Industria y Comercio

SUPERINTENDENCIA

Delegatura de propiedad Industrial

División de Nuevas Creaciones

SOLICITUD

PATENTE DE INVENCIÓN

21 **EXPEDIENTE No.** 04-67547

54 **TÍTULO** Detención y parcheo Automatico de Archivos Vulnerables

51 **CLASIFICACIÓN INTERNACIONAL** ⁷ G06f 9/445

71 **SOLICITANTE** microsoft - corporation

DOMICILIO Redmond, washington, Estados Unidos

74 **APODERADO** Dario Cardenas Navas

22 **BOGOTÁ, D. C.,** 2004 - VII - 15

(FORMA P 10)

PETITORIO

Industria y Comercio
SUPERINTENDENCIA

SUPERINDUSTRIA Y COMERCIO
Radicacion : 04067547 00000000 Folios: 223
Fecha (AMD): 2004-07-15 16:58:53
Tramite : 002 PATENTES D 1 REGISTRO/ 411 PRESENTAC
Dependencia: 0020 DIVISION DE NUEVAS CREACIONES

FORMULARIO UNICO DE SOLICITUD DE PATENTE
2000-3

1 SOLICITUD DE:

[X] Patente de Invención [ ] Patente de Modelo de Utilidad

2 SOLICITANTE (71)

Nombre:	MICROSOFT CORPORATION
Dirección:	One Microsoft Way Redmond, Washington 98052, Estados Unidos de America
Teléfono:	3123600 Fax: 3122410
E-Mail:	general@cardenasycardenas.com
Lugar de Constitución	Estado de Washington, Estados Unidos de América
Domicilio:	REDMOND, WASHINGTON, ESTADOS UNIDOS DE AMERICA.

IDENTIFICACION
C.C. [ ] NIT [ ]
C.E. [ ] Otro [X]
Número

3 REPRESENTANTE O APODERADO (74)

Nombre:	DARIO CARDENAS NAVAS
Dirección:	Carrera 7 No 71-52 Torre B Piso 9
Teléfono:	3123600 Fax: 3122410
E-Mail:	general@cardenasycardenas.com

IDENTIFICACION
C.C. [X] NIT [ ]
C.E. [ ] Otro [ ]
Número 17.068.629 BTA
TP 1.250 C.S.J.

4 INVENTOR(ES) (72)

Nombre:	1. OLEG IVANOV 2. SERGEI IVANOV
Dirección:	1. 25923 SE 37th Way, Issaquah, Washington 98029 2. 25923 SE 37th Way, Issaquah, Washington 98029
Teléfono:	3123600 Fax: 3123600
E-Mail:	general@cardenasycardenas.com
Domicilio:	1. WASHINGTON, ESTADOS UNIDOS DE AMERICA 2. WASHINGTON, ESTADOS UNIDOS DE AMERICA

IDENTIFICACION
C.C. [ ] NIT [ ]
C.E. [ ] Otro [X]
Número

5 Título(54):

DETECCION Y PARCHEO AUTIMATICO DE ARCHIVOS VULNERABLES

6 Prioridad SI [X] NO [ ]

(33)País de Origen	(32)Fecha	(31)N° de Solicitud
USA	10/821,148	Julio 16, 2003

7 Para publicar a partir de la fecha de la presente solicitud a los:

[ ] 6 meses [ ] 12 meses [ ] 18 meses [ ] Otro

P-
JAG/AMR
amg

C 06F 17/60

DETECCION Y PARCHEO AUTIMATICO DE ARCHIVOS VULNERABLES

Comprobante de Pago No.: 04-55,265 / 04-55,266 Fecha Julio 13, 2004

ANEXOS

- [X] Comprobante de pago de la tasa de presentación de la solicitud. (No. 04-55,265)
- [ ] Comprobante de pago de la tasa por concepto de excedente de palabras en la publicación.
- [X] Comprobante de pago de la tasa por concepto de reivindicación de prioridad. (No 04-55,266)
- [X] Poderes, si fuere el caso. (Protocolo No 16.069)
- [X] Certificado de existencia y representación legal cuando el solicitante sea persona jurídica. Protocolo No 16.069)
- [X] Copia certificada de la primera solicitud, si se reivindica prioridad.
- [X] Traducción simple de la primera solicitud, si se reivindica prioridad.
- [X] Documento de cesión del inventor al solicitante o a su causante.
- [X] Descripción de la invención.
- [X] Una o más reivindicaciones.
- [X] Dibujos y/o planos necesarios.
- [ ] De ser el caso, copia del contrato de acceso.
- [ ] De ser el caso, documento que acredite la licencia o autorización de uso de conocimientos tradicionales de la comunidad de las comunidades indígenas.
- [ ] De ser el caso, certificado de depósito del material biológico.
- [ ] Arte final 12x12 y 6x6 cm.
- [X] Resumen y extracto para publicación.

10

NOMBRE: DARIO CARDENAS NAVAS

FIRMA:

C.C. 17.066.629 BTA TP. 1.250 C.S.J.

EV251222109

EN LA OFICINA DE PATENTES Y MARCAS DE LOS ESTADOS UNIDOS

SOLICITUD PARA TÍTULO DE PATENTE

Detección y Parcheo Automático de Archivos Vulnerables

Inventores:

Oleg Ivanov

Sergei Ivanov

REFERENCIA DE ABOGADO NO. MS1-1594US

4

CAMPO TÉCNICO

La presente descripción se relaciona de manera general con parcheo de archivos, y más particularmente, con una forma automática, comprensiva, confiable y libre de regresión de suministrar parches de seguridad para archivos de programas binarios vulnerables en ambientes de cómputo distribuidos, heterogéneos.

ANTECEDENTES

El desarrollo de software es un proceso que avanza por medio del cual el producto del software inicialmente liberado al público puede ser actualizado continuamente a través de revisiones del desarrollador/vendedor de software. Las revisiones de software son típicamente desembolsados de un vendedor de software en lo que son llamados "paquetes de servicios" que se pueden bajar u ordenar por un vendedor para instalación en un computador de usuario. Los paquetes de servicio contienen típicamente programas fijos (por ejemplo para un sistema operativo, un programa de aplicación, etc.) que repara problemas (es decir, "errores") descubiertos en el código de programa después de la liberación inicial del producto

o después de la última liberación del paquete de servicio.

Además de contener fijos para los errores del programa, los paquetes de servicio también pueden contener parches de seguridad desarrollados específicamente para reparar las vulnerabilidades encontradas en los archivos de programa. Las vulnerabilidades del programa descubiertas después de que es liberado un producto puede ser una amenaza de seguridad significativa de ataque de hackers y virus a nivel mundial. Por lo tanto, una vez se descubre la vulnerabilidad, la distribución puntual y de amplio esparcimiento y la instalación de los parches de seguridad a los computadores que tienen software vulnerable es de capital importancia. Teóricamente, el uso de paquetes de servicio para lograr tal distribución puntual y de amplio esparcimiento de parches de seguridad puede ser efectivo. Por ejemplo, cuando un vendedor de software descubre una vulnerabilidad y luego desarrolla un parche de seguridad, el parche puede ser colocado en el último paquete de servicio en un sitio Web del vendedor para que los usuarios lo bajen e instalen inmediatamente. Esto podría frustrar la mayoría de los

hackers y virus que pretenden explotar la vulnerabilidad descubierta. Sin embargo, los administradores del sistema y otros usuarios del producto de software habitualmente enfrentan varios inconvenientes y/o dificultades relacionadas con accesar e instalar parches de seguridad. Estas dificultades dan como resultado típicamente una distribución significativamente inferior de tales parches de la que es pretendida por el vendedor quien desarrolla el parche. El resultado es que las vulnerabilidades en muchos computadores a nivel mundial son dejadas sin parche, exponiendo tales computadores a un riesgo significativo.

Una dificultad con el acceso e instalación de parches de seguridad es que los métodos habituales para detectar si un computador está corriendo un software con una vulnerabilidad conocida requiere el uso activo y el involucramiento del computador. Por ejemplo, los métodos habitualmente disponibles pueden determinar si versiones particulares de productos de software en un computador están necesitadas de ser actualizadas (por ejemplo con un parche de seguridad). Sin embargo, solamente aquellos productos de software que corren activamente sobre el computador están incluidos en esta determinación. Los

sistemas operativos secundarios y las aplicaciones que no están corriendo activamente en un computador no son consideradas, y por lo tanto podrían tener una vulnerabilidad de seguridad que pasa desapercibida y no es fija. Para aquellos productos que corren activamente en un computador, un usuario puede revisar una lista de actualizaciones disponibles y seleccionar las actualizaciones para instalación. Algunas actualizaciones pueden ser actualizaciones críticas diseñadas para proteger un computador de vulnerabilidades de seguridad conocidas. Varias actualizaciones requieren que el usuario reinicie el computador antes de que se complete la instalación. Además, un usuario debe seleccionar activamente las actualizaciones e instalarlas. Por estas y otras razones, los métodos habituales para acceder e instalar los parches de seguridad son menos que efectivos.

Otra dificultad para accesar e instalar parches de seguridad es aquella de saber si un parche de seguridad es necesario en un computador. Algunas veces es difícil para los usuarios saber si en sus computadores está corriendo software que es vulnerable. Adicionalmente, los métodos disponibles para detectar si en un computador

está corriendo software con una vulnerabilidad conocida puede no ser capaz de detectar ciertas configuraciones de un producto de software conocido por ser vulnerable. Por ejemplo, las versiones compartidas de algunos productos de software se pueden distribuir como parte de otros productos. De esta manera, aunque la versión compartida de un producto puede contener la misma vulnerabilidad que la versión completa del producto, la versión compartida puede no ser reconocida como un producto que necesita una actualización del parche de seguridad. Así, las versiones compartidas de los productos de software que son conocidas por tener vulnerabilidades de seguridad a menudo no son fijas.

Otros problemas con el accesamiento y la instalación de parches de seguridad se relaciona con el método convencional de "paquete de servicio" por medio del cual tales parches son entregados. Bajar e instalar paquetes de programa es un proceso que consume tiempo y es manual que muchos administradores de sistema simplemente no tienen tiempo de efectuar. Por lo tanto, aún cuando los administradores pretendan instalar parches de seguridad, el tiempo entre la liberación del parche de seguridad y su instalación en un sistema dado puede ser de semanas,

meses, o años. Así, el riesgo de ataque a través de una vulnerabilidad de seguridad puede no aliviarse en tales sistemas hasta mucho después de que el vendedor de software ha lanzado un parche de seguridad.

Adicionalmente, los administradores de sistema a menudo escogen no bajar e instalar los paquetes de servicio que contienen parches de seguridad, aunque ellos entienden los riesgos de seguridad relevantes. La razón para esto es que la instalación de un paquete de servicio mismo conlleva el riesgo de regresiones del sistema que puede introducir cambios no deseados en el comportamiento del sistema. Los administradores a menudo dedican tiempo y esfuerzo significativo a depurar un sistema de tal forma que este funciones como se desea. Como se mencionó anteriormente, sin embargo, los paquetes de servicio representan una evolución de una versión previa de un producto de software que incluye las actualizaciones más recientes a una base de código de producto (es decir el alcance de los cambios no está restringido solamente a los parches de seguridad). Además de introducir comportamientos nuevos y pretendidos en un sistema, las actualizaciones de código recientes en un paquete de servicio pueden introducir errores desconocidos en un

sistema que puede originar que el sistema se comporte de manera inesperada, lo cual, a su vez puede crear problemas significativos para un administrador de sistemas. De esta manera, los sistemas frecuentemente no son actualizados con los parches de seguridad importantes destinados a fijarse en los archivos de programa vulnerables, por que los administradores no quieren riesgos de regresiones.

De acuerdo con esto, subsiste la necesidad de una forma de implementar parches de vulnerabilidades de seguridad en archivos de programa de una manera automática, comprensiva, confiable y libre de regresiones.

RESUMEN

Los Parches de seguridad automáticos, comprensivos, confiables y libres de regresión de los archivos de programa binarios se describen aquí.

De acuerdo con una implementación, es recibida una firma binaria de un parche de vulnerabilidad y uno de seguridad. Un archivo binario vulnerable se identifica en

un computador con base en la firma binaria de una vulnerabilidad. El archivo binario vulnerable en un computador es actualizado con el parche de seguridad.

De acuerdo con otra implementación, es recibida una firma binaria que identifica una vulnerabilidad de seguridad en un archivo binario. El parche de seguridad configurado para fijarse a la vulnerabilidad de seguridad también se recibe. La forma binaria y el parche de seguridad son distribuidos a una pluralidad de servidores.

De acuerdo con otra implementación, se recibe una firma binaria de un servidor que se utiliza para buscar archivos binarios. Se envía una solicitud de un parche de seguridad al servidor si la firma binaria se encuentra en un archivo binario. El archivo binario es luego actualizado con el parche de seguridad.

BREVE DESCRIPCION DE LOS DIBUJOS

Los mismos numerales de referencia son utilizados a lo largo de los dibujos con referencia a componentes y características similares.

La figura 1 ilustra un ambiente de red de ejemplo adecuada para implementar la detección automática y el parcheo de vulnerabilidades de seguridad en archivos binarios.

La figura 2 ilustra una modalidad de ejemplo de un servidor de distribución, un servidor de exploración de parches, y un computador de cliente adecuado para implementar la detección automática y el parcheo de vulnerabilidad de seguridad en archivos binarios.

La figura 3 ilustra otra modalidad de ejemplo de un servidor de distribución, un servidor de exploración de parches, y un computador de cliente adecuado para implementar la detección automática y el parcheo de vulnerabilidades de seguridad en archivos binarios.

Las figuras 4-6 ilustran diagramas de bloques de métodos de ejemplo para implementar la detección automática y el parcheo de vulnerabilidades de seguridad en archivos binarios.

La figura 7 ilustra un ambiente de cómputo de ejemplo adecuado para implementar un servidor de

distribución, un servidor de exploración de parche, y un computador de cliente.

DESCRIPCION DETALLADA

Revisión

La siguiente discusión está dirigida a sistemas y métodos que posibilitan parchear vulnerabilidad de seguridad en archivos binarios. La detección y el parcheo de archivos binarios vulnerables es automática, confiable, libre de regresión, y comprensiva a través de las redes a una escala ilimitada. Estas ventajas se pueden evidenciar de varias maneras incluyendo, por ejemplo, al apalancar infraestructura antivirus habitual que es ampliamente desplegada a través de la Internet. Una divergencia de parches de seguridad alejada de los paquetes de servicio convencionales suministra la posibilidad de producción de fijaciones libres de regresión para vulnerabilidades de seguridad en archivos binarios.

El descubrimiento confiable de archivos binarios vulnerables (por ejemplo en sistemas operativos,

programas de aplicación, etc.) se logra a través del uso de firmas binarias que han sido asociadas con vulnerabilidades de seguridad. Las firmas binarias asociadas con vulnerabilidades de seguridad en archivos binarios, junto con parches de seguridad desarrollados para fijarse a tales vulnerabilidades de seguridad, son cargadas a un servidor de distribución central. El servidor de distribución se configura para distribuir las firmas binarias y los parches de seguridad de manera amplia a través de varias redes tales como la Internet. El uso de un servidor de distribución central para actualizar los servidores de redes (por ejemplo a través de la Internet) suministra cubrimiento de parche comprensivo y automático a una escala ilimitada. Los servidores de red que reciben tales actualizaciones pueden explorar los computadores de los clientes dentro de redes subordinadas para localizar los archivos vulnerables de acuerdo con las firmas binarias, y luego actualizar aquellos computadores que se encuentre que tengan archivos de seguridad vulnerables que utilizan los parches de seguridad correspondientes que se fijarán a los archivos vulnerables. Loa servidores de red también se puede comunicar con los computadores de cliente para transferir formas binarias y parches de seguridad a los

computadores de tal forma que la exploración y actualización se pueda desarrollar por medio de los computadores mismos. Niveles multianidados de redes subordinadas también pueden existir.

Ambiente de Ejemplo

La figura 1 ilustra un ambiente de red de ejemplo 100 adecuado para implementar la detección automática y el parcheo de vulnerabilidades de seguridad en archivos binarios. En el ambiente de red del ejemplo 100, un servidor de distribución central 102 está acoplado a múltiples servidores de exploración/parcheo 104 por vía de una red 106(a). Un servidor de exploraciones/parcheo 104 está típicamente acoplado a través de una red 106(b) a una pluralidad de computadores de cliente 108(1)-108(n). La red 106 está destinada a representar cualquiera de una variedad de topologías y tipos de red convencional (que incluyen redes y ópticas, alambicas y/o inalámbricas); que emplea cualquiera de una variedad de protocolos de red convencional (que incluye protocolos públicos y/o de propietarios). La red 106 puede incluir, por ejemplo, la Internet así como también posiblemente al menos porciones de una o más redes de área local (LAN)

y/o redes de área amplia (WAN). Las redes 106(a) y 106(b) puede ser la misma red tal como la Internet, o ellas pueden ser redes aisladas unas de otras tal como la Internet y una LAN corporativa.

El servidor de distribución 102 y los servidores de exploración/parcheo 104 son típicamente implementados como servidores de red estándar, y pueden ser cada uno cualquiera de una variedad de dispositivos de cómputos convencionales, que incluyen PCs de escritorio, notebook o computadores portátiles, estaciones de trabajo, computadores gigantes, accesorios para Internet, combinaciones de estos, y así sucesivamente. Uno o más de los servidores 102 y 104 pueden ser los mismos tipos de dispositivos, o alternativamente diferentes tipos de dispositivos. Un ambiente de cómputo de ejemplo para implementar un servidor de distribución 102 y un servidor de exploración/parcheo 104 se describe con más detalle aquí adelante con referencia a la figura 7.

Los computadores de cliente 108 funcionan en una relación cliente/servidor típica con un servidor 104 en donde clientes múltiples 108 hacen solicitudes a un servidor 104 que sirve las solicitudes. Los computadores

de cliente 108 pueden ser cualquiera de una variedad de dispositivos de cómputo convencionales, incluyendo PCS de escritorio, notebook o computadores portátiles, estaciones de trabajo, computadores grandes, consolas de juego, PCs tipo agenda, teléfonos celulares u otros dispositivos de comunicación inalámbricos, ayudantes digitales personales (PDA), combinaciones de estos, y así sucesivamente. Uno o más de los computadores de cliente 108 pueden ser los mismos tipos de dispositivos, o alternativamente diferentes tipos de dispositivos. Un ambiente de cómputo de ejemplo para implementar un computador de cliente 108 se describe con más detalle aquí adelante con referencia a la figura 7.

En general, la detección automática y comprensiva y el parcheo de archivos binarios vulnerables en computadores de clientes 108 se logra a través de actualizaciones hechas a través de servidor de distribución 102 que incluye firmas binarias para identificar archivos binarios vulnerables y parches de seguridad configurados para fijarse en archivos vulnerables. Como se discute con mayor detalle adelante con respecto a las siguientes modalidades de ejemplo, las firmas binarias y los parches de seguridad son

distribuidos para explorar/parchear servidores 104 lo cual a su vez, exploran activamente y actualizan archivos binarios vulnerables en los computadores de clientes 108, o derriban las firmas binarias y los parches de seguridad a los computadores de clientes 108 de tal forma que los computadores de cliente 108 puedan desarrollar la exploración y el parcheo de los archivos binarios vulnerables.

Modalidades de Ejemplo

La figura 2 ilustra una modalidad de ejemplo de un servidor de distribución 102, un servidor de exploración-parcheo 104 y un computador de cliente 108 adecuado para implementar detección automática y parcheo de vulnerabilidades de seguridad en archivos binarios. El servidor de distribución 112 incluye un módulo de distribución 200 y una base de datos 202 para recibir y mantener firmas binarias y parches de seguridad. La base de datos 202 puede ser actualizada con firmas binarias y parches de seguridad de una variedad de manera que incluye, por ejemplo, a través de un medio de almacenamiento portátil (no mostrado, pero ver figura 7) o a través de un dispositivo de cómputo (no mostrado)

acoplado al servidor 102 y configurado para cargar firmas binarias y parches de seguridad a la base de datos 202.

Un escenario típico en el cual la base de datos 202 podría ser actualizada inicia con una investigación de un producto de software (por ejemplo un sistema operativo, un programa de aplicación, etc.) iniciado por el desarrollador del producto de software. Por ejemplo, el desarrollador puede contratar una firma de consultoría en seguridad para intentar encontrar vulnerabilidades de seguridad en un producto de software recientemente liberado. Si se descubre una vulnerabilidad en el producto de software a través de "hacking" o por algunos otros medios, el patrón de bit exacto de la función vulnerable dentro del producto puede ser identificado. El patrón de bit representa una firma binaria de la sección vulnerable en el archivo binario, que es un componente de un producto de software.

Una vez se descubre la vulnerabilidad de seguridad y se analiza, se puede desarrollar una fijación que eliminará la vulnerabilidad. Tales fijaciones son denominadas parches de seguridad y ellas representan módulos de código revisados compilados en ejecutables

binarios. Los parches de seguridad se pueden instalar en computadores que son identificados a través de la firma binaria como software que corre que tiene la vulnerabilidad de seguridad. La instalación del parche de seguridad fijará la vulnerabilidad de seguridad. El servidor de distribución 102 le posibilita a los vendedores de producto de software y a otros cargar firmas binarias de archivos binarios vulnerables junto con los parches de seguridad diseñados para fijar los archivos binarios vulnerables, en la base de datos 202 para distribución.

El módulo de distribución 200 se configura para distribuir firmas binarias y parches de seguridad de bases de datos 202 a varios servidores de exploración-parcheo 104 por vía de la re 106. El módulo de distribución 200 típicamente funciona automáticamente para distribuir firmas binarias y parches de seguridad de la base de datos 202 siempre y cuando la base de datos 202 esté actualizada con las firmas y los parches adicionales. La distribución automática se puede lograr de una variedad de maneras incluyendo, por ejemplo, a través de comunicación desde el módulo de distribución 200 a los servidores de exploración-parcheo 104 que

indican que están disponibles las firmas binarias actualizada y los parches de seguridad y esperan por solicitudes para enviar las firmas binarias y los parches de seguridad, o al enviar automáticamente firmas binarias actualizadas y parches de seguridad a los servidores de exploración-parcheo 104 configurados para aceptar las actualizaciones.

En la modalidad De la figura 2, el servidor de exploración-parcheo 104 incluye un módulo de exploración-parcheo 204 y una base de datos 206 para recibir y soportar firmas binaras y parches de seguridad. La base de datos 206 es típicamente actualizada automáticamente con nuevas firmas binarias y parches de seguridad a través de la comunicación entre el módulo exploración-parcheo 204 y el módulo de distribución 200 en el servidor de distribución 102. Además de actualizar la base de datos 206 con firmas binaras y parches de seguridad, el módulo de exploración-parcheo 204 se configura para accesar computadores de clientes 108 y explorar archivos binarios 208 para firmas binarias. Explorar archivos binarios 208 puede incluir buscar una firma binaria en archivos binarios presentes en cualquier forma de medios puentes en o accesibles por el computador

de cliente 108. Los archivos binarios 208 típicamente incluyen códigos de computador/leíbles por procesador, compilados tal como un sistema operativo o un archivo de programa de aplicación. Sin embargo, se nota que los archivos binarios 208 pueden ser cualquier forma de información binaria que incluye instrucciones de computador/leíbles por procesador, estructuras de datos, módulos de programa, y otros datos para computador de cliente 108.

Como se anotó anteriormente en la discusión en referencia al ambiente del computador de ejemplo de la figura 7, tales medios en un computador de cliente 108 pueden incluir cualquier medio disponible que sean accesibles por el computador del cliente 108, tal como medios volátiles y no volátiles así como también medios removibles y no removibles. Tales medios leíbles por computador/procesador pueden incluir memoria volátil, tal como la memoria de acceso aleatorio (RAM) y/o memoria no volátil, tal como la memoria de solo lectura (ROM). Los medios leíbles por computador/procesador pueden también incluir otros medios de almacenamiento en computador removibles/no removibles, volátiles/no volátiles, tales como, por ejemplo, unidad de disco duro para lectura

desde y escritura hacia los medios magnéticos no removibles, no volátiles, una unidad de disco magnético para lectura desde y escritura hacia un disco removible, no volátil (por ejemplo, un "floppy disk"), una unidad de disco óptico para lectura desde y/o escritura hacia un disco óptico removible, no volátil tal como un CD-ROM, DVD-ROM u otro medio óptico, otro dispositivo de almacenamiento magnético, tarjetas de memoria flash, memoria de solo lectura programable eléctricamente borrable (EEPROM), almacenamiento unido a red, y similar. Todos los medios leíbles por computador/procesador provistos con almacenamiento tanto volátil como no volátil de cualquier forma de archivos binarios 208, que incluye instrucciones leíbles por computador/procesador, estructuras de datos, módulos de programa, y otros datos para computador de cliente 108, son accesibles para exploración por medio de servidor de exploración-parcheo 104 por vía del módulo de exploración-parcheo 204.

El módulo de exploración-parcheo 204 busca así archivos binarios 208 en el computador del cliente 108 para determinar sí una firma binaria que identifica una vulnerabilidad de seguridad está presente en cualquier información binaria localizada en el computador del

cliente 108. Sí el patrón de bit de la firma binaria se encuentra en un archivo binario 208, el módulo de exploración-parcheo 204 opera para fijar la vulnerabilidad de seguridad de seguridad en el archivo binario 208 al instalar un parche de seguridad correspondiente en el computador del cliente 108. La instalación de un parche de seguridad en el computador del cliente 108 sobre escribe o de otra manera elimina el archivo binario a una porción del archivo binario que contiene la vulnerabilidad de seguridad.

La figura 3 ilustra otra modalidad de ejemplo de un servidor de distribución 102, un servidor de exploración-parcheo 104 y un computador de cliente 108 adecuado para implementar el parcheo de las vulnerabilidades de seguridad en los archivos binarios. En general, en la modalidad de la figura 3, las firmas binarias y los parches de seguridad son derribados, o redistribuidos, desde el servidor 104 del computador del cliente 108, y la exploración de archivos vulnerables por seguridad y el parcheo de los archivos vulnerables por seguridad se desarrollan mediante un computador de cliente 108 en lugar del servidor de exploración-parcheo 104.

En la modalidad de la figura 3, el servidor de distribución 102 es configurado de la misma manera como se discutió anteriormente con respecto a la modalidad de la figura 2. Así, la base de datos 202 se puede actualizar para incluir firmas binarias recientemente descubiertas que identifican vulnerabilidades de seguridad en archivos binarios. La base de datos 202 también se puede actualizar con los parches de seguridad correspondientes que han sido desarrollados para fijar tales vulnerabilidades de seguridad.

El servidor de exploración-parcheo 102 de la figura 3 se configura de alguna forma de la misma manera como aquel que se discutió anteriormente con respecto a la figura 2. Así, el servidor de exploración-parcheo 102 de la figura 3 incluye una base de datos 206 para recibir y mantener firmas binarias y parches de seguridad. La base de datos 206 es típicamente actualizado automáticamente con nuevas firmas binarias y parches de seguridad a través de las comunicaciones entre el servidor de exploración-parcheo 104 y el servidor de distribución 102. Sin embargo, la comunicación entre el servidor de exploración-parcheo 104 y el servidor de distribución 102 es conducida a través de un módulo de redistribución 300

en lugar de un módulo de exploración-parcheo 204 como se discutió con respecto a la modalidad de a figura 2.

El módulo de redistribución 300, además de actualizar la base de datos 206 con firmas binarias y parches de seguridad, se configura para comunicarse con el módulo de exploración-parcheo 302 en el computador del cliente 108 y transferir una firma binaria al computador del cliente 108. el módulo de exploración-parcheo 302 se configura para recibir la firma binaria y para explorar los archivos binarios 208 para determinar si la firma binaria está presente en cualquier información binaria localizada en el computador del cliente 108. Así, el módulo de el módulo de exploración-parcheo 302 de la figura 3 funciona de manera similar al módulo de el módulo de exploración-parcheo 204 discutido anteriormente con referencia a la figura 2.

Sí el patrón de bit de la firma binaria es encontrado en el archivo binario 208 en el computador de cliente 108, el módulo de el módulo de exploración-parcheo 302 envía una solicitud al módulo de redistribución 300 en el servidor 102. La solicitud para hacer que el módulo de redistribución 300 suspenda el

parche de seguridad que se corresponde con la firma binaria al computador del cliente 108. El módulo de redistribución 300 responde a la solicitud al enviar el parche de seguridad apropiado al computador del cliente 108. El módulo de el módulo de exploración-parcheo 302 recibe el parche de seguridad y opera para fijar la vulnerabilidad de seguridad en el archivo binario 208 al instalar el parche de seguridad en el computador del cliente 108. Como en la modalidad de la figura 2, la instalación de un parche de seguridad en el computador del cliente 108 sobre escribe o de otra manera elimina el archivo binario o una porción del archivo binario que contiene la vulnerabilidad de seguridades cubierta.

Métodos de Ejemplo

Los métodos de ejemplo para implementar la detección y el parcheo automático de vulnerabilidades de seguridad en archivos binarios será ahora descrita con referencia primaria a los diagramas de flujo de las figuras 4-6. Los métodos aplican de manera general a las modalidades de ejemplo discutidas anteriormente con respecto a las figuras 1-3. Los elementos de los métodos descritos se pueden desarrollar mediante cualquiera medios apropiados

incluyendo, por ejemplo, mediante bloques lógicos de hardware en un ASIC o mediante la ejecución de instrucciones leíbles por procesador definidas en un medios leíble por procesador.

Un "medio leíble por procesador", como se utiliza aquí, puede ser cualquiera medios que puedan contener, almacenar, comunicar, propagar, o transportar instrucciones para uso por o ejecución de un procesador. Un medio leíble por procesador puede ser, sin limitación, sistema, aparato, dispositivo o medio de propagación electrónico, magnético, óptico, electromagnético, infrarrojo o semiconductor. Ejemplos más específicos de un medio leíble por procesador incluye, entre otros, una conexión eléctrica (electrónica) que tiene uno o más alambres, un disquete portátil de computador (magnético), una memoria de acceso aleatoria (RAM) (magnético), una memoria de solo lectura (ROM) (magnético), una memoria de solo lectura programable borrable (EPROM o memoria Flash), una fibra óptica (óptica) un disco compacto reescribible (CD-RW) (óptico), y una memoria de solo lectura de disco compacto portátil (CDROM) (óptico).

La figura 4 muestra un método de ejemplo 400 para implementar la detección y el parcheo automáticos de vulnerabilidades de seguridad en archivos binarios. Los archivos binarios están típicamente localizados o almacenados en un computador de cliente que está siendo servido por un computador servidor, pero ellos también pueden estar localizados en el computador servidor mismo, o en cualquier otro dispositivo de cómputo accesible por medio de un computador servidor. En el bloque 402 del método 400 se recibe una firma binaria. La firma binaria es un patrón de bit que se ha asociado con una vulnerabilidad de seguridad en un archivo binario particular, tal como un programa de aplicación ejecutable o un sistema operativo que corre en un computador de cliente. La firma binaria se recibe de un servidor de distribución central 102 mediante un servidor subordinado 104.

En el bloque 404, se recibe un parche de seguridad. El parche de seguridad es típicamente código ejecutable compilado que ha sido desarrollado como un fijo a la vulnerabilidad de seguridad del archivo binario particular. El parche de seguridad también es recibido del servidor de distribución central 102 mediante un

servidor subordinado 104. En el bloque 406, se identifica un archivo binario vulnerable basado en la firma binaria. La identificación del archivo binario vulnerable es típicamente lograda mediante información binaria explorable almacenada en medios varios de un computador, tal como un computador de cliente 108, y luego comparar el o los patrones en la firma binaria con la información binaria encontrada en los medios. La identificación puede ocurrir de varias maneras incluyendo, por ejemplo, mediante el servidor 104 que explora y compara toda la información binaria presente en el computador del cliente. La identificación de un archivo binario vulnerable también se puede lograr al tener el servidor 104 derribada la firma binaria al computador del cliente de tal forma que el computador del cliente puede desarrollar la exploración y comparación.

En el bloque 408 del método 400, se utiliza el parche de seguridad para actualizar el archivo binario vulnerable. La actualización se puede lograr de varias maneras incluyendo, por ejemplo, mediante el servidor 104 que instala el parche de seguridad en el computador del cliente 108. Si el computador del cliente 108 ha desarrollado la exploración y ha identificado el archivo

binario vulnerable, el computador del cliente 108 puede solicitar que el servidor 104 envíe el parche de seguridad al computador 108, en cuyo caso el computador 108 puede instalar el parche de seguridad para fijar el archivo binario vulnerable.

La figura 5 muestra otro método de ejemplo 500 para implementar detección y parcheo automáticos de vulnerabilidades de seguridad en archivos binarios. El método 500 ilustra de manera general la distribución de firmas binarias para vulnerabilidades de seguridad y parches de seguridad desarrollados para fijar aquellas vulnerabilidades de seguridad. En el bloque 502 del método 500, se recibe una firma binaria que identifica una vulnerabilidad de seguridad de un archivo binario. La firma binaria es típicamente cargada a un servidor de distribución 102 como un patrón de bit recientemente descubierto que identifica una vulnerabilidad de un archivo binario de un producto de software que puede ser ampliamente distribuido a través de muchos computadores en una red tal como la Internet. La carga celebra típicamente desde un computador acoplado al servidor de distribución 102 o desde un medio almacenable portátil insertado en el servidor de distribución 102. En el

bloque 504, se recibe un parche de seguridad configurado para fijar la vulnerabilidad de seguridad mediante el servidor de distribución 102 de una manera similar que la firma binaria.

En el bloque 506, la firma binaria y el parche de seguridad son distribuidos a una pluralidad de servidores subordinados 104 desde el servidor de distribución 102. Esta distribución ocurre automáticamente y se puede lograr de varias maneras. Por ejemplo, al recibir una firma binaria cargada y el parche de seguridad, el servidor de distribución 102 puede automáticamente despachar la firma binaria y el parche de seguridad sobre la red a todos los servidores subordinados 104 configurados para recibir las firmas binarias actualizadas y los parches de seguridad. El servidor de distribución 102 también podría enviar una nota a los servidores 104 indicando que se ha descubierto una vulnerabilidad de seguridad y que el parche de seguridad fue disponible para fijar la vulnerabilidad. Los servidores subordinados 104 pueden solicitar entonces que el servidor de distribución 102 envíe la firma binaria que identifica la vulnerabilidad de seguridad y el parche de seguridad. Al recibir una solicitud, el servidor de

distribución 102 puede enviar la firma binaria y el parche de seguridad a los servidores que lo solicitan 102.

La figura 6 muestra otro método de ejemplo 600 para implementar la detección automática y el parcheo de las vulnerabilidades de seguridad n los archivos binarios. El bloque 602 del método 600, un computador de cliente 108 recibe una firma binaria de un servidor 104. La firma binaria está asociada con una vulnerabilidad de seguridad en un archivo binario que puede estar presente en el computador del cliente 108. En el bloque 604, el computador del cliente 108 explora toda la información binaria actualmente disponible en este y la compara con el o los patrones en la firma binaria con la información binaria, La información binaria explorada por el computador del cliente 108 está típicamente en la forma de instrucciones leíbles por computador/procesador y/o ejecutables, estructuras de datos, módulos de programa, y otros datos útiles para el computador del cliente 108, y puede residir tanto en los medios de almacenamiento volátiles como no volátiles de varios tipos.

En el bloque 606, si el computador del cliente 608 encuentra un archivo binario que contiene la firma binaria, este envía una solicitud al servidor 104 para hacer transferir el parche de seguridad. En el bloque 608, el computador del cliente 108 recibe del parche de seguridad, y en el bloque 610, del computador del cliente 108 instala el parche de seguridad con el fin de fijar la vulnerabilidad de seguridad en el archivo binario que contiene la información binaria que casa con el o los patrones de la firma binaria.

Aunque uno o más métodos se han descrito por medio de los diagramas de flujo y el texto asociado con los bloques de los diagramas de flujo, se debe entender que los bloques no necesariamente tienen que ser desarrollados en el orden en el cual ellos fueron presentados, y que un orden u órdenes alternativos pueden resultar en ventajas similares. Adicionalmente, los métodos no son exclusivos y si se pueden desarrollar solos o en combinación uno con el otro.

35

Computador de Ejemplo

La figura 7 ilustra un ambiente de cómputo de ejemplo adecuado para implementar un servidor de distribución 102, un servidor de exploración-parche 104 y un computador de cliente 108, como se discutió anteriormente con referencia a las figuras 1-3. Aunque una configuración específica se muestra e la figura 7, el servidor de distribución 102, el servidor de exploración-parcheo 104, y el computador de cliente 108 se puede implementar en otras configuraciones de cómputo.

El ambiente de cómputo 700 incluye un sistema de cómputo del propósito general en la forma de un computador 702. Los componentes del computador 702 pueden incluir, pero no están limitados a, uno o más procesadores a unidades de procesamiento 704, una memoria el sistema 706, y un bus del sistema 708 que acopla varios componentes del sistema incluyendo el procesador 704 a la memoria del sistema 706.

El bus del sistema 708 representa una o más de cualquiera de los varios tipos de estructuras de bus, incluyendo un bus de memoria o un controlador de memoria,

54

un bus periférico, un puerto de gráfico 2 acelerado, y un procesador o un bus local que utiliza cualquiera de una variedad de arquitecturas de bus. Un ejemplo de un bus del sistema 708 sería un bus de Interconexión de Componente Periférico (PCI) también conocido como bus Mezanine.

El computador 702 incluye típicamente una variedad de medios leíbles por computador. Tales medios pueden ser cualquiera medios disponibles que sean accesibles por computador 702 y que incluyen tanto medios volátiles como no volátiles, medios removibles como no removibles. La memoria del sistema 706 incluye medios leíbles por computador en la forma de memoria volátil, tal como memoria de acceso aleatorio (RAM) 710, y/o memoria no volátil, tal como memoria de solo lectura (ROM) 712. Un sistema de entrada/salida básico (BIOS) 714, que contiene las rutinas básicas que ayudan a transferir la información entre elementos dentro del computador 702, tal como durante el arranque, se almacena en la ROM 712. La RAM 710 contiene típicamente datos y/o módulos de programa que son inmediatamente accesibles a y/o actualmente operados mediante la unidad de procesamiento 704.

El computador 702 puede también incluir otros medios de almacenamiento en computador removibles/no removibles, volátiles/no volátiles. Por vía de ejemplo, la figura 7 ilustra una unidad de disco duro 714 para leer desde y escribir hacia medios magnéticos no volátiles (no mostrados), una unidad de disco magnético 718 para leer desde y escribir hacia un disco magnético removible, no volátil 720 (por ejemplo un "floppy disk"), y una unidad de disco óptico 722 para leer desde y/o escribir hacia un disco óptico removible, no volátil 724 tal como un CD-ROM, DVD-ROM, u otros medios ópticos. La unidad de disco duro 716, la unidad de disco magnético 718, la unidad de disco óptico 722 y cada uno conectado al bus del sistema 708 por uno o más interfaces de medios de datos 726. Alternativamente, la unidad de disco duro 716, la unidad de disco magnético 718, y la unidad de disco óptico 722 se pueden conectar al bus del sistema 708 mediante una interfase SCSI (no mostrada).

Las unidades de disco y sus medios leíbles por computador asociados suministran almacenamiento no volátil de instrucciones leíbles por computador, estructuras de datos, módulos de programa, y otros datos

para computador 702. Aunque el ejemplo ilustra un disco duro 716, un disco magnético removible 720, y un disco óptico removible 724, se debe apreciar que otros tipos de medios leíbles por computador que pueden almacenar datos que son accesibles por un computador tal como cartuchos magnéticos y otros dispositivos de almacenamiento magnético, tarjetas de memoria flash, CD-ROM, discos versátiles digitales (DVD) u otros almacenamientos ópticos, memorias de acceso aleatorio (RAM), memorias de solo lecturas (ROM), memorias de solo lectura programables eléctricamente borrables (EEPROM), y similares, también Se pueden utilizar para implementar el sistema de cómputo de ejemplo y el ambiente.

Cualquier número de módulos de programa se puede almacenar en el disco duro 716, el disco magnético 720, el disco óptico 724, ROM 712, y/o RAM 710, incluyendo por ejemplo, un sistema operativo 726, uno o más programas de aplicación 728, otros módulos de programa 730, y datos de programa 732. Cada uno de tales sistemas operativos 726 uno o más programas de aplicación 728, otros módulos de programa 730, y los datos de programa 732 (o alguna combinación de estos) puede incluir una modalidad de un

esquema de caché para utilizar información de acceso a redes.

El computador 702 puede incluir una variedad de medios leíbles por computador/procesador identificados como medios de comunicación. Los medios de comunicación típicamente son modalidades de instrucciones leíbles por computador, estructuras de datos, módulos de programa, u otros datos en señales de datos modulados tales como una onda portadora u otro mecanismo de transporte e incluye cualquier medio de suministro de información. El término "señal de datos modulada" significa una señal que tiene una o más de sus características establecida o cambiada de tal manera que codifica la información en la señal. Por vía de ejemplo, y no de limitación, los medios de comunicación incluyen medios alámbricos tales como redes alámbricas o conexiones alámbricas directas, y medios inalámbricos tales como medios acústicos, RF, infrarrojos, y otros medios inalámbricos. Las combinaciones de cualquiera de los anteriores también están incluidos dentro del alcance de los medios leíbles por computador.

Un usuario puede ingresar comandos de información dentro de un sistema de un sistema de computador 702 por vía de los dispositivos de entrada tales como un teclado 734 y un dispositivo de punteo 736 (por ejemplo, un "ratón"). Otros dispositivos de entrada 738 (no específicamente mostrados) pueden incluir un micrófono, joistick, game pad, disco satelital, puerto serial, explorador, y/o similar. Estos y otros dispositivos de entrada están conectados a la unidad de procesamiento 704 por vía de las interfases entrada/salida 740 que están acopladas al bus del sistema 708, pero que puede estar conectados mediante otra interfase y estructuras de bus, tal como un puerto paralelo, un puerto de juegos, o un bus serial universal (USB).

Un monitor 742 u otro tipo de dispositivo de exhibición se pueden conectar al bus del sistema 708 por vía de una interfase, tal como un adaptador de video 144. Además del monitor 742, otros dispositivos periféricos de salida pueden incluir componentes tales como parlantes (no mostrados) y una impresora 746 que puede estar conectada al computador 702 por vía de las interfases de entrada/salida 740.

El computador 702 puede operar en un ambiente de red utilizando conexiones lógicas a uno o más computadores remotos, tal como un dispositivo de cómputo remoto 748. Por vía de ejemplo, el dispositivo de cómputo remoto 748 puede ser un computador personal, computador portátil, un servidor, un enrutador, un computador de red, un dispositivo homólogo u otro nodo de red común, y similares. El dispositivo de computo remoto 748 se ilustra como un computador portátil que puede incluir muchos o todos los elementos y características descritas aquí con relación al sistema de computador 702.

Las conexiones lógicas entre el computador 702 y el computador remoto 748 se describen como una red de área local (LAN) 750 y una red de área amplia general (WAN) 752. Tales ambientes de red son sitios comunes en oficinas, redes de computadores amplias de empresas, intranet, y la Internet. Cuando se implementa en un ambiente de red LAN, el computador 702 está conectado a una red local 750 por vía de una interfase de red o adaptador 754. Cuando se implementa en un ambiente de red WAN, el computador 702 incluye típicamente un modem 756 u otros medios para establecer comunicación sobre una red amplia 752. El modem 756, puede ser interno o externo al

computador 702, puede estar conectado al bus del sistema 708 por vía de las interfases de entrada/salida 740 u otros mecanismos apropiados. Se debe apreciar que las conexiones de red ilustradas son ejemplos y que otros medios de establecer enlaces de comunicación entre los computadores 702 y 748 se pueden emplear.

En un ambiente de red, tal como aquel ilustrado con el ambiente del cómputo 700, los módulos del programa descritos con relación al computador 702, o porciones de estos, se pueden almacenar en un dispositivo de almacenamiento o de memoria remoto. Por vía de ejemplo, los programas de aplicación remota 758 residen en un dispositivo de memoria del computador remoto 748. Para propósitos de ilustración, los programas de aplicación y otros componentes de programa ejecutables, tales como el sistema operativo, son ilustrados aquí como bloques discretos aunque se reconoce que tales programas y componentes residen en varios tiempos en diferentes componentes de almacenamiento del sistema de computador 702, y son ejecutados por el o los procesadores de datos del computador.

Conclusión

Aunque la invención se ha descrito en lenguaje específico a las características estructurales y/o actos metodológicos, se debe entender que la invención definida en las reivindicaciones finales no está necesariamente limitada a las características específicas o actos descritos. En su lugar, las características y actos específicos son descritos como formas de ejemplo de implementar la invención reivindicada.

44

REIVINDICACIONES

1. Un medio leíble por procesador que comprende instrucciones ejecutables por procesador configuradas para:

   recibir una firma binaria;
   recibir un parche de seguridad;
   identificar un archivo binario vulnerable en un computador basado en firma binaria; y
   actualizar el archivo binario vulnerable en el computador con el parche de seguridad.

2. Un medio leíble por procesador como se cita en la reivindicación 1, en donde el identificar un archivo binario vulnerable en un computador incluye comparar un patrón de bit de las firma binaria contra los archivos binarios localizados en el computador, el patrón de bit asociado con una vulnerabilidad de seguridad.

3. Un medio leíble por procesador como se cita en la reivindicación 1, en donde la actualización del archivo binario vulnerable es en el computador

53

incluye instalar el parche de seguridad en el computador.

4. Un medio leíble por procesador como se cita en la reivindicación 1, en donde el identificar un archivo binario vulnerable en un computador incluye enviar la firma binaria al computador.

5. El medio leíble por procesador como se cita en la reivindicación 4, en donde la actualización del archivo binario vulnerable en el computador incluye:

Recibir una solicitud de un computador para enviar el parche de seguridad; y enviar el parche de seguridad al computador.

6. Un medio leíble por procesador como se cita en la reivindicación 1, en donde el computador es un computador de cliente y el recibir incluye recibir la firma binaria y el parche de seguridad desde un servidor de distribución configurado para distribuir al computador del cliente, firmas binarias que identifican archivos vulnerables y

parches de seguridad configurados para fijar los archivos vulnerables.

7. Un servidor que comprende un medio leíble por procesador como se cita e la reivindicación 1.

8. Un medio leíble por procesador que comprende instrucciones ejecutables por procesador configurados para:

recibir una firma binaria que identifica una vulnerabilidad de seguridad en un archivo binario;
recibir un parche de seguridad configurado para fijar la vulnerabilidad de seguridad en el archivo binario; y
distribuir la firma binaria y el parche de seguridad a una pluralidad de servidores.

9. El medio leíble por procesador como se cita en la reivindicación 8, en donde el distribuir incluye:
enviar una nota a cada una de una pluralidad de servidores con relación a la vulnerabilidad de seguridad y el parche disponible;

47

recibir una solicitud para enviar la firma binaria y el parche de seguridad; y

enviar la firma binaria y el parche de seguridad en respuesta a la solicitud.

10. Un servidor de distribución que comprende el medio leíble por procesador como se cita la reivindicación 8.

11. Un medio leíble por procesador que comprende instrucciones ejecutables por procesador configuradas para:

recibir una firma binaria de un servidor;

buscar la firma binaria en archivos binarios;

enviar una solicitud al servidor de un parche de seguridad si se encuentra que el archivo binario incluye la firma binaria;

recibir el parche de seguridad del servidor; y

actualizar el archivo binario con el parche de seguridad.

48

48

12. Un computador de cliente que comprende un medio leible por procesador como se cita en la reivindicación 11.

13. Un método que comprende:

recibir una firma binaria;
buscar un archivo vulnerable basado en la firma binaria;
si se encuentra un archivo vulnerable, solicitar un parche de seguridad; y
fijar el archivo vulnerable con el parche de seguridad.

14. Un método como se cita en la reivindicación 13, en donde la solicitud incluye enviar una solicitud a un servidor de un parche de seguridad, el método además comprende recibir el parche de seguridad del servidor en respuesta a la solicitud.

15. Un método como se cita en la reivindicación 14, en donde el recibir incluye recibir la firma binaria del servidor.

67

49

16. El método como se cita en la reivindicación 13, en donde el fijar incluye instalar el parche de seguridad en el computador.

17. Un método como se cita en la reivindicación 13, en donde el buscar incluye comparar la firma binaria a la información con la información binaria en un medio de almacenamiento de un computador.

18. Un método como se cita en la reivindicación 17, en donde la información binaria se selecciona del grupo que comprende:

un sistema operativo;
un archivo de programa de aplicación; y
un archivo de datos.

19. Un método como se cita en la reivindicación 17, en donde el medio de almacenamiento se selecciona del grupo que comprende:

un disco duro;

68

50

un disco floppy magnético;
un disco óptico;
una tarjeta de memoria flash;
una memoria de solo lectura programable eléctricamente borrable; y
un almacenamiento unido a red.

20. Un método que comprende:

recibir una firma binaria y un parche de seguridad de un servidor de distribución;
buscar en un computador de cliente un archivo vulnerable asociado con la firma binaria; y
si se encuentra el archivo vulnerable, fijar el archivo vulnerable con el parche de seguridad.

21. Un método como se citó en la reivindicación 20, en donde la búsqueda incluye transferir la firma binaria al computador del cliente, el computador del cliente configurado para buscar un archivo vulnerable asociado con la firma binaria.

22. Un método como se citó en la reivindicación 21, en donde el fijar incluye:

51

51

Recibir una solicitud de un computador de cliente para transferir el parche de seguridad, el computador de cliente teniendo localizado un archivo vulnerable; y

Transferir el parche de seguridad al computador del cliente en respuesta a la solicitud.

23. Un computador que comprende:

medios para recibir una firmas binaria;

medios para buscar un archivo vulnerable basados en la firma binaria;

medios para solicitar un parche de seguridad si se encuentra un archivo vulnerable; y

medios para fijar el archivo vulnerable con el parche de seguridad.

24. Un servidor que comprende:

Medios para recibir una firma binaria y un parche de seguridad de un servidor de distribución;

medios para explorar un computador de cliente en busca de un archivo vulnerable asociado con la firma binaria; y

medios para fijar el archivo vulnerable con el parche de seguridad si se encuentra el archivo vulnerable.

25. Un computador que comprende:

Información binaria;

Un módulo de exploración configurado para recibir una firma binaria y explorar la información binaria en busca de la firma binaria; y

Un módulo de parche configurado para solicitar un parche de seguridad e instalar el parche de seguridad si la firma binaria se encuentra n la información binaria.

26. Un computador como se cita en la reivindicación 25, que comprende además un medio de almacenamiento configurado para retener la información binaria.

27. Un computador como se cita en la reivindicación 25, en donde la información binaria se selecciona del grupo que comprende:

53

un sistema operativo;

un archivo de programa de aplicación; y

un archivo de datos.

28. Un computador que comprende:

archivos binarios;

una firma binaria; y

un módulo de parche de seguridad configurado para recibir una firma binaria de un servidor y para explorar los archivos binarios en la búsqueda de la firma binaria.

29. Un computador como se cita en la reivindicación 28, que comprende además:

un archivo binario que incluye la firma binaria; y

un parche de seguridad;

en donde el módulo de parche de seguridad está configurado además para solicitar el parche de seguridad del servidor luego de localizar la firma binaria dentro del archivo binario, y

52

54

aplicar el parche de seguridad al archivo binario.

30. Un servidor de distribución que comprende:

una base de datos; y

un módulo de distribución configurado para recibir una firma binaria y un parche de seguridad, almacenar la firma binaria y el parche de seguridad en la base de datos, y distribuir la firmas binaria y el parche de seguridad a una pluralidad de servidores.

31. Un servidor de distribución como se cita en la reivindicación 30, en donde el módulo de distribución es además configurado para recibir una solicitud de un servidor de la firma binaria y el parche de seguridad y distribuir la firma binaria y el parche de seguridad al servidor en respuesta a la solicitud.

32. Un servidor que comprende:

73

55

una firma binaria asociada con una vulnerabilidad de seguridad en un archivo binario;

un parche de seguridad configurado para fijar la vulnerabilidad de seguridad en el archivo binario; y

un módulo de exploración configurado para explorar los archivos binarios en un computador de cliente en busca de la firma binaria y actualizar el archivo binario con el parche de seguridad si se encuentra la firma binaria.

33. Un servidor como se cita en la reivindicación 32, que comprende además:

una base de datos;

un módulo de exploración configurado además para recibir la firma binaria y el parche de seguridad de un servidor de distribución y almacenar la firma binaria y el parche de seguridad en la base de datos.

56

67547

RESUMEN

Se describe el sistema y métodos para posibilitar el parcheo de vulnerabilidades de seguridad en archivos binarios. La detección y el parcheo de los archivos binaros vulnerables es automática, confiable, libre de regresión, y comprensiva a través de redes a escala ilimitadas. Estas ventajas se pueden evidenciar de varias maneras que incluyen, por ejemplo, al apalancar infraestructuras antivirus habituales que son ampliamente despegadas a través de la Internet. Descubrimiento confiable de archivos binarios vulnerables (por ejemplo en sistemas operativos, programas de aplicación, etc.) que se logra a través del uso de firmas binarias que han estado asociadas con vulnerabilidades de seguridades cubiertas. Una divergencia de parches de seguridad alejada de los paquetes de servicio convencional que suministra la posibilidad de producción de fijos libres de regresión para vulnerabilidad de seguridad e archivos binarios.

75

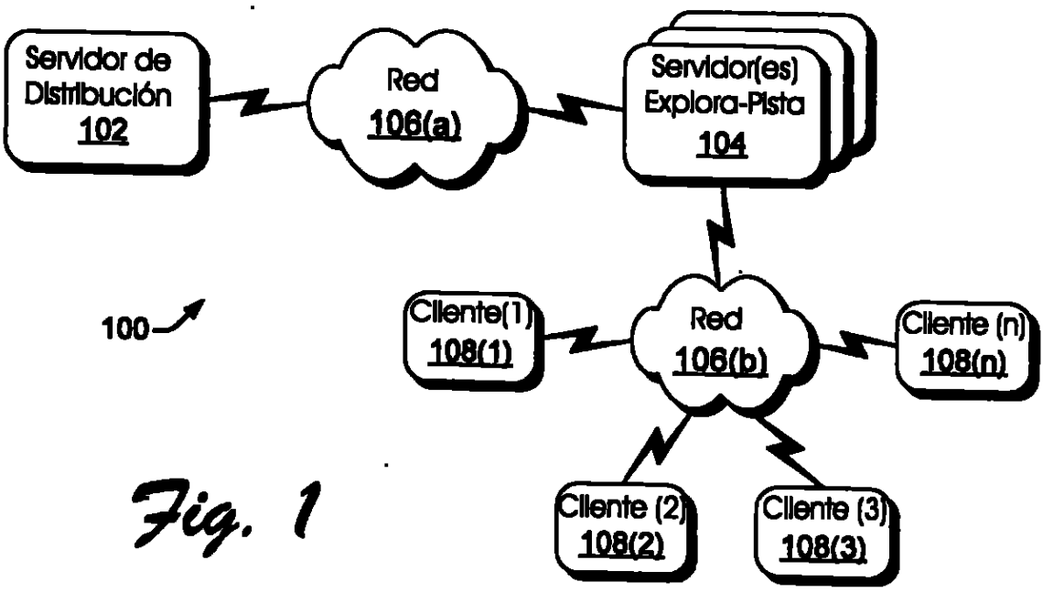

Servidor de Distribución 102
Red 106(a)
Servidor(es) Explora-Pista 104
100
Cliente(1) 108(1)
Red 106(b)
Cliente (n) 108(n)
Cliente (2) 108(2)
Cliente (3) 108(3)

Fig. 1

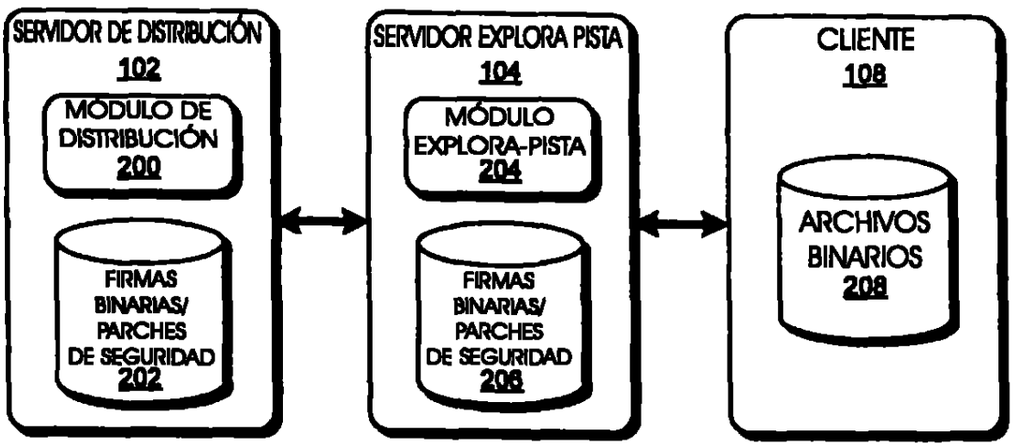

SERVIDOR DE DISTRIBUCIÓN
102
MÓDULO DE DISTRIBUCIÓN
200
FIRMAS BINARIAS/ PARCHES DE SEGURIDAD
202
SERVIDOR EXPLORA PISTA
104
MÓDULO EXPLORA-PISTA
204
FIRMAS BINARIAS/ PARCHES DE SEGURIDAD
206
CLIENTE
108
ARCHIVOS BINARIOS
208

Fig. 2

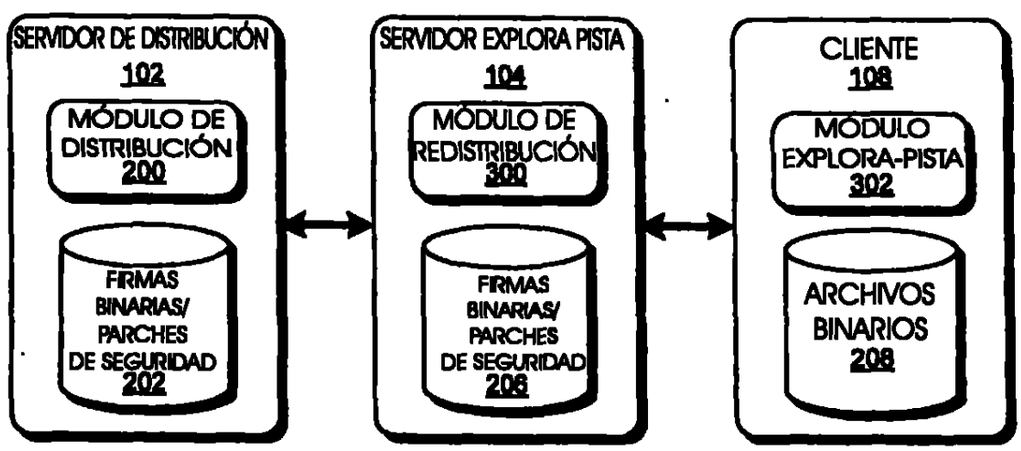

SERVIDOR DE DISTRIBUCIÓN
102
MÓDULO DE DISTRIBUCIÓN
200
FIRMAS BINARIAS/ PARCHES DE SEGURIDAD
202
SERVIDOR EXPLORA PISTA
104
MÓDULO DE REDISTRIBUCIÓN
300
FIRMAS BINARIAS/ PARCHES DE SEGURIDAD
206
CLIENTE
108
MÓDULO EXPLORA-PISTA
302
ARCHIVOS BINARIOS
208

Fig. 3

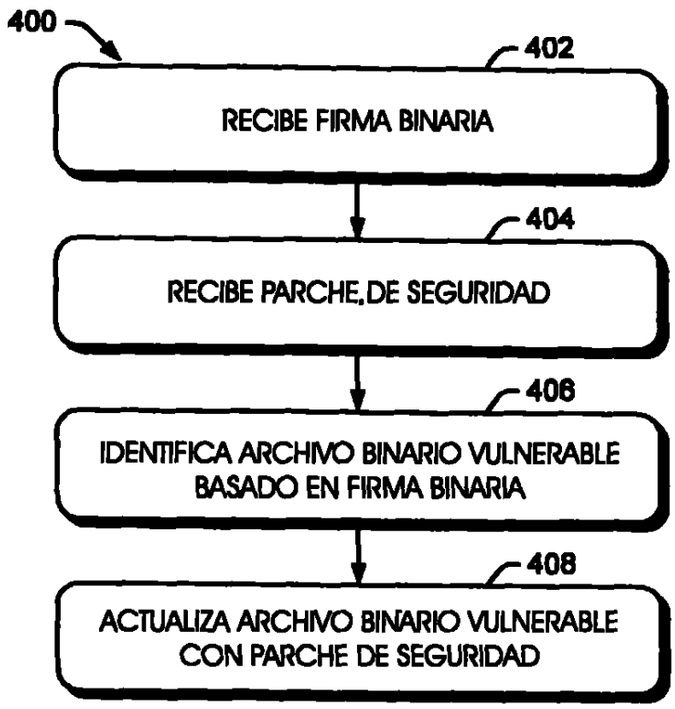

400
402
RECIBE FIRMA BINARIA
404
RECIBE PARCHE DE SEGURIDAD
406
IDENTIFICA ARCHIVO BINARIO VULNERABLE BASADO EN FIRMA BINARIA
408
ACTUALIZA ARCHIVO BINARIO VULNERABLE CON PARCHE DE SEGURIDAD

Fig. 4


500
502
RECIBE FIRMA BINARIA QUE IDENTIFICA VULNERABILIDAD DE SEGURIDAD
504
RECIBE PARCHE DE SEGURIDAD CONFIGURADO PARA ARREGLAR VULNERABILIDAD DE SEGURIDAD
506
DISTRIBUYE FIRMA BINARIA Y PARCHE DE SEGURIDAD A PLURALIDAD DE SERVIDORES
-ENVIAR NOTA DE VULNERABILIDAD Y PARCHE
-RECIBE SOLICITUD PARA FIRMA Y PARCHE
-ENVÍA FIRMA Y PARCHE

Fig. 5

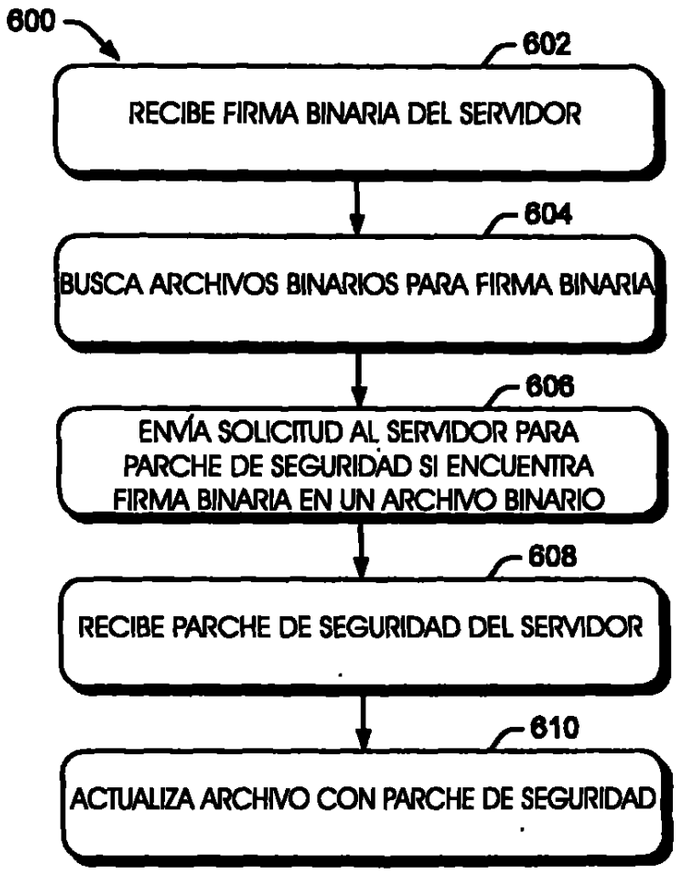

600
602
RECIBE FIRMA BINARIA DEL SERVIDOR
604
BUSCA ARCHIVOS BINARIOS PARA FIRMA BINARIA
606
ENVÍA SOLICITUD AL SERVIDOR PARA PARCHE DE SEGURIDAD SI ENCUENTRA FIRMA BINARIA EN UN ARCHIVO BINARIO
608
RECIBE PARCHE DE SEGURIDAD DEL SERVIDOR
610
ACTUALIZA ARCHIVO CON PARCHE DE SEGURIDAD

Fig. 6

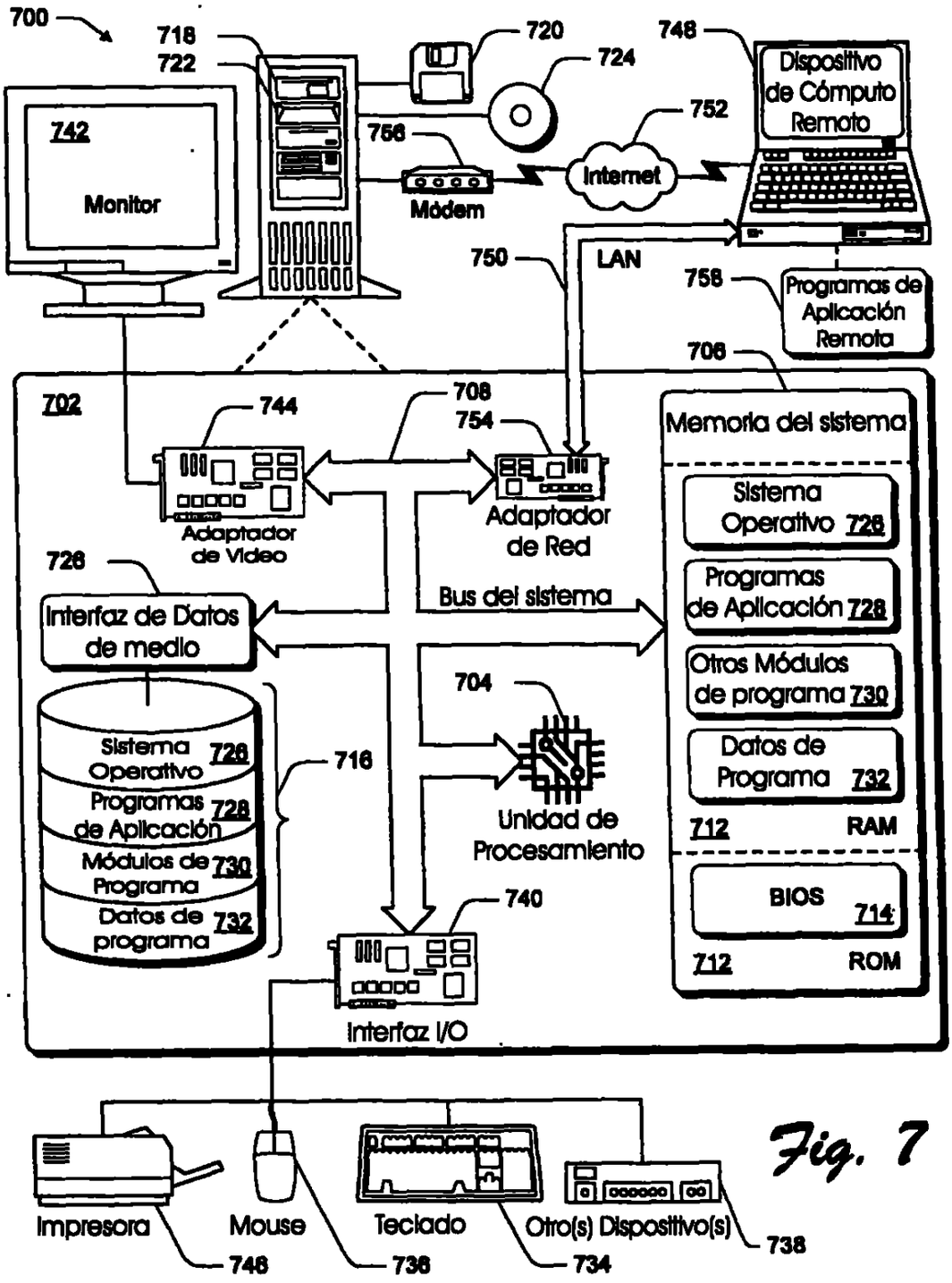

700
718
722
720
724
748
Dispositivo de Cómputo Remoto
752
742
Monitor
756
Módem
Internet
750
LAN
758
Programas de Aplicación Remota
706
702
744
708
754
Memoria del sistema
Adaptador de Video
Adaptador de Red
Sistema Operativo 726
Programas de Aplicación 728
Otros Módulos de programa 730
Datos de Programa 732
728
Interfaz de Datos de medio
Bus del sistema
704
Sistema Operativo 726
Programas de Aplicación 728
Módulos de Programa 730
Datos de programa 732
716
Unidad de Procesamiento
712 RAM
BIOS 714
712 ROM
740
Interfaz I/O
Impresora
Mouse
Teclado
Otro(s) Dispositivo(s)
738
746
736
734

Fig. 7

62

PA 1179806

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

June 07, 2004

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.

APPLICATION NUMBER: ***10/621,148***
FILING DATE: ***July 16, 2003***

By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

L. Edelen

L. EDELEN
Certifying Officer

NOTARIA 16
CARLOS MARTIN LEDESMA D.G.
OTTO BARRIOS GALVIS
3 JUL. 2004

60

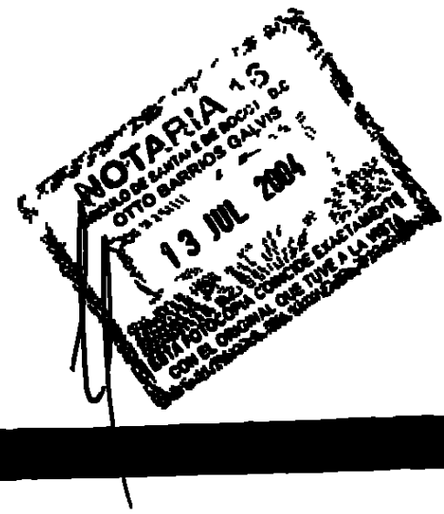
63

PATENT APPLICATION SERIAL NO _____

U S DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

07/21/2003 SDENBOB1 00000021 120769 10621148

01 FC 1001 750.00 DA
02 FC:1202 234.00 DA
03 FC 1201 672.00 DA



PTO-1556
(5/87)

*U.S. Government Printing Office: 2002 — 489-357/60033

91

64

07/16/03

Please type a plus sign (+) inside this box → ⊞

PTO/SB/05 (03-01)
Approved for use through 10/31/2002. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.	MS1-1594US
First Inventor	Ivanov
Title	Automatic Detection And Patching Of Vulnerable Files
Express Mail Label No.	EV251222109

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

1. ☑ Fee Transmittal Form (e.g., PTO/SB/17) (Submit an original and a duplicate for fee processing)
2. ☐ Applicant claims small entity status. See 37 CFR 1.27.
3. ☑ Specification (preferred arrangement set forth below) [Total Pages 32]
   - Descriptive title of the invention
   - Cross Reference to Related Applications
   - Statement Regarding Fed sponsored R & D
   - Reference to sequence listing, a table, or a computer program listing appendix
   - Background of the Invention
   - Brief Summary of the Invention
   - Brief Description of the Drawings (if filed)
   - Detailed Description
   - Claim(s)
   - Abstract of the Disclosure
4. ☑ Drawing(s) (35 U.S.C. 113) [Total Sheets 5]
5. Oath or Declaration [Total Pages 2]
   a. ☑ Newly executed (original or copy)
   b. ☐ Copy from a prior application (37 CFR 1.63 (d)) (for continuation/divisional with Box 18 completed)
      i. ☐ DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
6. ☐ Application Data Sheet. See 37 CFR 1.76

ADDRESS TO: Mail Stop Patent Application, Commissioner for Patents/P.O. Box 1450, Alexandria, VA 22313-1450

7. ☐ CD-ROM or CD-R in duplicate, large table or Computer Program (Appendix)
8. Nucleotide and/or Amino Acid Sequence Submission (If applicable, all necessary)
   a. ☐ Computer Readable Form (CRF)
   b. Specification Sequence Listing on:
      i. ☐ CD-ROM or CD-R (2 copies); or
      ii. ☐ paper
   c. ☐ Statements verifying identity of above copies

17364 U.S. PTO 10/621148

ACCOMPANYING APPLICATION PARTS

9. ☑ Assignment Papers (cover sheet & document(s))
10. ☐ 37 CFR 3.73(b) Statement (when there is an assignee) ☑ Power of Attorney
11. ☐ English Translation Document (if applicable)
12. ☑ Information Disclosure Statement (IDS)/PTO-1449 ☑ Copies of IDS Citations
13. ☐ Preliminary Amendment
14. ☑ Return Receipt Postcard (MPEP 503) (Should be specifically itemized)
15. ☐ Certified Copy of Priority Document(s) (if foreign priority is claimed)
16. ☐ Nonpublication Request under 35 U.S.C. 122 (b)(2)(B)(i). Applicant must attach form PTO/SB/35 or its equivalent.
17. ☐ Other: .....

18. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment, or in an Application Data Sheet under 37 CFR 1.76:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: ____ / ____

Prior application information: Examiner ____ Group Art Unit: ____

For CONTINUATION OR DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

19. CORRESPONDENCE ADDRESS

☑ Customer Number or Bar Code Label [redacted] or ☐ Correspondence address below

Name	22801 PATENT TRADEMARK OFFICE				
Address					
City		State		Zip Code	
Country		Telephone		Fax	

Name (Print/Type)	Nathan R. Rieth	Registration No. (Attorney/Agent)	44,302
Signature	[signature]	Date	[illegible]

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

[redacted]

10

65

EV20177271U4

07/16/03

PTO/SB/17 (01-03)
Approved for use through 04/30/2003. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL for FY 2003

Effective 01/01/2003. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT	($) 1,696.00

Complete if Known	
Application Number	
Filing Date	
First Named Inventor	Ivanov
Examiner Name	
Art Unit	
Attorney Docket No.	MS1-1594US

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None

☑ Deposit Account:

Deposit Account Number	12-0769
Deposit Account Name	Lee & Hayes, PLLC

The Commissioner is authorized to: *(check all that apply)*

☑ Charge fee(s) indicated below ☑ Credit any overpayments

☑ Charge any additional fee(s) during the pendancy of this application

☐ Charge fee(s) indicated below, except for the filing fee

to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	750	2001	375	Utility filing fee	750
1002	330	2002	165	Design filing fee	
1003	520	2003	260	Plant filing fee	
1004	750	2004	375	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

SUBTOTAL (1) ($) 750.00

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

	Extra Claims		Fee from below		Fee Paid
Total Claims	33	-20** = 13	x 18	=	234
Independent Claims	11	- 3** = 8	x 84	=	672
Multiple Dependent				=	

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	84	2201	42	Independent claims in excess of 3
1203	280	2203	140	Multiple dependent claim, if not paid
1204	84	2204	42	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 906.00

**or number previously paid, if greater; For Reissues, see above

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for ex parte reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	410	2252	205	Extension for reply within second month	
1253	930	2253	465	Extension for reply within third month	
1254	1,450	2254	725	Extension for reply within fourth month	
1255	1,970	2255	985	Extension for reply within fifth month	
1401	320	2401	160	Notice of Appeal	
1402	320	2402	160	Filing a brief in support of an appeal	
1403	280	2403	140	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,300	2453	650	Petition to revive - unintentional	
1501	1,300	2501	650	Utility issue fee (or reissue)	
1502	470	2502	235	Design issue fee	
1503	630	2503	315	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	40
1809	750	2809	375	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	750	2810	375	For each additional invention to be examined (37 CFR 1.129(b))	
1801	750	2801	375	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid SUBTOTAL (3) ($) 40.00

SUBMITTED BY				(Complete if applicable)	
Name (Print/Type)	Nathan R. Rieth	Registration No. (Attorney/Agent)	44,302	Telephone	(509) 324-9256
Signature	Nathan R. Rieth			Date	7/16/03

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS.

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

[redacted]

101

66

EV251222109

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

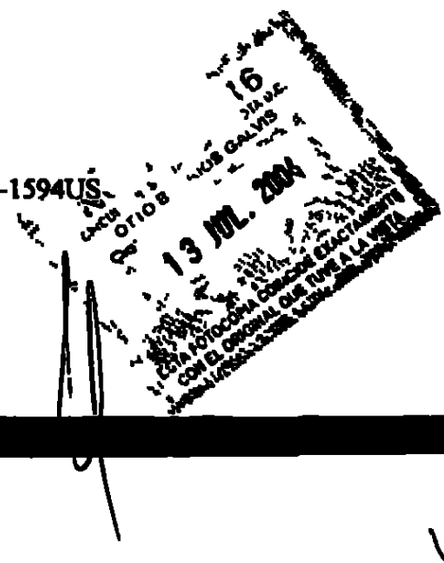
Automatic Detection And Patching Of Vulnerable Files

Inventors

Oleg Ivanov

Sergei Ivanov

ATTORNEY'S DOCKET NO MS1-1594US



TECHNICAL FIELD

The present disclosure generally relates to patching files, and more particularly, to an automatic, comprehensive, reliable and regression-free way of providing security patches for vulnerable binary program files in distributed, heterogeneous computing environments

BACKGROUND

Software development is an ongoing process whereby a software product initially released to the public can be continually updated through revisions from a software developer/vendor Software revisions are typically disbursed from a software vendor in what are called "service packs" that can be downloaded or ordered from a vendor for installation on a user's computer Service packs typically contain program fixes (e g , for an operating system, application program, etc) that repair problems (i e , "bugs") discovered in the program code after the initial release of the product or after the last service pack release

In addition to containing fixes for program bugs, service packs can also contain security patches developed specifically to repair vulnerabilities found in program files Program vulnerabilities discovered after a software product is released can pose significant security threat of attack from hackers and viruses on a world-wide basis Therefore, once a vulnerability is discovered, the prompt and wide-spread distribution and installation of security patches to computers having vulnerable software is of paramount importance Theoretically, the use of service packs to achieve such prompt and wide-spread distribution of security patches could be effective For example, when a software vendor discovers a vulnerability and then develops a security patch, the patch can be posted in the latest service

pack on a vendor Web site for users to immediately download and install This could thwart most hackers and viruses that are intent on exploiting the discovered vulnerability However, system administrators and other software product users currently face several drawbacks and/or difficulties related to accessing and installing security patches These difficulties typically result in a significantly lower distribution of such patches than is intended by the vendor who develops the patch The result is that vulnerabilities on many computers world-wide are left un-patched, exposing such computers to significant risk.

One difficulty with accessing and installing security patches is that current methods for detecting whether a computer is running software with a known vulnerability require the active use and involvement of the computer For example, currently available methods can determine whether particular versions of software products on a computer are in need of being updated (e g , with a security patch) However, only those software products actively running on the computer are included in this determination Secondary operating systems and applications that are not actively running on a computer are not considered, and therefore may have a security vulnerability that goes un-noticed and un-fixed For those products actively running on a computer, a user can review a list of available updates and select updates for installation Some updates may be critical updates designed to protect a computer from known security vulnerabilities Various updates require a user to restart the computer before the installation is complete In addition, a user must actively select the updates and install them For these and other reasons, current methods for accessing and installing security patches are less than effective

Another difficulty in accessing and installing security patches is that of knowing whether or not a security patch is needed on a computer It is sometimes difficult for users to know if their computers are running software that is vulnerable Furthermore, current methods for detecting whether a computer is running software with a known vulnerability may not be able to detect certain configurations of a software product known to be vulnerable For example, shared versions of some software products can be distributed as part of other products Thus, although a shared version of a product may contain the same vulnerability as the full version of the product, the shared version may not be recognized as a product that needs a security patch update Thus, shared versions of software products that are known to have security vulnerabilities often go un-fixed

Other problems with accessing and installing security patches relate to the conventional "service pack" method by which such patches are delivered Downloading and installing services packs is a time intensive and manual process that many system administrators simply do not have time to perform Therefore, even when administrators intend to install security patches, the time between the release of a security patch and its installation on a given system can be weeks, months, or years Thus, the risk of attack through a security vulnerability may not be alleviated in such systems until long after the software vendor has issued a security patch

Furthermore, system administrators often choose not to download and install service packs containing security patches, even though they understand the relevant security risks The reason for this is that the installation of a service pack itself brings the risk of system regressions that can introduce unwanted changes in system behavior Administrators often devote significant time and effort toward

debugging a system so that it functions as desired As mentioned above, however, service packs represent an evolution of a previous version of a software product that includes the most recent updates to a product's code base (i e , the scope of changes is not restricted to security patches only) In addition to introducing new and intended behaviors into a system, recent code updates in a service pack may introduce unknown bugs into a system that can cause the system to behave unexpectedly, which, in turn, can create significant problems for a system administrator Thus, systems frequently are not updated with important security patches intended to fix vulnerable program files, because administrators do not want to risk regressions

Accordingly, a need exists for a way to implement patching of security vulnerabilities in program files in an automatic, comprehensive, reliable and regression-free manner

SUMMARY

Automatic, comprehensive, reliable and regression-free security patching of binary program files is described herein

In accordance with one implementation, a binary signature of a vulnerability and a security patch are received. A vulnerable binary file is identified on a computer based on the binary signature of a vulnerability The vulnerable binary file on the computer is updated with the security patch

In accordance with another implementation, a binary signature is received that identifies a security vulnerability in a binary file. A security patch configured to fix the security vulnerability is also received The binary signature and the security patch are distributed to a plurality of servers

V06

In accordance with another implementation, a binary signature is received from a server and used to search binary files A request for a security patch is sent to the server if the binary signature is found in a binary file The binary file is then updated with the security patch

BRIEF DESCRIPTION OF THE DRAWINGS

The same reference numerals are used throughout the drawings to reference like components and features

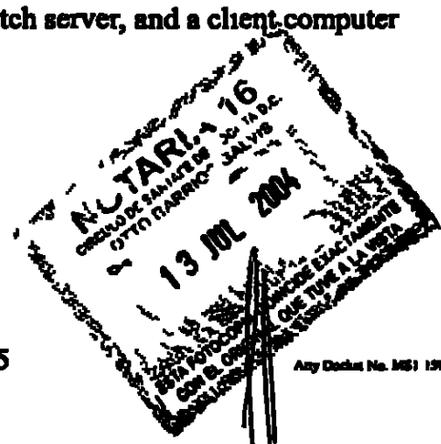
Fig 1 illustrates an exemplary network environment suitable for implementing automatic detection and patching of security vulnerabilities in binary files

Fig 2 illustrates an exemplary embodiment of a distribution server, a scan-patch server, and a client computer suitable for implementing automatic detection and patching of security vulnerabilities in binary files

Fig 3 illustrates another exemplary embodiment of a distribution server, a scan-patch server, and a client computer suitable for implementing automatic detection and patching of security vulnerabilities in binary files

Figs 4 - 6 illustrate block diagrams of exemplary methods for implementing automatic detection and patching of security vulnerabilities in binary files

Fig 7 illustrates an exemplary computing environment suitable for implementing a distribution server, a scan-patch server, and a client computer



DETAILED DESCRIPTION

Overview

The following discussion is directed to systems and methods that enable patching of security vulnerabilities in binary files The detection and patching of vulnerable binary files is automatic, reliable, regression free, and comprehensive across networks on an unlimited scale These advantages can be realized in various ways including, for example, by leveraging current anti-virus infrastructure that is widely deployed across the Internet A divergence of security patches away from conventional service packs provides for the possibility of production of regression-free fixes for security vulnerabilities in binary files

Reliable discovery of vulnerable binary files (e g , in operating systems, application programs, etc) is achieved through the use of binary signatures that have been associated with security vulnerabilities Binary signatures associated with security vulnerabilities in binary files, along with security patches developed to fix such security vulnerabilities, are uploaded to a central distribution server The distribution server is configured to distribute the binary signatures and security patches on a wide-scale basis across various networks such as the Internet. Use of a central distribution server to update network servers (e g , across the Internet) provides comprehensive and automatic patch coverage on an unlimited scale Network servers receiving such updates can scan client computers within subordinate networks to locate vulnerable files according to binary signatures, and then update those computers found to have security vulnerable files using corresponding security patches that will fix the vulnerable files Network servers can also communicate with client computers to transfer binary signatures and security patches to the computers so that the scanning and updating can be

performed by the computers themselves Multiple nested levels of subordinate networks may also exist.

Exemplary Environment

Fig 1 illustrates an exemplary network environment 100 suitable for implementing automatic detection and patching of security vulnerabilities in binary files In the exemplary network environment 100, a central distribution server 102 is coupled to multiple scan/patch servers 104 via a network 106(a) A scan/patch server 104 is typically coupled through a network 106(b) to a plurality of client computers 108(1) - 108(n) Network 106 is intended to represent any of a variety of conventional network topologies and types (including optical, wired and/or wireless networks), employing any of a variety of conventional network protocols (including public and/or proprietary protocols) Network 106 may include, for example, the Internet as well as possibly at least portions of one or more local area networks (LANs) and/or wide area networks (WANs) Networks 106(a) and 106(b) may be the same network such as the Internet, or they may be networks isolated from one another such as the Internet and a corporate LAN

Distribution server 102 and scan/patch servers 104 are typically implemented as standard Web servers, and can each be any of a variety of conventional computing devices, including desktop PCs, notebook or portable computers, workstations, mainframe computers, Internet appliances, combinations thereof, and so on One or more of the servers 102 and 104 can be the same types of devices, or alternatively different types of devices An exemplary computing environment for implementing a distribution server 102 and a scan/patch server 104 is described in more detail herein below with reference to Fig 7

Client computers 108 function in a typical client/server relationship with a server 104 wherein multiple clients 108 make requests to a server 104 that services the requests Client computers 108 can be any of a variety of conventional computing devices, including desktop PCs, notebook or portable computers, workstations, mainframe computers, gaming consoles, handheld PCs, cellular telephones or other wireless communications devices, personal digital assistants (PDAs), combinations thereof, and so on One or more of the client computers 108 can be the same types of devices, or alternatively different types of devices An exemplary computing environment for implementing a client computer 108 is described in more detail herein below with reference to Fig 7

In general, automatic and comprehensive detection and patching of vulnerable binary files on client computers 108 is achieved through updates made through distribution server 102 that include binary signatures for identifying vulnerable binary files and security patches configured to fix vulnerable files As discussed in greater detail below with respect to the following exemplary embodiments, the binary signatures and security patches are distributed to scan/patch servers 104 which in turn, either actively scan for and update vulnerable binary files on client computers 108, or push the binary signatures and security patches down to the client computers 108 so the client computers 108 can perform the scanning for and patching of vulnerable binary files

Exemplary Embodiments

Fig 2 illustrates an exemplary embodiment of a distribution server 102, a scan-patch server 104 and a client computer 108 suitable for implementing automatic detection and patching of security vulnerabilities on binary files

Distribution server 102 includes a distribution module 200 and a database 202 for receiving and holding binary signatures and security patches Database 202 can be updated with binary signatures and security patches in a variety of ways including, for example, through a portable storage medium (not shown, but see Fig 7) or through a computer device (not shown) coupled to the server 102 and configured to upload binary signatures and security patches to database 202

A typical scenario in which a database 202 might be updated begins with an investigation of a software product (e g , a operating system, application program, etc) initiated by the developer of the software product For example, a developer may hire a security consultancy firm to attempt to find security vulnerabilities in a newly released software product If a security vulnerability is discovered in a software product through hacking or by some other means, an exact bit pattern of the vulnerable function within the product can be identified The bit pattern represents a binary signature of the vulnerable section in the binary file, which is a component of a software product.

Once a security vulnerability is discovered and analyzed, a fix can be developed that will eliminate the vulnerability Such fixes are called security patches and they represent revised code modules compiled into binary executables. Security patches can be installed on computers that are identified through the binary signature as running software that has the security vulnerability Installation of the security patch will fix the security vulnerability The distribution server 102 enables software product vendors and others to upload binary signatures of vulnerable binary files along with the security patches designed to fix the vulnerable binary files, into the database 202 for distribution

Distribution module 200 is configured to distribute binary signatures and security patches from database 202 to various scan-patch servers 104 via a network 106. Distribution module 200 typically functions automatically to distribute binary signatures and security patches from database 202 whenever the database 202 is updated with additional signatures and patches. Automatic distribution may be achieved in a variety of ways including, for example, through communication from distribution module 200 to scan-patch servers 104 indicating that updated binary signatures and security patches are available and waiting for requests to send the binary signatures and security patches, or by automatically forwarding updated binary signatures and security patches to scan-patch servers 104 configured to accept the updates.

In the embodiment of Fig. 2, a scan-patch server 104 includes a scan-patch module 204 and a database 206 for receiving and holding binary signatures and security patches. Database 206 is typically updated automatically with new binary signatures and security patches through communications between the scan-patch module 204 and the distribution module 200 on distribution server 102. In addition to updating database 206 with binary signatures and security patches, scan-patch module 204 is configured to access client computer 108 and scan binary files 208 for binary signatures. Scanning binary files 208 can include searching for a binary signature in binary files present on any form of media present on or accessible by client computer 108. Binary files 208 typically include compiled, computer/processor-readable code such as an operating system or an application program file. However, it is noted that binary files 208 can be any form of binary information including computer/processor-readable instructions, data structures, program modules, and other data for client computer 108.

As noted below in the discussion referring to the exemplary computer environment of Fig. 7, such media on a client computer 108 can include any available media that is accessible by client computer 108, such as volatile and non-volatile media as well as removable and non-removable media. Such computer/processor-readable media can include volatile memory, such as random access memory (RAM) and/or non-volatile memory, such as read only memory (ROM). Computer/processor-readable media can also include other removable/non-removable, volatile/non-volatile computer storage media, such as, for example, a hard disk drive for reading from and writing to a non-removable, non-volatile magnetic media, a magnetic disk drive for reading from and writing to a removable, non-volatile magnetic disk (e.g., a "floppy disk"), an optical disk drive for reading from and/or writing to a removable, non-volatile optical disk such as a CD-ROM, DVD-ROM, or other optical media, other magnetic storage devices, flash memory cards, electrically erasable programmable read-only memory (EEPROM), network-attached storage, and the like. All such computer/processor-readable media providing both volatile and non-volatile storage of any form of binary files 208, including computer/processor-readable instructions, data structures, program modules, and other data for client computer 108, is accessible for scanning by scan-patch server 104 via scan-patch module 204.

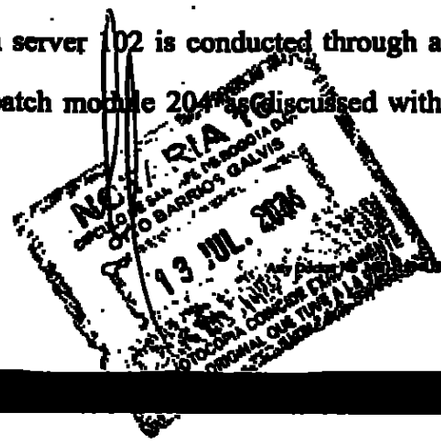
Scan-patch module 204 thus searches binary files 208 on client computer 108 to determine if a binary signature identifying a security vulnerability is present in any binary information located on client computer 108. If the bit pattern of the binary signature is found in a binary file 208, scan-patch module 204 operates to fix the security vulnerability in the binary file 208 by installing a corresponding

security patch on client computer 108. Installation of a security patch on client computer 108 overwrites or otherwise eliminates the binary file or a portion of the binary file containing the security vulnerability.

Fig. 3 illustrates another exemplary embodiment of a distribution server 102, a scan-patch server 104 and a client computer 108 suitable for implementing patching of security vulnerabilities in binary files. In general, in the Fig. 3 embodiment, binary signatures and security patches are pushed down, or redistributed, from the server 104 to the client computer 108, and the scanning for security vulnerable files and the patching of security vulnerable files is performed by the client computer 108 instead of the scan patch server 104.

In the Fig. 3 embodiment, distribution server 102 is configured in the same manner as discussed above with respect to the embodiment of Fig. 2. Thus, database 202 can be updated to include newly discovered binary signatures that identify security vulnerabilities in binary files. Database 202 can also be updated with corresponding security patches that have been developed to fix such security vulnerabilities.

The scan-patch server 102 of Fig. 3 is configured in somewhat the same manner as that discussed above with respect to Fig. 2. Thus, scan-patch server 102 of Fig. 3 includes a database 206 for receiving and holding binary signatures and security patches. Database 206 is typically updated automatically with new binary signatures and security patches through communications between the scan-patch server 104 and the distribution server 102. However, the communication between the scan-patch server 104 and the distribution server 102 is conducted through a redistribution module 300 instead of a scan-patch module 204 as discussed with respect to the Fig. 2 embodiment.



The redistribution module 300, in addition to updating database 206 with binary signatures and security patches, is configured to communicate with scan-patch module 302 on client computer 108 and transfer a binary signature to the client computer 108. Scan-patch module 302 is configured to receive the binary signature and to scan binary files 208 to determine if the binary signature is present in any binary information located on client computer 108. Thus, the scan-patch module 302 of Fig. 3 functions in a manner similar to the scan-patch module 204 discussed above with reference to Fig. 2.

If the bit pattern of the binary signature is found in a binary file 208 on client computer 108, scan-patch module 302 sends a request to the redistribution module 300 on server 102. The request is to have the redistribution module 300 send the security patch corresponding with the binary signature down to the client computer 108. The redistribution module 300 responds to the request by sending the appropriate security patch to client computer 108. The scan-patch module 302 receives the security patch and operates to fix the security vulnerability in the binary file 208 by installing the security patch on client computer 108. As in the Fig. 2 embodiment, installation of a security patch on client computer 108 overwrites or otherwise eliminates the binary file or a portion of the binary file containing the discovered security vulnerability.

Exemplary Methods

Example methods for implementing automatic detection and patching of security vulnerabilities in binary files will now be described with primary reference to the flow diagrams of Figs. 4 - 6. The methods apply generally to the exemplary embodiments discussed above with respect to Figs. 1 - 3. The elements

of the described methods may be performed by any appropriate means including, for example, by hardware logic blocks on an ASIC or by the execution of processor-readable instructions defined on a processor-readable medium.

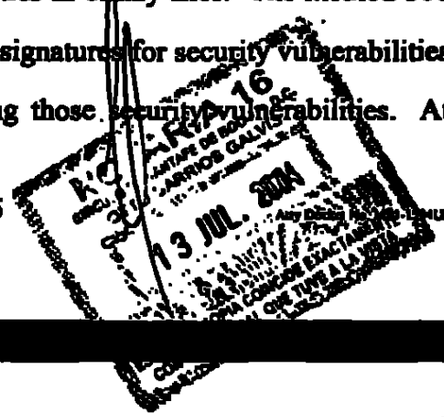
A "processor-readable medium," as used herein, can be any means that can contain, store, communicate, propagate, or transport instructions for use by or execution by a processor. A processor-readable medium can be, without limitation, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, device, or propagation medium. More specific examples of a processor-readable medium include, among others, an electrical connection (electronic) having one or more wires, a portable computer diskette (magnetic), a random access memory (RAM) (magnetic), a read-only memory (ROM) (magnetic), an erasable programmable-read-only memory (EPROM or Flash memory), an optical fiber (optical), a rewritable compact disc (CD-RW) (optical), and a portable compact disc read-only memory (CDROM) (optical).

Fig. 4 shows an exemplary method 400 for implementing automatic detection and patching of security vulnerabilities in binary files. The binary files are typically located or stored on a client computer being served by a server computer, but they may also be located on the server computer itself, or any other computing device accessible by the server computer. At block 402 of method 400, a binary signature is received. The binary signature is a bit pattern that has been associated with a security vulnerability in a particular binary file, such as an executable application program or operating system running on a client computer. The binary signature is received from a central distribution server 102 by a subordinate server 104.

At block 404, a security patch is received. The security patch is typically compiled executable code that has been developed as a fix to the security vulnerability of the particular binary file. The security patch is also received from the central distribution server 102 by the subordinate server 104. At block 406, a vulnerable binary file is identified based on the binary signature. The identification of the vulnerable binary file is typically achieved by scanning binary information stored on various media of a computer, such as client computer 108, and then comparing the pattern(s) in the binary signature with the binary information found on the media. The identification can happen in various ways including, for example, by the server 104 scanning and comparing all the binary information present on the client computer. The identification of a vulnerable binary file can also be achieved by having the server 104 push the binary signature down to the client computer so that the client computer can perform the scan and comparison.

At block 408 of method 400, the security patch is used to update the vulnerable binary file. The update can be achieved in various ways including, for example, by the server 104 installing the security patch on the client computer 108. If the client computer 108 has performed the scan and identified the vulnerable binary file, the client computer 108 may request that the server 104 send the security patch to the computer 108, in which case the computer 108 can install the security patch to fix the vulnerable binary file.

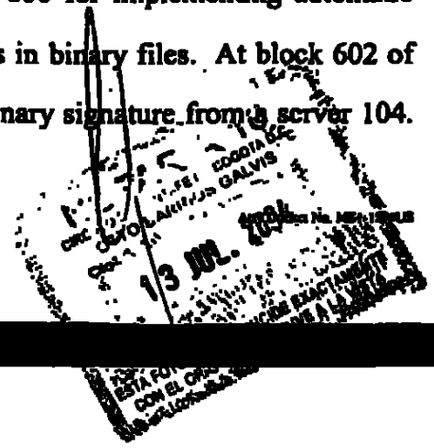
Fig. 5 shows another exemplary method 500 for implementing automatic detection and patching of security vulnerabilities in binary files. The method 500 generally illustrates the distribution of binary signatures for security vulnerabilities and the security patches developed for fixing those security vulnerabilities. At



block 502 of method 500, a binary signature is received that identifies a security vulnerability of a binary file. The binary signature is typically uploaded to a distribution server 102 as a newly discovered bit pattern that identifies a vulnerability in a binary file of a software product that may be widely distributed across many computers on a network such as the Internet. The upload is typically achieved from a computer coupled to the distribution server 102 or from a portable storage medium inserted into the distribution server 102. At block 504, a security patch configured to fix the security vulnerability is received by the distribution server 102 in a similar manner as the binary signature.

At block 506, the binary signature and the security patch are distributed to a plurality of subordinate servers 104 from distribution server 102. This distribution occurs automatically and can be achieved in various ways. For example, upon receiving an uploaded binary signature and security patch, the distribution server 102 can automatically send the binary signature and security patch out over the network to all subordinate servers 104 configured to receive updated binary signatures and security patches. The distribution server 102 might also send a notice to servers 104 indicating that a security vulnerability has been discovered and that a security patch is available to fix the vulnerability. Subordinate servers 104 can then request that the distribution server 102 send the binary signature that identifies the security vulnerability and the security patch. Upon receiving a request, the distribution server 102 can forward the binary signature and the security patch to requesting servers 102.

Fig. 6 shows another exemplary method 600 for implementing automatic detection and patching of security vulnerabilities in binary files. At block 602 of method 600, a client computer 108 receives a binary signature from a server 104.



The binary signature is associated with a security vulnerability in a binary file that may be present on the client computer 108. At block 604, the client computer 108 scans all the binary information presently available to it and compares the pattern(s) in the binary signature with the binary information. The binary information scanned by the client computer 108 is typically in the form of computer/processor-readable and/or executable instructions, data structures, program modules, and other data useful for client computer 108, and can reside on both volatile and non-volatile storage media of various types.

At block 606, if the client computer 108 finds a binary file that contains the binary signature, it sends a request to the server 104 to have the security patch transferred. At block 608, the client computer 108 receives the security patch, and at block 610, the client computer 108 installs the security patch in order to fix the security vulnerability in the binary file containing binary information matching the pattern(s) in the binary signature.

While one or more methods have been disclosed by means of flow diagrams and text associated with the blocks of the flow diagrams, it is to be understood that the blocks do not necessarily have to be performed in the order in which they were presented, and that an alternative order(s) may result in similar advantages. Furthermore, the methods are not exclusive and can be performed alone or in combination with one another.

Exemplary Computer

Fig. 7 illustrates an exemplary computing environment suitable for implementing a distribution server 102, a scan-patch server 104, and a client computer 108, as discussed above with reference to Figs. 1-6. Although one

specific configuration is shown in Fig. 7, distribution server 102, scan-patch server 104, and client computer 108 may be implemented in other computing configurations.

The computing environment 700 includes a general-purpose computing system in the form of a computer 702. The components of computer 702 can include, but are not limited to, one or more processors or processing units 704, a system memory 706, and a system bus 708 that couples various system components including the processor 704 to the system memory 706.

The system bus 708 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. An example of a system bus 708 would be a Peripheral Component Interconnects (PCI) bus, also known as a Mezzanine bus.

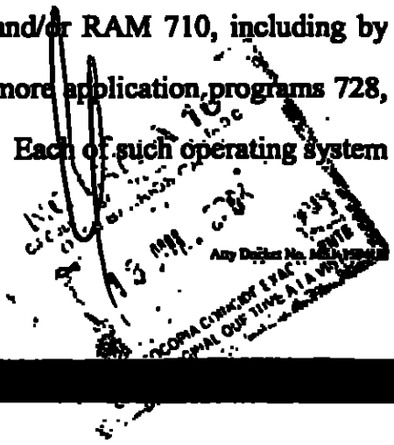
Computer 702 typically includes a variety of computer-readable media. Such media can be any available media that is accessible by computer 702 and includes both volatile and non-volatile media, removable and non-removable media. The system memory 706 includes computer readable media in the form of volatile memory, such as random access memory (RAM) 710, and/or non-volatile memory, such as read only memory (ROM) 712. A basic input/output system (BIOS) 714, containing the basic routines that help to transfer information between elements within computer 702, such as during start-up, is stored in ROM 712. RAM 710 typically contains data and/or program modules that are immediately accessible to and/or presently operated on by the processing unit 704.

Computer 702 can also include other removable/non-removable, volatile/non-volatile computer storage media. By way of example, Fig. 7

illustrates a hard disk drive 716 for reading from and writing to a non-removable, non-volatile magnetic media (not shown), a magnetic disk drive 718 for reading from and writing to a removable, non-volatile magnetic disk 720 (e.g., a "floppy disk"), and an optical disk drive 722 for reading from and/or writing to a removable, non-volatile optical disk 724 such as a CD-ROM, DVD-ROM, or other optical media. The hard disk drive 716, magnetic disk drive 718, and optical disk drive 722 are each connected to the system bus 708 by one or more data media interfaces 726. Alternatively, the hard disk drive 716, magnetic disk drive 718, and optical disk drive 722 can be connected to the system bus 708 by a SCSI interface (not shown).

The disk drives and their associated computer-readable media provide non-volatile storage of computer readable instructions, data structures, program modules, and other data for computer 702. Although the example illustrates a hard disk 716, a removable magnetic disk 720, and a removable optical disk 724, it is to be appreciated that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes or other magnetic storage devices, flash memory cards, CD-ROM, digital versatile disks (DVD) or other optical storage, random access memories (RAM), read only memories (ROM), electrically erasable programmable read-only memory (EEPROM), and the like, can also be utilized to implement the exemplary computing system and environment.

Any number of program modules can be stored on the hard disk 716, magnetic disk 720, optical disk 724, ROM 712, and/or RAM 710, including by way of example, an operating system 726, one or more application programs 728, other program modules 730, and program data 732. Each of such operating system



726, one or more application programs 728, other program modules 730, and program data 732 (or some combination thereof) may include an embodiment of a caching scheme for user network access information.

Computer 702 can include a variety of computer/processor readable media identified as communication media. Communication media typically embodies computer readable instructions, data structures, program modules, or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared, and other wireless media. Combinations of any of the above are also included within the scope of computer readable media.

A user can enter commands and information into computer system 702 via input devices such as a keyboard 734 and a pointing device 736 (e.g., a "mouse"). Other input devices 738 (not shown specifically) may include a microphone, joystick, game pad, satellite dish, serial port, scanner, and/or the like. These and other input devices are connected to the processing unit 704 via input/output interfaces 740 that are coupled to the system bus 708, but may be connected by other interface and bus structures, such as a parallel port, game port, or a universal serial bus (USB).

A monitor 742 or other type of display device can also be connected to the system bus 708 via an interface, such as a video adapter 744. In addition to the monitor 742, other output peripheral devices can include components such as

speakers (not shown) and a printer 746 which can be connected to computer 702 via the input/output interfaces 740.

Computer 702 can operate in a networked environment using logical connections to one or more remote computers, such as a remote computing device 748. By way of example, the remote computing device 748 can be a personal computer, portable computer, a server, a router, a network computer, a peer device or other common network node, and the like. The remote computing device 748 is illustrated as a portable computer that can include many or all of the elements and features described herein relative to computer system 702.

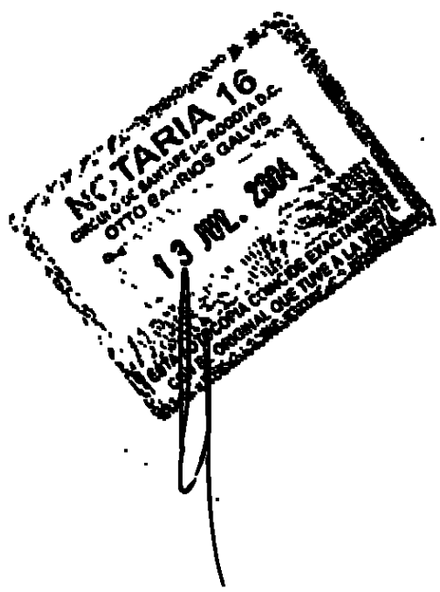
Logical connections between computer 702 and the remote computer 748 are depicted as a local area network (LAN) 750 and a general wide area network (WAN) 752. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet. When implemented in a LAN networking environment, the computer 702 is connected to a local network 750 via a network interface or adapter 754. When implemented in a WAN networking environment, the computer 702 typically includes a modem 756 or other means for establishing communications over the wide network 752. The modem 756, which can be internal or external to computer 702, can be connected to the system bus 708 via the input/output interfaces 740 or other appropriate mechanisms. It is to be appreciated that the illustrated network connections are exemplary and that other means of establishing communication link(s) between the computers 702 and 748 can be employed.

In a networked environment, such as that illustrated with computing environment 700, program modules depicted relative to the computer 702, or portions thereof, may be stored in a remote memory storage device. By way of

example, remote application programs 758 reside on a memory device of remote computer 748. For purposes of illustration, application programs and other executable program components, such as the operating system, are illustrated herein as discrete blocks, although it is recognized that such programs and components reside at various times in different storage components of the computer system 702, and are executed by the data processor(s) of the computer.

Conclusion

Although the invention has been described in language specific to structural features and/or methodological acts, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or acts described. Rather, the specific features and acts are disclosed as exemplary forms of implementing the claimed invention.



CLAIMS

1. A processor-readable medium comprising processor-executable instructions configured for:

receiving a binary signature;

receiving a security patch;

identifying a vulnerable binary file on a computer based on the binary signature; and

updating the vulnerable binary file on the computer with the security patch.

2. A processor-readable medium as recited in claim 1, wherein the identifying a vulnerable binary file on a computer includes comparing a bit pattern of the binary signature against binary files located on the computer, the bit pattern associated with a security vulnerability.

3. A processor-readable medium as recited in claim 1, wherein the updating the vulnerable binary file on the computer includes installing the security patch on the computer.

4. A processor-readable medium as recited in claim 1, wherein the identifying a vulnerable binary file on a computer includes sending the binary signature to the computer.

5. A processor-readable medium as recited in claim 4, wherein the updating the vulnerable binary file on the computer includes:

receiving a request from the computer to send the security patch; and

sending the security patch to the computer.

6. A processor-readable medium as recited in claim 1, wherein the computer is a client computer and the receiving includes receiving the binary signature and the security patch from a distribution server configured to distribute to the client computer, binary signatures that identify vulnerable files and security patches configured to fix the vulnerable files.

7. A server comprising the processor-readable medium as recited in claim 1.

8. A processor-readable medium comprising processor-executable instructions configured for:

receiving a binary signature that identifies a security vulnerability in a binary file;

receiving a security patch configured to fix the security vulnerability in the binary file; and

distributing the binary signature and the security patch to a plurality of servers.

9. A processor-readable medium as recited in claim 8, wherein the distributing includes:

sending a notice to each of the plurality of servers regarding the security vulnerability and the available patch;

receiving a request to send the binary signature and the security patch; and

sending the binary signature and the security patch in response to the request.

10. A distribution server comprising the processor-readable medium as recited in claim 8.

11. A processor-readable medium comprising processor-executable instructions configured for:

receiving a binary signature from a server;

searching for the binary signature in binary files;

sending a request to the server for a security patch if a binary file is found that includes the binary signature;

receiving the security patch from the server; and

updating the binary file with the security patch.

12. A client computer comprising the processor-readable medium as recited in claim 11.

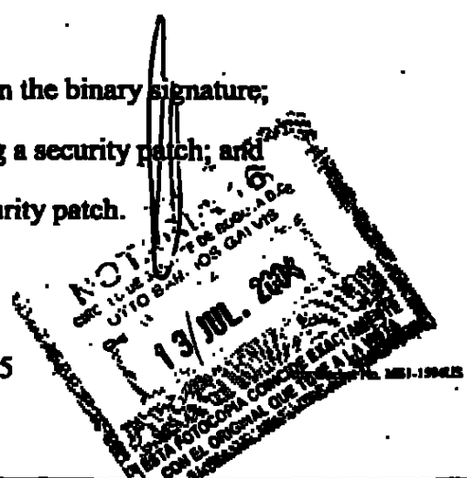
13. A method comprising:

receiving a binary signature;

searching for a vulnerable file based on the binary signature;

if a vulnerable file is found, requesting a security patch; and

fixing the vulnerable file with the security patch.



14. A method as recited in claim 13, wherein the requesting includes sending a request to a server for the security patch, the method further comprising receiving the security patch from the server in response to the request.

15. A method as recited in claim 14, wherein the receiving includes receiving the binary signature from the server.

16. A method as recited in claim 13, wherein the fixing includes installing the security patch on a computer.

17. A method as recited in claim 13, wherein the searching includes comparing the binary signature to binary information on a storage medium of a computer.

18. A method as recited in claim 17, wherein the binary information is selected from the group comprising:

an operating system;

an application program file; and

a data file.

19. A method as recited in claim 17, wherein the storage medium is selected from the group comprising:

a hard disk;

a magnetic floppy disk;

an optical disk;

a flash memory card;

an electrically erasable programmable read-only memory; and

network-attached storage.

20. A method comprising:

receiving a binary signature and a security patch from a distribution server;

searching on a client computer for a vulnerable file associated with the binary signature; and

if a vulnerable file is found, fixing the vulnerable file with the security patch.

21. A method as recited in claim 20, wherein the searching includes transferring the binary signature to the client computer, the client computer configured to search for a vulnerable file associated with the binary signature.

22. A method as recited in claim 21, wherein the fixing includes:

receiving a request from the client computer to transfer the security patch, the client computer having located a vulnerable file; and

transferring the security patch to the client computer in response to the request.

23. A computer comprising:

means for receiving a binary signature;

means for searching for a vulnerable file based on the binary signature;

means for requesting a security patch if a vulnerable file is found; and

means for fixing the vulnerable file with the security patch.

24. A server comprising:

means for receiving a binary signature and a security patch from a distribution server;

means for scanning a client computer for a vulnerable file associated with the binary signature; and

means for fixing the vulnerable file with the security patch if a vulnerable file is found.

25. A computer comprising:

binary information;

a scan module configured to receive a binary signature and scan the binary information for the binary signature; and

a patch module configured to request a security patch and install the security patch if the binary signature is found in the binary information.

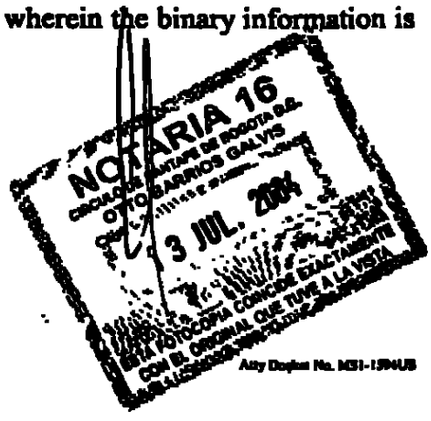
26. A computer as recited in claim 25, further comprising a storage medium configured to retain the binary information.

27. A computer as recited in claim 25, wherein the binary information is selected from the group comprising:

an operating system;

an application program file; and

a data file.



28. A computer comprising:

binary files;

a binary signature; and

a security patch module configured to receive the binary signature from a server and to scan the binary files in search of the binary signature.

29. A computer as recited in claim 28, further comprising:

a binary file that includes the binary signature; and

a security patch;

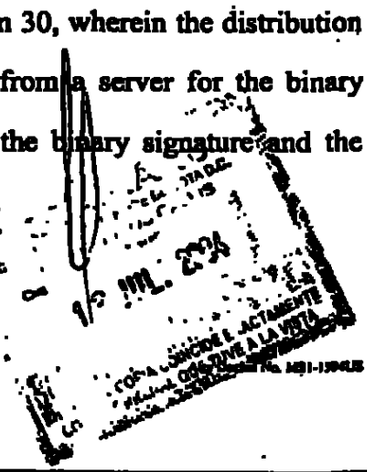
wherein the security patch module is further configured to request the security patch from the server upon locating the binary signature within the binary file, and to apply the security patch to the binary file.

30. A distribution server comprising:

a database; and

a distribution module configured to receive a binary signature and a security patch, store the binary signature and the security patch in the database, and distribute the binary signature and the security patch to a plurality of servers.

31. A distribution server as recited in claim 30, wherein the distribution module is further configured to receive a request from a server for the binary signature and the security patch and to distribute the binary signature and the security patch to the server in response to the request.



32. A server comprising:

a binary signature associated with a security vulnerability in a binary file;

a security patch configured to fix the security vulnerability in the binary file; and

a scan module configured to scan binary files on a client computer for the binary signature and to update the binary file with the security patch if the binary signature is found.

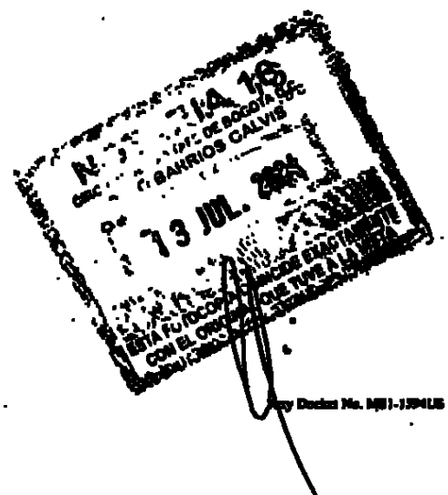
33. A server as recited in claim 32, further comprising:

a database;

the scan module further configured to receive the binary signature and the security patch from a distribution server and to store the binary signature and the security patch in the database.

ABSTRACT

Systems and methods are described that enable patching of security vulnerabilities in binary files. The detection and patching of vulnerable binary files is automatic, reliable, regression free, and comprehensive across networks on an unlimited scale. These advantages can be realized in various ways including, for example, by leveraging current anti-virus infrastructure that is widely deployed across the Internet. Reliable discovery of vulnerable binary files (e.g., in operating systems, application programs, etc.) is achieved through the use of binary signatures that have been associated with discovered security vulnerabilities. A divergence of security patches away from conventional service packs provides for the possibility of production of regression-free fixes for security vulnerabilities in binary files.



98

JUL 16 2003 12:50 FR MICROSOFT DEPT 4

EV251222109

PATENT APPLICATION

DECLARATION AND POWER OF ATTORNEY

ATTORNEY DOCKET NO. MS1-1594US MS DOCKET NO. 3045921

As a below named inventor, I hereby declare that:

My residence/post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (i plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: Automatic Detection And Patching Of Vulnerable Files

the specification of which is filed herewith unless the following box is checked:

( ) was filed on as US Application Serial No. or PCT International Application Number and was amended on _____ (if applicable).

I hereby state that I have reviewed and understood the contents of the above-identified specification, including the claims, as amended by any amendment(s) referred to above. I acknowledge the duty to disclose all information which is material to patentability as defined in 37 CFR 1.56.

Foreign Application(s) and/or Claim of Foreign Priority

I hereby claim foreign priority benefits under Title 35, United States Code Section 119 of any foreign application(s) for patent or inventor(s) certificate listed below and have also identified below any foreign application for patent or inventor(s) certificate having a filing date before that of the application on which priority is claimed:

COUNTRY	APPLICATION NUMBER	DATE FILED	PRIORITY CLAIMED UNDER 35 U.S.C. 119
			YES: NO:
			YES: NO:

Provisional Application

I hereby claim the benefit under Title 35, United States Code Section 119(e) of any United States provisional application(s) listed below:

APPLICATION SERIAL NUMBER	FILING DATE

U.S. Priority Claim

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matte of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

APPLICATION SERIAL NUMBER	FILING DATE	STATUS(patented/pending/abandoned)

POWER OF ATTORNEY:

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) listed below and those associated wi

Customer No. 22801

22801

PATENT TRADEMARK OFFICE

to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

Daniel D. Crouse, Reg. No. 32022	David Bartley Eppenauer, Reg. No. 35499	Martin L. Shively, Reg. No. 33553	Ronald O. Zink, Reg. No. 35744
Patricia E. Bomes, Reg. No. 37038	Michael W. Bodanowski, Reg. No. 28692	Danielle J. Holmes, Reg. No. 39720	

Send correspondence to:
Nathaniel Gleth
Lee & Hayes PLLC
421 W. Riverside Ave., Suite 500
Spokane, WA 99201

22801
PATENT TRADEMARK OFFICE

Direct Telephone Call To:
Nathaniel Gleth
Phone Number: (509) 324-9256

13 JUL. 2004

[redacted]

DECLARATION AND POWER OF ATTORNEY

ATTORNEY DOCKET NO. MS1-1594US **MS DOCKET NO. 304592.1**

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Inventor: Oleg Ivanov **Citizenship: Ukraine/USA**

Residence: Issaquah, WA

Post Office Address: 25923 SE 37th Way Issaquah, WA 98029

Oleg Ivanov
Inventor's Signature

7/16/2003
Date

Full Name of Inventor: Sergei Ivanov **Citizenship: Ukraine/USA**

Residence: Issaquah, WA

Post Office Address: 25923 SE 37th Way Issaquah, WA 98029

Sergei Ivanov
Inventor's Signature

7/16/2003
Date

NOTARIA 16
CIRCULO DE SANTAFE DE BOGOTA D.C.
OTTO BARRIOS GALVIS
13 JUL 2003

100

Docket No.: MS1-1594US
Inventors: Oleg Ivanov, Sergei Ivanov
Title: Automatic Detection And Patching
Of Vulnerable Files

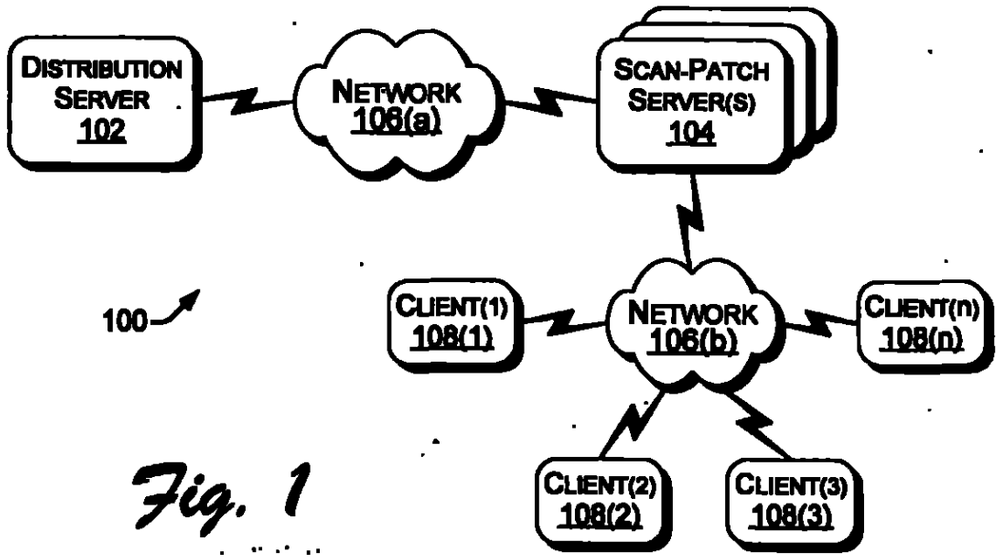
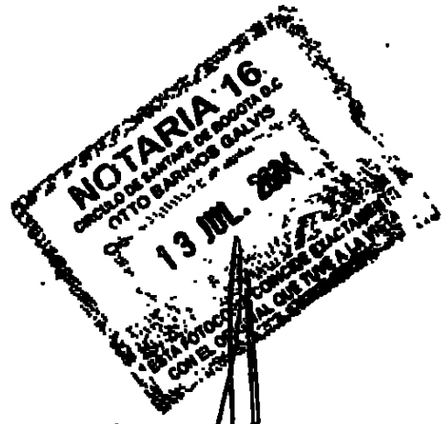


Fig. 1



736

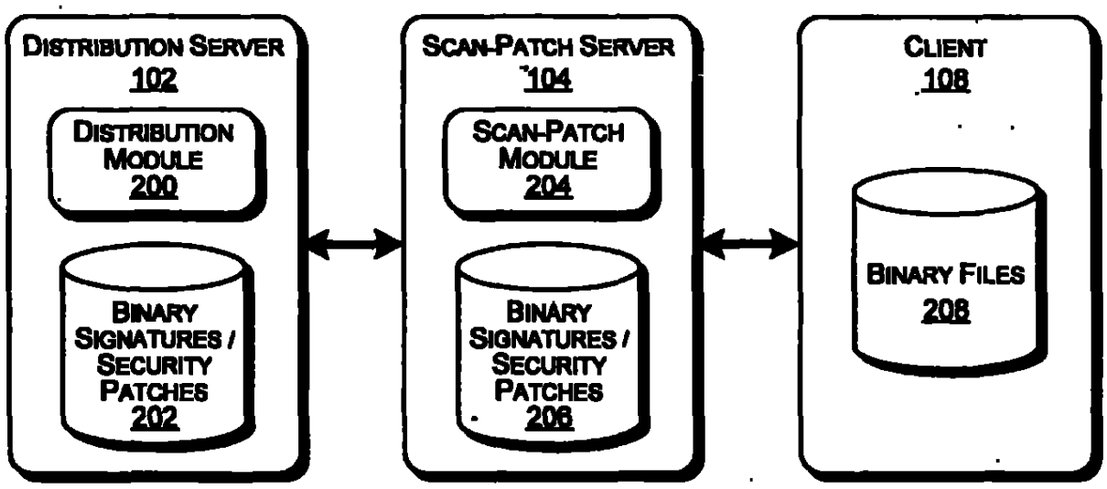


Fig. 2

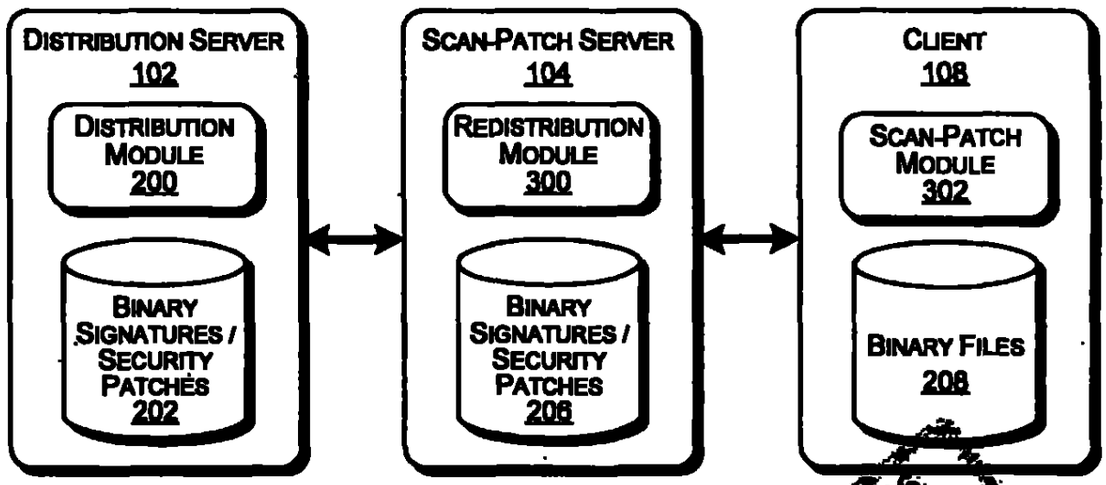


Fig. 3

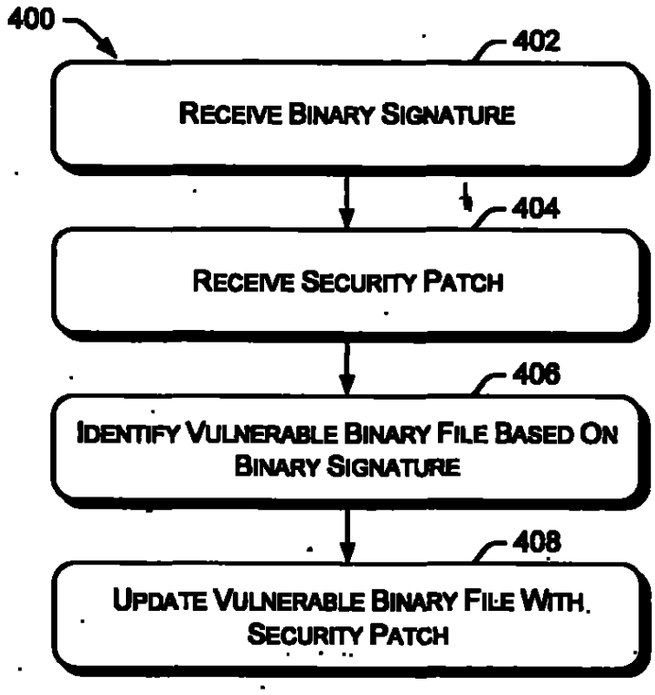


Fig. 4

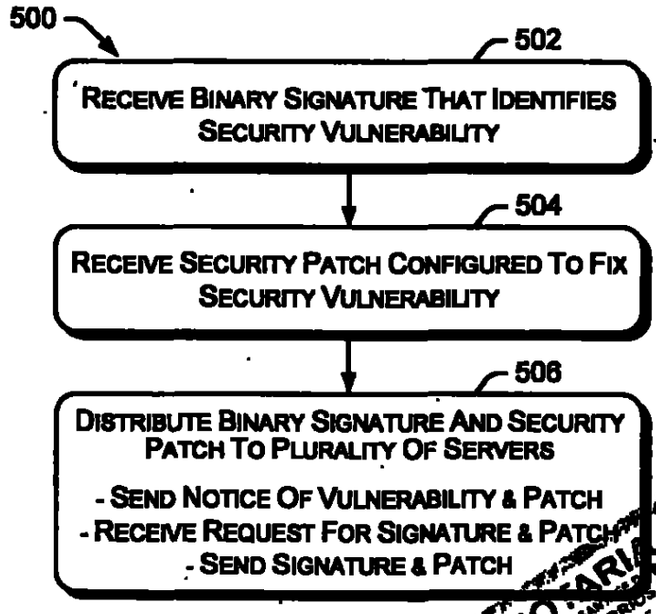


Fig. 5

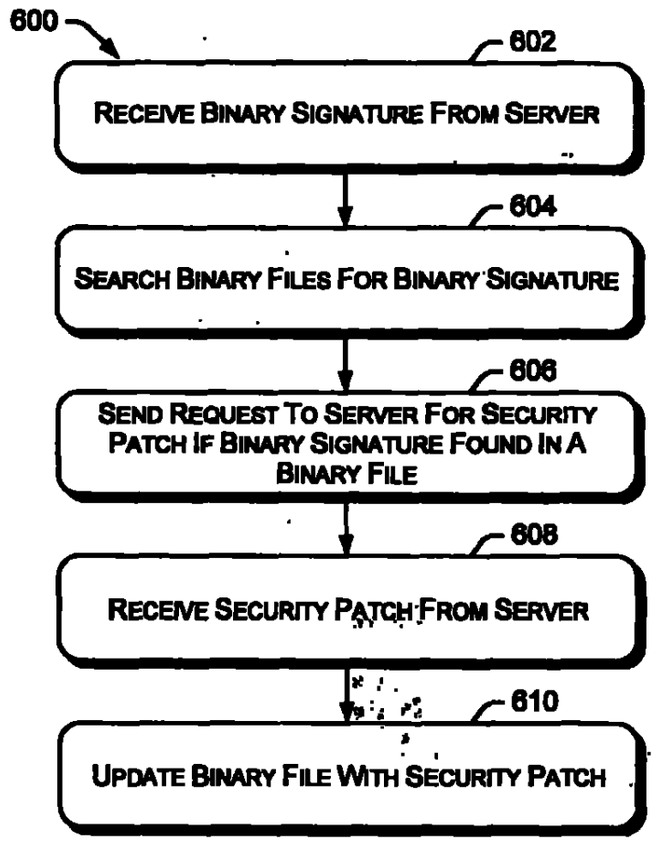
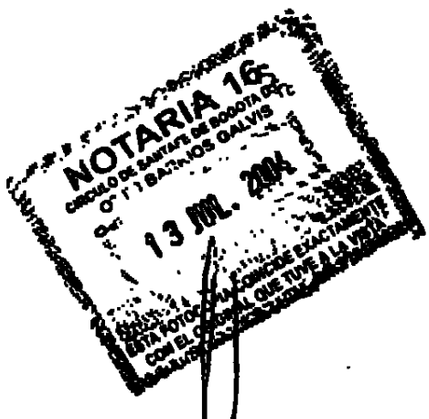


Fig. 6



Docket No.: MS1-1594US
Inventors: Oleg Ivanov Sergei Ivanov
Title: Automatic Detection And Patching Of Vulnerable Files

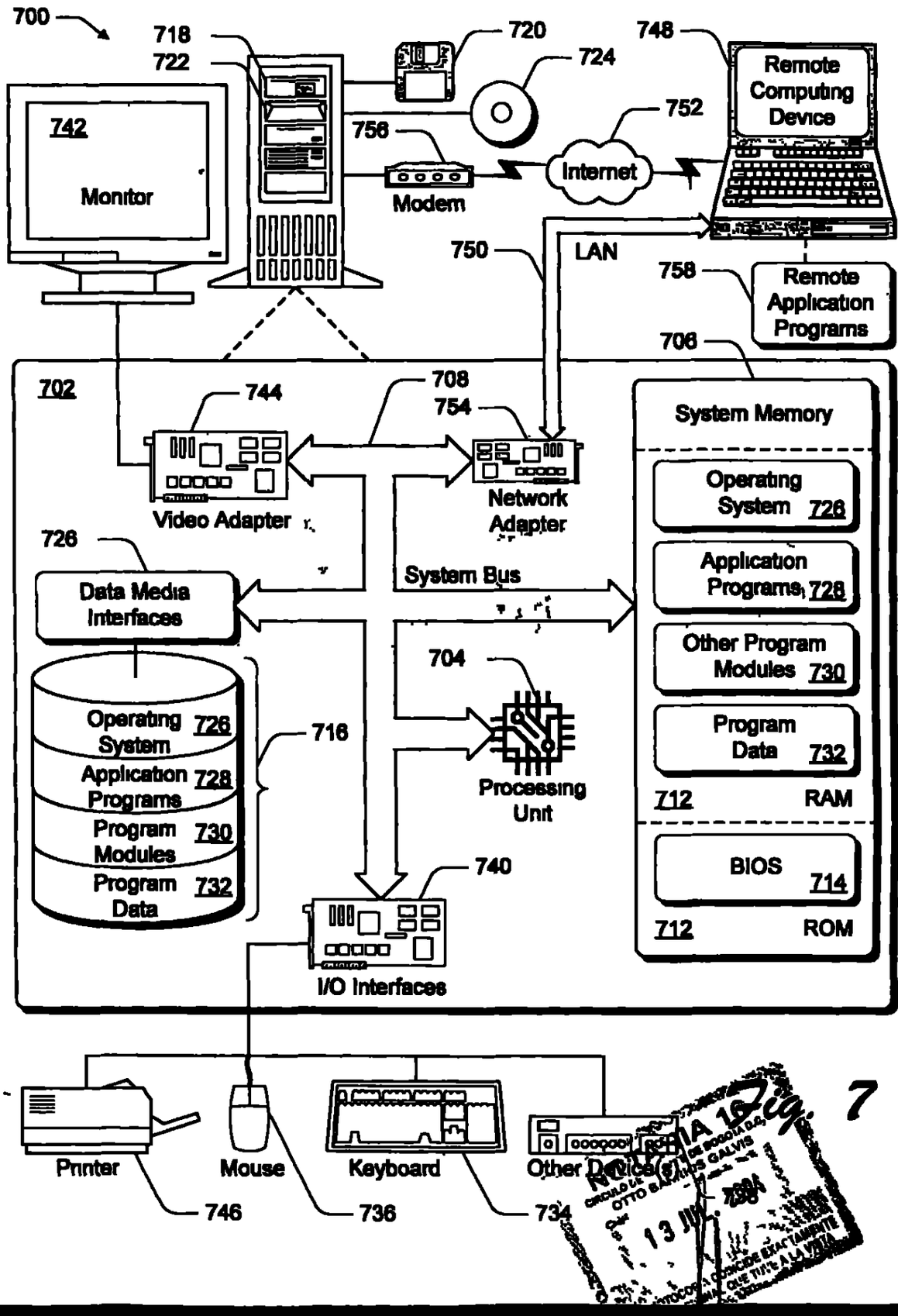


Fig. 7

Traducción Oficial del Inglés al Español de una solicitud de patente identificada en lo que sigue y realizada el 12de Julio de 2004

105

LOS ESTADOS UNIDOS DE AMERICA

A TODO AQUEL A QUIEN CONCIERNA LO PRESENTE

DEPARTAMENTO DE COMERCIO DE LOS ESTADOS UNIDOS

Oficina de Patentes y Marcas de los Estados Unidos

Junio 07, 2004

LA PRESENTE ES PARA CERTIFICAR QUE ADJUNTA A LA MISMA HAY UNA COPIA FIEL DE LOS REGISTROS DE LA OFICINA DE PATENTES Y MARCAS DE LOS ESTADOS UNIDOS DE LOS DOCUMENTOS DE LA SOLICITUD DE PATENTE IDENTIFICADA EN LO QUE SIGUE, QUE CUMPLEN LOS REQUISITOS PARA OTORGARLE UNA FECHA DE PRESENTACIÓN BAJO 35 USC 111.

NÚMERO DE SOLICITUD: *10/621,148*
FECHA DE PRESENTACIÓN: *Julio 16, 2003*

Por la Autoridad del
COMISARIO DE PATENTES Y MARCAS

(Aparece Firma:
L. Edelen
Oficial Certificador

BEATRIZ EUGE... ALARCON GUTIERREZ
TRADUCTORA E INTERPRETE PUBLICO
RESOLUCION 2513 DE SEPT. 15/93 MINJUSTICIA

4

106

SOLICITUD DE PATENTE NO. DE SERIE _____

DEPARTAMENTO DE COMERCIO

OFICINA DE MARCAS Y PATENTES

HOJA DE REGISTRO DE TASAS

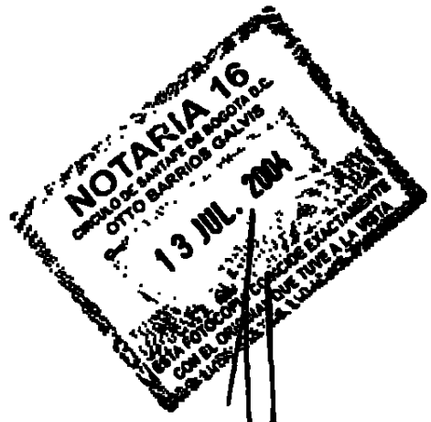
07/21/2003 SDENBOBI 00000021 120769 10621148

01 FC:1001	750.00 DA
02 FC: 1202	234.00 DA
03 FCX:1201	672.00 DA

PTO-1556

(5/87)

*U.S. Government Printing Office 2002 - 489-267/69033



142

107

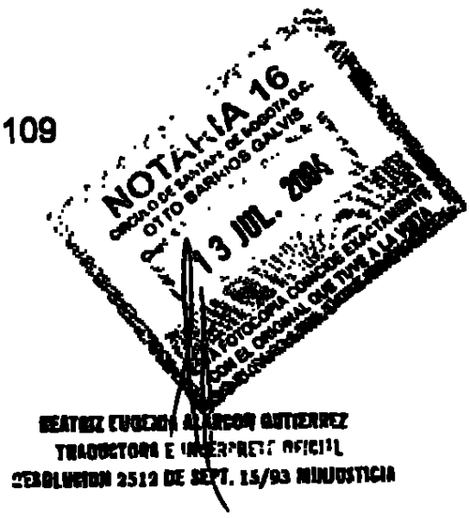
Por favor escriba a máquina un signo más (+) en esta casilla → ⊞

PTO/SB/05(03-01)
Aprobado para uso durante 10/31/2002. OMB 0651-0032
Oficina de Marcas y Patentes de los Estados Unidos;
DEPARTAMENTO DE COMERCIO DE LOS ESTADOS UNIDOS
De acuerdo con el acta de reducción de papel de 1995, no se requiere que ninguna persona responda para una recolección de información a menos que este muestre un número de control OMB válido.

TRASPASO DE SOLICITUD DE PATENTE DE UTILIDAD

(Solo para nuevas solicitudes no provisionales de acuerdo con el 37 C.F.R. 1,53(b))

Registro de abogado	No. MS1-1594US
Primer Inventor	Ivanov
Titulo	DETECCIÓN Y PARCHEO AUTOMÁTICO DE ARCHIVOS VULNERABLES
Etiqueta de correo expreso No.	EV251222109



143

108

ELEMENTOS DE LA SOLICITUD

Ver capítulo 600 MPEP que se relacionan con los contenidos de la solicitud de patente de utilidad.

DIRIGIDO A:

Solicitud de Patente Parada de Correo/P.O. Box 1450

Comisionado para Patentes/P.O. Box 1450

Alexandria, VA 22313 - 1450

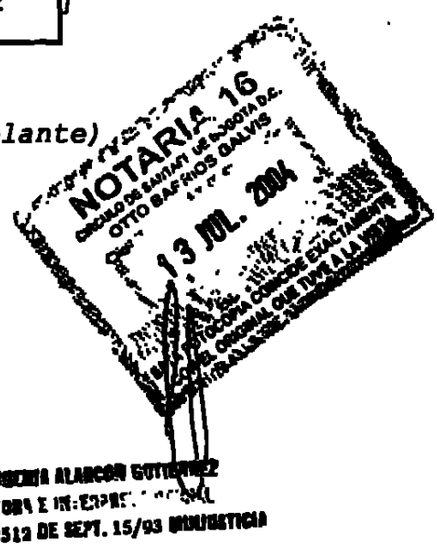
1. ☑ Tasa del Formulario de Traspaso (por ejemplo, PTO/SB/17)
   (Presentar un original y un duplicado para la tasa de procesamiento)

2. ☐ El solicitante reivindica el estado de pequeña entidad.
   Ver 37 CFR 1.27.

3. ☑ Descripción *[total de hojas* [32] *]*

(La disposición preferida se establece adelante)

- Título descriptivo de la invención



BEATRIZ EUGENIA ALARCON GUTIERREZ
TRADUCTORA E INTERPRETE OFICIAL
RESOLUCION 2512 DE SEPT. 15/93 MINJUSTICIA

144

109

- Referencia Cruzada a Solicitudes Relacionadas
- Declaración con respecto a patrocinio suministrado por R&D
- Referencia al listado de secuencias, una tabla, o un apéndice de lista de programa de computador
- Antecedentes de la Invención
- Breve Resumen de la Invención
- Breve Descripción de los Dibujos (si se presentan)
- Descripción Detallada
- Reivindicación(es)
- Resumen de la Descripción

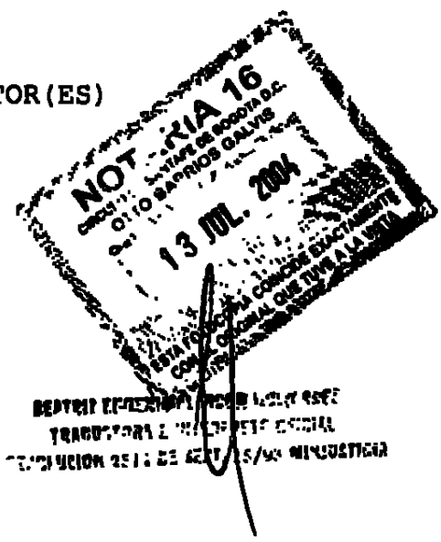
4. ☑ Dibujo (s) (35 U.S.C. 113) [Total Hojas [5]]

5. Juramento o Declaración [total de páginas [2]]

a. ☑ Formalizado nuevamente (original o copia)

b. ☐ copia de solicitud anterior (37 C.F.R. 1.63(d)) (para continuación/divisional con la casilla 18 Completada)

i. ☐ SUPRESIÓN DE INVENTOR(ES)



145

110

declaración firmada adjunta suprimiendo inventor(es) mencionados en la solicitud anterior, ver 37 C.F.R. 1.63(d)(2) y 1.33(b).

6. Hoja de Datos de la Solicitud. Ver 37 CFR 1.76

7. ☐ CD-ROM o CD-R en duplicado, tabla grande o programa de Computador (Apéndice)

8. ☐ Presentación de secuencias de ácido amino y/o nucleótido *(si aplica, todo lo necesario)*

   a. ☐ Copia legible en Computador (CRF)

   b. ☐ listado de la Especificación de Secuencia en:

      i. ☐ CD-ROM o CD-R (2 copias); o

      ii. ☐ papel

   c. ☐ Declaración que verifica la identidad de las anteriores copias

146

111

PARTES DE LA SOLICITUD QUE LAS ACOMPAÑAN

9. ☑ Documentos de Traspaso (portadas & documento(s))

10. ☐ 37 C.F.R. 3.73(b) Declaración (cuando exista cesionario) ☑ Poder de Abogado

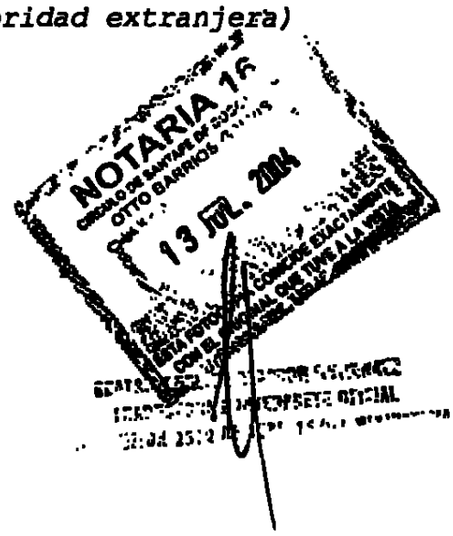
11. ☐ Documento de traducción al Inglés (si aplica)

12. ☑ Declaración de descripción de la información (IDS/PTO-1449) ☑ Copias de citaciones de IDS

13. ☐ Correcciones preliminares

14. ☑ Retorno postal de recibo (MPEP 503) *(debe estar itemizado específicamente)*

15. ☐ Copia Certificada de Documento(s) de Prioridad *(si se reivindica prioridad extranjera)*



147

112

16. ☐ Solicitud de no publicación de acuerdo con el U.S.C. 35 122 (b)(2)(B)(i). El Solicitante debe adjuntar formulario PTO/SB/35 o su equivalente.

17. ☐ Otro:.....

18. Si una SOLICITUD DE CONTINUACIÓN, marque la casilla apropiada, y suministre la información solicitada adelante y en una corrección preliminar, o en una Hoja de Datos de Solicitud de acuerdo con el 37 CFR 1.76:

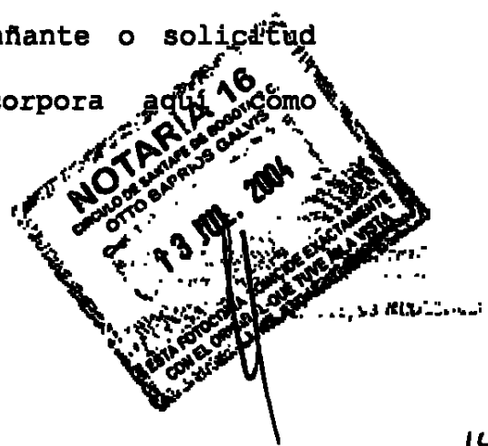
☐ Continuación. ☐ Divisional. ☐ Continuación en parte (CIP) de una solicitud anterior No. _____

Información de la solicitud anterior: _____

Examinador: _____

Grupo/Unidad Técnica _____

Solo para SOLICITUDES de CONTINUACIÓN o DIVISIONAL: La descripción completa de la solicitud anterior, a partir de la cual un juramento o declaración se suministra de acuerdo con la Casilla 5b, se considera parte de la descripción de la continuación acompañante o solicitud divisional y por lo tanto se incorpora aquí como



148

113

referencia. La incorporación solo puede ser confiada cuando una porción de las partes de la solicitud presentada ha sido omitida inadvertidamente.

19. DIRECCIÓN DE CORRESPONDENCIA

☑ Número de Cliente o Etiqueta de Código de Barras o

☐ Dirección de Correspondencia adelante

(Aparece Código de barras:

22801

OFICINA DE MARCAS Y PATENTES)

Nombre

Dirección

Ciudad

País

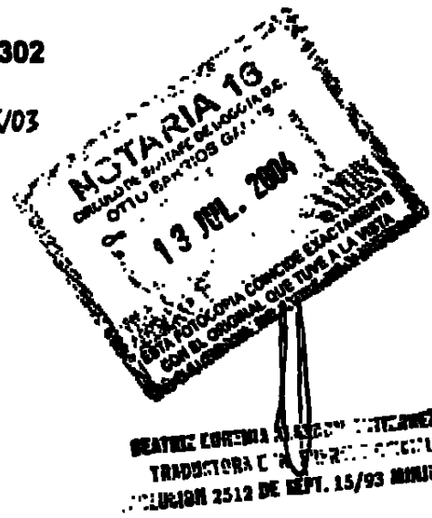
Teléfono Fax Código Zip

Nombre (Impreso/escrito a máquina) **Nathan R Rieth**

Firma (aparece firma ilegible)

Registro No. (Abogado/Agente) **44,302**

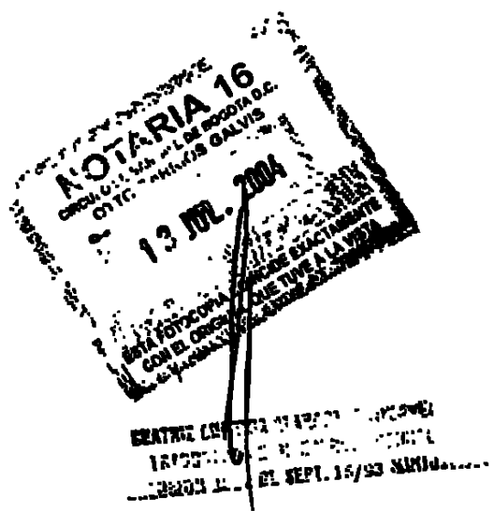
Fecha *1/16/03*



149

114

Declaración de Límite de Tiempo: se estima que este formulario toma 0,2 horas en completarse. El tiempo variará dependiendo de las necesidades en cada caso individual. Cualquier comentario sobre la cantidad de tiempo que usted necesita para completar este formulario se debe enviar al Oficial Jefe de Información, Oficina de Marcas y Patentes en los Estados Unidos, Washington, DC 20231. NO ENVIAR TASAS O FORMULARIO COMPLETOS A ESTA DIRECCIÓN. ENVIAR A: Comisionado Asistente para Patentes, Casilla Solicitud de Patente, Washington, DC 20231.



180

115

PTO/SB/17(01-03)
Aprobado para uso hasta 04/30/2003. OMB 0651-0032
Oficina de Marcas y Patentes de los Estados Unidos;
DEPARTAMENTO DE COMERCIO DE LOS ESTADOS UNIDOS
De acuerdo con el acta de reducción de papel de 1995, no se requiere que ninguna persona responda para una recolección de información a menos que este muestre un número de control OMB válido.

REMISIÓN DE TASAS para FY 2003

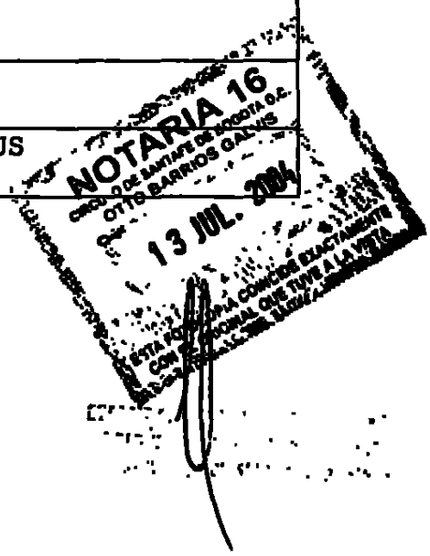
Vigente 01/01/2003. Las tasas de patentes están sujetas a revisión anual.

☐ El Solicitante reivindica el estado de pequeña entidad. Ver 37 CFR 1.27

CANTIDAD TOTAL DE PAGO ($)1.696,00

Complete si Conoce

Número de solicitud	
Fecha de Presentación	
Primer Inventor Mencionado	Ivanov
Nombre del Examinador	
Unidad Técnica	
Registro de Abogado No.	MS1-1594US



151

116

MÉTODO DE PAGO *(Marque todo lo que aplique)*

☐ Cheque ☐ Tarjeta de Crédito ☐ Giro Postal

☐ Otro ☐ Ninguno ☑ Cuenta de Depósito:

Número Cuenta de Depósito:	12-0769
Nombre Cuenta de Depósito:	Lee & Hayes, PLLC

Se autoriza al Comisionado a: *(Revisar toda la solicitud)*

☑ El cargo de tasa(s) se indica adelante

☑ Acreditar cualquier sobrepago

☑ Cargar cualquier tasa adicional durante la pendencia de esta solicitud

☐ El cargo de tasa(s) se indica adelante, **excepto para la tasa de presentación** de la cuenta de depósito identificada anteriormente.

NOTARIA 16
CIRCULO DE SANTAFE DE BOGOTA D.C.
OTTO BARRIOS GALVIS
13 JUL. 2004
ESTA FOTOCOPIA COINCIDE EXACTAMENTE CON EL ORIGINAL QUE TUVE A LA VISTA
BEATRIZ EUGENIA ALARCON RETTERIEZ
TRADUCTORA E INTERPRETE OFICIAL
RESOLUCION 2512 DE SEPT. 15/93 MINJUSTICIA

152

117

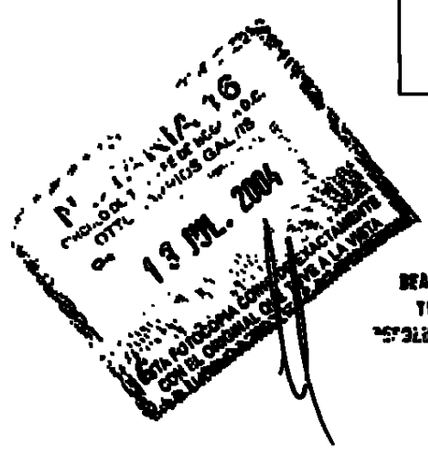
CÁLCULO DE TASAS

1. TASA DE PRESENTACIÓN BÁSICA

Gran Entidad		Pequeña entidad			
Código tasa	**($) tasa**	**Código tasa**	**($) tasa**	**Descripción de la tasa**	**Tasa pagada**
1001	750	2001	375	Tasa de presentación Utilidad	750
1002	330	2002	165	Tasa de presentación Diseño	
1003	520	2003	260	Tasa de presentación Planta	
1004	750	2004	375	Tasa de presentación Republicación	
1005	160	2005	80	Tasa de presentación Provisional	
				SUBTOTAL (1) ($)	750,00

2. TASAS PARA REIVINDICACIONES EXTRA PARA UTILIDAD Y REPUBLICACIÓN

					Tasa de abajo		Tasa pagada
Total Reivindicaciones	33	-20**=	13	X	18	=	234
Reivindicaciones Independientes	11	-3**=	8	X	84	=	672
Dependiente múltiple						=	



153

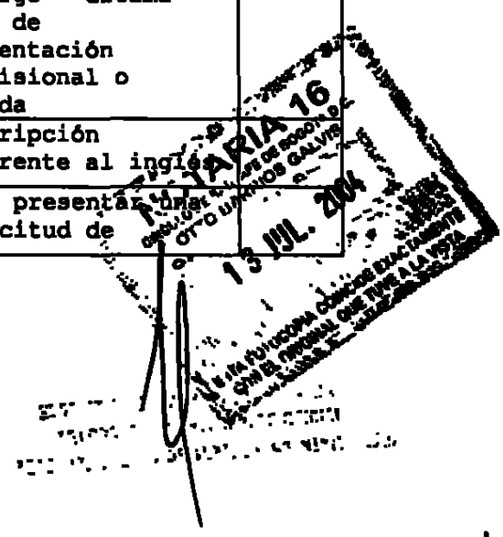
118

Gran Entidad		Pequeña Entidad			
Código tasa	**($) tasa**	**Código tasa**	**($) tasa**	**Descripción de la tasa**	
1202	18	2202	9	Reivindicaciones en exceso de 20	
1201	84	2201	42	Reivindicaciones en exceso de 3	
1203	280	2203	140	Múltiples reivindicaciones dependientes si no pagó	
1204	84	2204	42	** Reivindicaciones independientes republicadas sobre la patente original	
1205	18	2205	9	** Reivindicaciones en exceso de 20 republicadas sobre la patente original	
				SUBTOTAL (2) ($)	906,00
***o número pagado previamente, si es mayor; Para Republicaciones, ver arriba*					

CÁLCULO DE TASAS (CONTINUACIÓN)

3. TASAS ADICIONALES

Gran Entidad		Pequeña entidad			
Código tasa	**($) tasa**	**Código tasa**	**($) tasa**	**Descripción de la tasa**	**Tasa pagada**
1051	130	2051	65	Recargo - última tasa de presentación o Juramento	
1052	50	2052	25	Recargo - última tasa de presentación provisional o potada	
1053	130	1053	130	Descripción diferente al inglés	
1812	2.520	1812	2.520	Para presentar una solicitud de	



154

119

				reexamen ex parte	
1804	920*	1804	920*	Solicitar publicación de SIR antes que la acción del examinador	
1805	1840*	1805	1.840*	Solicitud de publicación de SIR después de la acción del examinador	
1251	110	2251	55	Extensión para respuesta dentro del primer mes	
1252	410	2252	205	Extensión para respuesta dentro del segundo mes	
1253	930	2253	465	Extensión para respuesta dentro del tercer mes	
1254	1.450	2254	725	Extensión para respuesta dentro del cuarto mes	
1255	1.970	2255	985	Extensión para respuesta dentro del quinto mes	
1401	320	2401	160	Nota de apelación	
1402	320	2402	160	Presentar resumen como apoyo a una apelación	
1403	280	2403	140	Solicitud de interrogatorio oral	
1451	1.510	1451	1.510	Petición al instituto de un procedimiento de uso público	
1452	110	2452	55	Petición para reactivar - ineludible	
1453	1.300	2453	650	Petición para reactivar - no intencional	
1501	1.300	2501	650	Tasa de publicación de utilidad (o republicación)	
1502	470	2502	235	Tasa de publicación de Diseño	
1503	630	2503	315	Tasa de publicación de planta	
1460	130	1460	130	Peticiones al Comisionado	

155

120

1807	50	1807	50	Tasa de procesamiento de acuerdo con el 37 CFR 1.17(q)	
1806	180	1806	180	Presentación de la declaración de Descripción de Información	
8021	40	8021	40	Registro de cada traspaso de patente por propiedad (número de ve3ces de propiedad)	40
1809	750	2809	375	Presentar una solicitud después de rechazo final (37 CFR 1.129(a))	
1810	750	2810	375	Para cada invención adicional a ser examinad (37 CFR 1.129(b))	
1801	750	2801	375	Solicitud de Examen Continuado (RCE)	
1802	900	1802	900	Solicitud para acelerar examen de una solicitud de diseño	
* Reducido por tasa de presentación básica pagada				SUBTOTAL (3) ($)	40,00

PRESENTADA POR

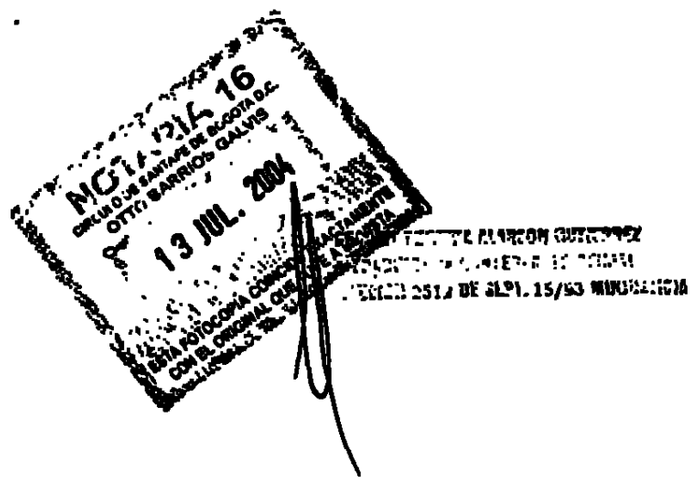
Nombre (Impreso/escrito a máquina) **Nathan R. Rieth**

Firma (aparece firma ilegible)

Registro No. (Abogado/Agente) **44,302**

Teléfono **(509) 324-9256**

Fecha *7/16/03*



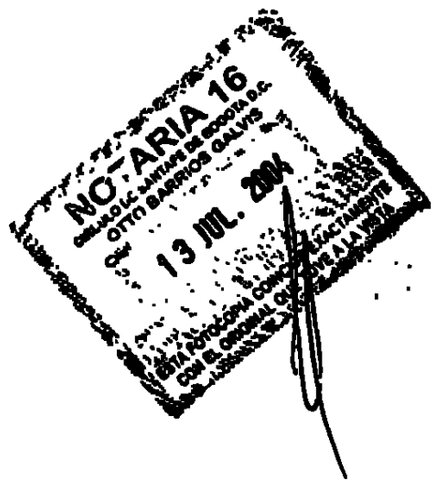
156

121

ADVERTENCIA: La información contenida en este formulario puede llegar a ser pública. La información de la Tarjeta de Crédito no se debe incluir en este formulario. Suministre la Información de Tarjeta de Crédito y autorización en el PTO-2038.

Esta recolección de información se requiere por el 37 CFR 1.17 y 1.27. Se requiere la información para obtener o retener un beneficio por el público que es registrar (y mediante el USITO para procesar) una solicitud. Regido confidencialmente por 35 U.S.C. 122 y 37 CFR 1.14. esta recolección se estima toma 12 minutos para completarse, e incluye reunir, preparar, y presentar el formulario de solicitud completo del USPTO. El tiempo variará del caso individual. Cualquier comentario sobre la cantidad de tiempo que usted necesita para completar este formulario y/o sugerencias para reducir este límite, se debe enviar al Oficial Jefe de Información, Oficina de Marcas y Patentes en los Estados Unidos, Departamento de Comercio de los Estados Unidos, Washington, DC 20231. NO ENVIAR TASAS O FORMULARIO COMPLETOS A ESTA DIRECCIÓN.

Si usted necesita ayuda en la terminación del formulario, calle al 1-800-PTO-9199 (1-800-786-9199) y seleccione la opción 2.



157

122

EV251222109

EN LA OFICINA DE PATENTES Y MARCAS DE LOS ESTADOS UNIDOS

SOLICITUD PARA TÍTULO DE PATENTE

Detección y Parcheo Automático de Archivos Vulnerables

Inventores:

Oleg Ivanov

Sergei Ivanov

REFERENCIA DE ABOGADO NO. MS1-1594US

158

123

CAMPO TÉCNICO

La presente descripción se relaciona de manera general con parcheo de archivos, y más particularmente, con una forma automática, comprensiva, confiable y libre de regresión de suministrar parches de seguridad para archivos de programas binarios vulnerables en ambientes de cómputo distribuidos, heterogéneos.

ANTECEDENTES

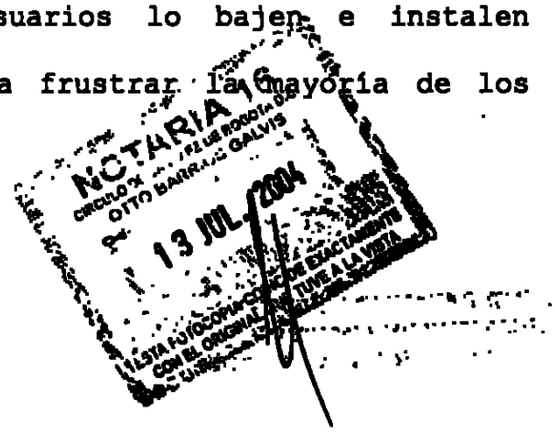
El desarrollo de software es un proceso que avanza por medio del cual el producto del software inicialmente liberado al público puede ser actualizado continuamente a través de revisiones del desarrollador/vendedor de software. Las revisiones de software son típicamente desembolsados de un vendedor de software en lo que son llamados "paquetes de servicios" que se pueden bajar u ordenar por un vendedor para instalación en un computador de usuario. Los paquetes de servicio contienen típicamente programas fijos (por ejemplo para un sistema operativo, un programa de aplicación, etc.) que repara problemas (es decir, "errores") descubiertos en el código de programa después de la liberación inicial del producto

139

124

o después de la última liberación del paquete de servicio.

Además de contener fijos para los errores del programa, los paquetes de servicio también pueden contener parches de seguridad desarrollados específicamente para reparar las vulnerabilidades encontradas en los archivos de programa. Las vulnerabilidades del programa descubiertas después de que es liberado un producto puede ser una amenaza de seguridad significativa de ataque de hackers y virus a nivel mundial. Por lo tanto, una vez se descubre la vulnerabilidad, la distribución puntual y de amplio esparcimiento y la instalación de los parches de seguridad a los computadores que tienen software vulnerable es de capital importancia. Teóricamente, el uso de paquetes de servicio para lograr tal distribución puntual y de amplio esparcimiento de parches de seguridad puede ser efectivo. Por ejemplo, cuando un vendedor de software descubre una vulnerabilidad y luego desarrolla un parche de seguridad, el parche puede ser colocado en el último paquete de servicio en un sitio Web del vendedor para que los usuarios lo bajen e instalen inmediatamente. Esto podría frustrar la mayoría de los

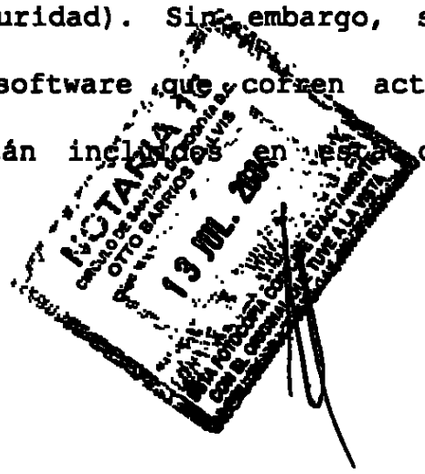


LCO

125

hackers y virus que pretenden explotar la vulnerabilidad descubierta. Sin embargo, los administradores del sistema y otros usuarios del producto de software habitualmente enfrentan varios inconvenientes y/o dificultades relacionadas con accesar e instalar parches de seguridad. Estas dificultades dan como resultado típicamente una distribución significativamente inferior de tales parches de la que es pretendida por el vendedor quien desarrolla el parche. El resultado es que las vulnerabilidades en muchos computadores a nivel mundial son dejadas sin parche, exponiendo tales computadores a un riesgo significativo.

Una dificultad con el acceso e instalación de parches de seguridad es que los métodos habituales para detectar si un computador está corriendo un software con una vulnerabilidad conocida requiere el uso activo y el involucramiento del computador. Por ejemplo, los métodos habitualmente disponibles pueden determinar si versiones particulares de productos de software en un computador están necesitadas de ser actualizadas (por ejemplo con un parche de seguridad). Sin embargo, solamente aquellos productos de software que corren activamente sobre el computador están incluidos en esta determinación. Los



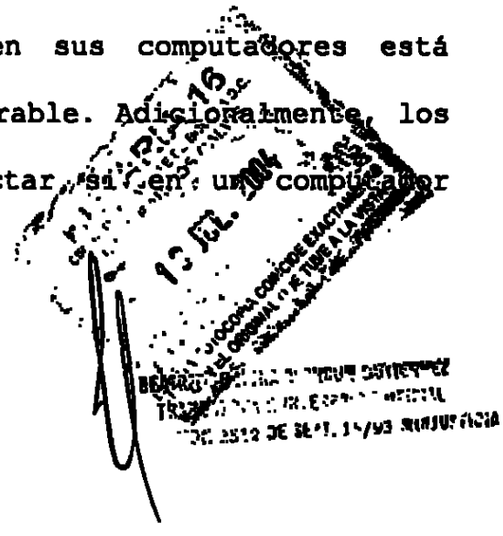
BEATRIZ EUGENIA ALARCON GUTIERREZ
TRADUCTORA E INTERPRETE OFICIAL
DECRETO 2512 DE SEPT. 15/93 JURISTICA

161

126

sistemas operativos secundarios y las aplicaciones que no están corriendo activamente en un computador no son consideradas, y por lo tanto podrían tener una vulnerabilidad de seguridad que pasa desapercibida y no es fija. Para aquellos productos que corren activamente en un computador, un usuario puede revisar una lista de actualizaciones disponibles y seleccionar las actualizaciones para instalación. Algunas actualizaciones pueden ser actualizaciones críticas diseñadas para proteger un computador de vulnerabilidades de seguridad conocidas. Varias actualizaciones requieren que el usuario reinicie el computador antes de que se complete la instalación. Además, un usuario debe seleccionar activamente las actualizaciones e instalarlas. Por estas y otras razones, los métodos habituales para acceder e instalar los parches de seguridad son menos que efectivos.

Otra dificultad para accesar e instalar parches de seguridad es aquella de saber si un parche de seguridad es necesario en un computador. Algunas veces es difícil para los usuarios saber si en sus computadores está corriendo software que es vulnerable. Adicionalmente, los métodos disponibles para detectar si en un computador

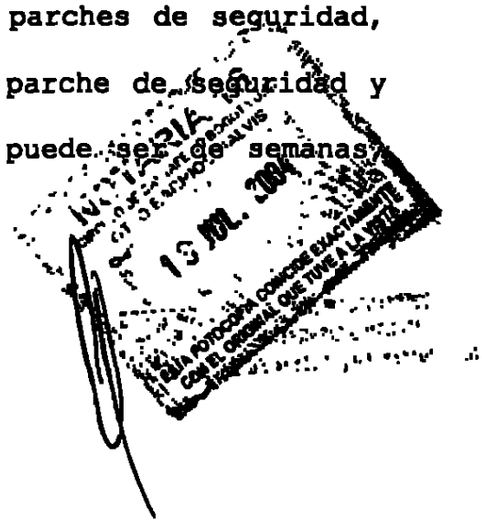


162

127

está corriendo software con una vulnerabilidad conocida puede no ser capaz de detectar ciertas configuraciones de un producto de software conocido por ser vulnerable. Por ejemplo, las versiones compartidas de algunos productos de software se pueden distribuir como parte de otros productos. De esta manera, aunque la versión compartida de un producto puede contener la misma vulnerabilidad que la versión completa del producto, la versión compartida puede no ser reconocida como un producto que necesita una actualización del parche de seguridad. Así, las versiones compartidas de los productos de software que son conocidas por tener vulnerabilidades de seguridad a menudo no son fijas.

Otros problemas con el accesamiento y la instalación de parches de seguridad se relaciona con el método convencional de "paquete de servicio" por medio del cual tales parches son entregados. Bajar e instalar paquetes de programa es un proceso que consume tiempo y es manual que muchos administradores de sistema simplemente no tienen tiempo de efectuar. Por lo tanto, aún cuando los administradores pretendan instalar parches de seguridad, el tiempo entre la liberación del parche de seguridad y su instalación en un sistema dado puede ser de semanas,

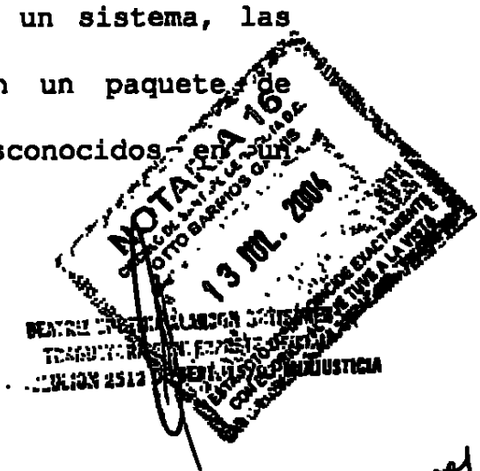


163

128

meses, o años. Así, el riesgo de ataque a través de una vulnerabilidad de seguridad puede no aliviarse en tales sistemas hasta mucho después de que el vendedor de software ha lanzado un parche de seguridad.

Adicionalmente, los administradores de sistema a menudo escogen no bajar e instalar los paquetes de servicio que contienen parches de seguridad, aunque ellos entienden los riesgos de seguridad relevantes. La razón para esto es que la instalación de un paquete de servicio mismo conlleva el riesgo de regresiones del sistema que puede introducir cambios no deseados en el comportamiento del sistema. Los administradores a menudo dedican tiempo y esfuerzo significativo a depurar un sistema de tal forma que este funciones como se desea. Como se mencionó anteriormente, sin embargo, los paquetes de servicio representan una evolución de una versión previa de un producto de software que incluye las actualizaciones más recientes a una base de código de producto (es decir el alcance de los cambios no está restringido solamente a los parches de seguridad). Además de introducir comportamientos nuevos y pretendidos en un sistema, las actualizaciones de código recientes en un paquete de servicio pueden introducir errores desconocidos en un



129

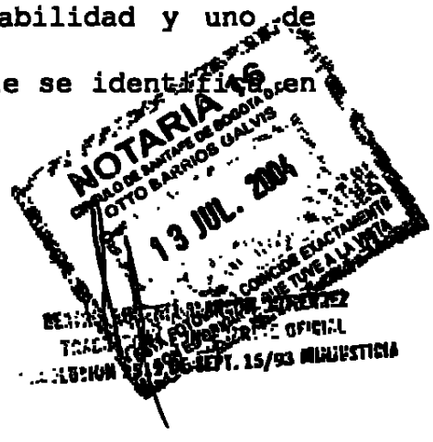
sistema que puede originar que el sistema se comporte de manera inesperada, lo cual, a su vez puede crear problemas significativos para un administrador de sistemas. De esta manera, los sistemas frecuentemente no son actualizados con los parches de seguridad importantes destinados a fijarse en los archivos de programa vulnerables, por que los administradores no quieren riesgos de regresiones.

De acuerdo con esto, subsiste la necesidad de una forma de implementar parches de vulnerabilidades de seguridad en archivos de programa de una manera automática, comprensiva, confiable y libre de regresiones.

RESUMEN

Los Parches de seguridad automáticos, comprensivos, confiables y libres de regresión de los archivos de programa binarios se describen aquí.

De acuerdo con una implementación, es recibida una firma binaria de un parche de vulnerabilidad y uno de seguridad. Un archivo binario vulnerable se identifica en



165

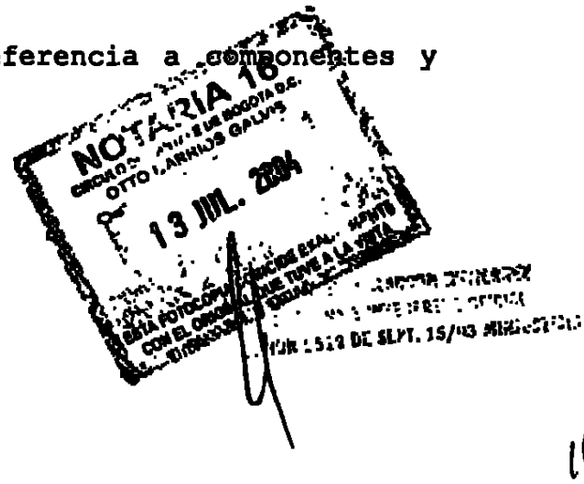
un computador con base en la firma binaria de una vulnerabilidad. El archivo binario vulnerable en un computador es actualizado con el parche de seguridad.

De acuerdo con otra implementación, es recibida una firma binaria que identifica una vulnerabilidad de seguridad en un archivo binario. El parche de seguridad configurado para fijarse a la vulnerabilidad de seguridad también se recibe. La forma binaria y el parche de seguridad son distribuidos a una pluralidad de servidores.

De acuerdo con otra implementación, se recibe una firma binaria de un servidor que se utiliza para buscar archivos binarios. Se envía una solicitud de un parche de seguridad al servidor si la firma binaria se encuentra en un archivo binario. El archivo binario es luego actualizado con el parche de seguridad.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

Los mismos numerales de referencia son utilizados a lo largo de los dibujos con referencia a componentes y características similares.



131

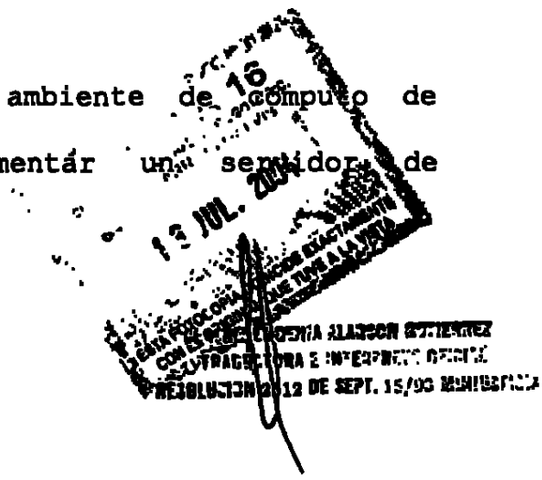
La figura 1 ilustra un ambiente de red de ejemplo adecuada para implementar la detección automática y el parcheo de vulnerabilidades de seguridad en archivos binarios.

La figura 2 ilustra una modalidad de ejemplo de un servidor de distribución, un servidor de exploración de parches, y un computador de cliente adecuado para implementar la detección automática y el parcheo de vulnerabilidad de seguridad en archivos binarios.

La figura 3 ilustra otra modalidad de ejemplo de un servidor de distribución, un servidor de exploración de parches, y un computador de cliente adecuado para implementar la detección automática y el parcheo de vulnerabilidades de seguridad en archivos binarios.

Las figuras 4-6 ilustran diagramas de bloques de métodos de ejemplo para implementar la detección automática y el parcheo de vulnerabilidades de seguridad en archivos binarios.

La figura 7 ilustra un ambiente de cómputo de ejemplo adecuado para implementar un servidor de



162

132

distribución, un servidor de exploración de parche, y un computador de cliente.

DESCRIPCIÓN DETALLADA

Revisión

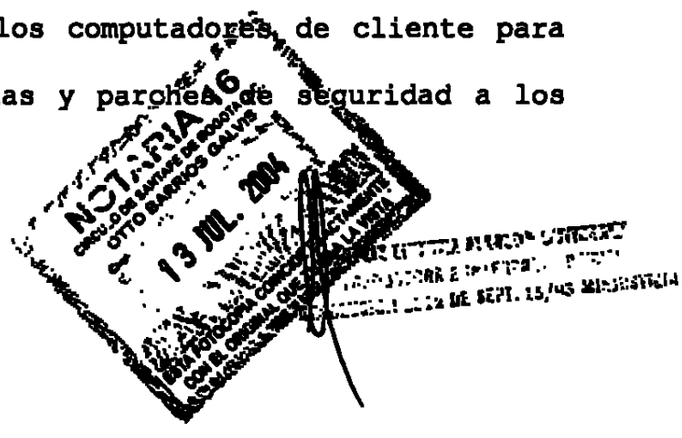
La siguiente discusión está dirigida a sistemas y métodos que posibilitan parchear vulnerabilidad de seguridad en archivos binarios. La detección y el parcheo de archivos binarios vulnerables es automática, confiable, libre de regresión, y comprensiva a través de las redes a una escala ilimitada. Estas ventajas se pueden evidenciar de varias maneras incluyendo, por ejemplo, al apalancar infraestructura antivirus habitual que es ampliamente desplegada a través de la Internet. Una divergencia de parches de seguridad alejada de los paquetes de servicio convencionales suministra la posibilidad de producción de fijaciones libres de regresión para vulnerabilidades de seguridad en archivos binarios.

El descubrimiento confiable de archivos binarios vulnerables (por ejemplo en sistemas operativos,

168

133

programas de aplicación, etc.) se logra a través del uso de firmas binarias que han sido asociadas con vulnerabilidades de seguridad. Las firmas binarias asociadas con vulnerabilidades de seguridad en archivos binarios, junto con parches de seguridad desarrollados para fijarse a tales vulnerabilidades de seguridad, son cargadas a un servidor de distribución central. El servidor de distribución se configura para distribuir las firmas binarias y los parches de seguridad de manera amplia a través de varias redes tales como la Internet. El uso de un servidor de distribución central para actualizar los servidores de redes (por ejemplo a través de la Internet) suministra cubrimiento de parche comprensivo y automático a una escala ilimitada. Los servidores de red que reciben tales actualizaciones pueden explorar los computadores de los clientes dentro de redes subordinadas para localizar los archivos vulnerables de acuerdo con las firmas binarias, y luego actualizar aquellos computadores que se encuentre que tengan archivos de seguridad vulnerables que utilizan los parches de seguridad correspondientes que se fijarán a los archivos vulnerables. Loa servidores de red también se puede comunicar con los computadores de cliente para transferir formas binarias y parches de seguridad a los



169

134

computadores de tal forma que la exploración y actualización se pueda desarrollar por medio de los computadores mismos. Niveles multianidados de redes subordinadas también pueden existir.

Ambiente de Ejemplo

La figura 1 ilustra un ambiente de red de ejemplo 100 adecuado para implementar la detección automática y el parcheo de vulnerabilidades de seguridad en archivos binarios. En el ambiente de red del ejemplo 100, un servidor de distribución central 102 está acoplado a múltiples servidores de exploración/parcheo 104 por vía de una red 106(a). Un servidor de exploraciones/parcheo 104 está típicamente acoplado a través de una red 106(b) a una pluralidad de computadores de cliente 108(1)-108(n). La red 106 está destinada a representar cualquiera de una variedad de topologías y tipos de red convencional (que incluyen redes y ópticas, alambicas y/o inalámbricas); que emplea cualquiera de una variedad de protocolos de red convencional (que incluye protocolos públicos y/o de propietarios). La red 106 puede incluir, por ejemplo, la Internet así como también posiblemente al menos porciones de una o más redes de área local (LAN)

120

135

y/o redes de área amplia (WAN). Las redes 106(a) y 106(b) puede ser la misma red tal como la Internet, o ellas pueden ser redes aisladas unas de otras tal como la Internet y una LAN corporativa.

El servidor de distribución 102 y los servidores de exploración/parcheo 104 son típicamente implementados como servidores de red estándar, y pueden ser cada uno cualquiera de una variedad de dispositivos de cómputos convencionales, que incluyen PCs de escritorio, notebook o computadores portátiles, estaciones de trabajo, computadores gigantes, accesorios para Internet, combinaciones de estos, y así sucesivamente. Uno o más de los servidores 102 y 104 pueden ser los mismos tipos de dispositivos, o alternativamente diferentes tipos de dispositivos. Un ambiente de cómputo de ejemplo para implementar un servidor de distribución 102 y un servidor de exploración/parcheo 104 se describe con más detalle aquí adelante con referencia a la figura 7.

Los computadores de cliente 108 funcionan en una relación cliente/servidor típica con un servidor 104 en donde clientes múltiples 108 hacen solicitudes a un servidor 104 que sirve las solicitudes. Los computadores

171

186

de cliente 108 pueden ser cualquiera de una variedad de dispositivos de cómputo convencionales, incluyendo PCS de escritorio, notebook o computadores portátiles, estaciones de trabajo, computadores grandes, consolas de juego, PCs tipo agenda, teléfonos celulares u otros dispositivos de comunicación inalámbricos, ayudantes digitales personales (PDA), combinaciones de estos, y así sucesivamente. Uno o más de los computadores de cliente 108 pueden ser los mismos tipos de dispositivos, o alternativamente diferentes tipos de dispositivos. Un ambiente de cómputo de ejemplo para implementar un computador de cliente 108 se describe con más detalle aquí adelante con referencia a la figura 7.

En general, la detección automática y comprensiva y el parcheo de archivos binarios vulnerables en computadores de clientes 108 se logra a través de actualizaciones hechas a través de servidor de distribución 102 que incluye firmas binarias para identificar archivos binarios vulnerables y parches de seguridad configurados para fijarse en archivos vulnerables. Como se discute con mayor detalle adelante con respecto a las siguientes modalidades de ejemplo, las firmas binarias y los parches de seguridad son

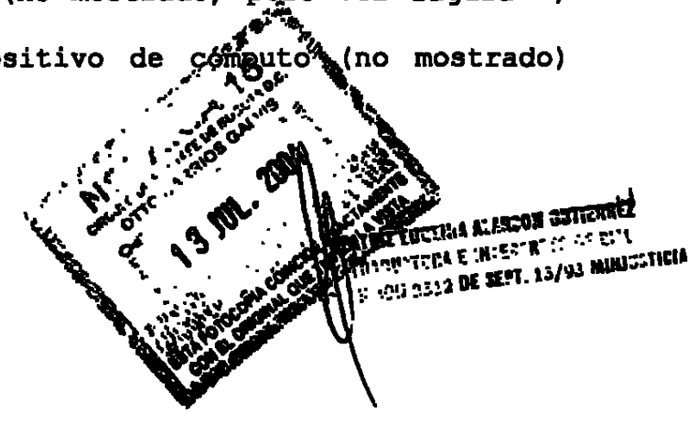
132

187

distribuidos para explorar/parchear servidores 104 lo cual a su vez, exploran activamente y actualizan archivos binarios vulnerables en los computadores de clientes 108, o derriban las firmas binarias y los parches de seguridad a los computadores de clientes 108 de tal forma que los computadores de cliente 108 puedan desarrollar la exploración y el parcheo de los archivos binarios vulnerables.

Modalidades de Ejemplo

La figura 2 ilustra una modalidad de ejemplo de un servidor de distribución 102, un servidor de exploración-parcheo 104 y un computador de cliente 108 adecuado para implementar detección automática y parcheo de vulnerabilidades de seguridad en archivos binarios. El servidor de distribución 112 incluye un módulo de distribución 200 y una base de datos 202 para recibir y mantener firmas binarias y parches de seguridad. La base de datos 202 puede ser actualizada con firmas binarias y parches de seguridad de una variedad de manera que incluye, por ejemplo, a través de un medio de almacenamiento portátil (no mostrado, pero ver figura 7) o a través de un dispositivo de cómputo (no mostrado)

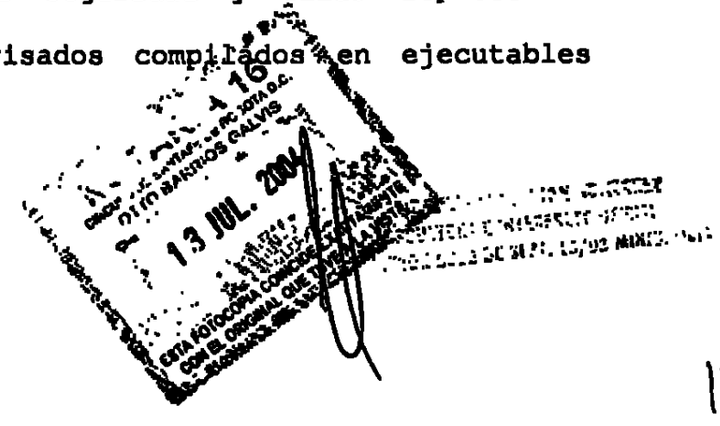


183

acoplado al servidor 102 y configurado para cargar firmas binarias y parches de seguridad a la base de datos 202.

Un escenario tipico en el cual la base de datos 202 podría ser actualizada inicia con una investigación de un producto de software (por ejemplo un sistema operativo, un programa de aplicación, etc.) iniciado por el desarrollador del producto de software. Por ejemplo, el desarrollador puede contratar una firma de consultoria en seguridad para intentar encontrar vulnerabilidades de seguridad en un producto de software recientemente liberado. Si se descubre una vulnerabilidad en el producto de software a través de "hacking" o por algunos otros medios, el patrón de bit exacto de la función vulnerable dentro del producto puede ser identificado. El patrón de bit representa una firma binaria de la sección vulnerable en el archivo binario, que es un componente de un producto de software.

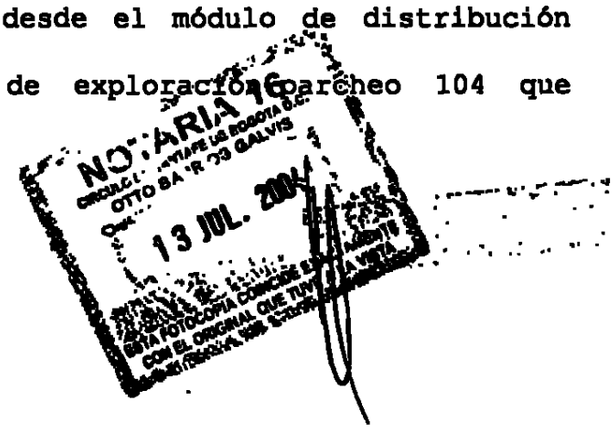
Una vez se descubre la vulnerabilidad de seguridad y se analiza, se puede desarrollar una fijación que eliminará la vulnerabilidad. Tales fijaciones son denominadas parches de seguridad y ellas representan módulos de código revisados compilados en ejecutables



189

binarios. Los parches de seguridad se pueden instalar en computadores que son identificados a través de la firma binaria como software que corre que tiene la vulnerabilidad de seguridad. La instalación del parche de seguridad fijará la vulnerabilidad de seguridad. El servidor de distribución 102 le posibilita a los vendedores de producto de software y a otros cargar firmas binarias de archivos binarios vulnerables junto con los parches de seguridad diseñados para fijar los archivos binarios vulnerables, en la base de datos 202 para distribución.

El módulo de distribución 200 se configura para distribuir firmas binarias y parches de seguridad de bases de datos 202 a varios servidores de exploración-parcheo 104 por vía de la re 106. El módulo de distribución 200 típicamente funciona automáticamente para distribuir firmas binarias y parches de seguridad de la base de datos 202 siempre y cuando la base de datos 202 esté actualizada con las firmas y los parches adicionales. La distribución automática se puede lograr de una variedad de maneras incluyendo, por ejemplo, a través de comunicación desde el módulo de distribución 200 a los servidores de exploración-parcheo 104 que



1253

140

indican que están disponibles las firmas binarias actualizada y los parches de seguridad y esperan por solicitudes para enviar las firmas binarias y los parches de seguridad, o al enviar automáticamente firmas binarias actualizadas y parches de seguridad a los servidores de exploración-parcheo 104 configurados para aceptar las actualizaciones.

En la modalidad De la figura 2, el servidor de exploración-parcheo 104 incluye un módulo de exploración-parcheo 204 y una base de datos 206 para recibir y soportar firmas binaras y parches de seguridad. La base de datos 206 es típicamente actualizada automáticamente con nuevas firmas binarias y parches de seguridad a través de la comunicación entre el módulo exploración-parcheo 204 y el módulo de distribución 200 en el servidor de distribución 102. Además de actualizar la base de datos 206 con firmas binaras y parches de seguridad, el módulo de exploración-parcheo 204 se configura para accesar computadores de clientes 108 y explorar archivos binarios 208 para firmas binarias. Explorar archivos binarios 208 puede incluir buscar una firma binaria en archivos binarios presentes en cualquier forma de medios puentes en o accesibles por el computador

186

141

de cliente 108. Los archivos binarios 208 típicamente incluyen códigos de computador/leíbles por procesador, compilados tal como un sistema operativo o un archivo de programa de aplicación. Sin embargo, se nota que los archivos binarios 208 pueden ser cualquier forma de información binaria que incluye instrucciones de computador/leíbles por procesador, estructuras de datos, módulos de programa, y otros datos para computador de cliente 108.

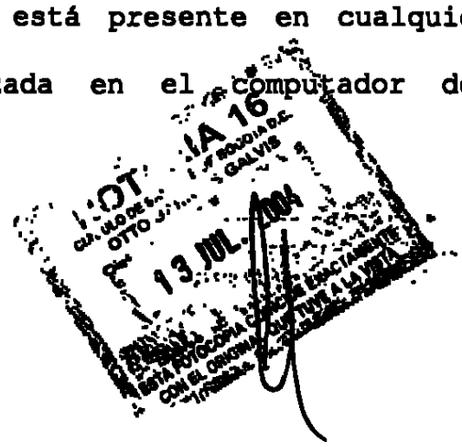
Como se anotó anteriormente en la discusión en referencia al ambiente del computador de ejemplo de la figura 7, tales medios en un computador de cliente 108 pueden incluir cualquier medio disponible que sean accesibles por el computador del cliente 108, tal como medios volátiles y no volátiles así como también medios removibles y no removibles. Tales medios leíbles por computador/procesador pueden incluir memoria volátil, tal como la memoria de acceso aleatorio (RAM) y/o memoria no volátil, tal como la memoria de solo lectura (ROM). Los medios leíbles por computador/procesador pueden también incluir otros medios de almacenamiento en computador removibles/no removibles, volátiles/no volátiles, tales como, por ejemplo, unidad de disco duro para lectura

187

142

desde y escritura hacia los medios magnéticos no removibles, no volátiles, una unidad de disco magnético para lectura desde y escritura hacia un disco removible, no volátil (por ejemplo, un "floppy disk"), una unidad de disco óptico para lectura desde y/o escritura hacia un disco óptico removible, no volátil tal como un CD-ROM, DVD-ROM u otro medio óptico, otro dispositivo de almacenamiento magnético, tarjetas de memoria flash, memoria de solo lectura programable eléctricamente borrable (EEPROM), almacenamiento unido a red, y similar. Todos los medios leíbles por computador/procesador provistos con almacenamiento tanto volátil como no volátil de cualquier forma de archivos binarios 208, que incluye instrucciones leíbles por computador/procesador, estructuras de datos, módulos de programa, y otros datos para computador de cliente 108, son accesibles para exploración por medio de servidor de exploración-parcheo 104 por vía del módulo de exploración-parcheo 204.

El módulo de exploración-parcheo 204 busca así archivos binarios 208 en el computador del cliente 108 para determinar si una firma binaria que identifica una vulnerabilidad de seguridad está presente en cualquier información binaria localizada en el computador del

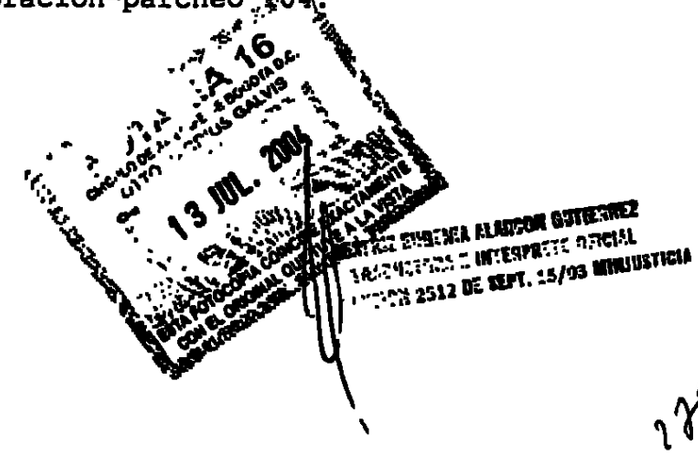


178

143

cliente 108. Si el patrón de bit de la firma binaria se encuentra en un archivo binario 208, el módulo de exploración-parcheo 204 opera para fijar la vulnerabilidad de seguridad de seguridad en el archivo binario 208 al instalar un parche de seguridad correspondiente en el computador del cliente 108. La instalación de un parche de seguridad en el computador del cliente 108 sobre escribe o de otra manera elimina el archivo binario a una porción del archivo binario que contiene la vulnerabilidad de seguridad.

La figura 3 ilustra otra modalidad de ejemplo de un servidor de distribución 102, un servidor de exploración-parcheo 104 y un computador de cliente 108 adecuado para implementar el parcheo de las vulnerabilidades de seguridad en los archivos binarios. En general, en la modalidad de la figura 3, las firmas binarias y los parches de seguridad son derribados, o redistribuidos, desde el servidor 104 del computador del cliente 108, y la exploración de archivos vulnerables por seguridad y el parcheo de los archivos vulnerables por seguridad se desarrollan mediante un computador de cliente 108 en lugar del servidor de exploración-parcheo 104.



281

144

En la modalidad de la figura 3, el servidor de distribución 102 es configurado de la misma manera como se discutió anteriormente con respecto a la modalidad de la figura 2. Así, la base de datos 202 se puede actualizar para incluir firmas binarias recientemente descubiertas que identifican vulnerabilidades de seguridad en archivos binarios. La base de datos 202 también se puede actualizar con los parches de seguridad correspondientes que han sido desarrollados para fijar tales vulnerabilidades de seguridad.

El servidor de exploración-parcheo 102 de la figura 3 se configura de alguna forma de la misma manera como aquel que se discutió anteriormente con respecto a la figura 2. Así, el servidor de exploración-parcheo 102 de la figura 3 incluye una base de datos 206 para recibir y mantener firmas binarias y parches de seguridad. La base de datos 206 es típicamente actualizado automáticamente con nuevas firmas binarias y parches de seguridad a través de las comunicaciones entre el servidor de exploración-parcheo 104 y el servidor de distribución 102. Sin embargo, la comunicación entre el servidor de exploración-parcheo 104 y el servidor de distribución 102 es conducida a través de un módulo de redistribución 300

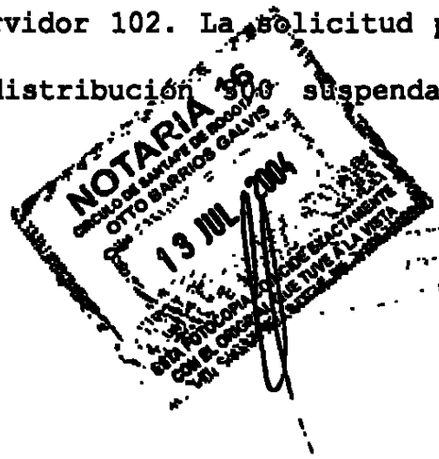
180

145

en lugar de un módulo de exploración-parcheo 204 como se discutió con respecto a la modalidad de a figura 2.

El módulo de redistribución 300, además de actualizar la base de datos 206 con firmas binarias y parches de seguridad, se configura para comunicarse con el módulo de exploración-parcheo 302 en el computador del cliente 108 y transferir una firma binaria al computador del cliente 108. el módulo de exploración-parcheo 302 se configura para recibir la firma binaria y para explorar los archivos binarios 208 para determinar si la firma binaria está presente en cualquier información binaria localizada en el computador del cliente 108. Así, el módulo de el módulo de exploración-parcheo 302 de la figura 3 funciona de manera similar al módulo de el módulo de exploración-parcheo 204 discutido anteriormente con referencia a la figura 2.

Si el patrón de bit de la firma binaria es encontrado en el archivo binario 208 en el computador de cliente 108, el módulo de el módulo de exploración-parcheo 302 envía una solicitud al módulo de redistribución 300 en el servidor 102. La solicitud para hacer que el módulo de redistribución 300 suspenda el



181

146

parche de seguridad que se corresponde con la firma binaria al computador del cliente 108. El módulo de redistribución 300 responde a la solicitud al enviar el parche de seguridad apropiado al computador del cliente 108. El módulo de el módulo de exploración-parcheo 302 recibe el parche de seguridad y opera para fijar la vulnerabilidad de seguridad en el archivo binario 208 al instalar el parche de seguridad en el computador del cliente 108. Como en la modalidad de la figura 2, la instalación de un parche de seguridad en el computador del cliente 108 sobre escribe o de otra manera elimina el archivo binario o una porción del archivo binario que contiene la vulnerabilidad de seguridades cubierta.

Métodos de Ejemplo

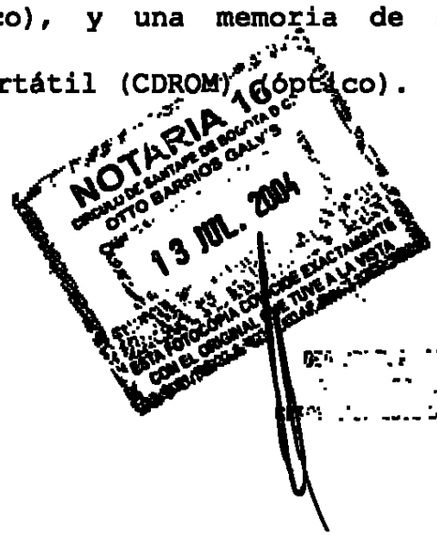
Los métodos de ejemplo para implementar la detección y el parcheo automático de vulnerabilidades de seguridad en archivos binarios será ahora descrita con referencia primaria a los diagramas de flujo de las figuras 4-6. Los métodos aplican de manera general a las modalidades de ejemplo discutidas anteriormente con respecto a las figuras 1-3. Los elementos de los métodos descritos se pueden desarrollar mediante cualquiera medios apropiados

18

147

incluyendo, por ejemplo, mediante bloques lógicos de hardware en un ASIC o mediante la ejecución de instrucciones leíbles por procesador definidas en un medios leíble por procesador.

Un "medio leíble por procesador", como se utiliza aquí, puede ser cualquiera medios que puedan contener, almacenar, comunicar, propagar, o transportar instrucciones para uso por o ejecución de un procesador. Un medio leíble por procesador puede ser, sin limitación, sistema, aparato, dispositivo o medio de propagación electrónico, magnético, óptico, electromagnético, infrarrojo o semiconductor. Ejemplos más especificos de un medio leíble por procesador incluye, entre otros, una conexión eléctrica (electrónica) que tiene uno o más alambres, un disquete portátil de computador (magnético), una memoria de acceso aleatoria (RAM) (magnético), una memoria de solo lectura (ROM) (magnético), una memoria de solo lectura programable borrable (EPROM o memoria Flash), una fibra óptica (óptica) un disco compacto reescribible (CD-RW) (óptico), y una memoria de solo lectura de disco compacto portátil (CDROM) (óptico).



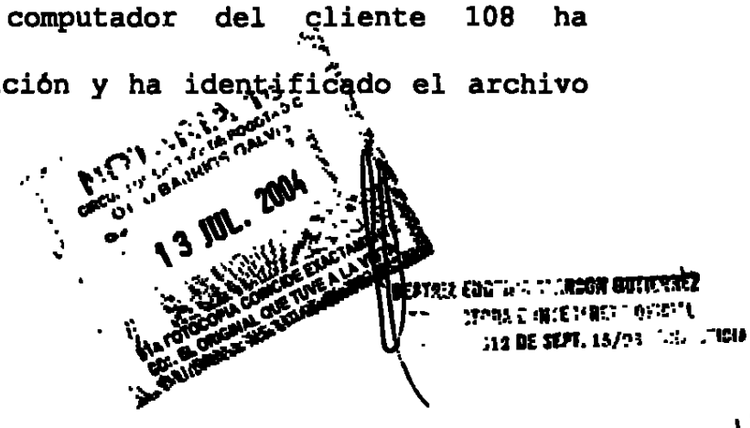
183

La figura 4 muestra un método de ejemplo 400 para implementar la detección y el parcheo automáticos de vulnerabilidades de seguridad en archivos binarios. Los archivos binarios están típicamente localizados o almacenados en un computador de cliente que está siendo servido por un computador servidor, pero ellos también pueden estar localizados en el computador servidor mismo, o en cualquier otro dispositivo de cómputo accesible por medio de un computador servidor. En el bloque 402 del método 400 se recibe una firma binaria. La firma binaria es un patrón de bit que se ha asociado con una vulnerabilidad de seguridad en un archivo binario particular, tal como un programa de aplicación ejecutable o un sistema operativo que corre en un computador de cliente. La firma binaria se recibe de un servidor de distribución central 102 mediante un servidor subordinado 104.

En el bloque 404, se recibe un parche de seguridad. El parche de seguridad es típicamente código ejecutable compilado que ha sido desarrollado como un fijo a la vulnerabilidad de seguridad del archivo binario particular. El parche de seguridad también es recibido del servidor de distribución central 102 mediante un

servidor subordinado 104. En el bloque 406, se identifica un archivo binario vulnerable basado en la firma binaria. La identificación del archivo binario vulnerable es típicamente lograda mediante información binaria explorable almacenada en medios varios de un computador, tal como un computador de cliente 108, y luego comparar el o los patrones en la firma binaria con la información binaria encontrada en los medios. La identificación puede ocurrir de varias maneras incluyendo, por ejemplo, mediante el servidor 104 que explora y compara toda la información binaria presente en el computador del cliente. La identificación de un archivo binario vulnerable también se puede lograr al tener el servidor 104 derribada la firma binaria al computador del cliente de tal forma que el computador del cliente puede desarrollar la exploración y comparación.

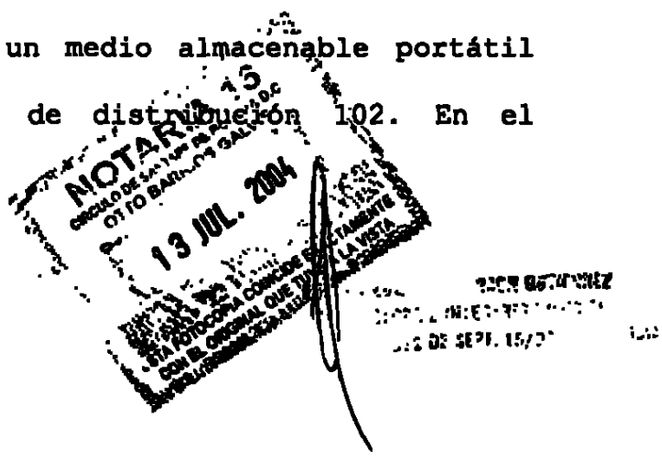
En el bloque 408 del método 400, se utiliza el parche de seguridad para actualizar el archivo binario vulnerable. La actualización se puede lograr de varias maneras incluyendo, por ejemplo, mediante el servidor 104 que instala el parche de seguridad en el computador del cliente 108. Si el computador del cliente 108 ha desarrollado la exploración y ha identificado el archivo



150

binario vulnerable, el computador del cliente 108 puede solicitar que el servidor 104 envíe el parche de seguridad al computador 108, en cuyo caso el computador 108 puede instalar el parche de seguridad para fijar el archivo binario vulnerable.

La figura 5 muestra otro método de ejemplo 500 para implementar detección y parcheo automáticos de vulnerabilidades de seguridad en archivos binarios. El método 500 ilustra de manera general la distribución de firmas binarias para vulnerabilidades de seguridad y parches de seguridad desarrollados para fijar aquellas vulnerabilidades de seguridad. En el bloque 502 del método 500, se recibe una firma binaria que identifica una vulnerabilidad de seguridad de un archivo binario. La firma binaria es típicamente cargada a un servidor de distribución 102 como un patrón de bit recientemente descubierto que identifica una vulnerabilidad de un archivo binario de un producto de software que puede ser ampliamente distribuido a través de muchos computadores en una red tal como la Internet. La carga celebra típicamente desde un computador acoplado al servidor de distribución 102 o desde un medio almacenable portátil insertado en el servidor de distribución 102. En el

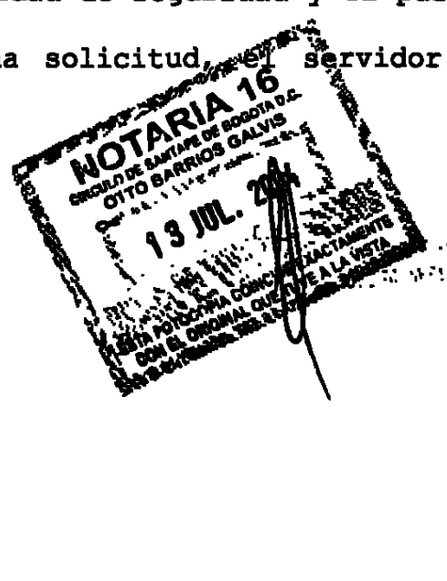


160

151

bloque 504, se recibe un parche de seguridad configurado para fijar la vulnerabilidad de seguridad mediante el servidor de distribución 102 de una manera similar que la firma binaria.

En el bloque 506, la firma binaria y el parche de seguridad son distribuidos a una pluralidad de servidores subordinados 104 desde el servidor de distribución 102. Esta distribución ocurre automáticamente y se puede lograr de varias maneras. Por ejemplo, al recibir una firma binaria cargada y el parche de seguridad, el servidor de distribución 102 puede automáticamente despachar la firma binaria y el parche de seguridad sobre la red a todos los servidores subordinados 104 configurados para recibir las firmas binarias actualizadas y los parches de seguridad. El servidor de distribución 102 también podría enviar una nota a los servidores 104 indicando que se ha descubierto una vulnerabilidad de seguridad y que el parche de seguridad fue disponible para fijar la vulnerabilidad. Los servidores subordinados 104 pueden solicitar entonces que el servidor de distribución 102 envíe la firma binaria que identifica la vulnerabilidad de seguridad y el parche de seguridad. Al recibir una solicitud, el servidor de



180

152

distribución 102 puede enviar la firma binaria y el parche de seguridad a los servidores que lo solicitan 102.

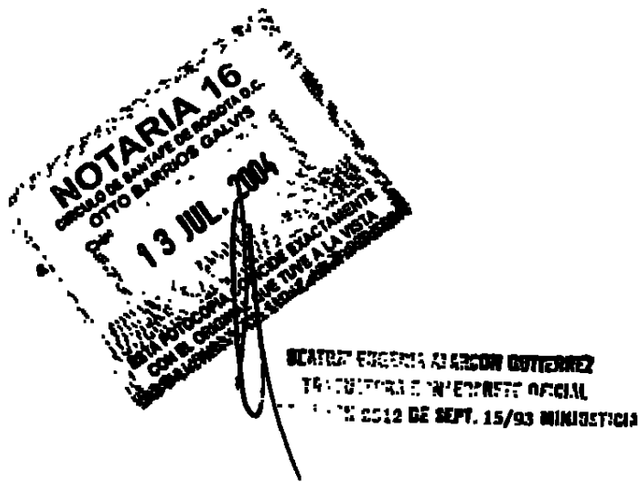
La figura 6 muestra otro método de ejemplo 600 para implementar la detección automática y el parcheo de las vulnerabilidades de seguridad n los archivos binarios. El bloque 602 del método 600, un computador de cliente 108 recibe una firma binaria de un servidor 104. La firma binaria está asociada con una vulnerabilidad de seguridad en un archivo binario que puede estar presente en el computador del cliente 108. En el bloque 604, el computador del cliente 108 explora toda la información binaria actualmente disponible en este y la compara con el o los patrones en la firma binaria con la información binaria, La información binaria explorada por el computador del cliente 108 está típicamente en la forma de instrucciones leíbles por computador/procesador y/o ejecutables, estructuras de datos, módulos de programa, y otros datos útiles para el computador del cliente 108, y puede residir tanto en los medios de almacenamiento volátiles como no volátiles de varios tipos.

188

153

En el bloque 606, si el computador del cliente 608 encuentra un archivo binario que contiene la firma binaria, este envia una solicitud al servidor 104 para hacer transferir el parche de seguridad. En el bloque 608, el computador del cliente 108 recibe del parche de seguridad, y en el bloque 610, del computador del cliente 108 instala el parche de seguridad con el fin de fijar la vulnerabilidad de seguridad en el archivo binario que contiene la información binaria que casa con el o los patrones de la firma binaria.

Aunque uno o más métodos se han descrito por medio de los diagramas de flujo y el texto asociado con los bloques de los diagramas de flujo, se debe entender que los bloques no necesariamente tienen que ser desarrollados en el orden en el cual ellos fueron presentados, y que un orden u órdenes alternativos pueden resultar en ventajas similares. Adicionalmente, los métodos no son exclusivos y si se pueden desarrollar solos o en combinación uno con el otro.



189

154

Computador de Ejemplo

La figura 7 ilustra un ambiente de cómputo de ejemplo adecuado para implementar un servidor de distribución 102, un servidor de exploración-parche 104 y un computador de cliente 108, como se discutió anteriormente con referencia a las figuras 1-3. Aunque una configuración específica se muestra e la figura 7, el servidor de distribución 102, el servidor de exploración-parcheo 104, y el computador de cliente 108 se puede implementar en otras configuraciones de cómputo.

El ambiente de cómputo 700 incluye un sistema de cómputo del propósito general en la forma de un computador 702. Los componentes del computador 702 pueden incluir, pero no están limitados a, uno o más procesadores a unidades de procesamiento 704, una memoria el sistema 706, y un bus del sistema 708 que acopla varios componentes del sistema incluyendo el procesador 704 a la memoria del sistema 706.

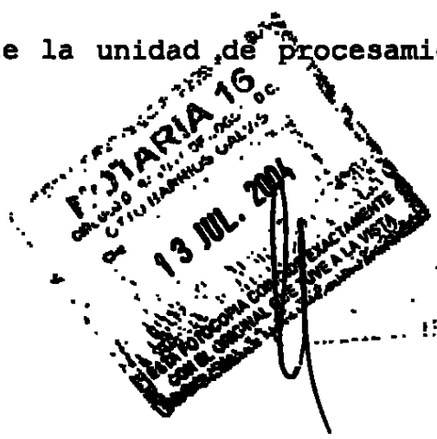
El bus del sistema 708 representa una o más de cualquiera de los varios tipos de estructuras de bus, incluyendo un bus de memoria o un controlador de memoria,

190

165

un bus periférico, un puerto de gráfico 2 acelerado, y un procesador o un bus local que utiliza cualquiera de una variedad de arquitecturas de bus. Un ejemplo de un bus del sistema 708 sería un bus de Interconexión de Componente Periférico (PCI) también conocido como bus Mezanine.

El computador 702 incluye típicamente una variedad de medios leíbles por computador. Tales medios pueden ser cualquiera medios disponibles que sean accesibles por computador 702 y que incluyen tanto medios volátiles como no volátiles, medios removibles como no removibles. La memoria del sistema 706 incluye medios leíbles por computador en la forma de memoria volátil, tal como memoria de acceso aleatorio (RAM) 710, y/o memoria no volátil, tal como memoria de solo lectura (ROM) 712. Un sistema de entrada/salida básico (BIOS) 714, que contiene las rutinas básicas que ayudan a transferir la información entre elementos dentro del computador 702, tal como durante el arranque, se almacena en la ROM 712. La RAM 710 contiene típicamente datos y/o módulos de programa que son inmediatamente accesibles a y/o actualmente operados mediante la unidad de procesamiento 704.

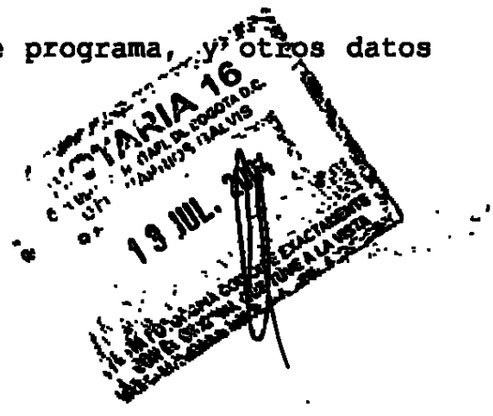


141

156

El computador 702 puede también incluir otros medios de almacenamiento en computador removibles/no removibles, volátiles/no volátiles. Por vía de ejemplo, la figura 7 ilustra una unidad de disco duro 714 para leer desde y escribir hacia medios magnéticos no volátiles (no mostrados), una unidad de disco magnético 718 para leer desde y escribir hacia un disco magnético removible, no volátil 720 (por ejemplo un "floppy disk"), y una unidad de disco óptico 722 para leer desde y/o escribir hacia un disco óptico removible, no volátil 724 tal como un CD-ROM, DVD-ROM, u otros medios ópticos. La unidad de disco duro 716, la unidad de disco magnético 718, la unidad de disco óptico 722 y cada uno conectado al bus del sistema 708 por uno o más interfaces de medios de datos 726. Alternativamente, la unidad de disco duro 716, la unidad de disco magnético 718, y la unidad de disco óptico 722 se pueden conectar al bus del sistema 708 mediante una interfase SCSI (no mostrada).

Las unidades de disco y sus medios leíbles por computador asociados suministran almacenamiento no volátil de instrucciones leíbles por computador, estructuras de datos, módulos de programa, y otros datos

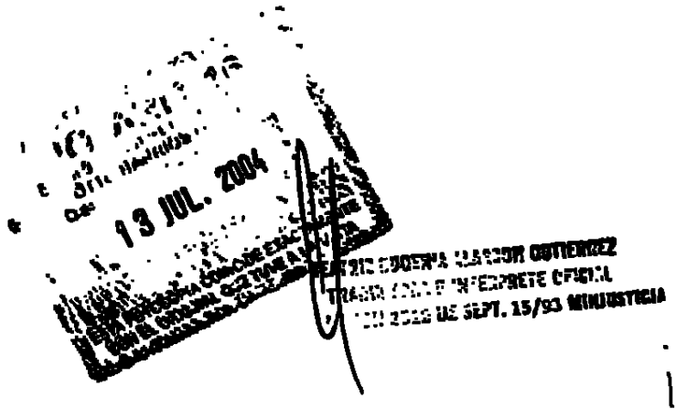


147

159

para computador 702. Aunque el ejemplo ilustra un disco duro 716, un disco magnético removible 720, y un disco óptico removible 724, se debe apreciar que otros tipos de medios leíbles por computador que pueden almacenar datos que son accesibles por un computador tal como cartuchos magnéticos y otros dispositivos de almacenamiento magnético, tarjetas de memoria flash, CD-ROM, discos versátiles digitales (DVD) u otros almacenamientos ópticos, memorias de acceso aleatorio (RAM), memorias de solo lecturas (ROM), memorias de solo lectura programables eléctricamente borrables (EEPROM), y similares, también Se pueden utilizar para implementar el sistema de cómputo de ejemplo y el ambiente.

Cualquier número de módulos de programa se puede almacenar en el disco duro 716, el disco magnético 720, el disco óptico 724, ROM 712, y/o RAM 710, incluyendo por ejemplo, un sistema operativo 726, uno o más programas de aplicación 728, otros módulos de programa 730, y datos de programa 732. Cada uno de tales sistemas operativos 726 uno o más programas de aplicación 728, otros módulos de programa 730, y los datos de programa 732 (o alguna combinación de estos) puede incluir una modalidad de un

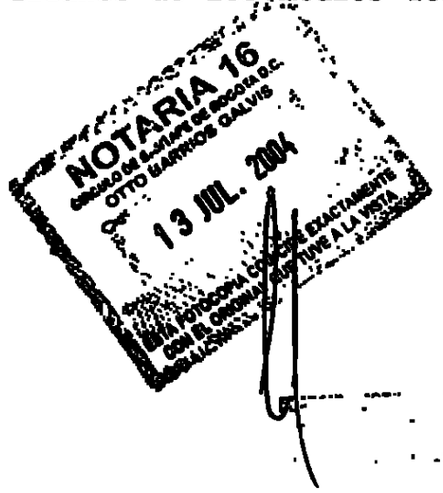


193

158

esquema de caché para utilizar información de acceso a redes.

El computador 702 puede incluir una variedad de medios leíbles por computador/procesador identificados como medios de comunicación. Los medios de comunicación típicamente son modalidades de instrucciones leíbles por computador, estructuras de datos, módulos de programa, u otros datos en señales de datos modulados tales como una onda portadora u otro mecanismo de transporte e incluye cualquier medio de suministro de información. El término "señal de datos modulada" significa una señal que tiene una o más de sus características establecida o cambiada de tal manera que codifica la información en la señal. Por vía de ejemplo, y no de limitación, los medios de comunicación incluyen medios alámbricos tales como redes alámbricas o conexiones alámbricas directas, y medios inalámbricos tales como medios acústicos, RF, infrarrojos, y otros medios inalámbricos. Las combinaciones de cualquiera de los anteriores también están incluidos dentro del alcance de los medios leíbles por computador.



159

Un usuario puede ingresar comandos de información dentro de un sistema de un sistema de computador 702 por vía de los dispositivos de entrada tales como un teclado 734 y un dispositivo de punteo 736 (por ejemplo, un "ratón"). Otros dispositivos de entrada 738 (no específicamente mostrados) pueden incluir un micrófono, joistick, game pad, disco satelital, puerto serial, explorador, y/o similar. Estos y otros dispositivos de entrada están conectados a la unidad de procesamiento 704 por vía de las interfases entrada/salida 740 que están acopladas al bus del sistema 708, pero que puede estar conectados mediante otra interfase y estructuras de bus, tal como un puerto paralelo, un puerto de juegos, o un bus serial universal (USB).

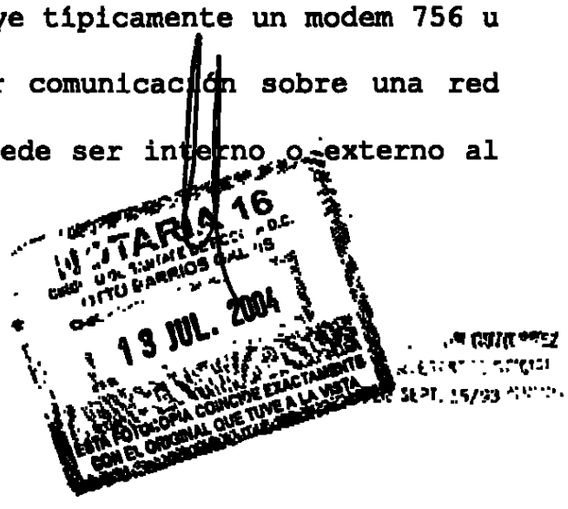
Un monitor 742 u otro tipo de dispositivo de exhibición se pueden conectar al bus del sistema 708 por vía de una interfase, tal como un adaptador de video 144. Además del monitor 742, otros dispositivos periféricos de salida pueden incluir componentes tales como parlantes (no mostrados) y una impresora 746 que puede estar conectada al computador 702 por vía de las interfases de entrada/salida 740.

193

160

El computador 702 puede operar en un ambiente de red utilizando conexiones lógicas a uno o más computadores remotos, tal como un dispositivo de cómputo remoto 748. Por vía de ejemplo, el dispositivo de cómputo remoto 748 puede ser un computador personal, computador portátil, un servidor, un enrutador, un computador de red, un dispositivo homólogo u otro nodo de red común, y similares. El dispositivo de computo remoto 748 se ilustra como un computador portátil que puede incluir muchos o todos los elementos y características descritas aquí con relación al sistema de computador 702.

Las conexiones lógicas entre el computador 702 y el computador remoto 748 se describen como una red de área local (LAN) 750 y una red de área amplia general (WAN) 752. Tales ambientes de red son sitios comunes en oficinas, redes de computadores amplias de empresas, intranet, y la Internet. Cuando se implementa en un ambiente de red LAN, el computador 702 está conectado a una red local 750 por vía de una interfase de red o adaptador 754. Cuando se implementa en un ambiente de red WAN, el computador 702 incluye típicamente un modem 756 u otros medios para establecer comunicación sobre una red amplia 752. El modem 756, puede ser interno o externo al



196

161

computador 702, puede estar conectado al bus del sistema 708 por vía de las interfases de entrada/salida 740 u otros mecanismos apropiados. Se debe apreciar que las conexiones de red ilustradas son ejemplos y que otros medios de establecer enlaces de comunicación entre los computadores 702 y 748 se pueden emplear.

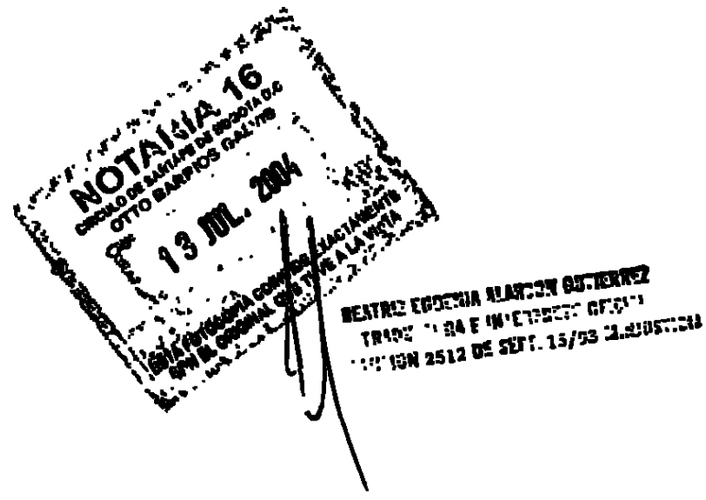
En un ambiente de red, tal como aquel ilustrado con el ambiente del cómputo 700, los módulos del programa descritos con relación al computador 702, o porciones de estos, se pueden almacenar en un dispositivo de almacenamiento o de memoria remoto. Por vía de ejemplo, los programas de aplicación remota 758 residen en un dispositivo de memoria del computador remoto 748. Para propósitos de ilustración, los programas de aplicación y otros componentes de programa ejecutables, tales como el sistema operativo, son ilustrados aquí como bloques discretos aunque se reconoce que tales programas y componentes residen en varios tiempos en diferentes componentes de almacenamiento del sistema de computador 702, y son ejecutados por el o los procesadores de datos del computador.

198

162

Conclusión

Aunque la invención se ha descrito en lenguaje específico a las características estructurales y/o actos metodológicos, se debe entender que la invención definida en las reivindicaciones finales no está necesariamente limitada a las características específicas o actos descritos. En su lugar, las características y actos específicos son descritos como formas de ejemplo de implementar la invención reivindicada.



198

163

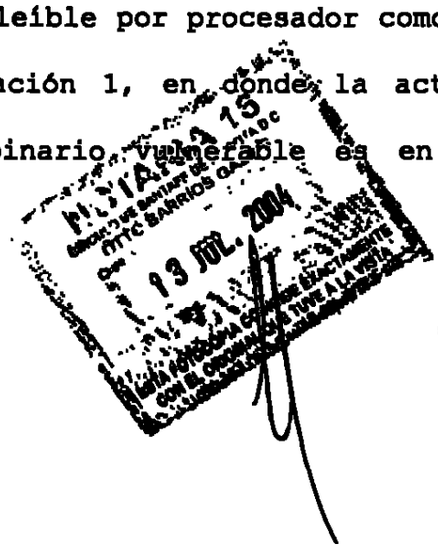
REIVINDICACIONES

1. Un medio leíble por procesador que comprende instrucciones ejecutables por procesador configuradas para:

   recibir una firma binaria;
   recibir un parche de seguridad;
   identificar un archivo binario vulnerable en un computador basado en firma binaria; y
   actualizar el archivo binario vulnerable en el computador con el parche de seguridad.

2. Un medio leíble por procesador como se cita en la reivindicación 1, en donde el identificar un archivo binario vulnerable en un computador incluye comparar un patrón de bit de las firma binaria contra los archivos binarios localizados en el computador, el patrón de bit asociado con una vulnerabilidad de seguridad.

3. Un medio leíble por procesador como se cita en la reivindicación 1, en donde la actualización del archivo binario vulnerable es en el computador



164

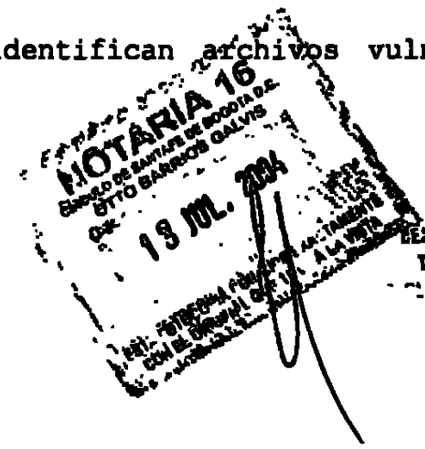
incluye instalar el parche de seguridad en el computador.

4. Un medio leíble por procesador como se cita en la reivindicación 1, en donde el identificar un archivo binario vulnerable en un computador incluye enviar la firma binaria al computador.

5. El medio leible por procesador como se cita en la reivindicación 4, en donde la actualización del archivo binario vulnerable en el computador incluye:

Recibir una solicitud de un computador para enviar el parche de seguridad; y enviar el parche de seguridad al computador.

6. Un medio leíble por procesador como se cita en la reivindicación 1, en donde el computador es un computador de cliente y el recibir incluye recibir la firma binaria y el parche de seguridad desde un servidor de distribución configurado para distribuir al computador del cliente, firmas binarias que identifican archivos vulnerables y



200

165

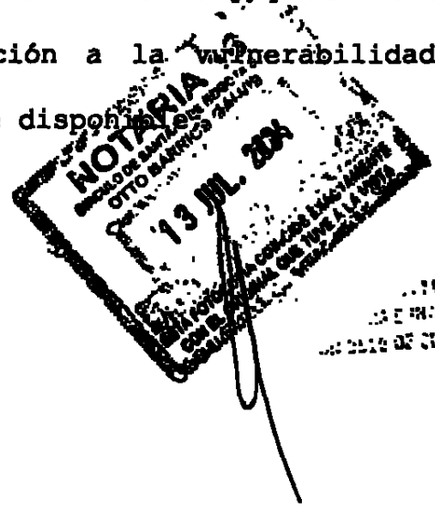
parches de seguridad configurados para fijar los archivos vulnerables.

7. Un servidor que comprende un medio leíble por procesador como se cita e la reivindicación 1.

8. Un medio leíble por procesador que comprende instrucciones ejecutables por procesador configurados para:

recibir una firma binaria que identifica una vulnerabilidad de seguridad en un archivo binario;
recibir un parche de seguridad configurado para fijar la vulnerabilidad de seguridad en el archivo binario; y
distribuir la firma binaria y el parche de seguridad a una pluralidad de servidores.

9. El medio leíble por procesador como se cita en la reivindicación 8, en donde el distribuir incluye: enviar una nota a cada una de una pluralidad de servidores con relación a la vulnerabilidad de seguridad y el parche dispon



vol

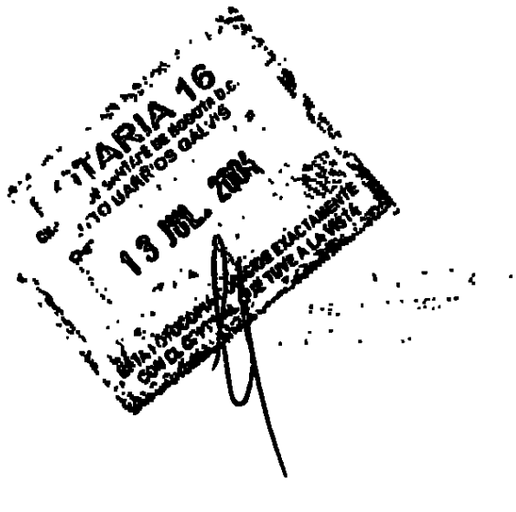
166

recibir una solicitud para enviar la firma binaria y el parche de seguridad; y
enviar la firma binaria y el parche de seguridad en respuesta a la solicitud.

10. Un servidor de distribución que comprende el medio leíble por procesador como se cita la reivindicación 8.

11. Un medio leíble por procesador que comprende instrucciones ejecutables por procesador configuradas para:

recibir una firma binaria de un servidor;
buscar la firma binaria en archivos binarios;
enviar una solicitud al servidor de un parche de seguridad si se encuentra que el archivo binario incluye la firma binaria;
recibir el parche de seguridad del servidor; y
actualizar el archivo binario con el parche de seguridad.



202

167

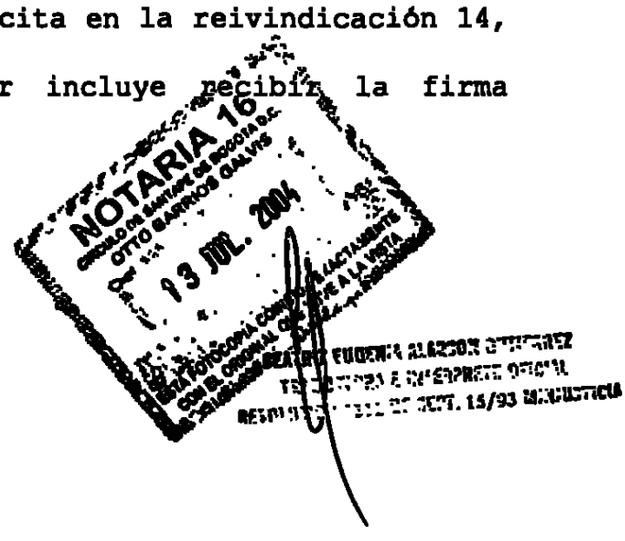
12. Un computador de cliente que comprende un medio leíble por procesador como se cita en la reivindicación 11.

13. Un método que comprende:

recibir una firma binaria;
buscar un archivo vulnerable basado en la firma binaria;
si se encuentra un archivo vulnerable, solicitar un parche de seguridad; y
fijar el archivo vulnerable con el parche de seguridad.

14. Un método como se cita en la reivindicación 13, en donde la solicitud incluye enviar una solicitud a un servidor de un parche de seguridad, el método además comprende recibir el parche de seguridad del servidor en respuesta a la solicitud.

15. Un método como se cita en la reivindicación 14, en donde el recibir incluye recibir la firma binaria del servidor.



203

168

16. El método como se cita en la reivindicación 13, en donde el fijar incluye instalar el parche de seguridad en el computador.

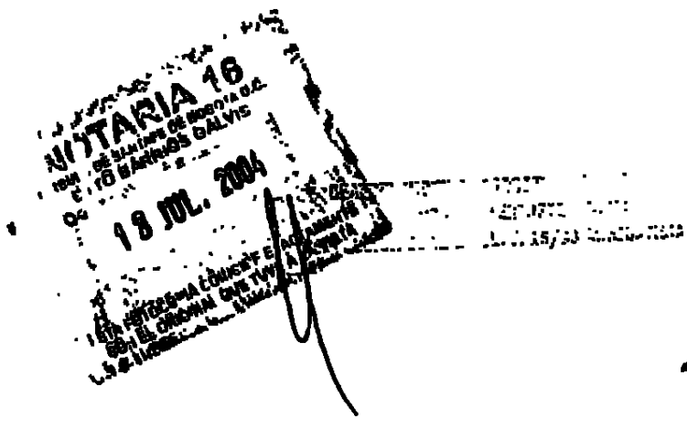
17. Un método como se cita en la reivindicación 13, en donde el buscar incluye comparar la firma binaria a la información con la información binaria en un medio de almacenamiento de un computador.

18. Un método como se cita en la reivindicación 17, en donde la información binaria se selecciona del grupo que comprende:

un sistema operativo;
un archivo de programa de aplicación; y
un archivo de datos.

19. Un método como se cita en la reivindicación 17, en donde el medio de almacenamiento se selecciona del grupo que comprende:

un disco duro;



201

169

un disco floppy magnético;

un disco óptico;

una tarjeta de memoria flash;

una memoria de solo lectura programable eléctricamente borrable; y

un almacenamiento unido a red.

20. Un método que comprende:

recibir una firma binaria y un parche de seguridad de un servidor de distribución;

buscar en un computador de cliente un archivo vulnerable asociado con la firma binaria; y

si se encuentra el archivo vulnerable, fijar el archivo vulnerable con el parche de seguridad.

21. Un método como se citó en la reivindicación 20, en donde la búsqueda incluye transferir la firma binaria al computador del cliente, el computador del cliente configurado para buscar un archivo vulnerable asociado con la firma binaria.

22. Un método como se citó en la reivindicación 21, en donde el fijar incluye:

205

170

Recibir una solicitud de un computador de cliente para transferir el parche de seguridad, el computador de cliente teniendo localizado un archivo vulnerable; y

Transferir el parche de seguridad al computador del cliente en respuesta a la solicitud.

23. Un computador que comprende:

medios para recibir una firmas binaria;

medios para buscar un archivo vulnerable basados en la firma binaria;

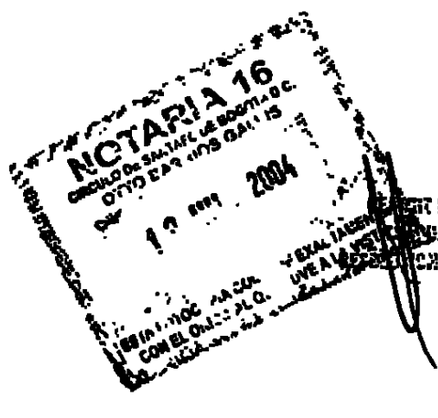
medios para solicitar un parche de seguridad si se encuentra un archivo vulnerable; y

medios para fijar el archivo vulnerable con el parche de seguridad.

24. Un servidor que comprende:

Medios para recibir una firma binaria y un parche de seguridad de un servidor de distribución;

medios para explorar un computador de cliente en busca de un archivo vulnerable asociado con la firma binaria; y



206

171

medios para fijar el archivo vulnerable con el parche de seguridad si se encuentra el archivo vulnerable.

25. Un computador que comprende:

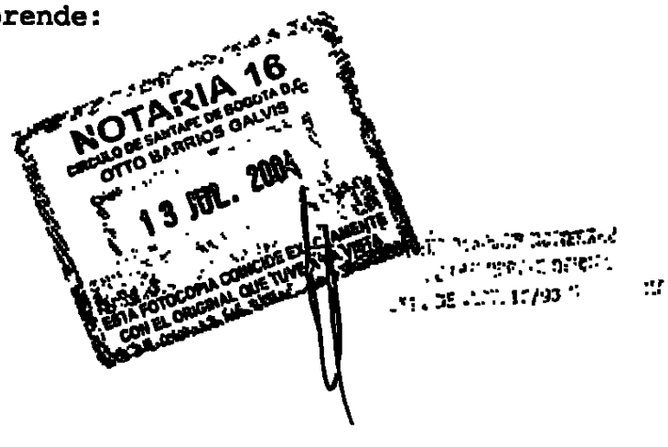
Información binaria;

Un módulo de exploración configurado para recibir una firma binaria y explorar la información binaria en busca de la firma binaria; y

Un módulo de parche configurado para solicitar un parche de seguridad e instalar el parche de seguridad si la firma binaria se encuentra n la información binaria.

26. Un computador como se cita en la reivindicación 25, que comprende además un medio de almacenamiento configurado para retener la información binaria.

27. Un computador como se cita en la reivindicación 25, en donde la información binaria se selecciona del grupo que comprende:



207

172

un sistema operativo;

un archivo de programa de aplicación; y

un archivo de datos.

28. Un computador que comprende:

archivos binarios;

una firma binaria; y

un módulo de parche de seguridad configurado para recibir una firma binaria de un servidor y para explorar los archivos binarios en la búsqueda de la firma binaria.

29. Un computador como se cita en la reivindicación 28, que comprende además:

un archivo binario que incluye la firma binaria; y

un parche de seguridad;

en donde el módulo de parche de seguridad está configurado además para solicitar el parche de seguridad del servidor luego de localizar la firma binaria dentro del archivo binario, y

178

173

aplicar el parche de seguridad al archivo binario.

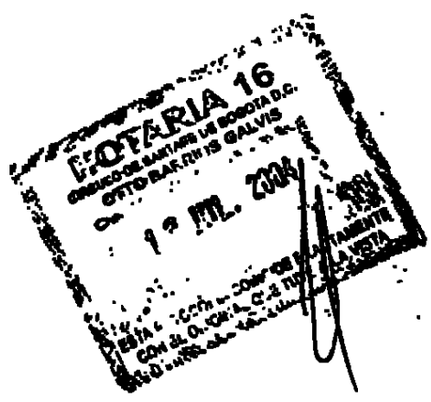
30. Un servidor de distribución que comprende:

una base de datos; y

un módulo de distribución configurado para recibir una firma binaria y un parche de seguridad, almacenar la firma binaria y el parche de seguridad en la base de datos, y distribuir la firmas binaria y el parche de seguridad a una pluralidad de servidores.

31. Un servidor de distribución como se cita en la reivindicación 30, en donde el módulo de distribución es además configurado para recibir una solicitud de un servidor de la firma binaria y el parche de seguridad y distribuir la firma binaria y el parche de seguridad al servidor en respuesta a la solicitud.

32. Un servidor que comprende:



una firma binaria asociada con una vulnerabilidad de seguridad en un archivo binario;

un parche de seguridad configurado para fijar la vulnerabilidad de seguridad en el archivo binario; y

un módulo de exploración configurado para explorar los archivos binarios en un computador de cliente en busca de la firma binaria y actualizar el archivo binario con el parche de seguridad si se encuentra la firma binaria.

33. Un servidor como se cita en la reivindicación 32, que comprende además:

una base de datos;

un módulo de exploración configurado además para recibir la firma binaria y el parche de seguridad de un servidor de distribución y almacenar la firma binaria y el parche de seguridad en la base de datos.

175

RESUMEN

Se describe el sistema y métodos para posibilitar el parcheo de vulnerabilidades de seguridad en archivos binarios. La detección y el parcheo de los archivos binaros vulnerables es automática, confiable, libre de regresión, y comprensiva a través de redes a escala ilimitadas. Estas ventajas se pueden evidenciar de varias maneras que incluyen, por ejemplo, al apalancar infraestructuras antivirus habituales que son ampliamente despegadas a través de la Internet. Descubrimiento confiable de archivos binarios vulnerables (por ejemplo en sistemas operativos, programas de aplicación, etc.) que se logra a través del uso de firmas binarias que han estado asociadas con vulnerabilidades de seguridades cubiertas. Una divergencia de parches de seguridad alejada de los paquetes de servicio convencional que suministra la posibilidad de producción de fijos libres de regresión para vulnerabilidad de seguridad e archivos binarios.

211

176

EV251222109

SOLICITUD DE PATENTE

DECLARACIÓN Y PODER DE ABOGADO

REGISTRO DE ABOGADO No. MS1-1594US Registro MS No. 304592.1

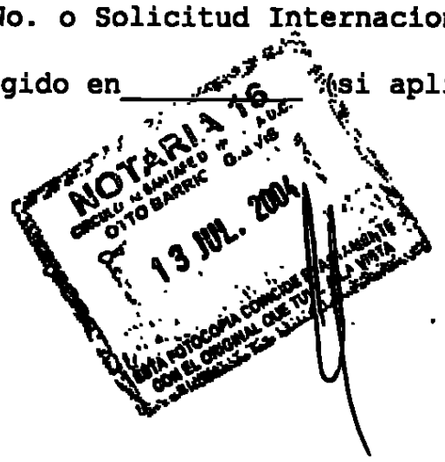
Como inventor nombrado adelante, Declaro que:

Mi residencia/dirección postal y ciudadanía son como se establece adelante junto a mi nombre:

Soy el primero, único y original inventor (si solo un nombre es listado adelante) o primeros, únicos y originales inventores (si se lista pluralidad de nombres adelante) de la materia de asunto que se reivindica y para el cual una patente se pretende sobre la invención titulada: **DETECCIÓN Y PARCHEO AUTOMÁTICO DE ARCHIVOS VULNERABLES** .

la descripción de la cual se adjunta aquí a menos que la siguiente casilla sea marcada:

( ) que fue presentada en_____como solicitud Estadounidense serie No. o Solicitud Internacional Número _____ y fue corregido en_____(si aplica).



212

177

Declaro por medio de la presente que he revisado y entendido los contenidos de la descripción identificada anteriormente, incluyendo las reivindicaciones, como se corrigió mediante cualquier corrección(es) referidas anteriormente. Reconozco la obligación de describir toda la información que es material patentable como se definió en el CFR 37 1.56.

Solicitud(es) Extranjera(s) y/o Reivindicación de Prioridad Extranjera

Por medio de la presente reivindico los beneficios de la prioridad extranjera de acuerdo con el Título 35, sección 119 del código de Estados Unidos de una solicitud(es) extranjera(s) para certificado de patente o inventor(es) listado adelante y tiene también identificado adelante cualquier solicitud extranjera para certificado de patente o inventor(es) que tiene una fecha de presentación antes de la solicitud sobre la que la prioridad es reivindicada:

213

178

PAÍS	NÚMERO DE SOLICITUD	FECHA DE PRESENTACIÓN	PRIORIDAD REIVINDICADA DE ACUERDO CON 35 U.S.C. 119
			Si: ____ No: .....
			Si: ____ No: .....

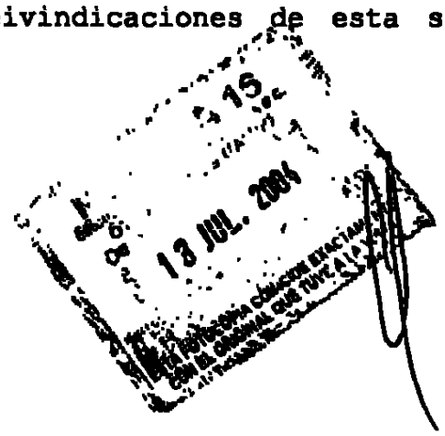
Solicitud Provisional

Por medio de la presente reivindico el beneficio de acuerdo con el título 35, sección 119(e) del código de Estados Unidos de cualquier solicitud(es) provisional(es) Estadounidenses listado adelante:

NÚMERO DE SOLICITUD	FECHA DE PRESENTACIÓN

Reivindicación de Prioridad Estadounidense

Por medio de la presente reivindico el beneficio acuerdo con el título 35, sección 120 del código de Estados Unidos de cualquier solicitud(es) Estadounidense listada adelante y, en la medida en que la materia asunto de cada una de las reivindicaciones de esta solicitud no está



214

179

descrita en la solicitud Estadounidense en la forma dada en el primer párrafo del título 35, del código de Estados Unidos sección 112,1 Reconozco la obligación de describir la información material como se definió en el título 37, Código de Regulaciones Federal, Sección 1.56(a) que ocurrió entre la fecha de presentación de la solicitud anterior y la nacional o fecha de presentación internacional PCT de esta solicitud:

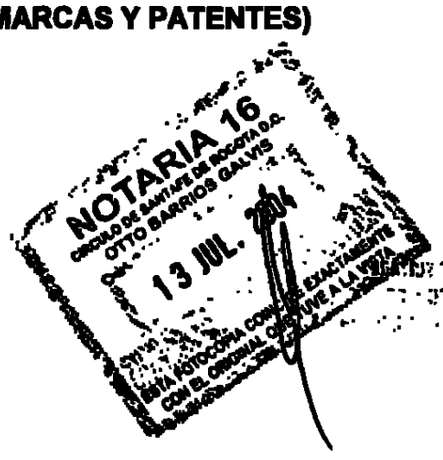
NÚMERO DE SOLICITUD	FECHA DE PRESENTACIÓN	ESTADO (patentado/pendiente/abandonado)

PODER DE ABOGADO

Como inventor, por medio de la presente nombro al siguiente(s) abogado(s) y/o agente(s) listado(s) y aquellos asociados con

cliente No. 22801

(aparece sello - código de barras
22801
OFICINA DE MARCAS Y PATENTES)



215

180

para procesar esta solicitud y transar todos los negocios en la Oficina de Marcas y Patentes relacionado con este.

Daniel D. Crouse, Reg. 32022

David Bartley Eppenauer, Reg No. 35499

Martin L. Shively, Reg. No. 33533

Ronald O. Zink, Reg. No. 35744

Patricia E. Bornes, Reg. No. 37038

Michael W. Bocianowski, Reg. No. 28692

Danielle J. Holmes, Reg. No. 39720

Enviar Correspondencia a:

Nathan R. Rieth

Lee & Hayes PLLC

421 W. Riverside Ave., Suite 500

Spokane, WA 99201

(aparece sello - código de barras
22801
OFICINA DE MARCAS Y PATENTES)

LLAMADAS A TELÉFONO DIRECTO A:

Nathan R. Rieth

Número telefónico (309)324-9256

NOTARIA 16
CIRCULO DE BOGOTA D.C.
OTTO BARRIOS GALVIS
13 JUL. 2004

216

181

DECLARACIÓN Y PODER DE ABOGADO

REGISTRO DE ABOGADO No. MS1-1594US Registro MS No. 304592.1

Por medio de la presente declaro que todos las afirmaciones hechas aquí de mi propio conocimiento son verdad y que todas las afirmaciones hechas sobre información y opinión son creídas para ser verdad; y adicionalmente que estas afirmaciones fueron hechas con el conocimiento de que las informaciones falsas intencionadas y similares pueden ser hechas punibles por multa o encarcelamiento, o ambos, de acuerdo con la Sección 1001 del título 18 del Código de Estados Unidos y que tales afirmaciones falsas intencionadas por consiguiente pueden arriesgar la validez de la solicitud o cualquier patente publicada.

Nombre completo del Inventor:	Oleg Ivanov
Ciudadanía:	Ucrania/Estados Unidos
Residencia:	Issaquah, WA
Dirección Oficina Postal:	25923 SE 37th Way, Issaquah, WA 98029

(aparece firma ilegible)
Firma del Inventor

NOTARIA 16
ESCUELA DE SANTAS DE PUERTO R.C.
OTTO BARRIOS GALVIS
18 JUL. 2004

DE SEPT. 15/03 MINISTERIO

118

182

7/16/2003

Fecha

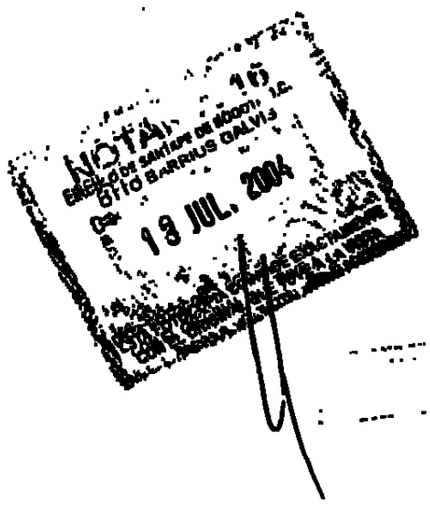
Nombre completo del Inventor:	Sergei Ivanov
Ciudadanía:	Ucrania/Estados Unidos
Residencia:	Issaquah, WA
Dirección Oficina Postal:	25923 SE 37th Way, Issaquah, WA 98029

(aparece firma ilegible)

Firma del Inventor

7/16/2003

Fecha



218

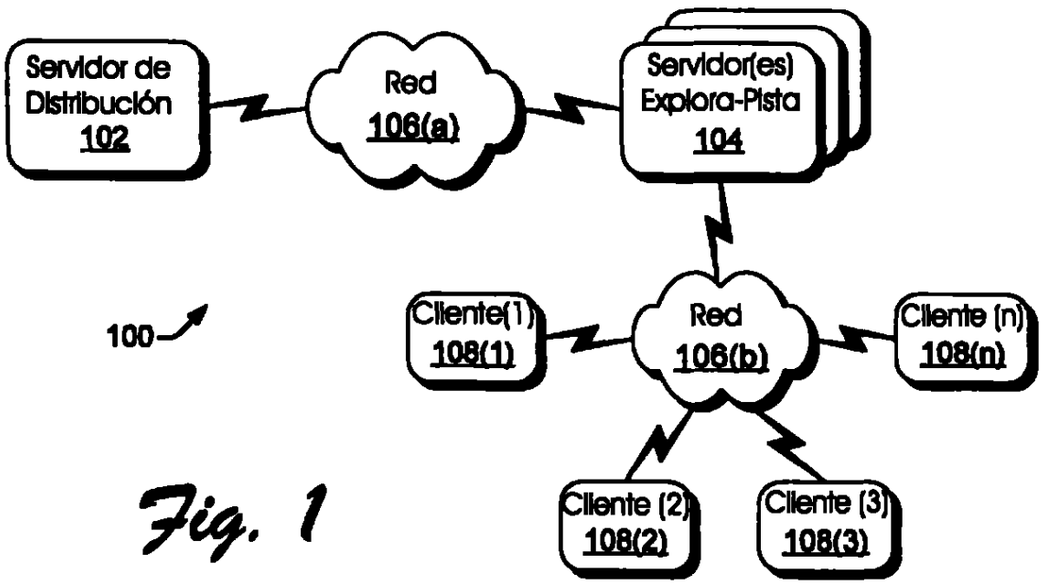


Fig. 1

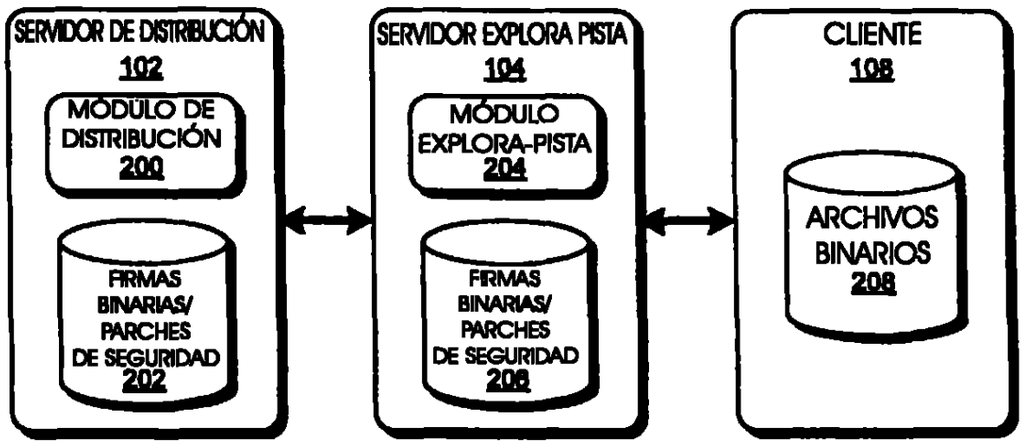


Fig. 2

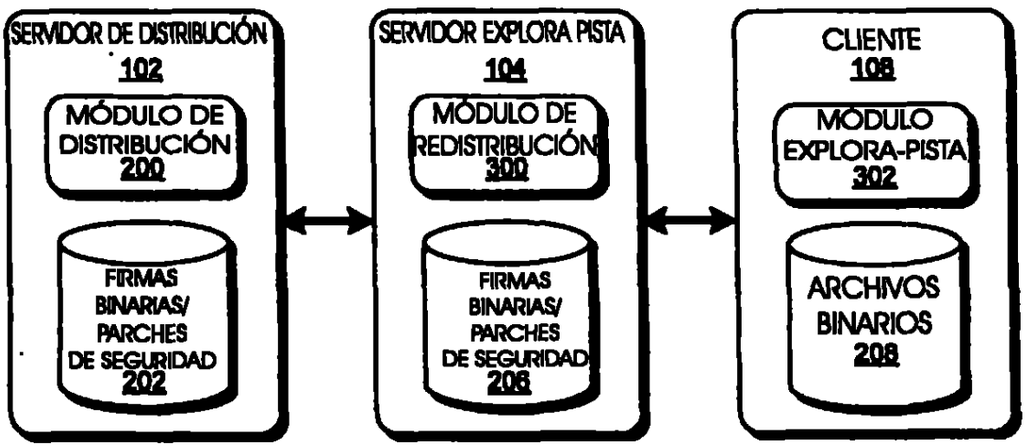


Fig. 3

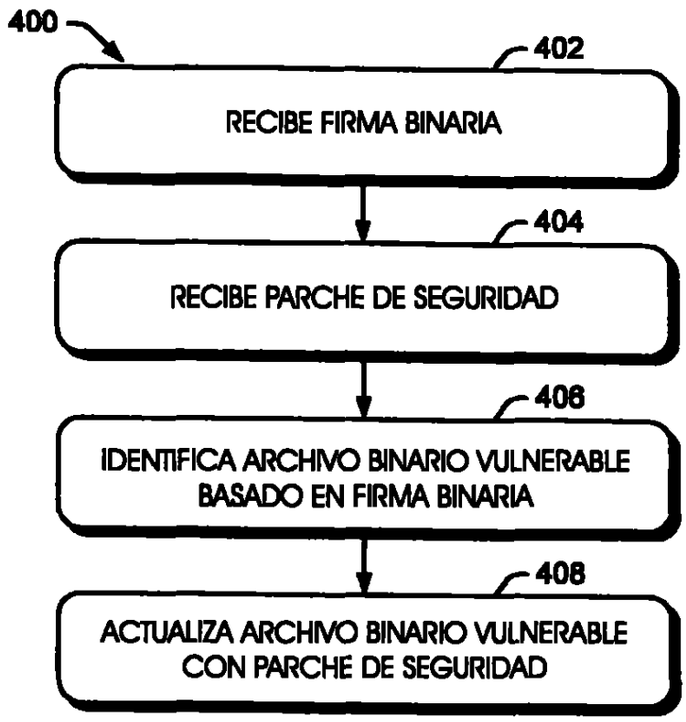

400
402
RECIBE FIRMA BINARIA
404
RECIBE PARCHE DE SEGURIDAD
406
IDENTIFICA ARCHIVO BINARIO VULNERABLE BASADO EN FIRMA BINARIA
408
ACTUALIZA ARCHIVO BINARIO VULNERABLE CON PARCHE DE SEGURIDAD

Fig. 4

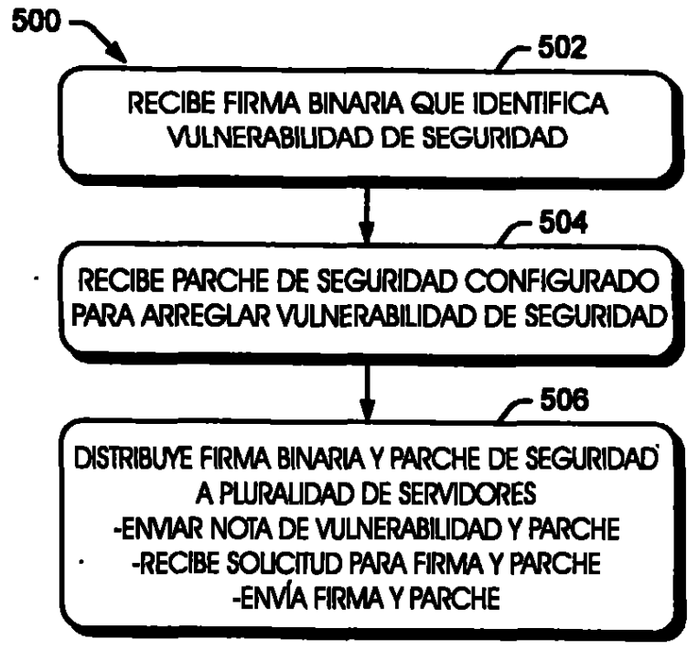
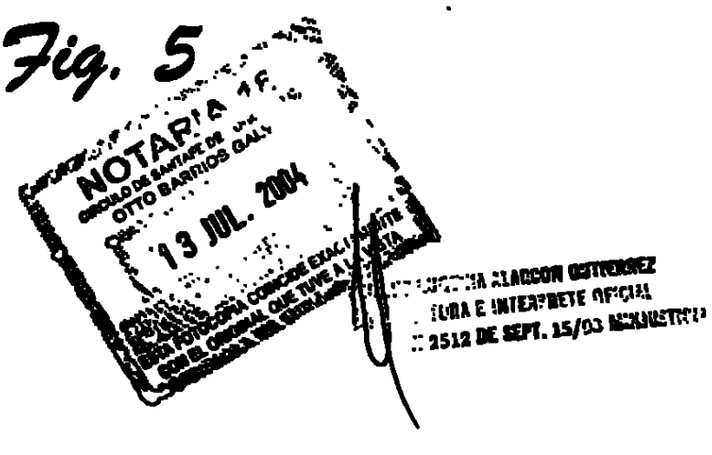

500
502
RECIBE FIRMA BINARIA QUE IDENTIFICA VULNERABILIDAD DE SEGURIDAD
504
RECIBE PARCHE DE SEGURIDAD CONFIGURADO PARA ARREGLAR VULNERABILIDAD DE SEGURIDAD
506
DISTRIBUYE FIRMA BINARIA Y PARCHE DE SEGURIDAD A PLURALIDAD DE SERVIDORES
-ENVIAR NOTA DE VULNERABILIDAD Y PARCHE
-RECIBE SOLICITUD PARA FIRMA Y PARCHE
-ENVÍA FIRMA Y PARCHE

Fig. 5


NOTARIA
OTTO BARRIOS GALL
13 JUL. 2004

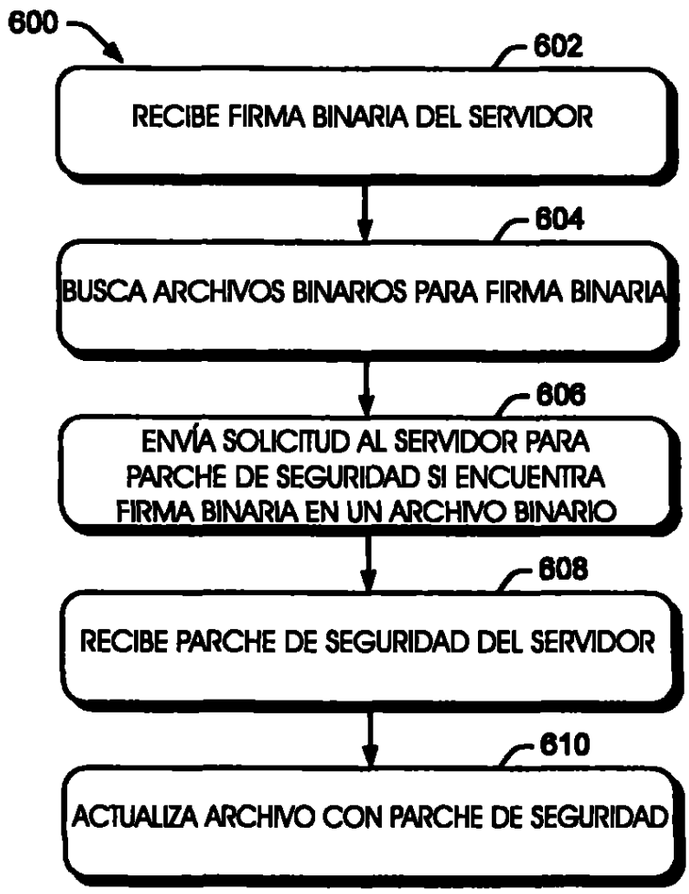


Fig. 6

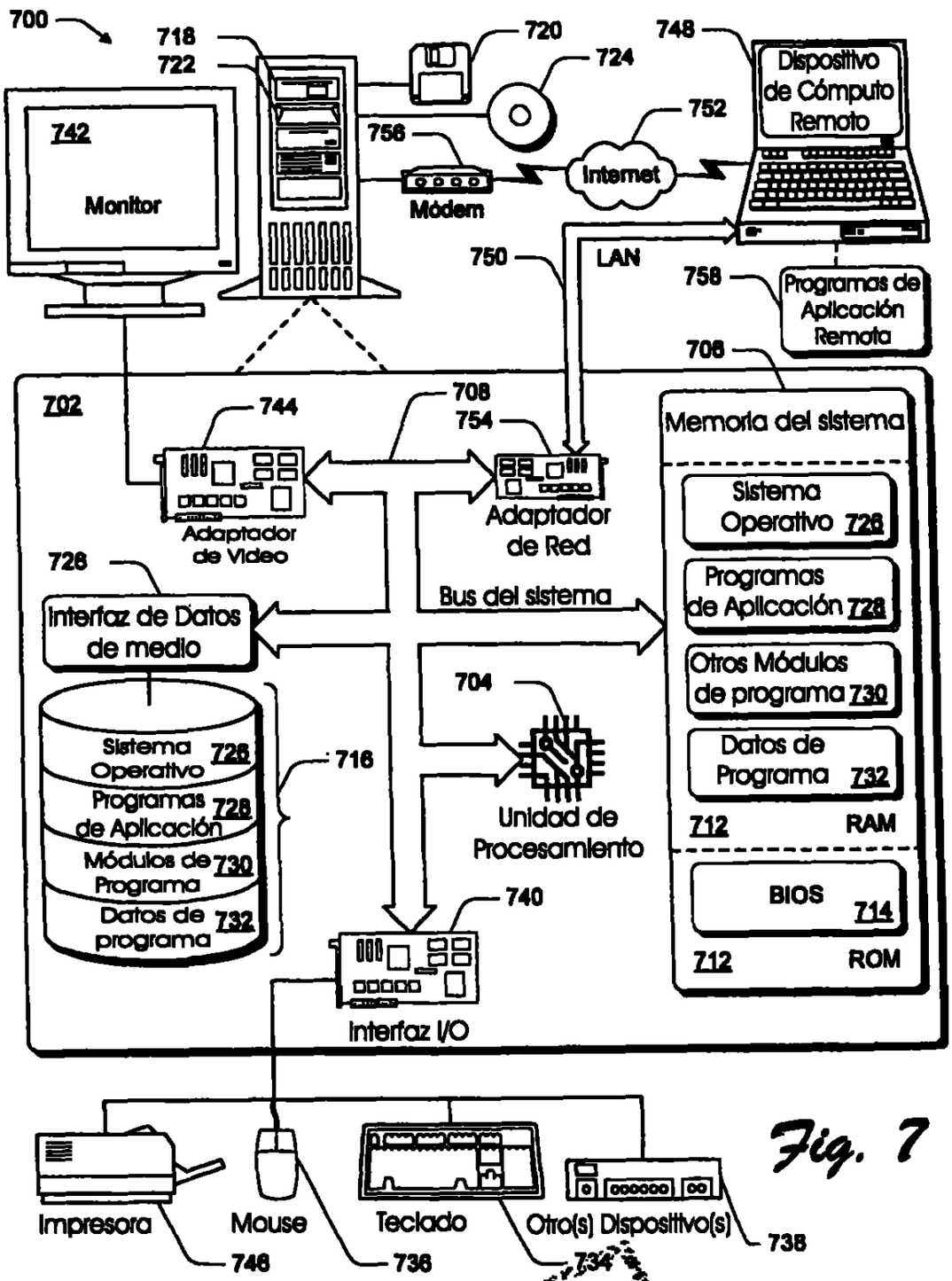


Fig. 7

UNITED STATES OF AMERICA

The State of Washington

Secretary of State

APOSTILLE

(Convention de la Haye du 5 Octobre 1961)

1. Country: **United States of America**

2. This public document has been signed by: LEANN M. SASSMAN

3. acting in the capacity of: Notary Public, state of Washington

4. bears the seal/stamp of: LEANN M. SASSMAN

CERTIFIED

5. at: Olympia, Washington

6. the: 16 day of June, 2004

7. by: Sam Reed, Secretary of State

8. No: 200419746

9. Seal/Stamp:

10. Signature:

Given under my hand and the Seal of the State of Washington at Olympia, the State Capital

Sam Reed

Sam Reed, Secretary of State

State of Washington)
ss:
County of Spokane)

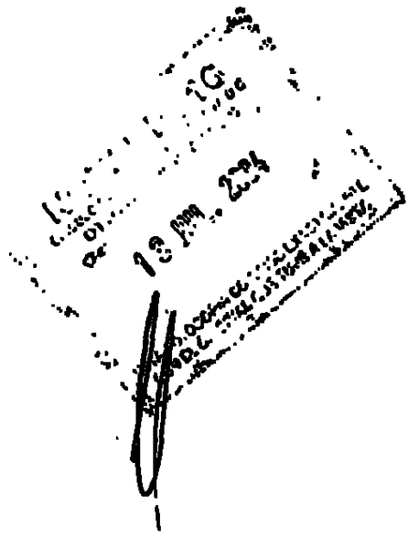
I certify that this is a true and correct copy of an Assignment document that grants Microsoft Corporation the right to file foreign applications directly in the name of Microsoft Corporation.

Dated 7 June 2004

LEANN M. SASSMAN
Commission Expires
NOTARY
PUBLIC
MAY 25, 2005
STATE OF WASHINGTON

Signature of Notary Public LeAnn M Sassman

My appointment expires 5/25/05



190

EV251222109

Form PTO-1595 (Rev. 03/01) 7-1603
OMB No. 0651-0027 (exp. 5/31/2002)
Tab settings ⇨ ⇨ ⇨

07-29-2003
102510427

U.S. DEPARTMENT OF COMMERCE
U.S. Patent and Trademark Office

To the Honorable Commissioner of Patents and Trademarks: Please record the attached original documents or copy thereof.

1. Name of conveying party(ies):
Oleg Ivanov
Sergei Ivanov
10/621148
Additional name(s) of conveying party(ies) attached? ☐ Yes ☒ No

2. Name and address of receiving party(ies)
Name: Microsoft Corporation
Internal Address:
Street Address: One Microsoft Way
City: Redmond State: WA Zip: 98052
Additional name(s) & address(es) attached? ☐ Yes ☒ No

17314 U.S. PTO 10/621148 07/18/03

3. Nature of conveyance:
☒ Assignment ☐ Merger
☐ Security Agreement ☐ Change of Name
☐ Other

Execution Date: 07/18/2003

4. Application number(s) or patent number(s):
If this document is being filed together with a new application, the execution date of the application is: 07/18/2003
A. Patent Application No.(s)
B. Patent No.(s)
Additional numbers attached? ☐ Yes ☒ No

5. Name and address of party to whom correspondence concerning document should be mailed:
Name: Nathan R. Rieth, Reg. No. 44,302
Internal Address: Lee & Hayes, PLLC
07/21/2003 DTUN11 00000035 12076? 10621148
01 FC:8021 40.00 DA
Street Address: 421 West Riverside Avenue
Suite 500
City: Spokane State: WA Zip: 99201

6. Total number of applications and patents involved: 1

7. Total fee (37 CFR 3.41).....$ 40.00
☐ Enclosed
☒ Authorized to be charged to deposit account

8. Deposit account number: 12-0769
(Attach duplicate copy of this page if paying by deposit account)

DO NOT USE THIS SPACE

9. Statement and signature.
To the best of my knowledge and belief, the foregoing information is true and correct and any attached copy is a true copy of the original document.

Nathan R. Rieth, Reg. No. 44,302 — Name of Person Signing
Nathan R. Rieth — Signature
7/16/03 — Date

Total number of pages including cover sheet, attachments, and documents: 3

Mail documents to be recorded with required cover sheet information to:
Mail Stop Assignment Recordation Services
Alexandria, VA 22313-1450

83

EV251222109

[redacted]

Serial Number .....
Filing Date .....
Inventorship .....Ivanov et al.
Applicant .....Microsoft Corporation
Attorney's Docket No. MS1-1594US
MSDocket No. 304592.1
Title: Automatic Detection And Patching Of Vulnerable Files

PATENT ASSIGNMENT

PARTIES TO THE ASSIGNMENT

Assignor(s):

Oleg Ivanov
25923 SE 37th Way
Issaquah, WA 98029

Sergei Ivanov
25923 SE 37th Way
Issaquah, WA 98029

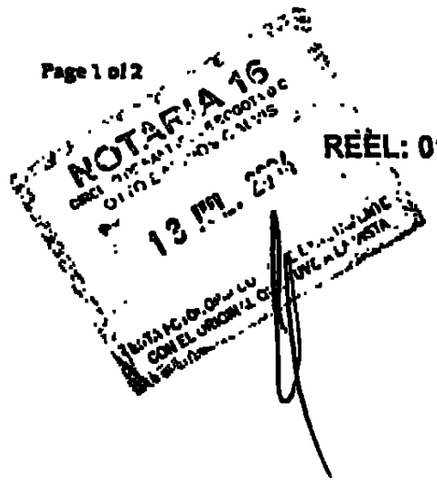
Assignee:

Microsoft Corporation
Corporation in the State of Washington
One Microsoft Way
Redmond, WA 98052

AGREEMENT

WHEREAS, ASSIGNOR(S) (listed above) are inventor(s) of an invention entitled "Automatic Detection And Patching Of Vulnerable Files," as described and claimed in the specification forming part of an application for United States letters patent referenced above;

WHEREAS, Microsoft Corporation (hereinafter referred to as ASSIGNEE), a corporation of the State of Washington having a place of business at One Microsoft Way, Redmond, WA 98052, is desirous of acquiring the entire right, title and interest in and to the invention and in and to any letters patent that may be granted therefor in the United States and in any and all foreign countries;



NOW, THEREFORE, in exchange for good and valuable consideration, the receipt of which is hereby acknowledged, ASSIGNOR(S) hereby sell, assign and transfer unto ASSIGNEE, the entire right, title and interest in and to said invention, said application and any and all letters patent which may be granted for said invention in the United States of America and its territorial possessions and in any and all foreign countries, and in any and all divisions, reissues and continuations thereof, including the right to file foreign applications directly in the name of ASSIGNEE and to claim priority rights deriving from said United States application to which said foreign applications are entitled by virtue of international convention, treaty or otherwise, said invention, application and all letters patent on said invention to be held and enjoyed by ASSIGNEE and its successors and assigns for their use and benefit and of their successors and assigns as fully and entirely as the same would have been held and enjoyed by ASSIGNOR(S) had this assignment, transfer and sale not been made. ASSIGNOR(S) hereby authorize and request the Commissioner of Patents and Trademarks to issue all letters patent on said invention to ASSIGNEE. ASSIGNOR(S) agree to execute all instruments and documents required for the making and prosecution of applications for United States and foreign letters patent on said invention, for litigation regarding said letters patent, or for the purpose of protecting title to said invention or letters patent therefor.

7/16/2003
Date

Oleg Ivanov

7/16/2003
Date

Sergei Ivanov

193

La presente es una traducción Oficial del Inglés al Español de una Apostilla dada en los Estados Unidos de América

ESTADOS UNIDOS DE AMÉRICA

El Estado de Washington

Ministro de Asuntos Exteriores

APOSTILLA

(Convención de la Haya de Octubre 5 de 1961)

1. País: **Estados Unidos de América**
2. Este documento público ha sido firmado por:

   LEANN M. SASSMAN
3. actuando en calidad de:

   Notario Público, Estado de Washington
4. lleva el sello de:

   LEANN M. SASSMAN

CERTIFICADO

5. en: Olimpia, Washington
6. el: día 16 de Junio, 2004
7. por: Sam Reed, Ministro de Asuntos Exteriores

16 JUN 2004

[illegible] ALARCON GUTIERREZ
[illegible] E INTERPRETE OFICIAL
[illegible] DE SEPT. 15/93 WASHINGTON

194

8. No.: 200419746

9. Sello:

10. Firma:

(Aparece sello:

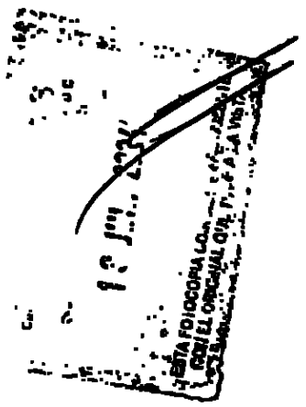
EL SELLO DEL ESTADO

DE WASHINGTON 1889)

Dado bajo mi firma y sello del Estado de Washington en Olimpia, la Capital del Estado

(Aparece Firma ilegible)

Sam Reed, Ministro de Asuntos Exteriores



MINISTERIO DE
RELACIONES EXTERIORES

239335

CUYA FIRMA APARECE EN EL
DOCUMENTO DESEMPEÑA
LAS FUNCIONES INDICADAS

NO SE ASUME
RESPONSABILIDAD DEL TEXTO

2004 JUL 1: 00

JEFE DE LEGALIZACIONES
BOGOTA D.C.

195

Estado de Washington)

ss:

Condado de Spokane) .

Certifico que esta es una copia fiel y exacta de un documento de Traspaso que otorga a Microsoft Corporation el derecho de registrar solicitudes extranjeras directamente a nombre de Microsoft Corporation

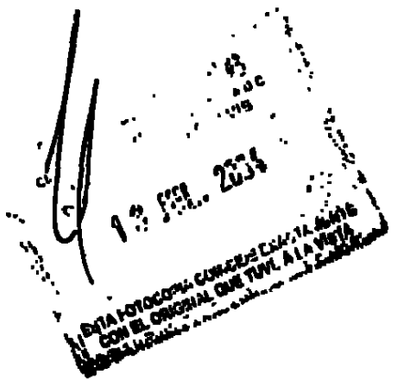
(Aparece sello:
LEANN M. SASSMAN
Estado de WASHINGTON
El Nombramiento Expira en
Mayo 25, 2005
NOTARIO PÚBLICO)

Fechado 7 de Junio 2004

Firma del

Notario Público Aparece firma

Mi nombramiento expira en 5/25/05



BEATRIZ AUSTRIA ALARCON GUTIERREZ
TRADUCTORA E INTERPRETE OFICIAL
RESOLUCION 2512 DE SEPT. 15/93 MINJUSTICIA

88

196

EV251222109

Forma 'PTO-1595
(Rev. 03/1) 7-1603
OMB No. 0651-0027 (exp.5/31/2002)

DEPARTAMENTO DE COMERCIO

DE LOS ESTADOS UNIDOS

Oficia de Marcas y Patentes

de los Estados Unidos

Al honorable Comisionado de Marcas y Patentes: Por favor registre los documentos originales adjuntos o copia de estos.

(Aparecen Números y Código de Barras:

07-29-2003

Código de Barras

102510427

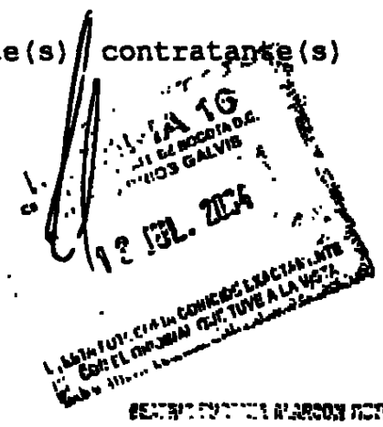
)

1: Nombre de la(s) parte(s) cedente(s)

Oleg Ivanov

Sergei Ivanov 10/621148

Nombre(s) Adicional(es) de parte(s) contratante(s) adjunta(s)? ☐ Si ☑ No



87

197

2. Nombre y dirección de la(s) parte(s) cesionaria(s):

Nombre	Microsoft Corporation
Dirección Interna	_____
Dirección	One Microfot Way
Ciudad	Redmond
Estado/Prov.:	WA
Zip	98052

Nombre(s) adicional(es) & dirección(es) adjunta(s)

☐ Si ☑ No

3. Naturaleza de la cesión:

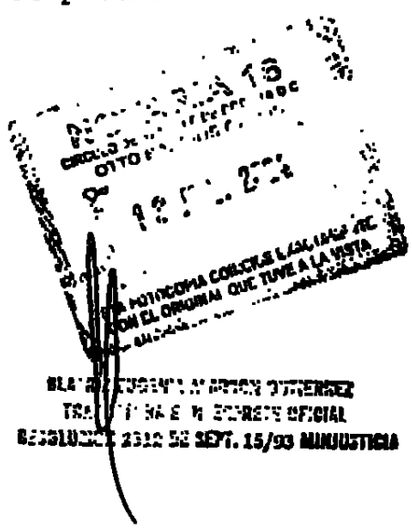
☑ Traspaso ☐ fusión

☐ Acuerdo de seguridad

☐ Cambio de nombre ☐ Otro _____

Fecha de Formalización: 07/16/2003

4. Número(s) de solicitud o número(s) de patente:



90

198

Si este documento está siendo presentado junto con una nueva solicitud, la(s) fecha(s) de formalización de la solicitud es: 7/16/2003

A. Solicitud de Patente Nos.

B. Patente No. (s)

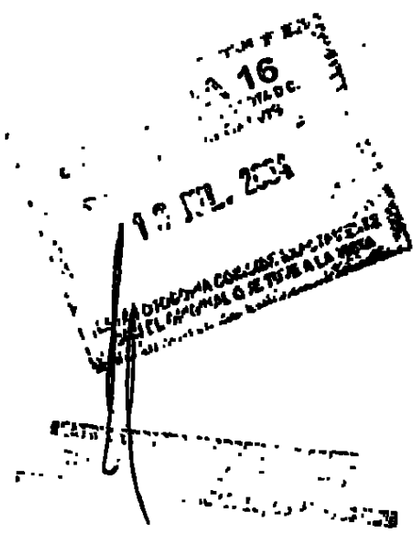
Números adicionales ☐ Si ☑ No

5. Nombre y dirección de la parte a quien corresponda los documentos concernientes que deban ser enviados por correo:

Nombre: Nathan R. Rieth, Reg. No. 44,302
Dirección Interna: Lee & Hayes, PLLC

07/28/2003 STOMIL 00000236 120769 10621148
01 FC: (aparecen números ilegibles) 40.00 DA

Dirección: 421 West Riverside Avenue
Suite 500
Ciudad: Spokane
Estado/Provincia: WA
ZIP: 99201



199

6. Número total de solicitudes y patentes involucradas:

[1]

7. Tarifa total (37 CFR 3.41):. $40.00

☐ Incluido

☑ Autorizado a ser cargado a cuenta de depósito

8. Cuenta de depósito número:

12-0769

(adjunte copia en duplicado de esta página si hay pago mediante cuenta de depósito)

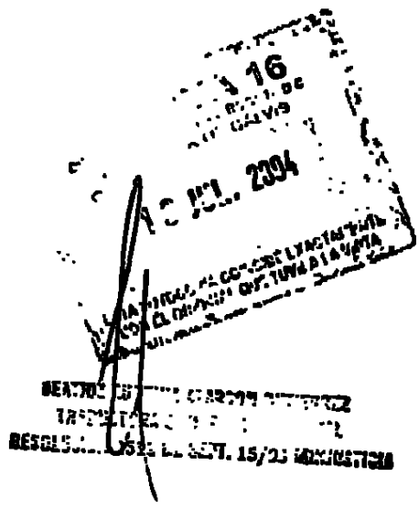
NO USE ESTE ESPACIO

9. Declaración y firma.

Según mi leal saber y entender, la información precedente es fiel y exacta y cualquier copia adjunta es una copia fiel del documento original.

Nathan R. Rieth, Reg. No. 44,302

Nombre de la persona firmante



200

(aparece firma ilegible)

Firma

7/16/03

Fecha

Número total de páginas presentadas incluyendo portada, adjuntos y Documentos: [3]

Documentos a ser registrados con la información de portada requerida a:
Servicios de Registro de Traspasos Parada de Correo
Alexandria, VA 22313 - 1450

PATENTE

ROLLO: 014308 CUADRO: 0208

Registro de Abogado No. MS1-1594US
Registro MS No. 304592.1
EV251222109

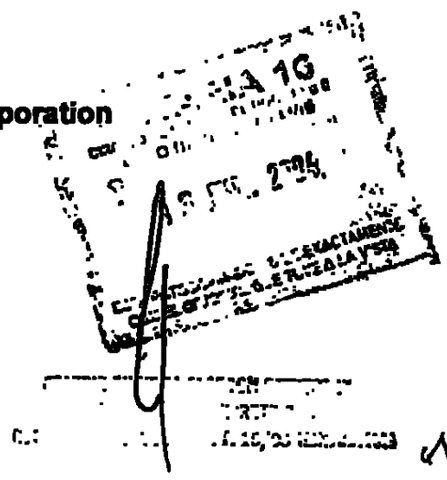
EN LA OFICINA DE MARCAS Y PATENTES DE LOS ESTADOS UNIDOS

Número de Serie:

Fecha de Presentación:

Inventores **Ivanov et al.**

Solicitante **Microsoft Corporation**



201

Registro de Abogado No. **MS1-1594US**

Registro MS No. **304592.1**

Título: **DETECCIÓN Y PARCHEO AUTOMÁTICO DE ARCHIVOS VULNERABLES**

TRASPASO DE PATENTE

PARTES EN EL TRASPASO

Cedente(s):

Oleg Ivanov

25923 SE 37th Way

Issaquah, WA 98029

Sergei Ivanov

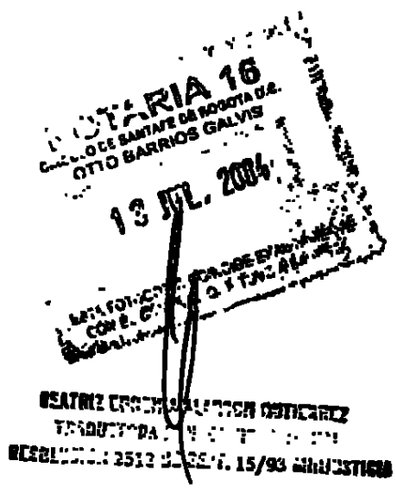
25923 SE 37th Way

Issaquah, WA 98029

Cesionario:

Microsoft Corporation

Corporation in the State of Washington



94

202

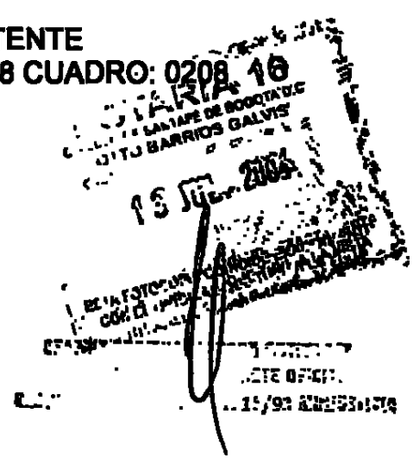
One Microsoft Way

Redmond, WA 98052

ACUERDO

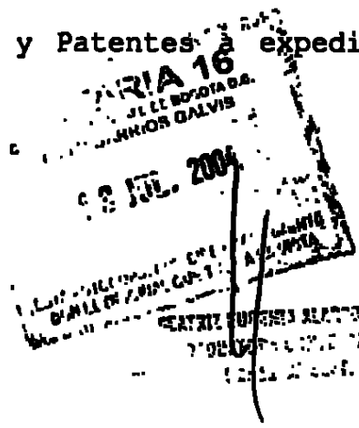
CONSIDERANDO, QUE EL(LOS) CEDENTE(S) (listados anteriormente) es(son) inventor(es) de la invención titulada **"DETECCIÓN AUTOMÁTICA Y PARCHEO DE ARCHIVOS VULNERABLES"**, como se describió y reivindicó en la especificación que forma parte de la solicitud para título de patente de Estados Unidos referenciadas anteriormente;

CONSIDERANDO, que Microsoft Corporation (llamado de aquí en adelante como CESIONARIO), una corporación del Estado de Washington que tiene lugar de negocio en One Microsoft Way, Redmond, WA 98052, deseosa de adquirir los derechos completos, título e interés en y a la invención y en cualquier título de patente que por lo tanto pueda ser otorgada en los Estados Unidos y en cualquiera y todos los países extranjeros;



203

POR LO TANTO, AHORA, en intercambio de buena y valiosa consideración, el recibo del cual se reconoce aquí, los CEDENTES por medio del presente venden, traspasan y transfieren al CESIONARIO, los derechos completos, título, beneficio en y a dicha invención, dicha solicitud y cualquiera y todos los cartas de patente que puedan ser otorgadas a dicha invención en los Estados Unidos de América, y sus posesiones territoriales y en cualquiera de los países extranjeros, y en cualquiera y todos las reotorgamientos y continuaciones de la misma, incluyendo el derecho a presentar solicitudes en el extranjero directamente a nombre del CESIONARIO y a reivindicar los derechos de propiedad derivados de dicha solicitud Estadounidense al cual dichas solicitudes extranjeras tienen derecho por virtud de la convención, tratado internacional, o de otra manera, dicha invención, solicitud y todas las cartas de patente sobre dicha invención a ser poseídos y disfrutados por los CESIONARIOS y sus sucesores y cede en su uso y beneficio y el de sus sucesores y cede tan completa y enteramente como si la misma hubiera sido poseída y disfrutada por el cedente si este traspaso o venta no hubiera sido hecho. Por medio de la presente los CEDENTES autorizan y solicitan al Comisionado de Marcas y Patentes a expedir



96

204

todas las cartas de patente sobre esta invención al CESIONARIO. Los CEDENTES están de acuerdo para firmar todos los instrumentos y documentos requeridos para efectuar y tramitar las solicitudes para los títulos de patente en los Estados Unidos y en el exterior, para litigar con respecto a dichos títulos de patente, o con el propósito de proteger el título a dicha invención o título de patente.

7/16/2003 (aparece firma ilegible)

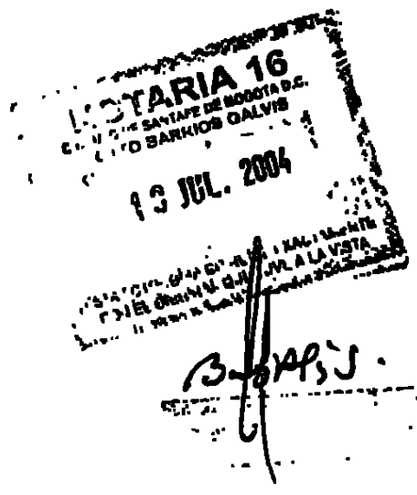
Fecha Oleg Ivanov

7/16/2003 (aparece firma ilegible)

Fecha Sergei Ivanov

REGISTRADO: 07/16/2003

PATENTE
ROLLO: 014308 CUADRO: 0208



9 8

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Automatic Detection And Patching Of Vulnerable Files

Inventors:

Oleg Ivanov

Sergei Ivanov

ATTORNEY'S DOCKET NO. MS1-1594US

TECHNICAL FIELD

The present disclosure generally relates to patching files, and more particularly, to an automatic, comprehensive, reliable and regression-free way of providing security patches for vulnerable binary program files in distributed, heterogeneous computing environments.

BACKGROUND

Software development is an ongoing process whereby a software product initially released to the public can be continually updated through revisions from a software developer/vendor. Software revisions are typically disbursed from a software vendor in what are called "service packs" that can be downloaded or ordered from a vendor for installation on a user's computer. Service packs typically contain program fixes (e.g., for an operating system, application program, etc.) that repair problems (i.e., "bugs") discovered in the program code after the initial release of the product or after the last service pack release.

In addition to containing fixes for program bugs, service packs can also contain security patches developed specifically to repair vulnerabilities found in program files. Program vulnerabilities discovered after a software product is released can pose significant security threat of attack from hackers and viruses on a world-wide basis. Therefore, once a vulnerability is discovered, the prompt and wide-spread distribution and installation of security patches to computers having vulnerable software is of paramount importance. Theoretically, the use of service packs to achieve such prompt and wide-spread distribution of security patches could be effective. For example, when a software vendor discovers a vulnerability and then develops a security patch, the patch can be posted in the latest service

pack on a vendor Web site for users to immediately download and install. This could thwart most hackers and viruses that are intent on exploiting the discovered vulnerability. However, system administrators and other software product users currently face several drawbacks and/or difficulties related to accessing and installing security patches. These difficulties typically result in a significantly lower distribution of such patches than is intended by the vendor who develops the patch. The result is that vulnerabilities on many computers world-wide are left un-patched, exposing such computers to significant risk.

One difficulty with accessing and installing security patches is that current methods for detecting whether a computer is running software with a known vulnerability require the active use and involvement of the computer. For example, currently available methods can determine whether particular versions of software products on a computer are in need of being updated (e.g., with a security patch). However, only those software products actively running on the computer are included in this determination. Secondary operating systems and applications that are not actively running on a computer are not considered, and therefore may have a security vulnerability that goes un-noticed and un-fixed. For those products actively running on a computer, a user can review a list of available updates and select updates for installation. Some updates may be critical updates designed to protect a computer from known security vulnerabilities. Various updates require a user to restart the computer before the installation is complete. In addition, a user must actively select the updates and install them. For these and other reasons, current methods for accessing and installing security patches are less than effective.

Another difficulty in accessing and installing security patches is that of knowing whether or not a security patch is needed on a computer. It is sometimes difficult for users to know if their computers are running software that is vulnerable. Furthermore, current methods for detecting whether a computer is running software with a known vulnerability may not be able to detect certain configurations of a software product known to be vulnerable. For example, shared versions of some software products can be distributed as part of other products. Thus, although a shared version of a product may contain the same vulnerability as the full version of the product, the shared version may not be recognized as a product that needs a security patch update. Thus, shared versions of software products that are known to have security vulnerabilities often go un-fixed.

Other problems with accessing and installing security patches relate to the conventional "service pack" method by which such patches are delivered. Downloading and installing services packs is a time intensive and manual process that many system administrators simply do not have time to perform. Therefore, even when administrators intend to install security patches, the time between the release of a security patch and its installation on a given system can be weeks, months, or years. Thus, the risk of attack through a security vulnerability may not be alleviated in such systems until long after the software vendor has issued a security patch.

Furthermore, system administrators often choose not to download and install service packs containing security patches, even though they understand the relevant security risks. The reason for this is that the installation of a service pack itself brings the risk of system regressions that can introduce unwanted changes in system behavior. Administrators often devote significant time and effort toward

debugging a system so that it functions as desired. As mentioned above, however, service packs represent an evolution of a previous version of a software product that includes the most recent updates to a product's code base (i.e., the scope of changes is not restricted to security patches only). In addition to introducing new and intended behaviors into a system, recent code updates in a service pack may introduce unknown bugs into a system that can cause the system to behave unexpectedly, which, in turn, can create significant problems for a system administrator. Thus, systems frequently are not updated with important security patches intended to fix vulnerable program files, because administrators do not want to risk regressions.

Accordingly, a need exists for a way to implement patching of security vulnerabilities in program files in an automatic, comprehensive, reliable and regression-free manner.

SUMMARY

Automatic, comprehensive, reliable and regression-free security patching of binary program files is described herein.

In accordance with one implementation, a binary signature of a vulnerability and a security patch are received. A vulnerable binary file is identified on a computer based on the binary signature of a vulnerability. The vulnerable binary file on the computer is updated with the security patch.

In accordance with another implementation, a binary signature is received that identifies a security vulnerability in a binary file. A security patch configured to fix the security vulnerability is also received. The binary signature and the security patch are distributed to a plurality of servers.

In accordance with another implementation, a binary signature is received from a server and used to search binary files. A request for a security patch is sent to the server if the binary signature is found in a binary file. The binary file is then updated with the security patch.

BRIEF DESCRIPTION OF THE DRAWINGS

The same reference numerals are used throughout the drawings to reference like components and features.

Fig. 1 illustrates an exemplary network environment suitable for implementing automatic detection and patching of security vulnerabilities in binary files.

Fig. 2 illustrates an exemplary embodiment of a distribution server, a scan-patch server, and a client computer suitable for implementing automatic detection and patching of security vulnerabilities in binary files.

Fig. 3 illustrates another exemplary embodiment of a distribution server, a scan-patch server, and a client computer suitable for implementing automatic detection and patching of security vulnerabilities in binary files.

Figs. 4 - 6 illustrate block diagrams of exemplary methods for implementing automatic detection and patching of security vulnerabilities in binary files.

Fig. 7 illustrates an exemplary computing environment suitable for implementing a distribution server, a scan-patch server, and a client computer.

DETAILED DESCRIPTION

Overview

The following discussion is directed to systems and methods that enable patching of security vulnerabilities in binary files. The detection and patching of vulnerable binary files is automatic, reliable, regression free, and comprehensive across networks on an unlimited scale. These advantages can be realized in various ways including, for example, by leveraging current anti-virus infrastructure that is widely deployed across the Internet. A divergence of security patches away from conventional service packs provides for the possibility of production of regression-free fixes for security vulnerabilities in binary files.

Reliable discovery of vulnerable binary files (e.g., in operating systems, application programs, etc.) is achieved through the use of binary signatures that have been associated with security vulnerabilities. Binary signatures associated with security vulnerabilities in binary files, along with security patches developed to fix such security vulnerabilities, are uploaded to a central distribution server. The distribution server is configured to distribute the binary signatures and security patches on a wide-scale basis across various networks such as the Internet. Use of a central distribution server to update network servers (e.g., across the Internet) provides comprehensive and automatic patch coverage on an unlimited scale. Network servers receiving such updates can scan client computers within subordinate networks to locate vulnerable files according to binary signatures, and then update those computers found to have security vulnerable files using corresponding security patches that will fix the vulnerable files. Network servers can also communicate with client computers to transfer binary signatures and security patches to the computers so that the scanning and updating can be

performed by the computers themselves. Multiple nested levels of subordinate networks may also exist.

Exemplary Environment

Fig. 1 illustrates an exemplary network environment 100 suitable for implementing automatic detection and patching of security vulnerabilities in binary files. In the exemplary network environment 100, a central distribution server 102 is coupled to multiple scan/patch servers 104 via a network 106(a). A scan/patch server 104 is typically coupled through a network 106(b) to a plurality of client computers 108(1) - 108(n). Network 106 is intended to represent any of a variety of conventional network topologies and types (including optical, wired and/or wireless networks), employing any of a variety of conventional network protocols (including public and/or proprietary protocols). Network 106 may include, for example, the Internet as well as possibly at least portions of one or more local area networks (LANs) and/or wide area networks (WANs). Networks 106(a) and 106(b) may be the same network such as the Internet, or they may be networks isolated from one another such as the Internet and a corporate LAN.

Distribution server 102 and scan/patch servers 104 are typically implemented as standard Web servers, and can each be any of a variety of conventional computing devices, including desktop PCs, notebook or portable computers, workstations, mainframe computers, Internet appliances, combinations thereof, and so on. One or more of the servers 102 and 104 can be the same types of devices, or alternatively different types of devices. An exemplary computing environment for implementing a distribution server 102 and a scan/patch server 104 is described in more detail herein below with reference to Fig. 7.

Client computers 108 function in a typical client/server relationship with a server 104 wherein multiple clients 108 make requests to a server 104 that services the requests. Client computers 108 can be any of a variety of conventional computing devices, including desktop PCs, notebook or portable computers, workstations, mainframe computers, gaming consoles, handheld PCs, cellular telephones or other wireless communications devices, personal digital assistants (PDAs), combinations thereof, and so on. One or more of the client computers 108 can be the same types of devices, or alternatively different types of devices. An exemplary computing environment for implementing a client computer 108 is described in more detail herein below with reference to Fig. 7.

In general, automatic and comprehensive detection and patching of vulnerable binary files on client computers 108 is achieved through updates made through distribution server 102 that include binary signatures for identifying vulnerable binary files and security patches configured to fix vulnerable files. As discussed in greater detail below with respect to the following exemplary embodiments, the binary signatures and security patches are distributed to scan/patch servers 104 which in turn, either actively scan for and update vulnerable binary files on client computers 108, or push the binary signatures and security patches down to the client computers 108 so the client computers 108 can perform the scanning for and patching of vulnerable binary files.

Exemplary Embodiments

Fig. 2 illustrates an exemplary embodiment of a distribution server 102, a scan-patch server 104 and a client computer 108 suitable for implementing automatic detection and patching of security vulnerabilities in binary files.

Distribution server 102 includes a distribution module 200 and a database 202 for receiving and holding binary signatures and security patches. Database 202 can be updated with binary signatures and security patches in a variety of ways including, for example, through a portable storage medium (not shown, but see Fig. 7) or through a computer device (not shown) coupled to the server 102 and configured to upload binary signatures and security patches to database 202.

A typical scenario in which a database 202 might be updated begins with an investigation of a software product (e.g., a operating system, application program, etc.) initiated by the developer of the software product. For example, a developer may hire a security consultancy firm to attempt to find security vulnerabilities in a newly released software product. If a security vulnerability is discovered in a software product through hacking or by some other means, an exact bit pattern of the vulnerable function within the product can be identified. The bit pattern represents a binary signature of the vulnerable section in the binary file, which is a component of a software product.

Once a security vulnerability is discovered and analyzed, a fix can be developed that will eliminate the vulnerability. Such fixes are called security patches and they represent revised code modules compiled into binary executables. Security patches can be installed on computers that are identified through the binary signature as running software that has the security vulnerability. Installation of the security patch will fix the security vulnerability. The distribution server 102 enables software product vendors and others to upload binary signatures of vulnerable binary files along with the security patches designed to fix the vulnerable binary files, into the database 202 for distribution.

Distribution module 200 is configured to distribute binary signatures and security patches from database 202 to various scan-patch servers 104 via a network 106. Distribution module 200 typically functions automatically to distribute binary signatures and security patches from database 202 whenever the database 202 is updated with additional signatures and patches. Automatic distribution may be achieved in a variety of ways including, for example, through communication from distribution module 200 to scan-patch servers 104 indicating that updated binary signatures and security patches are available and waiting for requests to send the binary signatures and security patches, or by automatically forwarding updated binary signatures and security patches to scan-patch servers 104 configured to accept the updates.

In the embodiment of Fig. 2, a scan-patch server 104 includes a scan-patch module 204 and a database 206 for receiving and holding binary signatures and security patches. Database 206 is typically updated automatically with new binary signatures and security patches through communications between the scan-patch module 204 and the distribution module 200 on distribution server 102. In addition to updating database 206 with binary signatures and security patches, scan-patch module 204 is configured to access client computer 108 and scan binary files 208 for binary signatures. Scanning binary files 208 can include searching for a binary signature in binary files present on any form of media present on or accessible by client computer 108. Binary files 208 typically include compiled, computer/processor-readable code such as an operating system or an application program file. However, it is noted that binary files 208 can be any form of binary information including computer/processor-readable instructions, data structures, program modules, and other data for client computer 108.

As noted below in the discussion referring to the exemplary computer environment of Fig. 7, such media on a client computer 108 can include any available media that is accessible by client computer 108, such as volatile and non-volatile media as well as removable and non-removable media. Such computer/processor-readable media can include volatile memory, such as random access memory (RAM) and/or non-volatile memory, such as read only memory (ROM). Computer/processor-readable media can also include other removable/non-removable, volatile/non-volatile computer storage media, such as, for example, a hard disk drive for reading from and writing to a non-removable, non-volatile magnetic media, a magnetic disk drive for reading from and writing to a removable, non-volatile magnetic disk (e.g., a "floppy disk"), an optical disk drive for reading from and/or writing to a removable, non-volatile optical disk such as a CD-ROM, DVD-ROM, or other optical media, other magnetic storage devices, flash memory cards, electrically erasable programmable read-only memory (EEPROM), network-attached storage, and the like. All such computer/processor-readable media providing both volatile and non-volatile storage of any form of binary files 208, including computer/processor-readable instructions, data structures, program modules, and other data for client computer 108, is accessible for scanning by scan-patch server 104 via scan-patch module 204.

Scan-patch module 204 thus searches binary files 208 on client computer 108 to determine if a binary signature identifying a security vulnerability is present in any binary information located on client computer 108. If the bit pattern of the binary signature is found in a binary file 208, scan-patch module 204 operates to fix the security vulnerability in the binary file 208 by installing a corresponding

security patch on client computer 108. Installation of a security patch on client computer 108 overwrites or otherwise eliminates the binary file or a portion of the binary file containing the security vulnerability.

Fig. 3 illustrates another exemplary embodiment of a distribution server 102, a scan-patch server 104 and a client computer 108 suitable for implementing patching of security vulnerabilities in binary files. In general, in the Fig. 3 embodiment, binary signatures and security patches are pushed down, or redistributed, from the server 104 to the client computer 108, and the scanning for security vulnerable files and the patching of security vulnerable files is performed by the client computer 108 instead of the scan patch server 104.

In the Fig. 3 embodiment, distribution server 102 is configured in the same manner as discussed above with respect to the embodiment of Fig. 2. Thus, database 202 can be updated to include newly discovered binary signatures that identify security vulnerabilities in binary files. Database 202 can also be updated with corresponding security patches that have been developed to fix such security vulnerabilities.

The scan-patch server 102 of Fig. 3 is configured in somewhat the same manner as that discussed above with respect to Fig. 2. Thus, scan-patch server 102 of Fig. 3 includes a database 206 for receiving and holding binary signatures and security patches. Database 206 is typically updated automatically with new binary signatures and security patches through communications between the scan-patch server 104 and the distribution server 102. However, the communication between the scan-patch server 104 and the distribution server 102 is conducted through a redistribution module 300 instead of a scan-patch module 204 as discussed with respect to the Fig. 2 embodiment.

The redistribution module 300, in addition to updating database 206 with binary signatures and security patches, is configured to communicate with scan-patch module 302 on client computer 108 and transfer a binary signature to the client computer 108. Scan-patch module 302 is configured to receive the binary signature and to scan binary files 208 to determine if the binary signature is present in any binary information located on client computer 108. Thus, the scan-patch module 302 of Fig. 3 functions in a manner similar to the scan-patch module 204 discussed above with reference to Fig. 2.

If the bit pattern of the binary signature is found in a binary file 208 on client computer 108, scan-patch module 302 sends a request to the redistribution module 300 on server 102. The request is to have the redistribution module 300 send the security patch corresponding with the binary signature down to the client computer 108. The redistribution module 300 responds to the request by sending the appropriate security patch to client computer 108. The scan-patch module 302 receives the security patch and operates to fix the security vulnerability in the binary file 208 by installing the security patch on client computer 108. As in the Fig. 2 embodiment, installation of a security patch on client computer 108 overwrites or otherwise eliminates the binary file or a portion of the binary file containing the discovered security vulnerability.

Exemplary Methods

Example methods for implementing automatic detection and patching of security vulnerabilities in binary files will now be described with primary reference to the flow diagrams of Figs. 4 - 6. The methods apply generally to the exemplary embodiments discussed above with respect to Figs. 1 - 3. The elements

of the described methods may be performed by any appropriate means including, for example, by hardware logic blocks on an ASIC or by the execution of processor-readable instructions defined on a processor-readable medium.

A "processor-readable medium," as used herein, can be any means that can contain, store, communicate, propagate, or transport instructions for use by or execution by a processor. A processor-readable medium can be, without limitation, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, device, or propagation medium. More specific examples of a processor-readable medium include, among others, an electrical connection (electronic) having one or more wires, a portable computer diskette (magnetic), a random access memory (RAM) (magnetic), a read-only memory (ROM) (magnetic), an erasable programmable-read-only memory (EPROM or Flash memory), an optical fiber (optical), a rewritable compact disc (CD-RW) (optical), and a portable compact disc read-only memory (CDROM) (optical).

Fig. 4 shows an exemplary method 400 for implementing automatic detection and patching of security vulnerabilities in binary files. The binary files are typically located or stored on a client computer being served by a server computer, but they may also be located on the server computer itself, or any other computing device accessible by the server computer. At block 402 of method 400, a binary signature is received. The binary signature is a bit pattern that has been associated with a security vulnerability in a particular binary file, such as an executable application program or operating system running on a client computer. The binary signature is received from a central distribution server 102 by a subordinate server 104.

At block 404, a security patch is received. The security patch is typically compiled executable code that has been developed as a fix to the security vulnerability of the particular binary file. The security patch is also received from the central distribution server 102 by the subordinate server 104. At block 406, a vulnerable binary file is identified based on the binary signature. The identification of the vulnerable binary file is typically achieved by scanning binary information stored on various media of a computer, such as client computer 108, and then comparing the pattern(s) in the binary signature with the binary information found on the media. The identification can happen in various ways including, for example, by the server 104 scanning and comparing all the binary information present on the client computer. The identification of a vulnerable binary file can also be achieved by having the server 104 push the binary signature down to the client computer so that the client computer can perform the scan and comparison.

At block 408 of method 400, the security patch is used to update the vulnerable binary file. The update can be achieved in various ways including, for example, by the server 104 installing the security patch on the client computer 108. If the client computer 108 has performed the scan and identified the vulnerable binary file, the client computer 108 may request that the server 104 send the security patch to the computer 108, in which case the computer 108 can install the security patch to fix the vulnerable binary file.

Fig. 5 shows another exemplary method 500 for implementing automatic detection and patching of security vulnerabilities in binary files. The method 500 generally illustrates the distribution of binary signatures for security vulnerabilities and the security patches developed for fixing those security vulnerabilities. At

block 502 of method 500, a binary signature is received that identifies a security vulnerability of a binary file. The binary signature is typically uploaded to a distribution server 102 as a newly discovered bit pattern that identifies a vulnerability in a binary file of a software product that may be widely distributed across many computers on a network such as the Internet. The upload is typically achieved from a computer coupled to the distribution server 102 or from a portable storage medium inserted into the distribution server 102. At block 504, a security patch configured to fix the security vulnerability is received by the distribution server 102 in a similar manner as the binary signature.

At block 506, the binary signature and the security patch are distributed to a plurality of subordinate servers 104 from distribution server 102. This distribution occurs automatically and can be achieved in various ways. For example, upon receiving an uploaded binary signature and security patch, the distribution server 102 can automatically send the binary signature and security patch out over the network to all subordinate servers 104 configured to receive updated binary signatures and security patches. The distribution server 102 might also send a notice to servers 104 indicating that a security vulnerability has been discovered and that a security patch is available to fix the vulnerability. Subordinate servers 104 can then request that the distribution server 102 send the binary signature that identifies the security vulnerability and the security patch. Upon receiving a request, the distribution server 102 can forward the binary signature and the security patch to requesting servers 102.

Fig. 6 shows another exemplary method 600 for implementing automatic detection and patching of security vulnerabilities in binary files. At block 602 of method 600, a client computer 108 receives a binary signature from a server 104.

21

The binary signature is associated with a security vulnerability in a binary file that may be present on the client computer 108. At block 604, the client computer 108 scans all the binary information presently available to it and compares the pattern(s) in the binary signature with the binary information. The binary information scanned by the client computer 108 is typically in the form of computer/processor-readable and/or executable instructions, data structures, program modules, and other data useful for client computer 108, and can reside on both volatile and non-volatile storage media of various types.

At block 606, if the client computer 108 finds a binary file that contains the binary signature, it sends a request to the server 104 to have the security patch transferred. At block 608, the client computer 108 receives the security patch, and at block 610, the client computer 108 installs the security patch in order to fix the security.vulnerability in the binary file containing binary information matching the pattern(s) in the binary signature.

While one or more methods have been disclosed by means of flow diagrams and text associated with the blocks of the flow diagrams, it is to be understood that the blocks do not necessarily have to be performed in the order in which they were presented, and that an alternative order(s) may result in similar advantages. Furthermore, the methods are not exclusive and can be performed alone or in combination with one another.

Exemplary Computer

Fig. 7 illustrates an exemplary computing environment suitable for implementing a distribution server 102, a scan-patch server 104, and a client computer 108, as discussed above with reference to Figs. 1 - 3. Although one

specific configuration is shown in Fig. 7, distribution server 102, scan-patch server 104, and client computer 108 may be implemented in other computing configurations.

The computing environment 700 includes a general-purpose computing system in the form of a computer 702. The components of computer 702 can include, but are not limited to, one or more processors or processing units 704, a system memory 706, and a system bus 708 that couples various system components including the processor 704 to the system memory 706.

The system bus 708 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. An example of a system bus 708 would be a Peripheral Component Interconnects (PCI) bus, also known as a Mezzanine bus.

Computer 702 typically includes a variety of computer-readable media. Such media can be any available media that is accessible by computer 702 and includes both volatile and non-volatile media, removable and non-removable media. The system memory 706 includes computer readable media in the form of volatile memory, such as random access memory (RAM) 710, and/or non-volatile memory, such as read only memory (ROM) 712. A basic input/output system (BIOS) 714, containing the basic routines that help to transfer information between elements within computer 702, such as during start-up, is stored in ROM 712. RAM 710 typically contains data and/or program modules that are immediately accessible to and/or presently operated on by the processing unit 704.

Computer 702 can also include other removable/non-removable, volatile/non-volatile computer storage media. By way of example, Fig. 7

illustrates a hard disk drive 716 for reading from and writing to a non-removable, non-volatile magnetic media (not shown), a magnetic disk drive 718 for reading from and writing to a removable, non-volatile magnetic disk 720 (e.g., a "floppy disk"), and an optical disk drive 722 for reading from and/or writing to a removable, non-volatile optical disk 724 such as a CD-ROM, DVD-ROM, or other optical media. The hard disk drive 716, magnetic disk drive 718, and optical disk drive 722 are each connected to the system bus 708 by one or more data media interfaces 726. Alternatively, the hard disk drive 716, magnetic disk drive 718, and optical disk drive 722 can be connected to the system bus 708 by a SCSI interface (not shown).

The disk drives and their associated computer-readable media provide non-volatile storage of computer readable instructions, data structures, program modules, and other data for computer 702. Although the example illustrates a hard disk 716, a removable magnetic disk 720, and a removable optical disk 724, it is to be appreciated that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes or other magnetic storage devices, flash memory cards, CD-ROM, digital versatile disks (DVD) or other optical storage, random access memories (RAM), read only memories (ROM), electrically erasable programmable read-only memory (EEPROM), and the like, can also be utilized to implement the exemplary computing system and environment.

Any number of program modules can be stored on the hard disk 716, magnetic disk 720, optical disk 724, ROM 712, and/or RAM 710, including by way of example, an operating system 726, one or more application programs 728, other program modules 730, and program data 732. Each of such operating system

726, one or more application programs 728, other program modules 730, and program data 732 (or some combination thereof) may include an embodiment of a caching scheme for user network access information.

Computer 702 can include a variety of computer/processor readable media identified as communication media. Communication media typically embodies computer readable instructions, data structures, program modules, or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared, and other wireless media. Combinations of any of the above are also included within the scope of computer readable media.

A user can enter commands and information into computer system 702 via input devices such as a keyboard 734 and a pointing device 736 (e.g., a "mouse"). Other input devices 738 (not shown specifically) may include a microphone, joystick, game pad, satellite dish, serial port, scanner, and/or the like. These and other input devices are connected to the processing unit 704 via input/output interfaces 740 that are coupled to the system bus 708, but may be connected by other interface and bus structures, such as a parallel port, game port, or a universal serial bus (USB).

A monitor 742 or other type of display device can also be connected to the system bus 708 via an interface, such as a video adapter 744. In addition to the monitor 742, other output peripheral devices can include components such as

speakers (not shown) and a printer 746 which can be connected to computer 702 via the input/output interfaces 740.

Computer 702 can operate in a networked environment using logical connections to one or more remote computers, such as a remote computing device 748. By way of example, the remote computing device 748 can be a personal computer, portable computer, a server, a router, a network computer, a peer device or other common network node, and the like. The remote computing device 748 is illustrated as a portable computer that can include many or all of the elements and features described herein relative to computer system 702.

Logical connections between computer 702 and the remote computer 748 are depicted as a local area network (LAN) 750 and a general wide area network (WAN) 752. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet. When implemented in a LAN networking environment, the computer 702 is connected to a local network 750 via a network interface or adapter 754. When implemented in a WAN networking environment, the computer 702 typically includes a modem 756 or other means for establishing communications over the wide network 752. The modem 756, which can be internal or external to computer 702, can be connected to the system bus 708 via the input/output interfaces 740 or other appropriate mechanisms. It is to be appreciated that the illustrated network connections are exemplary and that other means of establishing communication link(s) between the computers 702 and 748 can be employed.

In a networked environment, such as that illustrated with computing environment 700, program modules depicted relative to the computer 702, or portions thereof, may be stored in a remote memory storage device. By way of

example, remote application programs 758 reside on a memory device of remote computer 748. For purposes of illustration, application programs and other executable program components, such as the operating system, are illustrated herein as discrete blocks, although it is recognized that such programs and components reside at various times in different storage components of the computer system 702, and are executed by the data processor(s) of the computer.

Conclusion

Although the invention has been described in language specific to structural features and/or methodological acts, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or acts described. Rather, the specific features and acts are disclosed as exemplary forms of implementing the claimed invention.

CLAIMS

1. A processor-readable medium comprising processor-executable instructions configured for:

receiving a binary signature;

receiving a security patch;

identifying a vulnerable binary file on a computer based on the binary signature; and

updating the vulnerable binary file on the computer with the security patch.

2. A processor-readable medium as recited in claim 1, wherein the identifying a vulnerable binary file on a computer includes comparing a bit pattern of the binary signature against binary files located on the computer, the bit pattern associated with a security vulnerability.

3. A processor-readable medium as recited in claim 1, wherein the updating the vulnerable binary file on the computer includes installing the security patch on the computer.

4. A processor-readable medium as recited in claim 1, wherein the identifying a vulnerable binary file on a computer includes sending the binary signature to the computer.

5. A processor-readable medium as recited in claim 4, wherein the updating the vulnerable binary file on the computer includes:

receiving a request from the computer to send the security patch; and

sending the security patch to the computer.

6. A processor-readable medium as recited in claim 1, wherein the computer is a client computer and the receiving includes receiving the binary signature and the security patch from a distribution server configured to distribute to the client computer, binary signatures that identify vulnerable files and security patches configured to fix the vulnerable files.

7. A server comprising the processor-readable medium as recited in claim 1.

8. A processor-readable medium comprising processor-executable instructions configured for:

receiving a binary signature that identifies a security vulnerability in a binary file;

receiving a security patch configured to fix the security vulnerability in the binary file; and

distributing the binary signature and the security patch to a plurality of servers.

9. A processor-readable medium as recited in claim 8, wherein the distributing includes:

sending a notice to each of the plurality of servers regarding the security vulnerability and the available patch;

receiving a request to send the binary signature and the security patch; and

sending the binary signature and the security patch in response to the request.

10. A distribution server comprising the processor-readable medium as recited in claim 8.

11. A processor-readable medium comprising processor-executable instructions configured for:

receiving a binary signature from a server;

searching for the binary signature in binary files;

sending a request to the server for a security patch if a binary file is found that includes the binary signature;

receiving the security patch from the server; and

updating the binary file with the security patch.

12. A client computer comprising the processor-readable medium as recited in claim 11.

13. A method comprising:

receiving a binary signature;

searching for a vulnerable file based on the binary signature;

if a vulnerable file is found, requesting a security patch; and

fixing the vulnerable file with the security patch.

14. A method as recited in claim 13, wherein the requesting includes sending a request to a server for the security patch, the method further comprising receiving the security patch from the server in response to the request.

15. A method as recited in claim 14, wherein the receiving includes receiving the binary signature from the server.

16. A method as recited in claim 13, wherein the fixing includes installing the security patch on a computer.

17. A method as recited in claim 13, wherein the searching includes comparing the binary signature to binary information on a storage medium of a computer.

18. A method as recited in claim 17, wherein the binary information is selected from the group comprising:

an operating system;

an application program file; and

a data file.

19. A method as recited in claim 17, wherein the storage medium is selected from the group comprising:

a hard disk;

a magnetic floppy disk;

an optical disk;

a flash memory card;

an electrically erasable programmable read-only memory; and

network-attached storage.

20. A method comprising:

receiving a binary signature and a security patch from a distribution server;

searching on a client computer for a vulnerable file associated with the binary signature; and

if a vulnerable file is found, fixing the vulnerable file with the security patch.

21. A method as recited in claim 20, wherein the searching includes transferring the binary signature to the client computer, the client computer configured to search for a vulnerable file associated with the binary signature.

22. A method as recited in claim 21, wherein the fixing includes:

receiving a request from the client computer to transfer the security patch, the client computer having located a vulnerable file; and

transferring the security patch to the client computer in response to the request.

23. A computer comprising:

means for receiving a binary signature;

means for searching for a vulnerable file based on the binary signature;

means for requesting a security patch if a vulnerable file is found; and

means for fixing the vulnerable file with the security patch.

24. A server comprising:

means for receiving a binary signature and a security patch from a distribution server;

means for scanning a client computer for a vulnerable file associated with the binary signature; and

means for fixing the vulnerable file with the security patch if a vulnerable file is found.

25. A computer comprising:

binary information;

a scan module configured to receive a binary signature and scan the binary information for the binary signature; and

a patch module configured to request a security patch and install the security patch if the binary signature is found in the binary information.

26. A computer as recited in claim 25, further comprising a storage medium configured to retain the binary information.

27. A computer as recited in claim 25, wherein the binary information is selected from the group comprising:

an operating system;

an application program file; and

a data file.

28. A computer comprising:

binary files;

a binary signature; and

a security patch module configured to receive the binary signature from a server and to scan the binary files in search of the binary signature.

29. A computer as recited in claim 28, further comprising:

a binary file that includes the binary signature; and

a security patch;

wherein the security patch module is further configured to request the security patch from the server upon locating the binary signature within the binary file, and to apply the security patch to the binary file.

30. A distribution server comprising:

a database; and

a distribution module configured to receive a binary signature and a security patch, store the binary signature and the security patch in the database, and distribute the binary signature and the security patch to a plurality of servers.

31. A distribution server as recited in claim 30, wherein the distribution module is further configured to receive a request from a server for the binary signature and the security patch and to distribute the binary signature and the security patch to the server in response to the request.

32. A server comprising:

a binary signature associated with a security vulnerability in a binary file;

a security patch configured to fix the security vulnerability in the binary file; and

a scan module configured to scan binary files on a client computer for the binary signature and to update the binary file with the security patch if the binary signature is found.

33. A server as recited in claim 32, further comprising:

a database;

the scan module further configured to receive the binary signature and the security patch from a distribution server and to store the binary signature and the security patch in the database.

ABSTRACT

Systems and methods are described that enable patching of security vulnerabilities in binary files. The detection and patching of vulnerable binary files is automatic, reliable, regression free, and comprehensive across networks on an unlimited scale. These advantages can be realized in various ways including, for example, by leveraging current anti-virus infrastructure that is widely deployed across the Internet. Reliable discovery of vulnerable binary files (e.g., in operating systems, application programs, etc.) is achieved through the use of binary signatures that have been associated with discovered security vulnerabilities. A divergence of security patches away from conventional service packs provides for the possibility of production of regression-free fixes for security vulnerabilities in binary files.

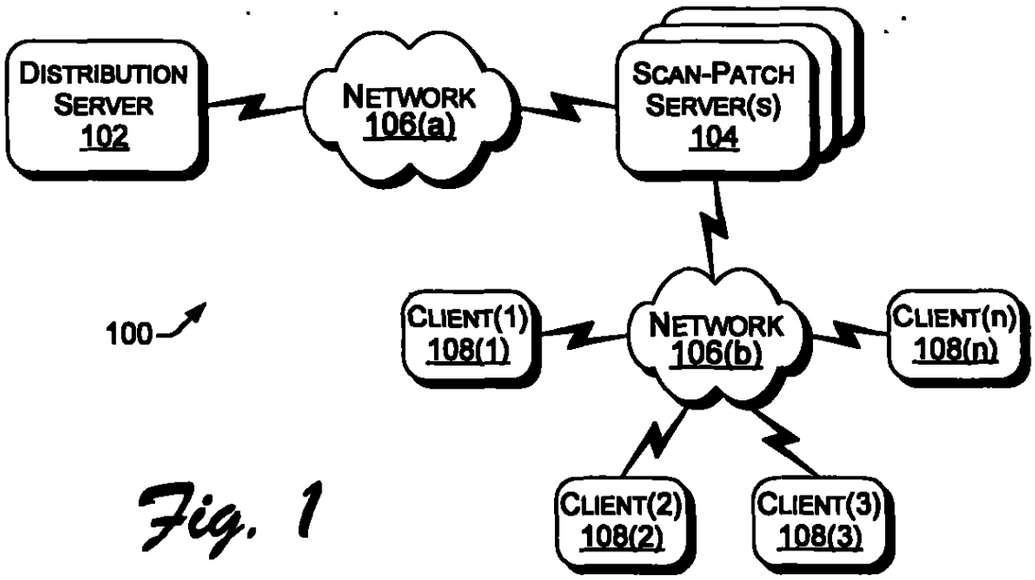

DISTRIBUTION SERVER 102
NETWORK 106(a)
SCAN-PATCH SERVER(S) 104
100
CLIENT(1) 108(1)
NETWORK 106(b)
CLIENT(n) 108(n)
CLIENT(2) 108(2)
CLIENT(3) 108(3)

Fig. 1

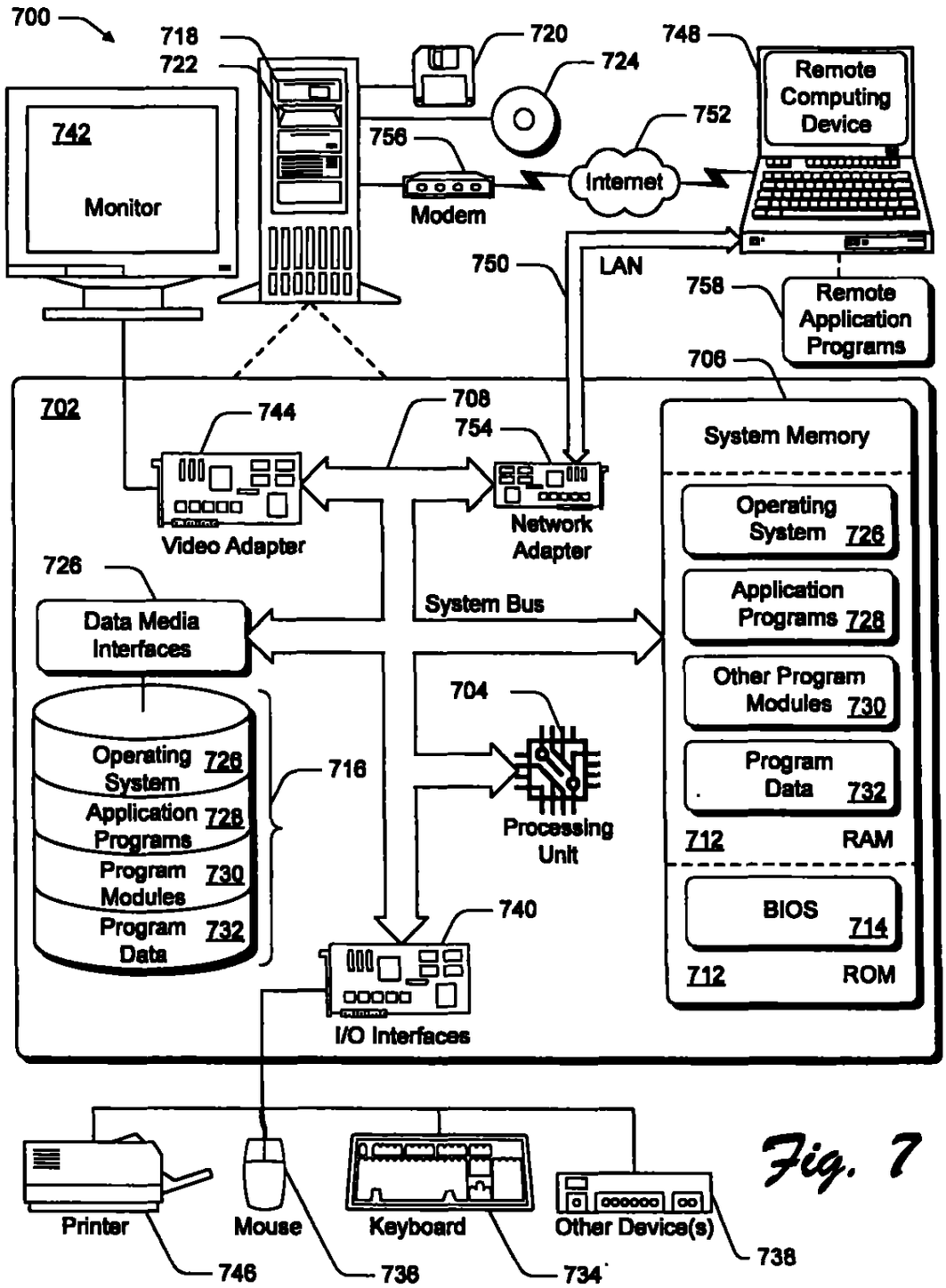

700
718
722
720
724
748
Remote Computing Device
742
Monitor
752
756
Modem
Internet
750
LAN
758
Remote Application Programs
706
702
744
708
754
System Memory
Video Adapter
Network Adapter
Operating System 726
Application Programs 728
Other Program Modules 730
Program Data 732
726
Data Media Interfaces
System Bus
704
716
Operating System 726
Application Programs 728
Program Modules 730
Program Data 732
Processing Unit
712 RAM
BIOS 714
712 ROM
740
I/O Interfaces
Printer
Mouse
Keyboard
Other Device(s)
746
736
734
738

Fig. 7

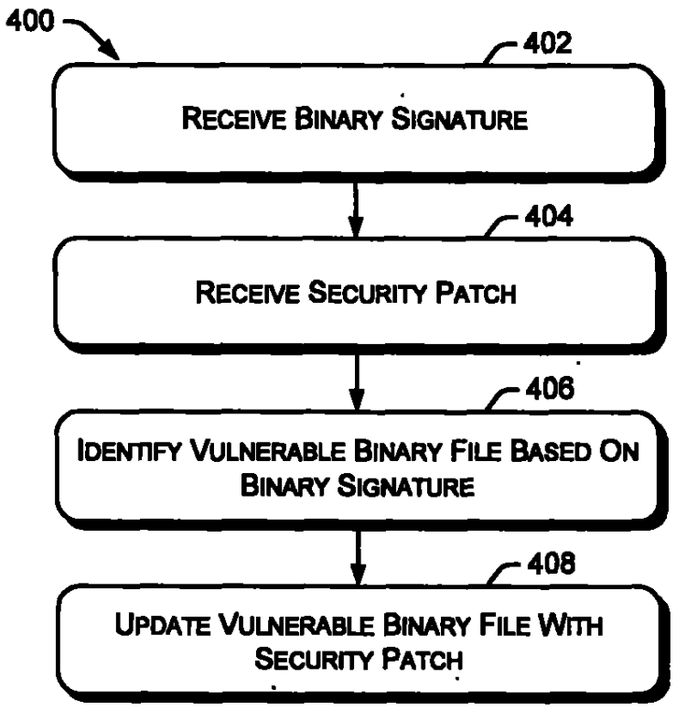

400
402
RECEIVE BINARY SIGNATURE
404
RECEIVE SECURITY PATCH
406
IDENTIFY VULNERABLE BINARY FILE BASED ON BINARY SIGNATURE
408
UPDATE VULNERABLE BINARY FILE WITH SECURITY PATCH

Fig. 4

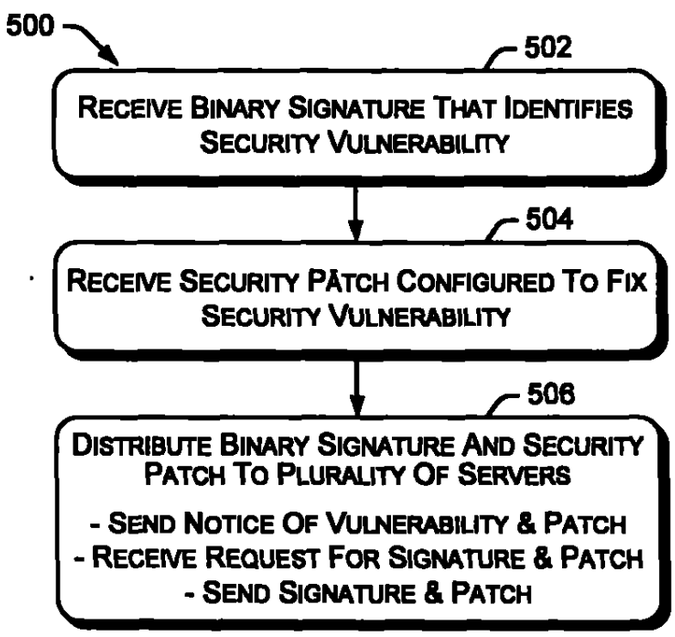

500
502
RECEIVE BINARY SIGNATURE THAT IDENTIFIES SECURITY VULNERABILITY
504
RECEIVE SECURITY PATCH CONFIGURED TO FIX SECURITY VULNERABILITY
506
DISTRIBUTE BINARY SIGNATURE AND SECURITY PATCH TO PLURALITY OF SERVERS
- SEND NOTICE OF VULNERABILITY & PATCH
- RECEIVE REQUEST FOR SIGNATURE & PATCH
- SEND SIGNATURE & PATCH

Fig. 5

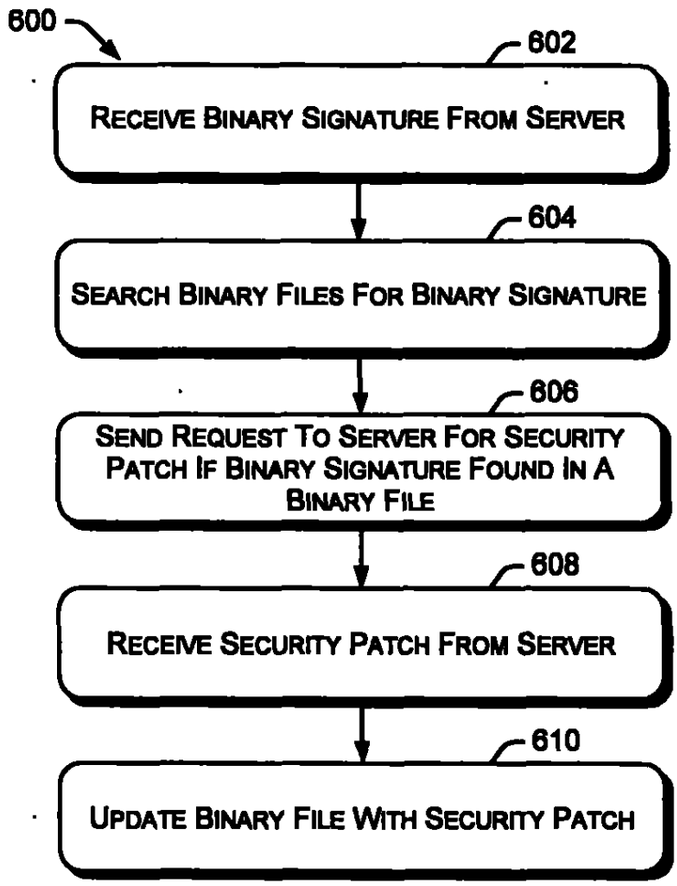

600
602
RECEIVE BINARY SIGNATURE FROM SERVER
604
SEARCH BINARY FILES FOR BINARY SIGNATURE
606
SEND REQUEST TO SERVER FOR SECURITY PATCH IF BINARY SIGNATURE FOUND IN A BINARY FILE
608
RECEIVE SECURITY PATCH FROM SERVER
610
UPDATE BINARY FILE WITH SECURITY PATCH

Fig. 6

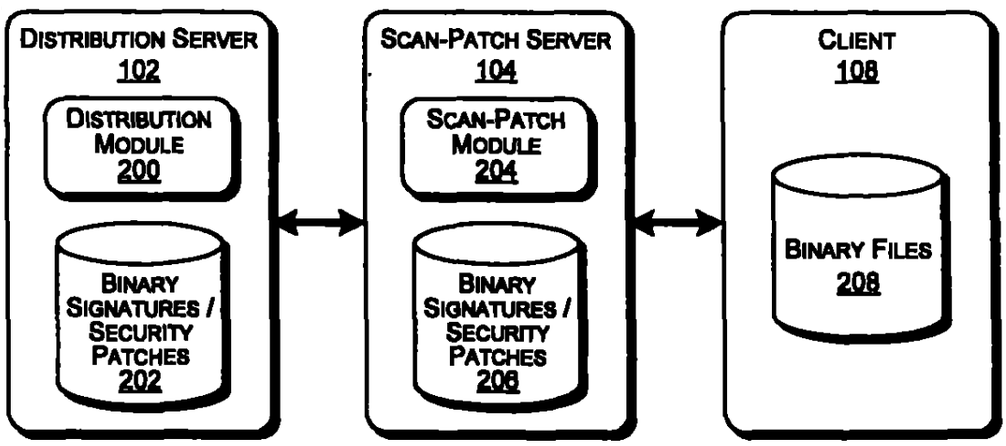

DISTRIBUTION SERVER
102
DISTRIBUTION MODULE
200
BINARY SIGNATURES / SECURITY PATCHES
202
SCAN-PATCH SERVER
104
SCAN-PATCH MODULE
204
BINARY SIGNATURES / SECURITY PATCHES
206
CLIENT
108
BINARY FILES
208

Fig. 2

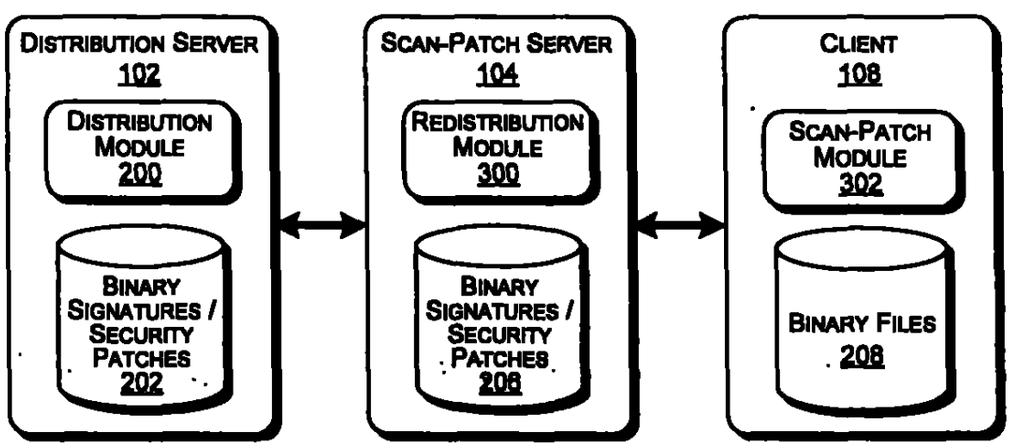

DISTRIBUTION SERVER
102
DISTRIBUTION MODULE
200
BINARY SIGNATURES / SECURITY PATCHES
202
SCAN-PATCH SERVER
104
REDISTRIBUTION MODULE
300
BINARY SIGNATURES / SECURITY PATCHES
206
CLIENT
108
SCAN-PATCH MODULE
302
BINARY FILES
208

Fig. 3

242

SUPERINDUSTRIA Y COMERCIO

Radicacion : 04067547 00000000 Folios: 223

Fecha (AMD): 2004-07-13 16:58:53

Tramite : 002 PATENTES D 1 REGISTRO/ 411 PRESENTAC

Dependencia: 2020 DIVISION DE NUEVAS CREACIONES

NIT : 800176089-2

RECIBO OFICIAL DE CAJA : 04 - 55,265
FECHA : JULIO 13 DE 2004

***** C O N S I G N A C I O N *****

DEPOSITANTE	TIPO PAGO	BANCO	CUENTA	No. PAGO	FECHA PGO	Vr. PAGO
CARDENAS Y CARDENAS P.I.	CONSIGNACION	BANCO POPULAR	050-00110-6	24340628	13/07/2004	9,504,850.00

***** C O N C E P T O *****

CANT.	RENTISTICO	CONCEPTO	TOTAL CONCEPTO
1	50005-01-01 SOLICITUDES	1 TRAMITES DE SOL. DE PATENTE DE IN	422,000.00
		TOTAL : $	422,000.00

SON: CUATROCIENTOS VEINTIDOS MIL PESOS

RESPONSABLE : _____

CARDENAS & CARDENAS

RECIBO DE CAJA APLICADO AL EXPEDIENTE No._____

DETECCION Y PARCHEO AUTOMATICO DE ARCHIVOS VULNERABLES

3

243

SUPERINDUSTRIA Y COMERCIO
Radicacion : 04057847 00000000 Folios: 223
Fecha (AMD): 2004-07-15 16:58:53
Tramite : 1 OMC PATENTES D 1 REGISTRO/ 411 PRESENTAC
Dependencia: 2000 DIVISION DE NUEVAS CREACIONES

NIT : 800176089-2

RECIBO OFICIAL DE CAJA : 04 - 55,266
FECHA : JULIO 13 DE 2004

***** C O N S I G N A C I O N *****

DEPOSITANTE	TIPO PAGO	BANCO	CUENTA	No. PAGO	FECHA PGO	Vr. PAGO
CARDENAS Y CARDENAS P.I.	CONSIGNACION	BANCO POPULAR	050-00110-6	24340628	13/07/2004	9,504,850.00

***** C O N C E P T O *****

CANT.	RENTISTICO	CONCEPTO	TOTAL CONCEPTO
1	50005-01-01 SOLICITUDES	82 INVOCACION DE UNA PRIORIDAD EN NU	118,000.00
		TOTAL : $	118,000.00

SON: CIENTO DIEZ Y OCHO MIL PESOS

RESPONSABLE : _____

CARDENAS & CARDENAS
ABOGADOS

RECIBO DE CAJA APLICADO AL EXPEDIENTE No._____

DETECCION Y PARCHEO AUTOMATICO
DE ARCHIVOS VULNERABLES

4

244

SUPERINDUSTRIA Y COMERCIO
Radicacion : 04067547 00000000 Folios: 223
Fecha (DMD): 2004-07-15 16 50:53
Tramite : 002 PATENTES D 1 REGISTRO/ 411 PRESENTA
Dependencia 2000 DIVISION DE NUEVAS CREACIONES

REQUISITOS NECESARIOS PARA PRESENTACION Y AGILIZACION DEL TRAMITE

	USUARIO	RADICADOR
PATENTE DE INVENCIÓN MODELO DE UTILIDAD	( )	( )
- Indicación de que se solicita la concesión de una patente	( )	( )
- Datos de identificación del solicitante o de la persona que se presenta la solicitud.	( )	( )
- Descripción de la invención.	( )	( )
Dibujos de ser estos pertinentes.	( )	( )
Comprobante de Pago de las tasas establecidas.		
COMPLETA ( ) **INCOMPLETA**	( )	( )
DISEÑO INDUSTRIAL ( )		
- Indicación de que se solicita el registro de Diseño Industrial	( )	( )
- Datos de identificación del solicitante o de la persona que presenta	( )	( )
- la solicitud		
- Representación gráfica o fotográfica del Diseño Industrial o muestra del Material que incorpora el diseño.	( )	( )
Comprobante de pago de las tasas establecidas	( )	( )
COMPLETA ( ) **INCOMPLETA**	( )	( )
ESQUEMA DE TRAZADO ( )		
- Indicación de que solicita el registro de un Esquema de trazado	( )	( )
- Datos de identificación del solicitante o de la persona que presenta La solicitud	( )	( )
Representación gráfica del esquema de trazado	( )	( )
Comprobante de pago de las tasas establecidas	( )	( )
COMPLETA ( ) **INCOMPLETA**	( )	( )

Firma Responsable:

Fecha:

245

Industria y Comercio
SUPERINTENDENCIA

Folio 220

2020
Bogotá DC

Doctor(a)
DARIO CARDENAS NAVAS
Apoderado(a)

REFERENCIA:	Radicación No.	04-67547
	Trámite	002
	Evento	001
	Actuación	416
	Oficio No.	1242
	Folios	001

Teniendo en cuenta que la solicitud de privilegio de patente que se tramita bajo el expediente indicado en la referencia, reúne los requisitos de forma establecidos en los artículos 26 y 27 de la Decisión 486 de la Comisión de la Comunidad Andina, y en las demás disposiciones vigentes sobre la materia, se envía l extracto de la solicitud a la Oficina de Comunicaciones para efecto de su publicación en la Gaceta de Propiedad Industrial.

De acuerdo con lo previsto por el artículo 40 de la Decisión 486/00, la publicación en este caso se hará a partir del **día 15 de enero de 2005.**

Cúmplase
Dado en Bogotá DC, a los 30 JUL. 2004

ALIX CARMENZA CÉSPEDES DE VERGEL
Jefe de la División de Nuevas Creaciones

Carrera 13 No. 27-00 Pisos 5, 7 y 10
Conmutador: 382 0840
Fax: 350 5220
E-mail: info@sic.gov.co
www.sic.gov.co
Bogotá, D.C., Colombia