(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 423 491**

(51) Int. CI.:

H04L 29/06 (2006.01)
G06F 21/00 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **12.11.2004 E 04810752 (8)**

(97) Fecha y número de publicación de la concesión europea: **29.05.2013 EP 1682990**

(54) Título: **Aparato, procedimiento y medio para detectar una anomalía de carga útil usando la distribución en n-gramas de datos normales**

(30) Prioridad:

12.11.2003 US 518742 P
28.09.2004 US 613637 P

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
20.09.2013

(73) Titular/es:

THE TRUSTEES OF COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK (100.0%)
535 WEST 116TH STREET
NEW YORK, NY 10027, US

(72) Inventor/es:

STOLFO, SALVATORE J. y
WANG, KE

(74) Agente/Representante:

ARIAS SANZ, Juan

DESCRIPCIÓN

Aparato, procedimiento y medio para detectar una anomalía de carga útil usando la distribución en n-gramas de datos normales

Información de prioridad y referencia cruzada con aplicaciones relacionadas

Esta solicitud reivindica prioridad de la solicitud provisional de Estados Unidos Nº 60 / 518.742 presentada el 12 de noviembre de 2003, y de la solicitud provisional de Estados Unidos Nº 60 / 613.637 presentada el 28 de septiembre de 2004.

Declaración con respecto a investigación patrocinada federalmente

Esta invención se realizó con el apoyo del Gobierno de los Estados Unidos, según el acuerdo número F30602-02-2-0209 concedido por DARPA. El Gobierno de los Estados Unidos tiene ciertos derechos en esta invención.

Trasfondo de la invención

Campo técnico

La presente invención se refiere al análisis de datos y, más específicamente, a la detección de transmisiones de datos anómalos.

Descripción de la técnica relacionada

Los sistemas de ordenadores en red consisten en sedes de procesamiento (p. ej., ordenadores anfitriones) que intercambian datos entre sí. Hay varios protocolos usados por los ordenadores para intercambiar datos. Por ejemplo, TCP / IP es un protocolo de red que proporciona el transporte de datos entre ordenadores que están conectados por una red. A cada ordenador anfitrión se asigna una única dirección del protocolo de Internet (IP), y los datos son intercambiados entre direcciones de IP de origen y direcciones de IP de destino, a un puerto de destino en el anfitrión de destino y desde un puerto de origen en el anfitrión de origen. Un número de puerto corresponde a un servicio o aplicación específica que "está a la escucha" de datos enviados al mismo en ese puerto, desde algún anfitrión remoto de origen. Algunos puertos están estandarizados y tienen asignado un servicio habitual bien conocido. Por ejemplo, los servidores basados en la Red tienen habitualmente asignado el puerto 80 para la transmisión de solicitudes de la Red, entregadas mediante paquetes de TCP / IP, con información de control según los comandos del protocolo de transferencia de hipertexto (HTTP) que espera el servidor de la Red. El TCP / IP transfiere tales datos en forma de "paquetes de red" que consisten en la identificación de la dirección de IP, los números de puertos, la información de control y la carga útil. La carga útil son los datos efectivos esperados por el servicio o la aplicación. En el caso del tráfico de la Red, la carga útil puede consistir, por ejemplo, en solicitudes GET para páginas de la Red representadas por los URL.

Según las redes, tal como Internet, se tornan más accesibles para los usuarios, aumenta significativamente la cantidad de datos transmitidos. Esto presenta una oportunidad para que los individuos causen daño a los ordenadores de usuarios desprevenidos. Los gusanos y los virus, en particular, son causas bien conocidas de grietas de seguridad en sistemas de ordenadores. Estos están constituidos por datos maliciosos enviados a un servicio o aplicación, que se aprovechan de una vulnerabilidad (tal como un desborde de un almacén temporal que proporciona acceso como usuario privilegiado al programa ejecutable del gusano) que provoca que el servicio o aplicación quede inhabilitado, casque o brinde privilegios no autorizados a un atacante. Algunos ejemplos comunes incluyen los recientes gusanos y virus Código Rojo, Nimda y Sobig. Los sistemas convencionales diseñados para detectar y defender los sistemas ante estos sucesos maliciosos e intrusivos dependen de "rúbricas" o "huellas dactilares" que son desarrolladas por seres humanos, o por medios semi-automatizados provenientes de gusanos o virus malignos anteriormente conocidos. Actualmente, los sistemas son protegidos después de que ha sido detectado un gusano, y que ha sido desarrollada y distribuida una rúbrica a los detectores basados en rúbricas, tal como un detector de virus o una regla de cortafuegos.

A fin de reducir la potencial amenaza de los ataques, se establece un cortafuegos para proteger los ordenadores dentro de una red. Los cortafuegos con sistemas de ordenadores que habitualmente permanecen en la pasarela de una red de ordenadores, o que residen en una red, enfrente de un anfitrión u ordenador servidor crítico, y que inspeccionan el tráfico a y desde la red o servidor, y determinan qué tráfico puede avanzar, y qué tráfico será filtrado. Los cortafuegos también pueden ser implementados en forma de software en ordenadores individuales. Como ejemplo, los gusanos que se propagan son habitualmente filtrados por cortafuegos que han sido pre-cargados con una "regla de rúbrica" que detecta la apariencia de un gusano específico. Cuando un paquete y su carga útil "coinciden" con una cadena de rúbrica conocida asociada a un gusano, el cortafuegos bloqueará los paquetes de TCP / IP que el gusano entregó, impidiendo que el servidor sea atacado por ese gusano.

Este enfoque padece de dos problemas fundamentales. En primer lugar, las cadenas de rúbrica asociadas a los gusanos solamente pueden ser construidas después de que el gusano ha sido detectado. Esto significa que el gusano no fue

efectivamente detectado en su primera aparición y, lógicamente, atacó al menos a un servidor, causando daño al servidor. La protección no es posible hasta que un tercero haya construido una cadena de rúbrica y la haya desplegado extensamente en todas las sedes y cortafuegos de la red. Puede perderse un tiempo precioso durante este proceso, que habitualmente puede requerir muchos días. Durante este tiempo, el gusano se habrá extendido con éxito ampliamente por Internet, dañando a muchos miles, si no millones, de anfitriones. Esto es porque los gusanos, en particular, se propagan rápidamente en Internet e infectan y destruyen sistemas a velocidades muy altas. En segundo lugar, hay muchísimos gusanos que han aparecido en Internet, y cada uno de estos ha tenido diversas cadenas de rúbrica construidas para su detección, que están cargadas en todos los cortafuegos. Esto implica que, a lo largo del tiempo, los cortafuegos deben crecer en complejidad a fin de almacenar, procesar y correlacionar muchas cadenas de rúbrica con cada carga útil de paquete entregada a la pasarela o al servidor.

Se han hecho varios intentos para detectar gusanos analizando la velocidad de escaneo y sondeando desde orígenes externos que indicarían que una propagación de gusano está en marcha. Desafortunadamente, este enfoque detecta el comienzo temprano de una propagación y, por definición, el gusano ya ha penetrado con éxito en un sistema, lo ha infectado y ha iniciado su daño y propagación.

En base a lo precedente, sería beneficioso proporcionar un sistema capaz de detectar datos potencialmente dañinos transmitiéndose a través de una red. También sería beneficioso proporcionar un sistema capaz de determinar si datos potencialmente dañinos son o no un programa malicioso. Sería adicionalmente beneficioso proporcionar rúbricas para filtrar programas maliciosos, tales como gusanos y virus, tras una aparición inicial de tales programas.

C. Krügel, T. Toth y E. Kirda describen un Procedimiento Específico de Servicios de Detección de Anomalías para la Detección de Intrusiones por Red, el 30 de abril de 2002 (2002-04030).

Matthew V. Mahoney describe un Procedimiento de Detección de Anomalías de Tráfico en Red en base a Octetos de Paquetes, el 12 de marzo de 2003 (2003-03-12).

Matthew V. Mahoney y Philip K. Chan describen Modelos de Aprendizaje del Tráfico en Red para Detectar Ataques Noveles, en agosto de 2002 (2002-08).

Carol Taylor y Jim Alves-Foss describen el NATE: Análisis en Red de Sucesos Anómalos de Tráfico, un Enfoque de Bajo Coste, el 13 de septiembre de 2001 (2001-09-13).

Sumario de la invención

Estas y otras necesidades son abordadas por la presente invención, en la cual pueden ser detectados datos potencialmente dañinos transmitiéndose a través de una red. Una o más realizaciones de la presente invención utilizan el análisis estadístico de los datos contenidos en una carga útil a fin de determinar si la carga útil es o no potencialmente dañina. El análisis estadístico puede tener la forma de una distribución de valores de octetos de los datos contenidos en la carga útil. Los datos transmitidos a través de la red son comparados con un modelo de datos “normales” previamente recibidos por la red, a fin de determinar su probabilidad de ser dañinos. Los datos normales recibidos por la red pueden ser determinados modelando el tráfico recibido durante un periodo de tiempo fijado. De tal modo, los datos normales representan el flujo regular del tráfico a través de la red y, por lo tanto, pueden incluir buenos datos, datos potencialmente dañinos y ruido. Estos datos normales pueden luego ser recogidos y procesados para crear una distribución estadística modelo que se compara con la distribución estadística de datos recientemente recibidos.

Los aspectos y realizaciones de la invención se exponen según las reivindicaciones adjuntas.

Según una o más implementaciones, las diferencias entre la distribución estadística de dicha al menos una carga útil y la distribución modelo se determinan en base a una métrica de distancia entre las dos. La métrica de distancia, optativamente, puede ser calculada en base a diversas técnicas que incluyen, por ejemplo, una distancia de Mahalanobis. Otras implementaciones de la invención son capaces de determinar si una carga útil anómala es o no un gusano o un virus. Las rúbricas, optativamente, pueden ser generadas para toda carga útil determinada como un gusano o un virus.

Según una o más realizaciones de la presente invención, se proporciona un procedimiento para modelar datos de carga útil recibidos en una red. El procedimiento comprende las etapas de: recibir una pluralidad de datos de carga útil en la red; crear una distribución de longitudes de carga útil para todos los datos de carga útil recibidos; dividir la distribución de longitudes de carga útil en una pluralidad de gamas de carga útil; generar una distribución estadística para cada dato de carga útil recibido; y construir una carga útil modelo para cada gama de carga útil, en base a las distribuciones estadísticas de todos los datos de carga útil recibidos, abarcados por la gama de longitudes de carga útil.

Según al menos una implementación específica, la carga útil modelo se construye en base a los datos de carga útil recibidos más recientemente, o los actuales. Además, una o más implementaciones de la presente invención pueden detectar automáticamente cuándo han sido recogidos suficientes datos de carga útil para construir la carga útil modelo.

Por tanto, han sido esbozadas las características más importantes de la invención y varias, pero no todas, las realizaciones, a fin de que la descripción detallada a continuación pueda ser mejor entendida, y a fin de que la presente contribución a la técnica pueda ser mejor apreciada. Hay, por supuesto, características adicionales de la invención que serán descritas más adelante en la presente memoria, y que formarán la materia objeto de las reivindicaciones adjuntas.

A este respecto, antes de explicar una o más realizaciones de la invención en mayor detalle, ha de entenderse que la invención no está limitada en su aplicación a los detalles de construcción ni a las disposiciones de los componentes expuestos en la siguiente descripción, o ilustrados en los dibujos. Antes bien, la invención es capaz de otras realizaciones y de ser puesta en práctica y llevada a cabo de diversas maneras. Además, ha de entenderse que la fraseología y terminología empleadas en la presente memoria son con fines de descripción, y no deberían considerarse como limitadoras.

De tal modo, los expertos en la técnica apreciarán que la concepción sobre la cual se basa esta revelación puede ser utilizada inmediatamente como una base para el diseño de otras estructuras, procedimientos y sistemas para llevar a cabo los diversos propósitos de la presente invención. Es importante, por lo tanto, que las reivindicaciones sean consideradas como incluyentes de tales construcciones equivalentes, en tanto no se aparten del espíritu y del alcance de la presente invención.

Estas, y varias, características novedosas que caracterizan la invención son señaladas con especificidad en las reivindicaciones adjuntas, que forman parte de esta revelación. Para una mejor comprensión de la invención, sus ventajas operativas y los beneficios específicos logrados por sus usos, debería hacerse referencia a los dibujos adjuntos y a las realizaciones preferidas de la invención, que ilustran la mejor modalidad contemplada para poner en práctica la invención.

Breve descripción de los dibujos

La Figura 1 es un diagrama de bloques que ilustra conceptualmente un sistema para detectar cargas útiles anómalas, de acuerdo a al menos una realización ejemplar de la presente invención.

La Figura 2 es un diagrama de flujo que ilustra las etapas realizadas para modelar datos de carga útil recibidos en una red, según una o más realizaciones de la presente invención.

La Figura 3A es un gráfico que ilustra una distribución de longitudes para datos de carga útil, de acuerdo a una o más realizaciones de la presente invención.

La Figura 3B es un gráfico que ilustra una distribución de longitudes para datos de carga útil, de acuerdo a una o más realizaciones de la presente invención.

La Figura 4 es una distribución estadística de datos contenidos en cargas útiles ejemplares.

La Figura 5A es un total de frecuencias de octetos, ordenados por categoría, de los datos contenidos en cargas útiles.

La Figura 5B es un total de frecuencias de octetos, ordenados por categoría, de los datos mostrados en la Figura 4.

La Figura 6A es una cadena Z de rúbrica de carga útil ejemplar, correspondiente a los datos en la Figura 5A.

La Figura 6B es una cadena Z de rúbrica de carga útil ejemplar, correspondiente a los datos en la Figura 5B.

La Figura 7 es un diagrama de flujo que ilustra las etapas realizadas para modelar datos de carga útil, de acuerdo a una o más realizaciones de la presente invención.

La Figura 8 es un diagrama de flujo que ilustra la manera de la cual los datos de carga útil son recogidos automáticamente, de acuerdo a una o más realizaciones de la presente invención.

La Figura 9 es un diagrama de flujo que ilustra las etapas realizadas para detectar cargas útiles anómalas transmitidas a través de una red.

La Figura 10 es un diagrama de flujo que ilustra las etapas realizadas para detectar cargas útiles anómalas, según una o más realizaciones de la presente invención.

La Figura 11 es un diagrama que ilustra la detección de un ejemplo de gusano.

La Figura 12 es un diagrama de bloques que ilustra conceptualmente la entrega de distintos tipos de ficheros a un ordenador, por una red.

La Figura 13 es un diagrama de flujo que ilustra las etapas realizadas para identificar tipos de ficheros, según una o más realizaciones de la presente invención.

Las Figuras 14A a 14I son gráficos que ilustran la distribución estadística de distintos tipos de ficheros.

La Figura 15 es un diagrama de flujo que ilustra las etapas realizadas para modelar un tipo de fichero según una o más realizaciones de la presente invención.

La Figura 16 es un diagrama de flujo que ilustra las etapas realizadas para verificar tipos de ficheros, según una o más realizaciones de la presente invención.

La Figura 17 es un diagrama de flujo que ilustra las etapas realizadas para identificar programas maliciosos, según una o más realizaciones de la presente invención.

La Figura 18 es un diagrama de bloques que ilustra conceptualmente un ataque por varios sistemas de ordenador.

La Figura 19 es un diagrama de flujo que ilustra las etapas realizadas para rastrear el origen de una transmisión, según una o más realizaciones de la presente invención.

Descripción detallada de la invención

Se hará ahora referencia en detalle a una o más realizaciones de la invención. Tales realizaciones se proporcionan a modo de explicación de la invención, que no está concebida para limitarse a las mismas. De hecho, los medianamente expertos en la técnica apreciarán, al leer la presente especificación y ver los presentes dibujos, que pueden hacerse diversas modificaciones y variaciones.

Por ejemplo, las características ilustradas o descritas como parte de una realización pueden ser usadas en otras realizaciones, para producir otra realización más. Adicionalmente, ciertas características pueden ser intercambiadas con dispositivos o características similares, no mencionados aún, que realizan las mismas, o similares, funciones. Por lo tanto, se pretende que tales modificaciones y variaciones estén incluidas dentro de la totalidad de la presente invención.

Antes de describir los detalles de la invención, se presentará una breve exposición de algunas de las notaciones y nomenclatura usadas en la descripción. Luego, se presentará una descripción de hardware ejemplar, utilizable al poner en práctica la invención.

Notaciones y nomenclatura

Las descripciones detalladas que siguen pueden ser presentadas en términos de procedimientos de programas ejecutados en un ordenador, o una red de ordenadores. Estas descripciones y representaciones procedimentales son un medio usado por los expertos en la técnica para transmitir efectivamente la sustancia de su obra a otros expertos en la técnica. A fin de ejecutar tales procedimientos, puede ser necesario extraer información de uno o más orígenes externos o dispositivos de entrada. La información también puede ser extraída desde diversos dispositivos de almacenamiento, que pueden estar situados tanto internamente como externamente. Al completar la fase de ejecución, la información puede ser emitida a diversos orígenes, tales como un dispositivo visor, un dispositivo de almacenamiento magnético, dispositivos de memoria no volátil, memoria volátil y / o impresoras. La información puede ser adicionalmente transmitida a dispositivos remotamente situados, usando diversos procedimientos y redes de comunicación, tales como las cableadas, inalámbricas, satelitales, ópticas, etc.

El término “medio legible por ordenador”, según se usa en la presente memoria, se refiere a cualquier medio que participe en el suministro de instrucciones a un procesador para su ejecución. Un medio de ese tipo puede adoptar muchas formas, que incluyen, sin limitación, medios no volátiles, medios volátiles y medios de transmisión. Los medios no volátiles incluyen, por ejemplo, discos ópticos o magnéticos. Los medios volátiles incluyen memoria dinámica instalada en el ordenador. Los medios de transmisión pueden incluir cables coaxiales, alambre de cobre y fibras ópticas. Los medios de transmisión también pueden adoptar la forma de ondas acústicas o lumínicas, tales como las generadas durante las comunicaciones de datos de frecuencia de radio (RF) e infrarrojos (IR). Las formas comunes de medios legibles por ordenador incluyen, por ejemplo, el disco rígido, la cinta magnética, cualquier otro medio magnético, un CD-ROM, un DVD, cualquier otro medio óptico, una RAM, una PROM, una EPROM, una FLASH-EPROM, cualquier otro chip o cartucho de memoria, una onda portadora según lo descrito más adelante en la presente memoria, o cualquier otro medio desde el cual pueda leer un ordenador.

Panorama general de la detección de anomalías basada en cargas útiles

La presente invención tiene, como un objetivo al menos, analizar los datos de carga útil recibidos en una red. El análisis puede ser usado para diversos propósitos, que incluyen, por ejemplo, modelar el flujo normal de tráfico a través de la red. Una o más realizaciones de la presente invención admiten detectar la primera ocurrencia de un gusano en una pasarela de un sistema de red, e impedirle entrar, en primer lugar. De tal modo, puede impedirse que el gusano se empeñe en sus acciones destructivas y en su propagación. En cambio, una o más realizaciones de la presente invención realizan, en parte, el análisis y la modelación de cargas útiles “normales”, de las que se espera que sean entregadas a un servicio o

aplicación de red.

En una o más realizaciones, la presente invención incluye una fase de “aprendizaje” que recoge cargas útiles “normales” y produce una distribución estadística de n-gramas (o “valores de octeto”) de esas cargas útiles, lo que sirve como un modelo para las cargas útiles “normales”. Las cargas útiles recogidas no son necesariamente cargas útiles seguras. Más bien, estas cargas útiles representan información transmitida a través de la red durante un periodo regular de tiempo. Esto se denomina una carga útil modelo. Después de que esta carga útil modelo ha sido producida en la fase de aprendizaje, comienza una fase de detección de anomalías. La fase de detección de anomalías captura cargas útiles entrantes al servicio, o a la aplicación, de la red. La carga útil entrante es sometida a pruebas para determinar diferencias con respecto a la carga útil modelo. Esto puede hacerse, por ejemplo, probando la carga útil en cuanto a su consistencia (o distancia) con respecto a la carga útil modelo. En una o más realizaciones de la presente invención, se calcula un “centroide” para todos los datos de carga útil que han sido recogidos durante la fase de aprendizaje. El modelo de centroide funciona entonces como una carga útil modelo para detectar cargas útiles anómalas.

Según una o más realizaciones de la presente invención, cualquiera carga útil determinada como demasiado distinta a la carga útil normal (p. ej., esperada) se considera anómala y es filtrada, o atrapada de otro modo, para que no sea enviada al servicio, o a la aplicación. Esto puede prevenir una infección potencial si se determina posteriormente que la carga útil es un gusano o un virus. El nivel de tolerancia admitido puede ser fijado por el usuario, o fijado automáticamente en base a criterios predeterminados. En una o más realizaciones de la presente invención, el modelo de centroide puede ser calculado en base a la distribución del total de frecuencias de octetos del conjunto de las cargas útiles “normales” analizadas durante la fase de aprendizaje (o de entrenamiento). Al menos una de las métricas de distancia que pueden ser usadas es la métrica de distancia de Mahalanobis. En una o más realizaciones de la presente invención, la distancia de Mahalanobis puede ser aplicada a un histograma discreto finito de frecuencias de valores (o caracteres) de octetos, calculadas en la fase de entrenamiento. Hay numerosas opciones de ingeniería posibles para implementar las técnicas de la presente invención en un sistema, e integrar el detector con un cortafuegos estándar, cortafuegos agentes de aplicaciones, u otra tecnología de seguridad informática basada en una red, o basada en un anfitrión, para impedir la primera aparición de un gusano entrando a un sistema protegido, de red o anfitrión. El sistema de detección de anomalías de la presente invención puede estar basado en la distribución estadística de valores de octeto en la carga útil de la conexión de red. Según una o más realizaciones de la presente invención, puede construirse un perfil para la carga útil normal de la conexión, que sea específica para la sede y puerto, y usarse luego para detectar cualquier desviación significativa de la carga útil de la nueva conexión, como un posible ataque malicioso.

Como una primera etapa, las cargas útiles que han atravesado cortafuegos y han sido entregadas a servicios y puertos anfitriones pueden ser recogidas mediante un archivo (o registro), o auditadas en tiempo real. Estos datos (es decir, las cargas útiles recibidas) constituyen los datos de entrenamiento a partir de los cuales se obtiene un modelo normal (o carga útil modelo). El conjunto de cargas útiles puede tener una amplia gama de longitudes, según, por ejemplo, la cantidad de datos entregados al servicio.

Como una segunda etapa, la distribución de las longitudes de cargas útiles de las conexiones puede ser modelada, y la distribución de longitudes puede ser dividida en múltiples gamas. En general, la longitud de las conexiones puede variar sobre una gama muy amplia, p. ej., desde unos pocos octetos a millones de octetos. Las distintas gamas de longitudes, por lo tanto, pueden tener distintos tipos de carga útil. Por ejemplo, las conexiones breves del HTTP contienen usualmente letras y dígitos, y las conexiones muy largas del HTTP a menudo contienen imágenes, recortes de vídeo, ficheros ejecutables, etc., que contienen un montón de caracteres no imprimibles. Por tanto, según un aspecto de la presente invención, pueden ser construidos modelos de carga útil para distintas gamas de longitudes de conexión. Esto puede tener ventajas sobre la construcción de solamente un modelo para todas las conexiones.

Según una o más realizaciones de la presente invención, la distribución de las longitudes de conexión puede ser modelada sin ninguna hipótesis previa, mediante el uso de estimaciones de núcleo. Las estimaciones de núcleo allanan la contribución de cada punto de datos observados en una vecindad local de ese punto. Pueden ser generadas curvas de probabilidad para las conexiones del HTTP entrantes / salientes. Las curvas de probabilidad pueden ser divididas según valores acumulativos de probabilidad. Las gamas de alta probabilidad tendrán pequeños tamaños de contenedor, mientras que las gamas de baja probabilidad tendrán grandes tamaños de contenedor.

Adicionalmente, hay otras técnicas que pueden ser usadas para dividir las gamas de longitudes. Por ejemplo, pueden ser usados algoritmos de agrupamiento de diversas clases, por los cuales las métricas de distancia empleadas pueden estar basadas en las métricas de distancia aplicadas a distribuciones de caracteres de n-gramas, según se describe más completamente más adelante. Una vez que la distribución de longitudes ha sido dividida en gamas específicas, los datos de entrenamiento de las cargas útiles normales se dividen en subconjuntos disjuntos de cargas útiles que tienen longitudes definidas por la división. Por ejemplo, si una división de longitudes identifica una gama como entre 0 y 50, entonces todas las cargas útiles de longitud acotada por 50 octetos serán incluidas en este subconjunto de cargas útiles.

La siguiente etapa implica modelar todas las cargas útiles normales dentro de cada gama de longitudes. Para aquellas

conexiones cuya longitud esté dentro de alguna gama de longitudes, puede calcularse el total de frecuencias medias de octetos, para los octetos ASCII entre 0 y 255. Esta distribución de frecuencias de octetos individuales se llama una distribución de 1-gramas. Las distribuciones también pueden ser construidas para 2-gramas (dos octetos consecutivos), 3-gramas (tres octetos consecutivos), etc. Además, la distribución de frecuencias puede ser de hetero-gramas, lo que significa que se usa una mezcla, por ejemplo, de distribuciones de 1-gramas y 2-gramas. Según se usa en la presente memoria, una distribución de hetero-gramas es una mezcla de modelos de n-gramas de distinto tamaño dentro de los datos. Los ejemplos de otras mezclas pueden incluir, sin limitación: 1-gramas y 3-gramas; 1-gramas y 4-gramas; 2-gramas y 3-gramas; 2-gramas y 5-gramas; 3-gramas y 4-gramas; 3-gramas y 7-gramas; 1-gramas, 3-gramas y 4-gramas; 1-gramas, 5-gramas, 7-gramas, 3-gramas, 2-gramas, 5-gramas, 3-gramas, 4-gramas; etc. Puede usarse prácticamente cualquier mezcla. El uso de 1-gramas, como ejemplo, si la frecuencia de caracteres está ordenada desde la más alta a la más baja, es usualmente similar a una distribución Zipf, con una larga cola.

Al menos una técnica (pero no la única técnica) para detectar una carga útil anómala es determinar si algún valor de carácter o de octeto aparece o no en la carga útil en cuestión, con una frecuencia de ocurrencia muchas veces mayor o menor que la que se esperaría de los datos de entrenamiento observados. Tal frecuencia de caracteres, junto con la varianza de cada octeto, puede caracterizar la carga útil dentro de alguna gama. Otra forma de representarla es usar una "cadena de rúbrica de carga útil normal", que es la correspondiente cadena de octetos ASCII de la anterior distribución ordenada de frecuencias, donde los caracteres o valores de octetos que tienen una frecuencia cero son ignorados. Obviamente, si se considera la varianza de cada octeto, puede obtenerse una "vecindad de cadenas de rúbrica de carga útil normal", lo que significa que cada octeto tiene una vecindad que contiene otros diversos octetos que pueden ser posiblemente ordenados en ese lugar si se considera la varianza, además de la frecuencia promediada.

Las nuevas cargas útiles pueden ser probadas para detectar en cuánto se apartan del modelo normal. Esto puede hacerse, por ejemplo, comparando la distribución de frecuencias de octetos de la nueva carga útil con el modelo de carga útil normal. Una vez que está construido el perfil, hay múltiples maneras de comparar la carga útil de alguna nueva conexión con dicho perfil, para detectar cualquier gran diferencia o desviación. La métrica de distancia de Mahalanobis es una de esas funciones de distancia que calculan la similitud entre dos distribuciones de frecuencia.

La fórmula de la distancia de Mahalanobis es $D(h_1, \bar{h}) = (h_1 - \bar{h})^T A(h_1 - \bar{h})$, donde $\bar{h}$ es el vector de características del perfil, calculado a partir de una fase de entrenamiento previa. La matriz *B* de Covarianza, $b_{ij} = Cov(\bar{h}_i, \bar{h}_j)$, y $A = B^{-1}$. Suponiendo que los octetos sean estadísticamente independientes, la matriz *B* quedará diagonal y los elementos son solamente la varianza de la frecuencia promedio de cada octeto. Para hacer que el cálculo sea sencillo y rápido, puede obtenerse la distancia simplificada de Mahalanobis, como $D(h_1, \bar{h}) = \sum_{i=0}^{n-1} (h_1[i] - \bar{h}[i]) / \bar{\sigma}_i$, donde *n* es igual a 256 si se usa un 1-grama. Si se aplica este procedimiento a la distribución de frecuencias de 2-gramas, entonces habría $n = 256^2$, o 65.536, contenedores. Pueden usarse diversos procedimientos para reducir significativamente este número. En general, el cálculo será lineal con respecto a la longitud de la carga útil de la conexión que está siendo probada.

Modelación de datos de carga útil

Con referencia a los dibujos, e inicialmente a la Figura 1, se muestra un sistema 100 para detectar cargas útiles anómalas, según una o más realizaciones de la presente invención. El sistema 100 de detección de cargas útiles de la Figura 1 incluye un servidor 110 que recibe datos (p. ej., cargas útiles o datos de cargas útiles) desde orígenes externos, tales como, por ejemplo, Internet 116. El servidor 110 también puede incluir un cortafuegos 112 que ayuda a proteger al servidor 110 ante ataques potenciales. El cortafuegos 112 funciona para filtrar ciertos datos, a fin de reducir la posibilidad de que se transmitan virus y gusanos al servidor 110 desde Internet 116. El servidor 110 también puede estar acoplado con una o más estaciones 114 de trabajo y / u otros servidores 118. Las estaciones 114 de trabajo se conectan, e interactúan, con Internet 116 a través del servidor 110. Más específicamente, cada estación 114 de trabajo transmite datos al servidor 110, y el servidor 110 transmite posteriormente estos datos a un destino mediante Internet 116. Los datos de diversos orígenes pueden ser recibidos por el servidor 110, y filtrados a través del cortafuegos 112. Una vez que los datos han sido filtrados, el servidor 110 remite los datos a las estaciones 114 de trabajo a fin de facilitar la interacción con dispositivos remotamente situados.

El servidor 110 genera una distribución estadística para los datos de cargas útiles recibidos desde la red (o desde Internet 116), según se expone en mayor detalle más adelante. El servidor 110 puede almacenar una pluralidad de distribuciones modelo (es decir, cargas útiles modelo) que representan, o corresponden a, las distribuciones estadísticas de las cargas útiles normales recibidas por el servidor 110. La distribución estadística de los nuevos datos de carga útil recibidos por el servidor 110 es comparada con una carga útil modelo seleccionada. La carga útil modelo es seleccionada, al menos en parte, en base al tamaño de los datos de la carga útil actual, recibidos por el servidor 110. Por ejemplo, si los datos de la carga útil actual, recibidos por el servidor 110, tienen 256 octetos, entonces la carga útil modelo seleccionada por el

servidor 100 incluirá al menos una gama que abarque 256 octetos.

El servidor 110 compara las distribuciones a fin de identificar cargas útiles anónimas. Habitualmente, una carga útil anómala tendrá ciertas diferencias en su distribución estadística, con respecto a la carga útil modelo. Según una o más realizaciones de la presente invención, el servidor 110 es capaz de procesar adicionalmente las cargas útiles, o datos, anómalos, a fin de determinar si corresponden a programas maliciosos tales como, y sin limitaciones, los gusanos o virus. Si se detecta un gusano o un virus, el servidor 110 puede, optativamente, generar un patrón de virus o rúbrica de gusano, que pueda ser usado protegerse a sí mismo y a otras máquinas. Por ejemplo, según una o más realizaciones de la presente invención, cuando el servidor 110 detecta y genera un patrón de virus (o rúbrica de gusano), actualiza automáticamente las reglas para el cortafuegos 112, de modo que el virus o gusano identificado sea filtrado si es adicionalmente transmitido desde Internet 116, o desde otras redes. Otras realizaciones (o realizaciones solapadas) de la presente invención permiten al servidor 110 transmitir los patrones de virus y las rúbricas de gusanos a otros servidores remotos 118. Los servidores remotos 118 pueden estar conectados con el servidor 110 a través de una conexión segura y / o directa, según lo ilustrado en la Figura 1. Alternativamente, los patrones y rúbricas de virus pueden ser transmitidos a través de Internet 116 a los servidores remotos 118. Una vez que los servidores remotos 118 reciben los patronos y rúbricas de virus, pueden actualizar sus reglas de filtrado, de modo que puedan protegerse a sí mismos y a los dispositivos conectados ante aplicaciones maliciosas transmitidas por Internet 116.

Según una o más realizaciones de la presente invención, múltiples servidores (p. ej., 110 y 118) pueden usar el sistema 110 de detección de cargas útiles para identificar datos anómalos. Cada servidor individual (por ejemplo, los de números 110 y 118 de referencia) realizaría las mismas técnicas para identificar datos anómalos, y determinaría adicionalmente si corresponden a gusanos o virus. Sin embargo, debido a que los servidores 110, 118 están remotamente situados, es probable que reciban datos distintos desde la red, o Internet 116. Por lo tanto, cada servidor 110, 118 puede identificar potencialmente a distintos tipos de gusanos o virus. Además, cada servidor 110, 118 puede interactuar e intercambiar información con respecto a patrones de virus y rúbricas de gusanos, de modo que todos los servidores 110, 118 puedan actualizar sus reglas de cortafuegos para filtrar los gusanos o virus más recientemente descubiertos. Además, las estaciones 114 de trabajo individuales pueden implementar las técnicas de la presente invención a fin de proporcionar otra capa de seguridad y / o protegerse independientemente a sí mismas.

Según una o más realizaciones de la presente invención, el servidor 110 crea la distribución modelo en base a datos recibidos a través de la red 116. Por ejemplo, el servidor 110 puede implementar diversas técnicas para capturar, o fisgar, los datos que recibe o transmite. Esta información es tabulada y usada para representar el flujo normal de datos a través de la red. En consecuencia, el flujo normal de datos puede incluir, de forma concebible, ruido y / o programas maliciosos. El servidor 110 puede recoger los datos para un periodo de tiempo prescrito y generar posteriormente la distribución modelo de los datos que han sido recogidos.

La Figura 2 es un diagrama de flujo que ilustra las etapas realizadas para generar una carga útil modelo, según una o más realizaciones de la presente invención. Estas etapas pueden ser realizadas, por ejemplo, por el servidor 110, para generar la carga útil modelo que será comparada con los datos de cargas útiles recibidos por la red 116. En la etapa S210, el servidor recibe los datos de carga útil. Los datos de carga útil pueden ser recibidos desde una pluralidad de orígenes, que incluyen, por ejemplo, otras redes, Internet, redes inalámbricas, redes satelitales, etc. En la etapa S212, se crea una distribución de longitudes para los datos de carga útil que han sido recibidos. Debería observarse, sin embargo, que el servidor puede recibir continuamente datos de carga útil, hasta que un momento tal en que haya recogido una cantidad predeterminada de datos, suficiente para construir la carga útil modelo. La distribución de longitudes creada por el servidor en S212 corresponde a la distribución de longitudes de las cargas útiles individuales recibidas por el servidor durante un periodo de entrenamiento.

Con referencia adicional a la Figura 3A, se muestra un diagrama ejemplar de distribución de longitudes. El diagrama de distribución de longitudes de la Figura 3A muestra la distribución de los datos recibidos, en base al tamaño de la carga útil. Por ejemplo, según se muestra en la Figura 3A, el número de cargas útiles que tienen una longitud cercana al cero es muy baja. Sin embargo, el número de cargas útiles que tienen una longitud que sea de aproximadamente 200 octetos es significativamente mayor. Según se reduce el número de octetos, también se reduce el número de cargas útiles que tienen tal longitud. Esto puede atribuirse al hecho de que la mayoría de los datos recibidos corresponderá a texto y / o ficheros pequeños. Sin embargo, los ficheros más grandes corresponderán a imágenes y / o ficheros de vídeo o de sonido. La Figura 3B ilustra otra distribución de longitudes de datos de carga útil, que oscila entre 0 y 10.000 octetos.

Con referencia nuevamente a la Figura 2, en la etapa S214 se divide la distribución de longitudes. El proceso de división puede hacerse, en parte, para generar múltiples cargas útiles modelo que puedan ser comparadas selectivamente con los datos de carga útil recibidos. Según una o más realizaciones de la presente invención, al menos una ventaja de dividir la distribución de longitudes es reducir la cantidad de tiempo necesaria para calcular la diferencia entre la distribución estadística de la carga útil recibida, en comparación con la distribución de la carga útil modelo. Hay diversas técnicas que pueden ser usadas para dividir la distribución de longitudes. Por ejemplo, según una realización de la presente invención, al menos un algoritmo de agrupamiento puede ser usado para dividir la distribución de longitudes. La distribución de

longitudes también puede ser dividida usando estimaciones de núcleo.

En la etapa S216, se genera una distribución estadística para cada una de las particiones creadas a partir de la distribución de longitudes. La distribución estadística puede corresponder, por ejemplo, a la distribución de frecuencias de los caracteres ASCII (o los datos) contenidos en la carga útil. Con referencia adicional a la Figura 4, se ilustra una distribución estadística de cargas útiles ejemplares con una longitud de entre 150 y 155 octetos. Según la realización ejemplar de la Figura 4, la distribución estadística corresponde al total de frecuencias de octetos de los datos contenidos en la carga útil. Por ejemplo, el eje x representa el valor numérico del carácter ASCII, mientras que el eje y corresponde al número de veces que apareció un carácter específico en la carga útil. El eje y puede ser normalizado, en correspondencia con el porcentaje del número de apariciones de un carácter o valor de octeto específico.

Según una o más realizaciones de la presente invención, puede usarse una técnica de n-gramas para agrupar los octetos al generar la distribución estadística. Usando una técnica de ese tipo, la variable n corresponde a una agrupación específica de octetos, que puede adoptar distintos valores. Por ejemplo, en una distribución de 2-gramas, los pares adyacentes de octetos serían agrupados entre sí, como una característica. De manera similar, usando una distribución de 4-gramas, 4 octetos adyacentes serían agrupados como una característica. Debería observarse adicionalmente que una o más realizaciones de la presente invención pueden proveer la distribución de hetero-gramas de la carga útil, según lo descrito anteriormente. Por ejemplo, una parte de la distribución de longitudes puede ser agrupada como 2 octetos, mientras que otras partes son agrupadas como tres, o cuatro, etc. De tal modo, según la complejidad de la distribución de longitudes y / o de los datos recibidos por el servidor, una distribución en hetero-gramas puede ser usada para reducir la cantidad de tiempo necesaria para calcular la diferencia entre los datos de una carga útil recibida y las cargas útiles modelo.

Según una o más realizaciones de la presente invención, la distribución estadística puede ser dispuesta de diversas formas. Por ejemplo, un total de frecuencias de octetos ordenados por clasificación puede ser generado a partir de la distribución de valores de octetos. La Figura 5A ilustra un total de frecuencias de octetos ordenados por clasificación de las distribuciones de caracteres. En la Figura 5A, el carácter que aparecía con la mayor frecuencia es correlacionado con el carácter uno en el eje x. El siguiente carácter correlacionado, más frecuentemente recibido, estaba contenido en los datos de la carga útil. En consecuencia, en el gráfico de frecuencias de octeto, ordenado por clasificación, la parte de más a la derecha del gráfico está vacía. Además, para la longitud de conexión de la muestra y los datos de carga útil probados para este ejemplo, solamente estaban presentes 29 caracteres.

La Figura 5B ilustra otro gráfico ejemplar de frecuencias de octeto, ordenado por clasificación, para una longitud de conexión de entre 150 y 155 octetos (ilustrado en la Figura 4). Como puede verse en la Figura 5B, había más caracteres ASCII presentes en los datos de carga útil, en comparación con la Figura 5A. En particular, estaban presentes 83 caracteres ASCII. Por tanto, la parte más a la derecha del gráfico no tiene ningún valor.

Con referencia nuevamente a la Figura 2, en la etapa S218 se construye la carga útil modelo, o las cargas útiles modelo. Según el número de divisiones generadas, se construiría un número correspondiente de cargas útiles modelo. Por ejemplo, si la distribución de longitudes fuera dividida en 20 secciones, habría 20 distintas cargas útiles modelo construidas. Según una o más realizaciones de la presente invención, cada carga útil modelo puede ser generada en forma de una cadena de rúbrica de carga útil.

Con referencia a la Figura 6A, se muestra una rúbrica “cadena Z” 150 de carga útil ejemplar, correspondiente al total de frecuencias de octeto, ordenado por clasificación de la Figura 5A. La cadena Z de rúbrica de carga útil es un valor de cadena formado directamente a partir de los datos de distribución estadística que representan los valores específicos de octetos en orden de frecuencia, desde la más alta a la más baja. Además, las cadenas Z de rúbrica de carga útil de la presente invención pueden tener distintas longitudes, según el contenido de los datos. Como se muestra en la Figura 6A, la cadena 150 de rúbrica de carga útil incluye una pluralidad de caracteres ASCII. La tabla 160 ilustra en mayor detalle los datos correspondientes a aquellos caracteres que aparecieron con la mayor frecuencia. Como puede verse a partir de la Figura 6A, la tabla solamente incluye 29 entradas. Este valor corresponde al número de caracteres que aparecieron para la longitud de conexión de la muestra.

La Figura 6B ilustra una cadena de rúbrica de carga útil ejemplar para el gráfico de frecuencias ordenadas por clasificación de la Figura 5B. La cadena 170 de rúbrica también se muestra con la correspondiente tabla que contiene los valores de cada carácter del gráfico. De manera similar a la Figura 6B, solamente 83 entradas están presentes en la tabla 180. Este valor corresponde nuevamente al gráfico en la Figura 5B.

Una vez que han sido construidas las cargas útiles modelo, el servidor compara cada dato de carga útil recibido con las cargas útiles modelo, a fin de identificar cargas útiles anómalas. Además, según lo indicado anteriormente, los datos de carga útil recibidos son comparados con una carga útil modelo, que corresponde a la longitud de los datos de carga útil. Por ejemplo, si los datos de una carga útil recibida tuvieran una longitud de 40 octetos, serían comparados con una cadena de rúbrica de carga útil tal como la de la Figura 6A. Análogamente, si los datos de la carga útil recibida tienen una

longitud de 153 octetos, serían comparados con una cadena de rúbrica de carga útil tal como la de la Figura 6B.

Pasando ahora a la Figura 7, se ilustra un diagrama de flujo para construir cargas útiles modelo, según una o más realizaciones de la presente invención. En la etapa S250, son recibidos los datos de carga útil. Esto corresponde al servidor que recibe datos a través de la red. En la etapa S252, se crea una distribución de longitudes para los datos de carga útil. Como se ha expuesto anteriormente, el servidor recibirá una pluralidad de datos de carga útil, suficientes para crear cargas útiles modelo. Una vez que hayan sido recibidos la pluralidad de datos de carga útil, puede crearse la distribución de longitudes. Alternativamente, una cantidad mínima de datos puede ser recibida por el servidor, y la distribución de longitudes puede ser creada inicialmente en base a estos datos. Según se reciben los datos, la distribución de longitudes sería continuamente actualizada, y las cargas útiles modelo construidas serían refinadas para reflejar mejor el tipo de datos que están siendo recibidos actualmente a través de la red.

En la etapa S254, la distribución de longitudes es dividida en una pluralidad de divisiones. Según lo expuesto anteriormente, la distribución de longitudes puede ser dividida usando estimaciones de núcleo y / o diversas técnicas de agrupamiento. En la etapa S256, se crea una distribución de valores de octeto para cada división. En S258, los datos de carga útil son clasificados en las distintas divisiones. Por ejemplo, si una de las divisiones corresponde a datos de carga útil con una longitud de entre 0 y 50 octetos, entonces cualquier dato individual de carga útil que cayera dentro de esa gama sería clasificado en esa división específica. En S260, se construye una carga útil modelo para cada división.

Según una o más realizaciones de la presente invención, pueden aplicarse diversas técnicas para construir y / o refinar las cargas útiles modelo. Por ejemplo, según lo ilustrado en la Figura 7, los datos de carga útil contenidos en cada división pueden ser compilados en la etapa S262. Esto corresponde a una etapa donde todos los datos de carga útil en la división son combinados para un procesamiento adicional. Una vez que las cargas útiles en las divisiones están compiladas, se calcula un centroide para cada división en la etapa S264. El centroide puede ser calculado usando cualquiera entre una pluralidad de técnicas de cálculo. En la etapa S266, el centroide es designado como la carga útil modelo. En consecuencia, usando este procedimiento de refinación de la carga útil modelo, el centroide (es decir, la carga útil modelo recientemente designada) sería usado para determinar si los datos de carga útil entrantes son o no anómalos.

Alternativamente, en la etapa S268, se crea una pluralidad de distribuciones de longitudes de división. Una distribución de longitudes de división es sencillamente la distribución de los datos dentro de la división, según lo expuesto anteriormente. Una vez que está creada la distribución de longitudes de división, los datos son agrupados en la etapa S270, para generar una pluralidad de distribuciones de agrupamiento. En la etapa S272, se calcula un centroide para cada agrupación que haya sido generada. En la etapa S274, se calcula un centroide modelo. Según una o más realizaciones de la presente invención, el centroide modelo calculado en la etapa S247 corresponde al centroide de todos los centroides de agrupamiento que fueron calculados en la etapa S272. En consecuencia, el centroide modelo es el centroide de una pluralidad de centroides. En la etapa S276, el centroide modelo es designado como la distribución modelo. Así, los datos entrantes serían comparados con el centroide modelo a fin de determinar cargas útiles anómalas que, en potencia, podrían ser un programa malicioso.

Según una o más realizaciones de la invención, el algoritmo de agrupamiento usado conjuntamente con la etapa S270 puede ser un algoritmo incremental en tiempo real, y puede no ser necesario especificar el número de agrupamientos de antemano. Un número inicial de agrupamientos, K. puede ser fijado para que corresponda al máximo número admisible posible de agrupamientos. Por ejemplo, un valor de K = 10 puede ser suficiente para representar el número de distintas clases de tráfico de carga útil de red. Una nueva carga útil, que es analizada durante la fase de entrenamiento, puede ser usada para actualizar las estadísticas de un centroide previamente calculado, que sea el más similar a la nueva carga útil. Si todavía no hay ningún centroide calculado, o ningún centroide existente que sea similar a la nueva carga útil, entonces la nueva carga útil es usada como un nuevo centroide. Si el número total de centroides es mayor que K, entonces los dos centroides más similares pueden ser fundidos combinando sus estadísticas en una distribución. Cuando la fase de entrenamiento está completa, ciertos centroides pueden ser podados, reteniendo solamente aquellos centroides que fueron calculados con un número mínimo especificado de cargas útiles de entrenamiento. Tal poda de centroides “sub-entrenados” puede ayudar en la identificación del “ruido” en los datos de entrenamiento, lo que podría representar, posiblemente, una carga útil “mala” que, en otro caso, no sería identificada durante la fase de detección.

Según una o más realizaciones de la presente invención, el servidor es adicionalmente capaz de eliminar por antigüedad datos que han sido recibidos, de modo que la distribución modelo en uso pueda reflejar con precisión el tipo de datos que está fluyendo actualmente a través de la red. Por ejemplo, en la etapa S278, el servidor puede comprobar la fecha en las cargas útiles que han sido recibidas y usadas para generar la distribución modelo. En la etapa S280, el servidor determina si la fecha de los datos de una carga útil es mayor que, o más antigua que, un umbral predeterminado. Por ejemplo, a fin de mantener, o de conservar actual, el perfil de la carga útil, puede determinarse que solamente los datos de cargas útiles recibidos dentro de los últimos seis meses deberían ser usados para construir la distribución modelo. En base a un ejemplo de ese tipo, los datos de una carga útil que tengan una antigüedad de más de seis meses superarían el umbral. Si la fecha de los datos de carga útil supera el umbral, entonces el control pasa a la etapa S282, donde se descartan los datos de la carga útil. Alternativamente, si la fecha de los datos de carga útil no supera el valor de umbral, entonces el

servidor continúa sencillamente recibiendo datos de carga útil en la etapa S284.

Según una o más realizaciones de la presente invención, el servidor puede recibir y procesar continuamente datos de carga útil para refinar incrementalmente las distribuciones modelo. De tal modo, el servidor continuaría recibiendo los datos de carga útil y el control, optativamente, pasaría a la etapa S252, donde se crearía una nueva distribución de longitudes. Además, una o más realizaciones de la presente invención pueden fijar un marco temporal, para el cual se requeriría al servidor generar una nueva distribución modelo. Así, una vez que el marco temporal se ha agotado, el servidor recogería datos y crearía una nueva distribución de longitudes en la etapa S252, y redefiniría las cargas útiles modelo.

Entrenamiento y calibración automáticos

Según una o más realizaciones, la presente invención puede realizar entrenamiento y calibración automáticos. La presente invención también es capaz de detenerse automáticamente cuando ha sido suficientemente entrenada. Por ejemplo, puede diseñarse un proceso de entrenamiento de modo tal que esté totalmente automatizado. Pueden establecerse una vez un tamaño de era y un umbral, y el sistema decidiría independientemente cuándo ha sido recibido un entrenamiento suficiente. La era corresponde a una longitud de tiempo predeterminada, o a una cantidad de datos predeterminada. Además, el entrenamiento y la calibración pueden ser efectuados en base, por ejemplo, a umbrales especificados por el usuario. Alternativamente, el sistema podría determinar un umbral inicial para cada modelo de carga útil, probando los datos de entrenamiento y escogiendo el máximo valor de distancia, por ejemplo. El número de paquetes capturados para cada era, optativamente, puede ser ajustado por el usuario. Después de cada era de entrenamiento, los nuevos modelos que han sido recién calculados son comparados con los modelos calculados en la era anterior. El entrenamiento acaba cuando los modelos se tornan “estables”.

La Figura 8 es un diagrama de flujo que ilustra la manera en que el servidor puede detectar cuándo han sido recibidos suficientes datos de carga útil para construir la carga útil modelo, según una o más realizaciones de la presente invención. En la etapa S310, el servidor definiría una era actual. Una era corresponde a una longitud de tiempo predeterminada, durante la cual los datos pueden ser, o están siendo, recibidos. En la etapa S312, el servidor recibiría datos de carga útil de la forma normal. En la etapa S314, un modelo actual de carga útil es construido por el servidor. El modelo actual de carga útil corresponde a un modelo de carga útil para todos los datos de carga útil que hayan sido recibidos durante la era actual.

En la etapa S316, el modelo actual de carga útil es comparado con un modelo anterior de carga útil. En consecuencia, durante la primera era, no habría ningún modelo anterior de carga útil con el cual el modelo actual de carga útil pueda ser comparado. En una o más realizaciones de la presente invención, el servidor puede ser dotado de un modelo inicial de carga útil que ha sido recogido anteriormente. Así, durante la primera iteración, el modelo actual de carga útil sería comparado con el modelo inicial de carga útil guardado. La comparación entre el modelo actual de carga útil y el modelo anterior de carga útil puede ser hecha de muchas maneras, que incluyen, por ejemplo, el cálculo de una distancia estadística entre las dos distribuciones distintas.

En la etapa S318, se determina si la distancia entre el modelo actual de carga útil y el modelo anterior de carga útil es menor que un umbral predeterminado. Si la distancia es menor que el umbral predeterminado, entonces se han recogido datos suficientes para construir el modelo de carga útil. El proceso se detiene en la etapa S320. En consecuencia, el modelo actual de carga útil sería usado como la carga útil modelo para comparar los datos entrantes de carga útil. Alternativamente, si la distancia es mayor que el valor de umbral, entonces una nueva era se define en la etapa S322. En la etapa S324, el modelo actual de carga útil es designado como el modelo anterior de carga útil. El control vuelve entonces a la etapa S312, donde el servidor recibe datos de carga útil para la nueva era que ha sido configurada en la etapa S322. El proceso se repite iterativamente hasta que la distancia entre el modelo actual de carga útil y el modelo anterior de carga útil sea menor que el valor de umbral.

Según una o más realizaciones de la presente invención, la estabilidad para cada modelo, para cada puerto, puede ser decidida por dos métricas: la primera es el número de nuevos modelos (antes del agrupamiento) producidos en una era; la segunda es la distancia Manhattan sencilla de cada modelo, después de la era actual, al calculado en la anterior era de entrenamiento. Si ambas métricas están dentro de algún umbral, los modelos se consideran estables y el entrenamiento se detiene. Si se están monitorizando múltiples puertos, puede usarse una métrica adicional. Esta métrica adicional puede examinar el porcentaje de los puertos estables sobre los puertos totales que se están monitorizando. Si el porcentaje de puertos estables es mayor que algún umbral definido por el usuario, concluye todo el proceso de entrenamiento. Los modelos de los puertos “inestables”, optativamente, podrían ser descartados porque no están bien entrenados y no deberían ser usados para la detección de anomalías durante las pruebas.

Una vez que el entrenamiento está completo, puede determinarse un umbral de anomalía. En lugar de usar un umbral universal para todos los centroides, se selecciona un umbral distinto para cada uno. Tales umbrales de resolución fina pueden mejorar la precisión. Esto puede lograrse de diversas maneras. Por ejemplo, el muestreo puede ser realizado

durante la fase de entrenamiento. Las muestras pueden ser usadas luego para ayudar a decidir los umbrales iniciales usados durante el tiempo de detección automáticamente. Durante el proceso de entrenamiento, se mantiene un almacén temporal de muestras de carga útil para cada centroide. Hay un número mínimo y máximo de muestras, y una tasa s% de muestreo. Antes de alcanzar el número mínimo, cada carga útil de paquetes en este contenedor será puesta en muestras. Cada carga útil luego tiene una probabilidad, s, de ser puesta en muestras temporalmente almacenadas. Después de llenar el almacén temporal hasta su tamaño máximo, se usa un almacenamiento temporal del estilo "primero en entrar, primero en salir" (FIFO). La más antigua será eliminada por rotación cuando se inserte una nueva muestra. Después de que se acaba la fase entera de entrenamiento, las muestras son calculadas con respecto al centroide y se usa la máxima distancia como el umbral inicial de anomalía para ese centroide. Debido al muestreo de estilo FIFO, el umbral calculado refleja la tendencia más reciente de la carga útil, y realiza un aprendizaje adaptable para asimilar la deriva conceptual. Esto significa que los modelos, y las calibraciones, son calculados para favorecer el entorno más reciente en el cual ha sido incrustado el sistema.

En el mismo comienzo de las pruebas, la presente invención también puede ejecutarse en eras. Después de cada era, la tasa de alerta generada es comparada con algún número definido por el usuario. Si la tasa total de alerta es demasiado alta, el umbral se incrementará en un t%, y comienza la nueva era. Un ciclo de ese tipo se repite hasta que la tasa de alerta llegue a la tasa deseada. Después de esta fase de calibración, el sistema sería considerado estable y listo para ejecutarse en la modalidad de detección. Debería observarse que el sistema continúa entrenando un nuevo modelo para conservar los modelos refrescados y al día, a fin de reflejar el más reciente entorno.

Detección de cargas útiles anómalas

La Figura 9 es un diagrama de flujo que ilustra las etapas realizadas para detectar cargas útiles anómalas, transmitidas a través de una red, según una o más realizaciones de la presente invención. En la etapa S350, el servidor recibe datos de carga útil desde la red. Esto corresponde a datos que pueden ser recibidos, por ejemplo, desde una red, ya sea externa, interna, inalámbrica o satelital, etc., En la etapa S352, el servidor determina la longitud de los datos contenidos en la carga útil. En la etapa S354, se genera una distribución estadística para los datos contenidos en la carga útil. Por ejemplo, el servidor analizaría los datos contenidos en la carga útil y generaría, por ejemplo, una distribución estadística de los caracteres que aparecen en los datos, según lo expuesto anteriormente. En la etapa S356, la distribución estadística de los datos de carga útil es comparada con una distribución modelo. Por ejemplo, el servidor contendría una pluralidad de distribuciones modelo, según lo anteriormente expuesto, que pueden ser extraídas y aplicadas a tamaños adecuados de cargas útiles. En la etapa S358, el servidor identifica como cargas útiles anómalas aquellas cargas útiles que, por ejemplo, son suficientemente distintas de la distribución modelo, en base a criterios predeterminados del usuario. En consecuencia, cualquier carga útil que sea identificada como anómala sería descartada, o bien adicionalmente analizada.

La Figura 10 es un diagrama de flujo que ilustra la manera en que las cargas útiles anómalas pueden ser detectadas, según una o más realizaciones de la presente invención. En la etapa S410, la carga útil es recibida por el servidor. En la etapa S412, el servidor determina la longitud de los datos contenidos en la carga útil. En la etapa S414, se genera una distribución estadística para los datos de carga útil. Según una o más realizaciones de la presente invención, los datos de carga útil pueden ser distribuidos, por ejemplo, usando distribuciones de n-gramas o hetero-gramas. Esto es ilustrado en la etapa S416. Según una o más realizaciones de la presente invención, diversos factores de peso pueden ser asignados a distintos valores de octeto en la distribución estadística. Esto es ilustrado en la etapa S418. Los diversos factores de peso son seleccionados de modo que los valores de octeto que posiblemente puedan corresponder a códigos de operación de un ordenador, o dispositivo de red, son ponderados en mayor medida y, por tanto, examinados con un mayor escrutinio. Los factores de peso pueden mejorar, al menos en parte, la capacidad del servidor para detectar programas maliciosos, tales como los gusanos, que ejecutan diversos códigos de operación del ordenador o dispositivo. Por ejemplo, los códigos de operación pueden ser códigos de máquina para instrucciones de salto, o para escribir caracteres del idioma correspondientes a operaciones aritméticas, y así sucesivamente.

Según tales realizaciones de la invención, las cargas útiles anómalas con una mayor probabilidad de contener código de ordenador malicioso pueden ser identificadas rápidamente. De tal modo, cuando se genera una alerta para alguna carga útil, esa carga útil, optativamente, puede tener una prueba por separado, para determinar si contiene valores de octeto de interés especial o, alternativamente, el puntaje de una carga útil podría ser cambiado para aumentar su "distancia" desde la distribución normal si contiene valores de octeto "distinguidos". En consecuencia, la distancia de Mahalanobis puede ser modificada para tener en cuenta los valores ponderados, o bien puede usarse una función de distancia distinta que incluya como factores los pesos de ciertos valores de octeto. Al menos algunos de los beneficios de tales realizaciones incluyen: precisión mejorada para identificar código malicioso, reducción de los falsos positivos y asistencia para identificar rápidamente una anomalía de carga útil, como una auténtica gesta del día cero, o un gusano.

En la etapa S420, una distribución modelo es seleccionada por el servidor. La distribución modelo se selecciona de modo que abarque la longitud de los datos contenidos en la carga útil. Por ejemplo, según lo expuesto anteriormente, si una de las distribuciones modelo corresponde a una longitud de carga útil de entre 150 y 155 octetos, entonces cualquier dato de carga útil recibido que tenga una longitud que quede dentro de esa gama sería comparado con esa distribución modelo.

En la etapa 422, se determina si el perfil de la distribución modelo es o no un perfil decadente. Esto puede ocurrir, por ejemplo, en situaciones donde la distribución modelo está dispuesta en un total de frecuencias de octetos, ordenado por categoría. De tal modo, las primeras entradas tendrían un valor mayor, que decae hasta un valor pequeño hacia el final del gráfico.

Como se ha indicado anteriormente, la complejidad de cálculo es lineal con respecto a la longitud de la conexión. Para hacerlo más rápido, el cálculo puede ser iniciado a partir de la cola de la distribución de frecuencias de caracteres, y detenerse inmediatamente si la distancia es mayor que el umbral, tanto para la distancia de Mahalanobis como para la distancia de edición de cadena. La parte de la cola de la distribución son aquellos caracteres que nunca aparecieron en el conjunto de datos de entrenamiento (los de frecuencia cero), o aquellos que aparecieron muy rara vez (una frecuencia muy baja). Si tales caracteres aparecen en los datos de prueba, hay una alta probabilidad de que los datos sean anómalos y, por lo tanto, pueden ser maliciosos. En consecuencia, puede reducirse el tiempo para detectar la conexión maliciosa.

En consecuencia, si la distribución modelo tiene un perfil decadente, entonces, en la etapa S424, el servidor selecciona una opción para calcular la medición de distancia desde el final de la distribución. Alternativamente, si la distribución modelo no tiene un perfil decadente, entonces, en la etapa S426, el servidor selecciona la opción para medir la distancia desde el inicio de la distribución modelo. Las distancias medidas en las etapas S424 y S426 corresponden a la comparación hecha con la distribución modelo para determinar las diferencias entre los datos de carga útil recibidos y la distribución modelo. Como se ha expuesto anteriormente, pueden usarse diversas técnicas para calcular la distancia entre las dos distribuciones. En la etapa S428, se determina si la distancia es mayor que un valor de umbral predeterminado. Por ejemplo, el valor de umbral correspondería a una distancia mínima admitida entre la carga útil recibida y la distribución modelo. Si la distancia es menor que el valor de umbral, entonces el servidor identifica los datos de carga útil como datos normales en la etapa S430. Alternativamente, si el servidor determina que la distancia supera el valor de umbral, entonces la carga útil es identificada como anómala en la etapa S432. Si se considera que la carga útil es una carga útil normal en la etapa S430, entonces es sencillamente dirigida al destino identificado y el proceso acaba en la etapa S444. Sin embargo, para cargas útiles que se determinan como anómalas, el servidor puede realizar pruebas adicionales para identificar diversas características de los datos contenidos en la carga útil.

Por ejemplo, en la etapa S434 el servidor determina si la carga útil corresponde o no a un programa malicioso tal como, por ejemplo, un gusano o virus. Esto puede hacerse de varias maneras. Por ejemplo, el servidor puede comparar diversas características de los datos de carga útil con rúbricas conocidas de gusanos o virus. Alternativamente, los datos de carga útil pueden ser transmitidos a una sede controlada, donde puede permitirse que el programa sea ejecutado, o pueda ser emulado, de modo que pueda determinarse si el programa es efectivamente malicioso.

Según una o más realizaciones de la presente invención, el servidor puede identificar la cadena común más larga, o la sub-secuencia común más larga, hallada en cargas útiles que se consideran anómalas. Si los paquetes o cargas útiles entrantes (o de ingreso) son considerados anómalos, y los paquetes o cargas útiles salientes (o de egreso) son considerados anómalos, y los paquetes entrantes tienen la misma dirección de destino que los paquetes salientes, entonces las cargas útiles pueden ser comparadas para determinar las cadenas comunes más largas, o las sub-secuencias comunes más largas, de ambas cargas útiles anómalas. En base a la cadena común más larga, o a la sub-secuencia común más larga, el anfitrión generaría una rúbrica que identifica al gusano o virus específico, y que sirve como un filtro de contenido para el gusano o virus. Si se determina efectivamente que los datos anómalos no son un gusano o un virus, entonces se descartan en la etapa S436 y el proceso acaba para esos datos específicos de carga útil. Alternativamente, si se determina que la carga útil es un gusano o virus, entonces la rúbrica es generada en la etapa S438. En la etapa S440, todos los patrones de virus o rúbricas de gusanos que han sido generados por el servidor son transmitidos a otros servidores, encaminadores, estaciones de trabajo, etc., para el filtrado de contenidos.

Según una o más realizaciones de la presente invención, el servidor puede ajustar automáticamente el valor del umbral para ayudar y / o mejorar la capacidad de detectar datos anómalos de carga útil. Esto está ilustrado en la etapa S442. Por ejemplo, un procedimiento para ajustar automáticamente el valor del umbral requiere que el servidor fije un umbral de alerta. El umbral de alerta correspondería a un número predeterminado de alertas que el servidor generaría. Cada alerta correspondería a un dato de carga útil anómala. De tal modo, si el umbral de alerta es 100 para un periodo de tiempo de una hora, entonces el servidor ajustaría automáticamente el umbral si no se generan 100 alertas dentro de un periodo de tiempo de una hora. El servidor también puede tener un margen de error, tal como, por ejemplo, $\pm$5 alertas. Por lo tanto, si el servidor genera entre 95 y 105 alertas dentro de un periodo de una hora, no se hace ningún ajuste. Sin embargo, si el servidor genera solamente 80 alertas dentro del periodo de tiempo, esto sugeriría que el valor del umbral es demasiado alto y que la distancia entre las cargas útiles recibidas y las cargas útiles modelo no es lo bastante larga como para superar el umbral. Por lo tanto, el servidor reduciría el valor del umbral de modo que se generara un mayor número de alertas. Alternativamente, si el servidor está generando un mayor número de alertas, tal como 150, entonces el umbral puede ser aumentado, de modo que se generen menos alertas.

La Figura 11 es un gráfico que muestra la distancia de diversos datos de carga útil a las distribuciones modelo. La

pluralidad de los datos de carga útil caen dentro de una distancia predeterminada de la carga útil modelo. Sin embargo, gusano de código rojo tiene una distancia que supera ampliamente el agrupamiento general de las cargas útiles normales recibidas. Por tanto, en esta situación, el servidor identificaría fácilmente el gusano de código rojo como un ataque potencial al servidor.

Para el ejemplo mostrado en la Figura 11, la carga útil efectiva del gusano Código Rojo fue usada como el objetivo de la detección, para mostrar cuán efectiva puede ser esta técnica para detectar el gusano del día cero y los virus. Los datos de entrenamiento fueron rastreados a partir del tráfico de la Red a un servidor de la Red, durante un periodo de tiempo de 24 horas. Las cargas útiles de entrenamiento fueron divididas en distintos subconjuntos, según su división por longitudes, y fueron calculados los modelos normales. La carga útil del Código Rojo fue analizada luego. La distribución de valores de octeto en su carga útil fue calculada, y comparada con la distribución normal de perfiles de carga útil.

El gráfico en la Figura 11 muestra la distancia de Mahalanobis simplificada de las conexiones dentro de la gama de longitudes entre 380 y 385, tanto para las conexiones normales como para el ataque del Código Rojo. Como puede verse, la conexión del ataque del Código Rojo tiene una distancia mucho mayor que todas las otras conexiones normales. En consecuencia, dado un umbral predeterminado, puede ser detectado fácilmente como algo malicioso, y rechazado sin dañar al servidor de la Red. El umbral puede ser fijado durante la fase de entrenamiento. Un valor posible es el máximo de los valores de distancias de los datos de entrenamiento, más alguna tolerancia. Usando esta técnica, el anfitrión puede ser protegido de los virus / gusanos, incluso antes de que se publique cualquier rúbrica de virus.

En lugar de usar las métricas de distancia para calcular la similitud, también puede usarse la “cadena Z de rúbrica de carga útil normal” para lograr el mismo objetivo. Teniendo el perfil “cadena Z de rúbrica” y la cadena de octetos de nuevos datos de conexión a probar, puede usarse una sencilla distancia de edición de cadena para obtener su puntaje de similitud. La distancia de edición de cadena solamente cuenta cuántos caracteres están descolocados con respecto a la cadena de rúbrica del perfil. Una ventaja de la distancia de edición de cadena es el hecho de que no implica ningún cálculo numérico, sino solamente la comparación de equivalencia de cadenas. Esto puede dar como resultado un muy rápido cálculo de distancia.

Uso ejemplar de la invención

Dispositivos de red

Una o más realizaciones de la presente invención pueden ser implementadas en un sistema de ordenador que rastrea pasivamente y audita el tráfico de red en una red de ordenadores, o pueden ser implementadas en el mismo ordenador operando sobre un cortafuegos de red, o pueden ser implementadas en un ordenador anfitrión o servidor, para el cual ha sido calculado un perfil. Una o más realizaciones conciben construir un dispositivo de red capaz de calcular modelos de cargas útiles normales para una multitud de servicios y puertos, para tráfico tanto entrante como saliente. El dispositivo puede distribuir modelos de detección de anomalías a un cortafuegos, para filtrar el tráfico a fin de proteger a cualquier servicio en le red. Alternativamente, el sistema de detección de cargas útiles puede ser implementado en una tarjeta de interfaz de red de un ordenador anfitrión o servidor, sin necesidad de instalar nuevo software en el servidor o anfitrión, o de instalar un nuevo aparato o dispositivo en el sistema de la red.

Aprendizaje incremental

Según una o más realizaciones de la presente invención, un modelo de 1-gramas con distancia de Mahalanobis puede ser implementado como una versión incremental, solamente con poca información más almacenada en cada modelo. Una versión incremental de este procedimiento puede especialmente útil por varios motivos. En primer lugar, un modelo puede ser calculado sobre la marcha en una forma automática de “manos libres”. Ese modelo mejorará en su precisión según el tiempo avanza y más datos son muestreados. Además, una versión incremental en línea también puede “eliminar por antigüedad” los datos viejos del modelo, manteniendo una visión más precisa de las cargas útiles más recientes que fluyen a o desde un servicio.

Una o más realizaciones de la presente invención permiten que los ejemplos más viejos usados en el entrenamiento del modelo sean eliminados por antigüedad. Esto puede ser logrado, al menos en parte, especificando un parámetro de decadencia del modelo más viejo y potenciando las distribuciones de frecuencias que aparecen en las nuevas muestras. Esto permite la actualización automática del modelo para mantener una visión precisa de las cargas útiles normales vistas más recientemente.

El cálculo de la versión incremental de la distancia de Mahalanobis puede lograrse de varias maneras, según la implementación específica de la presente invención. Por ejemplo, la media y la desviación estándar se calculan para cada carácter ASCII visto para cada nueva muestra observada. Para la frecuencia media de un carácter, se calcula $\bar{x} = \sum_{i=1}^{N} x_i / N$ a partir de los ejemplos de entrenamiento. Optativamente, puede almacenarse el número de muestras

procesadas, *N*. Esto permite que la media sea actualizada como $\bar{x} = \frac{\bar{x} \times N + x_{N+1}}{N+1} = \bar{x} + \frac{x_{N+1} - \bar{x}}{N+1}$ cuando se observa una nueva muestra x_{N+1}. Dado que la desviación estándar es la raíz cuadrada de la varianza, el cálculo de la varianza puede ser reescrito usando el valor esperado E, como:

$$Var(X) = E(X - EX)^2 = E(X^2) - (EX)^2$$

Esta desviación estándar puede ser actualizada de manera similar si también se almacena el promedio de los x_i^2 en el modelo.

Según tales realizaciones de la presente invención, solamente es necesario mantener una formación adicional de 256 elementos en cada modelo que almacena el promedio de los x_i^2 y el número total de observaciones *N*. De tal modo, el modelo de distribución de octetos por n-gramas puede ser implementado como un sistema de aprendizaje incremental, fácilmente y muy eficazmente. El mantenimiento de esta información extra también puede ser usado en el agrupamiento de muestras, según se describe en mayor detalle más adelante.

Tamaño reducido del modelo por agrupamiento

Como se ha expuesto anteriormente, un modelo M_{ij} se calcula para cada contenedor *i* de longitudes observadas de las cargas útiles enviadas al puerto *j*. En ciertas circunstancias, una tal modelización de resolución fina podría introducir problemas. Por ejemplo, el tamaño total del modelo puede hacerse muy grande. Esto puede ocurrir cuando las longitudes de carga útil están asociadas a ficheros de medios que pueden ser medidos en gigaoctetos y se definen muchos contenedores de longitudes. Por consiguiente, debe calcularse un gran número de centroides. Además, la distribución de octetos para cargas útiles del contenedor *i* de longitudes puede ser muy similar a la de las cargas útiles de los contenedores *i-1* e *i+1* de longitudes, porque difieren en un octeto. El almacenamiento de un modelo para cada longitud puede a veces ser redundante y derrochador. Otro problema es que, para algunos contenedores de longitudes, puede no haber suficientes muestras de entrenamiento. La poca densidad implica que los datos generarán una distribución empírica que será una estimación imprecisa de la verdadera distribución, lo que lleva, potencialmente, a un detector defectuoso que genera demasiados errores.

El sistema de detección de anomalías de la presente invención proporciona varias soluciones posibles para abordar estos problemas. Según una o más realizaciones de la presente invención, una solución para abordar el problema de la poca densidad es relajar los modelos asignando un mayor factor de allanamiento a las desviaciones estándar. Esto puede permitir una mayor variabilidad de las cargas útiles. Al menos una realización adicional (o solapada) de la invención “pide prestados” datos de los contenedores vecinos para aumentar el número de muestras. En otras palabras, los datos de los contenedores vecinos son usados para calcular otros modelos “similares”. Dos modelos vecinos pueden ser comparados usando la distancia sencilla de Manhattan para medir la similitud de sus distribuciones medias de frecuencias de octetos. Si su distancia es menor que algún umbral *t*, se funden esos dos modelos. Esta técnica de agrupamiento se repite hasta que no puedan ser fundidos más modelos vecinos. Esta fusión también puede ser calculada usando el algoritmo incremental descrito anteriormente. Como se ha expuesto anteriormente, una técnica de ese tipo implica actualizar las medias y varianzas de los dos modelos, para producir una nueva distribución actualizada.

Para nuevos datos de prueba observados, con longitud *i*, enviados al puerto *j*, pueden usarse el modelo M_{ij}, o el modelo con el que fue fundido. Si la longitud de los datos de prueba está fuera de la gama de todos los modelos calculados, entonces se usa el modelo cuya gama de longitudes esté más cerca de la de los datos de prueba. En estos casos, el mero hecho de que la carga útil tenga una tal longitud inusual, no observada durante el entrenamiento, puede ser por sí mismo la causa para generar una alerta.

Debería observarse que tanto el algoritmo de modelación como el proceso de fusión de modelos son cálculos temporales lineales y, por tanto, la técnica de modelación es muy rápida y puede ser realizada en tiempo real. Adicionalmente, el algoritmo de aprendizaje en línea asegura que los modelos mejorarán con el tiempo, y que su precisión se mantendrá incluso cuando los servicios sean cambiados y se observen nuevas cargas útiles.

Tráfico correlacionado de ingreso y de egreso para detectar la propagación de gusanos y generar rúbricas

La auto-propagación es una característica clave y una condición necesaria para los gusanos. La auto-propagación significa que, una vez que un gusano infecta una máquina, comenzará a atacar otras máquinas automáticamente, intentando enviar una copia de sí mismo, o una variante de la misma, a otro anfitrión susceptible. Por ejemplo, si una máquina queda infectada por el gusano Código Rojo II a partir de alguna solicitud recibida en el puerto 80, entonces esta máquina comenzará a enviar la misma solicitud al puerto 80 de otras máquinas. Tal patrón de propagación es verdadero para casi todo gusano. Por lo cual, si puede ser detectado algún tráfico de egreso al puerto *i* que sea muy similar a esos tráficos anómalos de ingreso al puerto *i*, hay una alta probabilidad de que un gusano dirigido al puerto *i* esté propagándose.

Según una o más realizaciones de la presente invención, el tráfico malicioso entrante puede ser detectado, y una alerta puede ser generada. Al mismo tiempo, la carga útil puede ser proporcionada como una cadena en el almacén temporal para el puerto *i*, y comparada con el tráfico saliente, con respecto a todas las cadenas, para ver cuáles devuelven el más alto puntaje de similitud. Si el puntaje es mayor que algún umbral predeterminado, se supone una posible propagación de gusano. Además, la presente invención puede ser implementada en una máquina servidora, tal como, por ejemplo, un servidor de la Red. Los servidores de la Red, en general, tienen una gran cantidad de solicitudes entrantes, pero las solicitudes salientes son habitualmente improbables. Por lo cual, cualquier solicitud saliente ya es bastante sospechosa, y debería ser comparada con las cadenas maliciosas. Si la máquina anfitriona está funcionando a la vez como servidor y cliente, lo que significa que son comunes tanto las solicitudes entrantes como las solicitudes salientes, la misma técnica de modelación sería aplicada al tráfico saliente, y usada solamente para comparar el tráfico de egreso ya juzgado como malicioso.

Una o más realizaciones de la presente invención también proporcionan múltiples métricas que pueden ser usadas para decidir la similitud entre dos cadenas. Las más comunes son la sub-cadena común más larga (LCS) o la sub-secuencia común más larga (LCSec). La diferencia entre ellas es: la sub-cadena común más larga es contigua, mientras que la sub-secuencia común más larga no necesariamente lo es. LCSec tiene la ventaja de poder detectar gusanos “polimórficos” y “metamórficos”; pero pueden introducir falsos positivos. Otras técnicas, tales como los procedimientos de modelación de probabilidad que tienen en cuenta los tamaños de sub-cadena dependientes del contexto, también pueden ser aplicadas por la presente invención. El puntaje de similitud devuelto es el porcentaje de la longitud de la parte común dentro de la longitud total de la cadena de carga útil maliciosa. Una implementación alternativa (o solapada) para calcular una rúbrica sería calcular el conjunto de (al menos una de) las sub-cadenas que aparezcan más frecuentemente dentro de la carga útil que aparece en dos o más ejemplos de datos anómalos.

Según una o más realizaciones adicionales (o solapadas), la presente invención puede ser usada para generar automáticamente rúbricas de gusano. Calculando el puntaje de similitud, la sub-cadena o sub-secuencia coincidente, que representa la parte común del tráfico malicioso de ingreso y de egreso, también son calculadas. Dado que el tráfico que está siendo comparado ya está juzgado como malicioso, lo que significa que la carga útil es bastante distinta a las normales, estas cadenas comunes representan la rúbrica del contenido del gusano. Por tanto, correlacionando la carga útil maliciosa de ingreso y de egreso, la presente invención es capaz de detectar la muy inicial propagación del gusano, y de identificar su rúbrica o rúbrica parcial inmediatamente, sin ninguna implicación externa. Esto ayuda a resolver el problema del gusano del día cero. Tales rúbricas pueden luego ser comunicadas a otros anfitriones y dispositivos para el filtrado de contenidos, a fin de eliminar toda aparición adicional del gusano que infecte a cualquier otro anfitrión en la red.

Detección de gusanos más precisa por seguridad colaborativa

Según una o más realizaciones de la presente invención, el sistema de detección de anomalías puede ser implementado en múltiples anfitriones y dispositivos en una red. Los anfitriones y dispositivos pueden luego colaborar entre sí, por ejemplo, intercambiando alertas y posibles rúbricas de gusanos. En consecuencia, un gusano puede ser rápidamente identificado, e impedirse su extensión, porque múltiples anfitriones se informan mutuamente de la misma rúbrica de gusano. La rúbrica puede luego ser anunciada a fin de aplicar el filtrado de contenidos rápidamente por toda la red. Usando tales estrategias colaborativas de seguridad, es posible reducir la probabilidad de que los gusanos puedan extenderse a lo largo y a lo ancho, y ocupar la red.

Identificación de ficheros

Hay varias complicaciones que pueden resultar de los entornos en red, algunos referidos al hecho de que la red puede ser usada en un entorno de oficina. Para complicar estos problemas, están las altas velocidades a las cuales pueden ser transmitidos los datos por múltiples redes. En consecuencia, el personal de operaciones y / o de seguridad de una red puede desear conocer cómo son efectivamente usados los ordenadores en la red, y qué tipos de datos son comunicados entre los anfitriones dentro y fuera de la red. Esto puede entrañar, por ejemplo, determinar los tipos de ficheros y medios transmitidos entre ordenadores (y usuarios) dentro de la red. Si bien la mayoría de las transmisiones son generalmente inofensivas, a veces pueden proporcionar un camino para esparcir programas maliciosos, tales como virus y gusanos. Además, algunos empleadores mantienen información confidencial y / o personal que no debería ser diseminada fuera del lugar de trabajo. Tales empleadores a menudo promulgan políticas que advierten a los empleados no transmitir ciertos ficheros y / o información a ordenadores fuera de la red de la compañía. También puede ser el caso que los empleadores no deseen que ciertos tipos de ficheros sean recibidos desde (o transmitidos a) redes u ordenadores externos.

A fin de imponer algunas de estas políticas, el tráfico a través de la red de la compañía está habitualmente monitorizado, de modo que ciertos ficheros puedan ser bloqueados o examinados. Por ejemplo, un anexo “Patente25.doc” de correo electrónico puede ser examinado si los documentos de Word no deberían ser transmitidos a ordenadores externos. Sin embargo, es relativamente sencillo enmascarar (u ocultar) el verdadero tipo de un fichero, cambiando sencillamente la extensión asociada al nombre del fichero. Por tanto, un usuario puede evadir fácilmente la política de seguridad, cambiando el nombre del fichero, de Patente25.doc a FotosVacaciones.jpg, por ejemplo. Alternativamente, el fichero

podría recibir un sufijo distinto, indicativo, p. ej., de un fichero de imagen, y transmitido fuera de la red. Una vez recibido, el destinatario podría renombrar el fichero como Patente25.doc y abrirlo, por ejemplo, con el Word de Microsoft. Y al contrario, un anexo de correo electrónico entrante puede ser un virus o gusano que ha sido renombrado, por ejemplo, como Patente25.doc. Una vez abierto, el virus podría causar daño al sistema de ordenador.

La Figura 12 es un diagrama de bloques que ilustra ciertas situaciones donde los ficheros pueden ser transmitidos a un usuario furtivamente, o con falsas pretensiones. Por ejemplo, la estación 114 de trabajo está conectada con una red tal como Internet 116. Tres distintos anexos de ficheros han sido transmitidos a la estación de trabajo. El primer anexo 150 de fichero tiene el nombre “hello.doc”. Se supone que este fichero es un fichero de Word de Microsoft. Sin embargo, no es ese el caso. El fichero, en verdad, es un virus (sobig.exe) que ha sido renombrado para que parezca un documento de Word. El segundo anexo 152 de fichero se titula “junk.zip”. Este fichero puede no estar necesariamente renombrado, pero tiene la forma de un fichero de archivo que puede contener múltiples contenidos archivados. Los contenidos archivados no pueden ser vistos hasta que se acceda a, o se abra, el fichero 152 de archivo. Hay situaciones donde un sistema operativo o programa de correo puede acceder automáticamente al fichero 152 de archivo en cuanto es recibido. Así, si un virus está contenido dentro del fichero 152 de archivo, puede ser liberado automáticamente. El tercer anexo 154 se titula “document.pdf”, para que no sea identificado como un fichero de Word. Sin embargo, el fichero se llamaba originalmente “contract.doc”. Todos los ficheros pueden presentar problemas potenciales a la estación 114 de trabajo.

La Figura 13 ilustra un procedimiento de identificación de tipos de fichero transmitidos a través de una red, según una o más realizaciones de la presente invención. En la etapa S510, se recibe una transmisión a través de la red. La transmisión puede ser un mensaje convencional de correo electrónico, y puede incluir varios tipos de información, tales como, por ejemplo, texto, anexos, etc. En la etapa S512, se determina si la transmisión contiene o no algún fichero. Los ficheros pueden ser incluidos en la transmisión como parte de un anexo para el mensaje de correo electrónico. Si la transmisión no incluye fichero alguno, entonces el control avanza a la etapa S526, donde el proceso termina. Sin embargo, si se determina que la transmisión contiene uno o más ficheros, entonces, en la etapa S514, se genera una distribución estadística para los datos contenidos en cada uno de los ficheros en la transmisión.

En la etapa S516, se selecciona una distribución modelo. La distribución modelo corresponde a una o más distribuciones estadísticas que han sido generadas para tipos de ficheros predeterminados. Por ejemplo, una distribución específica de ficheros modelo podría ser generada para un fichero .gif. De manera similar, una distribución modelo podría ser generada para un fichero .pdf, un fichero .doc, un fichero .jpeg, etc., usando conceptos previamente descritos y / o los descritos más adelante. Con referencia adicional a las Figuras 14A a 14I, se ilustran las distribuciones modelo de diversos tipos de ficheros. En la etapa S518, la distancia entre la distribución estadística para el fichero es medida con respecto a la distribución modelo. Según lo expuesto anteriormente, pueden ser usados diversos procedimientos para medir la distancia, incluyendo, pero sin limitación, la distancia de Mahalanobis. Adicionalmente, si la transmisión contiene más de un fichero, entonces se aplicaría la prueba de distancia a cada fichero contenido en la transmisión. En la etapa S520, se determina si la distancia entre el fichero recibido y la distribución modelo es o no mayor que un umbral predeterminado. Si esta distancia es mayor que el umbral, entonces el control avanza a la etapa S522. Si la distancia es menor que el umbral, entonces el control avanza a la etapa S524. En este punto, el fichero recibido puede ser identificado como de un tipo específico. Por ejemplo, el tipo para la distribución modelo ya es conocido. Por tanto, puede determinarse que el fichero recibido sea del mismo tipo que la distribución modelo. El control avanzaría entonces a la etapa S526, donde el proceso termina.

En la etapa S522, se determina si hay o no distribuciones modelo adicionales. Si no hay más distribuciones modelo adicionales, entonces el proceso también termina sin haber identificado, o haber podido identificar, el tipo del fichero recibido. Sin embargo, si hay distribuciones modelo adicionales disponibles, entonces el control vuelve a la etapa S516, donde se selecciona la siguiente distribución modelo. El proceso continuaría hasta que los ficheros recibidos sean probados con respecto a todas las distribuciones modelo, y se determine un tipo, o bien no pueda determinarse un tipo. Según una o más realizaciones de la presente invención, si el tipo para el fichero no puede ser determinado, entonces el fichero puede ser descartado o identificado como un virus potencial o un programa malicioso.

La Figura 15 es un diagrama de flujo que ilustra un procedimiento para modelar tipos de ficheros, según una o más realizaciones de la presente invención. En la etapa S550, se recogen una pluralidad de ficheros. Se sabe que los ficheros tienen un tipo específico y / o que son creados como de tales tipos. Por ejemplo, puede recogerse una pluralidad de ficheros .pdf, o una pluralidad de ficheros .doc, ficheros .jpeg, ficheros .gif, etc., Mientras todos los ficheros sean del mismo tipo, pueden ser usados para generar el modelo adecuado. En la etapa S552, se genera una distribución estadística para cada uno de los ficheros que han sido recogidos. En la etapa S554, se combinan las distribuciones estadísticas. Esto puede lograrse de una amplia variedad de maneras, incluyendo, por ejemplo, la sencilla adición de la distribución para cada fichero recogido.

En la etapa S556, se forma una pluralidad de agrupamientos para las distribuciones estadísticas. En la etapa S558, se calcula un centroide para cada agrupamiento formado para las distribuciones estadísticas. En la etapa S560, se calcula un centroide modelo. El centroide modelo corresponde al centroide de la pluralidad de centroides de agrupamientos

calculados en la etapa S558. En la etapa S562, el centroide modelo es designado como el modelo para representar el tipo de fichero específico. En consecuencia, si están siendo modelados ficheros .pdf, entonces el centroide modelo correspondería a una distribución modelo para ficheros .pdf. En la etapa S566, el proceso termina. Según una o más realizaciones de la presente invención, el tipo de fichero modelo también puede basarse en la distribución estadística combinada para todos los ficheros que han sido recogidos. Esto está ilustrado en la etapa S564. De tal modo, las distribuciones estadísticas combinadas del fichero recogido serían asignadas como la distribución modelo para el tipo de fichero específico.

La Figura 16 es un diagrama de flujo que ilustra las etapas realizadas para verificar tipos de ficheros según una o más realizaciones de la presente invención. En la etapa S610, se recibe el fichero. El fichero puede ser recibido desde cualquiera entre una pluralidad de orígenes, incluyendo, por ejemplo, las transmisiones generales de red, el correo electrónico o los medios portátiles. En la etapa S612, una distribución estadística es generada para el fichero. En la etapa S614, se extrae la distribución modelo correspondiente al tipo de fichero recibido. Por ejemplo, si el tipo de fichero recibido está etiquetado (o designado usando una extensión específica) como un fichero .jpeg, entonces se extraería la distribución modelo adecuada para un fichero .jpeg. En la etapa S616, la distribución estadística para el fichero recibido es comparada con la distribución modelo extraída en la etapa S614. En la etapa S618, se determina si la distribución estadística para el fichero recibido está o no dentro del límite de tolerancia de la distribución modelo. Más específicamente, la distancia entre la distribución estadística para el fichero recibido y la distribución modelo es revisada para determinar si cae o no dentro del umbral de tolerancia.

Si la distancia para la distribución estadística para el fichero recibido está dentro de la tolerancia, entonces el fichero puede ser confirmado como del tipo especificado en el nombre de fichero. Esto se ilustra en la etapa S620. El proceso terminaría así al confirmar el tipo del fichero recibido. Alternativamente, si la distancia de la distribución estadística para el fichero extraído no está dentro de la tolerancia, entonces puede ser generada una alerta para indicar que el fichero no es efectivamente del tipo especificado en el nombre de fichero. Esto está indicado en la etapa S622. En la etapa S624, el fichero puede ser bloqueado, o bien descartado, para transmisiones adicionales a través de la red o estación de trabajo.

Según una o más realizaciones de la presente invención, al detectar que un fichero está nombrado inadecuadamente, y que corresponde a un tipo de fichero distinto, pueden realizarse pruebas adicionales para determinar si el fichero es efectivamente un virus fingiendo ser de un tipo de fichero distinto. El control volvería entonces a la etapa S624, donde el fichero puede nuevamente ser bloqueado o descartado para la propagación posterior a través de la red. El proceso termina entonces en la etapa S628.

La Figura 17 es un diagrama de flujo que ilustra las etapas realizadas para detectar y / o identificar programas maliciosos, tales como virus y gusanos, según una o más realizaciones de la presente invención, En la etapa S650, una transmisión es recibida a través de la red, o en una estación de trabajo. La transmisión puede ser una transmisión por una red entre múltiples ordenadores, dentro de la red, etc. Adicionalmente, la transmisión puede corresponder a transmisiones internas de datos dentro de una única máquina. Por ejemplo, la transmisión puede corresponder a la lectura de un fichero desde un medio portátil, hacia la memoria de la estación de trabajo.

En la etapa S652, se determina si hay o no algún fichero adosado a la transmisión. Si ningún fichero está adosado a la transmisión, entonces el proceso termina. Si algún fichero está presente en la transmisión, entonces el control avanza a la etapa S654. Se extrae la información con respecto al tipo de cada fichero. La información puede ser extraída, por ejemplo, examinando la extensión en el nombre de fichero. En la etapa S656, se genera una distribución estadística para el fichero. En la etapa S658, se extrae la distribución modelo correspondiente al tipo del fichero. En la etapa S660, la distribución estadística para el fichero es comparada con la distribución modelo extraída. En la etapa S662, se determina si la distribución estadística para el fichero está dentro del umbral de tolerancia. Si es así, entonces, probablemente, el fichero no es un virus y sería identificado como tal en la etapa 664. Sin embargo, si una distribución estadística para el fichero no está dentro de la tolerancia, entonces el fichero es identificado como un virus en la etapa S666. En la etapa S668, la distribución estadística para el fichero puede ser comparada con cualquier distribución estadística de virus conocidos.

Según una o más realizaciones de la presente invención, diversos factores de peso pueden ser asignados a distintos valores de octeto dentro de la distribución estadística. Esto está ilustrado en la etapa S670. Como se ha expuesto anteriormente, pueden ser asignados mayores factores de peso a valores de octeto que pueden corresponder, posiblemente, a códigos de ejecución de máquina, ficheros guión y / u otros programas que puedan causar daño a la máquina. En la etapa S672, se determina si la distribución estadística para el fichero coincide o no con alguna de las distribuciones de virus. Si hay una coincidencia, entonces el tipo de virus es identificado en la etapa S674. Si no se halla ninguna coincidencia, entonces el control avanza a la etapa S676. Los datos contenidos en el fichero son examinados a fin de identificar información concerniente al virus. En la etapa S678, son identificadas cadenas o sub-secuencias comunes cualesquiera dentro del fichero. En la etapa S680, las cadenas o sub-secuencias comunes son usadas para generar una rúbrica para el virus. En la etapa S684, el proceso termina. Según una o más realizaciones de la presente invención, en lugar de examinar los datos en el fichero para generar una rúbrica, la distribución estadística para el fichero puede ser usada como una cadena (o distribución) de rúbrica. Esto está ilustrado en la etapa S682, donde se genera una

rúbrica basada en la distribución para el fichero (es decir, el virus identificado).

Rastreo del origen de transmisión

Según una o más realizaciones, la presente invención puede ser usada para abordar varios problemas asociados al uso de grandes redes, tales como Internet. Un problema de ese tipo implica el uso de agentes de peldaño. Estos agentes son usados por atacantes (o piratas) para ocultar sus verdaderas ubicaciones, mientras lanzan ataques contra diversas máquinas. A menudo, un atacante iniciará el ataque desde una máquina "controlada remotamente" que ha sido previamente pirateada y cuyo control ha sido arrebatado. Estas máquinas controladas remotamente pueden posteriormente lanzar ataques de denegación de servicio sobre diversos ordenadores comerciales, servidores, sedes de la Red, etc. Además, el atacante puede hacer que una máquina controlada remotamente active una segunda máquina controlada remotamente, iniciando por ello un ataque. Una vez que el ataque está iniciado, el ordenador de destino solamente ve información de la máquina que transmite el comando de ataque.

Como el atacante ha tomado el control del ordenador controlado remotamente, el ordenador de destino solamente vería la dirección de IP del ordenador controlado remotamente que causa el ataque. Los piratas pueden usar múltiples niveles, o agentes de peldaño, desde los cuales lanzar tales ataques. Esto hace que sea cada vez más difícil rastrear la ubicación del atacante efectivo. Para complicar adicionalmente la situación, pueden darse a los ordenadores controlados remotamente horas específicas para tomar contacto automáticamente con otro ordenador controlado remotamente y / o iniciar un ataque.

La Figura 18 ilustra un tipo de situación de peldaño. El atacante 200 inicia un ataque contra un ordenador 250 de destino. El ordenador de destino puede estar en la misma vecindad, país o estado del atacante. Sin embargo, el atacante también puede estar situado en cualquier parte del mundo donde se proporciona una conexión a Internet. Según la situación en la Figura 16, el atacante ha tomado el control de cuatro ordenadores controlados remotamente. Estos incluyen el ordenador 210 controlado remotamente de la etapa 1, el ordenador 220 controlado remotamente de la etapa 2, el ordenador 230 controlado remotamente de la etapa 230 y el ordenador 240 controlado remotamente de la etapa 4. Todos estos ordenadores controlados remotamente están bajo el control del atacante. Como se ha expuesto anteriormente, durante las conexiones normales de red, una máquina solamente puede ver información transmitida desde la máquina inmediatamente anterior. Por ejemplo, el ordenador 250 de destino, que es el destino último del ataque, solamente ve información transmitida desde el ordenador 240 controlado remotamente de la etapa 4. De tal modo, el ordenador 250 de destino cree que un ataque está siendo lanzado desde el ordenador 240 controlado remotamente de la etapa 4. Análogamente, el ordenador 240 controlado remotamente de la etapa 4 ve información relacionada con el ordenador 230 controlado remotamente de la etapa 3. Yendo hacia atrás, el ordenador 230 controlado remotamente de la etapa 3 ve un ataque iniciado por el ordenador 220 controlado remotamente de la etapa 2. El ordenador 220 controlado remotamente de la etapa 2 ve un ataque iniciado por el ordenador 210 controlado remotamente de la etapa 1. El único ordenador dentro del enlace de conexión que conoce la verdadera dirección del atacante 200 es el ordenador 210 controlado remotamente de la etapa 1.

Según una o más realizaciones de la presente invención, la ubicación del atacante puede ser determinada analizando la distribución estadística para las cargas útiles de datos transmitidas a través de los múltiples ordenadores controlados remotamente. Los ordenadores controlados remotamente están conectados entre sí por una red, mediante un cierto número de proveedores 260 de servicios. Cada proveedor 260 de servicios mantiene un registro 270 de conexiones que contiene información con respecto a las transmisiones por la red. El registro 270 de conexiones puede incluir, por ejemplo, una dirección 272 de IP del sistema de ordenador que transmite información, la dirección 274 de destino del sistema de ordenador donde será entregada la información, y la información efectiva 276 que está siendo entregada. A fin de minimizar la cantidad de información contenida en el registro de conexiones, una distribución estadística puede ser generada para cada carga útil 276 de datos que se transmite. Así, la distribución estadística puede ser configurada de modo que sea almacenada dentro de una cadena corta, por ejemplo, de 256 octetos, según lo anteriormente expuesto con respecto a diversas realizaciones de la invención. Esto permite al proveedor 260 de servicios capturar y almacenar información con respecto al vasto número de transmisiones que circulan, sin desperdiciar espacio de almacenamiento. Como se expondrá en mayor detalle más adelante, la información mantenida por el proveedor de servicios puede ser usada para rastrear la ubicación física del atacante que inicia el ataque sobre la máquina de destino. Además, la distribución estadística puede ser generada para todo el registro 270 de conexiones, o solamente una parte del mismo.

La Figura 19 es un diagrama de flujo que ilustra las etapas realizadas para rastrear el origen de un mensaje transmitido, según una o más realizaciones de la presente invención. En la etapa S710, los registros de conexiones son creados por el proveedor de servicios. Según lo anteriormente expuesto, los registros de conexiones pueden incluir, por ejemplo, una dirección de un sistema de ordenador anterior, una carga útil de datos y una dirección para un sistema de ordenador posterior. En la etapa S712, los registros de conexiones son examinados, y las distribuciones estadísticas son generadas para los datos contenidos en cada registro de conexiones. En la etapa S714, una carga útil sospechosa es identificada en un ordenador de destino final. Más específicamente, la carga útil de datos sospechosos puede corresponder, por ejemplo, a un programa malicioso que fue usado para infectar, o bien iniciar un ataque sobre, el sistema de ordenador de destino.

En la etapa S716, una distribución estadística es generada para la carga útil de datos sospechosos. En la etapa S718, el ordenador de destino final es designado como ordenador sospechoso.

En la etapa S720, la distribución estadística para la carga útil de datos sospechosos es comparada con las distribuciones estadísticas de las cargas útiles de datos generadas en la etapa S712. En la etapa S722, se determina si la distancia de la distribución de la carga útil de datos sospechosos está o no dentro del umbral de tolerancia para la distribución del registro actual de conexiones. Si está dentro de la tolerancia, entonces puede identificarse una coincidencia. Si la distribución estadística para la carga útil sospechosa no está dentro de la tolerancia, entonces, en la etapa S724, se determina si hay o no registros adicionales de conexiones. Si hay registros adicionales de conexiones, entonces el control vuelve a la etapa S720, donde se hace una comparación con el próximo registro de conexiones. Si no hay más registros de conexiones, entonces el proceso terminará. Sin embargo, si la distribución estadística para la carga útil sospechosa está dentro de la tolerancia, entonces, en la etapa S726, se identifica la identidad del remitente anterior. Esto puede hacerse, por ejemplo, examinando el registro de conexiones a partir del cual fue generada la distribución. Dentro de los registros de conexiones son identificadas las direcciones de los sistemas de ordenadores remitente y destinatario. De tal modo, el sistema de ordenador sospechoso sería la dirección de destino y se identificaría la dirección del remitente anterior.

En la etapa S728, se determina si el sistema de ordenador anterior es o no el remitente original de la transmisión. Si el sistema de ordenador anterior es el remitente original de la transmisión, entonces se obtiene la identidad del remitente original y el proceso termina. Sin embargo, si la dirección del remitente anterior no corresponde al remitente original de la transmisión, entonces el control avanza a la etapa S732. El ordenador anterior es designado ordenador sospechoso. El control entonces retorna a la etapa S720, donde la distribución estadística para la carga útil sospechosa es comparada con la distribución estadística para los registros de conexiones almacenados por el ordenador sospechoso recientemente designado. El proceso puede repetirse hacia atrás a través de múltiples sistemas de ordenador, hasta que sea identificado el remitente original de la transmisión.

El sistema de detección de anomalías de la presente invención también puede ser implementado en ordenadores y servidores que usan diversos sistemas operativos, tales como la línea Windows de sistemas operativos, Linux, MacOS, etc. El código de programa necesario también puede ser producido usando cualquier lenguaje de programación, tal como C++, C#, Java, etc.

Las muchas características y ventajas de la invención son evidentes a partir de la especificación detallada y, por tanto, las reivindicaciones adjuntas están concebidas para cubrir todas las características y ventajas de ese tipo que caigan dentro del alcance de la invención. Además, dado que numerosas modificaciones y variaciones devendrán inmediatamente evidentes para los expertos en la técnica, la invención no debería estar limitada a la construcción y operación exactas ilustradas y descritas. En cambio, todas las modificaciones adecuadas, y los equivalentes, pueden considerarse como incluidos dentro del alcance de la invención reivindicada.

REIVINDICACIONES

1. Un procedimiento, llevado a cabo por un ordenador, de detección de cargas útiles anómalas transmitidas a través de una red, que comprende las etapas de:

recibir al menos una carga útil dentro de la red (S250; S350);

determinar una longitud para los datos contenidos en la al menos una carga útil (S252; S352);

generar una distribución estadística de valores de octeto de los datos contenidos en la al menos una carga útil recibida dentro de la red (S256; S354);

seleccionar, de entre una pluralidad de distribuciones estadísticas modelo de valores de octeto, una distribución estadística modelo de valores de octeto representativa de las cargas útiles normales transmitidas a través de la red, en base, al menos en parte, a la longitud determinada, en donde la distribución estadística modelo de valores de octeto tiene una gama de longitudes predeterminada y se selecciona de modo que la longitud determinada para los datos contenidos en la al menos una carga útil esté incluida dentro de la gama de longitudes predeterminada;

comparar al menos una parte de la distribución estadística generada de valores de octeto con una parte correspondiente de la distribución estadística modelo seleccionada de valores de octeto (S356); e

identificar si la al menos una carga útil es una carga útil anómala (S358), en base, al menos en parte, a las diferencias detectadas entre la al menos una parte de la distribución estadística generada de valores de octeto para la al menos una carga útil y la parte correspondiente de la distribución estadística modelo seleccionada de valores de octeto.

2. El procedimiento de la reivindicación 1, en el cual la gama de longitudes predeterminada de la distribución estadística modelo seleccionada de valores de octeto está basada en divisiones seleccionadas usando estimaciones de núcleo para la distribución estadística modelo seleccionada de valores de octeto de todas las cargas útiles normales (S254).

3. El procedimiento de la reivindicación 1, en el cual la gama de longitudes predeterminada de la distribución estadística modelo seleccionada de valores de octeto está basada en divisiones seleccionadas aplicando al menos un algoritmo de agrupamiento a la distribución estadística modelo seleccionada de valores de octeto de todas las cargas útiles normales.

4. El procedimiento de la reivindicación 1, en el cual la distribución estadística generada de valores de octeto de la al menos una carga útil es una distribución de valores de octeto de la frecuencia media y la varianza de los datos contenidos en la al menos una carga útil, y la distribución estadística modelo seleccionada de valores de octeto es una distribución de valores de octeto de la frecuencia media y la varianza, representativa de las cargas útiles normales.

5. El procedimiento de la reivindicación 1, en el cual la distribución estadística generada de valores de octeto de la al menos una carga útil es una distribución de valores de octeto de los datos contenidos en la al menos una carga útil, y la distribución estadística modelo seleccionada de valores de octeto es una distribución de valores de octeto representativa de las cargas útiles normales.

6. El procedimiento de la reivindicación 5, en el cual la distribución estadística generada de valores de octeto de la al menos una carga útil es un total de frecuencias de octetos de los datos contenidos en la al menos una carga útil, y la distribución estadística modelo seleccionada de valores de octeto es un total de frecuencias de octetos para cargas útiles normales.

7. El procedimiento de la reivindicación 5, en el cual la distribución estadística generada de valores de octeto de la al menos una carga útil es un total de frecuencias de octetos, ordenado por clasificación, de los datos contenidos en la al menos una carga útil, y la distribución estadística modelo seleccionada de valores de octeto es un total de frecuencias de octetos, ordenado por clasificación, para cargas útiles normales.

8. El procedimiento de la reivindicación 1, en el cual:

la etapa de comparación comprende adicionalmente una etapa de medir una métrica de distancia entre la distribución estadística generada de valores de octeto de los datos contenidos en dicha(s) carga(s) útil(es) y la distribución estadística modelo seleccionada de valores de octeto (S518); y

la etapa de identificación comprende adicionalmente una etapa de identificar cargas útiles anómalas como las cargas útiles que, al menos en parte, superan una métrica de distancia predeterminada (S520).

9. El procedimiento de la reivindicación 8, en el cual la métrica de distancia se calcula en base a una distancia de Mahalanobis entre la distribución estadística generada de valores de octeto de los datos contenidos en la al menos una carga útil y la distribución estadística modelo seleccionada de valores de octeto.

10. El procedimiento de la reivindicación 8, en el cual la distribución estadística generada de valores de octeto de los datos contenidos en la al menos una carga útil tiene un perfil decadente, y en el cual la medición de la métrica de distancia es iniciada en el extremo decadente.

11. El procedimiento de la reivindicación 8, que comprende adicionalmente las etapas de:

fijar un umbral de alerta correspondiente a un número deseado de alertas para cargas útiles anómalas detectadas; y ajustar automáticamente la métrica de distancia predeterminada hasta que se alcance el umbral de alerta.

12. El procedimiento de la reivindicación 1, en el cual la gama de longitudes predeterminada es seleccionada a partir de al menos una parte de prefijo de la distribución estadística modelo seleccionada de valores de octeto.

13. El procedimiento de la reivindicación 1, en el cual la gama de longitudes predeterminada es seleccionada a partir de al menos una parte de sufijo de la distribución estadística modelo seleccionada de valores de octeto.

14. El procedimiento de la reivindicación 1, en el cual la gama de longitudes predeterminada es seleccionada a partir de al menos una parte central de la distribución estadística modelo seleccionada de valores de octeto.

15. El procedimiento de la reivindicación 1, en el cual la gama de longitudes predeterminada de la distribución estadística modelo seleccionada de valores de octeto es seleccionada a partir de un grupo de divisiones que consiste en: de 0 a 50 octetos, de 50 a 150 octetos, de 150 a 155 octetos, de 0 a 255 octetos, menos de 1.000 octetos, más de 2.000 octetos y más de 10.000 octetos.

16. El procedimiento de la reivindicación 1, en el cual al menos una entre la distribución estadística generada de valores de octeto para la al menos una carga útil, la pluralidad de distribuciones estadísticas modelo de valores de octeto y la distribución estadística modelo seleccionada de valores de octeto se genera usando una distribución en n-gramas, en donde cada n-grama es una agrupación variable de octetos.

17. El procedimiento de la reivindicación 16, en el cual n es un valor mixto de agrupaciones de octetos.

18. El procedimiento de la reivindicación 16, en el cual n = 1.

19. El procedimiento de la reivindicación 16, en el cual n = 2.

20. El procedimiento de la reivindicación 16, en el cual n = 3.

21. El procedimiento de la reivindicación 1, que comprende adicionalmente una etapa de determinar si una carga útil anómala detectada es o no un virus o gusano (S434).

22. El procedimiento de la reivindicación 21, que comprende adicionalmente las etapas de:

identificar un puerto que haya sido sondeado por el virus o gusano;

comparar el tráfico de egreso con la distribución estadística generada de valores de octeto para el virus o gusano, a fin de detectar sondeos al mismo puerto en otras máquinas; y

detectar la propagación del gusano o virus en base a la etapa de comparación.

23. El procedimiento de la reivindicación 21, que comprende adicionalmente una etapa de asignar distintos factores de peso a valores de octeto seleccionados.

24. El procedimiento de la reivindicación 23, en el cual mayores factores de peso son asignados a valores de octeto correspondientes a códigos operativos de un sistema de ordenador.

25. El procedimiento de la reivindicación 21, que comprende adicionalmente una etapa de generar una rúbrica para cualquier carga útil anómala determinada como virus o gusano (S682).

26. El procedimiento de la reivindicación 25, en el cual la etapa de generar una rúbrica comprende adicionalmente las etapas de: identificar la cadena común más larga para los datos contenidos en la carga útil determinada como virus o gusano; y

generar la rúbrica en base, al menos en parte, a la cadena común más larga.

27. El procedimiento de la reivindicación 25, en el cual la etapa de generación comprende adicionalmente las etapas de:

identificar la sub-secuencia común más larga para los datos contenidos en la carga útil determinada como el virus o gusano; y

generar la rúbrica en base, al menos en parte, a la sub-secuencia común más larga.

28. El procedimiento de la reivindicación 25, que comprende adicionalmente una etapa de intercambiar rúbricas de virus o gusanos con uno o más servidores.

29. Un sistema para detectar cargas útiles anómalas transmitidas a través de una red, que comprende:

un ordenador acoplado a la red, y que recibe al menos una carga útil a través de la red; y

una o más distribuciones estadísticas modelo de valores de octeto, representativas de las cargas útiles normales recibidas a través de la red;

estando dicho ordenador configurado para:

determinar una longitud para los datos contenidos en dicha al menos una carga útil (S252, S352),

generar una distribución estadística de valores de octeto de los datos contenidos en dicha al menos una carga útil recibida dentro de la red (S256, S354),

seleccionar, entre la una o más distribución(es) estadística(s) modelo de valores de octeto, una distribución estadística modelo de valores de octeto basada, al menos en parte, en la longitud determinada, en donde la distribución estadística modelo de valores de octeto tiene una gama de longitudes predeterminada y se selecciona de modo que la longitud determinada para los datos contenidos en la al menos una carga útil esté incluida en la gama de longitudes predeterminada;

comparar al menos una parte de dicha distribución estadística generada de valores de octeto con una parte correspondiente de la distribución estadística modelo seleccionada de valores de octeto de dicha una o más distribución(es) estadística(s) modelo de valores de octeto (S356), e

identificar si dicha al menos una carga útil es una carga útil anómala (S358), en base, al menos en parte, a las diferencias detectadas entre la al menos una parte de la distribución estadística generada de valores de octeto para dicha al menos una carga útil y la parte correspondiente de dicha distribución estadística modelo seleccionada de valores de octeto.

30. El sistema de la reivindicación 29, en el cual dicho ordenador está adicionalmente configurado para:

medir una métrica de distancia entre la distribución estadística generada de valores de octeto de los datos contenidos en dicha al menos una carga útil y dicha distribución estadística modelo seleccionada de valores de octeto (S518); e

identificar cargas útiles anómalas como las cargas útiles que, al menos en parte, superan una métrica de distancia predeterminada (S520).

31. El sistema de la reivindicación 29, en el cual dicho ordenador está adicionalmente configurado para:

fijar un umbral de alerta correspondiente a un número deseado de alertas para cargas útiles anómalas detectadas; y

ajustar automáticamente dicha métrica de distancia predeterminada hasta que se alcance dicho umbral de alerta.

32. El sistema de la reivindicación 29, en el cual dicho ordenador comprende al menos un puerto y está adicionalmente configurado para:

determinar si una carga útil anómala detectada es o no un virus o gusano;

identificar al menos un puerto que ha sido sondeado por dicho virus o gusano; y

detectar la propagación de dicho gusano o virus comparando el tráfico de egreso con la distribución estadística generada de valores de octeto para dicho virus o gusano, a fin de detectar sondeos al al menos un puerto en al menos otro ordenador.

33. Un medio legible por ordenador que lleva instrucciones ejecutables por un ordenador, para modelar los datos de carga útil recibidos en una red, haciendo dichas instrucciones que dicho ordenador realice el procedimiento de:

recibir al menos una carga útil a través de la red (S250; S350);

determinar una longitud para los datos contenidos en la al menos una carga útil (S252; S352);

generar una distribución estadística de valores de octeto de los datos contenidos en la al menos una carga útil recibida dentro de la red;

seleccionar, entre una pluralidad de distribuciones estadísticas modelo de valores de octeto, una distribución estadística modelo de valores de octeto, representativa de las cargas útiles normales transmitidas a través de la red, en base, al menos en parte, a la longitud determinada, en donde la distribución estadística modelo de valores de octeto tiene una gama de longitudes predeterminada, y se selecciona de modo que la longitud determinada para los datos contenidos en la al menos una carga útil esté incluida dentro de la gama de longitudes predeterminada;

comparar al menos una parte de la distribución estadística generada de valores de octeto con una parte correspondiente de la distribución estadística modelo seleccionada de valores de octeto (S356) e

identificar si dicha al menos una carga útil es una carga útil anómala (S358), en base, al menos en parte, a las diferencias detectadas entre la al menos una parte de la distribución estadística generada de valores de octeto para la al menos una carga útil y la parte correspondiente de la distribución estadística modelo seleccionada de valores de octeto.

34. El medio legible por ordenador de la reivindicación 33, que comprende adicionalmente instrucciones para hacer que dicho ordenador realice el procedimiento de:

medir una métrica de distancia entre la distribución estadística generada de valores de octeto de los datos contenidos en dicha al menos una carga útil y dicha distribución estadística modelo seleccionada de valores de octeto (S518); e

identificar cargas útiles anómalas como las cargas útiles que, al menos en parte, superan una métrica de distancia predeterminada (S520).

35. El medio legible por ordenador de la reivindicación 33, que comprende adicionalmente instrucciones para hacer que dicho ordenador realice el procedimiento de:

fijar un umbral de alerta correspondiente a un número deseado de alertas para cargas útiles anómalas detectadas; y

ajustar automáticamente dicha métrica de distancia predeterminada hasta que se alcance dicho umbral de alerta.

36. El medio legible por ordenador de la reivindicación 33, que comprende adicionalmente instrucciones para hacer que dicho ordenador realice el procedimiento de:

determinar si una carga útil anómala detectada es o no un virus o gusano (S434);

identificar el al menos un puerto que ha sido sondeado por dicho virus o gusano; y

detectar la propagación de dicho gusano o virus comparando el tráfico de egreso con la distribución estadística generada de valores de octeto para dicho virus o gusano, a fin de detectar sondeos al al menos un puerto en al menos otro ordenador.

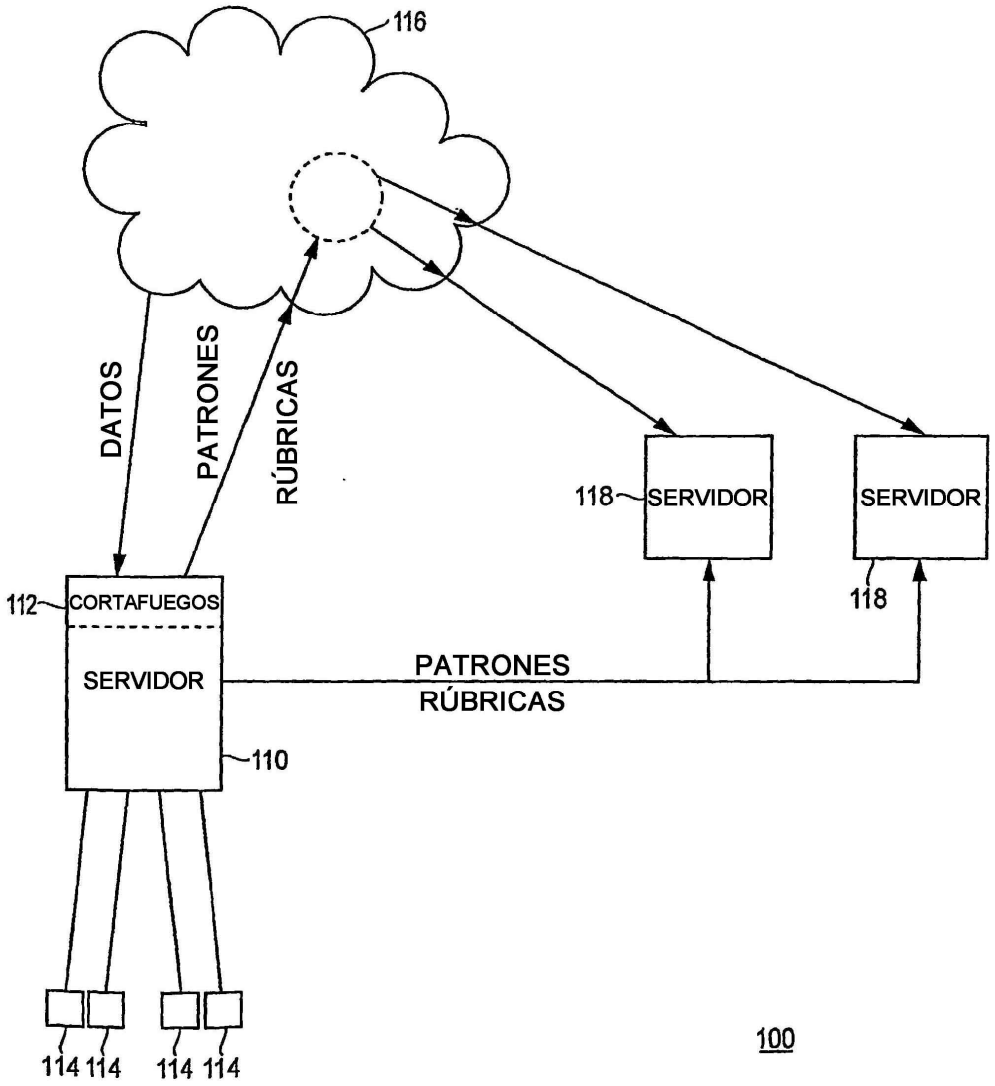
116
DATOS
PATRONES
RÚBRICAS
118
SERVIDOR
SERVIDOR
118
112
CORTAFUEGOS
SERVIDOR
PATRONES
RÚBRICAS
110
114 114 114 114
100

FIG. 1

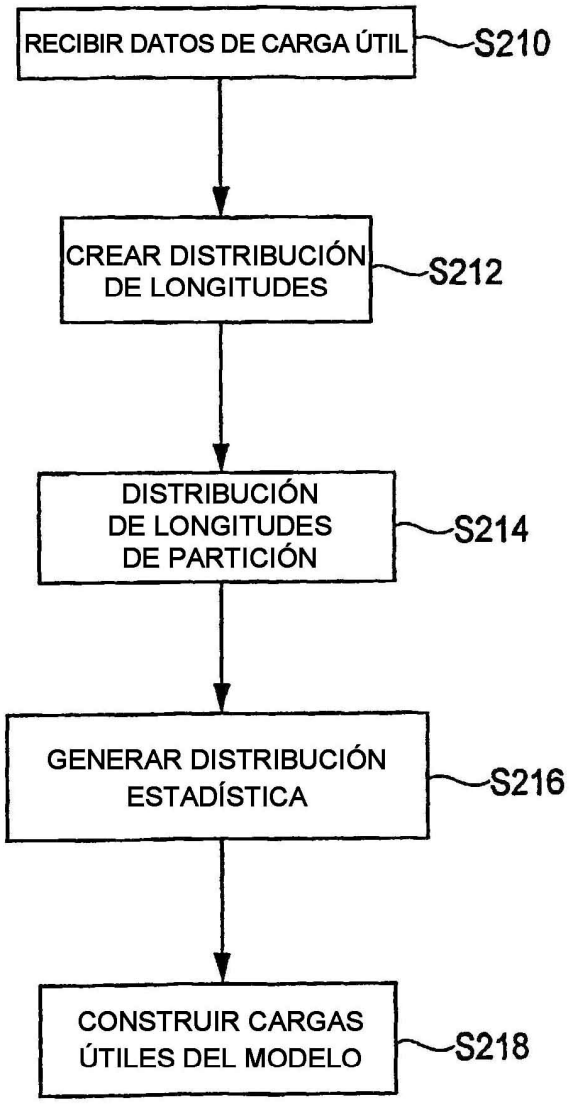
RECIBIR DATOS DE CARGA ÚTIL
S210
CREAR DISTRIBUCIÓN DE LONGITUDES
S212
DISTRIBUCIÓN DE LONGITUDES DE PARTICIÓN
S214
GENERAR DISTRIBUCIÓN ESTADÍSTICA
S216
CONSTRUIR CARGAS ÚTILES DEL MODELO
S218

FIG. 2

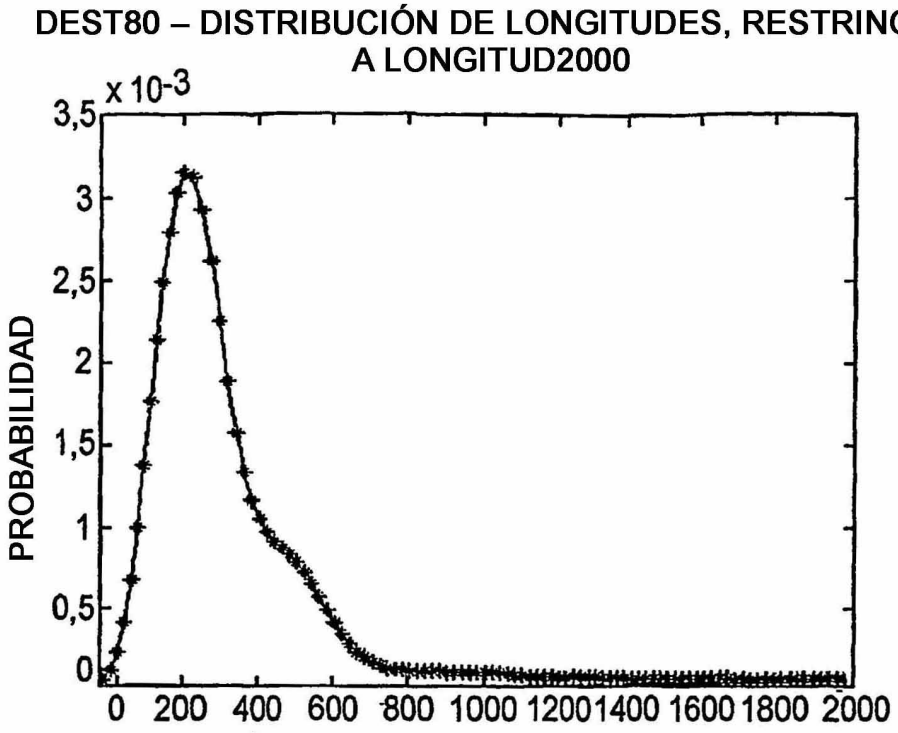
DEST80 – DISTRIBUCIÓN DE LONGITUDES, RESTRINGIR A LONGITUD2000
x 10-3
PROBABILIDAD
3,5
3
2,5
2
1,5
1
0,5
0
0 200 400 600 800 1000 1200 1400 1600 1800 2000

FIG. 3A

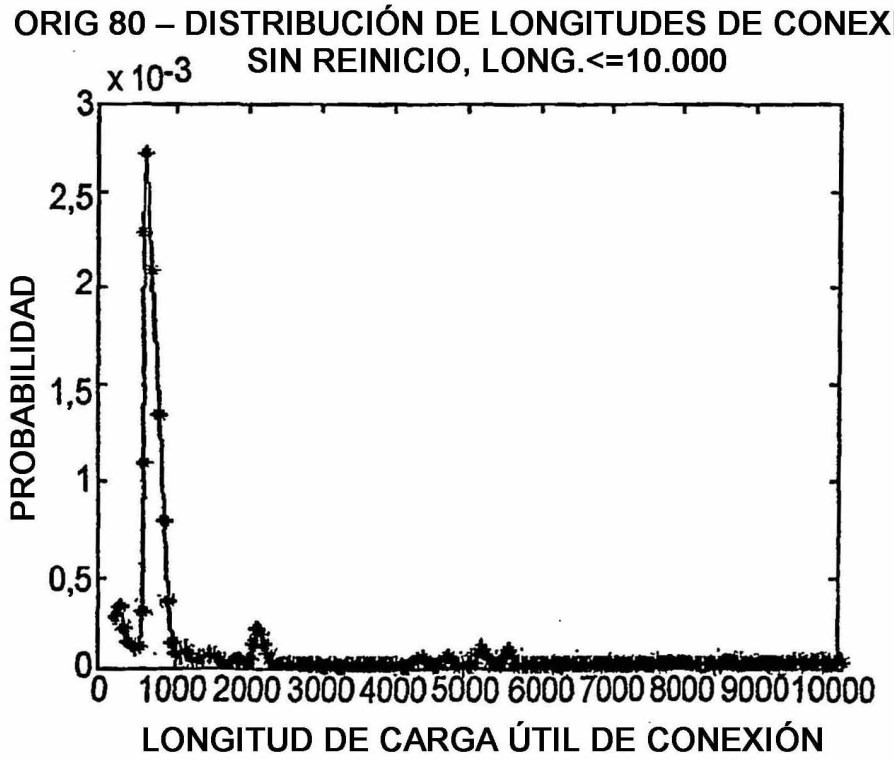
ORIG 80 – DISTRIBUCIÓN DE LONGITUDES DE CONEXIÓN, SIN REINICIO, LONG.<=10.000
x 10-3
PROBABILIDAD
3
2,5
2
1,5
1
0,5
0
0 1000 2000 3000 4000 5000 6000 7000 8000 9000 10000
LONGITUD DE CARGA ÚTIL DE CONEXIÓN

FIG. 3B

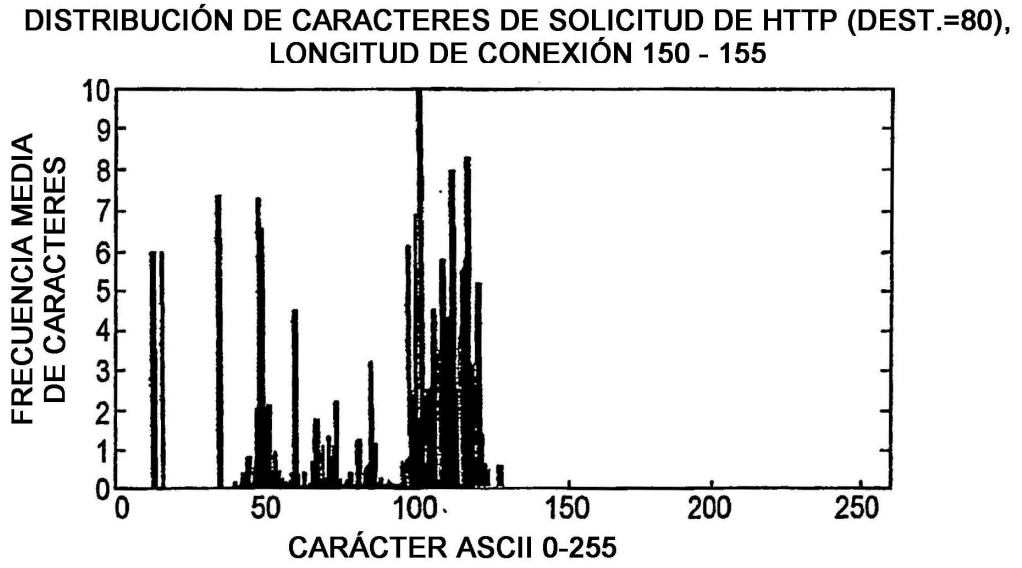
DISTRIBUCIÓN DE CARACTERES DE SOLICITUD DE HTTP (DEST.=80),
LONGITUD DE CONEXIÓN 150 - 155
FRECUENCIA MEDIA DE CARACTERES
0
1
2
3
4
5
6
7
8
9
10
0
50
100
150
200
250
CARÁCTER ASCII 0-255

FIG. 4

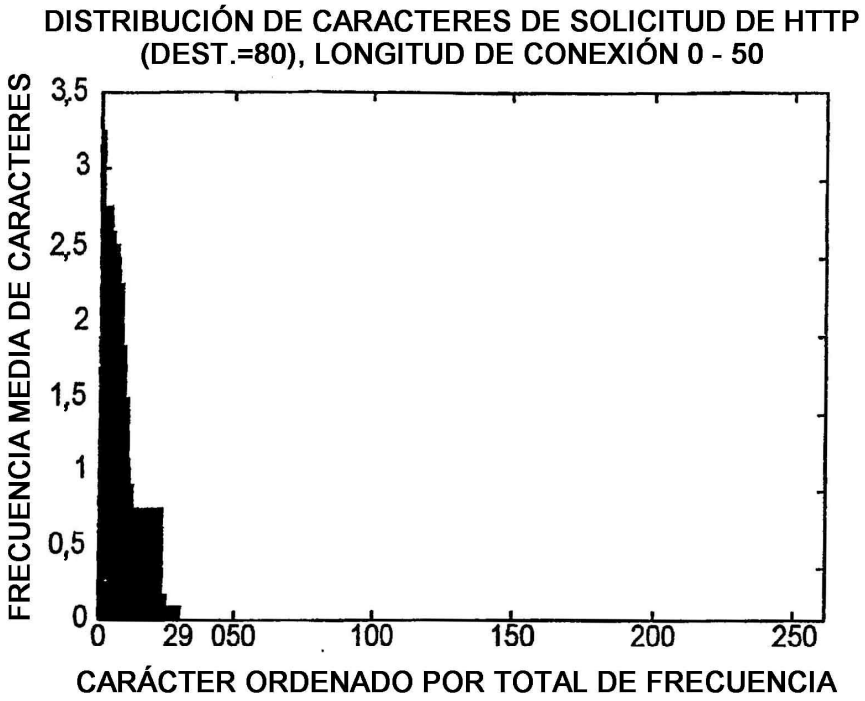
DISTRIBUCIÓN DE CARACTERES DE SOLICITUD DE HTTP
(DEST.=80), LONGITUD DE CONEXIÓN 0 - 50
FRECUENCIA MEDIA DE CARACTERES
3,5
3
2,5
2
1,5
1
0,5
0
0
29
050
100
150
200
250
CARÁCTER ORDENADO POR TOTAL DE FRECUENCIA

FIG. 5A

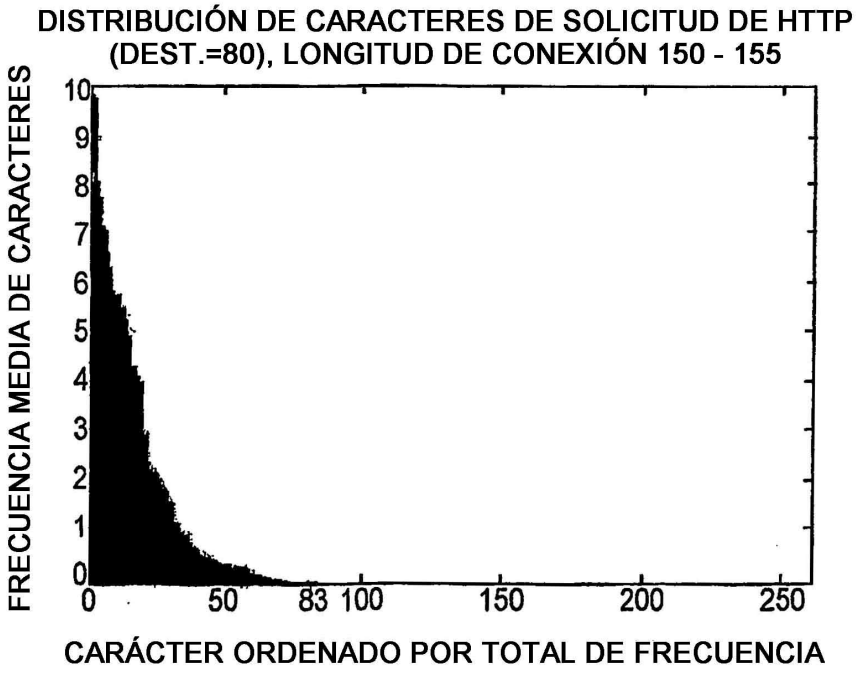
DISTRIBUCIÓN DE CARACTERES DE SOLICITUD DE HTTP
(DEST.=80), LONGITUD DE CONEXIÓN 150 - 155
FRECUENCIA MEDIA DE CARACTERES
10
9
8
7
6
5
4
3
2
1
0
0
50
83
100
150
200
250
CARÁCTER ORDENADO POR TOTAL DE FRECUENCIA

FIG. 5B

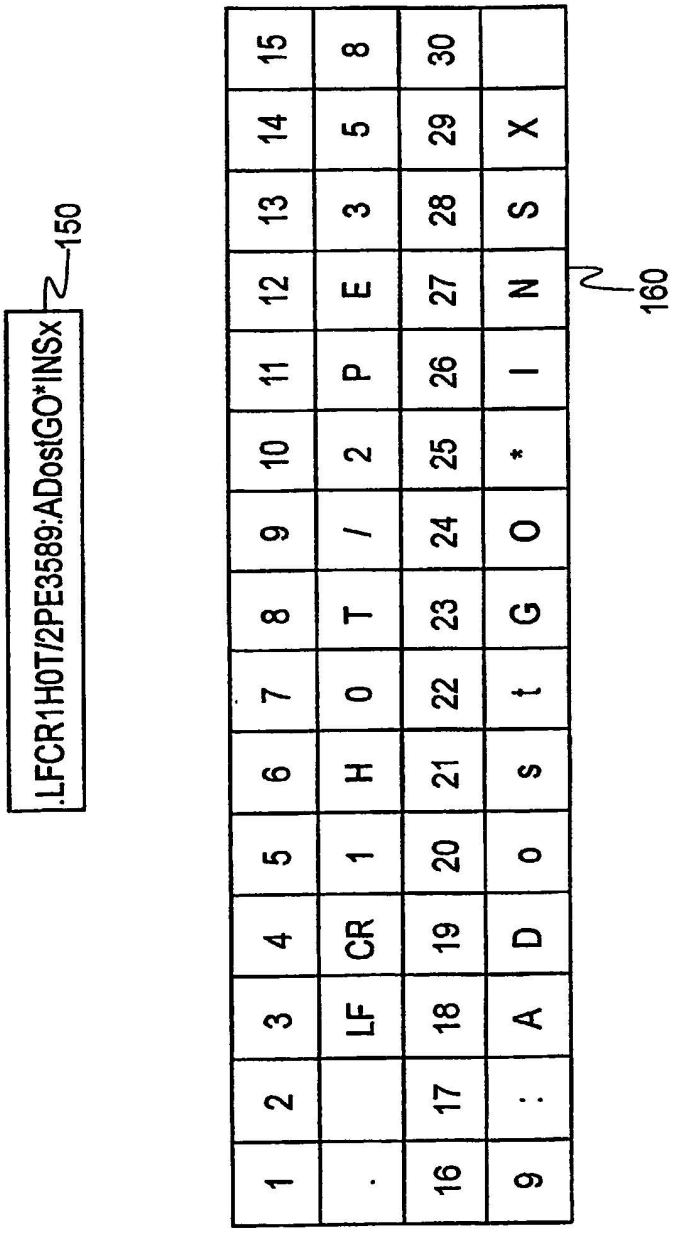
.LFCR1H0T/2PE3589:ADostGO*INSx
150
160
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
. LF CR 1 H 0 T / 2 P E 3 5 X
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
9 : A D o s t G O * I N S

FIG. 6A

exolac/ .siLFCRwmnp:rTh□~bgA0H1dxECkPfv;zGUM,()*~2S4Jy3W5FI@6=D_B7KLR9jZN[]8Vq%+?OXQ 170

180

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
e	t		o	l	a	c	/	.	s	i	LF	CR	w	m
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
n	p	:	r	T	h	DC1	-	b	g	A	0	H	I	d
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
x	E	C	k	P	f	v	;	z	G	U	M	,	(	)
46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
*	~	2	S	4	J	y	3	W	5	F	I	@	6	=
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
D	_	B	7	K	L	R	9	j	Z	N	[	]	8	V
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
q	%	+	?	O	X	Q	&							

FIG. 6B

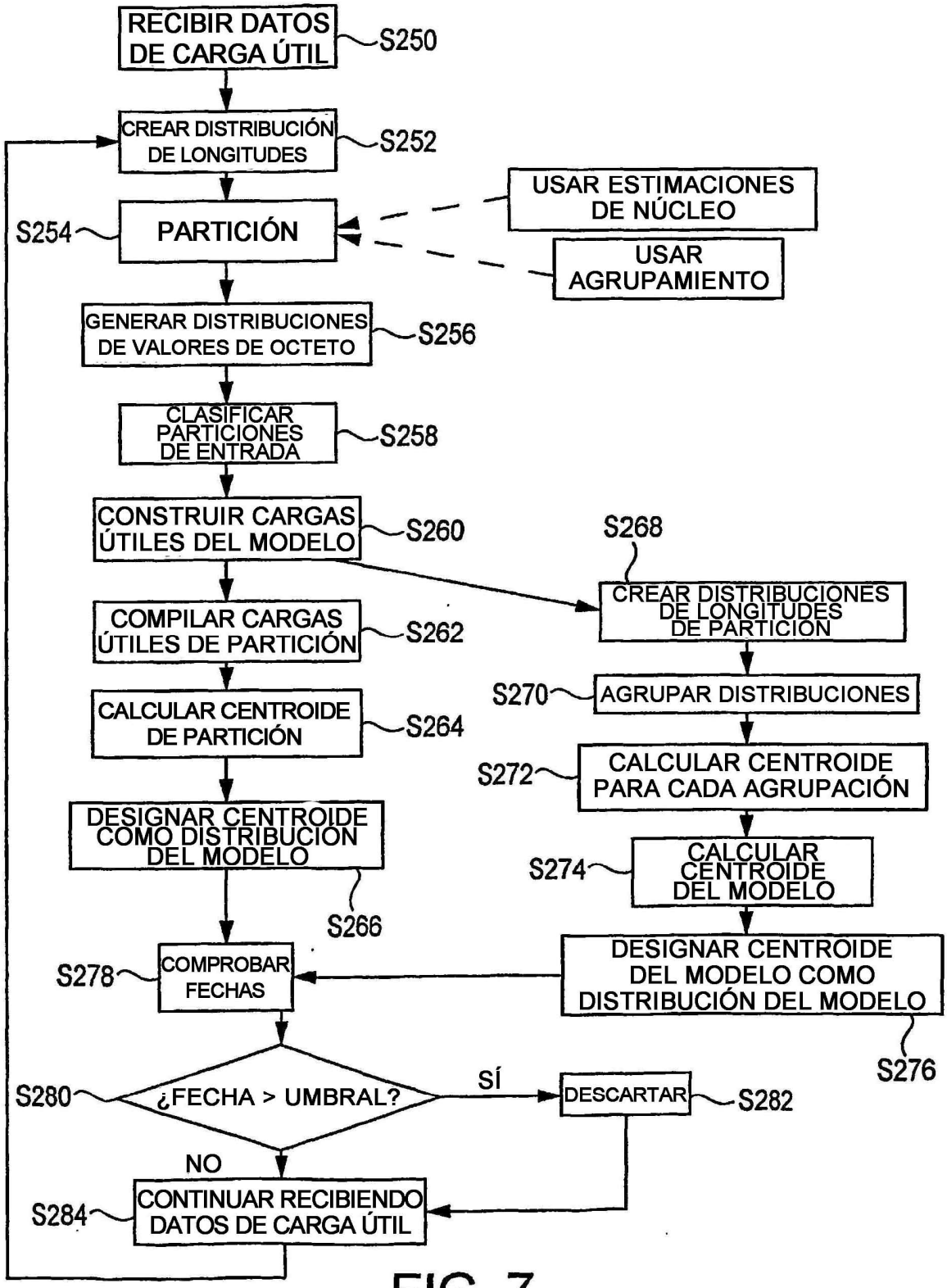
RECIBIR DATOS DE CARGA ÚTIL
S250
CREAR DISTRIBUCIÓN DE LONGITUDES
S252
S254
PARTICIÓN
USAR ESTIMACIONES DE NÚCLEO
USAR AGRUPAMIENTO
GENERAR DISTRIBUCIONES DE VALORES DE OCTETO
S256
CLASIFICAR PARTICIONES DE ENTRADA
S258
CONSTRUIR CARGAS ÚTILES DEL MODELO
S260
COMPILAR CARGAS ÚTILES DE PARTICIÓN
S262
CALCULAR CENTROIDE DE PARTICIÓN
S264
DESIGNAR CENTROIDE COMO DISTRIBUCIÓN DEL MODELO
S266
S268
CREAR DISTRIBUCIONES DE LONGITUDES DE PARTICIÓN
S270
AGRUPAR DISTRIBUCIONES
S272
CALCULAR CENTROIDE PARA CADA AGRUPACIÓN
S274
CALCULAR CENTROIDE DEL MODELO
DESIGNAR CENTROIDE DEL MODELO COMO DISTRIBUCIÓN DEL MODELO
S276
S278
COMPROBAR FECHAS
S280
¿FECHA > UMBRAL?
SÍ
DESCARTAR
S282
NO
S284
CONTINUAR RECIBIENDO DATOS DE CARGA ÚTIL

FIG. 7

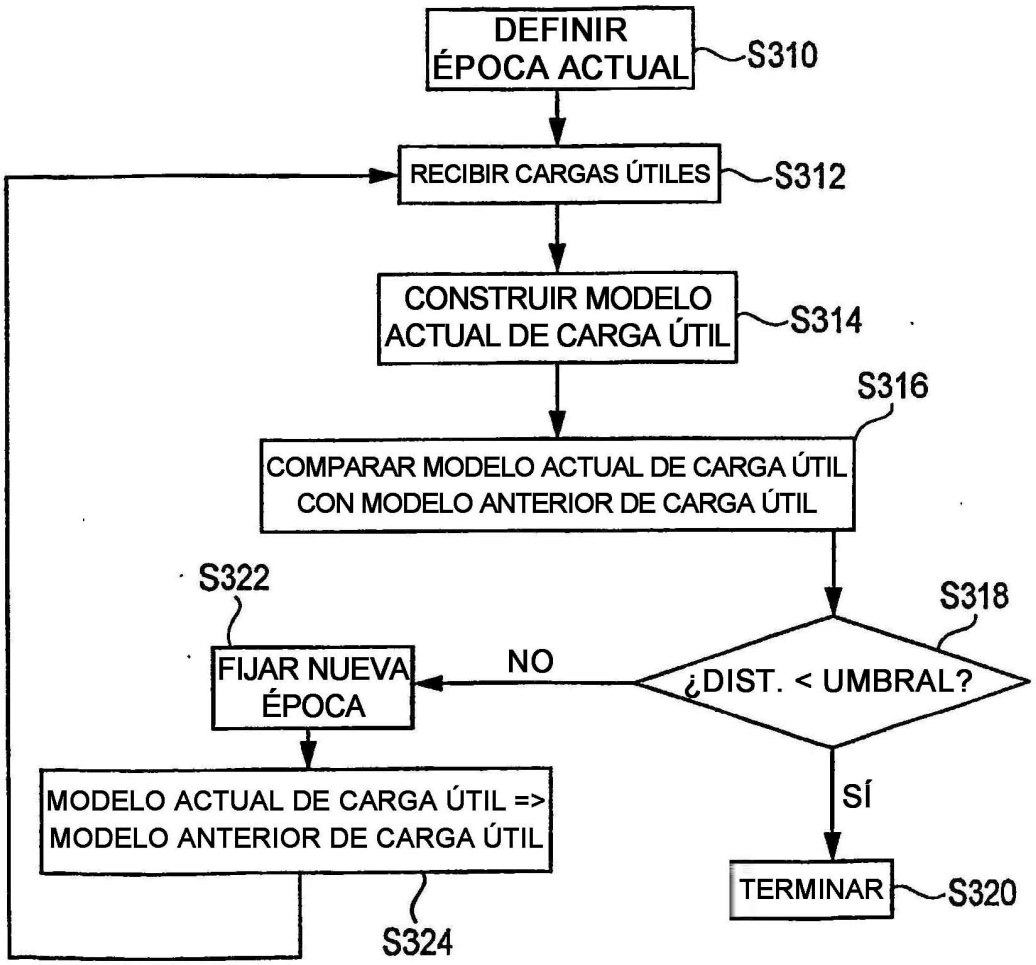
DEFINIR ÉPOCA ACTUAL
S310
RECIBIR CARGAS ÚTILES
S312
CONSTRUIR MODELO ACTUAL DE CARGA ÚTIL
S314
S316
COMPARAR MODELO ACTUAL DE CARGA ÚTIL CON MODELO ANTERIOR DE CARGA ÚTIL
S318
¿DIST. < UMBRAL?
NO
SÍ
S322
FIJAR NUEVA ÉPOCA
MODELO ACTUAL DE CARGA ÚTIL => MODELO ANTERIOR DE CARGA ÚTIL
S324
TERMINAR
S320

FIG. 8

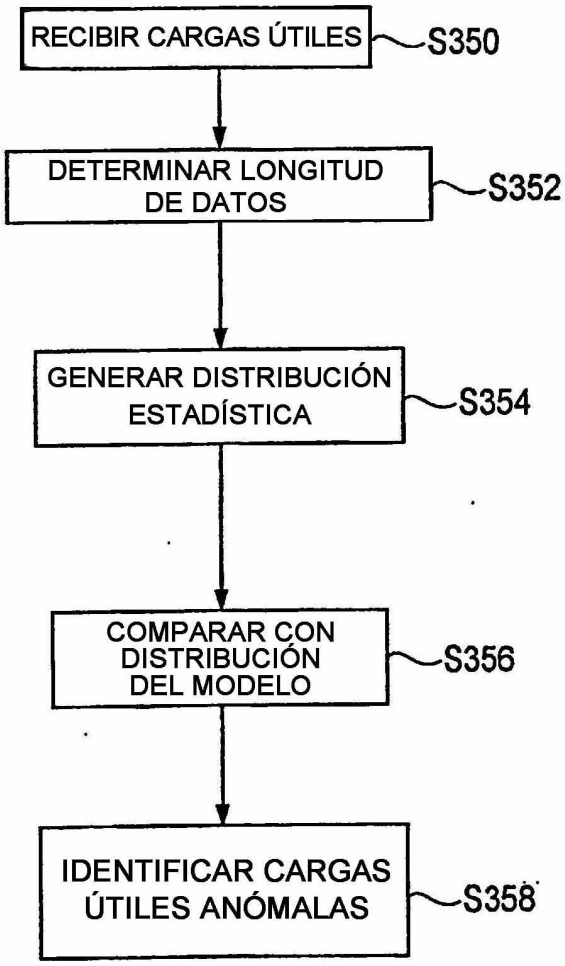
RECIBIR CARGAS ÚTILES
S350
DETERMINAR LONGITUD DE DATOS
S352
GENERAR DISTRIBUCIÓN ESTADÍSTICA
S354
COMPARAR CON DISTRIBUCIÓN DEL MODELO
S356
IDENTIFICAR CARGAS ÚTILES ANÓMALAS
S358

FIG. 9

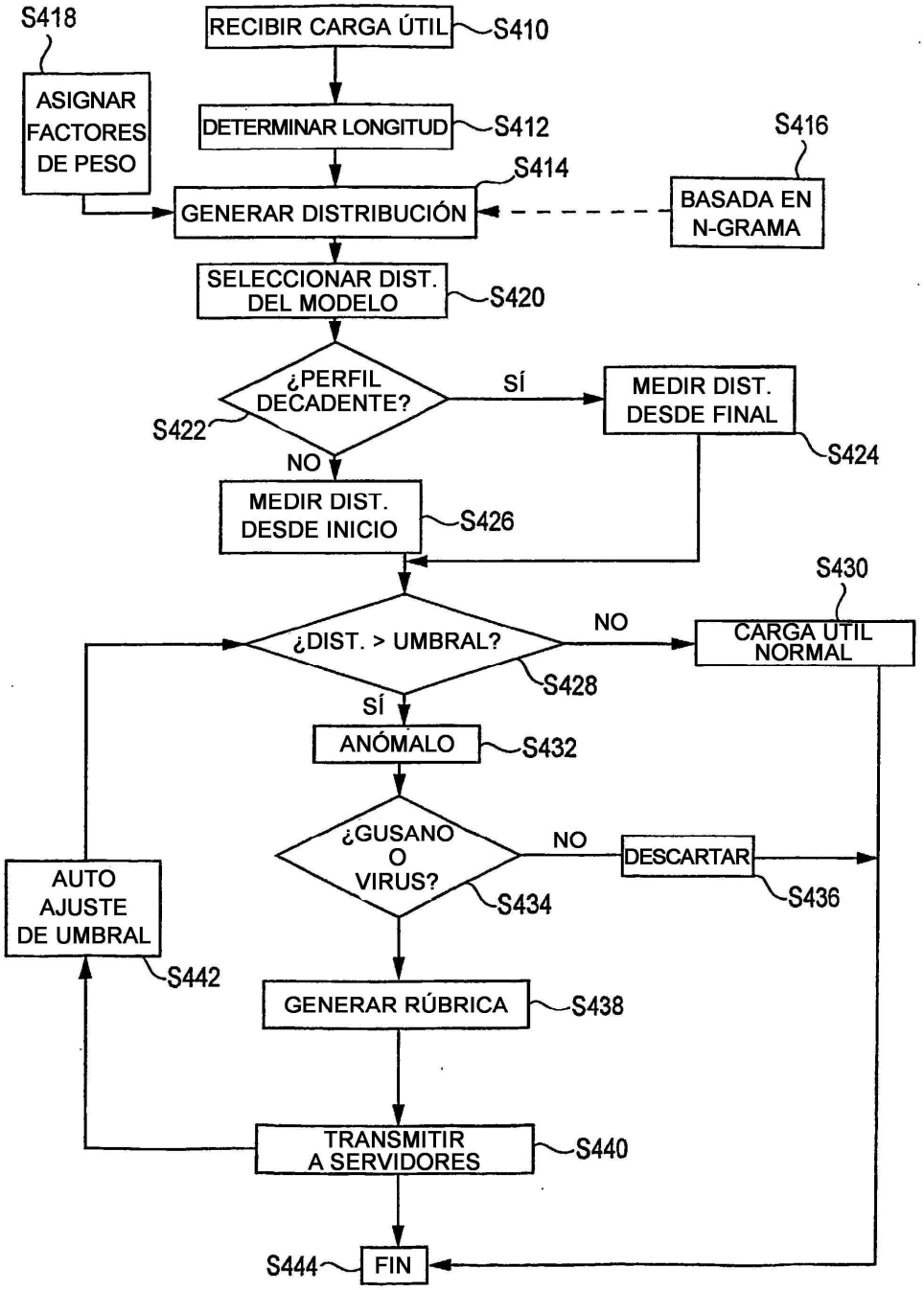
S418
RECIBIR CARGA ÚTIL
S410
ASIGNAR FACTORES DE PESO
DETERMINAR LONGITUD
S412
S416
S414
GENERAR DISTRIBUCIÓN
BASADA EN N-GRAMA
SELECCIONAR DIST. DEL MODELO
S420
¿PERFIL DECADENTE?
SÍ
MEDIR DIST. DESDE FINAL
S422
S424
NO
MEDIR DIST. DESDE INICIO
S426
S430
¿DIST. > UMBRAL?
NO
CARGA ÚTIL NORMAL
S428
SÍ
ANÓMALO
S432
¿GUSANO O VIRUS?
NO
DESCARTAR
AUTO AJUSTE DE UMBRAL
S434
S436
S442
GENERAR RÚBRICA
S438
TRANSMITIR A SERVIDORES
S440
S444
FIN

FIG 10

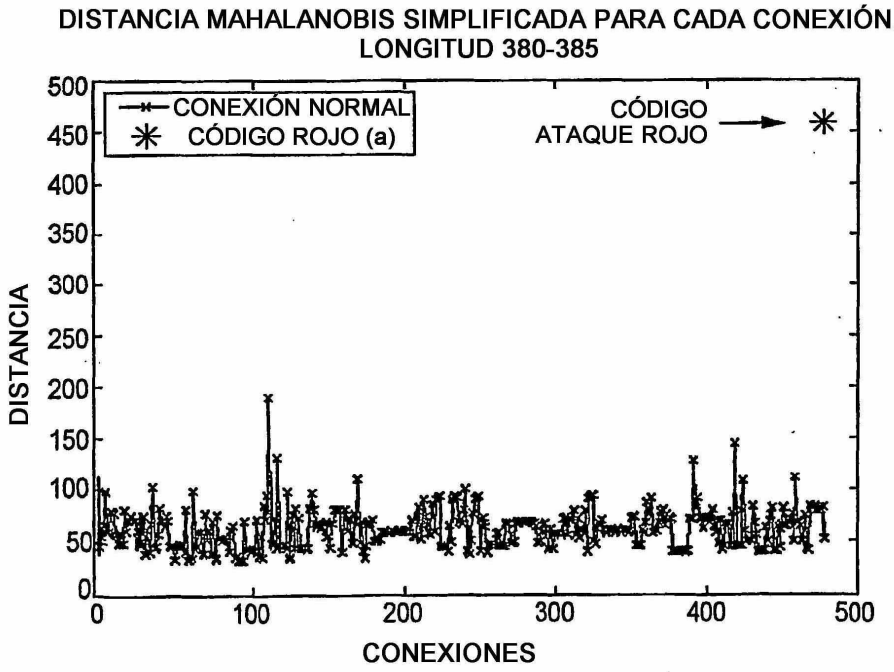
DISTANCIA MAHALANOBIS SIMPLIFICADA PARA CADA CONEXIÓN, LONGITUD 380-385
CONEXIÓN NORMAL
CÓDIGO ROJO (a)
CÓDIGO ATAQUE ROJO
DISTANCIA
CONEXIONES
0
50
100
150
200
250
300
350
400
450
500
0
100
200
300
400
500

FIG. 11

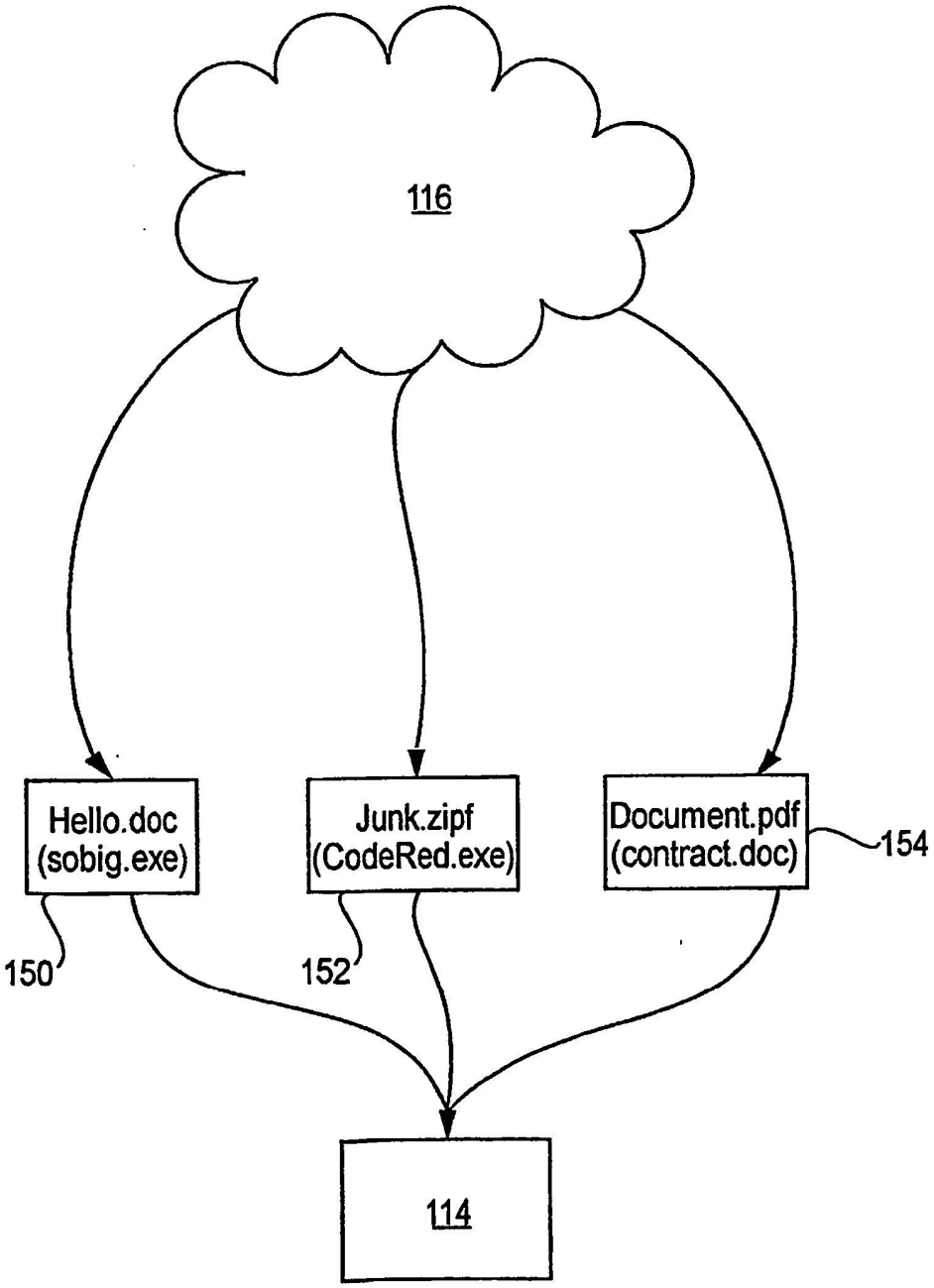
116
Hello.doc
(sobig.exe)
Junk.zipf
(CodeRed.exe)
Document.pdf
(contract.doc)
154
150
152
114

FIG. 12

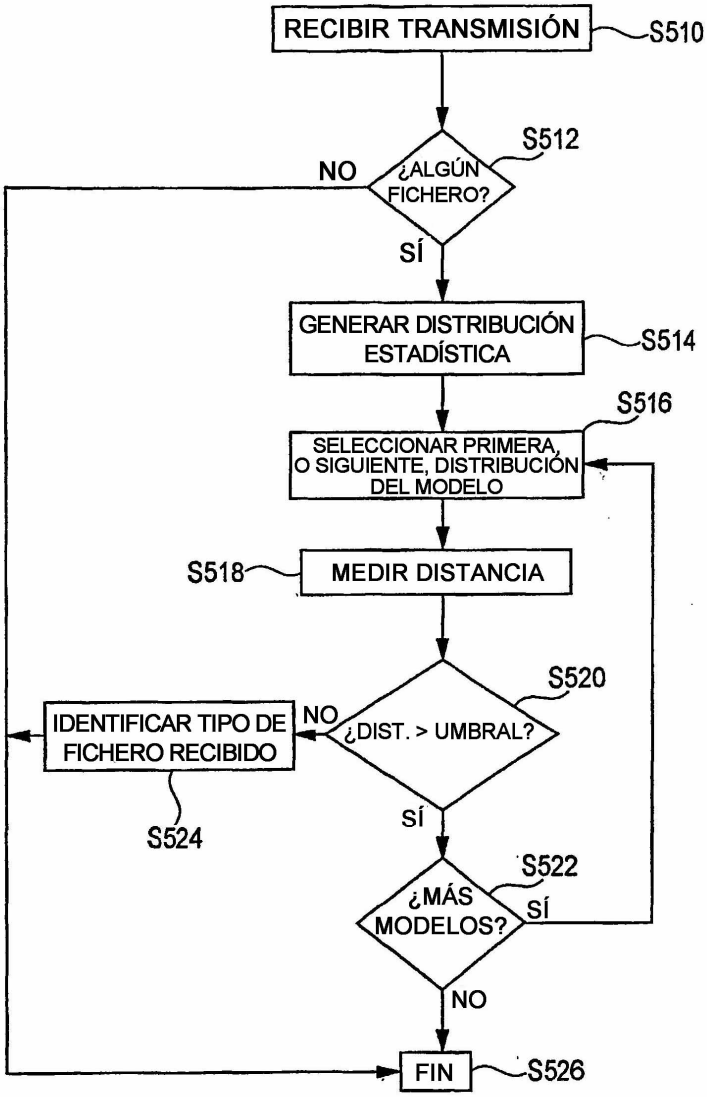
RECIBIR TRANSMISIÓN
S510
S512
¿ALGÚN FICHERO?
NO
SÍ
GENERAR DISTRIBUCIÓN ESTADÍSTICA
S514
S516
SELECCIONAR PRIMERA, O SIGUIENTE, DISTRIBUCIÓN DEL MODELO
S518
MEDIR DISTANCIA
S520
¿DIST. > UMBRAL?
NO
SÍ
IDENTIFICAR TIPO DE FICHERO RECIBIDO
S524
S522
¿MÁS MODELOS?
SÍ
NO
FIN
S526

FIG. 13

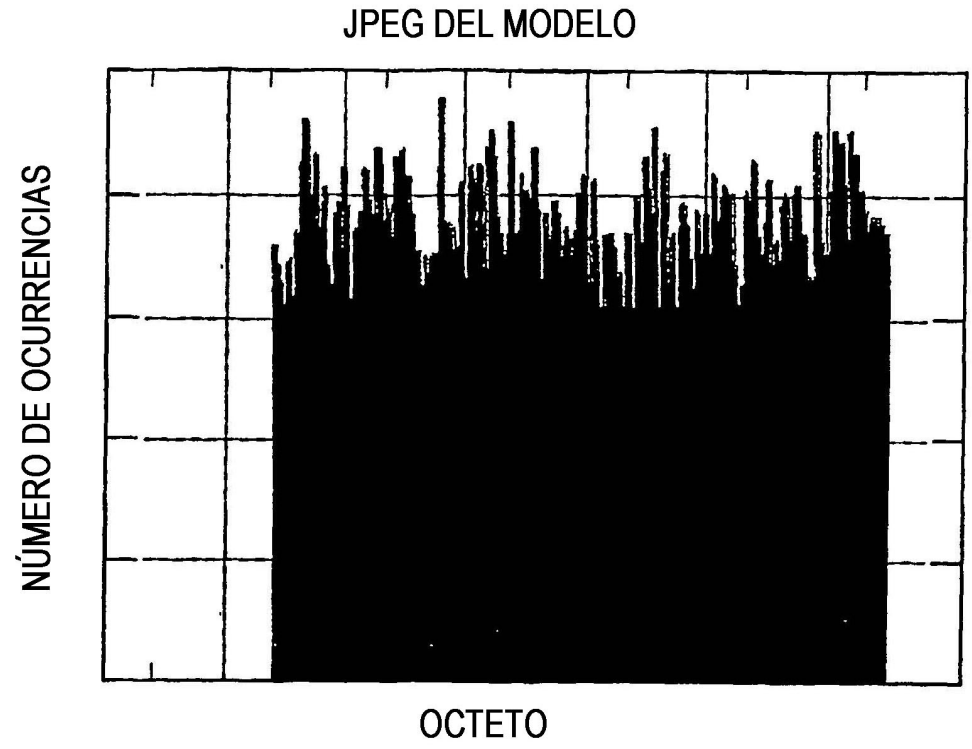

FIG. 14A

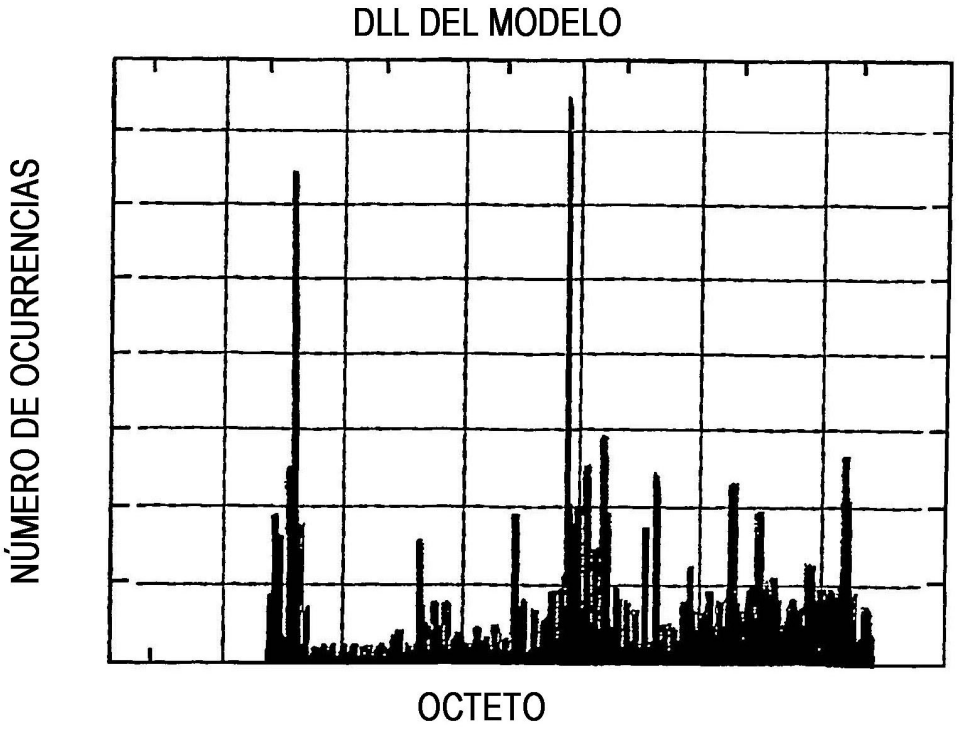

FIG. 14B

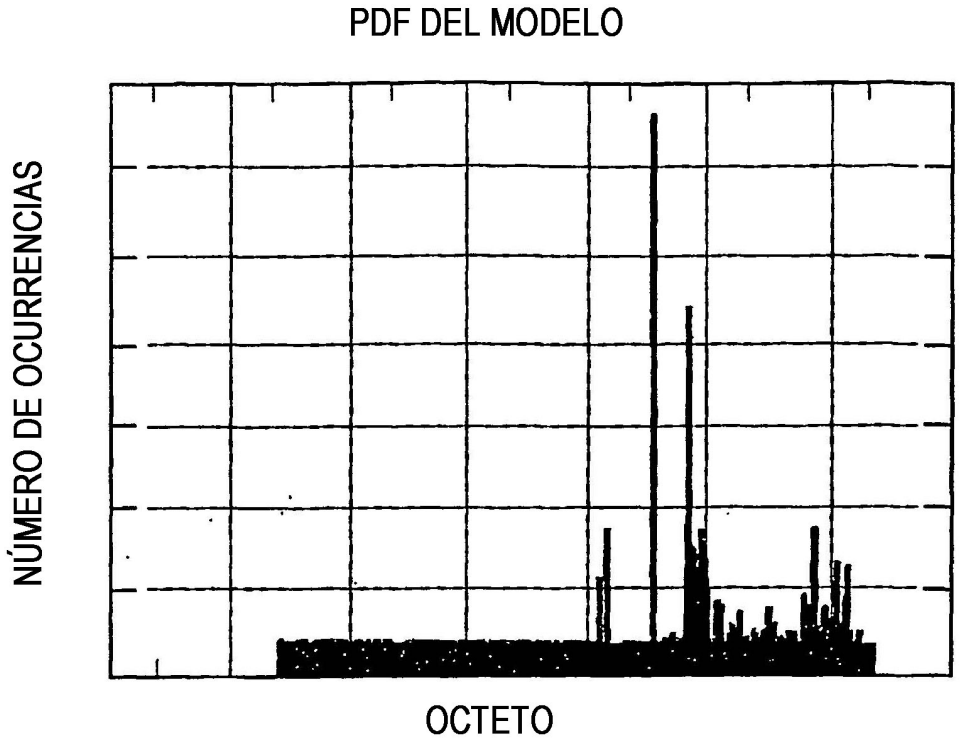

FIG. 14C

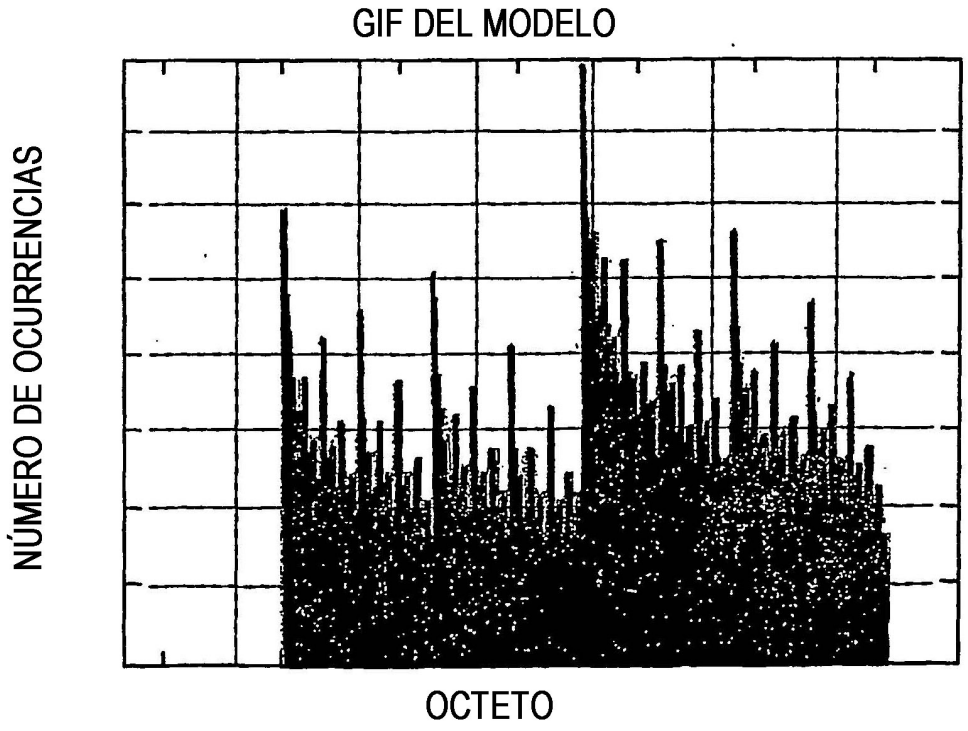

FIG. 14D

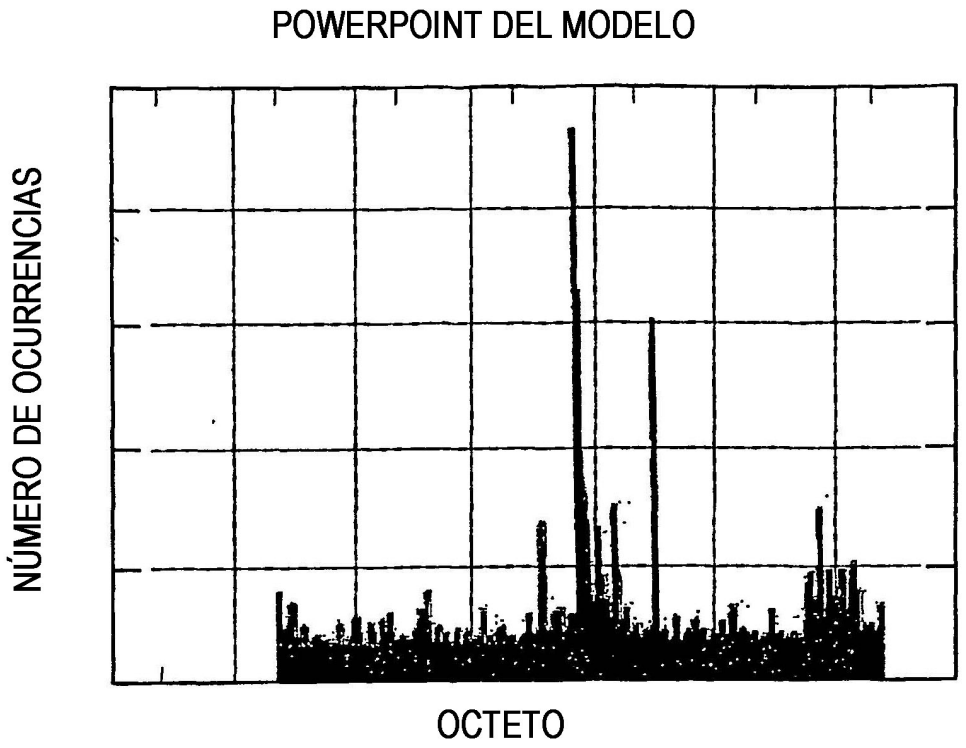

FIG. 14E

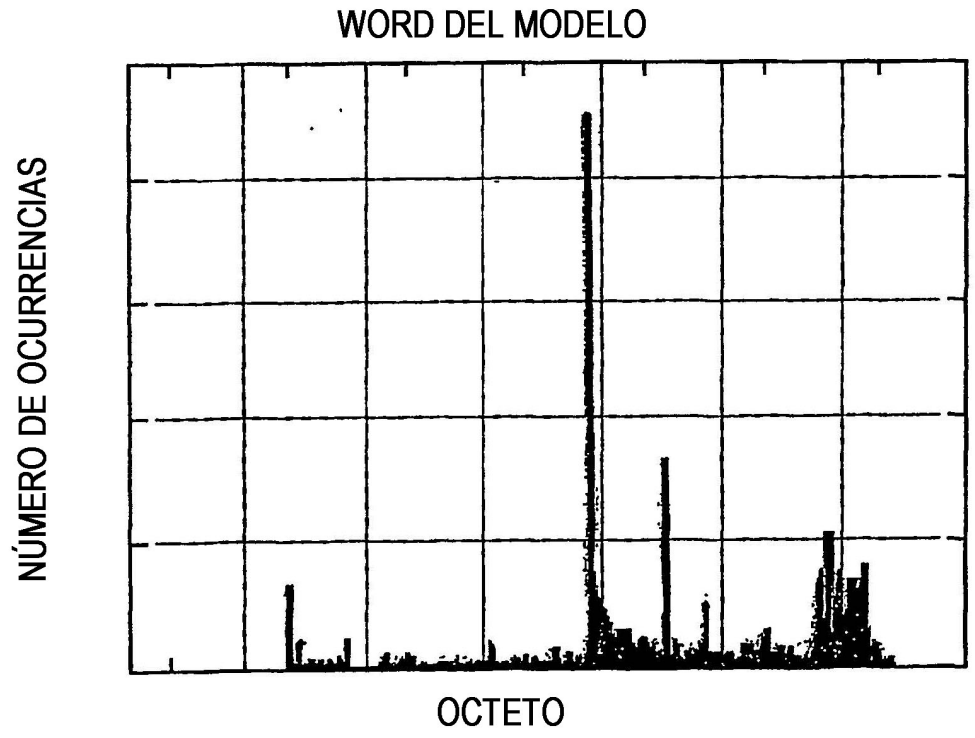

FIG. 14F

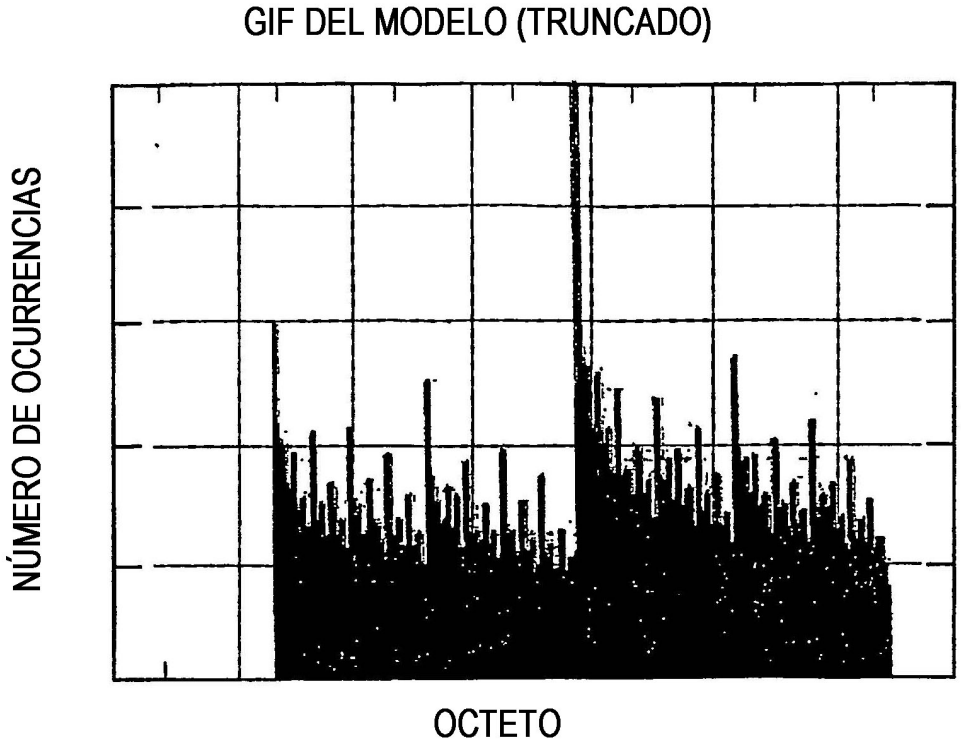

FIG. 14G

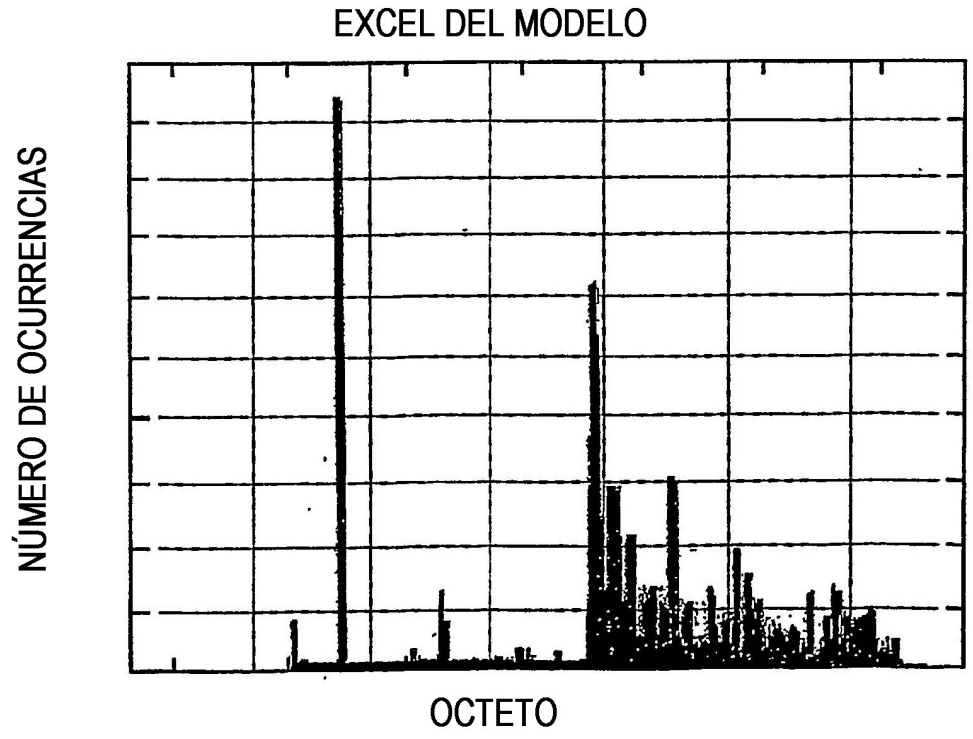

FIG. 14H

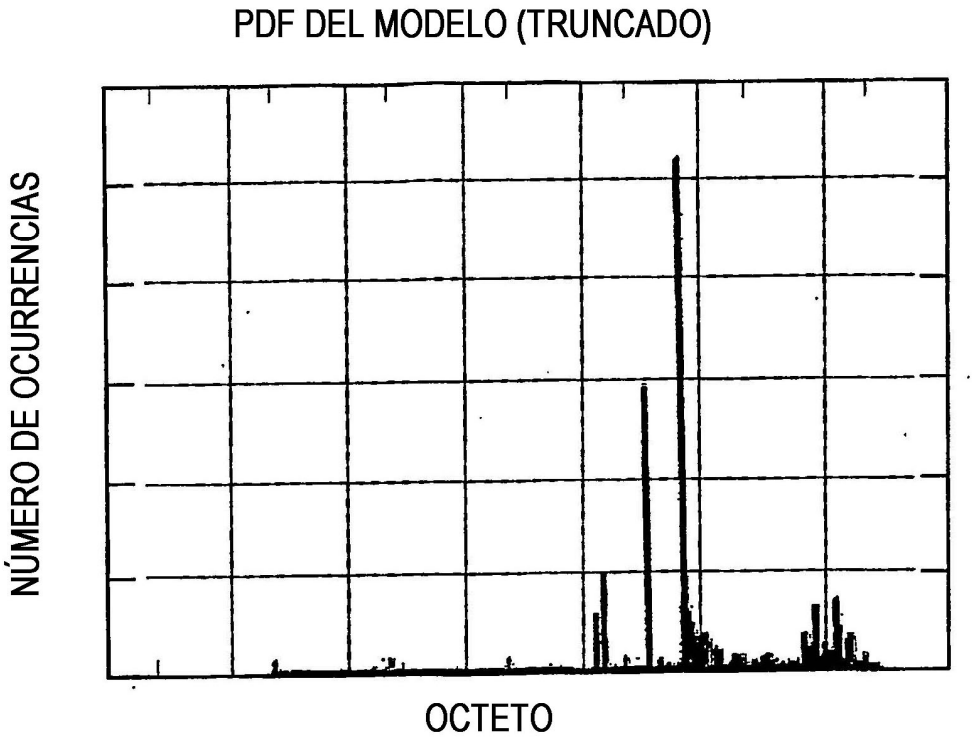

FIG. 14I

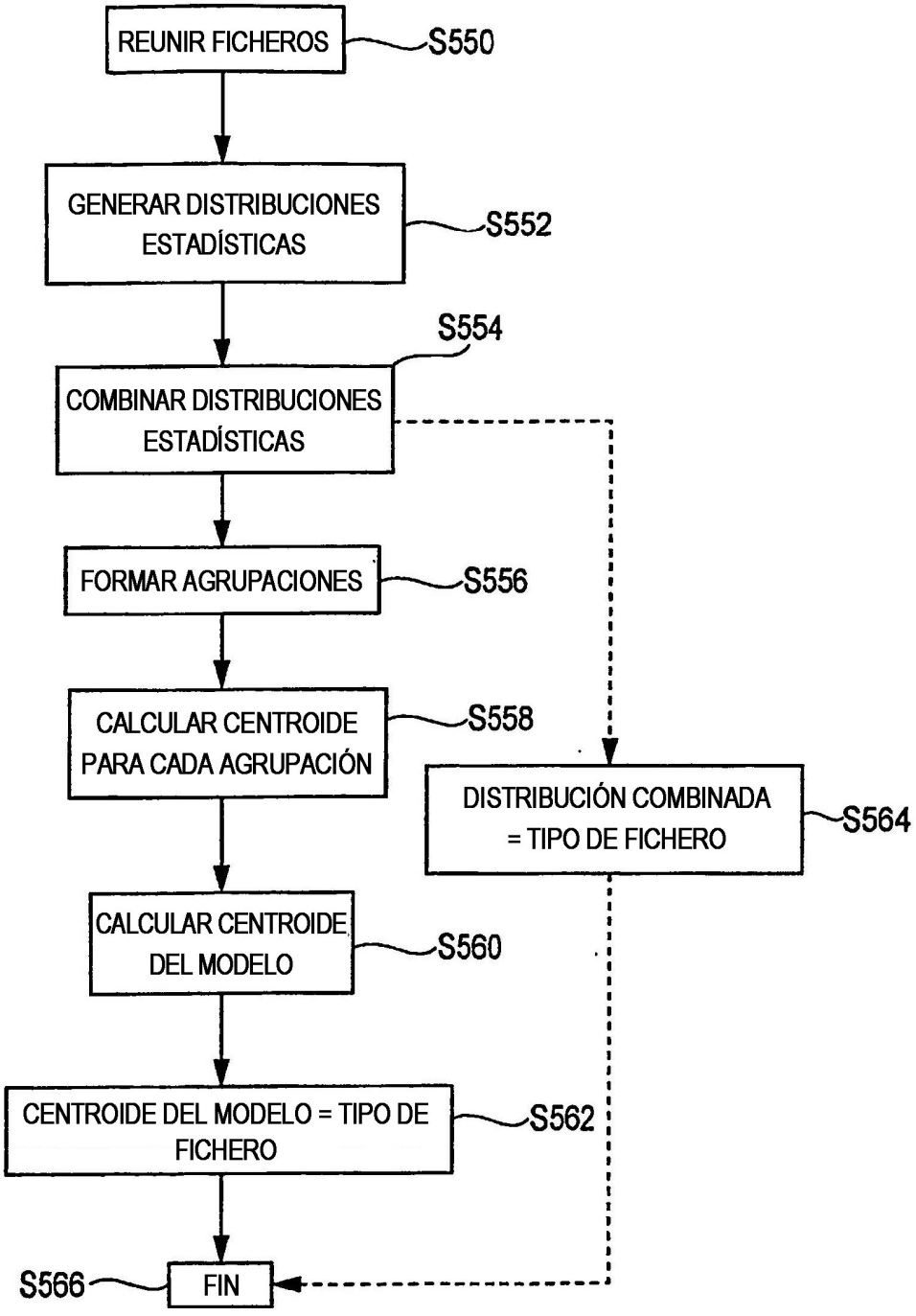
REUNIR FICHEROS
S550
GENERAR DISTRIBUCIONES ESTADÍSTICAS
S552
S554
COMBINAR DISTRIBUCIONES ESTADÍSTICAS
FORMAR AGRUPACIONES
S556
CALCULAR CENTROIDE PARA CADA AGRUPACIÓN
S558
DISTRIBUCIÓN COMBINADA = TIPO DE FICHERO
S564
CALCULAR CENTROIDE DEL MODELO
S560
CENTROIDE DEL MODELO = TIPO DE FICHERO
S562
S566
FIN

FIG. 15

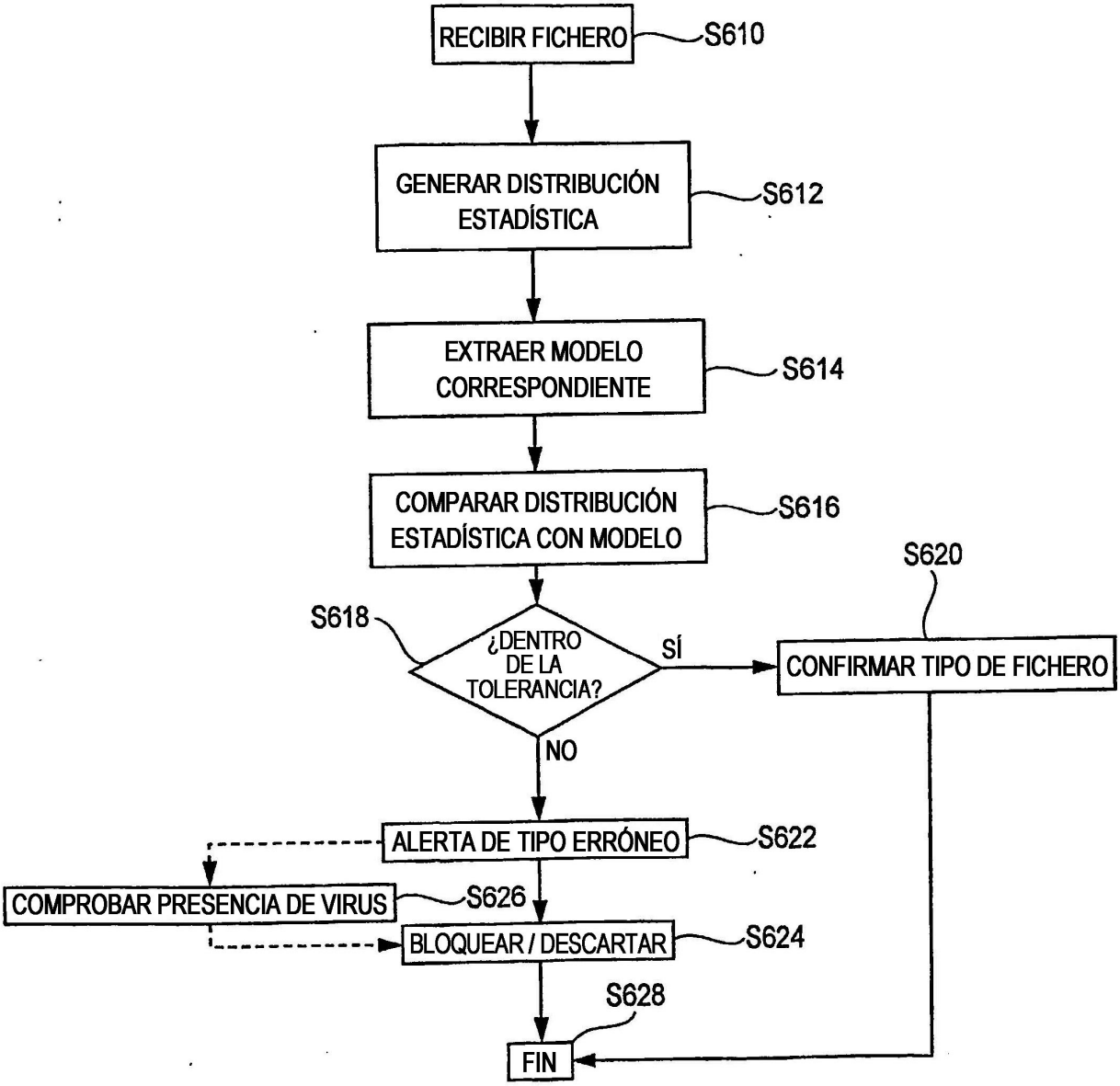
RECIBIR FICHERO
S610
GENERAR DISTRIBUCIÓN ESTADÍSTICA
S612
EXTRAER MODELO CORRESPONDIENTE
S614
COMPARAR DISTRIBUCIÓN ESTADÍSTICA CON MODELO
S616
S618
¿DENTRO DE LA TOLERANCIA?
SÍ
S620
CONFIRMAR TIPO DE FICHERO
NO
ALERTA DE TIPO ERRÓNEO
S622
COMPROBAR PRESENCIA DE VIRUS
S626
BLOQUEAR / DESCARTAR
S624
S628
FIN

FIG. 16

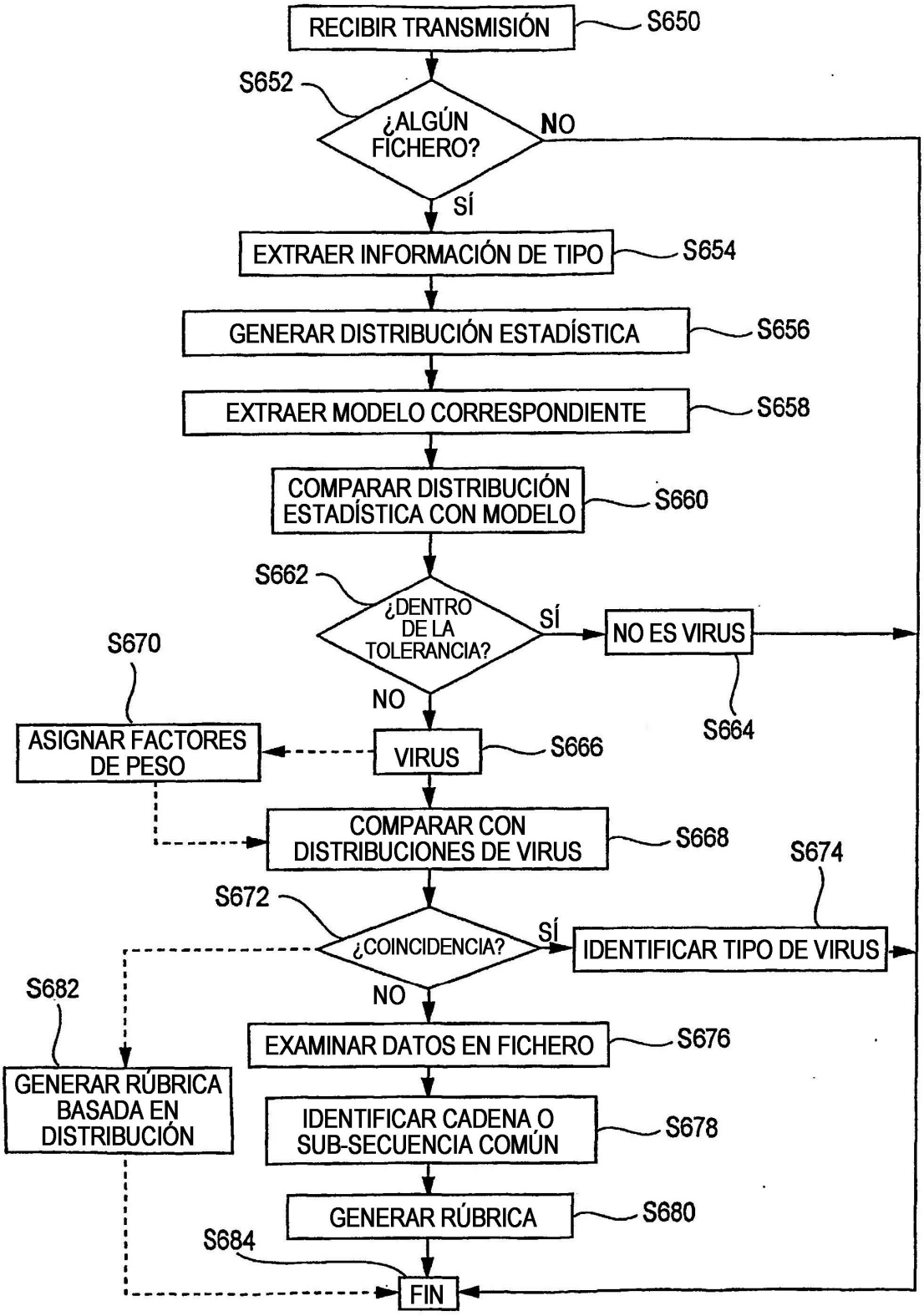
RECIBIR TRANSMISIÓN
S650
S652
¿ALGÚN FICHERO?
NO
SÍ
EXTRAER INFORMACIÓN DE TIPO
S654
GENERAR DISTRIBUCIÓN ESTADÍSTICA
S656
EXTRAER MODELO CORRESPONDIENTE
S658
COMPARAR DISTRIBUCIÓN ESTADÍSTICA CON MODELO
S660
S662
¿DENTRO DE LA TOLERANCIA?
SÍ
NO ES VIRUS
S664
NO
S670
ASIGNAR FACTORES DE PESO
VIRUS
S666
COMPARAR CON DISTRIBUCIONES DE VIRUS
S668
S674
S672
¿COINCIDENCIA?
SÍ
IDENTIFICAR TIPO DE VIRUS
NO
S682
EXAMINAR DATOS EN FICHERO
S676
GENERAR RÚBRICA BASADA EN DISTRIBUCIÓN
IDENTIFICAR CADENA O SUB-SECUENCIA COMÚN
S678
GENERAR RÚBRICA
S680
S684
FIN

FIG. 17

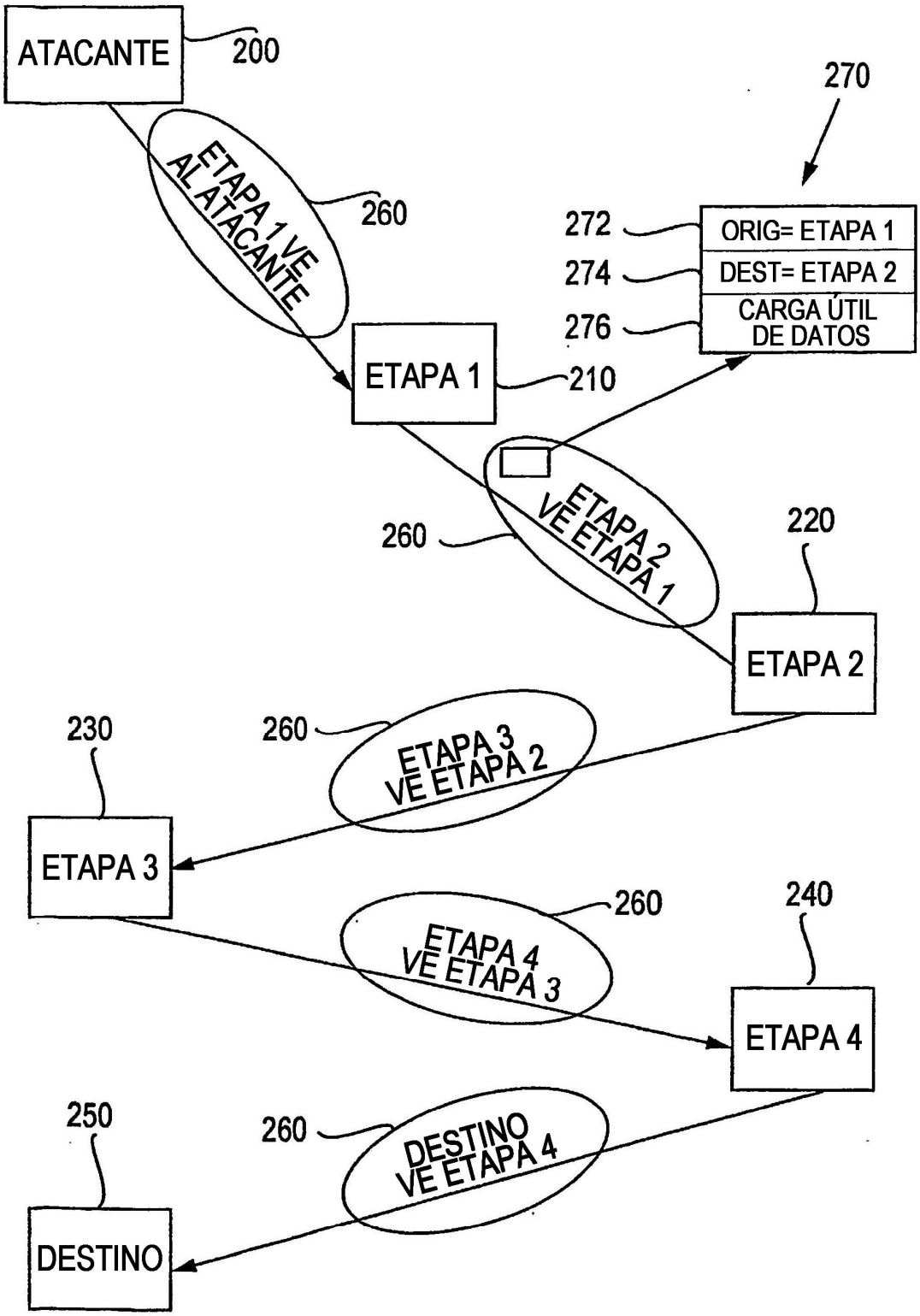
ATACANTE
200
270
ETAPA 1 VE AL ATACANTE
260
272
ORIG= ETAPA 1
274
DEST= ETAPA 2
276
CARGA ÚTIL DE DATOS
ETAPA 1
210
ETAPA 2 VE ETAPA 1
260
220
ETAPA 2
230
260
ETAPA 3 VE ETAPA 2
ETAPA 3
260
240
ETAPA 4 VE ETAPA 3
ETAPA 4
250
260
DESTINO VE ETAPA 4
DESTINO

FIG 18

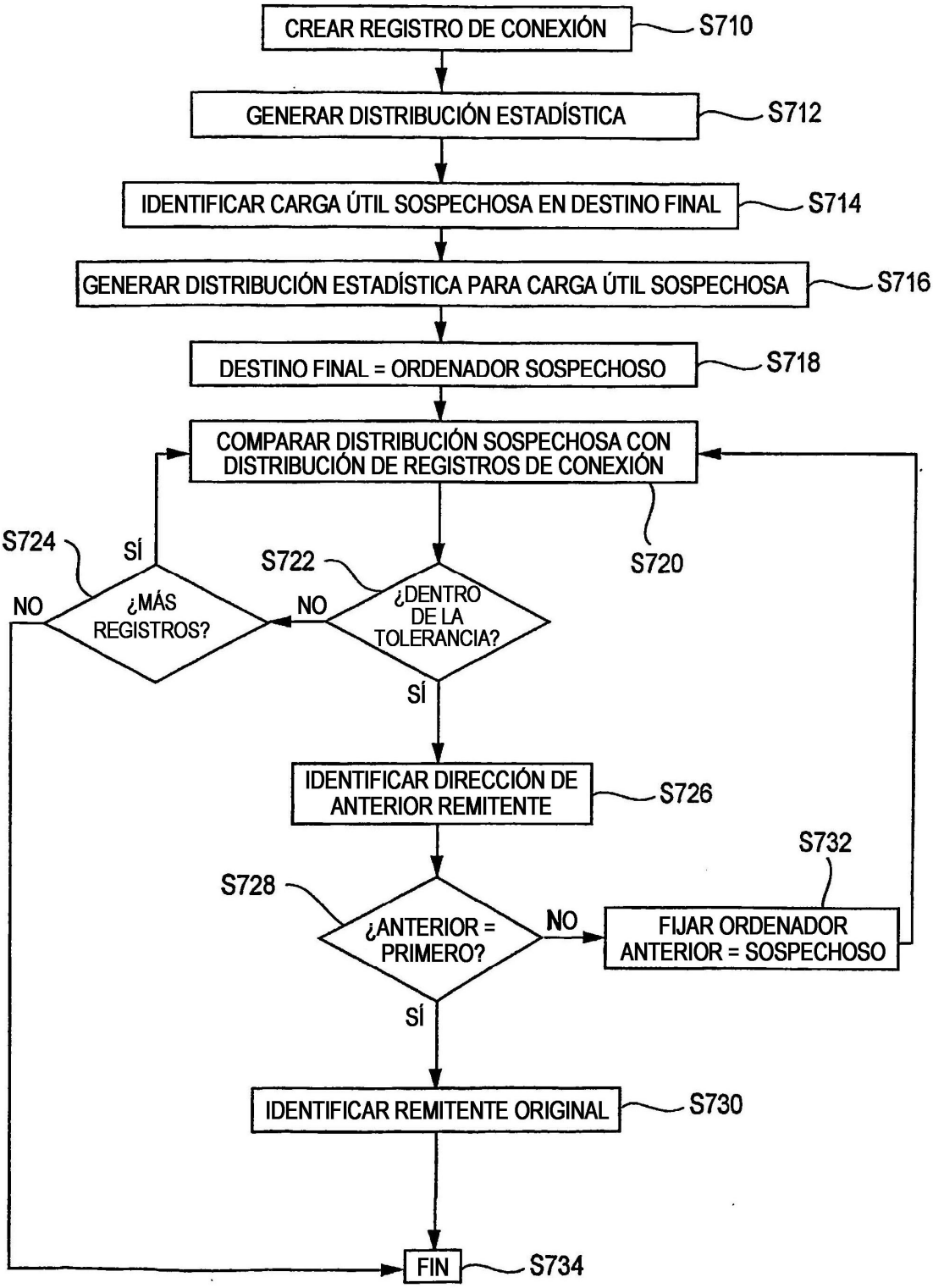
CREAR REGISTRO DE CONEXIÓN
S710
GENERAR DISTRIBUCIÓN ESTADÍSTICA
S712
IDENTIFICAR CARGA ÚTIL SOSPECHOSA EN DESTINO FINAL
S714
GENERAR DISTRIBUCIÓN ESTADÍSTICA PARA CARGA ÚTIL SOSPECHOSA
S716
DESTINO FINAL = ORDENADOR SOSPECHOSO
S718
COMPARAR DISTRIBUCIÓN SOSPECHOSA CON DISTRIBUCIÓN DE REGISTROS DE CONEXIÓN
S720
S722
¿DENTRO DE LA TOLERANCIA?
NO
SÍ
S724
¿MÁS REGISTROS?
NO
SÍ
IDENTIFICAR DIRECCIÓN DE ANTERIOR REMITENTE
S726
S728
¿ANTERIOR = PRIMERO?
NO
SÍ
S732
FIJAR ORDENADOR ANTERIOR = SOSPECHOSO
IDENTIFICAR REMITENTE ORIGINAL
S730
FIN
S734

FIG 19