(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 672 734**

(51) Int. CI.:

H04L 29/06 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **23.07.2014 E 14178205 (2)**

(97) Fecha y número de publicación de la concesión europea: **07.03.2018 EP 2830281**

(54) Título: **Procedimiento de asociación de fuentes de datos heterogéneos para la seguridad de redes de comunicación y sistema asociado**

(30) Prioridad:

25.07.2013 FR 1301787

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
15.06.2018

(73) Titular/es:

THALES (100.0%)
45, rue de Villiers
92200 Neuilly Sur Seine, FR

(72) Inventor/es:

HUYOT, BENOÎT;
MABIALA, YVES;
SANS, STÉPHANE y
CHOLLON, LAURENT

(74) Agente/Representante:

SALVA FERRER, Joan

DESCRIPCIÓN

Procedimiento de asociación de fuentes de datos heterogéneos para la seguridad de redes de comunicación y sistema asociado

[0001] La invención tiene como campo el de la seguridad de las redes de comunicación, y más particularmente, las redes de comunicación IP, tales como una red de área local.

[0002] En este campo, se conoce, por una parte, los componentes del *software* de monitorización de red, tales como los sistemas de detección de intrusión, también denominados sistemas IDS (según el acrónimo inglés "*Intrusion Detection System*"). Tal componente está parametrizado por el editor que incorpora en él una serie de reglas de alerta básica, así como también por el operador que define una biblioteca de reglas de alerta específicas. Cuando el componente detecta que un evento capturado en la red monitorizada cumple con todos los criterios de una regla de alerta, registra el evento sospechoso en una tabla de alertas, por ejemplo, almacenada en una base de datos.

[0003] En este campo, por otra parte, se conocen sondas de análisis del contenido de paquetes interceptados en la red, tales como sistemas de inspección profunda de paquetes, también denominados sistemas DPI (según el acrónimo en inglés *Deep Paquet Inspection*"). Dicha sonda es capaz de generar una pluralidad de archivos binarios, cada archivo recopila los datos de los paquetes que pertenecen a una misma sesión de comunicación. Los datos extraídos no sólo están relacionados con la parte del encabezado de los paquetes, sino también con la parte de carga útil de los paquetes.

[0004] En el presente documento, por "paquete de datos", se entiende un datagrama elemental transmitido a través de la red de comunicación monitorizada. Un paquete consta de una parte del encabezado, que contiene los datos que permiten el envío del paquete de un emisor a un destinatario a través de la red y una parte de carga útil, que contiene los datos de aplicación, también denominados mensaje, que el emisor desea compartir con el destinatario.

[0005] En el presente documento, por "sesión", se entiende el conjunto de paquetes de datos intercambiados, según el mismo protocolo (FTP, HTTP, etc.), entre un mismo emisor (definido por una dirección IP emisora y un puerto emisor) y un mismo destinatario (definido por una dirección IP de destino y un puerto de destino).

[0006] Por lo tanto, la tabla de alertas consta de informaciones genéricas acerca de un evento identificado como sospechoso. Sin embargo, un operador desearía poder disponer de informaciones adicionales que le permita analizar con más detalle un evento sospechoso, en particular, conocer cómo se inserta en una sesión de comunicación, y qué datos de aplicación transmite.

[0007] No obstante, los componentes de monitorización no permiten un análisis en profundidad del contenido del paquete correspondiente a un evento sospechoso, ni la reconstrucción de la sesión a la que pertenece este paquete.

[0008] Por el contrario, una sonda de análisis de contenido de paquetes intercepta el conjunto del flujo de datos que transitan en su punto de implantación en la red. Es adecuada para un tratamiento en tiempo real de paquetes según las reglas de recopilación mucho más simples que los componentes de monitorización. Los datos extraídos de un paquete son detallados y vuelven a las capas de la aplicación. Se almacenan directamente en un archivo que, en consecuencia, contiene una gran cantidad de entradas. Sin embargo, un operador desearía poder disponer de informaciones adicionales y más genéricas que le permita identificar si un paquete interceptado constituye un ataque. Sin embargo, la adición de medios de monitorización a una sonda conduciría a una disminución considerable en sus capacidades de procesamiento.

[0009] Por lo tanto, existe la necesidad de poder enriquecer las alertas generadas por un componente de monitorización, mediante datos recopilados por una sonda de análisis profundo de paquetes.

[0010] Sin embargo, los formatos de las fuentes de datos generados de salida, por una parte, de un componente de monitorización y, por otra parte, de una sonda de análisis de paquetes son heterogéneos.

[0011] Además, a partir del documento US 7 356 585 81, se conoce un procedimiento que asocia con cada puerto de la red, un sensor de monitoreo capaz de procesar los paquetes que circulan en el puerto asociado con el

objetivo de generar, para cada paquete, mensajes de datos de archivos en bruto, de alerta y/o de flujo de paquetes. Los mensajes procedentes de diferentes sensores son recopilados por un servidor, que consta de un motor que permite correlacionar los mensajes del mismo tipo entre sí. En particular, el servidor es capaz de asociar dos mensajes de flujo de paquetes asociados con paquetes diferentes en un mensaje combinado. Para conseguirlo, el servidor se basa en los criterios apropiados, tales como reglas de correlación, dirección IP de la fuente, mensaje de alerta asociado con cada mensaje de flujo de paquetes, etc. Cabe destacar que el valor de una función de creación de códigos de comprobación de cada mensaje de flujo de paquetes se calcula y sirve como un simple identificador de este mensaje.

[0012] La invención, en consecuencia, tiene por objeto responder a esta necesidad proponiendo un procedimiento que permita el cotejo automático de los datos contenidos en estos dos tipos de fuentes de datos.

[0013] Para ello, la invención tiene por objeto un procedimiento, un soporte de grabación de informaciones y un sistema de acuerdo con las reivindicaciones. La invención está definida en las reivindicaciones independientes. Otras características y ventajas de la invención resultarán más evidentes a partir de la siguiente descripción detallada de una realización particular, dada a título indicativo y de ninguna manera limitativo y con referencia a los dibujos anexos en los que:

- la figura 1 es una representación esquemática de una instalación de monitorización de una red de comunicación;
- la figura 2 es una representación esquemática de una tabla de creación de códigos de comprobación; y,
- la figura 3 es una representación, en forma de bloques, de una realización particular de un procedimiento de asociación de fuentes de datos de la instalación de la figura 1.

[0014] Con referencia a la figura 1, una instalación de monitorización 1 de una red de comunicación IP 2 consta de un componente de monitorización 3, capaz de generar una primera fuente de datos, en forma de una tabla de alertas 4, almacenada en una base de datos adecuada.

[0015] La instalación 1 también consta de una sonda de análisis del contenido de paquetes 5, capaz de generar una segunda fuente de datos 6 constituida por una pluralidad de archivos binarios F_m.

[0016] La instalación 1 consta de un motor de asociación de las primera y segunda fuentes de datos 4 y 6. Este motor está referenciado en la figura 1 por el número 10.

[0017] El motor 10 consta de un primer módulo 12 de generación de una tabla de elección arbitraria 14 a partir de las informaciones contenidas en la tabla de alertas 4.

[0018] El motor 10 también consta de un segundo módulo 16 de análisis del contenido de un archivo binario F_m y de reconciliación con la tabla 4.

[0019] El motor 10 es capaz de generar una tabla de alertas enriquecida 20.

[0020] El componente 3 es un sistema de detección de intrusión capaz de identificar eventos anormales en la red 2 monitorizada. Se trata por ejemplo del software libre SNORT.

[0021] El componente 3 es capaz de capturar, preferentemente en tiempo real, un flujo de paquetes de datos en la red 2.

[0022] El componente 3 consta de una biblioteca de reglas de alerta, que han sido configuradas por un operador o predefinidas por un editor. Cada regla de alerta se define por un conjunto de criterios que constituyen una firma de un posible ataque a la red 2.

[0023] El componente 3 es capaz de filtrar los paquetes capturados con el fin de verificar si responden, sí o no, al conjunto de criterios de una de las reglas de la biblioteca.

[0024] Si un evento cumple con el conjunto de criterios de una regla de alerta, el componente 3 almacena, en la tabla 4, este evento y los parámetros que lo caracterizan.

[0025] El formato IDMEF, según el acrónimo inglés "*Intrusion Detection Message Exchange Format*", permite formalizar el contenido de las tablas, tal como la tabla 4, generado en la salida de un componente de software de

monitorización. Este formato se describe por ejemplo en el documento RFC4765, disponible en línea en la dirección "http://tools.ietf.org/rfc/rfc4765.txt.".

[0026] Por lo tanto, el formato de la tabla 4 es el siguiente:

- un campo de identificación de un evento sospechoso E_i;
- la fecha en la que el paquete correspondiente a este evento fue capturado en la red;
- la dirección IP y el puerto del emisor del paquete capturado, indicados en la parte del encabezado del paquete capturado;
- la dirección IP y el puerto de destino del paquete capturado, indicados en la parte del encabezado del paquete capturado;
- la referencia de la regla de alerta que verifica el evento, esta referencia que permite volver a la regla de alerta en la biblioteca definida por el operador; y,
- un número entero que indica la gravedad de la alerta.

[0027] La sonda 5 realiza una inspección en profundidad de los paquetes interceptados en la red 2.

[0028] La sonda 5 permite la duplicación de todos los paquetes que circulan en la red 2 en el punto de implantación de la sonda 5 en la red 2 y que corresponden a una regla de intercepción. La regla de intercepción es definida por el operador.

[0029] Para cada paquete interceptado, la sonda 5 no sólo lee el contenido de la parte del encabezado de este paquete, sino también el contenido de la parte de carga útil de este paquete, es decir, el mensaje transmitido entre el emisor y el destinatario del paquete.

[0030] La sonda 5 también es adecuada para reconstruir una sesión de comunicación entre un emisor y un destinatario asociando los paquetes que comparten los mismos dos pares de dirección IP y puerto de máquina, y el mismo protocolo de comunicación. Un nuevo paquete es añadido a una sesión cuando el tiempo transcurrido entre la fecha en la que el nuevo paquete fue interceptado en la red y la fecha en la que el último paquete de la sesión fue interceptado en la red es inferior a un tiempo predeterminado. De lo contrario, se crea una nueva sesión.

[0031] La sonda 5 almacena el conjunto de datos resultantes de esta lectura y de esta reconstrucción en la segunda fuente de datos 6. La sonda 5 es capaz de generar, por sesión, un archivo binario F_m que consta de datos relacionados con la parte del encabezado de los paquetes (dirección IP y puerto de una primera máquina, dirección IP y puerto de una segunda máquina, así como el protocolo de comunicación utilizado para estas primera y segunda máquinas) y con la parte de carga útil de los paquetes interceptados (parámetros de las consultas intercambiadas entre las primera y segunda máquinas, valores de los parámetros en las respuestas intercambiadas, etc.).

[0032] El formato de un archivo F_m es, por ejemplo, del tipo ".pcap", generado, por ejemplo, por el *software* Winpcap de la sociedad Riverbed Technology.

[0033] El motor 10 es adecuado para implementar el procedimiento de asociación representado esquemáticamente en la figura 3.

[0034] El procedimiento de asociación 100 se descompone en dos partes, consiste en primer lugar en elaborar una tabla de elección arbitraria 14 a partir de la tabla 4, y acto seguido, una vez que se ha obtenido la tabla de elección arbitraria, analizar sucesivamente cada uno de los archivos F_m disponibles con el fin de intentar unirlo con uno de los eventos en la tabla 4. El procedimiento conduce a la generación de una tabla enriquecida 20.

[0035] En general, la implementación del procedimiento 100 proporciona la definición de una clase Alerta. La clase Alerta está definida por una pluralidad de variables que son:

- la dirección IP del emisor (IPE), codificada en cuatro bytes;
- la dirección IP del destinatario (IPD), codificada en cuatro bytes;
- el puerto utilizado por el emisor (PE), codificado en dos bytes;
- el puerto utilizado por el destinatario (PD), codificado en dos bytes; y,
- el protocolo de comunicación utilizado (P), codificado en un byte.

[0036] Un objeto Alerta A_i corresponde a una instancia de la clase Alerta, es decir, un grupo de valores para

las cinco variables mencionadas anteriormente. Por lo tanto, un objeto Alerta A_i está codificado en trece bytes.

[0037] La clase Alerta también se define mediante una función de creación de códigos de comprobación H. La función de creación de códigos de comprobación utilizada en la presente realización tiene la forma:

$$H = \sum_{q=0}^{N-1} S_q .37^{N-1-q}$$

en la que: N es el número de bytes utilizado para representar un objeto (N equivale a trece en la presente realización); y S_q es un peso que permite conferir a algunos bytes una mayor importancia en el valor de elección arbitraria obtenido a fin de reducir las colisiones (es decir, con el fin de evitar como máximo dos objetos diferentes que tienen el mismo valor de elección arbitraria). Por ejemplo, los dos últimos bytes de cada dirección IP se ven más afectados que los primeros dos bytes de cada dirección IP.

[0038] Finalmente, la clase Alerta está definida por una función de igualdad. La operación de igualdad se redefine para permitir la comparación de los primer y segundo objetos entre sí para verificar si son iguales. El procedimiento de ejecución de la función de igualdad consiste en verificar, sucesivamente para cada una de las variables del primer objeto, si su valor es igual al de la variable correspondiente del segundo objeto. Tan pronto como una verificación es negativa, el procedimiento se detiene y los primer y segundo objetos se denominan diferentes.

[0039] Ventajosamente, para acelerar el procedimiento de ejecución de la función de igualdad, que se utiliza sólo si dos objetos tienen el mismo valor de elección arbitraria, conviene ordenar las variables para que el proceso de ejecución de la función de creación de códigos de comprobación comience por la verificación de las variables cuyos bytes tienen un pequeño peso en la función de creación de códigos de comprobación (es decir, los bytes que no se tienen en cuenta o que apenas se tienen en cuenta en la función de creación de códigos de comprobación).

[0040] El procedimiento 100 consta de una primera parte 110 que corresponde a la ejecución del módulo 12 de generación de la tabla de elección arbitraria 14.

[0041] Para cada evento E_i de la tabla 4, un objeto Alerta A_i se instancia primero en una etapa 112.

[0042] A continuación, en una etapa 114, se utiliza la función de creación de códigos de comprobación H con el fin de calcular el valor de elección arbitraria del objeto Alerta A_i instanciado.

[0043] Finalmente, en una etapa 116, se actualiza la tabla de elección arbitraria 14.

[0044] Cuando el valor de elección arbitraria Val_p calculado para el objeto Alerta A_i aún no existe en la tabla 14, se crea un nuevo índice en la tabla 14, asociándose con el valor calculado Val_p, un puntero Add_p que apunta a una dirección en un espacio de memoria 18 dedicado. A continuación, se registra en esta dirección, un doblete HMap_i cuyo primer campo corresponde al objeto Alerta A_i y un segundo campo se reserva para posiblemente recibir posteriormente un identificador de un archivo F_m.

[0045] Cuando el valor de elección arbitraria calculado Val_q para el objeto Alerta A_k instanciado ya existe en el índice de la tabla 14, es decir, que al menos otro objeto A_j tiene por valor de elección arbitraria el valor Val_q, la actualización de la tabla 14 consiste en agregar, a la dirección indicada por el puntero Add_q asociado con el valor Val_q en la tabla 14, un doblete HMap_k después del doblete HMap_j que consta del objeto A_j. El doblete HMap_k consta del objeto A_k.

[0046] Las etapas 112, 114 y 116 son iteradas para todos los eventos E_i de la tabla 14. Una vez que el conjunto de la tabla 4 ha sido examinado, el espacio de memoria 18 del motor 10 consta de un objeto Alerta A_i (es decir, una instanciación de la clase Alerta) para cada uno de los eventos E_i de la tabla 14.

[0047] Luego, de forma asíncrona, se realiza la segunda parte 120 del procedimiento 100. Esto corresponde a la ejecución del módulo 16 del motor 10.

[0048] En una etapa 122, se analiza el contenido de un archivo binario F_m con el fin de extraer posibles valores de una dirección IP de un emisor, un puerto emisor, una dirección IP de un destinatario, un puerto de un

destinatario y de un protocolo. Esta etapa corresponde a la implementación de un algoritmo optimizado y paralelizado de análisis sintáctico ("*parsing*" en inglés) del archivo F_m.

[0049] Luego, en la etapa 124, la función de creación de códigos de comprobación H de la clase de Alerta se aplica a los valores extraídos en la etapa 122. Esto lleva a la obtención de un valor de elección arbitraria Val_m.

[0050] La etapa 126 consiste entonces en verificar si el valor Val_m corresponde a un índice de la tabla de elección arbitraria 14.

[0051] En el caso de que no sea así, el procedimiento 100 se interrumpe y se reanuda en la etapa 122 para analizar el siguiente archivo F_m + 1 de la segunda fuente de datos 6.

[0052] En cambio, en caso de que sea afirmativo (es decir, si existe en la tabla 14 un valor Val_n igual a Val_m) la etapa 128 consiste en verificar si hay igualdad entre los valores extraídos en la etapa 122 y el objeto o uno de los objetos almacenados en el espacio de memoria 18 en la dirección Add_n asociada con el índice Val_n de la tabla 14.

[0053] Para ello, la función de igualdad de la clase Alerta se aplica entre los valores extraídos en la etapa 122 y el primer campo del primer doblete contenido en la dirección Add_n.

[0054] En caso de igualdad (caso de n=p en la figura 2), en la etapa 130, se escribe un identificador del archivo F_m en el segundo campo del doblete HMap_i. El identificador del archivo F_m es, por ejemplo, la ruta de acceso a este archivo. El procedimiento 100 se interrumpe y se reanuda en la etapa 122 para analizar el siguiente archivo F_m + 1.

[0055] Si no hay igualdad, la función de igualdad de la clase Alerta se aplica entre los valores extraídos en la etapa 122 y el primer campo del segundo doblete contenido en la dirección Add_n. La etapa 128 se itera hasta que el conjunto de los dobletes en la dirección Add_n hayan sido examinados sin éxito. El procedimiento 100 se interrumpe y luego se reanuda en la etapa 122 para analizar el siguiente archivo F_m + 1 de la pluralidad de archivos 6.

[0056] Una vez que se han procesado los diferentes archivos F_m de la segunda fuente de datos 6, se obtiene una tabla enriquecida 20 a partir del contenido del espacio de memoria 18. La tabla enriquecida 20 asocia, cuando existe, el o cada identificador de un archivo F_m, con los parámetros del evento E_i mencionados en la tabla 4.

[0057] Esta tabla enriquecida constituye una combinación entre las primera y segunda fuentes de datos 4 y 6. Por lo tanto, el operador puede estudiar mejor un evento considerado como una alerta de seguridad por el componente de seguridad 3, consultando los datos de aplicación recopilados por la sonda 5 a la que retorna la tabla enriquecida.

REIVINDICACIONES

1. Procedimiento de asociación (100) de una primera fuente de datos con una segunda fuente de datos, la primera fuente de datos (4) constituye una tabla generada por un componente de monitorización (3) de una red de comunicación (2), cada línea de la tabla corresponde a un evento (E_i) en dicha red que cumple con una regla de alerta predefinida,
la segunda fuente de datos (6) está constituida por una pluralidad de archivos binarios (F_m), cada archivo es generado por una sonda (3) de análisis del contenido de paquetes interceptados en dicha red, un archivo que recopila los datos de los paquetes que pertenecen a una misma sesión de comunicación en dicha red, el procedimiento consta de las etapas que consisten en:

- proporcionar una clase, denominada clase Alerta, definida por una pluralidad de variables, una función de creación de códigos de comprobación y una función de igualdad;
- para cada evento (E_i) de la primera fuente de datos (4), instanciar un objeto Alerta (A_i) de la clase Alerta y actualizar una tabla de elección arbitraria (14) calculando el valor de la función de creación de códigos de comprobación para dicho objeto Alerta instanciado;
- para cada archivo (F_m) de la segunda fuente de datos (6), analizar el archivo para extraer posibles valores para las variables de la clase Alerta;
- utilizar la función de creación de códigos de comprobación y la función de igualdad de la clase Alerta, verificar, por medio de la tabla de elección arbitraria (14), si los valores extraídos corresponden a un objeto Alerta instanciado; y,
- en caso de que sea afirmativo, asociar en la memoria un identificador del archivo de dicho objeto Alerta instanciado correspondiente, según el cual la función de creación de códigos de comprobación de la clase Alerta se define por:

$$H = \sum_{q=0}^{N-1} S_q .37^{N-1-q}$$

en la que N es el número de bytes utilizados para representar un objeto de la clase Alerta y S_q es un peso que permite ponderar el quinto byte.

2. Procedimiento según la reivindicación 1, en el que la red (2) es una red de comunicación IP.

3. Procedimiento según una cualquiera de las reivindicaciones anteriores, en el que la tabla de la primera fuente de datos (4) cumple con el formato IDMEF.

4. Procedimiento según una cualquiera de las reivindicaciones anteriores, en el que cada archivo binario (F_m) de la segunda fuente de datos (6) cumple con el formato ".pcap".

5. Procedimiento según una cualquiera de las reivindicaciones anteriores, en el que las variables de la clase Alerta son:

- la dirección IP de un emisor de un paquete correspondiente al evento (E_i) considerado;
- la dirección IP de un destinatario de dicho paquete;
- el puerto utilizado por dicho emisor;
- el puerto utilizado por dicho destinatario; y,
- el protocolo de comunicación utilizado entre dicho emisor y dicho destinatario.

6. Procedimiento según una cualquiera de las reivindicaciones 1 a 5, en el que el proceso de ejecución de la función de igualdad prevé que las variables de los objetos Alerta a comparar estén ordenadas, la programación de las variables depende de los pesos utilizados en la función de creación de códigos de comprobación.

7. Soporte de grabación de informaciones, **caracterizado porque** consta de instrucciones para la ejecución de un procedimiento de acuerdo con el procedimiento según una cualquiera de las reivindicaciones 1 a 6, cuando estas instrucciones son ejecutadas por un ordenador electrónico.

8. Sistema que consta de una unidad de control, **caracterizado porque** dicha unidad de control está programada para ejecutar un procedimiento de acuerdo con el procedimiento según una cualquiera de las reivindicaciones 1 a 6.

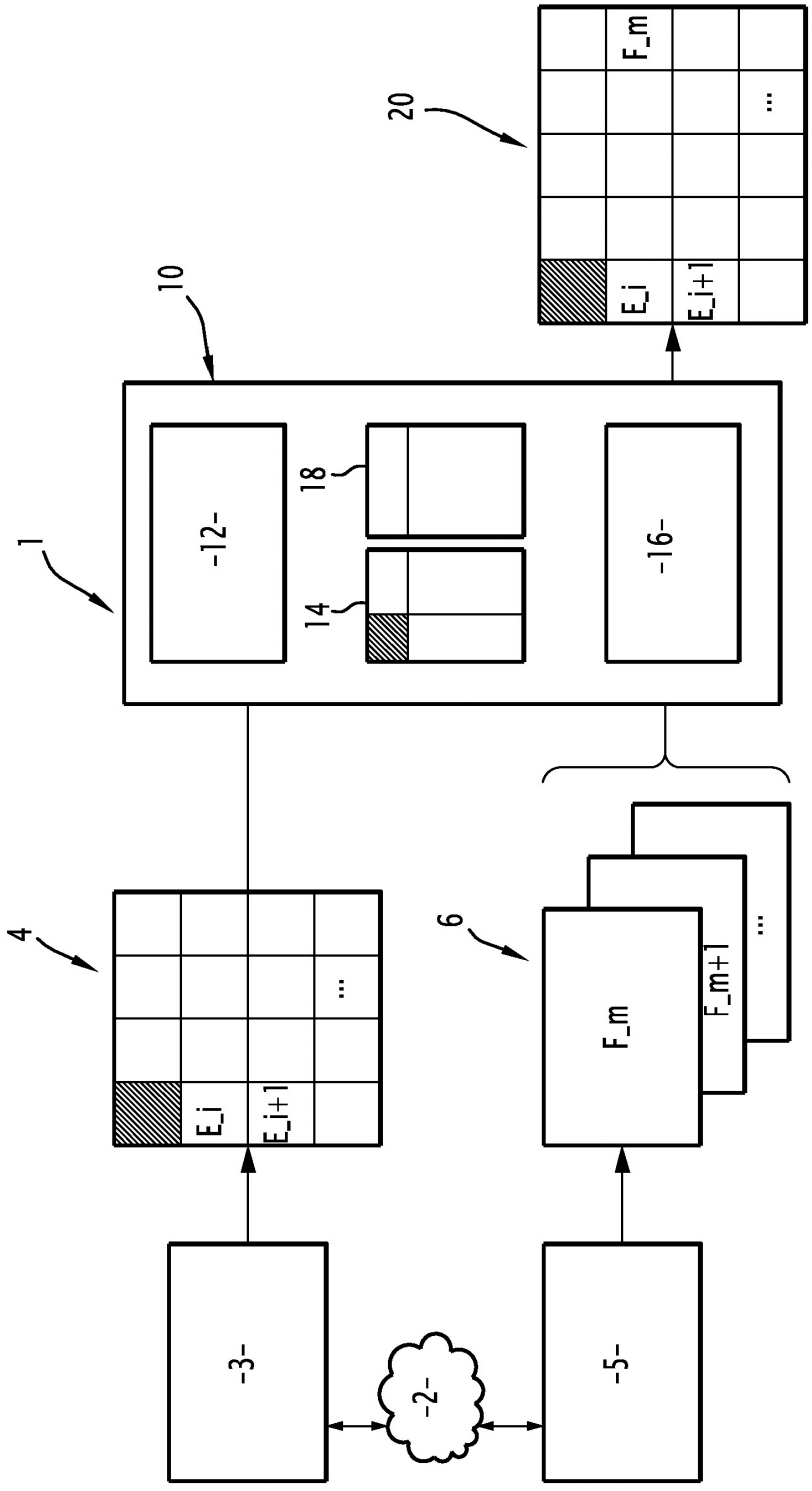

FIG.1

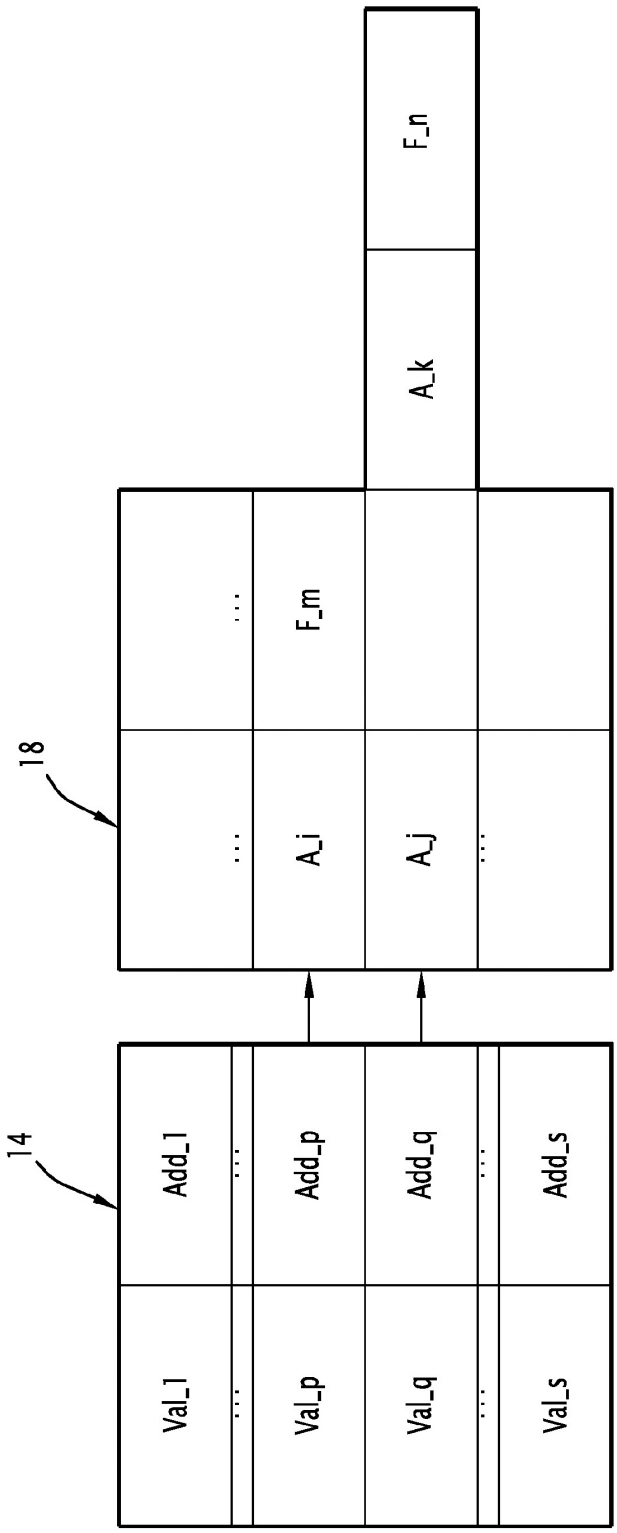

FIG.2

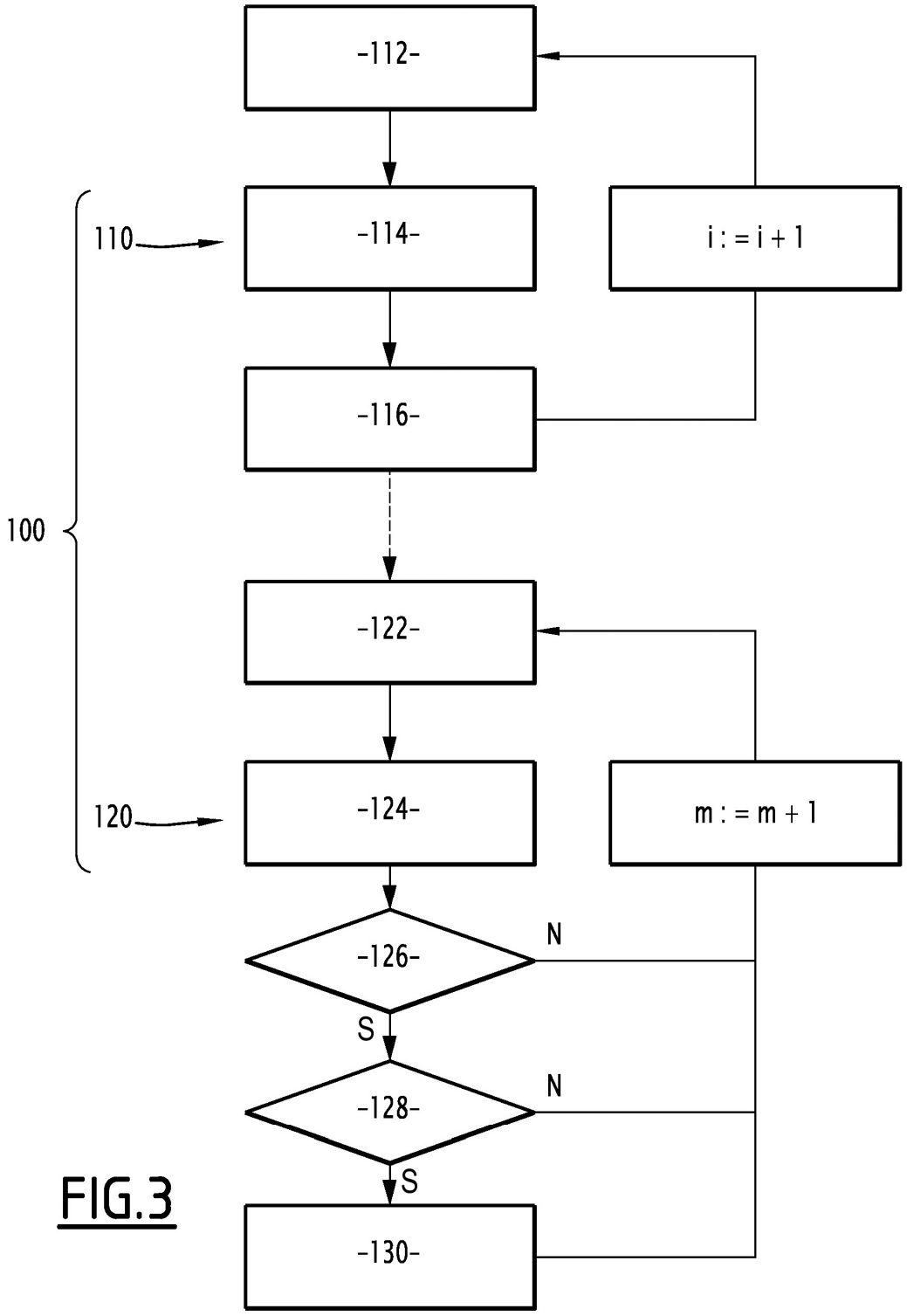

FIG.3