(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 577 143**

(51) Int. CI.:

H04L 29/06 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **23.12.2011 E 11805856 (9)**

(97) Fecha y número de publicación de la concesión europea: **06.04.2016 EP 2767056**

(54) Título: **Método y sistema para detectar software malintencionado**

(30) Prioridad:

14.10.2011 ES 201131650 P

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
13.07.2016

(73) Titular/es:

TELEFÓNICA, S.A. (100.0%)
Gran Vía, 28
28013 Madrid, ES

(72) Inventor/es:

ROMERO BUENO, FRANCISCO y
AMAYA CALVO, ANTONIO MANUEL

(74) Agente/Representante:

ARIZTI ACHA, Monica

Método y sistema para detectar software malintencionado

DESCRIPCIÓN

Campo de la técnica

La presente invención se refiere, en general, en un primer aspecto, a un método para detectar software malintencionado, realizándose dicha detección en un sistema de detección de anomalías, o ADS, analizando el comportamiento de una red y buscando desviaciones con respecto a una normalidad, indicando dicha normalidad el comportamiento común de usuarios de dicha red y definiéndose antes de dicha detección, y más particularmente a un método que comprende construir una pluralidad de modelos de detección, estando adaptado cada uno de dicha pluralidad de modelos de detección a diferentes entidades de dicha red y a diferentes algoritmos, implementando dichos diferentes algoritmos diferentes estrategias de detección y representando dicha pluralidad de modelos de detección dicha normalidad.

Un segundo aspecto de la invención se refiere a un sistema dispuesto para implementar el método del primer aspecto.

Estado de la técnica anterior

La detección de software malintencionado (malware) puede clasificarse de muchas maneras. Una de ellas es la categorización clásica que distingue entre la detección basada en host y basada en red. El primer tipo intenta encontrar evidencias de la existencia de virus, troyanos, etc. por medio de procesos, memoria física y varios otros análisis en host, mientras que el segundo se centra en las comunicaciones realizadas por tales virus o troyanos.

Surgieron varias estrategias en los comienzos de la detección basada en red. La más básica era bloquear determinadas comunicaciones que atraviesan un nodo de red especial, un cortafuego. Esta solución fue útil hasta que el malware, con el fin de ocultarse, empezó a usar protocolos usados ampliamente tales como HTTP, SMTP que no pueden bloquearse sin detener el negocio de los ISP y operadores.

Entonces, era obvio que no era suficiente analizar unos pocos campos de los paquetes de TCP/IP (protocolo, puertos de origen y destino y direcciones IP), y era necesario extender el proceso de monitorización a la cabecera de TCP/IP completa e incluso la carga útil de los paquetes. Es así como nacen los sistemas de detección de intrusión de red (NIDS o IDS) [15]. Los IDS se basan en firmas de tráfico, es decir, cadenas especiales que, cuando aparecen dentro de los paquetes de red, indican la presencia de un malware específico. Los IDS conocidos ampliamente son Snort [1] y Bro [2].

Aunque actualmente los IDS continúan desempeñando un papel importante en los procesos de monitorización de todos los operadores alrededor del mundo, puesto que tratan un porcentaje importante de detección de malware, los investigadores encontraron que algún malware permanecía indetectable a los cortafuegos e IDS. Estos virus, gusanos, etc. indetectables usaban tanto protocolos populares como contenido normal a priori dentro de la carga útil de paquetes. Algunos ejemplos son: spammers [3], bots orientados a la denegación de servicio (DoS) [4] o scanners [5]. Entonces, se observó que la única manera para detectar estas amenazas era analizar el comportamiento de la red con el fin de encontrar desviaciones con respecto a la normalidad, es decir, anomalías. La normalidad se define a través de una representación matemática de la realidad común, es decir, un modelo, que se construye en una etapa previa a la detección. Los sistemas de detección de anomalías de red (NADS) [14] o simplemente sistemas de detección de anomalías (ADS) tal como Proventia [6] demostraron rápidamente ser útiles para cubrir la laguna mencionada en el campo de monitorización. Y también en muchos otros escenarios, tales como detección de malware de día cero (nuevo malware existente para el que los IDS no tienen una firma válida) o tráfico cifrado (las firmas no son importantes cuando no puede verse la carga útil).

La detección de anomalías es todavía un área de investigación interesante que puede dar bastantes soluciones para los administradores de seguridad. Están apareciendo algoritmos innovadores de inteligencia artificial (AI), y el objetivo del modelado, que algunas veces es más importante que el propio algoritmo de detección, varía entre la granularidad más general, la totalidad de la red, hasta la mínima, el usuario final individual.

Ejemplos de sistemas de detección de anomalías pueden encontrarse en los documentos US2007/289013 y US8015133.

Pueden distinguirse dos tipos de problemas cuando se habla acerca de los ADS existentes, los relacionados con la eficacia y los relacionados con la eficiencia.

Los problemas de eficacia son evidentes en el día a día de todas las compañías. Sus sistemas ya implementados no lo detectan todo, y las cosas que deben detectar no se encuentran apropiadamente debido a estrategias y

mecanismos obsoletos.

- Modelado de comportamiento basado en red

Los modelos con respecto al comportamiento de redes monitorizadas siempre se basan en la agregación, contando el número de paquetes/flujos/bytes por unidad de tiempo y definiendo por tanto una referencia. Aunque es verdad que este enfoque es suficiente para afrontar amenazas importantes tales como ataques DoS [4], en los que se genera una gran cantidad de datos, no tiene nada que hacer con ataques más discretos y sofisticados basados en variaciones muy bajas de determinadas características de tráfico que sólo pueden apreciarse al nivel de entidad individual.

- Algoritmos de detección rudimentarios

Las reglas de condición-consecuencia y volúmenes/umbrales son casi todas las tecnologías implementadas en la mayoría de los NADS exitosos. Se trata de algoritmos muy básicos difíciles de mantener (deben considerarse nuevas condiciones cuando una amenaza evoluciona), mostrando muy baja flexibilidad (un umbral de 1000000 bytes superándose cuando se alcanzan 1000001 bytes) y una pobre auto-adaptación.

La mayoría de los problemas relacionados con la eficiencia con respecto a las soluciones existentes son debido a la falta de un marco común para fines de monitorización. Cada solución es propietaria y está muy cerrada a terceras partes, lo que hace que las organizaciones instalen una nueva batería de soluciones cada vez que surge una nueva amenaza.

- Nivel de integración nula entre proveedores

Excepto SIEM [9], una clase especial de sistemas no encargados de monitorización de red directa, pero encargados de la agregación y correlación de eventos de las diferentes fuentes, no se permite la integración entre sistemas de monitorización actuales (incluyendo los ADS), y no sólo sistemas de monitorización: herramientas de husmeado (sniffing), aplicaciones de registro, etc. Por ejemplo, es muy común que los IDS/IPS/ADS husmeen ellos mismos el tráfico de red, mientras que muchos sistemas de husmeado realizan la misma tarea.

- Bajo nivel de personalización/extensión

Uno de los principales problemas que tienen en particular los sistemas de monitorización y ADS con respecto a su arquitectura es la falta de flexibilidad. Las soluciones de detección de anomalías en investigación y comerciales actuales están muy cerradas y son propietarias, lo que hace casi imposible cualquier clase de personalización/extensión. Este bajo nivel de personalización puede parecer normal en cualquier otra clase de sistema de software, pero no en el campo de la monitorización, en el que los operadores de seguridad necesitan afrontar la evolución continua del malware por medio de nuevos algoritmos, estrategias, fuentes de datos, etc.

- Múltiples puntos de monitorización

Además, si una organización espera usar varios NADS (porque cada uno de estos tiene como objetivo una amenaza diferente) entonces cada NADS individual requerirá un punto de monitorización exclusivo desde el que obtener el tráfico sin procesar. Esto puede parecer un problema insignificante si proporcionar los puntos de monitorización no fuera costoso. Por un lado, las soluciones de derivación física deben cortar el cable durante unos pocos segundos, pero suficiente para detener aplicaciones críticas, y la división de la potencia óptica debe realizarse muy cuidadosamente con el fin de permitir que los nodos de extremo continúen negociando el enlace. Por otro lado, las derivaciones lógicas tales como la duplicación de puertos consumen una alta cantidad de capacidades de procesamiento en los nodos de red; adicionalmente, cada aplicación de monitorización necesita un puerto exclusivo en el nodo.

- Aplicaciones monolíticas

Finalmente, las soluciones existentes están diseñadas para ejecutarse en un único equipo, evitando una característica de distribución de procesamiento deseable. Las aplicaciones monolíticas habitualmente requieren grandes cantidades de recursos tales como CPU, memoria RAM y disco para afrontar redes de alta velocidad.

Descripción de la invención

Es necesario ofrecer una alternativa al estado de la técnica que cubra las lagunas encontradas en la misma, particularmente con relación a la falta de propuestas que en realidad permitan detectar todo el software malintencionado posible e implantar un marco común con fines de monitorización de manera que las organizaciones no necesiten instalar una nueva batería de soluciones cada vez que surge una nueva amenaza.

Para ello, la presente invención proporciona, en un primer aspecto, un método para detectar software malintencionado, realizándose dicha detección en un sistema de detección de anomalías, o ADS, analizando el comportamiento de una red y buscando desviaciones con respecto a una normalidad, indicando dicha normalidad el comportamiento común de los usuarios de dicha red y definiéndose antes de dicha detección.

A diferencia de las propuestas conocidas, el método de la invención, de una manera característica, comprende además construir una pluralidad de modelos de detección, estando adaptado cada uno de dicha pluralidad de modelos de detección a diferentes entidades de dicha red y a diferentes algoritmos, implementando dichos diferentes algoritmos diferentes estrategias de detección y representando dicha pluralidad de modelos de detección dicha normalidad.

Otras realizaciones del método del primer aspecto de la invención se describen según las reivindicaciones adjuntas 1 a 15, y en una sección posterior relativa a la descripción detallada de varias realizaciones.

Un segundo aspecto de la presente invención se refiere a un sistema para detectar software malintencionado, realizándose dicha detección en un sistema de detección de anomalías, o ADS, analizando el comportamiento de una red y buscando desviaciones con respecto a una normalidad, indicando dicha normalidad la realidad común de dicha red y definiéndose antes de dicha detección.

En el sistema del segundo aspecto de la invención, a diferencia de los sistemas conocidos mencionados en la sección de estado de la técnica anterior, y de una manera característica, éste comprende un módulo de sonda para la monitorización de tráfico de dicha red conectada a un módulo controlador encargado de realizar dicha detección, en el que dicho módulo controlador está dotado de una pluralidad de modelos de detección construidos por medio de un módulo compilador, estando adaptado cada uno de dicha pluralidad de modelos de detección a diferentes entidades de dicha red y a diferentes algoritmos, implementando dichos diferentes algoritmos diferentes estrategias de detección y representando dicha pluralidad de modelos de detección dicha normalidad.

El sistema del segundo aspecto de la invención está adaptado para implementar el método del primer aspecto.

Otras realizaciones del sistema del segundo aspecto de la invención se describen según las reivindicaciones adjuntas 16 a 22, y en una sección posterior relativa a la descripción detallada de varias realizaciones.

Breve descripción de los dibujos

Las anteriores y otras ventajas y características se entenderán de manera más completa a partir de la siguiente descripción detallada de realizaciones, con referencia a los dibujos adjuntos, que deben considerarse de una manera ilustrativa y no limitativa, en los que:

La figura 1 muestra los componentes y la interacción entre éstos, definiendo dichos componentes e interacciones una posible arquitectura del sistema propuesto, según una realización de la presente invención.
La figura 2 muestra el módulo de sonda, según una realización de la presente invención.
La figura 3 muestra el módulo de monitorización y la biblioteca de detectores, según una realización de la presente invención.
La figura 4 muestra el módulo compilador y la biblioteca de compiladores, según una realización de la presente invención.
La figura 5 muestra el módulo registrador y la biblioteca de registradores, según una realización de la presente invención.
La figura 6 muestra un posible diagrama de secuencia entre el módulo de monitorización y el módulo compilador, en el que el módulo de monitorización y el módulo compilador hacen uso de la interfaz compilador-monitorizador, según una realización de la presente invención.
La figura 7 muestra el diagrama de flujo para el detector de flujo de dominios basado en DNS, según una realización de la presente invención.
La figura 8 muestra una representación gráfica de agrupamientos cuando usan el agrupamiento de tráfico sospechoso, en el que se representan los vectores normales mediante “0”, las anomalías mediante “x” y los agrupamientos mediante círculos, según una realización de la presente invención.
La figura 9 muestra una representación gráfica de 2 características del algoritmo rápido de pertenencia al agrupamiento, según una realización de la presente invención.
La figura 10 muestra un ejemplo de implantación física distribuida de los componentes del sistema, según una realización de la presente invención.

Descripción detallada de varias realizaciones

La invención propuesta se refiere a un equipo de hardware y software (o conjunto de equipos si sus componentes se

distribuyen finalmente) que actúa como plataforma que permite mucha más eficacia y eficiencia en el campo de detección de anomalías de red.

La ganancia de eficacia se obtiene gracias a la combinación de dos enfoques innovadores:

- El uso de modelos de detección en un modo por usuario. En lugar de crear un único modelo para la totalidad de la red, cada entidad dentro de ésta se modela.
- El uso de modelos de detección en un modo por algoritmo. En lugar de tener una estrategia de detección única, múltiples ejemplos de detector se ejecutan en paralelo.

Los dos enfoques anteriores pueden verse como una matriz de modelo en la que se consideran N entidades ($E_1...E_N$) y M algoritmos ($A_1...A_M$), obteniendo NxM modelos diferentes ($M_{11}...M_{NM}$).

El modelo y sus respectivos detectores pueden ser o bien ADS del estado de la técnica (pueden considerarse matrices sencillas de 1x1, puesto que se usan una única entidad modelada y sólo un algoritmo) o bien algoritmos de detección innovadores, como los dos explicados más adelante basados en inteligencia artificial.

Además, otras características de arquitectura disponibles en la invención propuesta permiten mucha más eficiencia en términos de:

- La distribución del esfuerzo de procesamiento entre varios nodos físicos.
- Fácil integración de software relacionado con monitorización existente, tal como procesadores de tráfico de red, mediante el diseño de arquitectura.
- Un único punto de monitorización en la red.

• Componentes de la arquitectura

Los componentes del sistema y la interacción entre ellos se muestran en la figura 1. Los recuadros gris claro representan componentes SoA, el gris oscuro representa la SoA modificada y los recuadros blancos, que incluyen las bibliotecas de detectores y compiladores, son componentes innovadores dentro de la arquitectura. A continuación, se detallarán los componentes del sistema:

- SONDA

Es el único punto de monitorización que proporciona rastros de tráfico de red al resto de los componentes del sistema.

La detección de anomalías realizada por las aplicaciones de ADS de múltiples algoritmos se basa en el tráfico de red, específicamente buscando desviaciones respecto de la normalidad en los paquetes de TCP/IP que atraviesan la red monitorizada. Para realizar esto es necesario (1) capturar los paquetes de los medios físicos y (2) preparar los paquetes capturados para los algoritmos de detección, que habitualmente funcionan con flujos agregados (un flujo se define, al menos, por el protocolo de transporte, TCP o UDP, las direcciones IP de origen y destino y los puertos de origen y destino). La SONDA es el componente encargado de la captura de paquetes y del procesamiento en flujos, tal como se muestra en la figura 2.

Las SONDAS también pueden ser heredadas, para lo que es necesario simplemente implantar un adaptador con el fin de conseguir la interfaz entre las SONDAS y el MONITORIZADOR.

- MONITORIZADOR y BIBLIOTECA DE DETECTORES

El MONITORIZADOR es el controlador principal del sistema, recibe rastros de red desde la SONDA y es responsable (1) del almacenamiento de rastros si funciona en modo de entrenamiento y (2) la invocación de DETECTORES, pasándoles una copia de los rastros, si funciona en modo de detección. La BIBLIOTECA DE DETECTORES es una recopilación de DETECTORES que implementan cada uno un algoritmo de NADS.

Una vez que el tráfico de red se agrega en flujos está listo para analizarse usando una amplia diversidad de algoritmos de detección de anomalías. Sin embargo, debe recordarse que el tráfico puede o bien almacenarse (con el fin de construir los modelos normales) o bien usarse por los algoritmos de detección. Así, alguna entidad debe (1) conocer si el sistema está funcionando actualmente en modo de detección o de almacenamiento y (2) reenviar el tráfico al componente de almacenamiento o a las entidades de detección dependiendo del modo de funcionamiento. La invención propuesta implementa esto por medio del MONITORIZADOR.

Con respecto a la detección, puesto que la carpeta de algoritmos no puede estar completa (imposibilidad de implementar todos los algoritmos existentes, algoritmos que no se han descubierto todavía) sería muy conveniente

tener un mecanismo para ajustar el sistema de detección (añadir, eliminar o modificar algoritmos). La arquitectura del sistema proporciona un mecanismo acoplable de este tipo por medio de la BIBLIOTECA DE DETECTORES. La BIBLIOTECA DE DETECTORES es una recopilación discreta de algoritmos de detección, la recopilación actual, que tiene una interfaz definida con el MONITORIZADOR. Un nuevo algoritmo de detección, anteriormente un DETECTOR, puede incluirse en la recopilación si tal DETECTOR respeta la interfaz definida con el MONITORIZADOR. Todos los DETECTORES dentro de la biblioteca no tienen que usarse siempre, y si existe la posibilidad de enumerar el subconjunto deseado de algoritmos para el MONITORIZADOR. Esto puede realizarse, por ejemplo, por medio de archivos de configuración.

El MONITORIZADOR entonces reenvía simplemente tal como se dijo previamente, a través de la interfaz definida, los flujos recibidos a todos los DETECTORES dentro de la BIBLIOTECA DE DETECTORES. Finalmente, los DETECTORES comparan los flujos entrantes con los modelos de comportamiento normal almacenados.

La BIBLIOTECA DE DETECTORES está concebida originalmente, precisamente, como una biblioteca que se incorpora de manera nativa al software de MONITORIZADOR (bibliotecas estáticas o dinámicas en C/C++, archivos JAR en Java, etc.). El fin de esta decisión es mejorar la entrega de flujos a los DETECTORES, pero el desarrollador puede implementar la biblioteca como un conjunto de procesos que se ejecutan independientemente del MONITORIZADOR, por ejemplo; incluso, los DETECTORES pueden colocarse en dispositivos físicos separados.

- COMPILADOR y BIBLIOTECA DE COMPILADORES

El COMPILADOR es responsable de la invocación de COMPILADORES cuando el sistema funciona en modo de entrenamiento. Se ejecuta una vez que la fase de almacenamiento de rastros termina. La BIBLIOTECA DE COMPILADORES es una recopilación de COMPILADORES que implementan cada uno un mecanismo de modelado de comportamiento.

El COMPILADOR es el módulo de la arquitectura de ADS de múltiples algoritmos implicada en la generación de modelos de comportamiento normal y, por tanto, es el componente central de la invención propuesta. Tal como se conoce, los algoritmos de detección de anomalías generalmente comparan el tráfico observado con un modelo normal, buscando desviaciones entre uno y otro. Puesto que los algoritmos de detección pueden ser muy diferentes, entonces sus modelos de referencia también serán muy diferentes a pesar de que los datos usados para generarlos son los mismos (el tráfico capturado y agregado). El COMPILADOR es responsable de generar esta generación de modelos diferenciados a través de la BIBLIOTECA DE COMPILADORES, que contiene un generador de modelos, anteriormente un COMPILADOR, por algoritmo de detección en el sistema. Los COMPILADORES dentro de la biblioteca pueden habilitarse o deshabilitarse.

Sin embargo, el COMPILADOR es un componente opcional puesto que algunos DETECTORES podrían no necesitar un modelo personalizado (un modelo por defecto se usa siempre independientemente del comportamiento específico de los usuarios); o los modelos pueden haberse generado por otros medios; o incluso el DETECTOR no necesita un modelo.

- REGISTRADOR y BIBLIOTECA DE REGISTRADORES

El REGISTRADOR recibe alertas desde el MONITORIZADOR e invoca a los REGISTRADORES. La BIBLIOTECA DE REGISTRADORES es una recopilación de REGISTRADORES implementando cada uno una facilidad de registro.

Finalmente, las alarmas colaborativas generadas pueden registrarse, tarea para la cual el componente de REGISTRADOR se introduce en la arquitectura (una vez más, una carpeta de registro exhaustiva es inviable, por lo que se usa una BIBLIOTECA DE REGISTRADORES para añadir o eliminar dinámicamente facilidades de registro: archivos, bases de datos, syslog, etc.). Los REGISTRADORES existentes pueden habilitarse o deshabilitarse en la configuración.

• Interfaces de arquitectura

A continuación, se describirán las interfaces entre los componentes del sistema:

- SONDA - MONITORIZADOR

Esta interfaz define cómo los flujos generados en el componente SONDA se envían al componente MONITORIZADOR. Las comunicaciones, básicamente, pueden implementar dos esquemas diferentes dependiendo de manera proactiva de la SONDA:

1. Intercambio proactivo: la SONDA envía los flujos en el momento en que están listos para compartirse.

2. Intercambio a petición: los flujos se envían cuando el MONITORIZADOR los solicita.

El uso de uno u otro esquema tiene un importante impacto en el rendimiento en tiempo real. Por un lado, es obvio que el intercambio a petición no es compatible con el tiempo real puesto que el reenvío de datos depende de la disponibilidad del MONITORIZADOR. Por otro lado, puede pensarse que el intercambio proactivo es siempre compatible con el tiempo real; pero esto puede no ser cierto si el componente MONITORIZADOR realiza de nuevo un almacenamiento en memoria intermedia de los datos recibidos proactivamente.

En cualquier caso, debe establecerse el formato de los flujos intercambiados con el fin de garantizar la correcta interacción entre las SONDAS y el MONITORIZADOR. Si se usan fuentes de datos heredadas entonces debe implementarse un adaptador con el fin de garantizar su disponibilidad. Una lista no exhaustiva de campos que pueden intercambiarse entre las SONDAS y el MONITORIZADOR es la siguiente:

- Protocolo de capa 4 (TCP, UDP, etc.).
- Sellos de fecha y hora del primer y último paquete.
- Direcciones IP de origen y destino en el flujo.
- Puertos de TCP o UDP de origen y destino en el flujo.
- Número de paquetes enviados y recibidos.
- Número de bytes enviados y recibidos.
- Estado de TCP o, al menos, banderas de TCP implicadas en el flujo.
- Determinados datos de cabecera de aplicación, tales como DNS, HTTP, SMTP, SIP y otros.

- MONITORIZADOR - COMPILADOR

Como se indicó en las secciones de MONITORIZADOR y COMPILADOR específicas, la comunicación entre estos dos componentes de la arquitectura de la invención propuesta se realiza por medio de bases de datos. Dos son los motivos para esto:

1. La alta cantidad de flujos capturados no permite un almacenamiento en memoria intermedia sencillo.
2. La mayoría de los algoritmos de detección de anomalías necesitan amplias ventanas de tiempo para realizar sus modelos de comportamiento normal. Naturalmente, es posible una construcción de modelos en tiempo real en algunos algoritmos de detección de anomalías, pero, puesto que esta tarea no es crítica, siempre se realizará en un modo no en tiempo real.

La base de datos implementada debe permitir al MONITORIZADOR almacenar los flujos construidos en el formato de intercambio acordado entre la SONDA y el MONITORIZADOR (véase sección anterior). Estos flujos almacenados pueden consultarse por el COMPILADOR también, que generará los modelos y los almacenará en otra base de datos.

- MONITORIZADOR - REGISTRADOR

Las alarmas generadas por el MONITORIZADOR se envían al componente REGISTRADOR en un formato específico, para el que los campos candidatos podrían ser los siguientes:

- Identificador del atacante.
- Identificador del host atacado.
- Protocolo implicado.
- Sello de fecha y hora para la alerta.
- Tipo de ataque/tipo de anomalía.
- Identificador de DETECTOR.
- Información de realimentación tal como datos originales que provocan que se establezca la alarma en el DETECTOR, lista de objetivos extendida (si no hay uno solo), etc.

• Detectores

El sistema prevé el uso de algoritmos avanzados para mejorar la eficacia de los sistemas de detección de anomalías. Estos algoritmos avanzados proceden del campo de inteligencia artificial, siendo las redes neuronales y los algoritmos de agrupamiento los candidatos principales para usarse.

Las redes neuronales y otros algoritmos de aprendizaje de máquina supervisados [10] han demostrado varias veces su potencia en problemas de clasificación, debido a su capacidad para generalizar soluciones por medio de unos pocos ejemplos de entrenamiento; su adaptabilidad; y su baja tasa de falsos positivos.

Sin embargo, no siempre es posible usar algoritmos supervisados, especialmente cuando no hay un experto que

pueda etiquetar o clasificar previamente los ejemplos de entrenamiento. Cuando esto se produce, se necesitan los algoritmos no supervisados [11] tal como de agrupamiento. El agrupamiento es muy útil en auto-generación de clases de tráfico, comportamientos, etc.

En este caso se detallan un par de DETECTORES que forman parte de la BIBLIOTECA DE DETECTORES. Estos DETECTORES son ejemplos de los algoritmos de detección avanzados contemplados para el ADS de múltiples algoritmos propuesto. Otros algoritmos sencillos tales como monitorización de tráfico volumétrico, recuentos absolutos de paquetes o flujos entre nodos, y la detección periódica de flujos no se describen puesto que no son algoritmos innovadores aunque pueden desarrollarse perfectamente sobre la invención propuesta.

DETECTORES adicionales no documentados en el presente documento pueden añadirse al ADS de múltiples algoritmos en forma de extensiones de la patente actual.

- Detector de flujo de dominios basado en DNS

Este DETECTOR intenta detectar actividades de flujo de dominios [12] en el tráfico de DNS monitorizado, que puede indicar la presencia de un bot [7].

Los bots dentro de una botnet habitualmente implementan consultas de DNS con el fin de descubrir su servidor de mando y control (C&C); esto permite a los dueños de bots cambiar la ubicación real (la dirección IP) del servidor sin reconfigurar sus bots. Mientras que los FQDN fijos son fáciles de detectar y filtrar, los bots implementan la técnica de flujo de dominios, que genera dinámicamente una alta cantidad de FQDN para el servidor C&C con el fin de sincronizarse con el dueño. Estos FQDN dinámicamente generados pueden basarse en el sello de fecha y hora actual, o pueden ser un conjunto generado de manera pseudoaleatoria, siendo sólo una de las posibilidades la válida. En cualquier caso, esta técnica genera muchas respuestas de NX_DOMAIN (y otras) cuando se consulta el servidor DNS. Este detector analiza estas respuestas anómalas.

Por tanto, se analizan las respuestas de DNS y se extrae un conjunto de características dentro de un intervalo de tiempo, siendo lo siguiente un ejemplo de conjunto de características:

- Número de respuestas de NX_DOMAIN.
- Número de respuestas de FORMAT_ERROR.
- Número de respuestas de REFUSED.
- Número de respuestas de SERVER_FAILURE.
- Número de diferentes FQDN consultados.
- Número de FQDN de una capa
- Número de FQDN de dos capas.
- Número de FQDN de tres capas o más.
- Número de dominios de nivel superior sospechosos (TLD).
- Número de dominios de capa dos con longitud inferior a 6.
- Número de dominios de capa dos con longitud superior a 5 e inferior a 21.
- Número de dominios de capa dos con longitud superior a 20.
- Número de dominios de capa dos con número de vocales inferior al 0,3%
- Número de dominios de capa dos con número de dígitos superior al 0,5%

Estas características componen un vector de características que se evalúa usando una red neuronal. Esta red neuronal se entrena (supervisa) previamente usando ejemplos de vectores que contienen valores para todas las características anteriores y proporcionando un campo adicional, una etiqueta. Esta etiqueta proporcionará información con respecto a la conveniencia de establecer una alarma o no cuando se encuentre un vector parecido en el tráfico.

El DETECTOR usa, por tanto, un modelo único por defecto, construido previamente, la configuración de red neuronal específica que resulta tras la fase de entrenamiento.

Se proporcionó en la figura 7 un diagrama de flujo para ilustrar el DETECTOR propuesto, en el que se describen algunas variables en la siguiente tabla:

Nombre de variable	Descripción
f	Flujo
sid	Identificador de abonado
m	Modelo para el identificador de abonado (actualmente uno por defecto)
td	¿Está el flujo actual relacionado con un dominio superior?

ts	Sello de fecha y hora para el flujo actual
tw	Sello de fecha y hora de inicio de la ventana de tiempo actual
tw'	tw actualizado
s	Tamaño de la ventana de tiempo
ctw	¿Está el flujo actual en la ventana de tiempo actual?
a	Alerta
v	Vector de características que se obtiene del flujo actual
gv	Vector acumulado global con respecto a la ventana de tiempo actual
gv'	gv actualizado

• Agrupamiento de tráfico sospechoso

Los fundamentos de este DETECTOR se basan en la construcción de un modelo personalizado con respecto a los valores de determinadas características dentro de un periodo de intervalo. A diferencia del DETECTOR anterior, no se usa ningún experto para entrenarlo, es decir, se va a utilizar un algoritmo no supervisado. Específicamente, un algoritmo de agrupamiento de maximización de la esperanza (EM) [13] definirá las clases de tráfico en las que cada usuario está comúnmente implicado. Entonces, en una segunda etapa, se detectarán las anomalías si aparece un patrón de tráfico diferente del normal. Gráficamente, este proceso se muestra en la imagen simplificada de 2 características de la figura 8, en la que los círculos representan agrupamientos de datos, la "o" representa un vector normal y una "x" es una anomalía.

Sin embargo, el punto clave no es el algoritmo sino las características de los vectores que alimentarán la implementación de EM. Características tales como el número de paquetes/bytes/flujos enviados y recibidos parecen ser candidatos válidos, pero estas características no son estadísticamente estables a lo largo del tiempo. Por el contrario, se necesitan características estables; son mejores aquéllas relacionadas, precisamente, con comportamientos anormales. Este DETECTOR considera al menos las siguientes (recuentos con respecto a un intervalo de tiempo):

- Número de flujos que tienen el destino ubicado en un país inusual.
- Número de flujos que tienen el destino incluido en una lista negra.
- Número de flujos que usan un protocolo inusual.
- Número de flujos de TCP sospechosos (sólo flujos SYN, flujos sólo SYN+ACK, flujos sólo FIN, flujos sólo FIN+ACK).
- Número de flujos de más de 10 KB de longitud.
- Número de flujos de más de 50 KB de longitud.
- Número de flujos muy pequeños (menos de la mitad de MTU).
- Número de flujos no generados por la entidad modelada.

La mayoría de las características anteriores deben tener valores diferentes de cero, pero muy próximos a cero, en el caso normal. Esto no es relevante puesto que un usuario puede acceder a uno o dos servidores legítimos ubicados en un país inusual, por ejemplo; lo importante es encontrar valores anormales.

Los países inusuales y protocolos inusuales se definen realizando una estadística previa con respecto a países a los que se accede más y protocolos más usados durante un determinado intervalo de tiempo.

El diagrama de flujo para el DETECTOR de agrupamiento de tráfico sospechoso es el mismo que en el DETECTOR de flujo de dominios basado en DNS, pero siendo la función de evaluación una diferente. En este caso la función de evaluación verifica si el vector acumulado se correlaciona con algún agrupamiento dentro del modelo (es comportamiento normal) o no (es una anomalía). En este sentido se propone un algoritmo rápido de pertenencia al agrupamiento.

Las funciones de pertenencia al agrupamiento pueden ser complejas, especialmente cuando se usan grandes vectores de características. En este caso se presenta una rápida función de evaluación, basándose en una simplificación de los agrupamientos obtenidos: para cada agrupamiento y para cada dimensión o característica se calculan sus límites, es decir, los valores mínimo y máximo que puede adoptar la característica con respecto al agrupamiento actual. Entonces, la función de pertenencia al agrupamiento es tan sencilla como verificar si cada característica dentro del vector evaluado está dentro de los límites de la misma característica del agrupamiento. La idea puede observarse fácilmente en un espacio de 2 características, tal como se muestra en la figura 9.

■ Ventajas de la invención

1. Ventajas de eficacia

- Modelado del comportamiento basándose en entidad y basándose en algoritmo

La invención propuesta permite el modelado del comportamiento basándose en entidad y basándose en algoritmo, abordando estadísticamente pocos y sofisticados ataques que no se detectarían basándose en red. Sin embargo, este ADS de múltiples algoritmos permite también la definición de modelos de comportamiento global.

- Algoritmos de detección avanzada

Los algoritmos detallados antes se basan en algoritmos de inteligencia artificial compleja y de aprendizaje de máquina que proporcionan mecanismos flexibles, de auto-adaptación, de autoaprendizaje y de precisión para detectar tanto anomalías generales como comportamientos anómalos específicos de tráfico, tales como flujo de dominios.

2. Ventajas de eficiencia

- Un marco común para el desarrollo de aplicaciones de monitorización

La arquitectura mostrada en este documento tiene como objetivo ser un modelo de referencia a la hora de implementar sistemas de detección colaborativa, específicamente aplicaciones de detección de anomalías colaborativas. Tal como puede verse en la sección de estado de la técnica anterior, muchos intentos para definir tal clase de aplicaciones han conducido a una confusión de conceptos que no ayuda a conocer si una arquitectura ha emergido de un algoritmo o viceversa, y lo peor, hace difícil integrar tanto nuevos algoritmos como nuevos requisitos de arquitectura. El sistema propuesto da a los desarrolladores un marco común para diseñar nuevas aplicaciones de detección de anomalías, pero también añade esas nuevas aplicaciones a las existentes en un modo aséptico. Esto es claramente una ventaja para los administradores de red (principalmente operadores de telecomunicaciones) que desean incluir en su portafolio de sistemas de detección una nueva estrategia o algoritmo de detección sin que interfiera con los existentes.

- Distribución de procesamiento

Todos los componentes de esta arquitectura pueden distribuirse fácilmente entre varios dispositivos físicos; el único requisito es que el desarrollador de los NADS implemente las interfaces entre los módulos usando una de las muchas tecnologías de comunicaciones tales como sockets, servicios web, RPC, RMI, etc. La figura 10 muestra un ejemplo de tal implantación distribuida en de cuatro servidores diferentes.

La distribución de procesamiento también ayuda en la escalabilidad del sistema desarrollado, puesto que la inclusión de un algoritmo de procesamiento intenso puede resolverse, por ejemplo, incluyendo un servidor especializado para ese algoritmo de detección e implementando una interfaz basada en TCP/IP con el MONITORIZADOR.

- Un solo punto de monitorización para una amplia diversidad de aplicaciones

El lector debe tener en cuenta que los puntos de monitorización están limitados debido a las restricciones técnicas (las derivaciones de fibra hacen más débil la señal óptica cuando se divide; la duplicación de puertos en encaminadores o conmutadores consume bastantes recursos que pueden restarse del procesamiento de tráfico; o directamente el negocio no puede detenerse cortando la línea, al fin y al cabo, debe realizarse o bien una derivación física o bien una duplicación lógica). Tal como puede observarse, el ADS de múltiples algoritmos sólo necesita un único punto de monitorización común a todas las aplicaciones implantadas.

- Integración de sondas heredadas

El componente SONDA no necesita desarrollarse desde cero. Pueden usarse muchos otros sistemas de generación de flujo ampliamente conocidos tales como Netflow [8]. El único requisito para realizar tal integración es incluir una etapa de normalización.

Un experto en la técnica puede introducir cambios y modificaciones en las realizaciones descritas sin apartarse del alcance de la invención tal como se define en las reivindicaciones adjuntas.

Siglas

ADS	Anomaly Detection System; sistema de detección de anomalías
AI	Artificial Intelligence; inteligencia artificial
CPU	Central Processing Unit; unidad central de procesamiento
C&C	Command and Control; mando y control

DNS	Domain Name System; sistema de nombres de dominio
DoS	Denial of Service; denegación de servicio
EM	Expectation-Maximization; maximización de la esperanza
FQDN	Fully Qualified Domain Name; nombre de dominio completamente calificado
HTTP	Hyper-Text Transfer Protocol; protocolo de transferencia de hipertexto
IP	Internet Protocol; protocolo de Internet
IDS	Intrusion Detection System; sistema de detección de intrusión
ISP	Internet Service Provider; proveedor de servicio de Internet
KB	Kilo Bytes; kilobytes
MTU	Maximum Transfer Unit; unidad de transferencia máxima
NADS	Network Anomaly Detection System; sistema de detección de anomalías de red
NIDS	Network Intrusion Detection System; sistema de detección de intrusión de red
RAM	Random Access Memory; memoria de acceso aleatorio
SIEM	Security Information and Events Management; gestión de eventos e información de seguridad
SMTP	Simple Mail Transfer Protocol; protocolo simple de transferencia de correo
SOA	Service Oriented Architecture; arquitectura orientada a servicio
TCP	Transport Control Protocol; protocolo de control de transporte
TLD	Top Layer Domain; dominio de capa superior
UDP	User Datagram Protocol; protocolo de datagramas de usuario

Bibliografía

[1] Snort IDS. http://www.snort.org/

[2] Bro IDS. http://bro-ids.org/

[3] “Spambot” en Wikipedia. http://en.wikipedia.org/wiki/Spambot

[4] “Denial-of-service attack” en Wikipedia. http://en.wikipedia.org/wiki/Denial-of-service_attack

[5] “Port scanner” en Wikipedia. http://en.wikipedia.org/wiki/Port_scanner

[6] Proventia ADS por IBM. http://www-935.ibm.com/services/uk/index.wss/offering/iss/y1026942

[7] “Botnets” en Wikipedia. http://en.wikipedia.org/wiki/Botnets

[8] Cisco IOS NetFlow. http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html

[9] “SEM” en Wikipedia. http://en.wikipedia.org/wiki/Security_event_manager

[10] “Supervised learning” en Wikipedia. http://en.wikipedia.org/wiki/Supervised_learning

[11] “Unsupervised learning” en Wikipedia. http://en.wikipedia.org/wiki/Unsupervised_learning

[12] “Fast-flux” en Wikipedia. http://en.wikipedia.org/wiki/Fast_flux

[13] “Expectation-maximization algorithm” en Wikipedia http://en.wikipedia.org/wiki/Expectation-maximization_algorithm

[14] “Anomaly detection” en Wikipedia http://en.wikipedia.org/wiki/Anomaly_detection

[15] “Intrusion detection system” en Wikipedia http://en.wikipedia.org/wiki/Intrusion-detection_system

REIVINDICACIONES

1. Método para detectar software malintencionado, realizándose dicha detección en un sistema de detección de anomalías, o ADS, analizando el comportamiento de una red y buscando desviaciones con respecto a una normalidad, indicando dicha normalidad el comportamiento común de usuarios de dicha red y definiéndose antes de dicha detección, comprendiendo dicho método:

- construir una pluralidad de modelos de detección para cada una de una pluralidad de diferentes entidades de dicha red, cada uno de dicha pluralidad de modelos de detección adaptado a dichas diferentes entidades de dicha red y a diferentes algoritmos, implementando dichos diferentes algoritmos diferentes estrategias de detección y representando dicha pluralidad de modelos de detección dicha normalidad, y
- representando dicha pluralidad de modelos de detección en una matriz bidimensional, correspondiendo una dimensión de dicha matriz a un número de dichas diferentes entidades de dicha red y correspondiendo la otra dimensión de dicha matriz a un número de dichos diferentes algoritmos empleados.

2. Método según la reivindicación 1, que comprende monitorizar el tráfico de dicha red, comprendiendo dicho tráfico paquetes que atraviesan dicha red, y preparar dichos paquetes para dichos diferentes algoritmos, o para dichas diferentes estrategias de detección, agregando dichos paquetes en flujos.

3. Método según la reivindicación 2, que comprende almacenar dicho tráfico con el fin de construir dicha pluralidad de modelos de detección cuando dicho ADS está operando en modo de almacenamiento.

4. Método según la reivindicación 2 o 3, que comprende:

- procesar dichos flujos según al menos parte de dichos diferentes algoritmos cuando dicho ADS está operando en modo de detección, definiéndose dichos diferentes algoritmos en una biblioteca de detectores, permitiendo dicha biblioteca de detectores al menos añadir, eliminar o modificar algoritmos; y
- comparar dichos flujos procesados con al menos parte de dicha pluralidad de modelos de detección.

5. Método según la reivindicación 4 cuando depende de la reivindicación 3, que comprende construir dicha pluralidad de modelos de detección según una biblioteca de compiladores que contiene un generador de modelos por algoritmo, definiendo cada generador de modelos un compilador que va a usarse con dicho tráfico almacenado cuando se opera en dicho modo de almacenamiento.

6. Método según la reivindicación 5, que comprende tener un generador de modelos por defecto contenido en dicha biblioteca de compiladores para construir al menos parte de dicha pluralidad de modelos de detección.

7. Método según la reivindicación 4, 5 o 6, que comprende registrar alarmas generadas cuando se detecta dicho software malintencionado según dicha comparación entre dichos flujos procesados con dicha al menos parte de dichos diferentes algoritmos y dicha al menos parte de dicha pluralidad de modelos de detección.

8. Método según la reivindicación 7, en el que dichas alarmas contienen al menos parte de la información de la siguiente lista no cerrada: identificador del atacante, identificador del host atacado, protocolo implicado, sello de fecha y hora para la alerta, tipo de ataque o tipo de anomalía, identificador de algoritmo e información de realimentación.

9. Método según la reivindicación 8, en el que se definen dichos diferentes algoritmos según redes neuronales, algoritmos de aprendizaje de máquina supervisados, algoritmos de agrupamiento y/o algoritmos sencillos de la siguiente lista no cerrada: monitorización de tráfico volumétrico, recuentos absolutos de paquetes o flujos entre nodos y detección periódica de flujos.

10. Método según la reivindicación 9, que comprende implementar un algoritmo dado según un detector de flujo de dominios basado en el sistema de nombres de dominio, o DNS, detectando dicho algoritmo dado actividades de flujo de dominios en un tráfico de DNS monitorizado analizando las respuestas de DNS, comprendiendo dicho análisis de respuestas de DNS:

- extraer una pluralidad de características a partir de dichas respuestas de DNS;
- construir un vector de características con dicha pluralidad de características;
- evaluar dicho vector de características usando una red neuronal, entrenándose previamente dicha red neuronal con ejemplos de vectores; y
- proporcionar, mediante dicha red neuronal, una etiqueta que indica la conveniencia de establecer una alarma según dicha evaluación.

11. Método según la reivindicación 10, en el que al menos parte de dicha pluralidad de características están

contenidas en la siguiente lista no cerrada: número de respuestas de NX_DOMAIN, número de respuestas de FORMAT_ERROR, número de respuestas de REFUSED, número de respuestas de SERVER_FAILURES, número de diferentes FQDN consultados, número de FQDN de una capa, número de FQDN de dos capas, número de FQDN de tres capas o más, número de dominios de nivel superior sospechosos, número de dominios de capa dos con longitud inferior a 6, número de dominios de capa dos con longitud superior a 5 e inferior a 21, número de dominios de capa dos con longitud superior a 20, número de dominios de capa dos con número de vocales inferior al 0,3 % y número de dominios de capa dos con número de dígitos superior al 0,5 %.

12. Método según la reivindicación 8, que comprende implementar un algoritmo concreto según un algoritmo de agrupamiento de maximización de esperanza, en el que dicho algoritmo concreto identifica las clases de tráfico, agrupa dichas clases de tráfico en agrupamientos y detecta una anomalía si aparece un patrón de tráfico que no pertenece a uno de dichos agrupamientos.

13. Método según la reivindicación 12, que comprende realimentar dicho algoritmo concreto con un vector de características de dichos flujos, siendo dichas características estables a lo largo del tiempo y estando contenidas en la siguiente lista no cerrada: número de flujos que tienen destino ubicado en un país inusual, número de flujos que tienen destino en una lista negra, número de flujos que usan un protocolo inusual, número de flujos de TCP sospechosos, número de flujos de más de 10 KB de longitud, número de flujos de más de 50 KB de longitud, número de flujos inferior a la mitad de la unidad de transmisión máxima y número de flujos no generados por una entidad modelada, en el que dichos flujos de TCP sospechosos comprenden flujos sólo SYN, flujos sólo SYN+ACK, flujos sólo FIN y flujos sólo FIN+ACK.

14. Método según la reivindicación 13, que comprende calcular los límites de cada uno de dichos agrupamientos o de cada característica de dicho vector de características, y detectar una anomalía si un valor de una característica de cada vector de características está fuera de los límites del agrupamiento correspondiente o característica correspondiente de dicho vector de características.

15. Sistema para detectar software malintencionado, realizándose dicha detección en un sistema de detección de anomalías, o ADS, analizando el comportamiento de una red y buscando desviaciones con respecto a una normalidad, indicando dicha normalidad el comportamiento común de los usuarios de dicha red y definiéndose antes de dicha detección, comprendiendo dicho sistema un módulo de sonda para la monitorización de tráfico de dicha red conectada a un módulo controlador encargado de realizar dicha detección, en el que dicho módulo controlador está dotado de una pluralidad de modelos de detección construidos por medio de un módulo compilador para cada una de una pluralidad de diferentes entidades de dicha red, cada uno de dicha pluralidad de modelos de detección adaptado a dichas diferentes entidades de dicha red y a diferentes algoritmos, implementando dichos diferentes algoritmos diferentes estrategias de detección y representando dicha pluralidad de modelos de detección dicha normalidad.

16. Sistema según la reivindicación 15, en el que una primera interfaz entre dicho módulo de sonda y dicho módulo controlador permite enviar rastros de tráfico en forma de flujos desde dicho monitorizador de sonda a dicho módulo controlador cada vez que dichos flujos están listos para compartirse o cuando dicho módulo controlador solicita dichos flujos.

17. Sistema según la reivindicación 16, en el que dicho módulo de sonda adapta dichos rastros de tráfico a los flujos y permite la disponibilidad en dichos flujos de al menos parte de los siguientes campos: protocolo de capa 4, sellos de fecha y hora del primer y último paquete, direcciones IP de origen y destino en el flujo, número de paquetes enviados y recibidos, número de bytes enviados y recibidos, banderas de TCP implicadas en el flujo y datos de cabecera de aplicación.

18. Sistema según la reivindicación 16 o 17 en el que dicho módulo controlador está dotado de una base de datos de flujos en la que se almacenan al menos dichos flujos y dicho módulo compilador está dotado de una base de datos de modelos en la que se almacenan al menos dicha pluralidad de modelos de detecciones.

19. Sistema según la reivindicación 18, en el que se implanta una segunda interfaz entre dicho módulo controlador y dicho módulo compilador con el fin de permitir la comunicación entre dicho módulo controlador y dicha base de datos de flujos, entre dicha base de datos de flujos y dicho módulo compilador y entre dicho módulo compilador y dicha base de datos de modelos.

20. Sistema según cualquiera de las reivindicaciones anteriores 15 a 19, en el que un módulo registrador conectado a dicho módulo controlador se proporciona para registrar alarmas generadas por dicho módulo controlador cuando se detecta dicho software malintencionado, comunicándose dicho módulo controlador y dicho módulo registrador por medio de una tercera interfaz.

21. Sistema según la reivindicación 20, en el que dicho módulo de sonda, dicho módulo controlador, dicho módulo

compilador y/o dicho módulo registrador están distribuidos en diferentes servidores o dispositivos físicos, usando dicha primera interfaz, dicha segunda interfaz y/o dicha tercera interfaz al menos una de las tecnologías de comunicación de la siguiente lista no cerrada: sockets, servicios web, comandos RPC e invocación de método remoto.

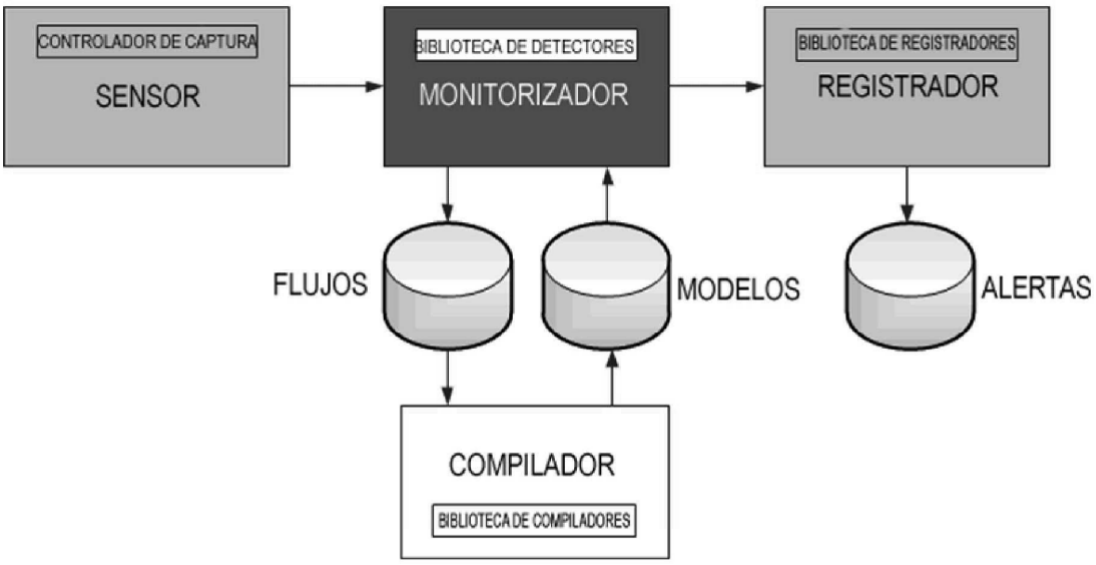

Figura 1

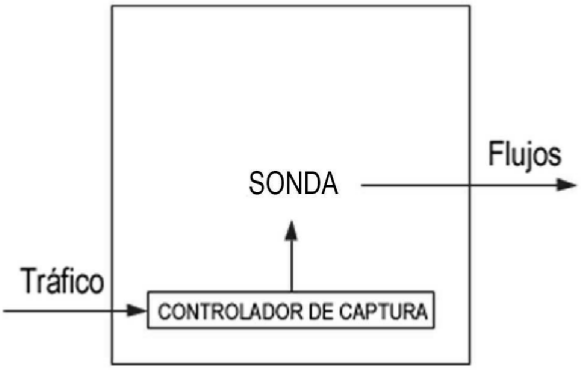

Figura 2

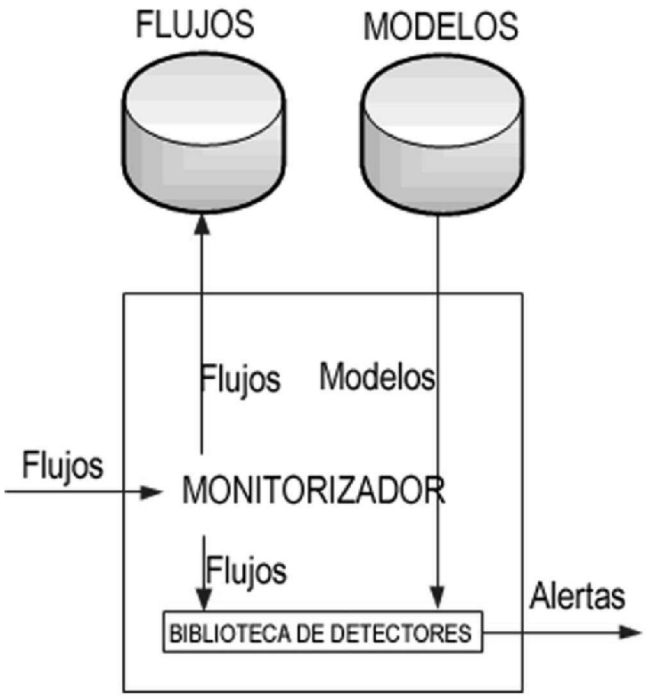

Figura 3

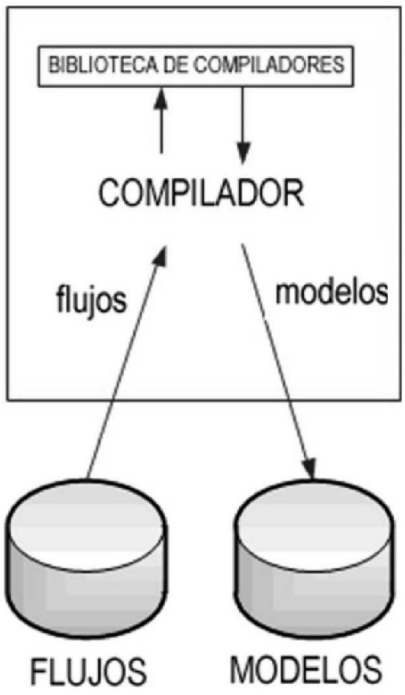

Figura 4

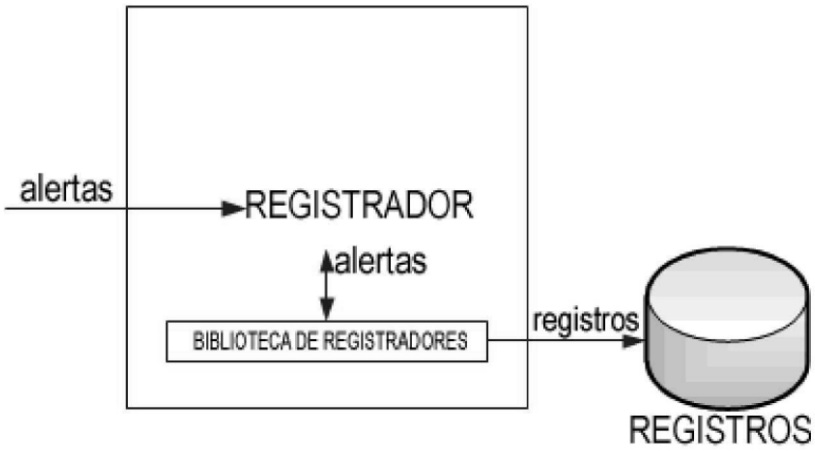

Figura 5

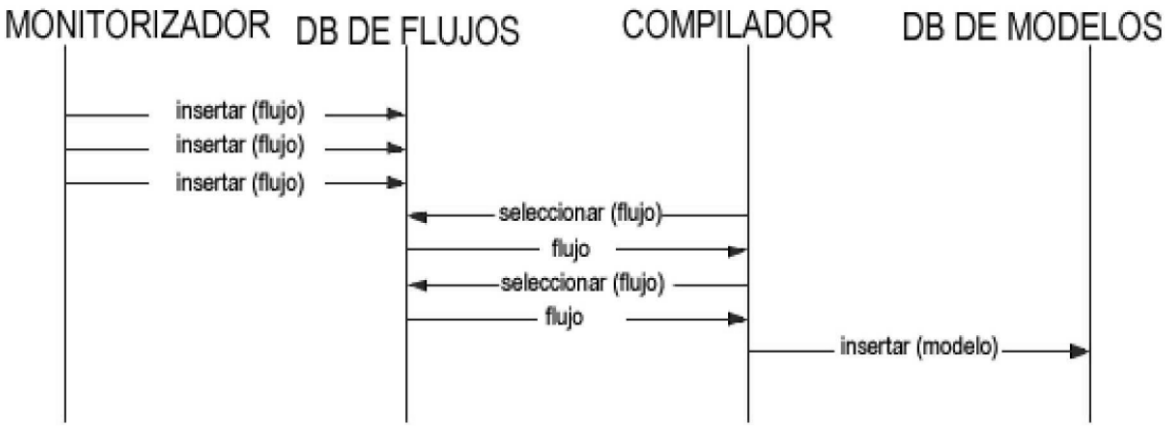

Figura 6

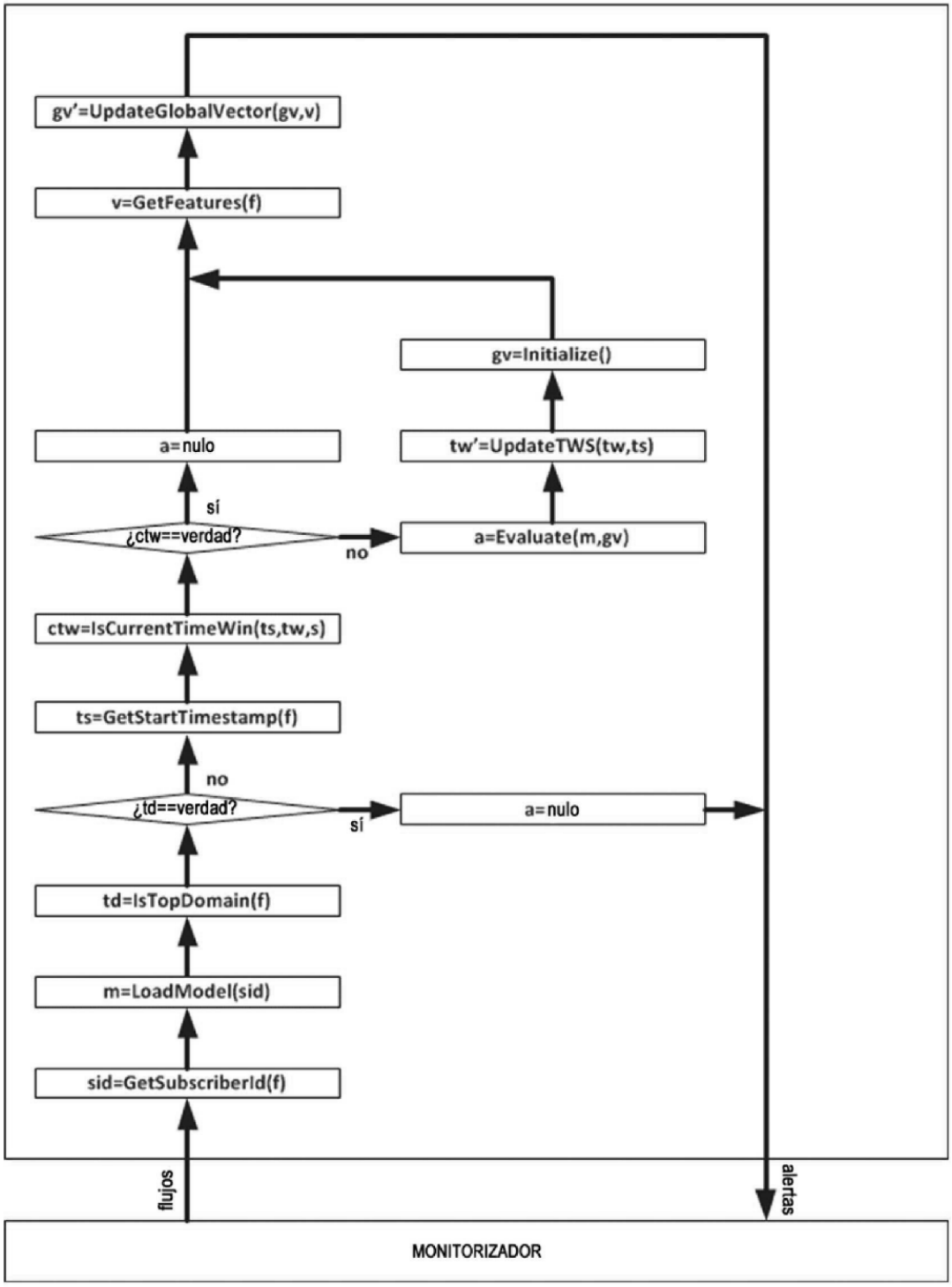
gv'=UpdateGlobalVector(gv,v)
v=GetFeatures(f)
gv=Initialize()
a=nulo
tw'=UpdateTWS(tw,ts)
sí
¿ctw==verdad?
no
a=Evaluate(m,gv)
ctw=IsCurrentTimeWin(ts,tw,s)
ts=GetStartTimestamp(f)
no
¿td==verdad?
sí
a=nulo
td=IsTopDomain(f)
m=LoadModel(sid)
sid=GetSubscriberId(f)
flujos
alertas
MONITORIZADOR

Figura 7

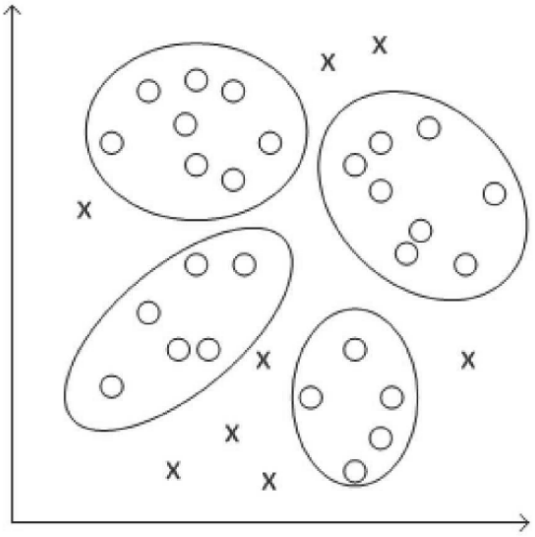

Figura 8

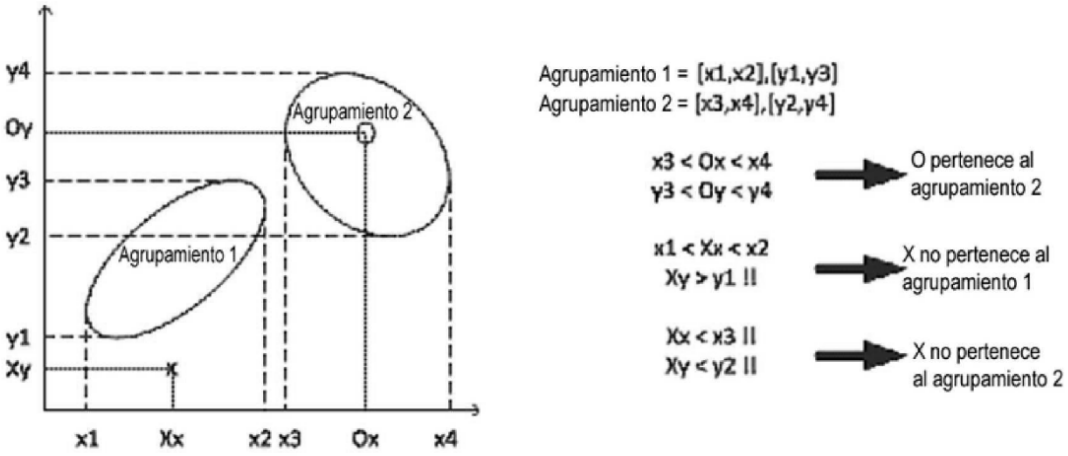

Figura 9

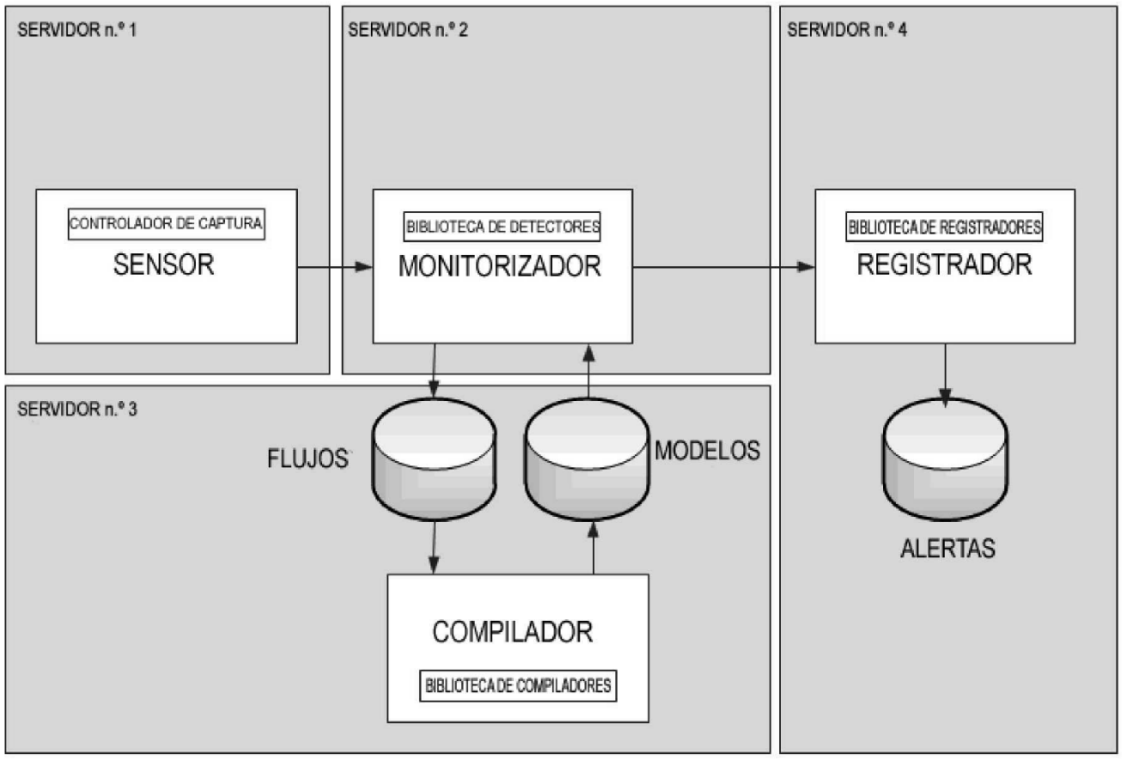
SERVIDOR n.º 1
CONTROLADOR DE CAPTURA
SENSOR
SERVIDOR n.º 2
BIBLIOTECA DE DETECTORES
MONITORIZADOR
SERVIDOR n.º 4
BIBLIOTECA DE REGISTRADORES
REGISTRADOR
SERVIDOR n.º 3
FLUJOS
MODELOS
ALERTAS
COMPILADOR
BIBLIOTECA DE COMPILADORES

Figura 10