(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 400 623**

(51) Int. CI.:

G06F 21/00 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **23.04.2008 E 08155001 (4)**

(97) Fecha y número de publicación de la concesión europea: **14.11.2012 EP 1986120**

(54) Título: **Sistemas, aparato, y métodos para detectar malware**

(30) Prioridad:

23.04.2007 US 738882

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
11.04.2013

(73) Titular/es:

MCAFEE, INC. (100.0%)
3965 Freedom Circle
Santa Clara, CA 95054, US

(72) Inventor/es:

ALME, CHRISTOPH

(74) Agente/Representante:

CARVAJAL Y URQUIJO, Isabel

DESCRIPCIÓN

Sistemas, aparato, y métodos para detectar malware

Campo de la Invención

La presente invención se relaciona con la seguridad de redes de ordenadores, y más particularmente, con un sistema y método para detectar posible malware.

Información de Antecedentes

El acceso incrementado a Internet ha tenido el efecto no deseado de aumentar el alcance de los programas de software que capturan información personal de los usuarios sin su autorización ("Spyware") o que corrompen los ordenadores sin el conocimiento y autorización del usuario ("Malware"). Además, ha surgido una industria artesanal en el software que automáticamente descarga y exhibe publicidad mientras que se está utilizando una aplicación ("Adware"). El término malware como se utiliza aquí incluye cualquier tipo de programas de software diseñados para infiltrarse o dañar un sistema de ordenador sin la autorización del propietario, independientemente de la motivación del software, e independientemente de los resultados provocados por el software sobre los dispositivos, sistemas, redes, o datos del propietario.

Cuando dichos programas se instalan en el ordenador del usuario, pueden espiar al usuario, recolectar información confidencial y, en algunos casos, tomar el control del ordenador del usuario. En algunos casos, estos programas de software envían mensajes a otros ordenadores o servidores, proporcionando un conducto para la transferencia de información potencialmente sensible.

Se pueden utilizar diversos programas de detección para intentar detectar la presencia de malware. En algunos casos, los programas de detección se basan en detectar una firma en un software que se examina para determinar si el programa es o contiene, malware. En algunos casos, un programa de detección utiliza un método con base en suma de control para determinar si un software es malware. Sin embargo, los autores del malware frecuentemente cambian partes de los programas con el fin de evitar la detección por firma o los métodos de suma de control. Se pueden crear nuevas variantes de malware conocido al reempacar o compilar dentro de intervalos de tiempo cortos con el fin de evadir la detección con base en firma o suma de control y tomar ventaja del retardo en la creación y distribución de firmas o sumas de control de detección actualizadas.

Los vendedores de software para detección tratan de contrarrestar la cantidad incrementada de nuevas variantes y ejemplos de malware al utilizar detecciones más genéricas, y detecciones mas heurísticas. Sin embargo, las detecciones genéricas tienen la deficiencia de requerir análisis manual de una, en la mayoría de casos de por lo menos dos variantes de malware con el fin de proporcionar una detección apropiada. Adicionalmente, las detecciones heurísticas tienen la deficiencia de falsos positivos

Breve Descripción de los Dibujos

La Figura 1 ilustra un sistema que incluye una puerta de enlace;

La Figura 2 ilustra un diagrama de un posible diseño para un archivo ejecutable;

La Figura 3 ilustra un diagrama de un posible diseño para un archivo ejecutable empacado en tiempo de ejecución;

La Figura 4 ilustra un diagrama de un archivo que incluye una porción 410 del archivo 402 que se ha dividido en una pluralidad de bloques;

La Figura 5 ilustra un diagrama que incluye una comparación de dos huellas ejecutables difusas de dos archivos diferentes;

La Figura 6 ilustra un diagrama que incluye una representación de bloques y sus valores de complejidad de un primer archivo y una representación de bloques y sus valores de complejidad de un segundo archivo;

La Figura 7 ilustra un diagrama que incluye una representación de bloques y sus valores de complejidad de un archivo;

La Figura 8 ilustra un diagrama de flujo para un método de acuerdo con diversas realizaciones; y

La Figura 9 ilustra un diagrama de flujo para un método de acuerdo con diversas realizaciones.

Descripción Detallada de la Invención

En la siguiente descripción detallada de las realizaciones preferidas, se hace referencia a los dibujos acompañantes que forman una parte de esta, y en los que se muestra por vía de ilustración las realizaciones específicas en las que se puede practicar la invención. Se debe entender que se pueden utilizar otras realizaciones y se pueden realizar cambios estructurales sin apartarse del alcance de la presente invención.

La Figura 1 ilustra un sistema 100 que incluye una puerta de enlace 120. La puerta de enlace 120 acopla una pluralidad de dispositivos protegidos 154 a una red 110. Los dispositivos protegidos 154 no se limitan a ningún número o tipos de dispositivos, y pueden incluir cualesquier dispositivos, tales como pero no limitados a ordenadores, servidores, y bases de datos que se acoplan a una red externa 110 a través de una puerta de enlace 110. Los dispositivos 152A y 152B a 152N representan cualquier número de dispositivos, como se ilustra por la línea discontinua 152C, que se acoplan a y se protegen por la puerta de enlace 120. En diversas realizaciones, los dispositivos 152A y 152B a 152N se pueden acoplar mediante una o más redes 150, que también acoplan los dispositivos 152A y 152B hasta 152N a la puerta de enlace 120.

Los dispositivos 152A y 152B a 152N se denominan como “protegidos” porque estos dispositivos son los dispositivos configurados para recibir la protección anti-malware proporcionada por la puerta de enlace 120. En diversas realizaciones, los dispositivos 152A y 152B a 152N se acoplan a través de la puerta de enlace 120 a una red 110, y a uno o más dispositivos 108 acoplados a una red 110. La red 110 no se limita a un tipo o número particular de redes. En diversas realizaciones, la red 110 incluye la Internet. Los dispositivos 108 no se limitan a cualquier tipo o número particular de dispositivos, y en diversas realizaciones incluyen servidores ilustrativos 102A-102N. Los servidores 102A-102N pueden proporcionar uno o más recursos, tales como archivos o páginas web, que pueden ser requeridos por los dispositivos protegidos 152A y 152B a 152N. Debido a que estas solicitudes se acoplan a través de la puerta de enlace 120, la puerta de enlace 120 es operable para explorar el paso de comunicaciones a través de la puerta de enlace 120, y detectar y bloquear malware en las comunicaciones que van hacia, o proceden desde, los dispositivos protegidos 152A y 152B a 152N.

La puerta de enlace 120 no se limita a solo un dispositivo físico o lógico; por ejemplo, puede consistir de un clúster de dispositivos de puerta de enlace, o comunicaciones entre dispositivos protegidos y la red se puede manejar principalmente por un grupo lógico de dispositivos de puerta de enlace, aunque la protección anti-malware del contenido que se maneja se descarga a otro grupo de dispositivos de puerta de enlace.

En diversas realizaciones, la puerta de enlace 120 se puede operar para interceptar comunicaciones entre cualquiera de los dispositivos protegidos 152A y 152B a 152N, y para detectar malware en bloque en estas comunicaciones. En diversas realizaciones, la puerta de enlace 120 puede sondear cualquiera de los dispositivos protegidos 152A y 152B a 152N para determinar si el dispositivo sondeado está contaminado con malware.

En diversas realizaciones, la puerta de enlace 120 incluye un motor anti-malware 122. En diversas realizaciones, la puerta de enlace 120 incluye una base de datos de huellas generadas difusas 124, un controlador de huellas difusas 126, y un generador de huellas difusas 128. La base de datos 124, el comparador 126, y el generador 128 son parte de, y se acoplan lógicamente a, un motor anti-malware 122. En diversas realizaciones, la base de datos 124 se puede operar para almacenar un conjunto de huellas ejecutables difusas. Las huellas ejecutables difusas almacenadas en base de datos 124 incluyen huellas ejecutables difusas generadas desde archivos conocidos por ser malware. La generación de huellas ejecutables difusas no se limita a ningún medio en particular, y puede incluir generar manualmente una huella ejecutable difusa, o generar automáticamente una huella ejecutable difusa al determinar que un ejecutable es malware con base en la comparación con otras huellas ejecutables difusas.

En diversas realizaciones, el generador 128 se puede operar para generar una huella ejecutable difusa a un archivo. En diversas realizaciones, el generador 128 genera la huella ejecutable difusa para un archivo recibido en la puerta de enlace 120 y para la cual se realiza una comparación para determinar si el archivo recibido es malware. El comparador 126 se puede operar para comparar huellas exigibles difusas suministradas por el generador 128 y comparar la huella ejecutable difusa generada por una o más de las huellas ejecutables difusas almacenadas en la base de datos 124.

En diversas realizaciones, el motor anti-malware 122 controla las operaciones de la base de datos 124, el comparador 126, y el generador 128. En diversas realizaciones, las configuraciones 130 incluyen parámetros de configuración y valores almacenados utilizados por el motor anti-malware 122 para controlar los procesos detención de malware. En diversas realizaciones, las configuraciones 130 almacenan uno o más valores de umbral utilizados en los procesos detección de malware, como se describe adicionalmente aquí.

En diversas realizaciones, todo o partes de la implementación del generador 128 y comparador 126 se pueden almacenar dentro de la base de datos 124, utilizando un lenguaje de script o código P. En diversas realizaciones, la puerta de enlace 120 se acopla al servidor de generador de huellas difusas 142 a través de servidor de suministro de actualización 140. En diversas realizaciones, el servidor 142 incluye uno o más conjuntos de formación almacenados 144A-144N. Los conjuntos de formación 144A-144N incluyen uno o más archivos almacenados que se conocen por incluir malware. A partir de los conjuntos de formación 144A-144N, el servidor 142 se puede operar para generar huellas ejecutables difusas para cualquiera de los archivos incluidos en los conjuntos de formación 144A-144N. El servidor de suministro de actualización 140 se puede operar para, en ciertos intervalos o cuando se descubren nuevas variantes de malware, se actualiza con nuevas huellas ejecutables difusas de la base de datos 124 en la puerta de enlace 120. En diversas realizaciones, el mecanismo de actualización 141 controla la actualización de la base de datos 124 con nuevas huellas ejecutables difusas desde el servidor 140.

En diversas realizaciones, el servidor 142 incluye conjunto de formación retirado 146. En varias realizaciones, el servidor 142 incluye una pluralidad de conjuntos de formación retirados representados por el conjunto de formación 146. En diversas realizaciones, el conjunto de formación retirado 146 incluye uno o más archivos que se eliminan cuando las huellas que se generan son falsos positivos. En diversas realizaciones, el conjunto de formación retirado 146 se genera de forma automática cuando se ha generado un falso positivo desde los conjuntos de formación.

En diversas realizaciones, el motor anti-malware 122 determinará que se ha descubierto una nueva variante de malware. En varios casos, un archivo desde uno de los dispositivos protegidos 152A, 152B-15N o sobre la red 110, por ejemplo desde el servidor malicioso 104, será procesado por el motor anti-malware y se determinará que incluye, o es, malware. En varios casos, una huella ejecutable difusa será generada para el archivo de malware detectado, y se puede agregar a la base de datos 124. En diversas realizaciones, el archivo de malware nuevamente descubierto se suministrará al servidor 142 de tal manera que este archivo de malware se pueda agregar a los conjuntos de formación almacenados en el servidor 142.

En diversas realizaciones, los archivos de malware descubiertos nuevamente proporcionados por la puerta de enlace 120 estarán disponibles para el sistema exterior 100 de otros sistemas de detección anti-malware (no mostrados en la Figura 1). En diversas realizaciones, el sistema exterior 100 de los sistemas de detección anti-malware, proporcionarán los archivos de malware descubiertos nuevamente al servidor 142. El servidor 142 puede generar huellas ejecutables difusas para estos archivos de malware suministrados desde el exterior, y actualizar las bases de datos 124 para incluir huellas ejecutables difusas para estos archivos de malware descubiertos en el exterior.

De esta forma, el sistema 100 se puede operar para actualizar automáticamente su propia base de datos 124 cuando se descubren malware conocidos en la puerta de enlace 120, y se pueden operar para proporcionar archivos de malware descubiertos nuevamente a otros sistemas de detección anti-malware. Además, el sistema 100 se puede actualizar de forma automática con huellas ejecutables difusas para archivos malware nuevamente descubiertos en el sistema exterior 100, incluso si los archivos descubiertos nuevamente no se detectan utilizando una comparación de huellas ejecutables difusas. En los casos donde el malware descubierto nuevamente ya no tiene una huella ejecutable difusa generada por el archivo, el servidor 142 o generador 128 se puede utilizar para generar la huella ejecutable difusa para el archivo.

En diversas realizaciones, la base de datos 124 puede incluir una o más tablas. En diversas realizaciones, la base de datos 124 incluye una tabla de precondición 125. En diversas realizaciones, la base de datos 124 incluye una tabla de salto 127. La tabla de precondición 125 y la tabla de salto 127 proporcionan mecanismos para acelerar la búsqueda de las huellas ejecutables difusas almacenadas en la base de datos 124.

En diversas realizaciones, la base de datos 124 incluye una tabla de conjunto de cadena 123, que proporciona un mecanismo para unificar porciones de cadena redundante en los nombres de malware adheridos a las huellas, con el fin de reducir la huella de memoria.

En diversas realizaciones, las huellas ejecutables difusas almacenadas en la base de datos 124 cada una incluye información meta, como tamaño de archivo y tipo de medios, un nombre de malware para ser asignado en pareja, y un conjunto de entidades que reflejan una aproximación de complejidad y ponderación para bloques de datos y código incluido en el archivo. Las huellas ejecutables difusas se pueden clasificar por esta información, tal como tamaño de archivo o tipo de medios. Se pueden normalizar las huellas ejecutables difusas, esto es, huellas ejecutables difusas muy similares se almacenan como una huella fusionada. Adicionalmente, se puede fusionar un número limitado de bloques para reducir la huella de memoria, dado que tienen ponderación baja a media y de complejidad similar; “fusionado” significa reemplazar los bloques por uno que cubra la longitud de todos estos bloques en conjunto, con su complejidad que es aquella del nuevo bloque, que es un bloque más grande.

Un criterio de clasificación de las huellas ejecutables difusas está respaldado por una tabla de precondición 125 y una tabla de salto 127, con el fin de mejorar de forma colaborativa de la localidad de memoria de huellas ejecutables

difusas prospectivas, y reducir errores de página al soportar el sistema operativo para intercambiar mejor las páginas no utilizadas de la base de datos (aquellas que se relacionan con archivos muy pequeños o muy grandes, como paquetes instaladores y métodos de descarga, respectivamente).

En diversas realizaciones, las comparaciones de un archivo contra la base de datos 124 primero necesita evaluar la tabla de precondición 125, con el fin de encontrar si se requiere una búsqueda en la base de datos 124 en todo. Por ejemplo, la tabla de precondición 125 contiene firmas de bytes mágicos para todos los tipos de medios cubiertos por la base de datos 124 - comparar una imagen GIF o JPEG contra una base de datos que solo contiene huellas ejecutables difusas para los archivos PE archivos no necesita ninguna búsqueda adicional, cuando no es posible ninguna coincidencia.

En diversas realizaciones después de pasar la tabla de precondición 125, la comparación utiliza la tabla de salto 127 para determinar el desplazamiento relativo de la primera huella ejecutable difusa que tiene una oportunidad para coincidir el archivo inquirido, por lo menos mediante su tamaño de archivo. De ahí en adelante, se iteran huellas ejecutables difusas posteriores y se comparan hasta que salgan del alcance de los posibles candidatos de coincidencia para el archivo comparado.

Luego de la coincidencia de una huella, se expande su nombre de malware asociado utilizando la tabla de conjunto de cadena 123. Por vía de ilustración, la tabla de conjunto de cadena puede contener entradas tales como “Win32.” En el índice 0 en la tabla, “Trojan.” en el índice 1, “Downloader.” en el índice 2, a través de “.gen” en el índice 7. En lugar de almacenar nombres de malwares extensos como “Trojan.Downloader. Win32.Agent.XY” o “Trajan.Win32.RBot.gen” asociados con una pluralidad de huellas, los nombres de malware en lugar se almacenan como “\1\2\0Agent.XY” y “\1\0RBot\7”, respectivamente (utilizando “\” como un carácter de escape de ejemplo aquí, seguido por el índice en la tabla de conjunto de cadena).

Es probable que los archivos con una alta complejidad (como ejecutables comprimidos o empacados en tiempo de ejecución) sean muy comunes en su diseño para diferenciar un archivo benigno desde un archivo comprimido malicioso o empacado en tiempo de ejecución. La complejidad Kolmogorov, por ejemplo la duración de la representación de programa más corta (d) de un conjunto (s), se utiliza para clasificar candidatos inapropiados de acuerdo con un umbral definible, por ejemplo

$$\frac{|d(s)|}{|s|} \geq 0.94$$

Una implementación real aproximará la complejidad Kolmogorov al, por ejemplo, utilizar un buen algoritmo de compresión. En diversas realizaciones, la puerta de enlace 120 se puede operar para registrar y reportar cualquier malware detectado y cualquier malware bloqueado.

La Figura 2 ilustra un diagrama 200 de un posible diseño para un archivo ejecutable 202. En diversas realizaciones, archivo ejecutable es un archivo que incluye, o es, malware. En diversas realizaciones, el archivo ejecutable 202 es un archivo que tendrá una huella ejecutable difusa generada para el archivo de tal manera que la huella ejecutable difusa para el archivo 202 se puede comparar con las huellas ejecutables difusas para archivos conocidos que incluyen, o son, malware.

En diversas realizaciones, el archivo 202 incluye la sección de código 210, sección de datos 240, y una sección de recurso 260. En diversas realizaciones, la sección 210 incluye tabla de dirección de importación 214, el directorio de depuración 216, el código de máquina 218, el directorio de importación 220, tabla de nombres de importación 222, y espacio de relleno 224. En diversas realizaciones, la sección de datos 240 incluye datos de inicialización 242 y espacio de relleno 244. En diversas realizaciones, la sección de recurso 260 incluye directorio de recurso 264 y espacio de relleno 266.

En diversas realizaciones, dependiendo del compilador y montador de generación, el nivel de complejidad varía a través de las diferentes porciones del archivo 202. El espacio de relleno 224, 244, y 266 son normalmente áreas de muy baja entropía de información, como se describe adicionalmente aquí y cuando se utilizan como una medición de la complejidad de la sección. La tabla de nombres de importación 222 y directorio de recurso 264 son de manera general áreas de baja o media entropía de información.

El código de máquina 218 es de manera general un área de entropía de información media. Además, el código de máquina 218 es un probable punto en el archivo 202 que incluye el archivo que origina programación 202 que es malware. En diversas realizaciones, el punto de entrada 226 se puede seleccionar en el código 218 para un punto de partida para comparar los bloques utilizados en la huella ejecutable difusa para el archivo 202, como se describe

adicionalmente aquí, y para los archivos que se comparan con el archivo 202 que utiliza una comparación entre una huella ejecutable difusa para el archivo 202 y una huella ejecutable difusa para el archivo comparado.

La Figura 3 ilustra un diagrama 300 de un posible diseño de un archivo ejecutable empacado en tiempo de ejecución 302, teniendo un programa de compresión llamado “UPX” como ejemplo. En diversas realizaciones, el archivo 302 es un archivo que incluye malware. En diversas realizaciones, el archivo 302 es un archivo que tendrá una huella ejecutable difusa generada para el archivo con el fin de que la huella ejecutable difusa para el archivo 302 se pueda comparar con las huellas ejecutables difusas para los archivos conocidos que incluyen malware.

En diversas realizaciones, el archivo 302 incluye la sección UPX0 310, sección UPX1 340, y una sección de recurso 370. En diversas realizaciones, la sección UPX0 310 no incluye datos en bruto 312, en cambio denota espacio de memoria requerido 314 para el código desempacado, y espacio de relleno 316. En diversas realizaciones, la sección UPX1 340 incluye código empacado 344, bucle decodificador 346, y espacio de relleno 348. En diversas realizaciones, la sección de recurso 370 incluye sección de datos en bruto 372, directorio de recurso 374, directorio de importación del programa de compresión 376, y espacio de relleno 378.

En diversas realizaciones, el nivel de complejidad varía a través de las diferentes porciones del archivo 302. El espacio de relleno 316, 348, y 378 son normalmente áreas de muy baja entropía de información. El código empacado 344 es de manera general un área de alta entropía de información. Los datos en el código empacado 344 son códigos ejecutables comprimidos, sin embargo, esto es bien detectable, pueden estar no empacados por un motor anti-malware, y no se empaca una cantidad incrementada de programas de malware, y la detección se enfoca en ejecutables no empacados. En diversas realizaciones, el punto de entrada 350 en el bucle decodificador, que no se empaca en tiempo de ejecución, se utiliza para un punto de partida para comparar bloques utilizados en la huella ejecutable difusa para el archivo 302, como se describe adicionalmente aquí, y para los archivos que se comparan con el archivo 302 utilizando una comparación entre una huella ejecutable difusa para el archivo 202 y una huella ejecutable difusa para el archivo comparado.

La Figura 4 ilustra un diagrama 400 de un archivo 402 que incluye una porción 410 del archivo 402 se ha dividido en una pluralidad de bloques 412. En diversas realizaciones, la pluralidad de bloques 412 puede incluir una pluralidad de filas de bloques, tales como bloques 420, 422, 424, 426, y 428 en una primera fila, y una pluralidad de filas adicionales 429, que incluye una última fila que incluye bloques 440, 442, 444, 446, y 448. El número de bloques en una fila no se limita a cualquier número particular de bloques y puede tener más o menos bloques que se ilustran en la Figura 4. El número de filas en la porción del archivo 410 no se limita a ningún número particular, y puede incluir cualquier número de filas como se ilustra por las líneas discontinuas 430 en la Figura 4.

Como se muestra en el diagrama 400, la porción ilustrativa del archivo 410 puede ser solo una porción de un archivo que se ha seleccionado desde un archivo completo con base en una porción del archivo completo esto es más favorable para utilizar en la detección de malware. Como una dicha implementación de ejemplo, se puede indicar un bloque, o su ponderación se asigna en cero, con el fin de ser omitido durante la comparación. Esto permite, por ejemplo, utilizar huellas difusas en formatos de archivo o diseños de archivo que de otra manera no permitirían su aplicación. Un ejemplo sería la creación de huellas ejecutables difusas para ejecutables empacados UPX al limitar los bloques cubiertos bloques a aquellos que pertenecen a su sección de recurso. La porción del archivo 410 se ha dividido en una pluralidad de bloques 412. El tamaño de cualquier bloque particular dentro de la pluralidad de bloques 412 no se limita a ningún tamaño particular. En diversas realizaciones, el tamaño de cualquier bloque particular dentro de la pluralidad de bloques 412 se determina mediante la ubicación del bloque dentro del archivo. En más realizaciones, una o más de la pluralidad de bloques 412 están en una ubicación diferente y pueden ser partes diferentes de la porción, del archivo 410 como uno o más de otros bloques en la pluralidad de bloques 412, y pueden proporcionar un tamaño diferente de acuerdo con esta ubicación diferente dentro de la porción del archivo 410. Por lo tanto, diferentes bloques o grupos de bloques dentro de la pluralidad de bloques 12 puede tener diferentes tamaños de bloques.

En diversas realizaciones, el tamaño de bloque depende del tamaño de archivo. En diversas realizaciones, se utilizan tamaños de bloques más pequeños para archivos pequeños, y se utilizan tamaños de bloques más grandes para archivos grandes. En diversas realizaciones, el tamaño de bloque también cambia dinámicamente a través de la porción 410, de acuerdo con su ubicación en el archivo. En diversas realizaciones, se utilizan tamaños de bloques más pequeños en la porción 410 en la sección de código más importante, mientras que se utilizan tamaños de bloques más grandes en la porción 410 para las áreas que solo incluyen datos y secciones de recurso del archivo 402.

En diversas realizaciones una ponderación se determina para cada una de la pluralidad de bloques 412 dentro de la porción 410. La ponderación para cualquier bloque dado dentro de la pluralidad de bloques 412 no se limita a ningún valor particular. En diversas realizaciones, la ponderación depende de una ubicación particular del bloque dentro del archivo 402, esto es, depende de cómo los datos importantes en los que la ubicación es para el formato de archivo dado y/o cómo los datos indicativos en la ubicación son para un archivo que es, o incluye, malware.

En diversas realizaciones, los archivos de entrada apropiados para tener una huella ejecutable difusa generada para el archivo se discuten en bloques, con cada tamaño de bloque y la ponderación depende de su ubicación en el archivo. Se utiliza la entropía de información de bloque ya que es indicativa de su complejidad.

En diversas realizaciones, para un valor de complejidad 429 se determina cada una de la pluralidad de bloques 412. En diversas realizaciones, se utiliza cada entropía de información de bloque para determinar un valor de complejidad 429 para un bloque dado. En diversas realizaciones, cada uno de los bloques entropía de información se calcula como:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

Donde

H(X) se basa en la información, contenido o auto- información de X, que es en sí misma una variable arbitraria; y

$p(x_i) = Pr(X=x_i)$ es la función de probabilidad de masa de X.

y se calcula con x = {0..255}, $x_i = i-1$ y n = |x| En diversas realizaciones, el bloque de datos es un conjunto de valores de 8-bit. Sin embargo, bloques no se limitan a ningún tamaño de palabra particular utilizado para calcular sus aproximaciones de complejidad, tales como bytes, palabras de 16-bit o 32-bit, y pueden utilizar diferentes tamaños de palabra por bloque en partes diferentes del archivo.

Al calcular un valor de complejidad para cada una de la pluralidad de bloques 412, una huella ejecutable difusa se crea para el archivo 402. La huella ejecutable difusa se considera que es difusa porque no tiene una representación exacta del archivo 402. Se espera que una “huella” de un archivo sea única, por ejemplo dos archivos diferentes no deben producir la misma huella - como, por ejemplo, la suma de control MD5 del archivo. Dichas huellas se adecuan perfectamente para detectar exactamente el mismo archivo de nuevo, pero se diseñan de forma explicita para que no sean ambiguas para los diferentes archivos - por ejemplo no son aplicables para detectar ligeras variantes del archivo de huella. Por supuesto, la detección de firmas es una solución bien conocida para la detección de malware, pero en contraste con las huellas, el malware tiene que ser analizado manualmente y la firma se tiene que crear manualmente, lo que hace más costoso (especialmente debido a la tremenda carga de nuevas variante de malware entrantes) la creación automática de huellas.

La ventaja de la huella ejecutable difusa es que si la huella ejecutable difusa se hace para un archivo, por ejemplo el archivo 402, esto es un archivo de malware conocido, la huella ejecutable difusa para el archivo 402 se puede utilizar para detectar variantes del malware, representado por o incluido en el archivo 402, en donde la variante del archivo 402 no es exactamente el mismo como el archivo 402, y de esta manera no sería detectado por una comparación de huella, pero se puede detectar como un archivo de malware variante al comparar una huella ejecutable difusa del archivo 402 con una huella ejecutable difusa de un archivo sospechoso.

La Figura 5 ilustra un diagrama 500 que incluye una comparación de dos huellas ejecutables difusas desde dos archivos diferentes. El diagrama 500 incluye un primer archivo 510 y un segundo archivo 550. En diversas realizaciones, el archivo 510 es una porción de un archivo 502, y el archivo 550 es una porción del archivo 552.

En diversas realizaciones, el primer archivo 510 incluye una pluralidad de bloques 512, que incluye bloques 520, 522, 224, 526, y 528 en la primera fila de la pluralidad de bloques 512. El número de filas de bloques en la pluralidad de bloques 512 no se limita a un número particular de filas, y se indica por las líneas discontinuas 530. El tamaño de los bloques en la pluralidad de bloques no se limita a ningún tamaño particular y puede ser cualquier tamaño como se describe aquí. En diversas realizaciones, el primer archivo 510 es un archivo conocido por ser malware, y para el cual se ha generado una huella ejecutable difusa, que incluye un valor de complejidad y una ponderación para cada uno de los bloques dentro de la pluralidad de bloques 512.

En diversas realizaciones, el segundo archivo 550 incluye una pluralidad de bloques 554, que incluye bloques 560, 562, 264, 566, y 568 en la primera fila de la pluralidad de bloques 554. El número de filas de bloques en la pluralidad de bloques 512 no se limita a un número particular de filas, y se indica por líneas discontinuas 580. Sin embargo, se utilizarán un número de filas incluidas en el segundo archivo 550 que corresponden al número de filas utilizadas en el primer archivo 510 con el fin de proporcionar un número casi correspondiente de bloques en los que se hacen una comparación entre el primer archivo 510 y el segundo archivo 550.

En diversas realizaciones, el tamaño de cada bloque en el segundo archivo 550 se realizará del mismo tamaño como un bloque correspondiente en el primer archivo 510. Por vía de ilustración, el tamaño de bloque 560 en el segundo archivo 550 tendrá el mismo tamaño como el bloque 520 en el primer archivo 510, el tamaño del bloque

562 en el segundo archivo 550 tendrá el mismo tamaño como el bloque 522 en el primer archivo 510, el tamaño del bloque 564 en el segundo archivo 550 tendrá el mismo tamaño como el bloque 524 en el primer archivo 510, el tamaño del bloque 566 en el segundo archivo 550 tendrá el mismo tamaño como el bloque 520 en el primer archivo 510, y el tamaño del bloque 568 en el segundo archivo 550 tendrá el mismo como el tamaño del bloque 528 en el primer archivo 510.

En diversas realizaciones, no se requiere que el número de bloques en el segundo archivo 550 es exactamente el mismo como el número de bloques en el primer archivo 510. Una diferencia de uno o más bloques con base por ejemplo en el número de bloques en un porción particular del primer archivo 510 y una porción correspondiente del segundo archivo 550 puede no resultar en el mismo número exacto de bloques en los dos archivos, pero no obstante no evitará la comparación entre los dos archivos desde que se realiza.

Una huella ejecutable difusa se genera para cada uno del primer archivo 510 y segundo archivo 550. La huella ejecutable difusa para el primer archivo 510 incluye un valor de complejidad y una ponderación para cada una de la pluralidad de bloques 512 en el primer archivo 510. En diversas realizaciones, la huella ejecutable difusa para el primer archivo 510 incluye información meta acerca del primer archivo 510, que incluye pero no se limita a un tamaño de archivo y un tipo de medios para el primer archivo 510. La huella ejecutable difusa para el segundo archivo 550 incluye un valor de complejidad y una ponderación para cada una de la pluralidad de bloques 554 en el segundo archivo 550. En diversas realizaciones, la huella ejecutable difusa para el segundo archivo 550 incluye información meta sobre el segundo archivo 550, que incluye pero no se limita a un tamaño de archivo y un tipo de medios para el segundo archivo 550.

Utilizando las huellas ejecutables difusas para el primer archivo 510 y el segundo archivo 550, se hace una comparación en forma bloque de las huellas ejecutables difusas del primer archivo 510, conocidas por ser malware, con la huella ejecutable difusa del segundo archivo 550. En diversas realizaciones, una comparación en forma de bloque inicia en un bloque particular en el primer archivo 510 y en un bloque particular en el segundo archivo 550.

En diversas realizaciones, la huella del primer archivo 510 utilizada en la comparación incluye una secuencia de bytes mágicos que debe existir en la misma desviación dada en el segundo archivo 550. En diversas realizaciones, la secuencia de bytes mágicos incluye comodines. En diversas realizaciones, la secuencia de bytes mágicos del primer archivo 510 está en o cerca del inicio del archivo 510.

En diversas realizaciones una comparación en forma de bloque incluye comparar un bloque desde el primer archivo 510 a una desviación de archivo dada a un bloque en el segundo archivo 550 en la misma desviación de archivo en el segundo archivo 550. En diversas realizaciones, se determina la similitud o disimilitud entre los valores de complejidad de los bloques comparados. Por ejemplo, el valor de complejidad del primer bloque desde el primer archivo 510 se compara con el valor de complejidad para los bloques comparados desde el segundo archivo 550. En diversas realizaciones, si el porcentaje diferente en el valor de complejidad está dentro de más o menos un valor de umbral predeterminado N, tal como un porcentaje de umbral, entonces se dice que los bloques comparados son similares. Por otro lado, si la diferencia de porcentaje en el valor de complejidad es más que más o menos el valor de umbral predeterminado N, se dice que los bloques comparados son disímiles.

Por vía de ilustración, el bloque 520 en el primer archivo 510 que se compara con el bloque 560 es el segundo archivo 550 que utiliza un valor de umbral de 90 por ciento. Si el valor de complejidad para el bloque 520 es 3.5, y el valor de complejidad para el bloque 560 es 3.7, el valor de porcentaje comparado de 3.5 dividido por 3.7 es aproximadamente 0.95, o 95 por ciento. Utilizando un valor de umbral de 90 por ciento, este porcentaje comparado de 95 por ciento es mayor que el valor de umbral predeterminado de 90 por ciento. Debido a que el valor de porcentaje comparado calculado para los bloques comparados excede el valor de umbral, los bloques comparados se considerarían “similares”.

Si el valor de complejidad del bloque 524 es 3.5, y el valor de complejidad del bloque comparado 564 es 4.0, el valor de porcentaje comparado de 3.5 dividido por 4.0 es aproximadamente 0.87, o 87 por ciento. De nuevo utilizando un valor de umbral de 90 por ciento, este porcentaje comparado de 87 por ciento es menor que el valor de umbral predeterminado de 90 por ciento. Debido a que el valor de porcentaje comparado calculado para los bloques comparados es menor que el valor de umbral, los bloques comparados se consideraría “disímiles.”

Como se ilustra en la Figura 5, el bloque 520 en el primer archivo 510 se compara con el bloque 560 en el segundo archivo 550, y se determina que es similar, como se indica por la designación “SIMILAR 1” incluida en cada uno de estos bloques. Como también se muestra, el bloque 522 en el primer archivo 510 se compara con el bloque 562 en el segundo archivo 550, y se determina que es similar, como se indica por la designación “SIMILAR 2” incluida en cada uno de estos bloques. Sin embargo, cuando el bloque 524 en el primer archivo 510 se compara con el bloque 564 en el segundo archivo 550, se determina que los bloques son disímiles, y se indican por la indicación “DISSIMILAR 1” en el bloque 524.

En diversas realizaciones, una comparación en forma de bloque entre archivos no se limita a una comparación de bloques en las mismas desviaciones exactas en los archivos que se comparan. En diversas realizaciones, una comparación en forma de bloque entre archivos incluye comparar un bloque en un primer archivo con uno o más bloques ubicados cerca del bloque correspondiente que tiene la misma desviación en el segundo archivo. Continuando con el ejemplo de la Figura 5, aunque se determina que el bloque 524 en el primer archivo 510 es disímil del bloque 564 en el segundo archivo 550, una comparación del bloque 526, el siguiente bloque en el primer archivo 510, con el bloque 564 del segundo archivo 550 resulta en una determinación de que estos bloques son similares. Esto se representa por la flecha 573 y la indicación “SIMILAR 3” incluida en los bloques 526 y 564.

Continuando con la comparación, el bloque 528 del primer archivo 510 se compara con el bloque 566 del segundo archivo 550, como se representa por la flecha 574. Como se muestra, utilizando la comparación, se determina que los bloques 528 y 566 son similares, como se ilustra por la indicación “SIMILAR 4” en los bloques 528 y 566. Así, una comparación en forma de bloque incluye comparar bloques desde el primer archivo 510 hasta los bloques del segundo archivo 550 en donde los bloques del segundo archivo no están necesariamente en la misma desviación o en una misma posición dentro de los bloques como el bloque utilizado en la comparación del primer archivo 510.

Las variaciones en los patrones de comparación no se limita a ningún patrón de comparación particular o ningunas variaciones particulares de los patrones de comparación entre los bloques en el primer archivo y los bloques en el segundo archivo. Las variaciones en los patrones en forma de bloque incluyen cualquier número de variación o esquemas que se determinan por ser apropiados para la detección de malware en archivos comparados.

Se realiza una comparación en forma de bloque al calcular un valor de similitud para los dos bloques de datos comparados. En diversas realizaciones, para cualquiera de los dos bloques la probabilidad de similitud de los bloques se calcula como la desviación real de ambos valores de complejidad de bloques en relación con una desviación máxima posible por un factor de proximidad para la comparación entre la desviación i en un primer archivo x_1 y la desviación j en un segundo archivo x_2, en donde:

$$d_{\max} = 2 \cdot N\% \cdot \max(H(b(x_1, i)), H(b(x_2, j)))$$

y

$$s(x_1, i, x_2, j) = \frac{d_{\max} - |H(b(x_1, i)) - H(b(x_2, j))|}{d_{\max}} \cdot \frac{\max(|x_1|, |x_2|) - |i - j|}{\max(|x_1|, |x_2|)}$$

En diversas realizaciones, la probabilidad de similitud se multiplica por la ponderación de los dos bloques. En diversas realizaciones, la ponderación sería 1 por defecto, y caería por debajo de 1 para bloques menos importantes, y se elevaría por encima de 1 para bloques importantes. Por vía de ilustración, los bloques para una sección de código de un archivo tendrían una ponderación de 1.2.

En la mayoría de las realizaciones, la probabilidad de similitud también tiene en cuanta una distancia es decir la diferencia entre la desviación de archivo de ambos bloques, denominados como un factor de proximidad anterior.

Las probabilidades de similitud individual se resumen utilizando la fórmula de Bayes para generar una probabilidad de similitud total. La fórmula de Bayes relaciona las probabilidades condicional y marginal de eventos estocásticos A y B:

$$\begin{aligned} \Pr(A|B) &= \frac{\Pr(B|A)\,\Pr(A)}{\Pr(B)} \\ &\propto L(A|B)\,\Pr(A) \end{aligned}$$

donde L(A|B) es la probabilidad de A dado B fijo. Nótese la relación: Cada término en la fórmula de Bayes tiene un nombre adicional:

$$\Pr(B|A) = L(A|B)$$

- Pr(A) es la probabilidad anterior o probabilidad marginal de A. Es “anterior” en el sentido de que no se tiene en cuenta ninguna información acerca de B.

- Pr(A|B) es la probabilidad condicional de A, dado B. También se denomina la probabilidad posterior debido a que se deriva de o depende del valor específico de B.

- Pr(B|A) es la probabilidad condicional de B dado A.

- Pr(B) es la probabilidad anterior o marginal de B, y actúa como una constante de normalización.

La probabilidad de similitud total proporciona una probabilidad de porcentaje de que un segundo archivo es una variante de un primer archivo. En diversas realizaciones, un usuario puede comparar la probabilidad de similitud total con un valor de umbral M para determinar si el segundo archivo se considera malware. Por ejemplo, si una probabilidad de similitud total para un segundo archivo iguala o excede un valor de umbral M, se puede determinar que el segundo archivo es una variante del primer archivo. Si el primer archivo es una malware conocido, luego se puede determinar que el segundo archivo es malware con base en la probabilidad de similitud total y el valor de umbral M. Por otra parte, si una probabilidad de similitud total para un segundo archivo cuando se compara con un primer archivo no excede el valor de umbral M, se hace una determinación de que el segundo archivo no es una variante para el primer archivo.

En algunas realizaciones, se puede utilizar un segundo umbral X, con X<M, de tal manera que los archivos con una probabilidad de similitud menor que X se consideran que no son una variante de malware, los archivos con probabilidad mayor que M se consideran una variante de malware, y aquellos entre X y M se consideran sospechosos y se advierte a los usuarios.

Un patrón como se muestra en la Figura 5 puede ser representativo del código insertado en el bloque 524, que puede ser el código insertado como malware. Además el bloque 524 puede haber sido retirado del archivo 550 en un intento de ocultar el archivo 550 de la detección como malware en casos donde el código 524 se conoce por ser malware y se utiliza para examinar los programas utilizados para detectar malware. Al retirar o cambiar el código en el bloque 524, un autor malware puede haber intentado derrotar la detección de una variante del malware ahora presente en el archivo 550. Una Técnica diferente que se basa en una huella exacta para archivos comparados, tal como una técnica de detección de suma de control, podría no detectar este segundo archivo como un mismo primer archivo de malware y archivo de malware conocido, y esto puede fallar en proporcionar una indicación de que de que el segundo archivo es una variante del primer y conocido archivo de malware. Al utilizar una huella ejecutable difusa, y al utilizar una o más variaciones de las comparaciones en forma de bloque, es detectable una variante de un archivo conocido que incluye malware.

La Figura 6 ilustra un diagrama que incluye una representación de un primer archivo 610 y una representación de un segundo archivo 650. Se muestra que el primer archivo 610 tiene una pluralidad de bloques que incluye filas 614 y columnas 616. Se muestra que el segundo archivo 650 tiene una pluralidad de bloques incluidos en filas 654 y columnas 656. El primer archivo 610 y el segundo archivo 650 tienen una estructura similar de bloques, en donde el primer archivo 610 tiene diez columnas de bloques, y el segundo archivo 650 tiene diez columnas de bloques. El primer archivo 610 tiene la fila quince de bloques completa, que incluyen bloques 621-630 en la fila más superior, y una fila dieciséis incompleta que incluye seis bloques 641-646. El segundo archivo 650 tiene la fila quince de bloques completa, que incluyen bloques 661-670 en la fila más superior, y una fila diecisiete incompleta que incluye cinco bloques 681-655. Como se muestra en la Figura 6, el segundo archivo 650 tiene un bloque menos que el primer archivo 610.

Cada bloque en el primer archivo 610 tiene un valor de complejidad 612 asociado con el bloque y representativo de un valor de complejidad calculado para el bloque dado. Cada bloque en el segundo archivo 650 tiene un valor de complejidad 652 asociado con el bloque y representativo de un valor de complejidad calculado para el bloque dado. Estos valores de complejidad se pueden utilizar para comparar el primer archivo 610, que se considera es ilustrativo de malware conocido, con el segundo archivo 650. Como se ilustra en la Figura 6, en muchos casos el valor de complejidad para un bloque en el primer archivo 610 está cercano en valor a un valor de complejidad para un bloque correspondiente en una misma, o cercana, posición de segundo archivo 650. Por vía de ilustración, el bloque 621 en el primer archivo 610 tiene un valor de complejidad de 3.70, en donde el bloque 661 en el segundo archivo 650 tiene un valor de complejidad de 3.82. Dependiendo de la configuración utilizada para el valor de umbral, se puede determinar que los bloques 621 y 661 son bloques similares.

Un mismo resultado, que de nuevo depende del valor de umbral utilizado en la comparación, se puede obtener al comparar los bloques 622-630 en la fila más superior de primer archivo 610 con los bloques 662-670 en la fila más superior del segundo archivo 650. En contraste, el bloque 690 en el primer archivo 610 tiene un valor de complejidad de 4.36, en donde el bloque correspondiente 695 en el segundo archivo 650 tiene un valor de complejidad de 1.50. Esta discrepancia en valores puede representar un bloque disímil, y puede indicar una modificación, una inserción, o una eliminación del código de programa en el segundo archivo 650 cuando se compara con el primer archivo 610. En otras posibles comparaciones, los bloques 691, 692 y 693 en el primer archivo 610 tienen valores de complejidad de 4.32, 4.25, y 4.65 respectivamente, en donde los bloques correspondientes 696, 697, y 698 tienen valores de

complejidad de 2.51, 2.08, y 1.07 respectivamente. De nuevo dependiendo del valor de umbral que se utiliza, estos conjuntos de bloques correspondientes de deben determinar disímiles.

Después de calcular una probabilidad de similitud como resultado de una comparación de huella ejecutable difusa entre estos archivos, se puede calcular una probabilidad de similitud total. Al comparar la probabilidad de similitud total generada con el valor de umbral para M, se puede hacer una determinación en cuanto a si el segundo archivo 650 es una variante de un archivo de malware conocido representado por archivo 610.

La generación de las huellas ejecutables difusas no requiere ningún análisis manual, y no requiere ninguna creación de firma de regla de detección con el fin de detectar nuevas variantes del malware conocido. Además, el uso de huellas ejecutables difusas genera pocos falsos positivos que otros métodos no exactos, tales como los métodos de detección heurística.

En diversas realizaciones, si se cumplen ciertos parámetros durante un proceso de comparación utilizando huellas ejecutables difusas, los bloques completos del archivo no necesitan ser comparados con el fin de determinar que un archivo que se compara con una huella del archivo de malware conocido no es una variante del archivo de malware conocido.

La Figura 7 ilustra el diagrama 700 que incluye una representación de un archivo 710. Se muestra que el archivo 710 tiene una pluralidad de bloques incluidos en filas 714 y columnas 716. EL archivo 710 Tiene una estructura similar de bloques al primer archivo 610 en la Figura 6, en donde el archivo 710 tienen diez columnas de bloques, y el primer archivo 610 tiene diez columnas de bloques. El archivo 710 tiene la fila quince de bloques completa, que incluye los bloques 721-730 en la fila más superior, y una fila dieciséis incompleta que incluye tres bloques 741-743. El primer archivo 610 tiene la fila quince de bloques completa, que incluye los bloques 621-630 en la fila más superior, y una fila diecisiete incompleta que incluye seis bloques 641-646.

Como se muestra en la Figura 7, cada uno de los bloques en el archivo 710 tiene un valor de complejidad 712. En el archivo 710, los bloques 723-730 todos incluyen un valor de complejidad de 0.00. Los bloques correspondientes en el primer archivo 610, que incluye los bloques 623 a 630, que corresponden por lo menos en posición en el archivo a los bloques 723-730, incluyen valores de complejidad de 4.62, 4.67, 4.55, 4.53, 395, 3.60, 4.11, y 4.53 respectivamente. En la comparación los valores de complejidad de bloques 723-730 con los bloques 623-630, una determinación “disímil” que resulta de cada una de estas comparaciones podría resultar en ocho comparaciones en una fila que es disímil.

En diversas realizaciones, un valor de umbral Z se puede utilizar para determinar si ha ocurrido un número parcial de comparaciones que tienen un resultado de “disímil” en una fila durante cualquier comparación dada de huellas ejecutables difusas entre dos archivos. Si ocurre que el número de comparaciones que tiene un resultado de disímil en una fila iguala o excede el valor de umbral Z, el proceso de comparación se puede finalizar, y se puede hacer una determinación que el archivo probado, tal como archivo 710, no es una variante del archivo conocida como malware, tal como el archivo 610.

Esta comparación terminada con base en el número de comparaciones disímiles encontradas en una fila reducirían el tiempo utilizado para hacer una comparación, mientras que aún se hace una determinación de que el archivo que se prueba no es una variante de un archivo dado conocido por ser malware. El tiempo ahorrado se puede utilizar para comparar el archivo que se prueba con otras huellas ejecutables difusas de archivos de malware conocidos, y así se reduce el tiempo requerido para proba un archivo contra un conjunto de huellas de archivos conocidos por ser malware. Como se incrementa el número de archivos conocidos por ser malware, el tiempo ahorrado se hace más importante en ser capaz de procesar archivos entrantes que se van a probar en una forma oportuna.

La Figura 8 ilustra un diagrama de flujo de un método 800 de acuerdo con diversas realizaciones.

En el bloque 810, el método 800 incluye crear una primera huella difusa de un archivo de malware conocido. En diversas realizaciones, el bloque 810 incluye la primera huella difusa que incluye un primer conjunto de aproximaciones de complejidad calculadas y ponderaciones para cada una de una pluralidad de bloques dentro del archivo de malware conocido;

En el bloque 820, el método 800 incluye crear una segunda huella difusa de un archivo que se va a verificar. En diversas realizaciones, el bloque 820 incluye la segunda huella difusa que incluye un segundo conjunto de aproximaciones de complejidad calculadas y ponderaciones para cada una de una pluralidad de bloques dentro del archivo que se va a verificar;

En el bloque 830 el método 800 incluye comparar la segunda huella difusa to la primera huella difusa. En diversas realizaciones, el bloque 830 incluye comparar las aproximaciones de complejidad calculadas desde la segunda

huella difusa con una pluralidad de las aproximaciones de complejidad desde la primera huella difusa utilizando una comparación en forma de bloque; y

En el bloque 840, el método 800 incluye calcular una probabilidad de similitud para cada una de las comparaciones en forma de bloque. En diversas realizaciones, el bloque 840 incluye el cálculo que incluye una ponderación respectiva para cada una de la pluralidad de bloques dentro del archivo de malware conocido y para cada una de la pluralidad de bloques dentro del archivo que se va a verificar, y el cálculo incluye una distancia entre los bloques comparados; y

En el bloque 850, el método 800 incluye calcular una probabilidad de similitud total para la pluralidad de bloques comparada.

Diversas realizaciones del método 800 incluyen detener las comparaciones en forma de bloque sí un número de umbral de comparaciones en forma de bloque en una fila cae por debajo de un valor de umbral N para la probabilidad de similitud calculada.

Diversas realizaciones del método 800 incluyen calcular una aproximación de complejidad para cada una de una pluralidad de bloques con el archivo de malware conocido y calcular una aproximadamente de complejidad para cada una de la pluralidad de bloques dentro del archivo que se va a verificar que incluye calcular la entropía de información para un bloque dado utilizando la fórmula:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

Donde

H(X) se basa en el contenido de información o auto- información de X, que es en sí misma una variable arbitraria;

$p(x_i) = Pr(X=x_i)$ es la función de probabilidad de masa de X; y

y se calcula con x = {0..255}, $x_i = i-1$ y $n = |x|$.

Diversas realizaciones del método 800 incluyen en donde calcular un valor de similitud para cada una de las comparaciones en forma de bloque incluidas para cualquiera de los dos bloques dados que se comparan, calcular una desviación real de ambos valores de complejidad de bloques en relación con una desviación máxima posible por un factor de proximidad para la comparación entre un desplazamiento i en un primer archivo x_1 y un desplazamiento j en un segundo archivo x_2, en donde:

$$d_{max} = 2 \cdot N\% \cdot \max(H(b(x_1, i)), H(b(x_2, j)))$$

y

$$s(x_1, i, x_2, j) = \frac{d_{max} - |H(b(x_1, i)) - H(b(x_2, j))|}{d_{max}} \cdot \frac{\max(|x_1|, |x_2|) - |i - j|}{\max(|x_1|, |x_2|)}.$$

Diversas realizaciones del método 800 incluyen sumar utilizando la fórmula de Bayes para generar una probabilidad de similitud total, en donde la fórmula de Bayes relaciona las probabilidades condicional y marginal de eventos estocásticos A y B:

$$\begin{aligned} \Pr(A|B) &= \frac{\Pr(B|A)\Pr(A)}{\Pr(B)} \\ &\propto L(A|B)\Pr(A) \end{aligned}$$

donde L(A|B) es la probabilidad de A dando fijo a B.

Diversas realizaciones del método 800 incluyen en donde calcular la probabilidad de similitud total para la pluralidad de bloques comparada que incluye comparar la probabilidad de similitud total con un valor de umbral M para determinar si el segundo archivo se considera malware.

Diversas realizaciones del método 800 incluyen determinar que el archivo que se va a verificar es una variante del archivo de malware conocido cuando la probabilidad de similitud calculada total iguala o excede el valor de umbral M.

Diversas realizaciones del método 800 incluyen en donde calcular la probabilidad de similitud total para la pluralidad de bloques comparada que incluye comparar la probabilidad de similitud total con un valor de umbral X y un valor de umbral M para determinar si el segundo archivo se considera malware o se considera un archivo sospechoso, en donde el valor de umbral M es mayor que el valor de umbral X.

Diversas realizaciones del método 800 incluyen en donde cuando la probabilidad de similitud calculada total es menor que el valor de umbral X el archivo que se va a verificar se determina no es una variante del archivo de malware conocido, y cuando la probabilidad de similitud calculada total es mayor que el valor de umbral M el archivo que se va a verificar se determina es una variante del archivo de malware conocido, y el archivo que se va a verificar se considera un archivo sospechoso cuando la probabilidad de similitud calculada total es mayor que el valor de umbral X pero menor que o igual a el valor de umbral para M.

La Figura 9 ilustra un diagrama de flujo de un método 900 de acuerdo con diversas realizaciones.

En el bloque 910, el método 900 incluye almacenar por lo menos un conjunto de formación una pluralidad de archivos conocidos por ser malware;

En el bloque 920, el método 900 incluye generar para cada archivo de la pluralidad de archivos conocidos por ser malware una huella ejecutable difusa. En diversas realizaciones, el bloque 920 incluye cada huella ejecutable difusa que incluye un primer conjunto de aproximaciones de complejidad calculadas y ponderaciones para cada una de una pluralidad de bloques dentro de cada una de las pluralidades individuales de la pluralidad de archivos conocidos por ser malware; y

En el bloque 930, el método 900 incluye proporcionar a uno o más motores anti-malware las huellas generadas ejecutables difusas para cada archivo de la pluralidad de archivos.

Diversas realizaciones del método 900 incluyen recibir desde una cualquiera de por lo menos uno de los motores anti-malware una variante detectada de uno de los archivos conocidos por ser malware, y actualizar por lo menos un conjunto de formación para incluir la variante detectada de uno de los archivos conocidos por ser malware.

Diversas realizaciones del método 900 incluye generar una huella ejecutable difusa para la variante detectada de uno de los archivos conocidos por ser malware y actualizar cada uno de uno o más motores anti-malware que incluyen la huella ejecutable difusa generada para la variante detectada de uno de los archivos conocidos por ser malware.

Diversas realizaciones del método 900 incluyen en donde proporcionar uno o más motores anti-malware de la huella ejecutable difusa generada que incluye un proveedor del software anti-malware que suministra una versión actualizada de la base de datos de huellas ejecutables difusas generadas.

Se han descrito aquí diversas realizaciones de sistemas, aparatos, y métodos de detección de malware.

Diversas realizaciones incluyen un aparato que comprende una puerta de enlace que incluye un motor anti-malware acoplada a una base de datos de huellas difusas generadas que incluye una pluralidad de huellas para archivos de malware conocidos, un generador de huellas difusas acoplado al motor anti-malware, el generador de huellas difusas operable para producir una huella difusa que incluye una aproximación de complejidad para cada una de una pluralidad de bloques para un archivo proporcionado por el motor anti-malware, y un comparador de huella acoplado al motor anti-malware, el comparador de huella operable para comparar una huella producida desde el generador de huella con una cualquiera de la pluralidad de huellas para la base de datos de huellas difusas generadas y para producir una probabilidad de similitud sobre una base en forma de bloque.

Diversas realizaciones incluyen un sistema que comprende una pluralidad de dispositivos protegidos acoplada a red a través de una puerta de enlace, la puerta de enlace incluye un motor anti-malware, una base de datos de huellas generadas difusas acopladas al motor antimalware, la base de datos de huella generada incluye una pluralidad de huellas para archivos de malware conocidos acopladas al motor anti-malware, un generador de huellas difusas acoplado al motor anti-malware, el generador de huellas difusas operable para producir una huella ejecutable difusa que incluye una aproximación de complejidad para cada una de una pluralidad de bloques en un archivo suministrado por el motor anti-malware, y un controlador de huellas difusas acoplado al motor anti-malware, el controlador de huellas difusas operable para comparar una huella ejecutable difusa producida desde el generador de

huella con una cualquiera de any one of la pluralidad de huellas desde la base de datos de huella generada y para producir una probabilidad de similitud sobre una base bloque por bloque.

Aunque se han ilustrado y descrito aquí las realizaciones específicas, se apreciará por aquellos medianamente expertos en la técnica que cualquier disposición que se calcula para lograr el mismo propósito se puede sustituir para la realización específica mostrada. Esta solicitud está destinada a cubrir cualesquier adaptaciones o variaciones de la presente invención. Por lo tanto, se entiende que esta invención se limita solo por las reivindicaciones y los equivalentes de las mismas.

REIVINDICACIONES

1. Un método para detectar posible malware que comprende:

crear una primera huella difusa de un archivo de malware conocido, la primera huella difusa incluye un primer conjunto de aproximaciones de complejidad calculadas y ponderaciones para cada una de una pluralidad de bloques dentro del archivo de malware conocido;

crear una segunda huella difusa de un archivo que se va a verificar, la segunda huella difusa incluye un segundo conjunto de aproximaciones de complejidad calculadas y ponderaciones para cada una de una pluralidad de bloques dentro del archivo que se va a verificar;

comparar la segunda huella difusa con la primera huella difusa que incluye comparar las aproximaciones de complejidad calculadas desde la segunda huella difusa con una pluralidad de las aproximaciones de complejidad desde la primera huella difusa utilizando una comparación en forma de bloque;

cálculos de complejidad utilizados para calcular la primera y segunda huellas difusas derivadas de la entropía de información del bloque al calcular la suma de una función de probabilidad de masa por un logaritmo de la función de probabilidad de masa del bloque datos,

las ponderaciones para la primera y segunda huellas difusas derivadas desde por lo menos una ubicación del bloque en su formato de archivo particular y cómo la información indicativa en la ubicación del bloque dentro del archivo es para indicar que el archivo es malware;

calcular una probabilidad de similitud para cada una de las comparaciones en forma de bloque, el cálculo que incluye una ponderación respectiva para cada una de la pluralidad de bloques dentro del archivo de malware conocido y para cada una de la pluralidad de bloques dentro del archivo que se va a verificar, y el cálculo que incluye una distancia entre los bloques comparados; y

calcular una probabilidad de similitud total para la pluralidad de bloques comparada para determinar si el archivo que se va a verificar se considera un malware.

2. El método de la reivindicación 1, que incluye detener las comparaciones en forma de bloque sí un número de umbral de comparaciones en forma de bloque en una fila cada una cae por debajo de un valor de umbral N para la probabilidad de similitud calculada.

3. El método de la reivindicación 1, calcular una aproximación de complejidad para cada una de una pluralidad de bloques dentro del archivo de malware conocido y calcular una aproximación de complejidad para cada una de la pluralidad de bloques dentro del archivo que se va a verificar incluye calcular la entropía de información para un bloque dado utilizando la fórmula:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

donde

H(X) se basa en el contenido de información o auto- información de X, que es en sí misma una variable arbitraria;

$p(x_i) = Pr(X=x_i)$ es la función de probabilidad de masa de X; y

y se calcula con x = {0..255}, $x_i = i - l$ y $n = |x|$.

4. El método de la reivindicación 1, en donde calcular un valor de similitud para cada una de las comparaciones en forma de bloque incluye para cualquiera de los dos bloques dados que se comparan, calcular una desviación real de ambos valores de complejidad de bloques en relación con una desviación máxima posible por un factor de proximidad para la comparación entre un desplazamiento i en un primer archivo x, y un desplazamiento j en un segundo archivo x_2, en donde:

$$d_{max} = 2 \cdot N\% \cdot \max(H(b(x_1, i)), H(b(x_2, j)))$$

y

$$s(x_1, i, x_2, j) = \frac{d_{max} - |H(b(x_1, i)) - H(b(x_2, j))|}{d_{max}} \cdot \frac{\max(|x_1|, |x_2|) - |i - j|}{\max(|x_1|, |x_2|)}.$$

5. El método de la reivindicación 4, que incluye sumar utilizando la fórmula de Bayes para generar una probabilidad de similitud total, en donde la fórmula de Bayes relaciona las probabilidades condicional y marginal de eventos estocásticos A y B:

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)} \propto L(A|B)\Pr(A)$$

donde L(A|B) es la probabilidad de A dando fijo a B.

6. El método de la reivindicación 1, en donde calcular la probabilidad de similitud total para la pluralidad de bloques comparada que incluye comparar la probabilidad de similitud total con un valor de umbral M para determinar si el segundo archivo se considera malware.

7. El método de la reivindicación 6, que incluye determinar que el archivo que se va a verificar es una variante del archivo de malware conocido cuando la probabilidad de similitud calculada total iguala o excede el valor de umbral M.

8. El método de la reivindicación 1, en donde calcular la probabilidad de similitud total para la pluralidad de bloques comparada que incluye comparar la probabilidad de similitud total con un valor de umbral X y un valor de umbral M para determinar si el segundo archivo se considera malware o se considera un archivo sospechoso, en donde el valor de umbral M es mayor que el valor de umbral X.

9. El método de la reivindicación 8, en donde cuando la probabilidad de similitud calculada total es menor que el valor de umbral X, se determina que el archivo que se va a verificar no es una variante del archivo de malware conocido, y cuando la probabilidad de similitud calculada total es mayor que el valor de umbral M, se determina que el archivo que se va a verificar es una variante del archivo de malware conocido, y el archivo que se va a verificar se considera un archivo sospechoso cuando la probabilidad de similitud calculada total es mayor que el valor de umbral X pero menor que o igual al valor de umbral para M.

10. Un aparato para detectar posible malware que comprende:

una puerta de enlace que incluye un motor anti-malware acoplada a una base de datos de huellas difusas generadas que incluye una pluralidad de huellas para archivos de malware conocidos;

un generador de huellas difusas acoplado al motor anti-malware, el generador de huellas difusas operable para producir una huella difusa que incluye una aproximación de complejidad y una ponderación para cada una de una pluralidad de bloques para un archivo proporcionado por el motor anti-malware; y

un comparador de huella acoplado al motor anti-malware, el comparador de huella operable para comparar una huella producida desde el generador de huella con una cualquiera de la pluralidad de huellas para la base de datos de huellas difusas generadas y para producir una probabilidad de similitud sobre una base en forma de bloque para determinar si el archivo suministrado se considera un malware

los cálculos de complejidad utilizados para calcular la huella difusa derivada de la entropía de información del bloque al calcular la suma de una función de probabilidad de masa por un logaritmo de la función de probabilidad de masa del bloque datos,

las ponderaciones utilizadas para calcular la huella difusa derivada de por lo menos una ubicación del bloque en su formato de archivo particular y cómo la información indicativa en la ubicación del bloque dentro del archivo es para indicar que el archivo es malware.

11. El aparato de la reivindicación 10, que produce una aproximación de complejidad para cada una de una pluralidad de bloques incluye calcular la entropía de información para un bloque dado utilizando la fórmula:

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)$$

donde

H(X) se basa en el contenido de información o auto- información de X, que es en sí misma una variable arbitraria;

$p(x_i) = Pr(X=x_i)$ es la función de probabilidad de masa de X; y

y se calcula con x = {0..255}, $x_i = i - 1$ y n = |x|.

12. El aparato de la reivindicación 10, en donde producir la probabilidad de similitud sobre una base en forma de bloque incluye para cualquiera de los dos bloques dados que se comparan, calcular una desviación real de ambos valores de complejidad de bloques en relación con una desviación máxima posible por un factor de proximidad para la comparación entre un desplazamiento i en un primer archivo x_1 y un desplazamiento j en un segundo archivo x_2, en donde:

$$d_{\max} = 2 \cdot N\% \cdot \max(H(b(x_1, i)), H(b(x_2, j)))$$

y

$$s(x_1, i, x_2, j) = \frac{d_{\max} - \left|H(b(x_1, i)) - H(b(x_2, j))\right|}{d_{\max}} \cdot \frac{\max(|x_1|, |x_2|) - |i - j|}{\max(|x_1|, |x_2|)}.$$

13. El aparato de la reivindicación 12, que incluye sumar utilizando la fórmula de Bayes para generar una probabilidad de similitud total, en donde la fórmula de Bayes relaciona las probabilidades condicional y marginal de eventos estocásticos A y B:

$$\begin{aligned} \Pr(A|B) &= \frac{\Pr(B|A)\,\Pr(A)}{\Pr(B)} \\ &\propto L(A|B)\,\Pr(A) \end{aligned}$$

donde L(A|B) es la probabilidad de A dando fijo a B.

14. El aparato de la reivindicación 10, que incluye un mecanismo de actualización operable para recibir versiones actualizadas de huellas difusas generadas para almacenamiento en la base de datos de huellas difusas generadas.

15. El aparato de la reivindicación 10, en donde la base de datos de huellas difusas generadas incluye una tabla de precondición que contiene firmas de bytes mágicos para todos los tipos de medios cubiertos por la base de datos.

16. El aparato de la reivindicación 10, en donde la base de datos de huellas difusas generadas incluye una tabla de salto operable para determinar un desplazamiento relativo en la base de datos de huellas difusas generadas con base en un tamaño de archivo del archivo que se va a verificar.

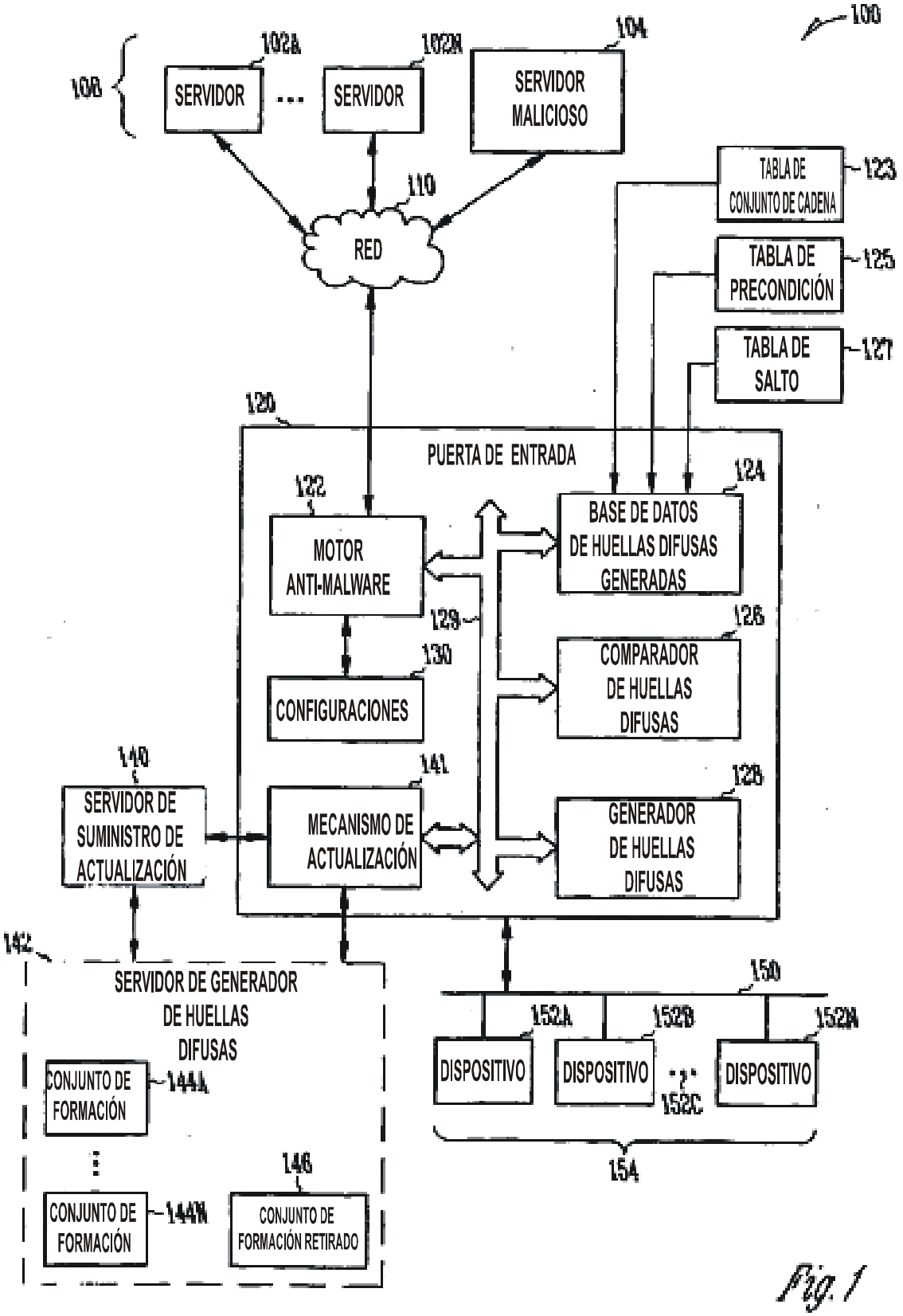

100
102A
102N
104
108
SERVIDOR
SERVIDOR
SERVIDOR MALICIOSO
110
RED
TABLA DE CONJUNTO DE CADENA
123
TABLA DE PRECONDICIÓN
125
TABLA DE SALTO
127
120
PUERTA DE ENTRADA
122
124
MOTOR ANTI-MALWARE
BASE DE DATOS DE HUELLAS DIFUSAS GENERADAS
129
126
COMPARADOR DE HUELLAS DIFUSAS
130
CONFIGURACIONES
140
141
128
SERVIDOR DE SUMINISTRO DE ACTUALIZACIÓN
MECANISMO DE ACTUALIZACIÓN
GENERADOR DE HUELLAS DIFUSAS
142
SERVIDOR DE GENERADOR DE HUELLAS DIFUSAS
150
152A
152B
152N
DISPOSITIVO
DISPOSITIVO
DISPOSITIVO
152C
CONJUNTO DE FORMACIÓN
144A
CONJUNTO DE FORMACIÓN
144N
146
CONJUNTO DE FORMACIÓN RETIRADO
154

Fig. 1

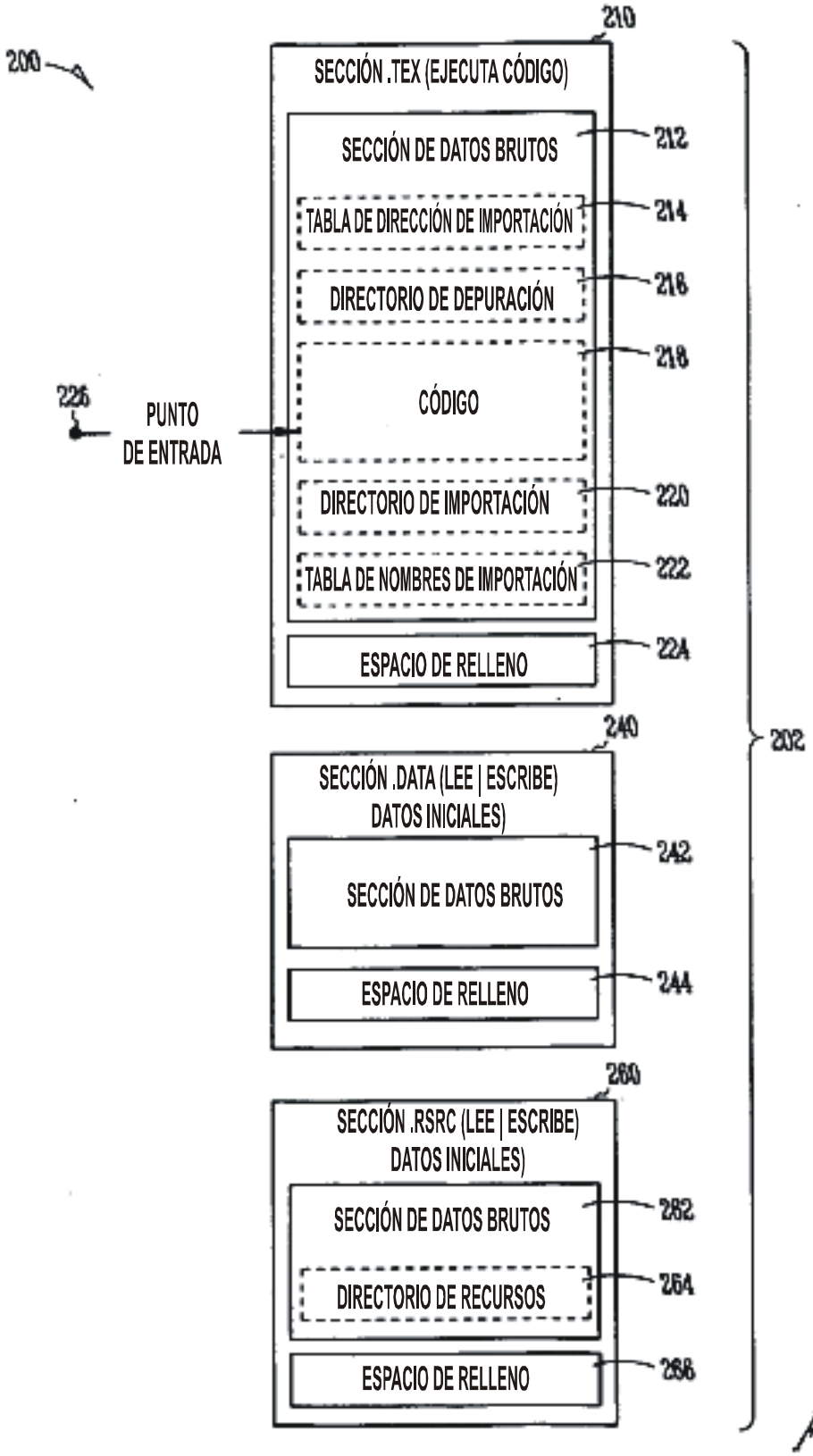

200
210
SECCIÓN .TEX (EJECUTA CÓDIGO)
SECCIÓN DE DATOS BRUTOS
212
TABLA DE DIRECCIÓN DE IMPORTACIÓN
214
DIRECTORIO DE DEPURACIÓN
216
CÓDIGO
218
226
PUNTO
DE ENTRADA
DIRECTORIO DE IMPORTACIÓN
220
TABLA DE NOMBRES DE IMPORTACIÓN
222
ESPACIO DE RELLENO
224
240
SECCIÓN .DATA (LEE | ESCRIBE)
DATOS INICIALES)
242
SECCIÓN DE DATOS BRUTOS
ESPACIO DE RELLENO
244
260
SECCIÓN .RSRC (LEE | ESCRIBE)
DATOS INICIALES)
SECCIÓN DE DATOS BRUTOS
262
DIRECTORIO DE RECURSOS
264
ESPACIO DE RELLENO
266
202

Fig.2

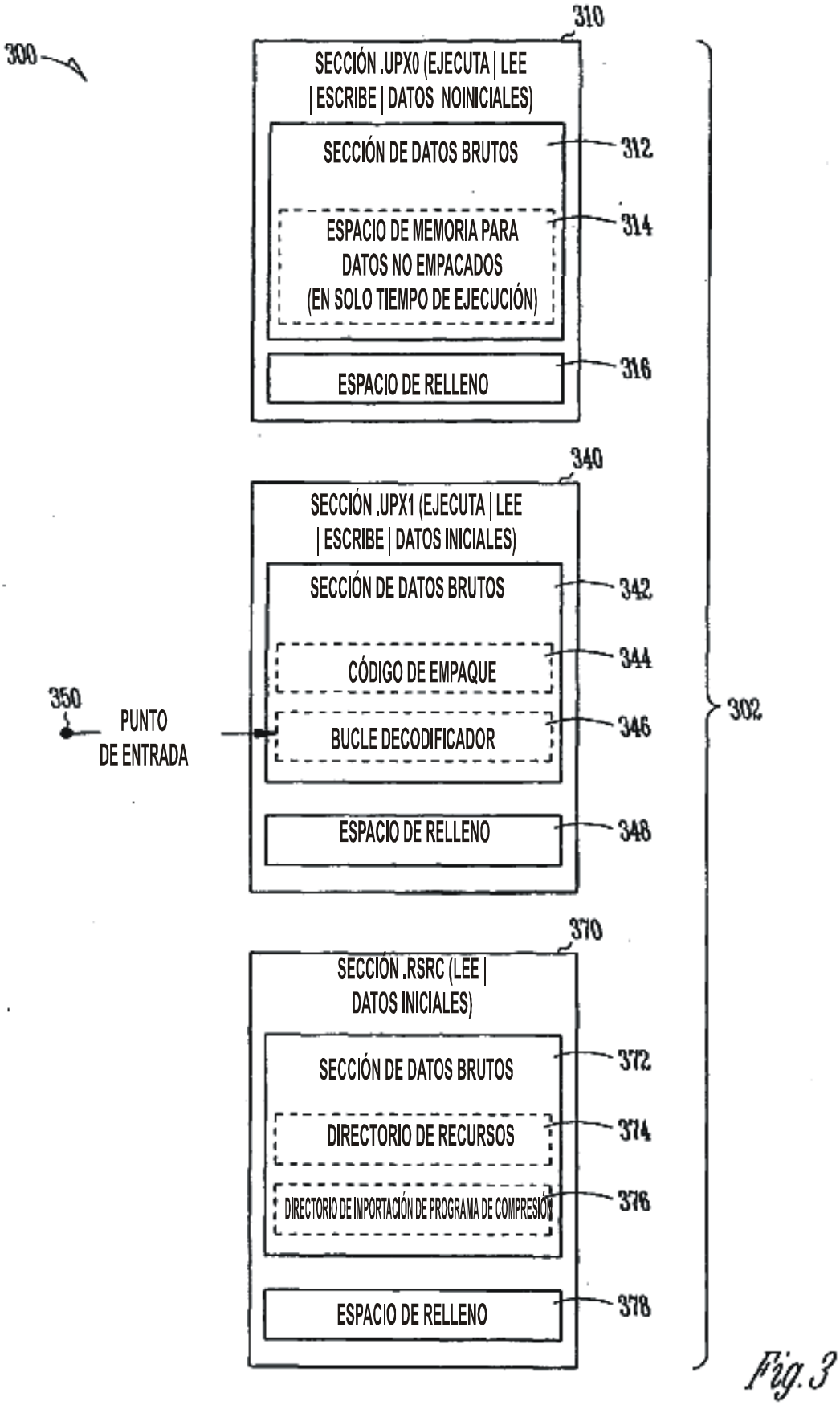

300
310
SECCIÓN .UPX0 (EJECUTA | LEE | ESCRIBE | DATOS NOINICIALES)
SECCIÓN DE DATOS BRUTOS
312
ESPACIO DE MEMORIA PARA DATOS NO EMPACADOS (EN SOLO TIEMPO DE EJECUCIÓN)
314
ESPACIO DE RELLENO
316
340
SECCIÓN .UPX1 (EJECUTA | LEE | ESCRIBE | DATOS INICIALES)
SECCIÓN DE DATOS BRUTOS
342
CÓDIGO DE EMPAQUE
344
350
PUNTO DE ENTRADA
BUCLE DECODIFICADOR
346
ESPACIO DE RELLENO
348
302
370
SECCIÓN .RSRC (LEE | DATOS INICIALES)
SECCIÓN DE DATOS BRUTOS
372
DIRECTORIO DE RECURSOS
374
DIRECTORIO DE IMPORTACIÓN DE PROGRAMA DE COMPRESIÓN
376
ESPACIO DE RELLENO
378

Fig. 3

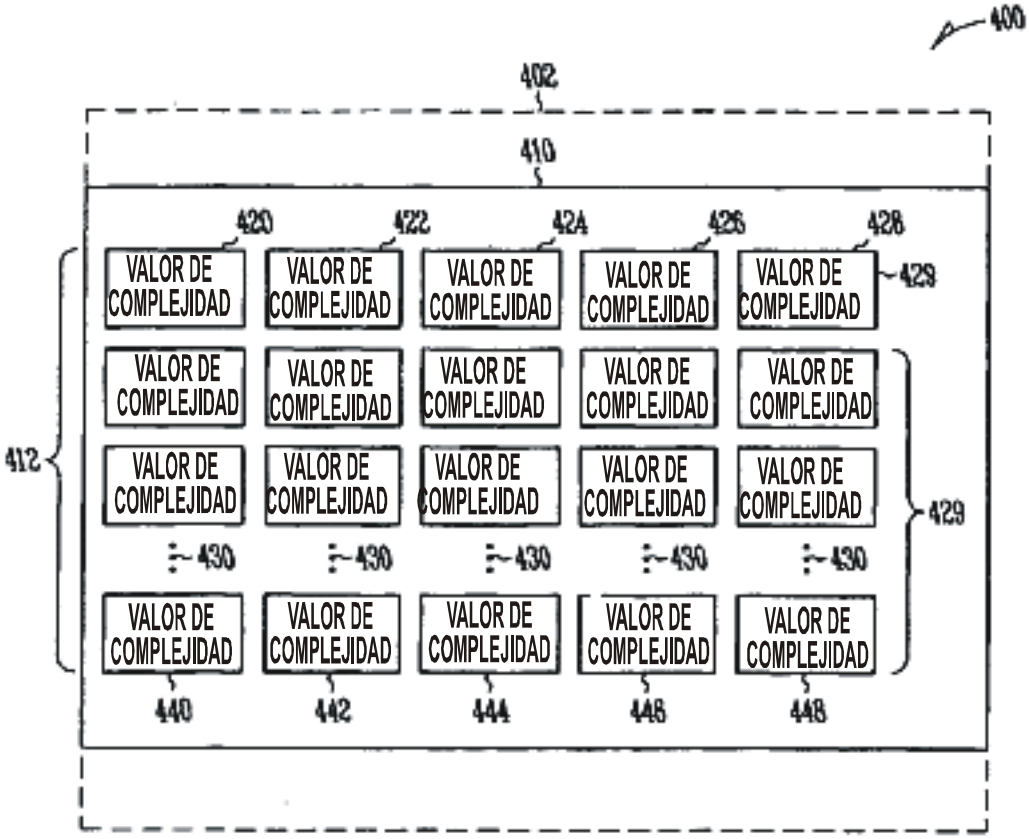

400
402
410
420
422
424
426
428
429
VALOR DE COMPLEJIDAD
412
430
440
442
444
446
448

Fig. 4

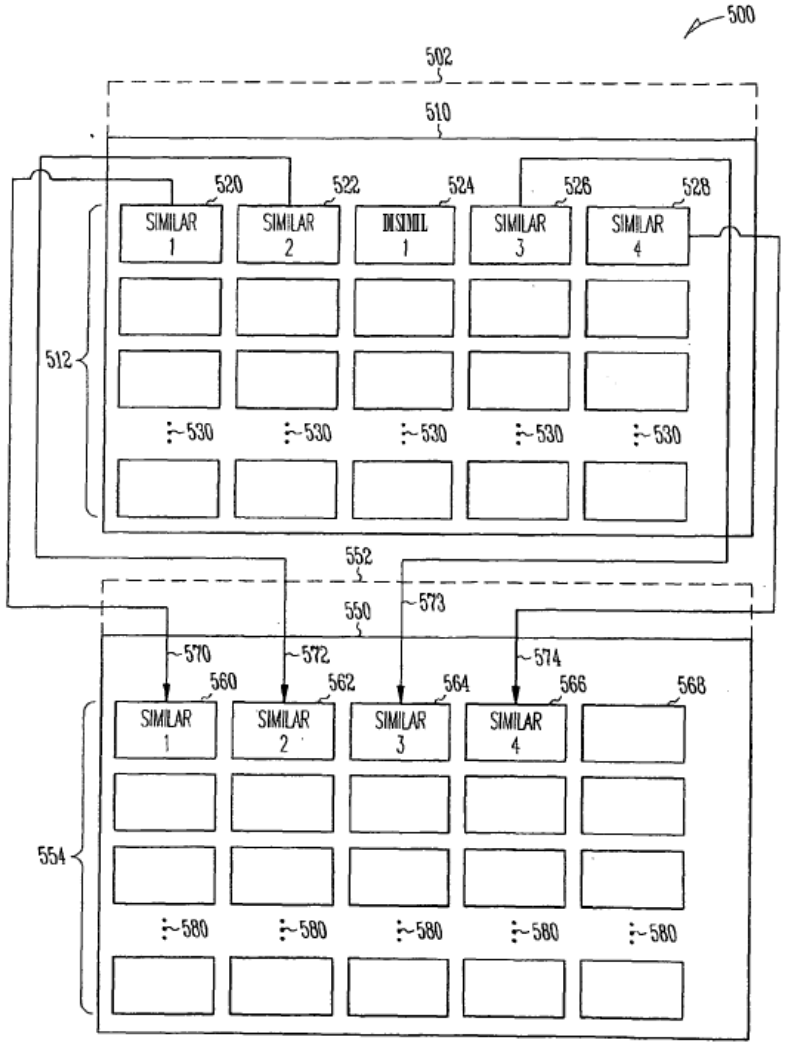

500
502
510
520
522
524
526
528
SIMILAR 1
SIMILAR 2
DISSIMIL 1
SIMILAR 3
SIMILAR 4
512
530
552
550
573
570
572
574
560
562
564
566
568
554
580

Fig. 5

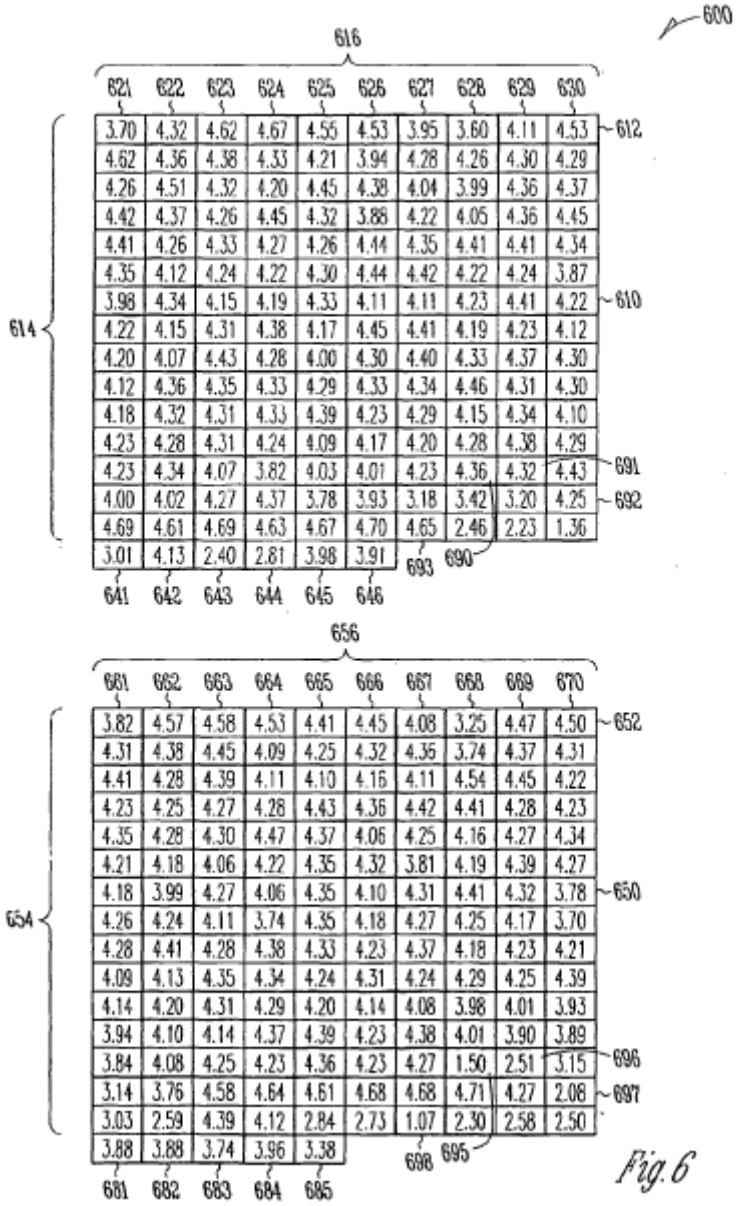
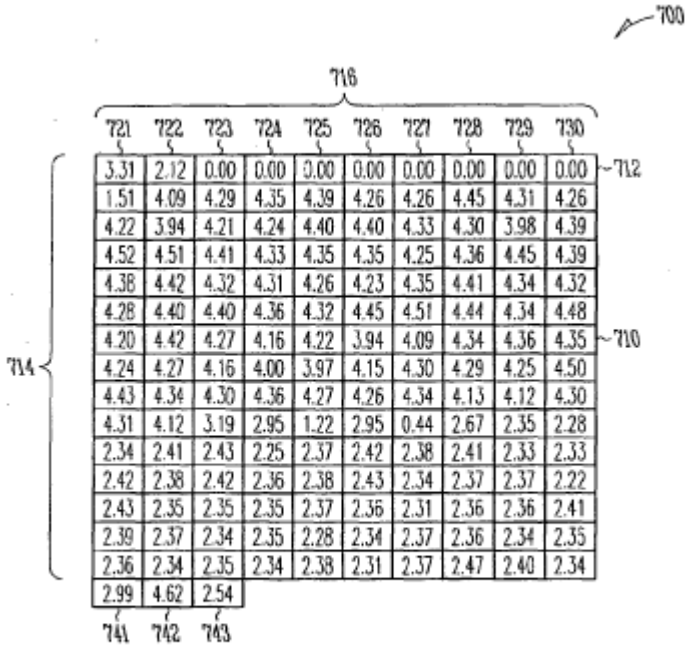

600
616
621 622 623 624 625 626 627 628 629 630
3.70 4.32 4.62 4.67 4.55 4.53 3.95 3.60 4.11 4.53 612
4.62 4.36 4.38 4.33 4.21 3.94 4.28 4.26 4.30 4.29
4.26 4.51 4.32 4.20 4.45 4.38 4.04 3.99 4.36 4.37
4.42 4.37 4.26 4.45 4.32 3.86 4.22 4.05 4.36 4.45
4.41 4.26 4.33 4.27 4.26 4.14 4.35 4.41 4.41 4.34
4.35 4.12 4.24 4.22 4.30 4.44 4.42 4.22 4.24 3.87
3.98 4.34 4.15 4.19 4.33 4.11 4.11 4.23 4.41 4.22 610
614
4.22 4.15 4.31 4.38 4.17 4.45 4.41 4.19 4.23 4.12
4.20 4.07 4.43 4.28 4.00 4.30 4.40 4.33 4.37 4.30
4.12 4.36 4.35 4.33 4.29 4.33 4.34 4.46 4.31 4.30
4.18 4.32 4.31 4.33 4.39 4.23 4.29 4.15 4.34 4.10
4.23 4.28 4.31 4.24 4.09 4.17 4.20 4.28 4.38 4.29
4.23 4.34 4.07 3.82 4.03 4.01 4.23 4.36 4.32 4.43 691
4.00 4.02 4.27 4.37 3.78 3.93 3.18 3.42 3.20 4.25 692
4.69 4.61 4.69 4.63 4.67 4.70 4.65 2.46 2.23 1.36
3.01 4.13 2.40 2.81 3.98 3.91
693 690
641 642 643 644 645 646
656
661 662 663 664 665 666 667 668 669 670
3.82 4.57 4.58 4.53 4.41 4.45 4.08 3.25 4.47 4.50 652
4.31 4.38 4.45 4.09 4.25 4.32 4.36 3.74 4.37 4.31
4.41 4.28 4.39 4.11 4.10 4.16 4.11 4.54 4.45 4.22
4.23 4.25 4.27 4.26 4.43 4.36 4.42 4.41 4.28 4.23
4.35 4.28 4.30 4.47 4.37 4.06 4.25 4.16 4.27 4.34
4.21 4.18 4.06 4.22 4.35 4.32 3.81 4.19 4.39 4.27
4.18 3.99 4.27 4.06 4.35 4.10 4.31 4.41 4.32 3.78 650
654
4.26 4.24 4.11 3.74 4.35 4.18 4.27 4.25 4.17 3.70
4.28 4.41 4.28 4.38 4.33 4.23 4.37 4.18 4.23 4.21
4.09 4.13 4.35 4.34 4.24 4.31 4.24 4.29 4.25 4.39
4.14 4.20 4.31 4.29 4.20 4.14 4.08 3.98 4.01 3.93
3.94 4.10 4.14 4.37 4.39 4.23 4.38 4.01 3.90 3.89
3.84 4.08 4.25 4.23 4.36 4.23 4.27 1.50 2.51 3.15 696
3.14 3.76 4.58 4.64 4.61 4.68 4.68 4.71 4.27 2.08 697
3.03 2.59 4.39 4.12 2.84 2.73 1.07 2.30 2.58 2.50
3.88 3.88 3.74 3.96 3.38
698 695
681 682 683 684 685

Fig. 6



721	722	723	724	725	726	727	728	729	730
3.31	2.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.51	4.09	4.29	4.35	4.39	4.26	4.26	4.45	4.31	4.26
4.22	3.94	4.21	4.24	4.40	4.40	4.33	4.30	3.98	4.39
4.52	4.51	4.41	4.33	4.35	4.35	4.25	4.36	4.45	4.39
4.38	4.42	4.32	4.31	4.26	4.23	4.35	4.41	4.34	4.32
4.28	4.40	4.40	4.36	4.32	4.45	4.51	4.44	4.34	4.48
4.20	4.42	4.27	4.16	4.22	3.94	4.09	4.34	4.36	4.35
4.24	4.27	4.16	4.00	3.97	4.15	4.30	4.29	4.25	4.50
4.43	4.34	4.30	4.36	4.27	4.26	4.34	4.13	4.12	4.30
4.31	4.12	3.19	2.95	1.22	2.95	0.44	2.67	2.35	2.28
2.34	2.41	2.43	2.25	2.37	2.42	2.38	2.41	2.33	2.33
2.42	2.38	2.42	2.36	2.38	2.43	2.34	2.37	2.37	2.22
2.43	2.35	2.35	2.35	2.37	2.36	2.31	2.36	2.36	2.41
2.39	2.37	2.34	2.35	2.28	2.34	2.37	2.36	2.34	2.35
2.36	2.34	2.35	2.34	2.38	2.31	2.37	2.47	2.40	2.34
2.99	4.62	2.54							
741	742	743							

Fig. 7

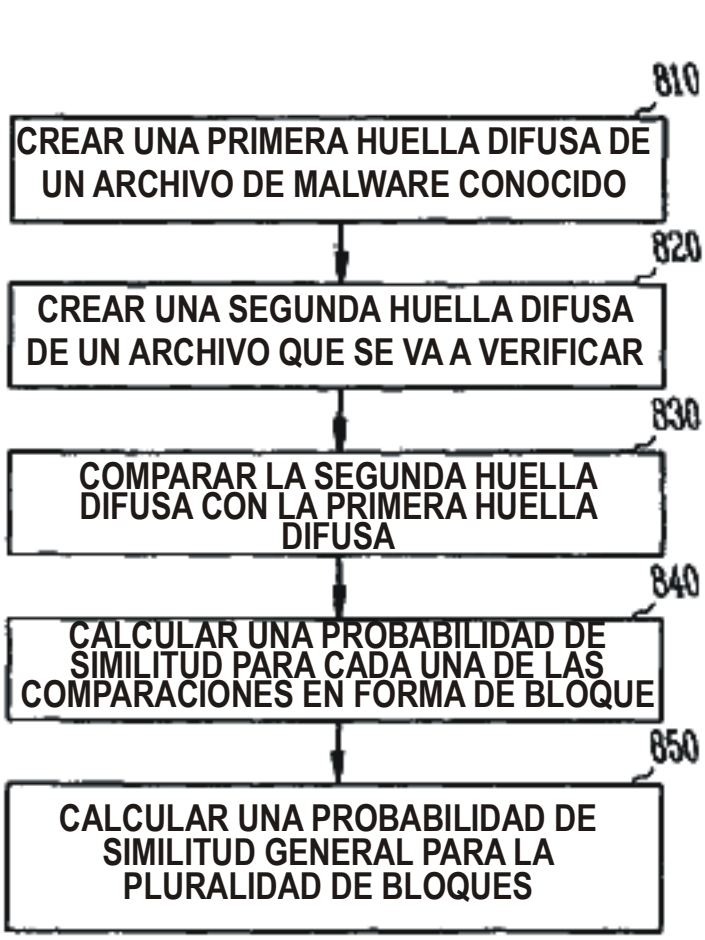

800
810
CREAR UNA PRIMERA HUELLA DIFUSA DE UN ARCHIVO DE MALWARE CONOCIDO
820
CREAR UNA SEGUNDA HUELLA DIFUSA DE UN ARCHIVO QUE SE VA A VERIFICAR
830
COMPARAR LA SEGUNDA HUELLA DIFUSA CON LA PRIMERA HUELLA DIFUSA
840
CALCULAR UNA PROBABILIDAD DE SIMILITUD PARA CADA UNA DE LAS COMPARACIONES EN FORMA DE BLOQUE
850
CALCULAR UNA PROBABILIDAD DE SIMILITUD GENERAL PARA LA PLURALIDAD DE BLOQUES

Fig. 8

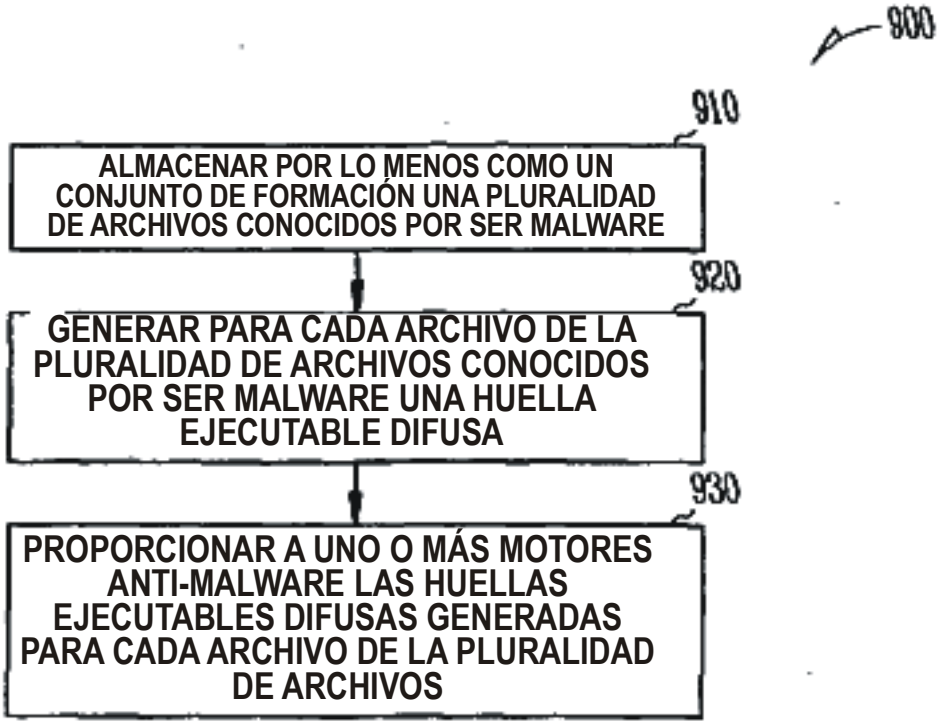

900
910
ALMACENAR POR LO MENOS COMO UN CONJUNTO DE FORMACIÓN UNA PLURALIDAD DE ARCHIVOS CONOCIDOS POR SER MALWARE
920
GENERAR PARA CADA ARCHIVO DE LA PLURALIDAD DE ARCHIVOS CONOCIDOS POR SER MALWARE UNA HUELLA EJECUTABLE DIFUSA
930
PROPORCIONAR A UNO O MÁS MOTORES ANTI-MALWARE LAS HUELLAS EJECUTABLES DIFUSAS GENERADAS PARA CADA ARCHIVO DE LA PLURALIDAD DE ARCHIVOS

Fig. 9