

(19) Organización Mundial de la
Propiedad Intelectual
Oficina internacional



(43) Fecha de publicación internacional
28 de marzo de 2019 (28.03.2019)

WIPO | PCT

(10) Número de publicación internacional
WO 2019/059754 A9

(51) Clasificación internacional de patentes:

G06F 21/00 (2013.01) H04L 9/00 (2006.01)

(21) Número de la solicitud internacional:

PCT/MX2018/050013

(22) Fecha de presentación internacional:

03 de junio de 2018 (03.06.2018)

(25) Idioma de presentación:

español

(26) Idioma de publicación:

español

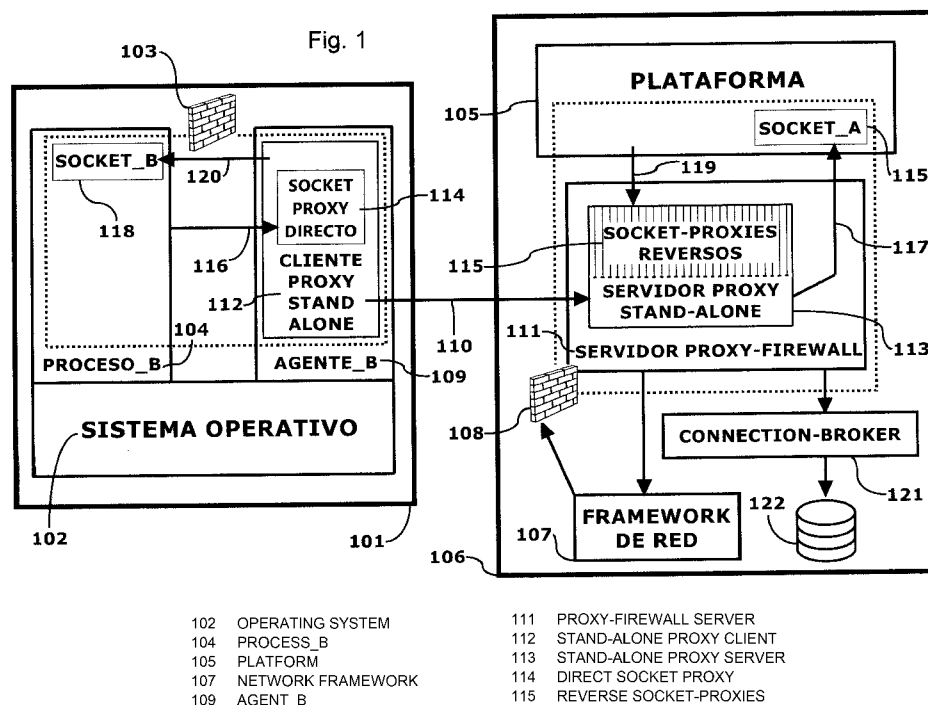
(72) Inventor; y

(71) Solicitante: **DE HOZ DIEGO, Jorge David** [MX/MX];
Circuito Hacienda de la Cadena 154, Haciendas del Cam-
pestre, Durango, 34194 (MX).

(81) Estados designados (a menos que se indique otra cosa,
para toda clase de protección nacional admisible): AE, AG,
AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW,
BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK,
DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM,
GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH,
KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA,
MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI,
NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU,
RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH,

(54) Title: SECURE COMMUNICATION METHOD AND SYSTEM USING NETWORK SOCKET PROXYING

(54) Título: MÉTODO Y SISTEMA DE COMUNICACIÓN SEGURA POR PROXIFICACIÓN DE SOCKETS DE RED



(57) Abstract: The present invention relates to the telecommunications sector and, in particular, to the field of telematics. More specifically, the invention describes a method and system for defining an improved network technology that can be used for communication and to operate secure communications between processes, based on the secure proxying of local area network sockets that may be between different devices. In particular, the system and method describe how socket proxies are established and managed between devices and how the security of said socket proxies in the local area thereof is enhanced and operated using security contexts. These contexts are configured based on privilege separation and local packet marking and filtering, and they allow applications to delegate all aspects relating to the security of the communications in the present invention.

TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) **Estados designados** (*a menos que se indique otra cosa, para toda clase de protección regional admisible*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), euroasiática (AM, AZ, BY, KG, KZ, RU, TJ, TM), europea (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declaraciones según la Regla 4.17:

- sobre la identidad del inventor (Regla 4.17(i))
- sobre el derecho del solicitante para solicitar y que le sea concedida una patente (Regla 4.17(ii))

Publicada:

- con informe de búsqueda internacional (Art. 21(3))
- antes de la expiración del plazo para modificar las reivindicaciones y para ser republicada si se reciben modificaciones (Regla 48.2(h))
- a pedido del solicitante, antes de la expiración del plazo a que se refiere en el Artículo 21.2)a)
- con información relativa a la autorización de rectificación de un error evidente según la Regla 91.3(b) (Regla 48.2(i))

(48) Fecha de publicación de esta versión corregida:

20 de junio de 2019 (20.06.2019)

(15) Información sobre la corrección:

véase la notificación del 20 de junio de 2019 (20.06.2019)

(57) Resumen: La presente invención se aplica al sector de las telecomunicaciones y está relacionada particularmente con el campo de la telemática. De forma más específica, la invención describe un método y sistema para definir una tecnología de red mejorada que permite comunicar y operar comunicaciones seguras entre procesos basándose en la utilización de proxificación segura de sockets de red locales que pueden ser entre diferentes dispositivos. Particularmente, el sistema y método presentados describen cómo establecer y gestionar proxificaciones entre dispositivos y cómo ampliar y administrar la seguridad de esas proxificaciones en el ámbito local de los mismos mediante contextos de seguridad. Estos contextos se configuran a partir de la separación de privilegios, marcado y filtrado de paquetes locales y permiten que las aplicaciones puedan delegar todos los aspectos relativos a la seguridad de las comunicaciones en la presente invención.

DESCRIPCIÓN

Título de la invención: Método y sistema de comunicación segura por proxificación de sockets de red

5 Sector técnico

La presente invención se refiere al sector de las telecomunicaciones y particularmente al área telemática.

Técnica anterior

10 El constante y creciente número de dispositivos y automatismos derivados del auge del Internet de las Cosas (IoT), junto con el despegue de la Industria 4.0 ponen de manifiesto la importancia de simplificar el control de las comunicaciones y la gestión de su seguridad. Anteriormente, el número de dispositivos desatendidos en cualquier tipo de proyectos era sensiblemente inferior, lo que suponía una interacción directa entre los dispositivos con el
15 usuario final o el operador de red en mayor o menor grado. Sin embargo, el elevado número de elementos que conforman estas redes y su previsión de crecimiento exponencial modifica este tipo de interacción directa, ya que resultaría inviable en la mayoría de las ocasiones y va en contra de una tendencia de automatización generalizada. Por ese motivo, se requiere que determinados aspectos de seguridad funcionen de forma sólida, estandarizada y confiable para
20 evitar que, por ejemplo, un supervisor de red tenga que analizar continuamente elemento a elemento el estatus de cada dispositivo a la par que se favorezca la escalabilidad e interconectividad potenciando las funcionalidades de esta tecnología.

La seguridad en las comunicaciones en el IoT, particularmente en dispositivos de alto nivel con capacidad de ser actualizados, actualmente se plantea como una prestación que debe
25 proporcionar el software/firmware de cada dispositivo en conjunción con la plataforma software que se encargue de gestionarlos, administrarlos y de la que derivan recursos y servicios. Independientemente del protocolo de comunicaciones empleado para ello, la solución actual utilizada implica embeber el software de autenticación y encriptación de las comunicaciones en el propio firmware del dispositivo o en la aplicación principal que éste utiliza. Este Dicho software
30 proporciona una protección criptográfica del canal de comunicaciones para evitar ataques a la seguridad en algún punto intermedio de la misma evitando, entre otros:

- Que la información que se intercambia pueda ser escuchada (eavesdropping).
- Que la información intercambiada pueda modificarse, replicarse, alterarse su orden o falsificarse sin que este hecho sea detectado por el receptor de la comunicación (tampering).

Comentado [David1]: Error evidente N° 1: Evitar repetición de palabra "Este". Se sugiere cambiar el término por "Dicho"

- Que el receptor con el que se comunica el emisor de la comunicación suplante la identidad del auténtico receptor de la misma, y este agente usurpador, a su vez, se haga pasar por el emisor de la comunicación de cara al receptor (man in the middle).

Este tipo de protección criptográfica se consigue, en la mayoría de las ocasiones, utilizando un framework de seguridad en el desarrollo de las aplicaciones que utilice Transport Layer Security (TLS) o algún protocolo de seguridad propietario. En ambos casos, una modificación que requiera actualizar algoritmos o determinadas configuraciones de los aspectos de seguridad, suele requerir actualizar la aplicación completa y, en muchos casos, también el firmware del dispositivo. Algunos ejemplos de este tipo de implementaciones se encuentran en dispositivos que utilicen COAP, MQTT, HTTP o HTTP/2 con seguridad.

Este tipo de diseños, en los que el protocolo de seguridad se encuentra integrado en el software, conduce a la necesidad de mantener activos los equipos de desarrollo de ese software para todas sus versiones y para cada modelo de dispositivo que actualmente haga uso de ello, por lo que las empresas fabricantes tienden a limitar en el tiempo el soporte que dan a los productos que comercializan animando a sustituir el dispositivo completo por su nueva versión en caso de quedarse obsoleto su firmware o la seguridad de la aplicación. En el medio plazo, esta idea frecuentemente no representa una alternativa real para el consumidor final; ya que debe contemplar como coste, no sólo el importe de los nuevos dispositivos, sino la sustitución de los antiguos por los nuevos, formación para los empleados y actualización de sus procesos internos entre otros.

La falta de previsión en estos costes por parte de las empresas fabricantes de tecnología IoT conduce a que los consumidores opten por seguir trabajando con dispositivos obsoletos en aspectos relativos a su seguridad. En muchas ocasiones, estos dispositivos no requieren manejar información sensible (ciertas redes de sensores) por lo que, aún en el caso de reconocerse esos dispositivos como vulnerables, el usuario final opta por mantenerlos ya que no le supone ningún riesgo real. Todo esto, conduce a un escenario en el que infraestructuras críticas ajenas acaban corriendo el riesgo de ser atacadas (DDoS) por botnets creadas a raíz precisamente de estos dispositivos IoT vulnerables que han logrado ser infectados con código malicioso, pero que aparentemente continúan funcionando con normalidad.

Resumen de la invención

La presente invención queda caracterizada por las reivindicaciones independientes y las reivindicaciones dependientes exponen ejemplos de realización de distintos aspectos de la invención.

El presente método y sistema presentado aportan soluciones para los problemas del estado del arte proponiendo un sistema de comunicación segura basado en proxificación que permita a

las aplicaciones de los dispositivos delegar las funcionalidades de seguridad en módulos de software independientes y stand-alone que se puedan actualizar sin afectar al resto del software de los dispositivos. Este objetivo, se consigue combinando una comunicación segura basada en proxificaciones entre dispositivos con un control de las comunicaciones locales empleando contextos de seguridad que permiten restringir el acceso a dichas proxificaciones en función del origen de la comunicación. Esta estrategia permite simplificar enormemente el desarrollo de las aplicaciones y el mantenimiento de la seguridad a medio y largo plazo, aun cuando el soporte de dichas aplicaciones deje de realizarse. Esto es posible gracias a que el sistema descrito en la invención no forma parte de las aplicaciones y puede actualizarse sin afectarlas.

En los dibujos y diagramas que se acompañan, se ilustran a modo de ejemplo distintos aspectos de posibles realizaciones de la presente invención. Para facilitar la comprensión de la invención, se adjunta la interpretación de los conceptos más relevantes utilizados en este documento:

- **Proxificar:** Acción de ofrecer el servicio o recurso provisto por un socket de red en otro socket de red diferente, pudiendo este socket ubicarse en un dispositivo remoto.
- **Proxificación o socket-proxy:** Resultado de la acción de proxificar en el que se define un nuevo socket de red, por medio del cual es posible acceder al socket origen de la proxificación de forma transparente y obtener los recursos o servicios ofrecidos por el mismo.
- **Servidor proxy-firewall:** (111) un módulo software que está configurado para:
 - o Recibir las solicitudes de establecimiento de sesión de otros dispositivos a través de un protocolo seguro por medio de un servidor proxy stand-alone.
 - o Configurar contextos de seguridad estrictos y amplios mediante llamadas al sistema al framework de red del sistema operativo del dispositivo_A.
 - o Procesar solicitudes de proxificación denominadas "socket-proxy" a través de sesiones establecidas o llamadas al sistema de procesos locales.
 - o Monitorizar los accesos a sockets locales mediante monitores de acceso.
 - o Gestionar las actualizaciones de seguridad de los dispositivos con los que mantenga una conexión.
 - o Establecer, mediante llamadas al sistema hacia el framework de red, protección de los sockets locales utilizados en las proxificaciones en el dispositivo_A.
- **Proxificación directa:** (114) Un tipo de proxificación que se produce cuando el socket-proxy resultado de la misma se genera en el dispositivo que inicia la sesión para ello, y ese socket-proxy ofrece recursos o servicios de un socket del dispositivo donde se dirige el establecimiento de sesión.
- **Proxificación reversa:** (115) Un tipo de proxificación que se produce cuando el socket-proxy resultado de la misma se genera en el dispositivo donde se dirige el establecimiento

de sesión para ello, y este socket-proxy ofrece recursos o servicios de un socket del dispositivo que inicia el establecimiento de sesión.

- **Espacio del usuario:** (418) Área de memoria donde operan los procesos de los usuarios del sistema operativo.

- **Espacio del kernel:** (419) Área de memoria donde operan exclusivamente los procesos del núcleo del sistema operativo.

- **Entorno del usuario:** Entorno del sistema operativo en el que aplican las configuraciones y privilegios asociados al usuario de dicho entorno.

- **Framework de red:** (107) (420) Conjunto de herramientas que interactúan con el sistema operativo o que forman parte de él y que permiten configurar el tratamiento que reciben los paquetes de las comunicaciones que se cursan del espacio usuario al espacio del kernel y viceversa.

- **Socket protegido:** Socket de red local cuyo acceso se encuentra restringido en base a reglas de filtrado del framework de red del sistema operativo del dispositivo.

- **Contexto de seguridad:** Entorno de red local caracterizado por unos permisos de acceso a ciertos sockets de red protegidos.

- **Contexto de seguridad estricto:** Contexto de seguridad aplicado entre un socket protegido y el espacio usuario de un usuario concreto del mismo sistema operativo. En la presente invención, se genera un contexto de seguridad estricto cuando se aplica un permiso_A_U en un socket-proxy del dispositivo_A por medio de reglas del framework de red. También se generan en el dispositivo_B cuando se aplican las reglas descritas en el perfil_B_A.

- **Contexto de seguridad amplio:** Contexto de seguridad aplicado entre un socket protegido y el espacio usuario de múltiples usuarios del mismo sistema operativo pertenecientes a un mismo grupo. En la presente invención, se genera un contexto de seguridad amplio cuando se aplica un permiso_A_G en un socket protegido.

- **Agente:** un módulo software que actúa como agente del servidor proxy-firewall desde un dispositivo remoto. Este software está configurado para:

- o Iniciar las solicitudes de establecimiento de sesión de comunicación hacia el servidor proxy-firewall a través de un protocolo seguro.
- o Realizar solicitudes de proxificación de sockets de forma directa e inversa.
- o Establecer, mediante llamadas al sistema hacia el framework de red, protección de los sockets locales utilizados en las proxificaciones en el dispositivo_B.

El agente se denomina de distintas maneras según sea el dispositivo en el que se encuentre:

- o Agente_A: (408) si se encuentra en un dispositivo_A. Sólo es necesario en el dispositivo_A si éste requiere realizar proxificaciones con otros dispositivos_A.
- o Agente_AF, si se encuentra en un dispositivo_AF.

Comentado [David2]:

Error evidente N° 2: Faltan los dos puntos “:”

- Agente_B: (109) si se encuentra en un dispositivo dispositivo_B.
- **Cliente proxy:** (112) (409) Módulo software complementario de cualquier agente que implementa el protocolo de comunicaciones seguro que se usa para el establecimiento de la sesión de comunicaciones y que se emplea para realizar las proxificaciones; tanto directas como inversas.
- **Servidor proxy:** (113) (407) Módulo software complementario del servidor proxy-firewall que implementa el protocolo de comunicación seguro que se usa para el establecimiento de sesión y que recibe y procesa las solicitudes de proxificaciones; tanto directas como inversas.
- **Connection-broker:** (121) (415) Módulo software empleado para mantener un registro y control sobre las proxificaciones realizadas y los accesos a las mismas en el dispositivo_A. También se ocupa de cerrar periódicamente los accesos inactivos.
- **Dispositivo_A:** (106) (405) Elemento hardware que alberga al menos: un servidor proxy-firewall, un connection-broker, un framework de red, una base de datos, un perfil_A y opcionalmente un agente_A. En la presente invención, si algún proceso requiere de dos dispositivos_A, el dispositivo sobre el que se describe el procedimiento principal se constituye como el dispositivo_A y el otro como dispositivo_AF.
- **Dispositivo_AF:** Elemento hardware equivalente en funcionalidades a un dispositivo_A. La razón de diferenciarlo como dispositivo_AF persigue el facilitar la explicación de las comunicaciones que se puedan producir entre dos dispositivos_A.
- **Dispositivo_B:** (101) Elemento hardware que alberga al menos un framework de red, un agente_B y un perfil_BA. En caso de albergar también un servidor proxy-firewall, la configuración de operación de este módulo software se incluye también en el perfil_BA, pues este servidor proxy-firewall tendrá funciones reducidas y orientadas a extender el control que el servidor proxy-firewall del dispositivo_A dispone de los módulos software del dispositivo_B, particularmente para procesos de actualización.
- **Base de datos:** ~~(112)~~ (414) Módulo software configurado para guardar y consultar distintos tipos de registros utilizados en la invención, entre los cuales se encuentran: registro_proxy, registro_acceso, registro_acceso_A, permiso_AU y el permiso_AG.
- **Perfil_BA:** Perfil de configuración del agente del dispositivo_B para configurar la sesión de conexión al servidor proxy-firewall del dispositivo_A. En este perfil también se encuentran las credenciales para conectarse al dispositivo_A utilizando la cuenta de usuario usuario_AB del sistema operativo del dispositivo_A. En este perfil también se encuentra Así mismo, también incluye información relativa a la capacidad del usuario_AB de solicitar proxificaciones en el dispositivo_A, siendo dicha capacidad validada por el servidor proxy-firewall tras cada solicitud de proxificación.
- **Perfil_A:** Perfil de configuración del servidor proxy_firewall y del connection-broker del dispositivo_A.

Comentado [David3]: Error evidente N° 3: La referencia numérica es incorrecta. La base de datos es la (122) y no la (112) y se sugiere corregirlo.

Comentado [David4]: Error evidente N° 4: Falta el guion bajo. Se sugiere cambiar el espacio por un guion bajo.

Comentado [David5]: Error evidente N° 5: "En este perfil también se encuentra" se repite dos líneas arriba. Se propone cambiarlo por "Así mismo, también incluye"

- **Perfil_UA_{AF}**: Perfil de configuración para el agente del dispositivo_A para configurar la sesión de conexión al proxy-firewall del dispositivo_AF. ~~En este perfil pertenece al~~ usuario_A y en él, también se encuentran las credenciales para conectarse al dispositivo_AF utilizando la cuenta de usuario usuario_A_{AF} del sistema operativo del dispositivo_AF. En este perfil también se encuentra información relativa a la capacidad del usuario_A_{AF} de solicitar proxificaciones en el dispositivo_AF, siendo dicha capacidad validada por el servidor proxy-firewall del dispositivo_AF tras cada solicitud de proxificación.
- **Usuario_A_B**: Usuario del sistema operativo del dispositivo A que es utilizado por el agente_B cuando éste se conecta al dispositivo_A utilizando las credenciales definidas en el perfil_B_A.
- **Usuario_A_{AF}**: Usuario del sistema operativo del dispositivo_AF que es utilizado por el agente_A cuando éste se conecta al dispositivo_A utilizando las credenciales definidas en el perfil_UA_{AF}.
- **Socket_A**: (115) Socket protegido del dispositivo_A que ofrece recursos o servicios a procesos locales y dispositivos remotos por medio de proxificaciones directas o reversas.
- **Socket_B**: (118) Socket protegido del dispositivo_B que ofrece recursos o servicios a dispositivos remotos a través de un socket-proxy reverso en el dispositivo_A.
- **Socket_AF**: Socket protegido del dispositivo_AF que ofrece recursos o servicios a procesos locales y dispositivos remotos por medio de proxificaciones directas o reversas.
- **Alias**: identificador referido a un socket concreto que sirve para ~~identificar-nombrar~~ el socket de un dispositivo. Es importante remarcar que el alias hace referencia al socket original que ofrece un servicio o recurso. Todas sus proxificaciones heredarán este alias, el cual puede ser de varios tipos:
 - o Alias_B: si el alias hace referencia a un socket_B de un proceso del dispositivo_B.
 - o Alias_A: si el alias hace referencia a un socket_A de un proceso del dispositivo_A.
- **Id_DispSocket**: Variable relativa al registro_socket de un socket protegido que identifica, junto con el alias del socket de dicho registro, el socket protegido del dispositivo. Esta variable hace referencia al id del dispositivo con el proceso que originó el socket que ofrece el recurso o servicio y puede almacenar tres tipos de identificadores:
 - o Un id_B, si el alias del registro_acceso refiere a un socket_B.
 - o Un id_AF, si el socket del registro_acceso comunica con un socket_proxy.
 - o Un id_A, si el alias del registro_acceso refiere a un socket_A.
- **Id_DispProxy**: Variable relativa al registro_socket de un socket protegido que identifica el id del dispositivo con el que se mantiene una sesión para poder acceder al socket referido por el alias y el id_DispSocket del registro. En el caso particular de que el registro_socket se refiera a un socket-proxy de otro socket-proxy, el id_DispSocket se referirá al id del dispositivo del segundo socket-proxy.

Comentado [David6]: Error evidente N° 6: Redacción incorrecta. Se sugiere cambiar "En este" por "Este"

Comentado [David7]: Error evidente N° 7: Repetición innecesaria de "identificar". Se sugiere "nombrar"

- **Tipo_socket:** Variable relativa al registro_socket de un socket protegido que identifica el tipo de socket al que hace referencia dicho registro pudiendo ser socket-proxy reverso, socket-proxy directo o socket_A. Los socket-proxy reversos y directos a su vez pueden ser también de tipo socket_A, socket_B y socket-proxy, indicando este último que el socket protegido es una proxificación de otro socket-proxy.
- **Registro_socket:** un tipo de registro del dispositivo_A relativo a un socket protegido. Se guarda en la base de datos del connection-broker y comprende de:
 - o Una variable denominada socket, que guarda el socket protegido al que hace referencia el presente registro_socket.
 - o Un alias del socket al que hace referencia el presente registro_socket.
 - o Un id_DispSocket relativo al dispositivo con el socket al que hace referencia el alias.
 - o Un id_DispProxy que identifica el dispositivo al que conduce la sesión en la que se genera el socket protegido.
 - o Un tipo_socket.
- **Con_Id:** Identificador del macado realizado en el tráfico de red relativo a la comunicación con un socket protegido. Este id determina la cuenta de usuario o el grupo de usuarios fuente de dicha comunicación registrada en la base de datos del connection-broker con destino ese socket protegido específico.
- **Id_Usuario:** Identificador de un usuario o de la cuenta relativa al mismo en el sistema operativo del dispositivo_A. Este identificador representa unívocamente al usuario y su cuenta de usuario, pudiéndose utilizar estos dos conceptos de forma indistinta en escenarios relativos a los contextos de seguridad del dispositivo_A.
- **Última_actividad:** Variable relativa a un registro_acceso de un usuario_A a un socket-proxy que define el último momento en el que hubo actividad relacionada con el socket protegido al que se hace referencia en el registro_acceso.
- **Registro_acceso:** un tipo de registro relativo a una cuenta de usuario del sistema operativo del dispositivo_A y que se genera por cada acceso concedido a un socket protegido de tipo socket-proxy directo o reverso, el cual comprende:
 - o Un con_Id, que guarda el id del acceso relativo a este registro.
 - o Una variable denominada ~~socket~~ que almacena el socket protegido al que se concede acceso y está formado por la dirección de red local y puerto.
 - o Un id_usuario, que guarda el id del usuario al que se concede el acceso.
 - o Una última_actividad que inicialmente guarda el momento en el que se crea el registro y posteriormente se actualiza periódicamente (si procede) por el monitor de accesos.
- **Registro_acceso_A:** un tipo de registro relativo a un grupo de usuarios del sistema operativo del dispositivo_A y que se genera por cada acceso concedido a un socket protegido de tipo socket_A, el cual comprende:

Comentado [David8]: Error evidente N° 8: Se sugieren comillas para diferenciar la variable

- Un con_Id, que guarda el id del acceso relativo a este registro.
- Una variable denominada `socket` que almacena el `socket_A`.
- Un id_Grupo, que guarda el id del usuario al que se concede el acceso.

Comentado [David9]: Error evidente N° 9: Se sugieren comillas para diferenciar la variable

- **Monitor de accesos:** Módulo software empleado para mantener un registro y control sobre las proxificaciones realizadas y los accesos a las mismas en el dispositivo_A. Este puede funcionar de dos maneras:

Comentado [David10]: Error evidente N° 10: Falta el punto y aparte

- Verificando periódicamente si existe tráfico relativo a un socket protegido, y en cuyo caso, actualiza el estatus de última actividad del registro_acceso pertinente.
- Actualizando periódicamente el estatus última actividad del registro_acceso de los accesos que monitoriza para garantizar la accesibilidad a los sockets protegidos respectivos mientras el proceso que solicitó los accesos siga en ejecución.

Cuando un monitor de accesos termina, bien al no haber actividad a lo largo de un periodo de tiempo determinado o porque el proceso responsable de las proxificaciones ha terminado, el monitor de accesos muere y deja de actualizar los registro_accesos de los que se encargaba. Esto provoca que el umbral de inactividad de esos registro_accesos se incremente y eventualmente sean cerrados por el connection-broker. Cuando esto ocurre, el servidor proxy-firewall es notificado para que borre las reglas de marcado pertinentes y cierre los accesos relativos a los registro_accesos borrados. Si los registro_accesos borrados se referían a una proxificación entre un dispositivo_A y un dispositivo_AF, el connection-broker también elimina el registro_proxy y el permiso_Au relativo a cada permiso_acceso.

Comentado [David11]: Error evidente N° 11: los registro_accesos no tienen una variable denominada "umbral_inactividad". (pag 7. Línea 33). Se sugiere cambiar "el umbral_inactividad" por "la inactividad"

- **Permiso_Au:** Este tipo de permiso sirve para otorgar acceso al usuario_A a un socket protegido que sea un socket-proxy a través de un contexto de seguridad estricto. Se guarda como registro en la base de datos del connection-broker y dicho registro comprende las siguientes variables:

- id_Usuario, que guarda el id del usuario del sistema operativo del dispositivo_A.
- Alias, que almacena el alias del socket protegido que se proxifica como socket-proxy en el dispositivo_A.
- id_DisSocket, que almacena el id del dispositivo con el socket_B ó socket_A relativo al alias del socket que ofrece el recurso a través del socket-proxy.

- **Permiso_Ag:** Este tipo de permiso sirve para otorgar el acceso de un grupo de usuarios del dispositivo_A a un socket_A a través de un contexto de seguridad amplio y comprende las siguientes variables:

- Id_grupo, que almacena el id del grupo de usuarios del sistema operativo del dispositivo_A al que se le otorga el permiso de acceso.
- Alias, que almacena el alias del socket_A.
- Socket, que guarda el valor del socket_A.

- **Sandbox-jail:** Conjunto de reglas de filtrado de paquetes que configuran una protección de acceso a los sockets protegidos. También restringe los destinos de las comunicaciones.
- **Acceso concedido a un socket protegido:** se denomina acceso concedido a un socket protegido, o simplemente "acceso concedido" a la configuración de un contexto de seguridad que habilite a un usuario de un sistema operativo acceder a un socket protegido local. Un solo acceso concedido habilita para cualquier número de conexiones concurrentes realizadas desde la cuenta de ese usuario al socket protegido al que hace referencia dicho acceso.
- **Aplicación final:** Aplicación utilizada por un usuario final.
- **Usuario final:** Persona física que requiere a través de una aplicación final un recurso o solicita un servicio de una plataforma.
- **Plataforma:** (105) (413) Módulo software que permite monitorizar y administrar la información y servicios que son ofrecidos por otros dispositivos de la red. Esta información y servicios que ofrece la plataforma, principalmente se obtienen a partir de los dispositivos_B. La función principal de la plataforma consiste en procesar estos datos y servicios para ponerlos a disposición de los usuarios finales en forma de recursos propios. Por otro lado, también puede actuar de forma directa sobre los elementos de la red que gestione para modificar su configuración o requerir actuaciones de los mismos utilizando los sockets protegidos locales a los que tenga acceso.
- **Host:** Máquina física o virtual con sistema operativo multiusuario que alberga una o varias plataformas.
- **Proxy:** (410) (422) Módulo software utilizado por cada tipo de protocolo empleado por las solicitudes de comunicación dirigidas a un socket protegido que no emplee el mismo tipo de protocolo de comunicación que el utilizado por el proceso origen de la solicitud. El proxy funciona como traductor entre protocolos para hacer la comunicación transparente y entre sus funciones figuran las siguientes:
 - o Autenticar las solicitudes de comunicación con credenciales de algún usuario del sistema operativo del dispositivo_A que disponga de permiso_A_U o que pertenezca a un grupo con permiso_A_G que le permita acceder al socket protegido.
 - o Habilitar un registro_acceso en caso de ser necesario mediante llamadas al sistema.
 - o Retirar las credenciales de autenticación de la comunicación para que ésta pueda enviarse al socket protegido y dicho proceso de autenticación sea transparente a este último;

Comentado [David12]: Error evidente N° 12: Falta el verbo

Comentado [David13]: Error evidente N° 13: Falta el determinante

Esta invención permite que los procesos establezcan comunicación entre ellos utilizando proxificaciones y sin tener que depender directamente de protocolos de seguridad. De este modo, los dos procesos que requieren comunicarse entre sí pueden realizar dicha comunicación de

forma local en un contexto de seguridad que permite que dicha comunicación sea además transparente gracias a la proxificación.

Contextos de seguridad

La comunicación entre los procesos mediante proxificación entre dos dispositivos consta de tres segmentos: El segmento intermedio que comunica ambos dispositivos y queda protegido por la sesión de comunicación entre el agente y el servidor proxy-firewall y los dos segmentos locales que comunican cada extremo de la proxificación con los procesos respectivos. Cada uno de estos dos segmentos, son vulnerables si no se aplica un contexto de seguridad, puesto que cualquier proceso puede acceder a los puertos que estén en escucha, lo cual resulta inseguro ya que las aplicaciones que se comunican delegaron toda responsabilidad de seguridad en el agente y el servidor proxy-firewall. Por ese motivo, la presente invención define el concepto de socket protegido y de contexto de seguridad.

Los contextos de seguridad consisten en reglas aplicadas por medio del framework de red de dos tipos: Reglas de marcado y reglas de filtrado de tráfico de red. Las reglas, a su vez, pueden ser generales o prioritarias y su combinación permite securizar las comunicaciones locales entre procesos y proxificaciones. Estos contextos son un tipo de sandbox-jail que impiden que se procesen las comunicaciones con sockets ajenos a los encontrados dentro de ese sandbox-jail.

La definición de las reglas de marcado y filtrado que definen los sockets protegidos y los contextos de seguridad utilizan dos conceptos: El socket local destino de la comunicación y el usuario o grupo cuyo proceso es origen de la comunicación de los paquetes de red. Este último dato se debe insertar en el tráfico de red mediante marcado de los paquetes ya que, en la mayoría de los protocolos de red, no existe una traslación de este dato al mismo. Por consiguiente, es necesario utilizar un framework de red que configure el sistema operativo para que introduzca ese dato en forma de marcado. Por otro lado, aclarar que el intentar identificar el origen de la comunicación por el socket origen resulta generalmente complicado debido a que las aplicaciones utilizan un socket diferente cada vez que solicitan establecer una comunicación. El contexto de seguridad, se define por tanto con el socket destino y el id de usuario origen o del grupo de usuarios origen de la comunicación. Esa información se traduce a un identificador de conexión al socket destino que se añade a cada paquete de tráfico de red para poder identificar el flujo de comunicación.

Una vez queda definido el procedimiento de marcado para las comunicaciones permitidas, el procedimiento de filtrado es sencillo: los sockets protegidos deben ser únicamente accesibles por comunicaciones marcadas por el sistema operativo en base a reglas del framework de red y dichas comunicaciones sólo pueden ser locales y pertenecientes a un rango de direcciones locales reservado y protegido. En este escenario, la vulnerabilidad de las comunicaciones queda

restringida a la posibilidad de que un usuario o proceso pueda obtener privilegios elevados que le permitan controlar el framework de red.

Protección del pool de direcciones reservadas

Un pool de direcciones locales del dispositivo_A queda reservada en el perfil_A específicamente para este sistema. Estas direcciones son protegidas al arrancar el sistema operativo del dispositivo_A por el framework de red a instancias del servidor proxy-firewall por medio de las siguientes reglas generales:

- Una regla general de marcado de paquetes que determine eliminar cualquier posible marcado de paquetes de red en cualquier comunicación producida entre sockets locales del dispositivo_A con direcciones locales pertenecientes al pool de direcciones locales reservado.
- Una regla general de marcado de paquetes que determine descartar cualquier paquete con origen y destino un socket local del dispositivo_A que carezca de un marcado siempre y cuando alguno de los sockets utilice una dirección de red local perteneciente al pool de direcciones locales reservado.
- Una regla general de filtrado de paquetes que determine descartar cualquier paquete con origen o destino una dirección local del pool de direcciones locales reservado para este método que tenga como origen o destino una dirección no local.

Además, se debe realizar la supresión, durante la secuencia de arranque del sistema operativo del dispositivo_A, de cada registro_acceso y permiso_Au de la base de datos relativo a los cada registro_socket cuyo tipo_socket sea socket-proxy y la supresión del propio registro_socket, ya que esos registro_accesos y registro_proxy son relativos a proxificaciones entre dispositivos_A y son considerados temporales.

Método de comunicación segura por proxificación de sockets de red:

A continuación, se expone el método que permite realizar comunicaciones seguras entre procesos por medio de proxificaciones de sockets de red. Inicialmente se establecen los sockets protegidos en el dispositivo_B y en el dispositivo_A y todo ello comprende las siguientes acciones:

- Durante el arranque del dispositivo_A, el servidor proxy-firewall ingresa unas reglas de marcado y filtrado a través del framework de red al sistema operativo del dispositivo_A para proteger el pool de direcciones asignado en el perfil_A. Estas reglas de carácter general comprenden:
 - o Eliminar cualquier posible marcado de paquetes de red generado por los procesos en cualquier comunicación producida entre sockets del dispositivo_A con direcciones de red pertenecientes al pool de direcciones locales reservado para este método.

Comentado [David14]: Error evidente N° 14: Falta la mayúscula

Comentado [David15]: Error evidente N° 15: La oración es muy larga y se sugiere dividir en dos.

Comentado [David16]: Error evidente N° 16: Falta la mayúscula

Comentado [David17]: Error evidente N° 17: Falta el punto y aparte

Comentado [David18]: Error evidente N° 18: Falta la mayúscula

Comentado [David19]: Error evidente N° 19: Falta el punto y aparte

Comentado [David20]: Error evidente N° 20: Falta la mayúscula

Comentado [David21]: Error evidente N° 21: Falta el punto y aparte

Comentado [David22]: Error evidente N° 22: La oración no está bien redactada. Se sugiere sustituir "los" por "cada"

- Descartar cualquier paquete con origen y destino un socket local del dispositivo_A que carezca de un marcado, siempre y cuando alguno de los sockets utilice una dirección de red local perteneciente al pool de direcciones reservado para este método.
- 5 ○ Durante el arranque del dispositivo_B, el agente_B ingresa al menos, una regla al framework de red del sistema operativo del dispositivo_B para proteger cada socket_B definido en el perfil_B_A. Esta regla de carácter general consiste en descartar cualquier paquete de red que tenga una dirección origen no local y el destino sea alguno de los socket_B.
- 10 ○ Ingresar, durante la secuencia de arranque del sistema operativo del dispositivo_A y a través de un framework de red de este último, una regla general de filtrado de paquetes por parte del servidor proxy-firewall que determine descartar cualquier paquete con origen o destino una dirección local del pool de direcciones locales reservado para este método que tenga como origen o destino una dirección no local.
- 15 Una vez se han establecido las reglas de protección del pool de direcciones en el dispositivo_A, se registran los socket_A en la base de datos que aún no lo estén y se aplican las reglas de marcado pertinentes que definan los contextos de seguridad amplios que correspondan en el dispositivo_A para conceder acceso permanente a los grupos de usuario del dispositivo_A que se definan en el perfil_A. Todo ello comprende de las siguientes acciones:
- 20 - El ingreso, si no existe, durante la secuencia de arranque del sistema operativo del dispositivo_A de un registro_socket en la base de datos del dispositivo_A relativo a cada socket_A del dispositivo_A recogido en el perfil_A conteniendo cada registro_socket información que comprende:
 - El id_DisSocket, que en este caso sería el id_A.
 - 25 ○ El id_DisProxy, que en este caso sería también igual al id_A.
 - El alias, que en este caso sería igual al empleado por el socket_A a registrar.
 - El socket, que en este caso sería igual al socket local utilizado por el proceso_A para el socket_A definido en el perfil_A.
 - El tipo_socket, que en este caso sería igual a "socket_A".
- 30 - El ingreso, si no existen, de los permisos_A_G y de los registro_acceso_A tal y como se contemplen en el perfil_A.
- El ingreso de unas reglas de marcado prioritarias respectivas a cada acceso de cada socket_A concedidas permanentemente a los grupos de usuarios del dispositivo_A según esté definida cada una en forma de permiso en la base de datos del connection-broker.
- 35 Tras esto, el servidor proxy-firewall verifica a través del connection-broker que en la base de datos no existan accesos en vigor relativos a registros_socket de tipo socket_AF. En caso de existir, borrará cada registro_acceso asociado a dicho registro_socket incluyendo el propio

registro_socket. Estos registros son relativos a proxificaciones entre el dispositivo_A y dispositivos_AF son consideradas temporales y durante el inicio del sistema operativo del dispositivo_A deben suprimirse.

El proceso continúa cuando el servidor proxy-firewall del dispositivo_A recibe una petición de establecimiento de sesión del agente_B para realizar unas proxificaciones, todo lo cual comprende las siguientes acciones:

- Establecer una sesión de comunicaciones segura que permita efectuar proxificaciones de sockets entre el dispositivo_A y el dispositivo_B, donde el primero utilice para ello un servidor proxy-firewall que albergue y el segundo utilice un agente_B que inicie el establecimiento de dicha sesión empleando un cliente proxy stand-alone que implemente un protocolo de comunicación segura usando unos datos de configuración y credenciales del usuario_AB definidos por su perfil_BA. De este modo, el dispositivo_B establece la sesión con el dispositivo_A utilizando la cuenta usuario_AB;
- Posteriormente, proceder con el establecimiento de las proxificaciones directas. Estas permiten acceder a cada socket_A requerido desde el dispositivo_B. Estas proxificaciones las solicita el agente_B según el perfil_BA y quedan protegidas por el protocolo seguro utilizado en la sesión de comunicaciones entre el agente_B y el servidor socket-proxy.
- Establecer las proxificaciones reversas. Estas permiten que cada socket_B sea accesible por las aplicaciones del dispositivo_A de forma local siempre y cuando el usuario del sistema operativo que las ejecuta tenga permiso para ello. Cada socket_B que debe proxificarse está definido también en el perfil_BA;
- Registrar las proxificaciones reversas en una base de datos de un connection-broker del dispositivo_A generando un registro_socket por cada una de ellas.

Una vez finalizado este proceso, los procesos del dispositivo_B pueden acceder a cada socket_A proxificado de forma local y los procesos del dispositivo_A que dispongan de permiso_AU pueden acceder a los socket-proxy reversos una vez soliciten el acceso al servidor socket-proxy del dispositivo_B y se configuren los contextos de seguridad restringidos-estrictos pertinentes.

Comentado [David23]:

Error evidente Nº 23: El término correcto definido arriba es "contextos de seguridad estrictos" (Pag.4 Línea 17). Se sugiere cambiar la palabra "restringidos" por "estrictos"

Breve descripción de las figuras:

- Figura 1: Ejemplo de realización de un dispositivo_B y un dispositivo_A en el que el agente_A y el servidor socket-proxy mantienen una sesión por la que se efectúan proxificaciones. A través de estas proxificaciones, el proceso_B y la plataforma pueden comunicarse entre sí.
- Figura 2: Ejemplo de establecimiento de una proxificación directa de un socket_A en un dispositivo_B y de una proxificación reversa de un socket_B en un dispositivo_A.

- Figura 3: Ejemplo de comunicacion del proceso_A con un socket de red protegido.
- Figura 4: Ejemplo de realización de un dispositivo_A. Ejemplo de URL usada en la petición de un recurso de un socket protegido a través de un proxy del dispositivo_A.
- Figura 5: Ejemplo de realización de un dispositivo_A. Ejemplo de URL usada en la petición de un recurso de un socket protegido a través de un proxy del dispositivo_A.
- Figura 6: Ejemplo de solicitud de proxificación directa de sockets protegidos de un dispositivo_AF por parte de un proceso_A de un dispositivo_A.

Comentado [David24]: Error evidente Nº 24: La descripción de la figura 4 corresponde a la figura 5 y la descripción de la figura 5 corresponde a la 4

Descripción detallada de la invención

A continuación, se describe la realización preferida y varios ejemplos de la realización de distintos aspectos de la presente invención haciendo referencia a las figuras adjuntas para ilustrar su funcionamiento en diferentes escenarios.

Realización preferida de la invención

Un ejemplo de la realización de la invención puede encontrarse en la Figura 1. Esta realización está orientada a un ecosistema IoT. En este escenario, un ejemplo de dispositivo_B (101) correspondería con todos aquellos dispositivos IoT del ecosistema que sean considerados nodos del mismo. En este ejemplo, los dispositivos pueden funcionar con firmwares o sistemas operativos (102) de tipo POSIX (Portable Operating System Interface). Un ejemplo de este tipo de sistemas operativos son los de la familia Linux o alguna de sus variantes. En todo caso, disponen de un kernel que puede hacer uso de un framework de red como Netfilter, que puede estar en el Kernel o ser cargable mediante módulos externos y mediante el cual se ingresan reglas de marcado y filtrado (103). Este dispositivo_B posee al menos un proceso_B (104) que desarrolla la funcionalidad principal del dispositivo y que requiere de comunicación con un software de gestión como lo puede ser una plataforma (105) de un dispositivo_A (106). En el ecosistema IoT que se presenta, un ejemplo de dispositivo_A correspondería con el host IoT al que se conectan los nodos IoT. En este ejemplo, el dispositivo_A cuenta con un sistema operativo también tipo POSIX y también de la familia Linux, pero esto es algo que no resulta imprescindible, ya que al ser sistemas operativos de tipo POSIX, la interoperabilidad entre ellos es factible. En todo caso, el sistema operativo del dispositivo_A, al ser también de la familia Linux en este ejemplo, también cuenta con un kernel con framework Netfilter (107) mediante el que también se ingresan reglas de marcado y filtrado (108). El sistema operativo del dispositivo_A, al ser utilizado en un host IoT, es necesario que sea multiusuario y con capacidad de separación de privilegios. De ese modo, cada dispositivo_B que se conecte al mismo podrá utilizar una cuenta de sistema operativo diferente del dispositivo_A. Esto permite a Netfilter diferenciar los flujos de comunicación y crear contextos de seguridad que extiendan la protección de las comunicaciones en el área local de cada dispositivo.

Comentado [David25]: Error evidente Nº 25: Falta el verbo

La seguridad en las comunicaciones entre el dispositivo_A y el dispositivo_B la proporciona un agente_B (109) del dispositivo_A que establece una sesión segura (110) con un servidor proxy-firewall (111) del dispositivo_B. Esta sesión segura, utiliza un protocolo de seguridad que se ejecuta entre un cliente proxy stand-alone (112) y un servidor proxy stand-alone (113). El primero forma parte del agente_B y el segundo, del servidor proxy-firewall. La sesión que establecen entre ellos permite realizar proxificaciones entre los dos dispositivos. Un ejemplo de este tipo de protocolos es SSH, el cual dispone de implementaciones que funcionan como cliente proxy stand-alone y servidor proxy stand-alone como las requeridas en este ejemplo. Algunos ejemplos de estas implementaciones son las de OpenSSH y Dropbear.

El agente_B, utiliza la sesión establecida para proxificar conexiones entre ambos dispositivos según esté especificado en el perfil_BA que utilice dicho agente para establecer la sesión. De este modo, y según la información contenida en dicho perfil, el agente_B crea los socket-proxies directos (114) en el dispositivo_B y socket-proxies reversos (115) en el dispositivo_A que correspondan. El establecimiento de proxificaciones, por tanto, puede realizarse en ambos sentidos, aunque la sesión entre el agente y el servidor proxy-firewall la inicie el agente_B.

Una vez establecidas las proxificaciones, tanto el proceso_AB como la plataforma IoT pueden comunicarse utilizando los socket-proxies. Si la comunicación la inicia el proceso_AB, utilizará el socket-proxy directo local (114) que genera el agente_B para comunicarse con el socket_A (115). Dicha comunicación segura tiene tres partes:

- La primera (116), es un segmento local que comprende la comunicación entre el proceso_B y el socket-proxy directo del agente_B. Este segmento de comunicación queda protegido por el contexto de seguridad estricto configurado por el agente_B al arrancar el sistema operativo del dispositivo_B por medio de Netfilter.
- La segunda, entre el dispositivo_B y el dispositivo_A, está encriptada por la sesión SSH (110).
- La tercera (117), entre el servidor proxy-firewall y el socket_A de la plataforma queda protegida por otro contexto de seguridad amplio configurado por el servidor proxy-firewall a través del Netfilter (107) del dispositivo_A.

Cuando la plataforma IoT requiere iniciar una comunicación con el socket_B(118) del proceso del dispositivo_B entonces utiliza uno de los socket-proxies reversos (115) generados en el dispositivo_A; el correspondiente al alias del socket_B y al id_B del dispositivo_B (id_DevSocket) al que se pretende acceder por parte de la plataforma IoT. Esta comunicación también consta de tres partes: la primera (119), entre la plataforma IoT y el socket-proxy reverso, queda protegida por el contexto de seguridad estricto configurado por Netfilter en el dispositivo_A. La segunda, entre el dispositivo_B y el dispositivo_A se encuentra protegida por la sesión segura (110) y finalmente la tercera (120), entre el agente y el socket_B(118) del proceso_B del dispositivo_B se encuentra protegida por el contexto de seguridad estricto del dispositivo_B.

Comentado [David26]: Error evidente N° 26: El término correcto es "proceso_B". Se sugiere cambiar la "A" por la "B". La Figura 1 en la que se basa este ejemplo no muestra ningún "proceso_A"

Comentado [David27]: Error evidente N° 27: El término correcto es "proceso_B". Se sugiere cambiar la "A" por la "B"

La gestión de todas las sesiones establecidas, socket-proxies asignados y permisos de acceso a los mismos para usuarios del sistema operativo del dispositivo_A, son manejados por el connection-broker(121) y su base de datos(122) a instancias del servidor proxy-firewall.

5 El acceso a los socket_B en el dispositivo_B o a los sockets protegidos del dispositivo_A se produce mediante unas reglas de marcado prioritarias que definen los respectivos contextos de seguridad estrictos. Las reglas de marcado en este ejemplo se aplican incluyendo el con_id en el campo TOS del protocolo IP. El marcado de tráfico siguiendo esta estrategia facilita la operación de Netfilter y la implementación del servidor proxy-firewall, pero limita a 255 contextos de seguridad distintos aplicables a cada socket protegido. Esta cifra, en el caso de los sockets
10 reversos, es suficiente debido a que los dispositivos IoT, suelen ser dispositivos de recursos limitados que no suelen estar preparados para permitir múltiples conexiones simultáneas: la información y los servicios que ofrece se presentan al usuario final o a la aplicación final a través de recursos propios de la plataforma a la que se conectan.

Los sockets_A, son también sockets protegidos y también requieren de permisos de acceso
15 para las cuentas de cada dispositivo_B con procesos que requieran conectarse. El número de contextos estrictos de seguridad necesario en este caso por cada socket_A sería tan grande como el número de dispositivos_B que requieran comunicación dicho socket, y en la mayoría de los proyectos IoT sí se superan los 255. Por ello, en este caso, se aplica un solo contexto de seguridad amplio a todo un grupo de usuarios del dispositivo_A de modo que la cuenta de usuario
20 del dispositivo_A que utiliza cada dispositivo_B al conectarse compartan el permiso_AG para así poder agrupar esas cuentas de usuario en un grupo con dicho permiso.

Ejemplo 1: Establecimiento de una proxificación directa de un socket_A en un dispositivo_B y de una proxificación reversa de un socket_B en un dispositivo_A

En la figura 2, se describe un diagrama de secuencia en el que, en una realización de la
25 invención, el dispositivo_B requiere proxificar dos puertos: un socket_B de forma directa y un socket_A de forma reversa. En este ejemplo intervienen un dispositivo_B con un agente_B (201) y un dispositivo_A con un servidor proxy-firewall (202), un connection-broker (203) y una base de datos (204). Este ejemplo es generalizable para establecer cualquier número de proxificaciones directas y/o inversas en la misma sesión sin necesidad de realizarse
30 recurrentemente.

En primer lugar, el agente_B inicia la sesión (205) en el servidor proxy-firewall utilizando la configuración definida en un perfil_BA. Durante el proceso de autenticación (206), el servidor proxy-firewall determina qué tipo de proxificaciones y el número de ellas son las que el agente_B puede realizar a partir del perfil_BA usado en el establecimiento de sesión. Una vez se realiza la
35 autenticación y se establece la sesión, el servidor proxy-firewall realiza un fork de sí mismo (207) en la cuenta usuario_AB. De este modo, se genera una nueva instancia del servidor proxy-firewall

como fork (208) en esa cuenta, lo que permite una separación de privilegios. Este fork solicita (209) al connection-broker la reserva de un socket libre para establecer el socket-proxy reverso para el socket_B. Así mismo, dicha solicitud también contiene un requerimiento de resolución del socket_A para obtener el socket protegido asignado a dicho socket_A. El connection-broker consulta (210) en la base de datos si existe el permiso_A_G para el socket_A e inserta o actualiza el registro_socket para el socket_B según el perfil_B_A. Si este registro_socket existía previamente, se verifica si también existe algún registro_acceso en vigor relativo a ese registro_socket. En caso de existir, se mantiene el valor de socket asignado en el registro_proxy. Si no existe, el valor de socket para el registro_proxy se actualiza con otro nuevo que sea local; perteneciente siempre y en todo caso al pool de direcciones reservadas para este método y que el socket se encuentre disponible. En caso de no existir el registro_proxy o en el caso de tener que generar un registro_acceso completo, el socket se elige aleatoriamente entre los que se encuentren libres en el pool de direcciones reservados. Una vez terminadas estas acciones, se procede con la respuesta (211). Esta respuesta incluye la resolución del socket_proxy (el valor del socket asignado en el registro_socket para el socket_B) y la resolución del socket_A (el valor asignado en el registro_socket relativo al socket_A) en forma de registro_sockets. Tras esto, se notifica al fork de ello (212) y éste a su vez hace lo propio con el agente_B (213). Éste último, cursa una solicitud (214) al fork para que inicie un socket protegido en escucha que corresponda al socket-proxy reverso (215) previamente asignado. Así mismo, el agente_B inicia en escucha un socket local protegido del dispositivo_B que corresponde al socket-proxy directo (216).

Comentado [David28]:

Error evidente N° 28: Referencia equivocada. La solicitud cursada es la (214). La (215) se explica a continuación.

Ejemplo 2: Solicitud de un recurso de un socket protegido del dispositivo A por un proceso A

A continuación, se describe a modo de ejemplo un procedimiento por el que se concede un acceso a un socket protegido del dispositivo_A al usuario del proceso_A para que éste pueda comunicarse con dicho socket protegido. Este procedimiento se ilustra con un diagrama de secuencia representado en la Figura 3. El registro_acceso y el contexto de seguridad se aplican en la cuenta de usuario local del dispositivo_A relativo al usuario_A. Los sockets protegidos nunca pueden ser accesibles directamente desde máquinas remotas y sus accesos siempre son locales y por usuarios dentro del contexto de seguridad. Esto es posible gracias a la separación de privilegios de sistemas POSIX que permite configurar los contextos de seguridad.

Para poder acceder al socket protegido (301), la cuenta usuario_A en la que se que ejecuta el proceso_A (302) que requiere el acceso debe disponer del permiso de usuario para ello en la base de datos (303) del connection-broker (304). El proceso_A realiza una solicitud (305) al servidor proxy-firewall (306) para realizar la resolución del socket protegido a partir del alias del socket y de su id_DisSocket y de ese modo saber a qué socket local dirigir la petición del recurso. Dicha solicitud (305) se produce en este ejemplo mediante una llamada al sistema. El origen de esta llamada no requiere del manejo de credenciales de usuario_A ante el servidor

proxy-firewall en la propia solicitud ya que se aprovecha la separación de privilegios del sistema tipo POSIX mediante la cual, el servidor proxy-firewall puede identificar unívocamente la cuenta de procedencia de dicha llamada. Esta solicitud se valida (307) por el connection-broker, el cual inicialmente realiza una consulta (308) en la base de datos para verificar con su respuesta (309) si el usuario_A tiene permiso. Si el permiso es de tipo permiso_AG no se requiere de la concesión de un acceso y se procede con la respuesta (310) al servidor proxy-firewall en la que se indica que se dispone de acceso y cuál es el socket protegido. Sin embargo, si el permiso es de tipo permiso_AU, el connection-broker debe de conceder antes el acceso para el usuario_A al socket protegido. Para ello, el connection-broker genera un registro_acceso (311) y lo registra en la base de datos (312). Una vez recibida la confirmación (313) de la base de datos de que el registro se guardó correctamente y de que no existía uno previo, se comunica (314) al servidor proxy-firewall de la necesidad de configurar el contexto de seguridad estricto para el usuario_A. El servidor proxy-firewall utiliza la información del registro_acceso para definir la regla de marcado necesaria que conceda el acceso de las comunicaciones del usuario_A al socket protegido. Estas reglas de marcado se insertan (315) en el sistema operativo por medio del framework de red (316). Si hubiese un registro previo válido, no es necesario generar otro registro_acceso para el usuario_A porque el usuario_A ya dispone de acceso. Tras recibir confirmación (317) de la inserción de las reglas (en caso de haber sido necesario), el servidor proxy-firewall instancia un monitor de acceso (318) para actualizar periódicamente el estatus de inactividad del acceso. En este ejemplo, la monitorización se hace a intervalos de 5 minutos. Este monitor de acceso, al igual que las reglas de marcado y que el registro_acceso sólo se lleva a cabo si no existía otro registro_acceso válido anterior.

Una vez se confirma que el usuario_A dispone de un acceso al socket protegido, El servidor proxy-firewall notifica ~~(310)~~ (319) al proceso_A cuál es el socket protegido al que solicitaba acceder para que pueda enviar la solicitud (320) del recurso al socket protegido y recibir su respuesta (321).

Comentado [David29]: Error evidente N° 29: Doble referencia. La correcta es sólo la (319). La referencia (310) se explicó arriba (Pag.18, Línea 6).

Ejemplo 3: Acceso a un socket-proxy reverso por un proceso remoto a través de un proxy local del dispositivo A que convierte peticiones del protocolo de aplicación https a peticiones http de forma transparente

En la figura 4 se muestra un ejemplo de realización del dispositivo_A que permite poder acceder (401) (402) (403) a los socket-proxies (404) en el dispositivo_A (405) desde un proceso en un dispositivo diferente al dispositivo_A a través de conexiones proxificadas (406) en sesiones (407) establecidas con dispositivos remotos. El primer caso (401) es un ejemplo de este tipo de comunicación en el que las aplicaciones que utilizan HTTPS acceden a recursos del socket_B pero sin requerir que el proceso del socket_B de este dispositivo_B tenga que contar con soporte para HTTPS, ni esquema de autenticación. Este acceso se realiza a través de un socket-proxy

reverso en el dispositivo_A de un socket_B de un dispositivo_B que tiene una sesión establecida con el dispositivo_A.

La comunicación como tal entre la aplicación y el proceso del dispositivo_B es transparente y se efectúa a través de un proxy intermedio (410) que además convierte comunicaciones que utilizan HTTPS en comunicaciones HTTP. En primer lugar, la aplicación final solicita un recurso del dispositivo_B ofrecido por un socket_B proxificado en el dispositivo_A y dirige esta petición (401) hacia el proxy intermedio. Este software, que convierte la petición (501) de HTTPS a HTTP, puede estar implementado en multitud de lenguajes de programación del lado del servidor (411); como son por ejemplo PHP, Node.js, Go, Python, etc. Para posibilitar un acceso al socket-proxy correspondiente, el proxy recibe la información del destino de la comunicación por medio de subdominios (502) y también recibe las credenciales que use la aplicación final para conseguir un acceso al socket proxy; estas últimas incluidas en encabezados de la petición. El servidor web (412) debe estar configurado para redireccionar todas las peticiones en todos los subdominios posibles del proxy hacia la URL principal del proxy intermedio, incluyendo los subdominios como parámetros. De los encabezados, el proxy intermedio obtiene la credencial que la aplicación final adjunta y que permiten al proxy intermedio validar la petición y solicitar acceso al socket_proxy. Las credenciales van protegidas por HTTPS al estar en los encabezados. La petición que recibe el proxy intermedio cuenta con los siguientes elementos:

- El id_DisSocket, que en este caso corresponde con el id_B (503) del socket_B al que conduce el socket-proxy reverso, y que permite identificarlo junto al alias.
- El alias (504) del socket_B que permite identificar el socket-proxy junto al id_DisSocket.
- El dominio (505) del proxy.
- El recurso (506) que se solicita al socket_B.

La protección de las comunicaciones en cualquiera de los subdominios dinámicos que se requieran es posible con un único certificado wildcard SSL basado en certificados X.509 que habilita la protección de HTTPS para el dominio principal del proxy y de todos los subdominios posibles. Por otro lado, la comunicación ~~extrema-extremo~~ a extremo también es transparente e independiente de otras comunicaciones que pueda tener la aplicación final con otros socket-proxy al necesitar cada uno un subdominio diferente. Las credenciales que van en los encabezados, son utilizadas por el proxy intermedio para validar el acceso al socket-proxy para la cuenta de usuario en la que se ejecuta el proxy intermedio según los permisos en la base de datos (414) para esa cuenta de usuario. Este acceso se puede solicitar directamente al connection-broker (415) a través del servidor proxy-firewall (416) o requerir antes una validación previa de la plataforma. Una vez realizada la conversión de HTTPS a HTTP y haberse habilitado el acceso al socket-proxy para la cuenta de usuario que ejecuta el proxy intermedio, éste suprime los encabezados con las credenciales para hacer transparente el proceso de autenticación y envía la petición web HTTP (417) hacia el socket-proxy que corresponda.

Comentado [David30]: Error evidente N° 30: El término correcto es "extremo a extremo". Se sugiere cambiar "extrema" por "extremo"

Esta petición, al pasar del espacio de usuario (418) al espacio del kernel (419) es marcada por el framework de red (420) si el acceso se concedió correctamente y se insertó la regla de marcado correspondiente. Posteriormente, esta petición es enrutada y dirigida (421) hacia el socket-proxy y será cursada o descartada según las reglas de filtrado, pudiendo únicamente ingresar de nuevo al espacio usuario si posee un marcado válido. Tanto las reglas de marcado como de filtrado se introducen en el framework de red del sistema operativo por iniciativa del servidor proxy-firewall siguiendo las indicaciones descritas en los ejemplos 1 y 2. Finalmente, la petición es recibida por el socket-proxy del servidor proxy stand-alone (422) y transmitida como comunicación proxificada hacia el dispositivo_B.

Comentado [David31]: Error evidente Nº 31: Falta la preposición "en"

Ejemplo 4: Acceso a un socket-proxy por un proceso remoto a través de un proxy intermedio del dispositivo_A en el que el proxy convierte las peticiones entrantes de la versión segura de un protocolo de aplicación a la versión insegura del mismo:

En este ejemplo, se generaliza el ejemplo anterior. Siguiendo el mismo esquema, el ejemplo se puede aplicar a otros protocolos del nivel de aplicación utilizados por aplicaciones de dispositivos_B que requieren de seguridad pero que no la implementan. Algunos ejemplos de estos protocolos pueden ser HTTP/2, MQTT y COAP. Todos ellos tienen también una versión segura de sí mismo y con un proxy que convierta de forma transparente peticiones seguras a peticiones inseguras se hace factible que los dispositivos_B utilicen la versión insegura del protocolo junto con el sistema expuesto en la presente invención para conseguir una comunicación segura extremo a extremo.

Ejemplo 5: Acceso a socket-proxies por una plataforma que recopila información de ellos, la procesa, almacena y luego la ofrece a las aplicaciones finales en forma de recursos propios.

Comentado [David32]: Error evidente Nº 32: Falta el pronombre "la"

La comunicación (424) representa una modalidad de operación muy común en ecosistemas IoT. En este caso, la aplicación final no requiere un acceso directo a los recursos de los sockets proxificados, sino a la información y servicios que la plataforma elabora a partir de esos recursos que obtiene de forma autónoma. La plataforma, para ello sí establece comunicaciones periódicas con los sockets protegidos a los que tiene permiso de acceder y de los que recopila información con la que luego elabora sus recursos y servicios propios, los cuales podría seguir ofreciendo aun cuando todas las sesiones (407) estuviesen caídas temporalmente. El modo de operar del sistema propuesto por la presente invención en este caso es análogo al del ejemplo 2. En particular, el proceso_A (302) es aquel que ejecute la plataforma para comunicarse con el socket local que requiera en cada momento. La plataforma debe de disponer para ello de una cuenta de usuario del dispositivo_A con permisos_A₀ suficientes para poder acceder a todos los socket-proxy relativos a los dispositivos configurados en la plataforma y de los que obtiene los recursos periódicamente. Por otro lado, la plataforma también puede ofrecer información necesaria a otros dispositivos por medio de sockets seguros de tipo socket_A (425).

ejemplo 6: Solicitud de acceso a sockets protegidos de un dispositivo AF por medio de proxificaciones directas y a instancia de un proceso A

Este ejemplo ilustra cómo el dispositivo_A, a instancias del proceso_A (601), solicita por medio del servidor proxy-firewall (602) y del agente_A (603) una proxificación directa (604), (426) de unos sockets protegidos de un dispositivo_AF (605). Esto permite que el proceso_A pueda acceder de forma segura a recursos de sockets protegidos en ese dispositivo remoto. Para ello, el proceso_A solicita (606) al servidor ~~socket-proxy~~ proxy-firewall una resolución de sockets remotos suministrando el alias, el id_DispSocket y el id_DisProxy de cada socket protegido al que se desea acceder. En este caso, el id_DisSocket es distinto al id_DisProxy y por ese motivo, el servidor proxy-firewall determina que se requieren proxificar de forma directa esos proxies desde otros dispositivos_AF. En el ejemplo, todos los sockets a proxificar provienen del mismo dispositivo, pero no necesariamente tiene por qué ser así en otros casos.

La solicitud de proxificación directa realizada por el proceso_A al servidor proxy-firewall se realiza mediante una llamada al sistema que instancia el servidor proxy-firewall como proceso de ese entorno de usuario. Para proceder con la proxificación directa, el servidor proxy-firewall inicialmente solicita (607) al connection-broker (608) que reserve y registre (609) en la base de datos (610), unos sockets locales libres que pertenezcan al pool de direcciones protegido para asignar los socket-proxy directos. La reserva de estos sockets implica además la inserción de un registro_socket de tipo socket-proxy, y un registro_acceso si dispone de permisos_AU para cada uno por cada socket reservado. Una vez se confirma la reserva (611) de esos sockets locales, el servidor proxy-firewall recibe en la respuesta (612) del connection-broker los sockets reservados y solicita (613) al agente_A el establecimiento (614) de una sesión con el dispositivo_AF para resolver los sockets protegidos que se requieren proxificar. El agente_A utilizará un perfil_UA_AF relativo al usuario_A que se autentica en la cuenta usuario_AF del dispositivo_AF y que es suministrado por el servidor socket-proxy. Una vez se establece la sesión usando dichas credenciales, el dispositivo_AF realiza las acciones pertinentes (615) descritas en el ejemplo 7 para procesar la solicitud de proxificación recibida. Una vez el dispositivo_AF responde (616) al agente_A con los sockets protegidos resueltos, el agente_A notifica (617) al servidor proxy-firewall y procede a establecer las proxificaciones directas (604).

Aunque los socket-proxy directos ya se encuentran establecidos, el acceso a los mismos no es posible al ser sockets protegidos. Para poder conceder acceso al usuario_A del proceso_A es necesario aplicar los permisos_AU respectivos de la base de datos. Estos permisos, son traducidos por el servidor proxy-firewall en reglas de marcado que se insertan (618) a través del framework de red. Una vez se confirma (619) este ingreso, el servidor proxy-firewall responde (620) la petición del proceso_A con los sockets-proxy directos resueltos para que pueda conectarse a ellos. Paralelamente, el servidor proxy-firewall instancia un monitor de accesos (621) que mantiene activos los accesos mientras esté funcionando el proceso_A.

Comentado [David33]: Error evidente N° 33: Según la referencia (606), el término correcto es "servidor proxy-firewall" y se sugiere reemplazar "socket-proxy" por "proxy-firewall"

Ejemplo 7: Ejemplo de comunicación en el que un dispositivo_AF consigue acceso a unos sockets protegidos de un dispositivo_A por proxificación directa

Este ejemplo está vinculado con el anterior: en el ejemplo 6, un dispositivo_A se conectaba con un dispositivo_AF para requerir proxificaciones directas y este ejemplo ilustra lo que ocurre en ese dispositivo_AF. Sin embargo, para mantener la coherencia descriptiva, en el presente ejemplo ése dispositivo_AF será referido como dispositivo_A y el dispositivo que se conecta al dispositivo_A será referido como dispositivo_AF. Una vez aclarado este detalle, se procede a exponer cómo se concede en este ejemplo la proxificación directa de sockets protegidos a un dispositivo_AF.

La primera parte del proceso es similar a la descrita en el ejemplo 1, con la salvedad de que el dispositivo que inicia la sesión esta vez es un dispositivo_AF (en vez de ser un dispositivo_B) y que la solicitud consiste únicamente en proxificaciones directas. Para ello, El dispositivo_AF dispone de un agente_AF y de un perfil de conexión para conectarse al dispositivo_A que le permite establecer una sesión en la que se solicita se solicita la resolución de los sockets protegidos requeridos por el dispositivo_AF. El dispositivo_AF utiliza las credenciales del perfil_UA_{AF} para conectarse a la cuenta del usuario_A_{AF} en el sistema operativo del dispositivo_A. Esta cuenta debe tener permiso de acceso a los sockets protegidos que se quieren proxificar. Una vez autenticadas las credenciales por el servidor proxy-firewall del dispositivo_A, este software se instancia con un fork de sí mismo bajo la cuenta de usuario_A_{AF}.

Una vez llegado a este punto, el fork del servidor proxy-firewall debe evaluar si el usuario_A_{AF} dispone de los permisos necesarios para acceder a los sockets protegidos requeridos por el agente_AF. El procedimiento de comprobación, marcado y asignación de registro_acceso es igual al descrito en el ejemplo 2 y que comprende el diagrama de secuencia de la figura 3 desde (307) hasta (319). Una vez confirmados los permisos y en su caso, concedidos los accesos a los socket-proxies requeridos y definidas e insertadas las reglas de marcado pertinentes, se devuelven los sockets resueltos al agente_AF para que pueda proceder con el inicio de la proxificación directa. Para garantizar que la cuenta de usuario_A_{AF} disponga del acceso constantemente mientras dure la sesión, se instancia un monitor de accesos que periódicamente actualice todos los accesos a los sockets-proxy aun si no ha habido tráfico. Esto lo realizará mientras dure la sesión entre el dispositivo_AF y el dispositivo_A.

Ejemplo 8: Establecimiento de una proxificación de sockets protegidos desde un dispositivo_A hacia otro dispositivo_AF de forma reversa a instancias del proceso_A del dispositivo_A

Existe la posibilidad de que una plataforma de un dispositivo_AF requiera acceso a recursos proxificados de dispositivos_B con los que no mantiene sesión o que esos dispositivos_B requieran de recursos de un dispositivo_A foráneo con el que tampoco mantienen sesión. En

este caso, el dispositivo_A que mantiene sesión con los dispositivos_B, toma la iniciativa e inicia un proceso de proxificación reversa de puertos protegidos locales hacia el dispositivo_AF; bien por instancias de algún proceso o por algún tipo de configuración de red establecido que así lo indique. Con la proxificación reversa, el dispositivo_A pone a disposición del dispositivo_AF los sockets protegidos que proxifique. Para ello, el agente_A y establece una sesión de comunicación con el dispositivo_AF utilizando un perfil_UA_{AF} suministrado por el servidor proxy-firewall a instancias de un proceso_A.

En este ejemplo, el proceso_A solicita la proxificación reversa al servidor socket-proxy del dispositivo_A. Al igual que en el ejemplo 2, es imprescindible que el servidor socket-proxy del dispositivo_A se asegure de que el usuario_A que ejecuta el proceso_A cuente con los permisos necesarios para poder acceder a los sockets protegidos que solicita proxificar y que, cuando corresponda, conceda y registre los accesos pertinentes y elabore e inserte las reglas de marcado oportunas. El procedimiento es idéntico al descrito en el ejemplo 2 en el diagrama de secuencias que comprende desde el (307) hasta el (319). También se requiere que el servidor proxy-firewall instancie un ~~control de conexiones~~ monitor de accesos que verifique que los accesos permanezcan abiertos mientras se esté ejecutando el proceso_A actualizando periódicamente el estatus de cada registro_acceso para evitar que se cierren automáticamente por inactividad. Tras esto, el servidor proxy-firewall comunica al servidor proxy-firewall los sockets protegidos resueltos y en este momento, cualquier proceso del usuario_A_{AF} puede acceder a los sockets protegidos del dispositivo_A a través de los socket-proxy reversos establecidos. Dado que el servidor proxy-firewall se ha instanciado desde esa cuenta de usuario puede lanzar el agente_A utilizando el perfil_UA_{AF} que inicie el procedimiento de proxificación reversa de los sockets protegidos ya accesibles.

Comentado [David34]: Error evidente Nº 34: Falta el determinante "el"

Comentado [David35]: Error evidente Nº 35: Según la figura 3, el término correcto es "monitor de accesos" (318). Se sugiere cambiar "control de conexiones" por "monitor de accesos"

Ejemplo 9: Proxificación de sockets protegidos desde un dispositivo AF hacia un dispositivo A de forma reversa a instancias del agente AF del dispositivo AF

Este ejemplo 9 está muy vinculado con el ejemplo anterior, el 8. En el ejemplo 8, un dispositivo_A a instancias de un proceso_A inicia el establecimiento de proxificaciones reversas hacia un dispositivo_AF. En el presente ejemplo, se muestra lo que ocurre en ese dispositivo_AF, en el que, de nuevo, y en aras de conservar la coherencia descriptiva, ese dispositivo_AF será referido como dispositivo_A y viceversa.

Por consiguiente, el servidor proxy-firewall del dispositivo_A recibe una solicitud de establecimiento de sesión de un dispositivo_AF que utiliza un perfil_UA_{AF} en el que se solicita la realización de una proxificación reversa de determinados sockets protegidos del dispositivo_AF. Esta solicitud fuerza al servidor proxy-firewall a instanciarse como fork_A_{AF} en la cuenta de usuario_A_{AF} cuyas credenciales utilizó el agente_AF para conectarse en el dispositivo_A. El fork_A_{AF} requiere solicitar al connection-broker una la reserva de sockets protegidos locales en el dispositivo_A para poder realizar la proxificación reversa. Este procedimiento es similar al

seguido en el ejemplo 1 por el dispositivo_A en el que el dispositivo_A debe reservar unos sockets locales para las proxificaciones reversas que solicita realizar el dispositivo_B. La principal diferencia, es que estos sockets locales protegidos resultado de la proxificación son nuevos y no existen permisos relativos en la base de datos que den acceso a ellos de algún modo. Por ello, el connection-broker, además de reservar esos sockets, debe insertar registros_proxy y permisos_AU por cada socket-proxy reverso que habiliten al usuario_AAF de acceso a cada socket resultado de la proxificación de forma similar a como se realiza en el ejemplo 6 (607), (609), (611) y (612). Esta reserva, no obstante, se realiza únicamente para socket-proxies reversos.

Una vez reservados los sockets e insertados los permisos correspondientes, se deben generar las reglas de marcado pertinentes e insertarlas en el sistema para que el usuario_AAF tenga acceso a esos sockets protegidos. El servidor proxy-firewall instancia un monitor de accesos para garantizar que los registro_accesos relativos a cada proxificación mantengan una umbral de inactividad bajo-baja que impida su cierre automático mientras el proceso_A siga en ejecución. El servidor proxy-firewall, al estar instanciado desde la cuenta del usuario_AAF, forma parte del contexto de seguridad configurado al insertar las reglas de marcado y, por tanto, esta instancia dispone de acceso a los socket protegidos reservados en este ejemplo para la proxificación reversa. Una vez terminadas estas acciones, el servidor proxy-firewall realiza la proxificación reversa solicitada por el agente_AF y utilizando los sockets reservados para ello.

Ejemplo 10: Actualización remota de elementos software:

Una de las principales ventajas de la invención es la disociación de las funcionalidades de seguridad de las aplicaciones de los dispositivos_B y su correspondiente simplificación en su diseño y mantenimiento. En los dispositivos_A, los procesos que hacen uso del sistema propuesto por la invención, no requieren tampoco de frameworks de seguridad, pero sí deben incorporar en su programación el método descrito en este documento para poder comunicarse con los sockets protegidos. No obstante, la interfaz de comunicación se basa en llamadas al sistema y marcate de paquetes, por lo que es independiente de los protocolos de seguridad y por consiguiente, es factible actualizar en el dispositivo_A el sistema descrito en la presente invención también sin afectar a las aplicaciones del dispositivo_A.

En dispositivos_A y dispositivos_B similares a los propuestos en la realización preferida de la invención, es posible actualizar el agente del dispositivo_B desde el dispositivo_A sin afectar al resto del software. Para ello, es necesario que el dispositivo_A-B cuente con una versión simplificada del proxy-firewall que disponga del servidor proxy stand-alone y de una configuración adecuada en el perfil_BA que permita al dispositivo_A utilizar un socket-proxy reverso para actualizar el agente del dispositivo_B. Esta estrategia puede generalizarse para proporcionar servicios de actualización a otros elementos software ajenos al sistema descrito por la invención.

En el caso particular de la actualización del cliente proxy stand-alone, el proceso comprende:

Comentado [David36]: Error evidente N° 36: Expresión confusa. Se sugiere cambiar "mantengan un umbral de inactividad bajo" por "mantengan una inactividad baja"

Comentado [David37]: Error evidente N° 37: Se sugiere cambiar la "A" por la "B". En la descripción de Dispositivo_B (Pag.5 Línea 21) se especifica que es éste dispositivo el que cuenta con una versión del proxy-firewall reducida y que utiliza el perfil_BA.

- Solicitar, a través de la sesión de comunicaciones establecida entre el agente_B y el servidor socket-proxy del inicio de un procedimiento de la actualización del cliente proxy stand-alone del agente_B por iniciativa del servidor proxy-firewall.
- Recibir, por parte del agente_B, la nueva versión del cliente proxy y actualizar el perfil_B_A.
- Probar, por parte del agente_B, la nueva versión del cliente proxy con el perfil_B_A actualizado estableciendo una nueva sesión de comunicaciones.
- En caso de establecerse correctamente esta nueva sesión, eliminar la versión anterior del cliente proxy y reemplazarlo con la nueva versión.

REIVINDICACIONES

[Reivindicación 1] Un método de comunicación segura por proxificación de sockets de red que comprende:

5 establecer una sesión (110) de comunicaciones segura que permita efectuar proxificaciones de sockets entre dos dispositivos denominados "dispositivo_A" (106) y "dispositivo_B" (101), donde el primero utilice para ello un servidor proxy-firewall (111) que albergue y el segundo utilice un agente del servidor proxy-firewall denominado "agente_B" (109) que inicie el establecimiento de dicha sesión empleando un cliente proxy stand-alone (112) que implemente un protocolo de comunicación segura usando unos datos de configuración y credenciales de una cuenta de usuario denominada "usuario_AB" definidos por un perfil de configuración del dispositivo_B denominado "perfil_BA" que le permiten conectarse a la cuenta del usuario_A del dispositivo_A;

15 establecer unas proxificaciones denominadas "proxificaciones directas" (114) si estas se producen desde los sockets de aplicaciones que se encuentran escuchando en el dispositivo_A, como lo son los denominados "socket_A" (115), y según se describa en el perfil_BA;

20 establecer unas proxificaciones denominadas "proxificaciones reversas" (115) si estas se producen desde los sockets de las aplicaciones que se encuentran escuchando en el dispositivo_B denominados "socket_B" (118) y según se describa en el perfil_BA;

25 proteger a instancias del servidor proxy-firewall del dispositivo_A las comunicaciones de los socket_A y las proxificaciones reversas en el dispositivo_A mediante la configuración de contextos de seguridad configurados con reglas de marcado y filtrado de paquetes de red de esas comunicaciones locales, utilizando como parámetro delimitador del contexto el usuario del proceso del sistema operativo del dispositivo_A origen de cada comunicación local; y

30 proteger a instancias del agente del dispositivo_B las comunicaciones de los socket_B y las proxificaciones directas en el dispositivo_B mediante la configuración de contextos de seguridad configurados con reglas de marcado y filtrado de paquetes de red de esas comunicaciones locales, utilizando como parámetro delimitador de dichos contextos el usuario del proceso del sistema operativo del dispositivo_B origen de cada comunicación local.

[Reivindicación 2] El método según la reivindicación 1, donde se establecen unos contextos de seguridad para las proxificaciones directas y las reversas, además comprende:

35 registrar las proxificaciones reversas en una base de datos (122) de un connection-broker (121) del dispositivo_A como un registro denominado "registro_socket" donde se guarda la información relativa de cada proxificación reversa que comprende:

- Una variable denominada "id_DispSocket" que almacena el id de dispositivo con el socket a proxificar y que según esta reivindicación correspondería con el id del dispositivo_B denominado "id_B";
- Una variable denominada "id_DispProxy" y que identifica el dispositivo con el que mantiene la sesión que posibilita el socket protegido al que hace referencia este registro_socket y que según esta reivindicación correspondería con el id_B;
- una variable denominada "alias" y que en esta reivindicación guardaría el alias del socket a proxificar del id_DispSocket;
- una variable denominada "socket" y que según esta reivindicación guardaría el socket local libre reservado por el sistema operativo del dispositivo_A para la proxificación y que dispone de un valor de dirección de red perteneciente a un pool de direcciones de red locales del dispositivo_A reservados para este método; y
- una variable denominada "tipo_socket" que determina la clase de socket a la que hace referencia este registro, y que según esta reivindicación sería "socket_B";

el ingreso, durante la secuencia de arranque del sistema operativo del dispositivo_A, de una regla general de marcado de paquetes a través de su framework de red (107) por parte del servidor proxy-firewall que determine eliminar cualquier posible marcado de paquetes de red en cualquier comunicación producida entre sockets locales del dispositivo_A con direcciones locales pertenecientes al pool de direcciones locales reservado para este método;

el ingreso, durante la secuencia de arranque del sistema operativo del dispositivo_A, de una regla general de marcado de paquetes por parte del servidor proxy-firewall que determine descartar cualquier paquete con origen y destino un socket local del dispositivo_A que carezca de un marcado siempre y cuando alguno de los sockets utilice una dirección de red local perteneciente al pool de direcciones locales reservado para este método;

el ingreso, durante la secuencia de arranque del sistema operativo (102) del dispositivo_B y a través del framework de red de este último, de una regla general de filtrado por cada socket_B y proxificación directa en el dispositivo_B que determine el descarte de paquetes entrantes con destino ese socket_B con origen una dirección de red no local;

el ingreso, durante la secuencia de arranque del sistema operativo del dispositivo_A y a través de un framework de red (107) de este último, de una regla general de filtrado de paquetes por parte del servidor proxy-firewall que determine descartar cualquier paquete con origen o destino una dirección local del pool de direcciones locales reservado para este método que tenga como origen o destino una dirección no local;

el ingreso, si no existe, durante la secuencia de arranque del sistema operativo del dispositivo_A, de un registro_socket en la base de datos del dispositivo_A relativo a cada socket_A del dispositivo_A recogido en el perfil de configuración del socket-proxy y del connection-broker denominado "perfil_A", conteniendo cada registro_socket información que comprende:

- el id_DispSocket, que en esta reivindicación sería igual al id del dispositivo_A denominado "id_A";
- el id_DispProxy, que en esta reivindicación sería también igual al id_A;
- el alias, que en esta reivindicación sería igual al empleado por el socket_A a registrar;
- el socket, que en esta reivindicación sería igual al es igual al socket local utilizado por el proceso_A para el socket_A definido en el perfil_A; y
- el tipo_socket, que en esta reivindicación sería igual a "socket_A";

el ingreso, si no existe, durante la secuencia de arranque del sistema operativo del dispositivo_A, de al menos un registro denominado "registro_acceso_A" en la base de datos del dispositivo_A relativo a cada socket_A del dispositivo_A recogido en el perfil_A, conteniendo cada registro_acceso_A información que comprende:

- una variable denominada "con_Id" que almacena el identificador del acceso con el mismo nombre,
- una variable denominada "socket" que almacena el socket_A, y
- una variable denominada "id_Grupo" que almacena el id del grupo del usuario del sistema operativo del dispositivo_A correspondiente al usuario_A del proceso_A origen de la petición de comunicación al que se le ha concedido acceso;

el ingreso, durante la secuencia de arranque del sistema operativo del dispositivo_A y a través del connection-broker (121) a instancias del servidor proxy-firewall y según la configuración marcada en el perfil_A, de unos permisos_AG en la base (122) de datos que comprenden:

- el id_Grupo, que almacena el id del grupo de usuarios del sistema operativo del dispositivo_A al que se otorga el permiso de acceso;
- el alias, que almacena el alias del socket_A;
- el socket, que almacena el valor del socket_A;

el ingreso, durante la secuencia de arranque del sistema operativo del dispositivo_A y a través del framework de red (107) a instancias del servidor proxy-firewall de unas reglas de marcado prioritarias sobre las generales y que son respectivas a los permisos_AG registrados en esta reivindicación para cada socket_A para conceder accesos

permanentemente a los grupos de usuarios del dispositivo_A relativos a los anteriores permisos según se defina en el perfil_A; y

la supresión, durante la secuencia de arranque del sistema operativo del dispositivo_A, de cada registro_acceso y de sus permisos respectivos de la base de datos relativos a los registro_socket cuyo tipo_socket sea socket-proxy y la supresión del propio registro_socket.

[Reivindicación 3] El método según las reivindicaciones anteriores, donde se establece una comunicación segura entre sockets de red, en la que un recurso de red que es ofrecido por el socket_B de un proceso denominado "proceso_B" (118) del dispositivo_B es solicitado por un proceso denominado "proceso_A" (302) de un usuario denominado "usuario_A" del sistema operativo del dispositivo_A y cuyo recurso es accesible a través de un socket-proxy reverso (301), además comprende:

Solicitar (305) por parte del proceso_A al servidor proxy-firewall mediante una llamada al sistema la resolución del socket protegido a partir del id_B y del alias del socket_B para obtener el socket-proxy reverso local por el que poder acceder al socket_B;

solicitar (307) al connection-broker de parte del servidor proxy-firewall la comprobación de existencia de un permiso de acceso del usuario_A denominado "permiso_Au" que lo habilite para acceder al socket-proxy reverso requerido, la solicitud de resolución del socket-proxy reverso y la concesión de un acceso al socket-proxy reverso denominado "registro_acceso";

comprobar (308) por parte del connection-broker en la base de datos la existencia del permiso_Au, realizar la resolución del socket-proxy reverso;

generar (311) el registro_acceso, si aún no existe en la base de datos, en el que se asigna por parte del connection-broker un identificador de acceso de comunicación al socket-proxy reverso denominado "con_id" relativo al usuario_A y al socket protegido;

guardar (312), o actualizar si procede, en la base de datos por parte de connection-broker el registro_acceso generado, el cual comprende:

- el con_id relativo del acceso concedido,
- el socket al que se concede el acceso y está formado por la dirección de red local y el puerto,
- una variable denominada "id_usuario" que almacena el id del usuario del sistema operativo del dispositivo_A correspondiente al usuario_A del proceso_A origen de la petición de comunicación al que se le ha concedido acceso; y
- una variable denominada "última_actividad" que guarda el momento en el que se le ha concedido el acceso que se actualizará por un monitor de acceso reflejando su estatus de actividad;

habilitar (315) una regla específica prioritaria sobre las reglas generales de marcado de paquetes para el sistema operativo del dispositivo_A por parte del servidor proxy-firewall que determine que a todos paquetes de comunicaciones locales entre el usuario_A que ejecuta el proceso_A y el socket-proxy reverso se les incluya el identificador del acceso asignado con_Id definido en el registro_acceso correspondiente al socket-proxy reverso y al usuario_A;

instanciar (318) por parte del connection-broker un monitor de accesos que verifique si hay tráfico por el acceso hacia el socket-proxy reverso periódicamente y actualice el valor de ~~ultima_inactividad~~; y

proceder con la comunicación (320) entre proceso_A y el socket_B que ofrece el recurso a través del socket-proxy reverso existente en el dispositivo_A;

[Reivindicación 4] El método según las reivindicaciones anteriores, donde el proceso_A del usuario_A en el dispositivo_A solicita la proxificación de forma directa en el dispositivo_A de varios sockets protegidos de un dispositivo_A foráneo denominado "dispositivo_AF", además comprende:

solicitar (606) por parte del proceso_A al servidor proxy-firewall una resolución de ~~las proxificaciones directas~~ a realizar requiriendo una respuesta con los sockets locales asignados;

solicitar (607) al connection-broker por parte del servidor proxy-firewall una reserva de sockets locales sin usar para utilizarlos como socket-proxies directos una vez se resuelvan los sockets protegidos del dispositivo_AF que se requieren proxificar de forma directa por medio del agente_A;

registrar (609) en la base de datos, por parte del connection-broker, los permisos_A, pertinentes que permitan al usuario_A acceder a los socket-proxies directos;

registrar (609) en la base de datos cada registro_socket generado por cada reserva de socket a proxificar de forma directa y el cual comprende:

- el id_DispSocket, que en esta reivindicación guardaría el ~~id del dispositivo del~~ socket protegido correspondiente y que conoce el proceso_A;
- el id_DispProxy, que en esta reivindicación guardaría el id del dispositivo_AF denominado "id_AF";
- el alias, que en esta reivindicación guardaría el alias del socket que el proceso_A desea proxificar y que también conoce el proceso_A;
- el socket, que en esta reivindicación guardaría el socket local libre asignado correspondiente para realizar cada proxificación directa y que dispone de un valor de dirección de red perteneciente al pool de direcciones de red locales del dispositivo_A reservados para este método; y
- el tipo_socket, que en esta reivindicación sería igual a 'socket_proxy';

Comentado [David1]: Error evidente N° 1: La variable "ultima_inactividad" no existe en ningún elemento de la descripción o reivindicaciones. El término correcto es "ultima_actividad" y hace referencia a la variable del registro_acceso (pag 4, línea 34)

Comentado [David2]: Error evidente N° 2: Falta la "r" del infinitivo

Comentado [David3]: Error evidente N° 3: En la reivindicación 4 se habla de solicitar proxificación de varios sockets (Pag.5 línea 14 y 15) y por tanto, se debe poner en plural y sustituir "la proxificación directa" por "las proxificaciones directas"

Comentado [David4]: Error evidente N° 4: El termino Id_DispSocket hace referencia a un id de un dispositivo tal y como se menciona también en la reivindicación 12 (pag. 12, Línea 28). Se omitió por error y se sugiere incluir "id del dispositivo del"

solicitar (613) al agente_A el establecimiento de una sesión de comunicaciones con el dispositivo_AF utilizando unas credenciales de acceso al dispositivo_AF del usuario_A provistas por el servidor socket-proxy, sesión a través de la cual se realice la resolución de los sockets protegidos del dispositivo_AF a proxificar de forma directa en el dispositivo_A;

solicitar (614) al dispositivo_AF a través de la sesión establecida en esta reivindicación la resolución y el establecimiento de los accesos para los sockets protegidos requeridos del dispositivo_AF indicandos en la solicitud;

establecer (618) por parte del servidor proxy-firewall la regla de marcado pertinente a cada permiso_Au por medio del framework de red que permitan al usuario_A acceder a los socket-proxies directos;

solicitar el establecimiento de cada proxificación directa (604) en el dispositivo_A hacia cada socket protegido correspondiente del dispositivo_AF tal y como requiere el proceso_A y según se especifica en la reivindicación 4;

responder (620) al proceso_A por parte del servidor proxy-firewall la resolución de sockets solicitada; y

instanciar, por parte del servidor proxy-firewall, un monitor de accesos (621) que actualice periódicamente el tiempo_inactividad de cada registro_acceso insertado en la base de datos en esta reivindicación.

Comentado [David5]: Error evidente N° 5: Palabra mal escrita: Se sugiere sustituir "agentea_A" por "agente_A"

Comentado [David6]: Error evidente N° 6: La variable "tiempo inactividad" no existe en ningún elemento de la descripción o reivindicaciones. El término correcto es "ultima_actividad" y hace referencia a la variable del registro_acceso (pag 4, línea 34)

[Reivindicación 5] El método según las reivindicaciones 1-3, donde varios sockets protegidos del dispositivo_A son a su vez proxificados de forma directa hacia un dispositivo_AF por un agente del dispositivo_AF denominado "agente_AF" usando unos datos de configuración y credenciales de una cuenta de usuario denominada "usuario_AAF" definidos por un perfil de configuración del dispositivo_A denominado "perfil_UAAF" que le permiten conectarse a la cuenta del usuario_AAF del dispositivo_A, además comprende:

establecer una sesión segura entre el agente_AF y el servidor socket-proxy del dispositivo_A por iniciativa del agente_AF;

solicitar en el establecimiento de dicha sesión una resolución de cada socket protegido del dispositivo_A que sea requerido por el agente_AF;

instanciar un fork del servidor proxy-firewall en la cuenta usuario_AAF denominado "fork_AAF";

solicitar al connection-broker por parte del fork_AAF una resolución de los sockets protegidos a partir del alias y el id_A de cada socket_A y el alias y el id_B de cada socket_B indicados por el agente_AF;

verificar por parte del connection-broker del dispositivo_A la existencia de permisos suficientes para el usuario_AAF en la base de datos para poder acceder a los sockets protegidos;

solicitar al connection-broker por parte del fork_AAF un acceso de comunicación por cada socket protegido para la cuenta del usuario_AAF del sistema operativo del

dispositivo_A para todos aquellos sockets protegidos requeridos por el agente_AF con los que en usuario_AAF aún no cuente con acceso;

registrar cada registro_acceso de cada acceso concedido en la base de datos del connection-broker;

establecer por parte del fork_AAF la regla de marcado relativa a cada registro_acceso por medio del framework de red del sistema operativo del dispositivo_A que permitan al usuario_AF acceder a cada socket protegido; responder al agente_AF con el valor de cada socket protegido requerido para que posteriormente, el agente_AF pueda efectuar las proxificaciones directas; e

instanciar un monitor de accesos por parte del fork_AAF para actualizar periódicamente el ~~última_inactividad~~ de cada registro_acceso insertado en la base de datos en esta reivindicación.

Comentado [David7]: Error evidente N° 7: La variable "ultima_inactividad" no existe en ningún elemento de la descripción o reivindicaciones. El término correcto es "ultima_actividad" y hace referencia a la variable del registro_acceso (pag 4, línea 34)

[Reivindicación 6] El método según las reivindicaciones 1-3, donde varios sockets protegidos del dispositivo_A son proxificados de forma reversa en el dispositivo_AF por requerimiento del proceso_A del usuario_A del dispositivo_A a través del agente_A, el cual además comprende:

solicitar por parte del proceso_A al servidor proxy-firewall la realización de una proxificación reversa de unos sockets locales protegidos a los que tiene acceso;

solicitar al connection-broker la resolución de los sockets locales requeridos en la proxificación solicitada por el proceso_A;

comprobar, por parte del connection-broker, si el usuario_A dispone de los permisos necesarios para acceder a los sockets locales protegidos requeridos en la solicitud de proxificación;

insertar en la base de datos, por parte del connection-broker, unos accesos de comunicación a los sockets protegidos para la cuenta del usuario_A para todos aquellos sockets protegidos con los que aún no cuente con un acceso;

registrar los accesos de comunicación en la base de datos del connection-broker como registro_accesos;

establecer, por parte del servidor proxy-firewall, la regla de marcado pertinente a cada permiso_Ai insertada en la base de datos en esta reivindicación que permita al usuario_A acceder a todos los sockets fuente de las proxificaciones reversas;

solicitar al agente_A, por parte del servidor proxy-firewall, el establecimiento de una sesión de comunicaciones con el dispositivo_AF utilizando unas credenciales de acceso al dispositivo_AF del proceso_A provistas por el servidor proxy-firewall;

~~Solicitar~~, por parte del agente_A en el establecimiento de dicha sesión, una reserva de un socket protegido del dispositivo_AF por cada socket a proxificar de forma reversa hacia ese dispositivo;

Comentado [David8]: Error evidente N° 8: Se debe usar minúscula

instanciar, por parte del servidor proxy-firewall, un monitor de accesos por parte del servidor proxy-firewall para actualizar periódicamente la última actividad el timestamp de todos los registro_accesos del usuario_A necesarios en este método; y
 enviar al dispositivo_AF por parte del agente_A la solicitud de establecimiento de las proxificaciones reversas.

Comentado [David9]: Error evidente N° 9: Se debe usar minúscula

Comentado [David10]: Error evidente N° 10: Expresión repetida en la misma oración. Se sugiere eliminar la última.

Comentado [David11]: Error evidente N° 11: La variable "timestamp" no existe en ningún elemento de la descripción o reivindicaciones. El término correcto es "ultima_actividad" y hace referencia a la variable del registro_acceso (pag 4, línea 34)

[Reivindicación 7] El método según las reivindicaciones 1-4, donde varios sockets protegidos del dispositivo_AF son proxificados de forma reversa en el dispositivo_A por acción del agente_AF usando unos datos de configuración y credenciales de una cuenta "usuario_AAF" definidos por el "perfil_UAAF" que le permiten conectarse a la cuenta del usuario_AAF del dispositivo_AF, además comprende:

establecer una sesión segura entre el agente_AF y el servidor socket-proxy del dispositivo_A por iniciativa del agente_AF;

solicitar en el establecimiento de dicha sesión la reserva sockets locales protegidos del dispositivo_A para la realización de una proxificación reversa en el dispositivo_A de sockets del dispositivo_AF;

Comentado [David12]: Error evidente N° 12: Falta la "r" del infinitivo

instanciar un fork_AAF del servidor proxy-firewall en la cuenta de usuario_AAF;

solicitar al connection-broker, por parte del fork_AAF, una reserva de sockets locales sin usar para utilizarlos como los socket-proxies reversos indicados en la solicitud recibida;

Comentado [David13]: Error evidente N° 13: Falta el determinante "los"

registrar en la base de datos, por parte del connection-broker, los registro_socket relativos a los sockets reservados para la proxificación reversa;

agregar, por parte del connection-broker, un permiso_AU a la base de datos por cada socket reservado para el usuario_AAF;

insertar en el sistema, por parte del fork_AAF-servidor-proxy-firewall, una regla de marcado por cada permiso_AU por medio del framework de red;

Comentado [David14]: Error evidente N° 14: El "fork_AAF" es una instancia del "servidor proxy firewall" (línea 16 página 8). Se sugiere reemplazar "servidor proxy-firewall" por "fork_AAF" para mantener coherencia y evitar confusión.

Instanciar, por parte del fork_AAF-servidor-proxy-firewall, un monitor de accesos para mantener los accesos activos relativos a la proxificación reversa; y

Comentado [David15]: Error evidente N° 15: Se sugiere reemplazar "servidor proxy-firewall" por "fork_AAF"

establecer las proxificaciones reversas, por parte del fork_AAF-servidor-proxy-firewall, empleando los sockets reservados.

Comentado [David16]: Error evidente N° 16: Se sugiere reemplazar "servidor proxy-firewall" por "fork_AAF"

[Reivindicación 8] El método según las reivindicaciones 3, 4-7, comprende la actualización de la variable última_actividad a instancias del monitor_de_acceso en cada registro_acceso involucrado en el método en diferentes escenarios:

si el registro de acceso se crea según la reivindicación 3, la variable última_actividad del registro_acceso se actualizará a instancias del monitor de conexión instanciado por el servidor proxy-firewall siempre que haya tráfico de la cuenta de usuario_A hacia el socket-proxy; y

si el registro de acceso se crea según la reivindicación 4,5,6 o 7, su timestamp la variable última_actividad de cada registro_acceso se actualizará a instancias del monitor de conexión instanciado por el servidor proxy-firewall siempre que se mantenga en

Comentado [David17]: Error evidente N° 17: Las palabras "su timestamp" sobran y se sugiere su eliminación.

ejecución el proceso que solicitó la proxificación que causó la creación de cada registro_acceso en la base de datos.

[Reivindicación 9] El método de las reivindicaciones 1-3, donde un proceso denominado "proceso_C" de un dispositivo denominado "dispositivo_C", dirige una petición segura del recurso ofrecido por el socket_B del dispositivo_B a un proxy ejecutado por el usuario_A, además comprende:

La utilización de subdominios de red del dominio principal del proxy como parámetros de la petición que requiere el proxy para poder solicitar la resolución del socket protegido por medio del connection-broker;

un certificado wildcard X.509 configurado adecuadamente para ofrecer seguridad a los subdominios del proxy;

recibir las credenciales del proceso_C en los encabezados de la petición que permiten al proxy verificar la identidad del proceso_C y comprobar si dispone del permiso_Au requerido para cursar la petición;

establecer una comunicación segura entre el proceso_C y el proxy empleando un protocolo de comunicaciones seguro conocido por el proceso_C y el proxy;

suprimir los encabezados de la petición;y

establecer una comunicación entre el proxy y el socket protegido que comunique con el socket_B según la reivindicación 3.

[Reivindicación 10] El método según las reivindicaciones 3-9, donde se cierra el acceso de comunicación relativo al registro_acceso cuando la inactividad supera un umbral denominado "umbral_inactividad" en el perfil_A, además comprende:

eliminación de la regla de marcado del sistema operativo del dispositivo_A relativa al registro_acceso;

eliminación del registro-acceso de la base de datos del connection-broker del dispositivo_A; y

eliminación del permiso_Au relativo al registro_acceso en caso de que éste sea relativo a un socket-proxy directo.

[Reivindicación 11] El método según la reivindicación 3 donde se requiere actualizar la seguridad en las comunicaciones entre el cliente proxy stand-alone y el servidor proxy stand-alone, además comprende:

solicitar, a través de la sesión de comunicaciones establecida entre el agente_B y el servidor socket-proxy del inicio de un procedimiento de la actualización del cliente proxy del agente_B por iniciativa del servidor proxy-firewall;

recibir, por parte del agente_B, la nueva versión del cliente proxy y actualizar el perfil_BA;

probar, por parte del agente_B, la nueva versión del cliente proxy con el perfil_B_A actualizado estableciendo una nueva sesión de comunicaciones; y

en caso de establecerse correctamente esta nueva sesión, eliminar la versión anterior del cliente proxy y reemplazarlo con la nueva versión.

5 **[Reivindicación 12]** Un sistema para comunicar de forma segura sockets de red que comprende:

un módulo software denominado "servidor proxy-firewall" albergado en un dispositivo denominado "dispositivo_A" y que está configurado para:

- recibir las solicitudes de establecimiento de sesión de otros dispositivos a través de un protocolo seguro por medio de un servidor proxy stand-alone,
- configurar contextos de seguridad estrictos y amplios mediante llamadas al sistema al framework de red del sistema operativo del dispositivo_A,
- procesar solicitudes de proxificación o "socket-proxy" a través de sesiones establecidas o llamadas al sistema de procesos locales,
- monitorizar los accesos a sockets locales mediante monitores de acceso,
- gestionar las actualizaciones de seguridad de los dispositivos con los que mantenga una conexión, y
- establecer, mediante llamadas al sistema hacia el framework de red, protección de los sockets locales utilizados en las proxificaciones en el dispositivo_A;

un tipo de perfil de configuración del servidor proxy-firewall para definir los aspectos relativos al funcionamiento del servidor proxy-firewall y al connection-broker denominado "perfil_A";

un módulo software que funciona como agente del servidor proxy-firewall configurado para iniciar las solicitudes de establecimiento de sesión hacia el servidor proxy-firewall a través de un módulo software denominado cliente proxy stand-alone y que implementa un protocolo seguro, realizar solicitudes de proxificación de sockets de forma directa e inversa y establecer la protección de los sockets locales, siendo este software denominado como:

- agente_A, si se encuentra en un dispositivo_A;
- agente_AF, si se encuentra en un dispositivo_A foráneo denominado a si vez dispositivo_AF;
- agente_B, si se encuentra en un dispositivo denominado "dispositivo_B";

un tipo de perfil de configuración del agente denominado "Perfil_B_A", si el perfil es para configurar la sesión que el agente del dispositivo_B realiza en el dispositivo_A o Perfil_UA_{AF}, si el perfil es para configurar la sesión que el agente del dispositivo_A realiza con el dispositivo_AF a instancias del usuario_A;

un módulo software denominado "monitor de accesos" empleado para mantener un registro y control sobre las proxificaciones realizadas y los accesos a las mismas en el dispositivo_A;

un tipo de registro del dispositivo_A denominado "registro_socket" relativo a un socket protegido y el cual comprende:

- una variable denominada "socket" que guarda el socket protegido al que hace referencia el presente registro_socket y que puede ser de tres tipos:
 - o socket_B si el socket a registrar es un socket-proxy que ofrece un recurso de un dispositivo_B;
 - o socket_A, si el socket a registrar ofrece un recurso de un proceso del propio dispositivo_A y no es resultado de ningún tipo de proxificación;
 - o socket-proxy;
- una variable denominada "alias" que guarda el alias del socket protegido, el cual puede ser de dos tipos:
 - o el alias de un socket_B denominado "alias_B", y
 - o el alias de un socket_A denominado "alias_A";
- una variable denominada "id_DisSocket" que guarda el id del dispositivo que tiene el socket protegido localmente y que puede ser de tres tipos:
 - o el id de un dispositivo_B denominado "id_B" si el registro_socket es de tipo socket_B,
 - o el id de un dispositivo_AF denominado "id_AF" si el registro_socket es de tipo socket-proxy, y
 - o el id del propio dispositivo_A denominado "id_A" si el registro_socket es de tipo socket_A;
- una variable denominada "id_DisProxy" que identifica el id del dispositivo al que conduce el socket protegido; y
- una variable denominada "tipo_socket" que determina el tipo de socket protegido al a la que hace referencia este registro pudiendo dicha clase encontrarse entre: socket_A, socket_B y socket-proxy.

un tipo de registro denominado "registro_acceso" de un usuario del sistema operativo del dispositivo_A generado por cada acceso concedido a un socket protegido de tipo socket-proxy directo o reverso, el cual comprende:

- una variable denominada "con_Id" que guarda el identificador del acceso de comunicación al socket-proxy de un usuario_A,

Comentado [David19]: Error evidente N° 19: Se sugiere cambiar "a la" por "al"

Comentado [David20]: Error evidente N° 20: "Dicha clase" es redundante. Se sugiere retirar.

- una variable denominada "socket" que almacena el socket protegido al que se concede el acceso y está formado por la dirección de red local y el puerto,
- una variable denominada "id_usuario" que almacena el identificador del usuario_A del proceso_A origen de la petición de comunicación al que se le ha concedido acceso, y
- una variable denominada "última_actividad" que guarda el momento en el que se le ha concedido el acceso y que se actualizará por un monitor de acceso reflejando su estatus de actividad;

Comentado [David21]: Error evidente N° 21: Falta la conjunción "y"

un tipo de registro denominado "registro_acceso_A" de un grupo de usuarios del sistema operativo del dispositivo_A generado por cada acceso concedido a un socket protegido de tipo socket_A, el cual:

- una variable denominada "con_Id" que almacena el identificador del acceso con el mismo nombre,
- una variable denominada "socket" que almacena el socket_A, y
- una variable denominada "id_Grupo" que almacena el id del grupo del usuario del sistema operativo del dispositivo_A correspondiente al usuario_A del proceso_A origen de la petición de comunicación al que se le ha concedido acceso;

un tipo de registro para definir el permiso de acceso denominado "permiso_Au" que define un contexto de seguridad estricto al otorgar al usuario_A del dispositivo_A la posibilidad de acceder a un socket protegido de tipo socket-proxy directo o reverso y comprende:

Comentado [David22]: Error evidente N° 22: Falta la "r" del infinitivo

- una variable denominada "id_Usuario" que guarda el id del usuario_A del sistema operativo del dispositivo_A,
- una variable denominada "alias" que almacena el alias del socket protegido que se proxifica como socket-proxy en el dispositivo_A, y
- una variable denominada "id_DisSocket" que almacena el id del dispositivo con el socket_B ó socket_A relativo al alias del socket que ofrece el recurso a través del socket-proxy;

un tipo de registro para definir el permiso de acceso denominado "permiso_Ag" que define un contexto de seguridad amplio al otorgar la posibilidad de acceso de un grupo de usuarios del dispositivo_A a un socket_A y que comprende:

- una variable denominada "id_Grupo" que almacena el id del grupo de usuarios del sistema operativo del dispositivo_A,
- una variable denominada "alias" que almacena el alias del socket_A, y
- una variable denominada "socket" que guarda el valor del socket_A;

un módulo software denominado connection-broker empleado para mantener un registro y control sobre las proxificaciones realizadas y los accesos a las mismas en el dispositivo_A pudiendo cerrarlas si se encuentran inactivas;

un pool de direcciones locales del dispositivo_A reservadas específicamente para este sistema y que son protegidas al arrancar el sistema operativo del dispositivo_A por el framework de red a instancias del servidor proxy-firewall por medio de las siguientes reglas generales:

- una regla general de marcado de paquetes que determine eliminar cualquier posible marcado de paquetes de red en cualquier comunicación producida entre sockets locales del dispositivo_A con direcciones locales pertenecientes al pool de direcciones locales reservado;
- una regla general de marcado de paquetes que determine descartar cualquier paquete con origen y destino un socket local del dispositivo_A que carezca de un marcado siempre y cuando alguno de los sockets utilice una dirección de red local perteneciente al pool de direcciones locales reservado; y
- una regla general de filtrado de paquetes que determine descartar cualquier paquete con origen o destino una dirección local del pool de direcciones locales reservado para este método que tenga como origen o destino una dirección no local;

un tipo de contexto de seguridad denominado "contexto de seguridad estricto" definido por una configuración del framework de red que realiza un marcado en los paquetes del tráfico utilizando el con_Id asignado en el registro_acceso relativo al socket protegido destino del dispositivo_A, el usuario origen de la comunicación también del dispositivo_A y el permiso_AU que habilita el registro_acceso;

un tipo de contexto de seguridad denominado "contexto de seguridad amplio" definido por una configuración del framework de red que realiza un marcado en los paquetes del tráfico utilizando el con_Id asignado en el registro_acceso relativo al socket protegido destino del dispositivo_A y el grupo del usuario origen de la comunicación también del dispositivo_A y el permiso_AG que habilita el acceso; y

una base de datos del connection-broker configurada para almacenar distintos tipos de registros que comprenden: registro_proxy, registro_acceso, registro_acceso_A, permiso_AU, permiso_AG.

Comentado [David23]: Error evidente N° 23: La U y la G deben ser subíndices

[Reivindicación 13] El sistema de la reivindicación 12, en donde los dispositivos utilizan un sistema operativo tipo POSIX, emplean un cliente proxy y un servidor proxy stand-alone basado en el protocolo seguro SSH y donde el framework de red utilizado es Netfilter, realiza el marcado y filtrado de paquetes utilizando el campo TOS del protocolo IP.

5 [Reivindicación 14] El sistema de la reivindicación 12 y 13, en donde un dispositivo_B también dispone de un servidor proxy-firewall que permite incluir en el perfil_B_A aspectos de configuración que extienden el control que dispone el servidor proxy-firewall del dispositivo_A sobre todos los módulos software del dispositivo_B para facilitar el desarrollo de servicios que impliquen llamadas al sistema, como es la actualización de módulos software del dispositivo_B.

10 [Reivindicación 15] El sistema de las reivindicaciones 12 y 13, en el que el socket_B del dispositivo_B ofrece un recurso utilizando el protocolo HTTP y el dispositivo_A utiliza un proxy como módulo software de tipo HTTPS-HTTP para poder aceptar y procesar las solicitudes externas de comunicación hacia el socket protegido que se cursen en HTTPS, además comprende:

la configuración de un certificado SSL wildcard para proteger el dominio principal del proxy HTTPS-HTTP del dispositivo_A y todos sus subdominios posibles;

15 la configuración de un servidor web en el dispositivo_A para poder redirigir la petición de comunicación del recurso del socket_B de forma transparente a la URL principal del proxy HTTPS-HTTP aprovechando los subdominios como campos para identificar el socket-proxy destino de dicha petición y utilizando como subdominios el alias del socket_B y el identificador del dispositivo_B; y

20 una configuración adecuada del proxy HTTPS-HTTP que permita obtener de los encabezados de la petición las credenciales requeridas para validarla y eliminarlos de la misma antes de transmitirla hacia el socket-proxy.

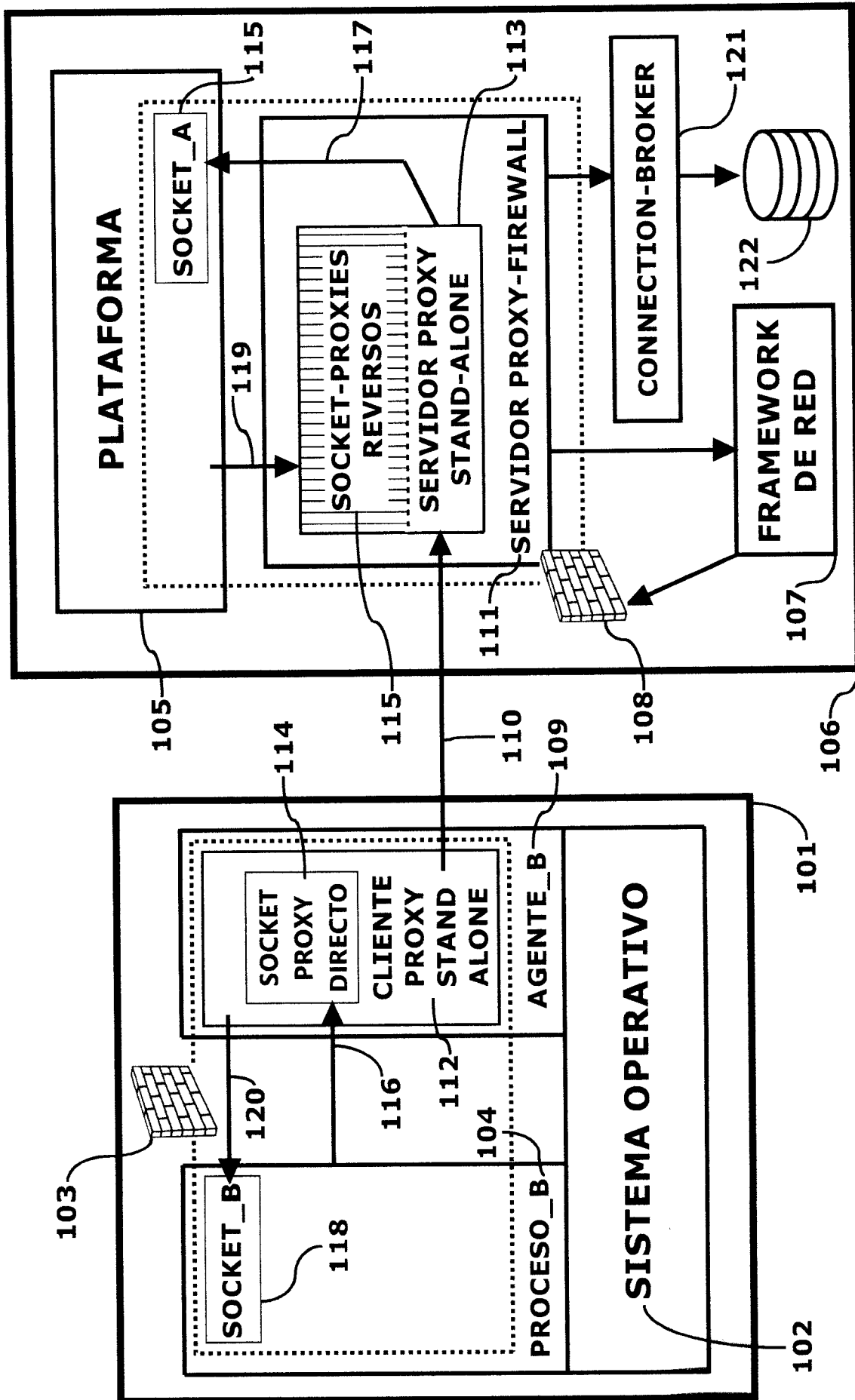


Fig. 1

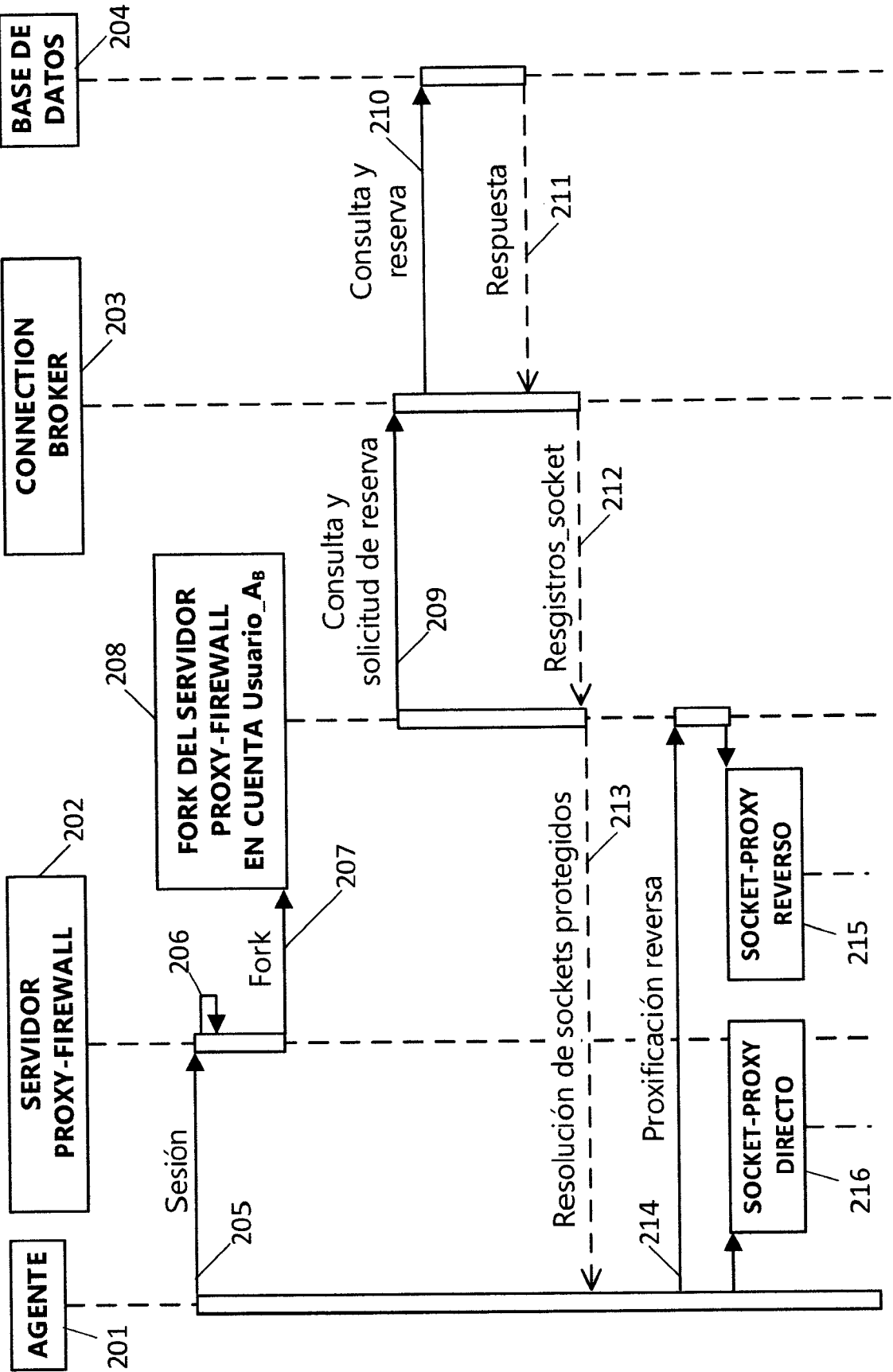


Fig. 2

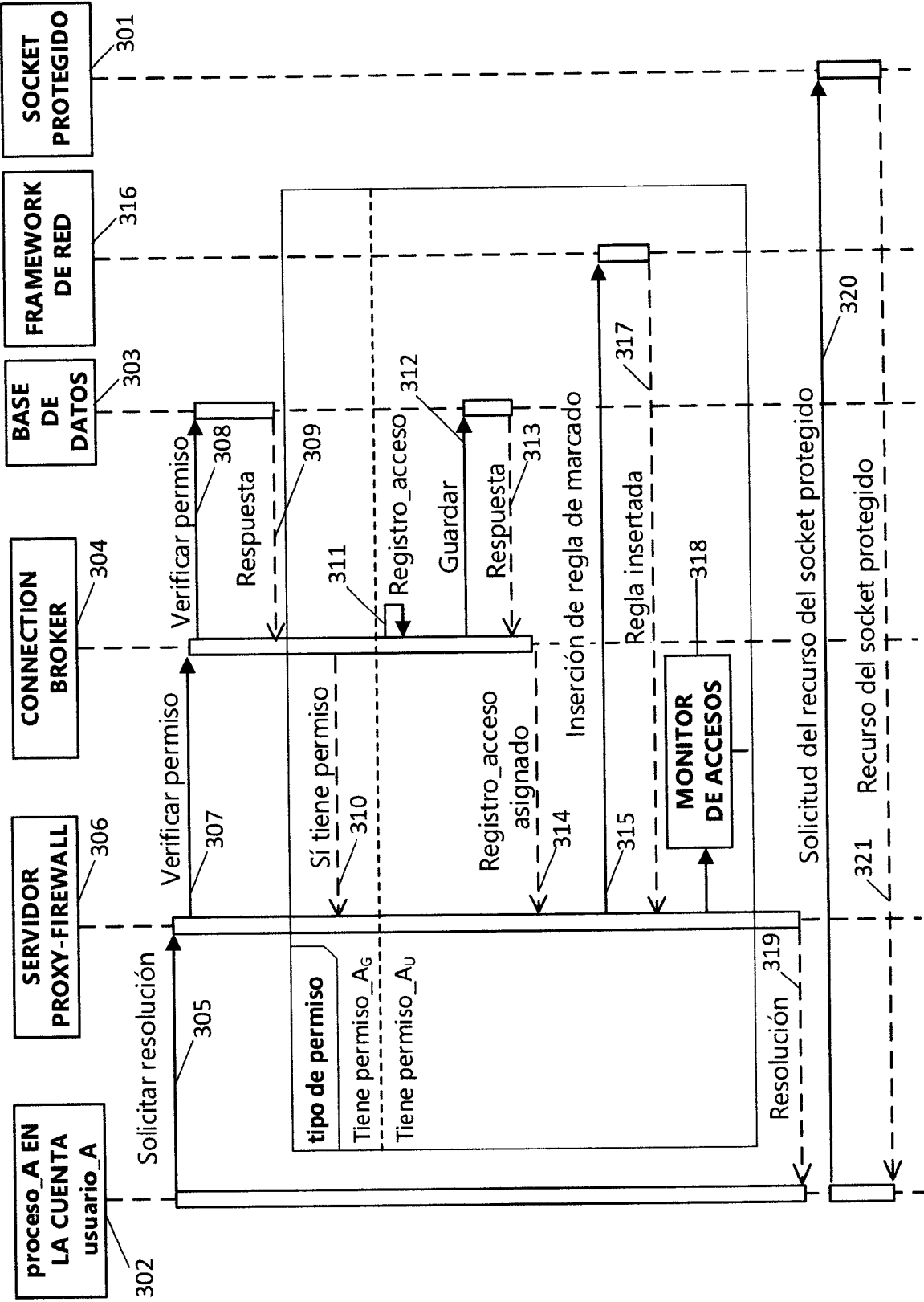


Fig. 3

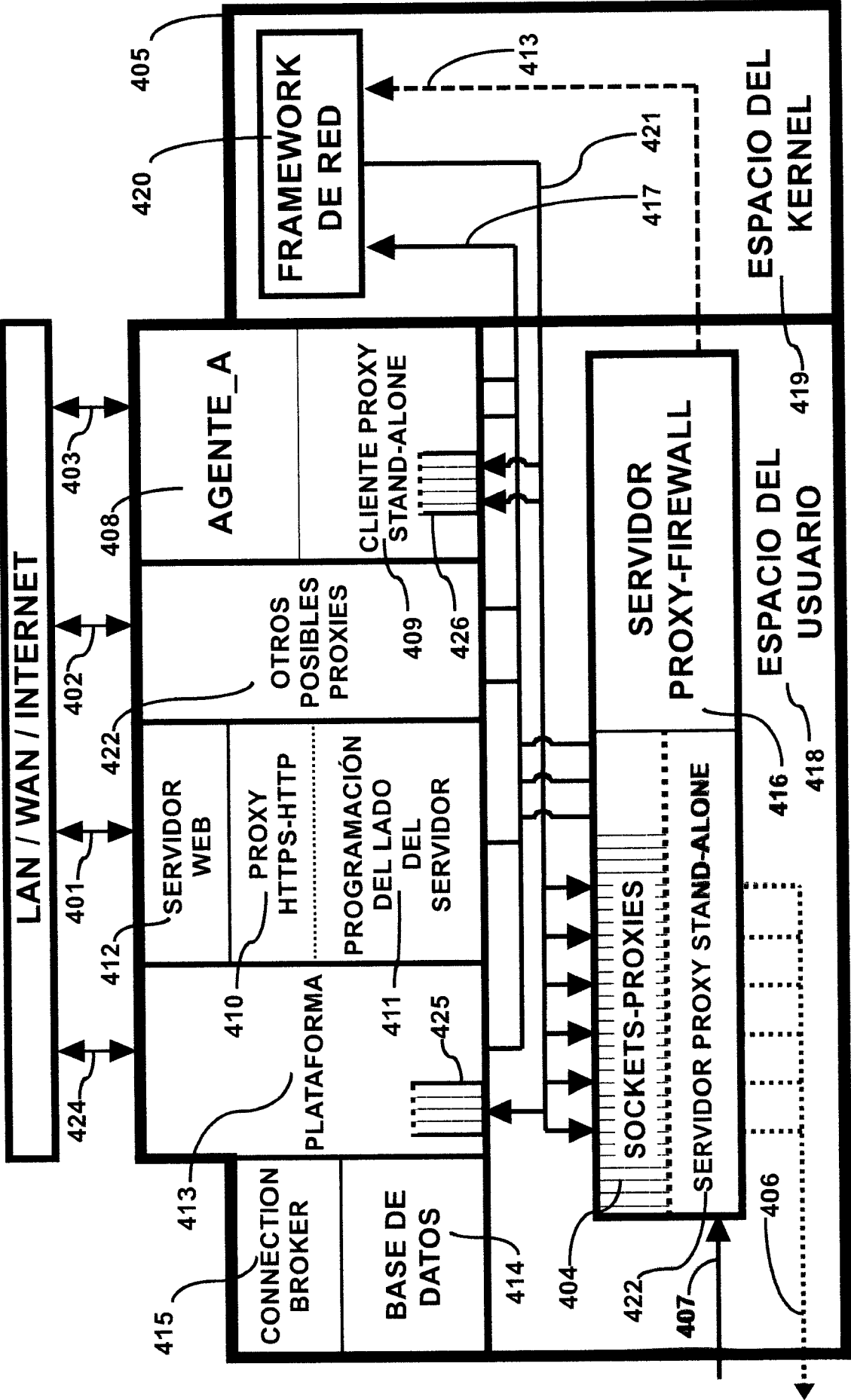


Fig. 4

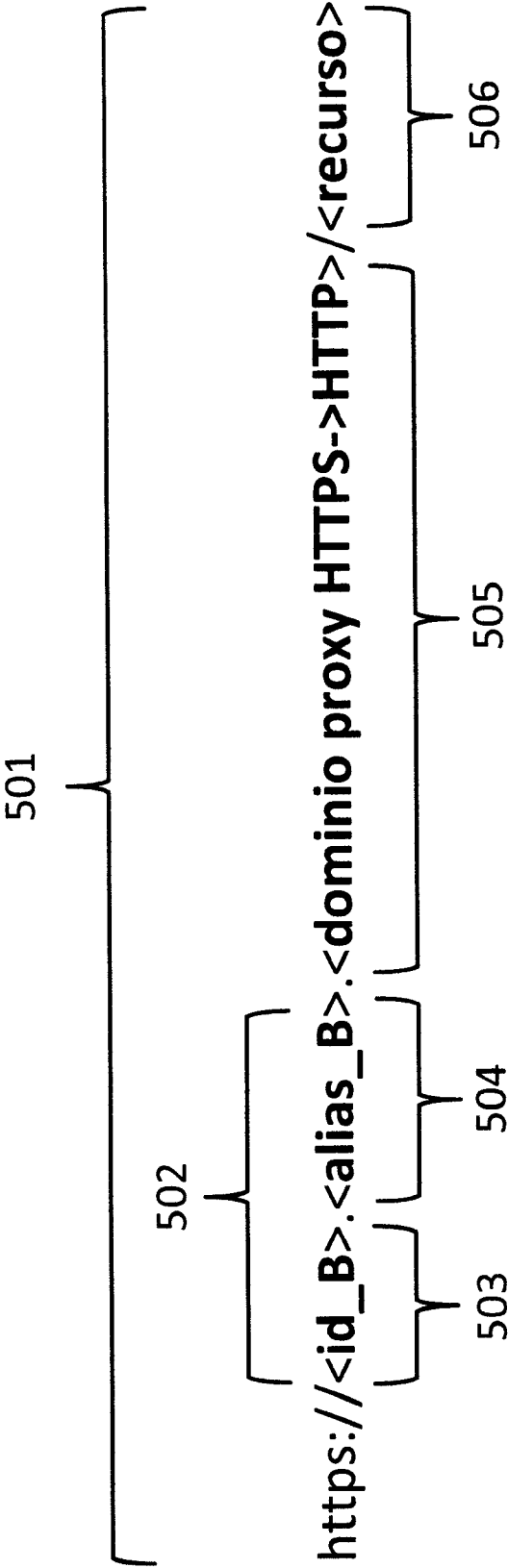


Fig. 5

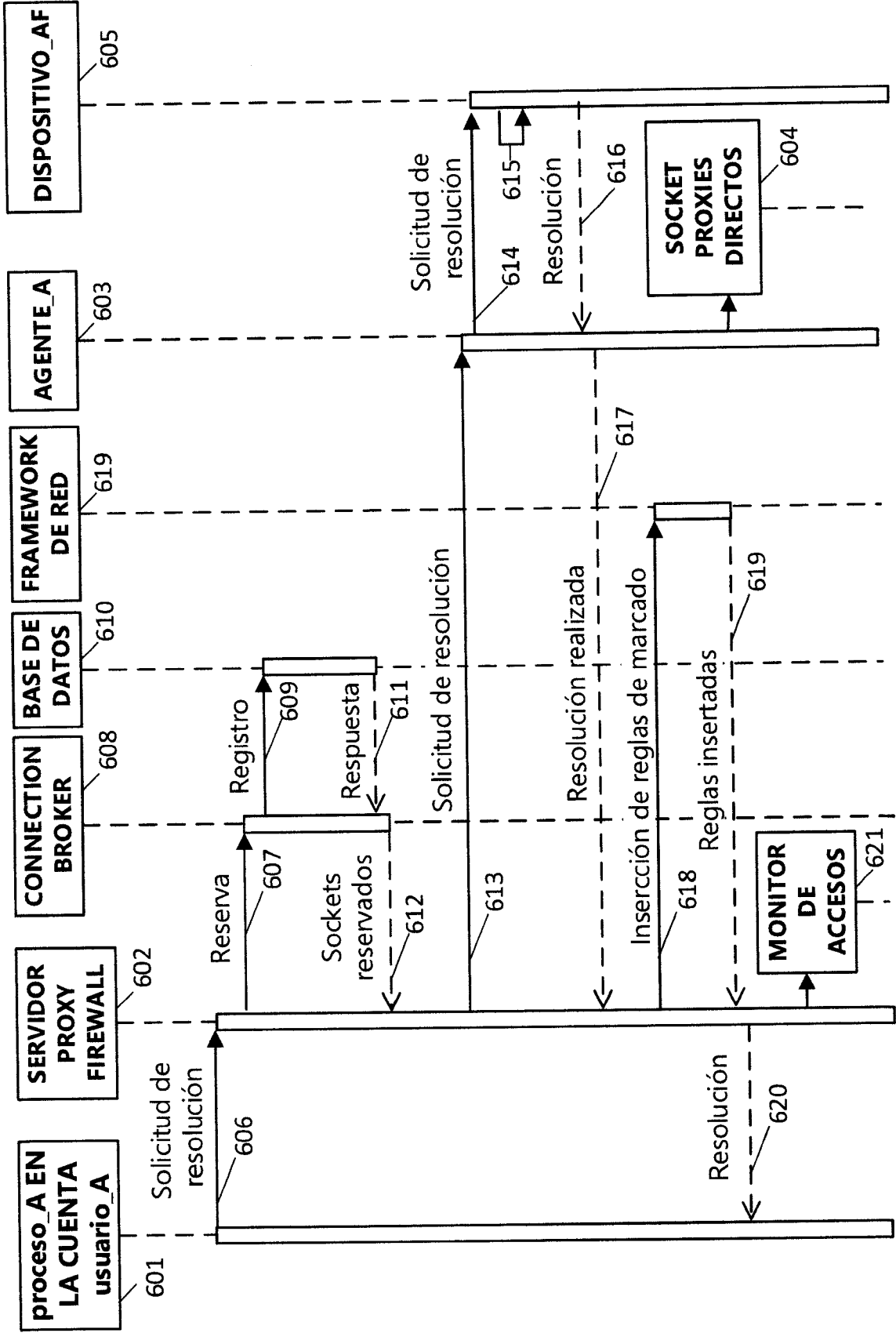


Fig. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/MX2018/050013

A. CLASSIFICATION OF SUBJECT MATTER

G06F21/00 (2013.01)

H04L9/00 (2006.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F, H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPODOC, INVENES, WPI, NPL, XPIEE, XPI3E

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2017272316 A1 (JOHNSON MICHAEL W ET AL.) 21/09/2017, the whole document.	1-15
A	PORTMANN M et al.. Selective security for TLS. Networks, 2001. Proceedings. Ninth IEEE International Conference on October 10-12, 2001, 20011010; 20011010 - 20011012 Piscataway, NJ, USA,IEEE, Los Alamitos, CA, USA. 10/10/2001, Pages 216 - 221, ISSN 1531-2216 ISBN 978-0-7695-1187-0 ; ISBN 0-7695-1187-2. The whole document.	1-15

☒ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance.	
"E" earlier document but published on or after the international filing date	
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"O" document referring to an oral disclosure use, exhibition, or other means.	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other documents , such combination being obvious to a person skilled in the art
"P" document published prior to the international filing date but later than the priority date claimed	"&" document member of the same patent family

Date of the actual completion of the international search
08/01/2019

Date of mailing of the international search report
(11/01/2019)

Name and mailing address of the ISA/

Authorized officer
M. Muñoz Sanchez

OFICINA ESPAÑOLA DE PATENTES Y MARCAS
Paseo de la Castellana, 75 - 28071 Madrid (España)
Facsimile No.: 91 349 53 04

Telephone No. 91 3495349

INTERNATIONAL SEARCH REPORT

International application No.

PCT/MX2018/050013

C (continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of documents, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	TSCHOFENIG H et al.. TLS/DTLS Profiles for the Internet of Things; draft-ietf-dice-profile-17.txt. TLS/DTLS Profiles for the Internet of Things; draft-ietf-dice-profile-17.txt, 20151019 Internet Engineering Task Force, IETF; StandardWorkingDraft, Internet Society (ISOC) 4, rue des Falaises CH- 1205 Geneva, Switzerland. 19/10/2015, Pages 1 - 59. The whole document	1-15
A	MILOSLAVSKAYA NATALIA et al.. Ensuring Information Security for Internet of Things. 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), 20170821 IEEE. 21/08/2017, Pages 62 - 69, <DOI:10.1109/FiCloud.2017.17>. The whole document.	1-15

INTERNATIONAL SEARCH REPORT

International application No.

Information on patent family members

PCT/MX2018/050013

Patent document cited in the search report	Publication date	Patent family member(s)	Publication date
US2017272316 A1	21.09.2017	US2018262388 A1	13.09.2018
		US2016344745 A1	24.11.2016
		US2016315824 A1	27.10.2016
		US2016112262 A1	21.04.2016
		US2016087933 A1	24.03.2016
		US9712486 B2	18.07.2017
		US2015113172 A1	23.04.2015
		US9231904 B2	05.01.2016
		US2015088982 A1	26.03.2015
		US2015052258 A1	19.02.2015
		US2015052253 A1	19.02.2015
		WO2013188835 A2	19.12.2013
		WO2013188835 A3	18.06.2015
		US2013339509 A1	19.12.2013
		US2013232243 A1	05.09.2013
		US9253031 B2	02.02.2016
		WO2008039481 A1	03.04.2008
		US2008077425 A1	27.03.2008
		US8447843 B2	21.05.2013
-----	-----	-----	-----

INFORME DE BÚSQUEDA INTERNACIONAL

Solicitud internacional nº

PCT/MX2018/050013

A. CLASIFICACIÓN DEL OBJETO DE LA SOLICITUD

G06F21/00 (2013.01)

H04L9/00 (2006.01)

De acuerdo con la Clasificación Internacional de Patentes (CIP) o según la clasificación nacional y CIP.

B. SECTORES COMPRENDIDOS POR LA BÚSQUEDA

Documentación mínima buscada (sistema de clasificación seguido de los símbolos de clasificación)

G06F, H04L

Otra documentación consultada, además de la documentación mínima, en la medida en que tales documentos formen parte de los sectores comprendidos por la búsqueda

Bases de datos electrónicas consultadas durante la búsqueda internacional (nombre de la base de datos y, si es posible, términos de búsqueda utilizados)

EPODOC, INVENES, WPI, NPL, XPIEE, XPI3E

C. DOCUMENTOS CONSIDERADOS RELEVANTES

Categoría*	Documentos citados, con indicación, si procede, de las partes relevantes	Relevante para las reivindicaciones nº
A	US 2017272316 A1 (JOHNSON MICHAEL W ET AL.) 21/09/2017, Todo el documento.	1-15
A	PORTMANN M et al.. Selective security for TLS. Networks, 2001. Proceedings. Ninth IEEE International Conference on October 10-12, 2001, 20011010; 20011010 - 20011012 Piscataway, NJ, USA, IEEE, Los Alamitos, CA, USA. 10/10/2001, Páginas 216 - 221, ISSN 1531-2216 ISBN 978-0-7695-1187-0 ; ISBN 0-7695-1187-2. Todo el documento.	1-15

☒ En la continuación del recuadro C se relacionan otros documentos

☒ Los documentos de familias de patentes se indican en el anexo

* Categorías especiales de documentos citados:	"T" documento ulterior publicado con posterioridad a la fecha de presentación internacional o de prioridad que no pertenece al estado de la técnica pertinente pero que se cita por permitir la comprensión del principio o teoría que constituye la base de la invención.
"A" documento que define el estado general de la técnica no considerado como particularmente relevante.	"X" documento particularmente relevante; la invención reivindicada no puede considerarse nueva o que implique una actividad inventiva por referencia al documento aisladamente considerado.
"E" solicitud de patente o patente anterior pero publicada en la fecha de presentación internacional o en fecha posterior.	"Y" documento particularmente relevante; la invención reivindicada no puede considerarse que implique una actividad inventiva cuando el documento se asocia a otro u otros documentos de la misma naturaleza, cuya combinación resulta evidente para un experto en la materia.
"L" documento que puede plantear dudas sobre una reivindicación de prioridad o que se cita para determinar la fecha de publicación de otra cita o por una razón especial (como la indicada).	"&" documento que forma parte de la misma familia de patentes.
"O" documento que se refiere a una divulgación oral, a una utilización, a una exposición o a cualquier otro medio.	
"P" documento publicado antes de la fecha de presentación internacional pero con posterioridad a la fecha de prioridad reivindicada.	

Fecha en que se ha concluido efectivamente la búsqueda internacional.
08/01/2019

Fecha de expedición del informe de búsqueda internacional.
11 de enero de 2019 (11/01/2019)

Nombre y dirección postal de la Administración encargada de la búsqueda internacional
OFICINA ESPAÑOLA DE PATENTES Y MARCAS
Paseo de la Castellana, 75 - 28071 Madrid (España)
Nº de fax: 91 349 53 04

Funcionario autorizado
M. Muñoz Sanchez
Nº de teléfono 91 3495349

INFORME DE BÚSQUEDA INTERNACIONAL

Solicitud internacional n°

PCT/MX2018/050013

C (Continuación). DOCUMENTOS CONSIDERADOS RELEVANTES		
Categoría *	Documentos citados, con indicación, si procede, de las partes relevantes	Relevante para las reivindicaciones n°
A	TSCHOFENIG H et al.. TLS/DTLS Profiles for the Internet of Things; draft-ietf-dice-profile-17.txt. TLS/DTLS Profiles for the Internet of Things; draft-ietf-dice-profile-17.txt, 20151019 Internet Engineering Task Force, IETF; StandardWorkingDraft, Internet Society (ISOC) 4, rue des Falaises CH- 1205 Geneva, Switzerland. 19/10/2015, Páginas 1 - 59. Todo el documento.	1-15
A	MILOSLAVSKAYA NATALIA et al.. Ensuring Information Security for Internet of Things. 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), 20170821 IEEE. 21/08/2017, Páginas 62 - 69, <DOI: 10.1109/FiCloud.2017.17>. Todo el documento.	1-15

INFORME DE BÚSQUEDA INTERNACIONAL

Informaciones relativas a los miembros de familias de patentes

Solicitud internacional n°

PCT/MX2018/050013

Documento de patente citado en el informe de búsqueda	Fecha de Publicación	Miembro(s) de la familia de patentes	Fecha de Publicación
US2017272316 A1	21.09.2017	US2018262388 A1	13.09.2018
		US2016344745 A1	24.11.2016
		US2016315824 A1	27.10.2016
		US2016112262 A1	21.04.2016
		US2016087933 A1	24.03.2016
		US9712486 B2	18.07.2017
		US2015113172 A1	23.04.2015
		US9231904 B2	05.01.2016
		US2015088982 A1	26.03.2015
		US2015052258 A1	19.02.2015
		US2015052253 A1	19.02.2015
		WO2013188835 A2	19.12.2013
		WO2013188835 A3	18.06.2015
		US2013339509 A1	19.12.2013
		US2013232243 A1	05.09.2013
		US9253031 B2	02.02.2016
		WO2008039481 A1	03.04.2008
		US2008077425 A1	27.03.2008
		US8447843 B2	21.05.2013
-----	-----	-----	-----