RNPS No. 2143
ISSN 2072-6260
Versión Digital

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

REPORTE TÉCNICO
Minería de Datos

SERIE GRIS

Técnicas y algoritmos de minería de datos empleados en sistemas de detección de intrusiones

José Manuel Bande Serrano y
José Hernández Palancar

RT_022 **junio 2014**

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

RNPS No. 2143
ISSN 2072-6260
Versión Digital

REPORTE TÉCNICO
Minería de Datos

SERIE GRIS

Técnicas y algoritmos de minería de datos empleados en sistemas de detección de intrusiones

José Manuel Bande Serrano y
José Hernández Palancar

RT_022 **junio 2014**

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

Tabla de contenido

Lista de figuras

Lista de tablas

Técnicas y algoritmos de minería de datos empleados en sistemas de detección de intrusiones

José Manuel Bande Serrano[1] y José Hernández Palancar[2]

[1] Equipo de Investigaciones de Minería de Datos, Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV), La Habana, Cuba
jbande@cenatav.co.cu
[2] Equipo de Investigaciones de Biometría, Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV), La Habana, Cuba
jpalancar@cenatav.co.cu

RT_022, Serie Gris, CENATAV
Aceptado: 17 de Enero de 2014

Resumen. En el presente trabajo se lleva a cabo una revisión y un análisis del estado del arte en el área de la detección de intrusiones informáticas. Las intrusiones informáticas, también conocidas como ataques informáticos, causan miles de millones de pérdidas económicas anualmente y en no pocos casos ponen el riesgo la seguridad y la paz entre naciones. En ese sentido, desde hace varios años se realizan investigaciones enfocadas en mejorar la eficiencia y eficacia de la detección de intrusiones. El presente artículo se enfoca principalmente en las Técnicas de Minería de Datos empleadas con tal objetivo. Adicionalmente, ofrecemos nuestra visión del problema y consideraciones respecto al camino futuro que se debe seguir en este campo de investigación.
Palabras clave: sistemas de detección de intrusiones de red, minería de datos, detección por anomalías, deteccion por uso inadecuado.

Abstract. In this work we make a review and analysis of the state of the art in Intrusion Detection. Intrusions, also known as informatics attacks cause thousands of millions of monetary looses every year. They also put in danger the national security and the peace between nations. Bearing this in mind, a lot of researching effort has been invested in improving the efficiency and efficacy of the intrusion detection systems. This article is mainly focused in Data Mining techniques applied to intrusion detection problem. In addition we offer our own vision of the problem and our considerations for the future of the intrusion detection.
Keywords: network intrusion detection systems, NIDS, data mining, anomaly-based detection, misuse-based detection .

1. Introducción

Un ataque informático o intrusión se define como una secuencia de eventos relacionados que persiguen interrumpir, denegar, degradar o destruir información y servicios residentes en una computadora o red de computadoras (Gorbani et. al, 2010)[1]. De manera más formal, una intrusión es toda actividad informática que vulnere la *Confidencialidad*, *Integridad*, *Autenticación*, y la *Disponibilidad* de la información. Estos términos constituyen los cuatro pilares de la seguridad informática.

La confidencialidad se refiere a que la información solo puede ser accesible para usuarios autorizados. La Integridad plantea que la información debe estar protegida contra cambios no autorizados. La autenticación indica que los usuarios deben ser quienes dicen ser. Y por último, la Disponibilidad establece

que los recursos y servicios deben estar disponibles para todos los usuarios autorizados. Adicionalmente, cualquier actividad que violente alguna política impuesta por la administración de la red se considerada también ataque informático.

Resulta apropiado destacar el hecho de que el ataque informático es más una secuencia actividades maliciosas relacionadas, que una única acción. Por eso, también se considera como intrusiones a los intentos fallidos de agresión informática y a los ataques no completados (Endorf et. al, 2004) [2]. Por otra parte, los ataques informáticos pueden ser ejecutados por uno o varios actores conocidos en el argot informático como *hackers*. Las motivaciones de estos sujetos u organizaciones son varias: dominación geopolítica, beneficios económicos, liberación de información vergonzosa y comprometedora, y hasta el mero entretenimiento. Los atacantes recurren a diferentes estrategias para lograr sus objetivos. Estas estrategias han sido recogidas en tres grandes grupos (Gorbani et. al, 2010)[1]: Ingeniería Social, Enmascaramiento, Explotación de Vulnerabilidades y Abuso de Funcionalidad.

La Ingeniería Social son acciones de tipo interpersonal que incluyen la manipulación y el engaño. Estas son minuciosamente diseñadas para ganar acceso a privilegios e información protegida. Por ejemplo, el robo de contraseñas o la manipulación de individuos con privilegios para que inserten programas dañinos con o sin su consentimiento. El Enmascaramiento, se produce cuando el atacante finge ser un usuario legítimo con el objetivo de escalar privilegios. Así puede conseguir acceder al sistema mediante contraseñas robadas, u obtener información haciéndose pasar por una autoridad. La Explotación de Vulnerabilidades se produce cuando el atacante se aprovecha de errores y malas prácticas de programación encontradas en programas legítimos. El objetivo más común es ganar acceso no autorizado al sistema mediante el programa comprometido, aunque también se puede agredir al sistema provocando mal funcionamiento del programa. Por último, el Abuso de Funcionalidad sucede cuando el atacante provoca que el sistema ejecute acciones legítimas pero de forma descontrolada, de manera que se excedan las capacidades de procesamiento y de memoria, provocando eventualmente la inutilización del sistema. Por ejemplo, en un ataque de negación de servicio se reciben más solicitudes de conexión que las que el sistema puede procesar, quedando imposibilitado de proveer otros servicios adecuadamente.

La detección de intrusiones es el proceso de analizar datos generados por eventos que ocurren en una red o en computadoras independientes, con propósito de encontrar indicios de agresión informática. Un Sistema de Detección de Intrusiones (IDS) es la automatización de dicho proceso (Scarfone and Mell, 2007) [3]. Los IDS pueden ser una herramienta o un conjunto de estas, que cooperan para llevar a cabo la detección. Estas no son solo de tipo software, sino que en algunos casos también incorporan arquitecturas de hardware dedicado, que en su mayoría son implementadas en dispositivos de hardware reconfigurable. El principal objetivo de estos aceleradores hardware es asistir en tareas que requieren alto desempeño computacional, como es el cotejo de grandes conjuntos de cadenas sobre flujos de datos a elevada velocidad. Algunos trabajos de investigación encontrados en esta área son [4], [5], [6].

El principio fundamental de la detección de intrusiones es que la actividad intrusiva es notablemente diferente de la actividad normal y, por lo tanto, es observable (Gorbani et. al, 2010)[1]. De forma más específica, la detección de intrusiones se apoya en las siguientes asunciones:

i. La información producida por la actividad intrusiva es cualitativamente y/o cuantitativamente, diferentes a la generada por la actividad normal.
ii. La mayor parte del tiempo las instancias de actividad es de tipo normal.
iii. El número de instancias de actividad normal es mayor que el número de instancias actividad intrusivas.

Aunque la mayoría de los IDS asumen estos principios como verdaderos, existen situaciones donde estos no se cumplen del todo, lo cual no indica que dejen de ser útiles, puesto que bajo otro conjunto de

condiciones sí se cumplen. Esto sugiere que aún falta formalización y generalidad en la descripción del problema de la detección de intrusiones. A tales situaciones nos referiremos a lo largo del artículo.

El método inicial de detección de intrusiones se basa en la comparación de patrones distintivos de los ataques con los datos. Junto con la evolución de internet, el tiempo de aparición de nuevos ataques se reduce, la cantidad de datos a analizar para el descubrimiento de los nuevos patrones de ataques se incrementa, y la complejidad de los ataques también aumenta. Esto hace que esta tarea supere las capacidades humanas y obliga a incorporar nuevas técnicas y algoritmos de aprendizaje para descubrir dichos patrones de forma automatizada, proporcionando mayor eficiencia y eficacia en la detección de intrusiones.

La Minería de Datos, la ciencia de descubrir conocimiento oculto en grandes volúmenes de información, se inserta en la detección de intrusos de tres maneras fundamentales. La primera es el descubrimiento de nuevos patrones de ataque a partir de datos pertenecientes al mismo. La segunda es la caracterización del comportamiento normal (no intrusivo), a partir de datos previamente clasificados como tal. Y por último, el descubrimiento de comportamiento anómalo (potencialmente intrusivo) sin emplear datos previamente clasificados. Las dos primeras formas caen en el dominio de la clasificación supervisada mientras que la última cae en el dominio de la clasificación no supervisada. La primera también se utiliza en lo que se denomina *Detección basada en uso inadecuado* (MIDS) del inglés *Misuse-based Intrusion Detecction Systems* dado que el conocimiento que se extrae está relacionado con el ataque. Las dos últimas se utilizan en la *Detección basada en anomalías* (AIDS) del inglés *Anomaly-based Intrusion Deteccion Systems*, dado que lo que se pretende es detectar aquello que se aleje del comportamiento normal, o sea, una anomalía.

El presente trabajo ofrece una revisión del estado actual de la Detección de Intrusiones. En la sección dos se abordan aspectos generales de los IDS. En la sección tres exponemos los trabajos en el estado del arte agrupados por los algoritmos propuestos. La sección cuatro estará dedicada a los sistemas de detección híbridos. Como su nombre lo indica, estos son sistemas que combinan varios métodos de detección. Por último, se ofrecen las conclusiones, donde aparecen nuestros análisis, criterios y proyecciones futuras.

2. Sistemas de detección de intrusiones

Esta sección expone los aspectos generales de los Sistemas de Detección de Intrusiones. Ponemos a consideración una taxonomía basada en otras anteriores y en nuestra propia visión del problema. También abordamos un enfoque emergente para la detección de intrusiones, que aunque algunos autores consideran que pertenece a la categoría de detección de anomalías pensamos que presenta marcadas diferencias que lo convierten en un campo nuevo e independiente. A este enfoque le denominaremos *Detección basada en teoría del daño* y debe su nombre a una nueva teoría sobre el funcionamiento del sistema inmunológico humano. A los demás enfoques de detección, MIDS y AIDS hemos dedicado las correspondientes subsecciones en las que se tratan los aspectos característicos de uno y otro. Adicionalmente hemos añadido una subsección que describe la taxonomía de tipos de intrusiones más representada en la literatura.

2.1. Taxonomía

En la literatura se describen dos tipos genéricos de IDS, según el enfoque usado para la detección. En el primero se cuenta con un conjunto de descriptores de patrones de ataques. Dichos descriptores de patrones se comparan con los datos y, en caso de encontrar coincidencias, se producen alertas. El término empleado para este enfoque es *Detección basada en uso inadecuado* (MIDS). No podemos afirmar, sin embargo, que haya un total acuerdo en cuanto al empleo de este término. Algunos autores denominan a este enfoque

como *Detección basada en firmas de ataques* (Signature-based) (Scarfone and Mell, 2007) [3], otros, *Detección basada en reglas* otros *Detección basada en cotejo de patrones* (Pattern-based) (Endorf et. al, 2004) [2], otros plantean que estos son casos particulares del primero (Gorbani et. al, 2010)[1].

El segundo enfoque, denominado *Detección basada en anomalías* (ANIDS), se basa en la creación de modelos de lo que es considerado comportamiento normal. Luego, estos modelos son comparados con los datos reales y cualquier desviación de los mismos se clasifica como anomalía. Se asume que las intrusiones son inherentemente anómalas y, por tanto, se alerta cada vez que se encuentra una anomalía. Bajo este enfoque se distingues dos categorías: la primera emplea datos previamente clasificados como normales para crear los modelos (clasificación supervisada); y la segunda emplea datos sin clasificar (clasificación no supervisada).

Adicionalmente, existen autores que defienden una tercera categoría denominada, *Stateful protocol análisis* o *Especification based apporaches*(Endorf et. al, 2004) [2], (Gorbani et. al, 2010)[1]. Esta se diferencia en que la base de conocimiento no recoge descriptores de patrones de ataque, sino descriptores de patrones del manejo adecuado de protocolos de internet. En estos, por el contrario, se alerta cuando no se encuentran correspondencias entre ellos y los datos observados. Desde nuestro punto de vista, este modelo sigue el paradigma de comparar un patrón pre-definido con los datos; por ende, consideramos que es una subcategoría o caso particular de la *Detección basada en uso inadecuado*.

En este trabajo se pone a consideración la taxonomía mostrada en la figura 1. La misma está compuesta por cinco dimensiones: alcance, entidad potencialmente atacante, enfoque de detección, modo de respuesta y tipo de arquitectura. El alcance, como su nombre lo indica, define el espacio físico bajo análisis. En esta dimensión se encuentran tres tipos de IDS: los que analizan localmente a una computadora (denominados en ingles *Host-based Intrusión Detection Systems* HIDS); los que analizan el comportamiento de la actividad en la red (excluyendo al comportamiento interno de las computadoras), denominados en inglés *Network-based Intrusion Detection Systems* NIDS; y los que analizan ambos espacios físicos, denominados *híbridos*.

Los HIDS solo pueden monitorear lo que ocurre dentro de la computadora, por eso se dice que son de alcance estrecho o (narrow-scope). Los NIDS, en cambio, pueden monitorear la red entera, por lo que son de alcance amplio, (wide-scope) (Endorf et. al, 2004) [2]. Esto implica que los HIDS son mejores en la detección de ataques internos, o sea, aquellos que se producen desde terminales conectadas a la red interna. Su mayor ventaja es que son capaces de detectar ataques antes que estos ingresen al entorno de la red. Los NIDS, por su parte, son mejores para detectar ataques desde redes externas a la red que protegen. Respecto a la complejidad de cada uno, se plantea que los HIDS son más difíciles de implementar, dada la heterogeneidad de las fuentes de datos con las que tienen que lidiar y además el hecho de que sean dependientes del sistema operativo. Los NIDS generalmente procesan paquetes de red, agregaciones de estos, o reportes basados en distintas métricas de tráfico.

Utilizamos el término *Entidad Potencialmente Atacante* (EPA) para definir a la unidad de información que es objeto de análisis, actualización y seguimiento por parte de los IDS. Ejemplo de EPAs empleados comúnmente por los IDS son los paquetes de red, las conexiones y los procesos en las computadoras. Adoptamos el término EPA para transmitir la idea de que esta puede ser cualquier abstracción de información que es analizada y sobre la cual se emitirá un veredicto respecto a su carácter malicioso o intrusivo. Sin perder generalidad, la detección de intrusiones puede verse como el proceso de analizar y clasificar EPAs como maliciosas o no maliciosas. Un EPA o un conjunto de estos clasificados como intrusivos pueden generar una alerta. Independientemente de que cada IDS defina su propio EPA, consideramos que en el entorno informático existe un conjunto de entidades, objetos o conceptos bien conocidos que pueden ser considerados EPAs. En la figura 2 se propone una jerarquía de dichas entidades.

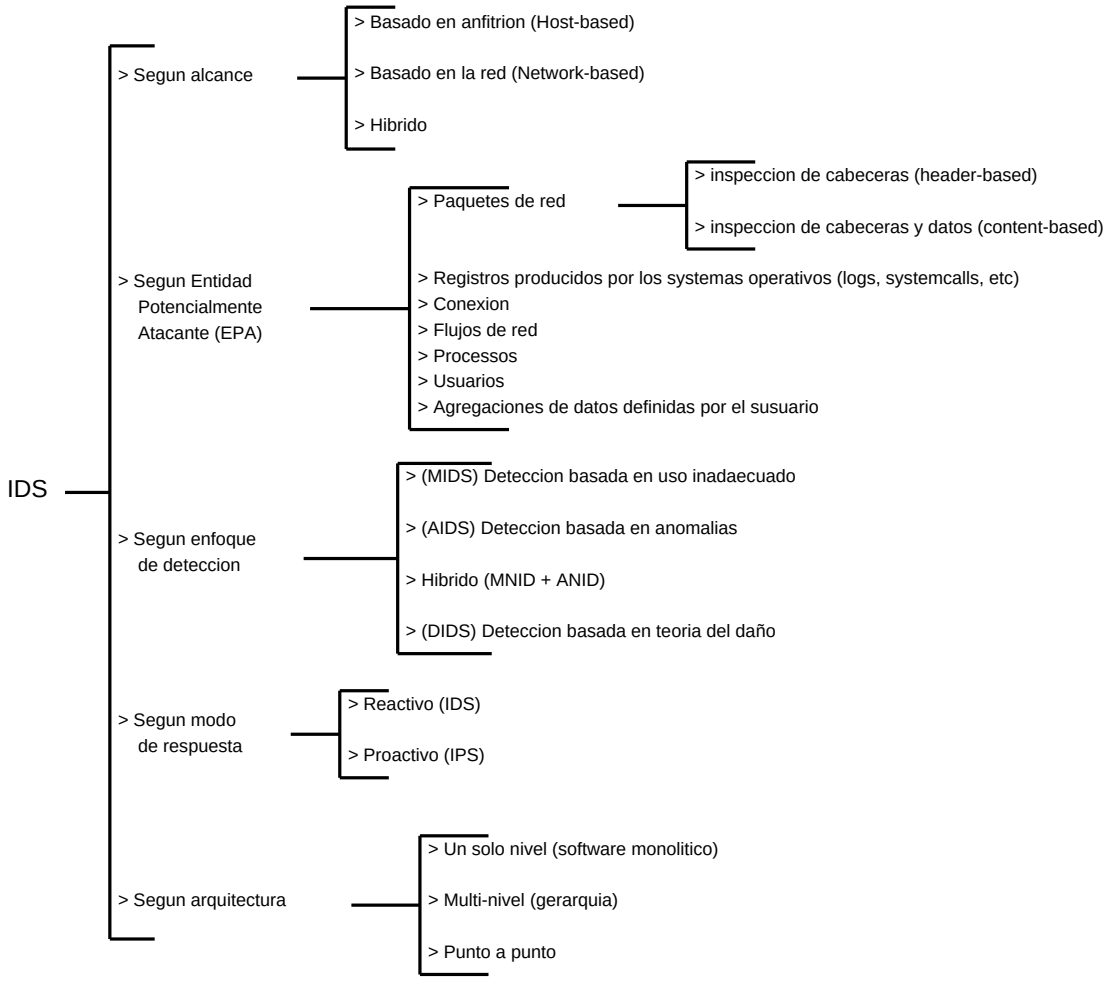

Fig. 1. Taxonomía de los Sistemas de Detección de Intrusiones según cinco dimensiones propuestas: alcance, entidad potencialmente atacante (EPS), enfoque de detección, modo de respuesta, y tipo de arquitectura.

La figura 2 se divide en dos partes, la de la derecha muestra aquellas entidades que habitan en una computadora, y la de la izquierda las que habitan la red. El grado de abstracción y el grado de agregación es mayor en la cima de la jerarquía y disminuye hacia los niveles inferiores. A la derecha se muestran los sistemas operativos actuales multi-usuario y multi-tarea, donde todos los procesos (tareas) están ligados a algún usuario. De forma inversa, cada usuario posee un conjunto de procesos. Lo anterior indica que toda actividad maliciosa es iniciada o llevada a cabo por algún proceso, ya sea de forma directa o indirecta. Por ejemplo, un usuario atacante ejecuta un proceso *shell* para insertar intrusiones maliciosas a otro proceso, lo que constituye un ataque del tipo *Buffer over-flow*. En este caso, el primer proceso es el responsable de iniciar la actividad maliciosa que el segundo proceso ejecutará. Los procesos, por su parte, generan información sobre la actividad que realizan, ya sea de manera directa, a través de los *logs* y llamadas del sistema, o de manera indirecta mediante los múltiples reportes y auditorías que podemos hacer al sistema. Por ejemplo, podemos hacer una lectura a la tabla de procesos y ver el consumo de memoria de cada proceso o en su conjunto.

En el lado izquierdo de la figura 2 corresponde al entorno de la red. En este encontramos al tráfico total generado por la red en la cima de la jerarquía. Le siguen los flujos de datos que bajo algún criterio puedan definirse, como pueden ser: flujos de sub-red, flujos de enlaces entre dispositivos de interconexión o agregaciones definidas por los encargados de la administración. Los flujos agrupan un conjunto de conexiones. Consideraremos como conexión al envío de, al menos, un paquete, desde una dirección IP fuente a una dirección IP destino. En ese sentido, cada conexión se compone de uno o más paquetes. Finalmente, en el nivel más bajo de abstracción y agregación de la jerarquía se encuentran los paquetes.

Los paquetes de red, los llamados *logs*, las llamadas de sistema y los reportes de auditorías al sistema, constituyen unidades de información sobre actividad reciente (Endorf, 2004)[2]. Es decir, que portan información lo más cercana posible al momento actual, sobre los eventos ocurridos en el sistema. Estos registros constituyen, además, las unidades primarias de información, pues son los que se encuentran en estado natural, o sea, sin procesamiento previo por parte de los sistemas de detección de intrusiones. Para referirnos a ellos emplearemos el término general *Registro de Actividad* (RA). Nótese que estas ocupan el menor nivel de abstracción en la jerarquía presentada en la figura 2.

En la dimensión referente al tipo de EPA de la taxonomía propuesta, se establecen varias categorías de IDS, según el tipo de entidad potencialmente atacante que analizan. En la figura 1 tomemos como ejemplo los IDS cuyos EPAs son los paquetes de red. Estos solo serán capaces de clasificar a cada paquete como malicioso o no malicioso, nada pueden decir respecto a la relación entre ellos. Este IDS de ejemplo, sería incapaz de detectar un ataque de negación de servicio, pues puede componerse de un número abrumador de paquetes perfectamente normales.

Consideremos ahora el nivel de abstracción inmediato superior, las conexiones. Un IDS donde los EPAs sean las conexiones necesita dar seguimiento a las conexiones activas para poder clasificarlas como maliciosas o no maliciosas. En otras palabras, necesita mantener objetos en memoria que representen y procesen la información relacionada con cada conexión. Además, por cada nueva unidad de información que arriba al sistema, dichos objetos necesitarían ser actualizados, lo cual indica la existencia de estados. Es por eso que en la literatura se afirma que estos IDS deben poseer la capacidad de mantener estados (Endorf et. al, 2004) [2]. Volviendo al ejemplo, un IDS que analiza conexiones sería capaz de detectar no solo ataques cuyo patrón se haya en la información contenida en los paquetes, sino también en la relación entre estos. Por ejemplo, este IDS podría observar que cierta conexión está teniendo un volumen de paquetes poco usual, lo cual podría constituir un ataque de negación de servicio o un ataque de escaneo de puertos. En la literatura se clasifica a estos IDSs como IDS orientados a paquetes o IDS orientados a conexión[7].

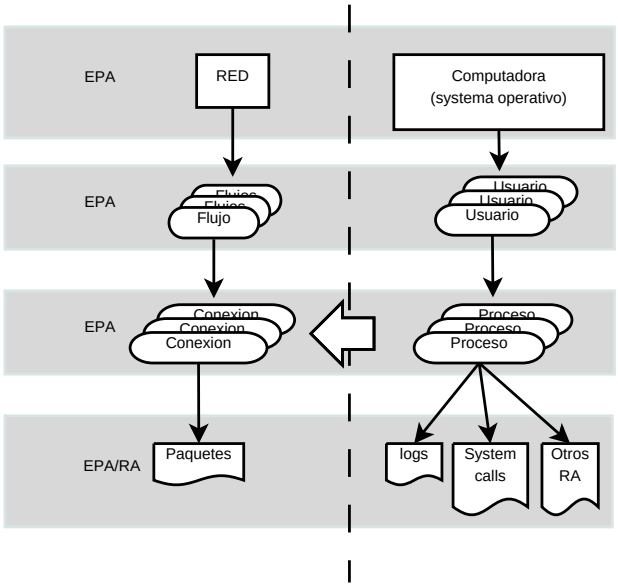

Fig. 2. Jerarquía de entidades o conceptos informáticos que se encuentran en las redes y sistemas informáticos modernos.

Los IDS orientados a paquetes son los más conocidos. En la figura 1 se observa que bajo esta categoría se encuentran otras dos clases de IDS, aquellos que solo chequean las cabeceras de los paquetes, más conocidos como *cortafuegos* (*firewalls*), y los que analizan tanto la cabecera, como la carga de datos. Estos tipos de IDS son los más antiguos, algunos autores emplean el término *Detección basada en firmas de ataques* (signature-based) (Scarfone y Mell, 2007) [3]. El término firma, se usa para describir una secuencia de bytes que es típica de algún ataque. Algunos autores también emplean la denominación *Escaneo basado en contenido* para afirmar que el contenido de datos de los paquetes también es inspeccionado (Endorf et. al, 2004) [2].

En dependencia del nivel de abstracción de los EPAs estos pueden conducir a distintas de agregaciones de datos. Lo anterior nos conduce al concepto de *resolución del análisis*. Cuando el análisis se lleva a cabo sobre EPAs de alto nivel en la jerarquía, el análisis tiene baja resolución. En cambio, el hacerlo sobre EPAs con bajo nivel jerárquico implica un análisis de alta resolución. Vale destacar que ambos tipos de análisis, de baja y de alta resolución, son de igual importancia. Un análisis de baja resolución permite detectar ataques que afectan a grandes cantidades de datos. Por el contrario, los de alta resolución son capaces de detectar intrusiones más sutiles, o sea, que no impliquen cambios significativos en los volúmenes de datos, o que más que presentar patrones cuantitativos característicos, presentan patrones cualitativos característicos, como es el caso de las firmas de ataque en los paquetes de red.

La siguiente dimensión en la taxonomía propuesta está relacionada con el método de detección. En esta área proponemos que se incluyan otras dos categorías, además de las ya existentes MIDS y AIDS. La primera son los enfoques híbridos MIDS-AIDS, y la segunda, es un enfoque emergente, denominado *Detección basada en daño* (DIDS) por sus siglas en inglés. Los sistemas híbridos combinan de alguna forma la detección basada en anomalías y la basada en uso inadecuado. En la sección 2.6 profundizamos sobre los sistemas híbridos.

Respecto a los DIDS, aunque algunos autores los incluyen en la categoría de detección de anomalías, consideramos que existe un aspecto que los distancia considerablemente de estos. Los AIDS detectan anomalías pero no pueden asegurar que estas sean intrusiones o falsos positivos, porque su principio de funcionamiento presupone que toda anomalía es una intrusión, lo cual no es siempre cierto. Los DIDS

pueden distinguir, al menos en principio, entre anomalías maliciosas y no maliciosas. Esto es así porque se basan en recientes descubrimientos sobre el funcionamiento del sistema inmunológico humano, específicamente en la *teoría del daño* (Aickelin, 2003)[8], la cual permite censar el mal funcionamiento o afectación al sistema. La idea de imitar el sistema inmunológico para la detección de intrusiones no es nueva. Desde hace algún tiempo existen los Sistemas Auto Inmunes (AIS), basados en teorías antiguas sobre el sistema inmunológico como la *Selección Negativa*. Los AIS basados en esta teoría no fueron muy exitosos, principalmente por las limitaciones de la teoría subyacente (Aickeling y Greensmith, 2007) [9]. En la subsección 2.4 se abordarán estas teorías con más profundidad.

La siguiente dimensión de la taxonomía propuesta se basa en la forma en que los IDS responden ante un ataque. Esta separa a los IDS en dos grandes grupos: (1) aquellos que solo alertan sobre la presencia total o parcial de un ataque; (2) los IPS, (del inglés *Intrusion Prevention Systems*) que, además de alertar, toman acciones para frenar o disminuir el ataque (Scarfone y Mell, 2007) [3]. Dichas acciones se basan en reglas pre-establecidas, que indican qué hacer en caso de haber detectado uno u otro incidente (Endorf et. al, 2004) [2].

La arquitectura física es la última dimensión en la taxonomía. Un IDS puede estar conformado por un único software, en cuyo caso se denomina arquitectura *mono-nivel*. Los HIDS, por ejemplo, son un ejemplo típico de arquitectura mono-nivel. La arquitectura también puede ser jerárquica-distribuida, llamada arquitectura *multi-nivel* (Endorf et. al, 2004) [2]. En esta existen agentes que ejecutan tareas específicas y que cooperan entre sí, y organizados de forma piramidal realizan diferentes tareas en cada nivel. Por ejemplo, la capa más baja la conforman los recolectores de datos; le siguen los agentes que limpian los datos y conforman agregaciones o abstracciones con mayor grado de información. Estos, a su vez, alimentan al siguiente nivel, constituido por evaluadores o analizadores, que son los que ejecutan el análisis sobre los datos y alertan sobre los ataques. En la cima se encuentra el módulo de gestión que recibe y procesa alertas, e interactúa con los administradores.

El último tipo de arquitectura se denomina *Punto-a-punto* (Endorf et. al, 2004) [2]. Esta arquitectura es distribuida pero no jerárquica, o sea, todos los agentes que la integran son capaces de ejecutar las mismas funcionalidades. Dichos elementos intercambian información entre ellos, que emplean para modificar su comportamiento, elevar el grado de protección, centrarse en protegerse de algún ataque en específico o difundir el conocimiento adquirido. Por ejemplo, suponga dos cortafuegos ubicados en dos puntos distintos de acceso a la red, el primer cortafuego detecta un ataque del tipo A, y manda un mensaje al segundo cortafuego que lo alerta sobre dicho ataque. El segundo cortafuegos puede configurarse para reaccionar de forma anticipada al ataque A. De forma similar puede actuar el segundo cortafuegos, cuando detecte otro tipo de ataque.

2.2. Detección de intrusiones basada en uso inadecuado

Los MNIDS comparan patrones de ataques conocidos contra los datos y, en caso de coincidir, generan alertas. Dichos patrones pueden describirse en forma de reglas. Las reglas son proposiciones lógicas condicionales en las que el antecedente es una combinación lógica de consultas realizadas a distintos atributos o características de los registros de actividad (RA). Mientras que el precedente, es una proposición simple, que indica el ataque relacionado. Tomemos como ejemplo el caso en que los RA son los paquetes de red. Por ejemplo, la proposición lógica 1 es una regla que afirma la existencia de una intrusión denom-

inada “Ataque1” cuando el campo IP fuente IPsrc del paquete contiene el valor “190.2.21.14”, el campo puerto de destino PortDest, contiene el valor 80, y la carga de datos del paquete contiene la cadena “/root”.

$$IPSrc(RA, 190.2.21.14) \wedge PortDest(RA, 80) \wedge PayloadContain(RA, \text{“}/root\text{”}) \Rightarrow AttackType(RA, Ataque1). \quad (1)$$

Ahora bien, como se puede observar en la regla anterior, toda la información necesaria para afirmar que el ataque está teniendo lugar se encuentra en un único paquete. En muchas ocasiones no basta con la información aportada por un solo RA para afirmar que cierto ataque está ocurriendo. Endorf et. al (2004) [2] afirma que los descriptores de patrones (reglas) pueden ser atómicos, cuando solo requieren un solo registro de actividad para activarse, o compuestos, cuando requieren varios registros.

Generalmente, la mayoría de los ataques constan de varios eventos relacionados en el tiempo. En la figura 3 se muestra el *grafo de un ataque*, donde cada nodo representa un evento o paso del ataque. Un grafo de ataque representa todas las posibles secuencias de vulnerabilidades que un atacante puede explotar durante una intrusión de múltiples pasos (Wang y Jajodia, 2008) [10]. Por ejemplo, un primer paso puede ser la identificación de los puertos abiertos en el objetivo. Luego de tener esta información, el atacante puede intentar conectarse mediante el protocolo *telnet* a uno de estos puertos para conocer el tipo de sistema operativo y la versión del objetivo, y así paso a paso hasta completar el ataque.

Es importante destacar que este grafo muestra la dependencia entre eventos que conforman el ataque, aunque no necesariamente la secuencia en que puedan ocurrir. El motivo es que eventos independientes entre sí no tienen por qué ser ejecutados con un orden fijo, como por ejemplo los eventos E1 y E2 que se muestran en la figura. Esto no sucede con los eventos dependientes de otros, que sí deben respetar un orden. En ese sentido, el grafo de la figura 3 se comporta como un *autómata finito no determinista* (NFA) por sus siglas en inglés. Los estados de este autómata se activan en la medida que se detectan los eventos correspondientes. El nodo terminal es el estado que se alcanza cuando todos los pasos del ataque se han cumplido. El hecho de que este autómata sea no determinista implica que un estado puede activar varios próximos estados, o sea, puede existir la concurrencia de eventos.

Los EPAs de jerarquía superiora a los RA son objetos con tiempo de vida, durante el cual relacionan información bajo un concepto. Por ejemplo, las conexiones relacionan a todos los paquetes intercambiados por los mismos extremos durante el tiempo de duración de la misma. Si definimos un conjunto de estados admisibles para los EPAs, la detección basada en uso inadecuado se resume a verificar cuando algún EPA ha alcanzado el estado terminal de algún grafo de ataque. Bajo este enfoque, las reglas simples ocupan el lugar de funciones de transición entre estados, dado que estas evalúan las condiciones bajo las cuales un evento ocurre, o no, y las reglas compuestas relacionan reglas simples en forma de grafo de ataque, el cual funciona como autómata finito no determinista.

Hasta hora se ha hablado del proceso de evaluación de reglas (fase de evaluación) pero nada se ha dicho del proceso de obtención de las mismas (fase de aprendizaje o entrenamiento). En principio, las reglas pueden obtenerse al analizar los datos relacionados con algún tipo de ataque, pero en la medida que aumenta la complejidad de los ataques y el volumen de datos a analizar, esta tarea se hace extremadamente engorrosa para los especialistas en seguridad. Desarrollar un escenario de intrusiones no es una tarea sencilla y requiere de elevada experiencia y conocimiento por parte de los especialistas. A ello se suma que encontrar relaciones entre reglas es una tarea difícil y no siempre se puede validar una regla con un ciento de certeza (Gorbani et. al, 2010) [1]. Para atacar estos problemas se ha recurrido a la Minería de Datos.

El volumen de información que es necesario procesar para encontrar nuevas reglas es elevado, supera las capacidades humanas para cumplir esta tarea de forma eficiente. Los algoritmos de Minería de Datos pueden sustituir a los humanos en la caracterización de nuevos ataques, pues brindan la posibilidad de

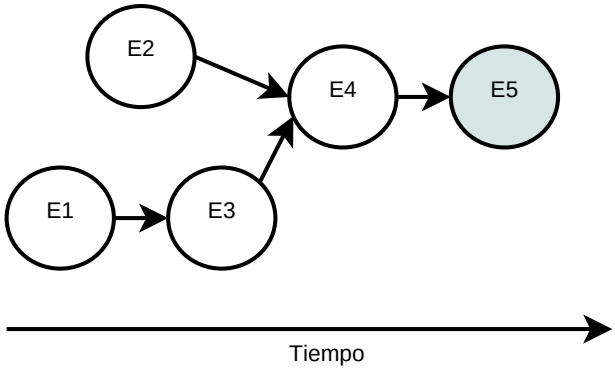

Fig. 3. Eventos relacionados que de conjunto conforman un ataque o intrusión informática. El evento más oscuro denota el evento final.

encontrar relaciones ocultas en grandes volúmenes de datos. Este es un problema de *Clasificación Supervisada*, donde a partir de un conjunto de datos pertenecientes a un ataque desconocido, se entrenan clasificadores que posteriormente serán capaces de reconocer dicho ataque.

Además de las reglas, existen otros tipos de clasificadores para la detección de intrusiones. En la sección sobre algoritmos veremos que también se emplean redes neuronales, máquinas de soporte vectorial, algoritmos de agrupamiento, entre otros. En ese sentido, aquellos clasificadores que constituyen cajas negras, o sea, cuyo conocimiento adquirido no es humanamente interpretable, constituyen una desventaja, dada su imposibilidad de mostrar el *modus operandi* de los ataques a los supervisores.

El hecho de requerir datos previamente clasificados puede constituir una desventaja, pues en ciertos entornos es muy difícil obtener datos con la calidad y en la cantidad requeridas. Adicionalmente, puede ocurrir que un atacante corrompa los datos de entrenamiento para que, una vez entrenado el clasificador, los ataques pasen inadvertidos. Actualmente existen instituciones dedicadas exclusivamente a la tarea de obtener nuevas reglas, como por ejemplo el proyecto de IDS *Snort* [11]. Este IDS de código abierto evalúa tanto las cabeceras como el contenido de datos de los paquetes de red.

Las investigaciones realizadas en la fase de evaluación de los MID se han enfocado en mejorar la eficiencia. El motivo es el aumento exponencial de la velocidad de los flujos de datos, y la proliferación de nuevos ataques, lo cual demanda capacidades de procesamiento superiores a las que pueden aportar tecnologías de procesamiento convencionales, como los procesadores de propósito general (GPP). Dado que cada vez se requiere evaluar mayor número de reglas en menor tiempo, los procesadores secuenciales se ven abrumados en ciertos entornos, como la detección de intrusos en las redes. Esta situación ha impulsado desde hace ya más de veinte años el desarrollo de propuestas que incorporen alto grado de paralelismo. En esa dirección se han realizado numerosos trabajos que emplean tecnologías como las GPU, multi-procesadores y los FPGAs, siendo estos últimos los que mejores resultados han proporcionado.

Los MIDs tienen como principal desventaja el hecho de que solo pueden detectar aquellos ataques recogidos en la base de reglas. Para detectar nuevas intrusiones, dicha base debe ser actualizada con las reglas correspondientes. Esto hace que los MIDs sean vulnerables ante ataques desconocidos. Ya sea obteniendo reglas de terceros, o implementado los propios mecanismos de descubrimiento automático de reglas, los MIDs dependen del análisis *aposteriori* de los ataques. En la siguiente sección discutiremos sobre el segundo enfoque para la detección de intrusiones, el cual permite descubrir ataques sin conocer de antemano su patrón de comportamiento.

2.3. Detección de intrusiones basada en anomalías

La detección basada en anomalías, del inglés *Anomaly-based Intrusion Detection*, se sustenta sobre la base de que la actividad intrusiva es cualitativa y cuantitativamente diferente de la actividad normal. En la detección basada en anomalías, se elaboran modelos de comportamiento normal a partir de datos que se consideran libres de intrusiones. Esta constituye la fase de entrenamiento. Luego, en la fase de análisis, los modelos se comparan con la información adquirida y cualquier desviación se asume como una intrusión (Gorbani et. al, 2010)[1]. Para la detección de anomalías se emplean métodos de clasificación supervisada y métodos de clasificación no supervisada. Para la clasificación supervisada, se realiza un entrenamiento con datos clasificados previamente como normales (libres de ataques). El enfoque no supervisado, en cambio, no requiere de datos previamente clasificados para crear sus modelos.

Los AIDS pueden clasificarse en dinámicos y estáticos (Endorf, 2004) [2]. Los AIDS dinámicos son aquellos que actualizan sus modelos al tiempo que ejecutan el análisis y, por ende, tienen una fase de entrenamiento constante. Como ejemplo podemos contar con (Olmeadow et al., 2004) [12] con su agrupamiento dinámico. Los AIDS estáticos, en cambio, tienen un tiempo de entrenamiento bien definido, donde se crean los modelos, que no se modifican durante el período de análisis. Un ejemplo es (Eskin et al., 2002) [13].

Los IDS encontrados en la literatura se distinguen, además, por el tipo de anomalía en que se enfocan. En ese sentido, hacemos una distinción entre anomalías temporales y anomalías espaciales. Las primeras son las desviaciones de un objeto respecto a un modelo que describe su comportamiento histórico. Las anomalías espaciales, por su parte, son desviaciones del comportamiento de un objeto con respecto al comportamiento de la mayoría de los objetos.

La detección de anomalías temporales como mecanismo de detección de intrusiones se basa en el principio (ii), donde se asume que la mayor parte del tiempo corresponde a condiciones libres de ataques. Será entonces imprescindible mantener un registro histórico de la evolución del comportamiento de las EPAs. Estos registros históricos también se conocen como series de tiempo. Los IDS que encuentran anomalías en series de tiempo emplean modelos estadísticos de pronóstico y detección de cambio [14] [15] [16] [17]. El cálculo del error de estimación y su comparación con umbrales preestablecidos es el método de decisión más empleado. También se utilizan técnicas propias del procesamiento de señales, como la transformada *wavelet*, que permite el análisis multiresolución para encontrar anomalías en distintos niveles de agregación de los datos [18]. La técnica de Análisis de Componentes Principales (PCA) es también utilizada como método de reducción de dimensionalidad y para mejorar la precisión del análisis [17]. Con el objetivo de correlacionar comportamientos históricos de varios EPAs, se emplea el análisis multivariado, útil para encontrar relaciones de dependencia entre procesos estocásticos.

Para el descubrimiento de anomalías espaciales se define un espacio n-dimensional poblado con EPAs, donde cada dimensión es un aspecto o atributo[13] [12]. De esta forma, cada punto de dicho espacio corresponde a un comportamiento específico. Ese espacio también se conoce como *espacio de características*. Sobre el espacio de características se aplican algoritmos de agrupamiento, creando grupos donde se maximiza la similitud entre sus miembros. A los objetos que quedan fuera de todo grupo se les denomina valores atípicos, (*outliers*, en inglés). Posteriormente, estos grupos y/o *outliers* son clasificados en anómalos o no.

El criterio empleado para clasificar los grupos se basa en el principio (iii), según el cual aquellos grupos mayormente poblados son clasificados como normales, mientras que aquellos pobremente poblados y los *outliers* son clasificados como anómalos. Una de las condiciones que viola este principio ocurre cuando el número de EPAs intrusivas supera al número de EPAs normales. Ello sucede en ataques masivos de negación de servicio, donde un número descomunal de conexiones son generadas, con el objetivo de

reducir la disponibilidad de los servicios. En estos casos los grupos de mayor número de EPAs pueden no corresponder a grupos normales.

La gran ventaja de los AIDs es que no requieren conocimiento previo respecto al patrón de las intrusiones; más bien se basan en aprender el comportamiento normal y alertar sobre el comportamiento extraño. Por eso, son capaces de detectar ataques totalmente desconocidos. Por otra parte, como desventaja se encuentra el número de falsos positivos que pueden generar, principalmente durante el tiempo en que no se ha aprendido lo suficiente sobre el comportamiento normal. Vale destacar que una intrusión es un comportamiento anómalo, pero un comportamiento anómalo no necesariamente es una intrusión. Otra desventaja es que estos son sensibles a ataques dirigidos a alterar los datos que se emplean en el entrenamiento.

Otro aspecto a considerar en los ANIDs lo constituye la interpretabilidad de las anomalías. La interpretabilidad se refiere a la capacidad que el sistema puede tener o no, de proveer información entendible por los humanos sobre una anomalía que haya sido detectada. Un sistema puede, simplemente, alertar cuando se ha encontrado una anomalía y dejar el escrutinio de datos anómalos a los especialistas humanos. Como capacidad adicional, el sistema puede automatizar dicho proceso de escrutinio de datos generando información de mayor nivel que informe a los especialistas cuáles son las características de la anomalía detectada. Claro está que la automatización de dicho proceso no es otra cosa que la aplicación de minería de datos para la obtención de reglas que describan dicha anomalía, es decir, un sistema híbrido AIDS-MIDS. Esto se corresponde con la mecánica natural del aprendizaje, donde el nuevo conocimiento es percibido, caracterizado y aplicado de forma más eficiente.

Otro campo activo de investigación se enfoca a determinar qué características en los datos son más sensibles al comportamiento anómalo y, por tanto, más relevantes en el proceso de detección. En ese sentido, en correspondencia con el principio (i), encontramos en la literatura las anomalías clasificadas como cuantitativas, también denominadas *anomalías de volumen* [15], y las cualitativas. En el caso de las redes, las anomalías de volumen son aquellas que ocurren sobre medidas de volumen de tráfico, como la cantidad de conexiones, cantidad de bytes transferidos, entre otras. Por otro lado, las anomalías cualitativas toman en cuenta la presencia de cierta información contenida en los paquetes de red, como los IPs, los puertos, interrelaciones y características distribucionales de los mismos. Algunos trabajos revisados en este reporte proponen, como parte del proceso de detección, algoritmos para extraer automáticamente las características más relevantes para elaborar los modelos de detección.

2.4. Deteccion de intrusiones basada en teoría del daño

La detección de intrusiones basada en teoría del daño se inspira en los mecanismos del sistema inmunológico humano, que permiten rechazar y eliminar entidades biológicas dañinas, incluso cuando estas son totalmente desconocidas por el organismo (Aickelin, 2003) [8]. Esto convierte al sistema inmunológico en un paradigma para la detección de intrusiones. Las entidades biológicas que ingresan al cuerpo humano se denominan antígenos. La eliminación de antígenos dañinos es responsabilidad de células denominadas *Linfocitos-B*. Dichas células generan anticuerpos que se adhieren a los antígenos, los inhabilitan y, eventualmente, los destruyen. Esto es posible porque los anticuerpos generados están dotados de secuencias proteicas, que se unen a las proteínas y conforman el cuerpo del antígeno. Este este es un proceso de cotejo de patrones, entre las proteínas de los anticuerpos y la de los antígenos.

Una de las cuestiones más difíciles de responder sobre el sistema inmunológico es su capacidad para distinguir entre entidades biológicas que forman parte del organismo y entidades biológicas que no lo forman. En ese sentido, se han propuesto dos teorías: la selección negativa y, más recientemente, la teoría del daño (Aickeling, 2007)[9].

La selección negativa afirma que los linfocitos-B tienen una etapa de maduración, donde son expuestos a las células del organismo, a la vez que producen anticuerpos (secuencias proteicas) de manera aleatoria. Aquellos que atacan las células del organismo son eliminados y los que no, ingresan al flujo sanguíneo. Esto responde a la pregunta de por qué el sistema inmunológico no agrede al propio organismo. Sin embargo, se ha demostrado que en realidad, el sistema inmunológico no ataca a cualquier entidad biológica externa, sino que reacciona únicamente ante aquellos que causan daño.

La teoría del daño aporta una solución a esta última cuestión. Así, sugiere que el sistema inmune es activado cuando recibe señales moleculares que indican daño o estrés sobre el tejido de células (Aickeling, 2007)[9]. En otras palabras, el daño ocurrido sobre las células del cuerpo es censado mediante la generación de señales supresoras y estimuladoras de la respuesta inmunológica. Cuando se detecta daño, se emiten señales (moléculas) que activan la respuesta de los Linfocitos-B que se encuentran en el área cercana, estimulando así la generación de anticuerpos.

Uno de los algoritmos basados en la teoría del daño, se centra en la existencia de células que, cual forenses, reconocen cuándo otras células han muerto por muerte natural (*apoptosis*), o cuándo han muerto de manera inesperada (*necrosis*). Estas células son denominadas *Células Dendríticas* (Greensmith, 2005) [19]. Cuando las células dendríticas detectan una necrosis, recolectan información (fragmentos proteicos), que luego transfieren a los Linfocitos-B. Estos por su parte, emplean la información recolectada para generar los anticuerpos apropiados para atacar al patógeno en cuestión. Adicionalmente, las células dendríticas emiten señales de daño que estimulan a los Linfocitos-B, que se encuentran a su alrededor, provocando inflamación y elevación de la temperatura, lo que propicia la concurrencia de otros linfocitos-B al área dañada o infectada.

Tanto la selección negativa como las células dendríticas, han sido traducidas a algoritmos informáticos. Haciendo analogía entre EPAs y entidades biológicas, los DIDS ofrecen grandes potencialidades en la detección de intrusiones, dada su capacidad de decidir entre lo que es dañino para el sistema y lo que no, y asociarlo a algún agente patógeno.

2.5. Tipos de intrusiones

A grandes rasgos, los ataques informáticos se pueden agrupar, según el papel que desempeñan, en: los que proveen información sobre el sistema víctima del ataque; los que permiten obtener el control sobre el sistema; y los que provocan daños o afectaciones al sistema. La taxonomía más difundida en la literatura reconoce a la adquisición de información como ataques del tipo *probing*. La obtención de control sobre el sistema se divide en dos ramas: una que provee acceso al sistema como usuario local, R2L del inglés *Remote to Local*, y otra que provee acceso de usuario local a super usuario, U2R del inglés *User to Root*. Los ataques generadores de daño pueden ser: por abuso de funcionalidad, también conocido como Negación de Servicio, DoS, del inglés *Deny of Service*, y aquellos que provocan mal funcionamiento del sistema o aplicaciones, como los programas malignos o virus.

- **Probing**: Probing, cuya traducción del inglés significa tanteo o exploración, es un ataque llevado a cabo generalmente como un paso previo a otros tipos de ataques, cuya intención es obtener conocimiento sobre su objetivo. Antes de penetrar una red, los atacantes necesitan recolectar información sobre la misma, conocer las direcciones IPs, conocer la topología de la red, los servicios que presta, los puertos que permanecen escuchando, identificar vulnerabilidades en los sistemas, entre otros datos. En este tipo de ataque es muy común realizar escaneo de IPs y de Puertos, por lo que se distingue por un aumento en el número de paquetes con destino a diferentes localizaciones en la red, que generalmente provienen de un mismo sitio. Con el objetivo de no ser detectados, los atacantes tratan de disminuir la

intensidad con que se lleva a cabo el escaneo, incrementando los períodos entre los paquetes que son enviados, o utilizando varias fuentes.

- **Dos/DDos**: Negación de Servicio por sus siglas en inglés, es un tipo de ataque que se basa en saturar a las computadoras que brindan un servicio determinado, dígase un sitio web o a la red misma, con un volumen de tráfico abrumador, de forma que el servicio o la red queden totalmente inhabilitados. Este tiene dos formas: (1) distribuido, cuando un número elevado de computadoras externas se encuentran involucradas en el ataque, y (2) no distribuido, cuando se trata de una única computadora atacante. Un ejemplo de ataque de negación de servicio es la Inundación de Paquetes SYN, donde se envían paquetes con las direcciones IP falsificadas provocando que las computadoras de la red generen paquetes TCP de inicio de conexión que nunca reciben respuesta. Similarmente, existen inundación de paquetes ICMP y UDP.
- **U2R**: Acceso de usuario a super-usuario, por sus siglas en inglés. Para tomar el control total de los recursos de una computadora u otro dispositivo es necesario ser super-usuario. Este tipo de ataque tiene ese objetivo. Una vez que los atacantes se han convertido en usuarios normales del sistema, llevan a cabo una serie de procedimientos que le dan privilegios de super-usuarios. Para ello emplean diferentes técnicas, como la ingeniería social, robo de contraseñas, desencriptación de información, explotan vulnerabilidades en servicios que dan privilegios administrativos, entre otras.
- **R2L**: Acceso remoto a usuario local, (*remote to local, R2L*), un atacante en una computadora externa a la red, gana acceso de super-usuario en una o varias computadoras conectadas a la red interna. Generalmente se lleva a cabo enviando una serie de paquetes a un servicio determinado y explotando alguna vulnerabilidad, que permita tener acceso local a la computadora remoto. Por ejemplo, una de las vulnerabilidades reportadas en el servicio SendMail es que si se envía un correo con el carácter pipe en ciertos campos, SendMail se ve forzado a ejecutar algunos comandos en la computadora remota.
- **Amenaza persistente avanzada**: Los mencionados aquí son los ataques más difíciles de detectar, pues su estrategia se basa en ir tomando el control de los recursos informáticos de una institución determinada de forma paulatina y desapercibida. Para ello no recurren solo a procedimientos informáticos, sino también a interacciones humanas, como la ingeniería social y el espionaje. Se le denomina persistentes, dado que su período de establecimiento puede durar entre uno y cinco años. Este tipo de ataque pasa por una serie de estadios de maduración bien definidos, donde entre otros se encuentran: la adquisición de información y el emplazamiento de Software, que abran puertas traseras para obtener control privilegiado en los sistemas. Este ataque puede combinar los ataques aquí citados. Los ataques más famosos de este tipo se han llevado a cabo entre gobiernos de naciones adversarias, y con motivo de espionaje industrial.

3. Algoritmos

A continuación presentamos un conjunto de los algoritmos encontrados en la literatura. Estos han sido agrupados en familias, de acuerdo con las bases teóricas los soportan. Las familias más representadas son las que emplean técnicas estadísticas, el agrupamiento y los algoritmos biológicamente inspirados. Adicionalmente, mostramos algoritmos y métodos que se han empleado no solo en tareas de detección, sino también en la solución de otros problemas dentro de la propia detección de intrusos, como la reducción de dimensionalidad y la interpretación de anomalías. Vale destacar que muchos de los trabajos revisados aplican combinaciones de algoritmos para obtener resultados superiores. Por eso, no será extraño encontrar trabajos citados en distintas secciones del texto, dada que cada sección trata de una familia de algoritmos en específico.

3.1. Enfoque estadístico

Desde la perspectiva de los IDS, el comportamiento de un EPA está definido por un conjunto de variables que denominamos atributos o características. Un flujo IP, por ejemplo, puede caracterizarse por el volumen de datos transferidos, por la distribución entre IPs origen y destino, por la distribución de paquetes por protocolo y por cualquier otra métrica diseñada por el usuario. Los valores adquiridos por dichos atributos pueden considerarse estados de procesos estocásticos. Esto significa que el próximo estado del proceso, o sea, el valor futuro del atributo, depende tanto de componentes predecibles, como de componentes impredecibles. Para analizar dichos procesos contamos con la ciencia de la probabilidad y la estadística.

El registro de los valores de una variable aleatoria en el tiempo se conoce como serie temporal. De las series temporales se desea conocer cuándo cierta variable está desarrollando un comportamiento poco usual, dado que en principio, este puede significar una intrusión. En la literatura se encuentran los términos detección de cambio o detección de comportamiento aberrante, para significar lo antes mencionado. Los algoritmos diseñados con tal objetivo se basan en crear un modelo con los datos observados previamente, que permita generar una predicción del valor futuro de la variable. Cuando esta predicción no concuerda dentro de umbrales predefinidos, con el valor real observado se indica una desviación, cambio inesperado o anomalía. A estos algoritmos se les denominan algoritmos de pronóstico o predicción sobre series de tiempo (Brockwell, 1996)[20]. En consecuencia, al modelo elaborado se le denomina predictor.

En las series temporales se asume que los valores observados no son independientes, su dispersión varía con el tiempo, y estos son frecuentemente gobernados por una tendencia central y por componentes cíclicas [21]. Un modelo estadístico, es un conjunto de funciones que se ajustan a la realidad observada para cada una de estos componentes. Existen múltiples modelos para describir a las series temporales, de ellos, el modelo aditivo es uno de los más empleados.

Sea una serie temporal cuyos valores, $y_1, y_2 \ldots, y_n$ son los resultados de la variable aleatoria Y_t, el modelo aditivo se define mediante la ecuación 2:

$$Y_t = T_t + Z_t + S_t + R_t. \tag{2}$$

En la ecuación anterior, T_t es una función monótona de t que recoge el crecimiento o decrecimiento del sistema, y se denomina tendencia. La componente Z_t es una influencia cíclica no aleatoria de largo término, mientras que S_t es una influencia cíclica aleatoria, pero de período corto. Por último, R_t es una variable aleatoria que recoge todas las desviaciones del modelo no estocástico. De todas estas componentes, solo R_t es estocástica (aleatoria) reflejando aquello que no es posible predecir del sistema, mientras que las demás representan esa interrelación oculta que existe en los datos, que desconocemos, pero que es determinista [21].

El análisis de series temporales se puede realizar en dos dominios diferentes, aunque complementarios: el dominio del tiempo y el dominio de la frecuencia. En el dominio de tiempo se desea encontrar el predictor más eficaz, con el objetivo de reducir la probabilidad de que dada una anomalía, esta sea un falso positivo. Por su parte, el análisis en la frecuencia permite descomponer la serie de tiempo en componentes cíclicas con frecuencia, amplitud y fase fijas. El análisis de las mismas puede ser útil para detectar cuándo el proceso se aleja de sus comportamientos cíclicos habituales y, por lo tanto, tiene un comportamiento anómalo.

Jake D. Brutlag [14] propone detectar el comportamiento aberrante en series de tiempo sobre medidas de tráfico, en una red de distribución de servicios televisivos por la web. Aunque en esencia no se trata de detección de intrusos, su objetivo persigue detectar mal funcionamiento y cuellos de botella en el sistema, lo cual puede ser perfectamente el producto de acciones intrusivas. Para ello, emplea el método

de pronóstico Holts-Winter [20], que es la extensión del predictor de Suavizado Exponencial [20] al caso de series temporales con componentes cíclicas y de tendencia.

El suavizado exponencial es uno de los métodos más simples de suavizado que puede emplearse, además, como predictor en series de tiempo, bajo ciertas condiciones (que la serie no tenga componentes cíclicas, ni tendencia variable). La predicción es el resultado de un promedio de pesado, sobre todas las observaciones anteriores. Dicho promedio pesado es tal que el grado de influencia dada a cada observación anterior decrece exponencialmente, desde la observación más reciente hasta la más antigua.

El método Holts-Winter es un predictor más complejo y eficaz, dado que emplea el suavizado exponencial, tomando en cuenta las componentes cíclicas y de tendencia de la serie temporal. Desde el punto de vista computacional, estos son métodos numéricos e incrementales, lo que los dota de una elevada eficiencia. Para detectar la anomalía, se define una ventana de tiempo y un rango de variabilidad en cada observación. El autor afirma que asumir como anomalía cada observación que queda fuera del rango predicho produce una elevada cantidad de falsos positivos. Como mejora, se propone un umbral dentro de dicha ventana de tiempo que establece el número máximo de observaciones fuera de rango que deben ocurrir para que sea considerado anomalía.

Paul Brandfor *et. al.* [16] emplea la Transformada Discreta de Wavelet, (DWT) por sus siglas en inglés, para llevar a cabo un análisis multi-resolución de las series de tiempo. La transformada wavelet permite obtener múltiples representaciones de una misma señal (serie de tiempo) en el dominio del tiempo y en distintas bandas de frecuencia. Dichas representaciones son denominadas coeficientes wavelet, y son el resultado de un proceso reiterativo de filtrado de la señal original a las altas y a las bajas frecuencias. La ventaja de este procedimiento radica en que los comportamientos anómalos de larga duración son más visibles en los coeficientes wavelets de baja frecuencia, mientras que en los de alta frecuencia, las anomalías de corta duración se perciben mejor.

Los autores utilizan tres coeficientes wavelet, a partir de la serie de tiempo de medidas de tráfico. Estos consisten en bajas, medias y altas frecuencias. Las señales a las bajas frecuencias capturan las variaciones de varios días; las medias frecuencias, las variaciones diarias; y las altas, aquellas fluctuaciones con horas de duración. Los autores proponen el algoritmo denominado *Deviation Score* (DS), para detectar las anomalías. En este, para cada tiempo t, y sobre cada coeficiente wavelet se analizan los datos en una ventana deslizante de tamaño igual a la duración de la anomalía que se quiere detectar. La variabilidad obtenida de cada coeficiente en la ventana es combinada y se compara con un umbral. Los autores realizan una comparación entre su método y el método Holts-Winter. De 39 anomalías, DS logró reportar 38, mientras que Holts-Winter reportó 37. Sin embargo, los autores afirman que no hicieron comparaciones en cuanto a la cantidad de falsos positivos, y reconocen que Holts-Winter es más sensible a anomalías potenciales (que pueden ser falsos positivos o no) que DS.

La gran dimensionalidad de los datos constituye una limitante para muchos algoritmos. En ese sentido, se han encontrado dos estrategias para reducir tal dimensionalidad. La primera ya la hemos mencionado, consiste en definir EPAs con mayor capacidad de agregación, en detrimento de la resolución del análisis. La otra, se basa en el empleo de técnicas estadísticas para la reducción de dimensionalidad. Krishnamurthy et al., (2003) [17] propone el uso de *Sketches*. Un Sketch es una estructura de datos que representa un resumen probabilístico basado en proyecciones aleatorias de un conjunto de variables aleatorias [22]. Esta estructura es más eficiente desde el punto de vista espacial (memoria) y temporal (procesamiento), a la vez que conserva las características probabilísticas de las variables originales. Un Sketch S, es, básicamente, una matriz de registros $T[i][j]$ donde cada fila lleva asociada una función de dispersión (*Hash*), h_i. Sea una secuencia de elementos, del tipo (llave, valor), para cada columna i se calcula: $T[i][h_i(llave)] += valor$. Esta operación se realiza en cada intervalo de tiempo I_t y para los n elementos (llave, valor) obtenidos en dicho intervalo.

En [17] se proponen tres módulos: a) módulo de generación de Sketches, b) módulo de predicción y c) módulo de detección. En el módulo de predicción se evalúan distintos métodos de predicción sobre los Sketches $S_o(t)$ observados por el módulo de generación. Los métodos empleados son: *Moving Average (MA)*, *S-shaped Moving Average (SMA)*, *Suavizado Exponencial Pesado*, *Non-Seasonal Holt-Winter (NSHW)* estos son métodos que se basan en suavizado. Adicionalmente se emplea un método denominado *Box-jenkins*, o también *Auto Regresive Integrated Moving Average*, (ARIMA). Este método captura la dependencia lineal de los valores recientes con los más antiguos. A continuación se muestra una tabla resumen de los métodos de suavizado empleados y luego una breve explicación del método ARIMA. El lector interesado en estos métodos puede remitirse a [21].

Tabla 1. Algunos métodos de pronostico en series temporales basados en suavizado.

Método	**Ecuaciones**
MA	$S_f(t) = \frac{\sum_{i=1}^{W} S_f(t-1)}{W}, W \geq 1$
SMA	$S_f(t) = \frac{\sum_{i=1}^{W} S_f(t-1)}{\sum_{i=1}^{W} w_i}, W \geq 1$
EWMA	$S_f(t) = \begin{cases} \alpha S_o(t-1) + (1-\alpha) S_f(t-1), & t > 2 \\ S_o(1), & t = 2 \end{cases}$
NSHW	$S_s(t) = \begin{cases} \alpha S_o(t-1) + (1-\alpha) S_f(t-1), & t > 2 \\ S_o(1), & t = 2 \end{cases}$ $S_t(t) = \begin{cases} \beta(S_s(t) - S_t(t-1)) + (1-\beta) S_s(t-1), & t > 2 \\ S_o(2) - S_o(1), & t = 2 \end{cases}$ $S_f(t) = S_s(t) + S_t(t)$

En la tabla, $S_f(t)$ es el valor predicho de la serie; $S_s(t)$ es el valor predicho para la componente cíclica; $S_t(t)$ es el valor predicho para la tendencia; y $S_o(t-1)$, es el valor observado en la lectura anterior. Los parámetros α y β son las constantes de suavizado, que generalmente se definen entre cero y uno.

El modelo Auto Regresivo (AR) y el Modelo de Promedio Desplazable (MA), por sus siglas en inglés, son aplicables a series estacionarias, es decir, series que no muestran tendencia, pero que fluctúan alrededor de un valor constante C denominado nivel, y que no poseen cíclicas. La ecuacion 3 define el modelo (AR) :

$$z_t = C + \Phi_1 z_{t-1} + \Phi_2 z_{t-2}, ... + \Phi_p z_{t-p} + a_t. \tag{3}$$

En esta ecuación, a_t representa una variable aleatoria independiente que recoge el error aleatorio en la predicción. Para esta variable se puede demostrar que dada una serie estacionaria 4:

$$z_t = a_t - \Theta_1 a_{t-1} - \Theta_2 a_{t-2}, ... - \Theta_q a_{t-q}. \tag{4}$$

El modelo ARMA, combina AR y AM, para series estacionarias definiendo un (p,q). Por ejemplo, para $p = 4$ y $q = 2$ se define como 5:

$$z_t = C + \Phi_1 z_{t-1} + \Phi_2 z_{t-2}, .. + \Phi_4 z_{t-4} + a_t - \Theta_1 a_{t-1} - \Theta_2 a_{t-2}. \tag{5}$$

En todas las ecuaciones Θ y Φ son coeficientes. Para el caso de MA se asume una progresión exponencial, tal que $\Phi = -\Theta^i$. El modelo ARIMA extiende ARMA para el caso de series con tendencias y componentes cíclicas, mediante la diferenciación. Se puede demostrar que mediante la diferenciación es posible eliminar las componentes cíclicas y de tendencia sin perder generalidad. Básicamente, sea Y_t una serie temporal, con tendencia variable en el tiempo y componente cíclica, su serie diferenciada con m pasos, se obtiene mediante la fórmula $d_t = (1-B)^m Y_t$, donde cada $B^i Y_t = Y_{t-i}$. El modelo ARIMA se

obtiene definiendo (p,q,m) y sustituyendo d_t por z_t en (5). Para más detalles sobre los modelos AR, AM, ARAM, ARIMA y otros modelos de pronostico en series temporales ver [23].

Es apropiado destacar la desventaja que implica la definición de parámetros *apriori*, como por ejemplo, los α, β, p, q, m, etc. Muchos de estos modelos han sido diseñados para análisis en ciencias económicas y sociales, donde se asume la constante intervención de un humano para la calibración de dichos parámetros. En nuestro contexto, tratamos minimizar tal intervención, por lo que contamos la predefinición de parámetros como un aspecto desventajoso para cualquier modelo de detección de anomalías.

El Análisis de Componentes Principales, (PCA, por sus siglas en inglés), es una técnica estadística exploratoria de datos. Sea un espacio n-dimensional donde se ubican los datos, esta técnica permite encontrar las coordenadas de máxima variabilidad de los mismos [24]. Estas coordenadas, conocidas como componentes principales, son combinaciones lineales de las dimensiones originales. Dado que por definición son ortogonales entre sí, conforman un nuevo espacio. La primera componente principal está orientada hacia la máxima variabilidad de los datos, la segunda hacia la máxima variabilidad de los datos no recogidos por la primera y así sucesivamente. Generalmente, la dimensionalidad del nuevo espacio obtenido, es menor que la del espacio original, dado que aquellas dimensiones que aportan poca información son eliminadas o no tenidas en cuenta. Varios trabajos emplean PCA como etapa de preprocesamiento de datos con el objetivo de que los algoritmos de detección de anomalías procesen los datos que más significativos y así ganar en eficiencia y eficacia.

Tal es el caso de Lakhina et. al, (2004) [15] donde se define una *matriz de trafico*, en la que cada columna pertenece a un enlace de la red, y cada fila corresponde con las medidas de volumen de tráfico hechas en un intervalo de tiempo. Sobre esta matriz se aplica PCA, tomando cada columna como una dimensión del espacio. Como resultado de la aplicación de PCA, se obtienen las componentes principales que capturan la máxima variabilidad en los datos del tráfico. De acuerdo con nuestra terminología, estas componentes constituyen los EPAs que son analizados en busca de anomalías. En las componentes de máxima variabilidad se observan patrones periódicos que son producto del comportamiento normal de la red. En las componentes de menor variabilidad, son más perceptibles las anomalías, puesto que estas se perciben como muestras muy alejadas de la tendencia central. Los autores dividen el conjunto de componentes en normal y anómalas, construyendo así un subespacio anómalo y uno normal.

El *Filtrado Kalman* es especialmente eficaz para separar las señales del ruido. Este tipo de filtrado emplea métodos estadísticos para construir modelos probabilísticos, que luego son utilizados para pronosticar valores futuros de señales aleatorias [25]. Soule et al., (2005) [18] emplea filtrado kalman en una primera etapa para predecir una matriz de tráfico basándose en la matriz actual. En la segunda etapa, la matriz predicha es comparada con la matriz real (que contiene las mediciones más recientes) y la diferencia (residuo) es inspeccionada en busca de anomalías. Con tal propósito, en este trabajo se evalúan cuatro métodos para el análisis de residuales, basados en diferentes patrones de cambio. El primer método compara el tráfico instantáneo residual con un umbral. El segundo método emplea *Deviation Score* sobre la señal residual. El tercero, aplica *wavelet* sobre la señal residual. Por último, el cuarto método emplea un Test de Razón de Probabilidad General. Este test evalúa los desplazamientos en la media que ocurren en la señal residual producto de las anomalías. Todos estos métodos tienen como factor común que requieren de umbrales para la toma de decisiones. Los autores ajustan dichos umbrales empleando las curvas ROC. Las curvas ROC son útiles para medir la precisión de los métodos de detección de anomalías, puesto que ellas representan la relación entre los falsos positivos y los falsos negativos.

Los IPs fuente y destino, los puertos, los protocolos y otros atributos de los paquetes involucrados en la conexión caracterizan a los paquetes. Según (Lakhina et al., 2005) [26], ejecutar técnicas de aprendizaje sobre estos atributos permite lograr mejores resultados que al hacerlo sobre métricas simples, como el volumen de tráfico. Esto se evidencia, primeramente, en que se facilita la detección de aquellas anomalías

que son difíciles de aislar del tráfico normal. Segundo, distribuciones inusuales de estas características revelan información valiosa respecto a la estructura de las anomalías, que no se puede obtener en los enfoques basados en volumen de tráfico. En tal sentido, el citado autor propone tener en cuenta el análisis distribucional de estas características. Para ello, por cada característica observada se crea una matriz similar a [15], salvo que en este caso, cada elemento contiene la medida de entropía por cada flujo de datos en un tiempo *t*. Las anomalías se detectan mediante técnicas de análisis *multi-variado*, con el cual es posible descubrir anomalías en combinaciones de características.

En Casas et. al, (2011) [27] [28] se conjugan la detección de anomalías temporales con detección de anomalías espaciales. El sistema propuesto se denomina UNADA (del inglés Unsupervised Network Anomaly Detection Algorithm). Este se divide en dos etapas, en la primera los flujos de tráfico son analizados con algoritmos de detección de cambio en series temporales, y aquellos que muestran anomalías son pasados al siguiente nivel. En la segunda etapa, con los tráficos anómalos se puebla un espacio, que luego se subdivide en múltiples espacios de menor dimensionalidad y sobre los cuales se realiza agrupamiento. En la sección dedicada a algoritmos de agrupamientos se ofrecen más detalles sobre esta segunda fase.

El muestreo es otra técnica estadística empleada para la reducción de dimensionalidad. El mismo consiste en tomar una muestra de una población lo suficientemente reducida como para hacer viable su análisis, y lo suficientemente significativa como para que refleje el comportamiento de la población en general. (Bakhoum, 2011) [29] demuestra que es posible mantener elevados los niveles de seguridad aún cuando en lugar de realizar una inspección exhaustiva de los paquetes de red se lleva a cabo una inspección selectiva. Para ello, se calcula la probabilidad de que la red esté siendo atacada, y la inspección se hace más o menos exhaustiva, en dependencia del resultado de este análisis. En cuanto a la selección en condiciones de alta probabilidad de ataque, se demuestra que para flujos IP de 1 a 4 paquetes la selección exhaustiva es necesaria para detectar el ataque; sin embargo, para flujos grandes, la frecuencia requerida es menor.

3.2. Algoritmos de agrupamiento

El agrupamiento es el proceso de separar en grupos objetos con información asociada, de forma tal que la similitud entre los objetos en un mismo grupo se maximice, mientras que la similitud entre objetos de grupos diferentes se minimice. Dado que los objetos pertenecientes a un grupo son muy parecidos entre sí, se asume que pertenecen a una misma clase. Para el análisis empleando agrupamiento, los objetos son descritos por un conjunto de atributos o características. Es común definir un espacio donde cada dimensión corresponde a una característica, y encontrar la similitud, o disimilitud, entre los objetos en función de la cercanía entre ellos en dicho espacio.

Para los grupos encontrados en un espacio poblado por EPAs las observación (iii) sugiere que los grupos más poblados, contienen muestras de actividad normal, mientras que los que posean pocas EPAs, incluyendo los *objetos atípicos*, es decir EPAs fuera de todo grupo, pueden ser clasificados como anómalos. En el enfoque de clasificación basada en tamaño del grupo, existen dos fases: (1) Generación de los grupos de comparación (entrenamiento), (2) Detección de EPAs anómalos (análisis o detección). En la primera etapa, los EPAs son agrupados según su similitud y los grupos resultantes son clasificados como anómalos o normales de acuerdo con la observación (iii). En la fase de detección, cada nueva instancia se clasifica, según el grupos más cercano, en anómala o normal.

Algunos trabajos que aplican el agrupamiento para la detección de anomalías en IDS son: Portnoy (2001) [30] y Eleazar et. al. (2002) [13], que emplean *k-menoids*; Oldmeadow et. al. (2004) [12]; Om y Kumdu (2012) [31] y Zanero et. al. (2004) [32], quienes usan *k-means*. Estos trabajos emplean algoritmos de agrupamiento basados en tendencias centrales, cuya principal desventaja es que los grupos que se

obtienen son de formas circulares, siendo poco efectivos para el descubrimiento de agrupaciones con formas más heterogéneas, como por ejemplo, los grupos alargados.

Otra clase de métodos de agrupamiento son los basados en mayas y los basados en densidad. Los algoritmos basados en mayas discretizan el espacio en celdas que en su conjunto conforman una maya o matriz. Es posible tener varios estratos de mayas con distintas resoluciones definidas sobre un mismo espacio. Cada maya puede definir celdas que resuman la información de un conjunto de celdas de la maya inmediata inferior. Esta forma de trabajo evita tener que analizar por separado cada objeto del espacio, por cada etapa en el crecimiento de los grupos, lo cual reduce considerablemente los tiempos de procesamiento. Es por eso que estos algoritmos son favorables para datos altamente dimensionales y de gran cantidad, como es el caso en los IDS.

Los métodos basados en densidad crecen los grupos tomando en cuenta el número de objetos en un área cercana al objeto analizado. De esta forma son buenos encontrando grupos de formas heterogéneas. Leung y Leckie (2005) [33] proponen un algoritmo denominado *fpMAFIA*, que es una extensión de *pMAFIA*, que a su vez es una variación optimizada de *CLIQUE*. Este último, es un algoritmo que combina mayas con cálculo de densidad. Su propuesta incluye el *Descubrimiento de Conjuntos de Elementos Frecuentes*, para lo cual emplean el método de *Crecimiento de Patrones Frecuentes, FP-grow*. Brahmi et. al. (2011) [34] emplean en la etapa de detección de anomalías de su sistema el algoritmo *AD-Clust*, el cual es una combinación de *k-means* y *DBSCAN*. Este último es un algoritmo clásico de agrupamiento basado en densidad.

En la sección dedicada a las series de tiempo abordamos el trabajo de Casas et al. (2011) [27] [28], el cual en una primera etapa detecta anomalías temporales mediante el empleo de algoritmos genéricos de detección de cambio en series temporales y, en una segunda etapa, realiza agrupamiento para detectar anomalías espaciales. La técnica empleada se denomina *Agrupamiento con Acumulación de Evidencia*. Esta consiste en descomponer el espacio de características, que es altamente dimensional, en muchos subespacios de baja dimensionalidad para analizarlos de forma independiente. El resultado de dicho análisis ofrece una solución parcial del problema, que es denominada evidencia. Posteriormente, la evidencia de todos los subespacios es combinada con algún algoritmo de correlación para ofrecer el veredicto final del análisis. Esta técnica reduce la dimensionalidad, a la vez que permite trabajar cada subespacio de forma paralela, y mejorar así la eficiencia. El algoritmo de agrupamiento empleado en cada sub-espacio es *DBSCAN* y la técnica para combinar las evidencias es la *correlación basada en distancia*. Específicamente emplean el algoritmo de Acumulación de Evidencia *AE4RO* que extrae los objetos atípicos intersubespaciales. La anomalía se alerta cuando la evidencia acumulada supera un umbral de disimilitud pre-establecido.

Zhao y Zhou (2013) [35] introducen una versión mejorada del algoritmo de agrupamiento *LegClust*. Este algoritmo es capaz de detectar grupos de cualquier forma, lo cual se logra tomando en cuenta la estructura local de las instancias datos y sus vecinos más cercanos, en conjunción con la aplicación de la *Entropía Cuadrática de Rengi*. Dicha entropía es una generalización del concepto de entropía para variables aleatorias y continuas. El algoritmo determina el proceso de mezclado de grupos de acuerdo a la relación de proximidad entre los datos en la matriz de disimilitud, la que a su vez es calculada empleando la Entropía de Rengi. La mejora al algoritmo se introduce adicionando distancia euclidiana en el proceso de mezclado de grupos, lo cual evita que grupos anómalos sean absorbidos por grupos normales. Esto es muy útil cuando los ataques son conformados por múltiples instancias (EPAs).

Las características del tráfico de red no son estacionarias, o sea, varían en el tiempo. Teniendo esto en cuenta, Oldmeadow et. al. (2004) [12] propone *agrupamiento dinámico*. Es decir, permitir que los grupos sean modificados en tiempo de detección, lo cual implica reagrupar el espacio por cada nuevo objeto insertado. De esta forma, los grupos cambiarán su forma y posición adecuándose más a las condiciones actuales del tráfico de la red. Los autores destacan el hecho de que una nueva conexión que se adicione a un

grupo densamente poblado, tiene menos influencia en la modificación del grupo que un objeto adicionado a un grupo menos poblado. Para controlar este fenómeno, a los nuevos objetos se les asigna un grado de influencia, en dependencia de la densidad del grupo más cercano. El algoritmo de agrupamiento que se emplea es *k-means*, donde el punto medio es recalculado cada vez que se adiciona un nuevo miembro al grupo.

El agrupamiento no solo ha sido empleado como método para la detección, también se ha recurrido a él para reducir la dimensionalidad de los datos. A esta estrategia le denominamos *Pre-agrupamiento* de los datos. Zanero et. al. (2004) [32] afirma que las técnicas de reducción de dimensionalidad como PCA no son eficientes para comprimir los datos, dado que tiende a la pérdida de información. Por otro lado, muchos NIDS ignoran la información contenida en la carga de datos de los paquetes por motivos de desempeño, procesando solo las cabeceras. Esto no es saludable, dado que muchos ataques solo se pueden detectar observando la carga de datos. Por ello, en la primera etapa de la arquitectura, los autores proponen clasificar empleando agrupamiento sobre carga de datos de los paquetes. Lo anterior reduce toda la carga de datos del paquete a una etiqueta, la cual será la misma para paquetes con cargas de datos similares, y diferente para paquetes con cargas de datos disimilares. Otros trabajos que emplean pre-agrupamiento son: Horng et. al. (2011) [36] y Chandrashekar and Raghuveer (2012) [37].

3.3. Máquina de soporte vectorial

La *Máquina de Soporte Vectorial* (SVM), por sus siglas en inglés, es una técnica de clasificación supervisada que, dado un espacio n-dimensional y un conjunto de objetos localizados en dicho espacio, encuentra un hiperplano frontera entre objetos que pertenecen a clases diferentes. La fase de entrenamiento de este algoritmo es el cálculo de dicho hiperplano. Esta se lleva a cabo identificando lo que se denomina como *Vectores Soporte*. Estos son objetos representativos localizados en las fronteras entre clases. Los vectores soporte sirven para calibrar lo que se denomina *función de kernel*. Dicha función representa el hiperplano que divide las clases. En la fase de prueba, por cada nuevo objeto, se calcula de qué parte del hiperplano se encuentra y se le asigna la clase según corresponda. Esto reduce la etapa de detección a un simple cálculo numérico, y convierte a SVM en uno de los métodos de clasificación más eficientes.

El principal reto que enfrenta SVM en la detección de intrusos es la gran dimensionalidad y volumen de los datos. Aunque inicialmente fue concebida como técnica supervisada, también existen versiones que extienden su uso a no supervisado. A continuación presentamos ambos casos del uso de SVM tanto en MNIDs como en ANIDs. Es conocido que SVN no escala favorablemente cuando la entrada es considerablemente grande y los datos son de elevada dimensionalidad. Con el objetivo de hacer SVN viable para la detección de intrusos, muchos trabajos incorporan alguna etapa de preprocesamiento para reducir la dimensionalidad y el volumen de los datos.

Horng et. al. (2011) [36]: emplea Pre-agrupamiento mediante el algoritmo jerárquico *BIRCH*, de manera que este actúa como filtro para limpiar y reducir el volumen de datos. La estrategia es proveer al algoritmo de SVN con menos datos, pero con mayor calidad. En consecuencia, se reduce el tiempo de entrenamiento y mejora el desempeño del clasificador resultante. BIRCH es un algoritmo de grupo jerárquico que genera un árbol, donde cada nodo representa un grupo, las hojas del árbol son aquellos grupos de menor tamaño o jerarquía, y un grupo de mayor jerarquía contiene a sus grupos hijos. La ventaja de BIRCH consiste en que no se guarda cada objeto en cada grupo (nodo), sino que se guarda información sumarizada de los objetos que lo componen, como por ejemplo el centroide y el radio de dicho grupo. La combinación entre la etapa de agrupamiento y SVM en este trabajo surge cuando las hojas del árbol (grupos de menor jerarquía) son reconocidas como instancias abstractas de datos y son pasadas a SVN como datos de entrenamiento.

Chandrashekar and Raghuveer (2012) [37] combinan *Conjuntos Difuso*, *Redes Neuronales* y SVM. Primeramente, se generan K grupos, empleando el algoritmo de grupos difuso *FC-mean*. Luego, para cada grupo se entrena una Red Neuronal con los datos contenidos en el mismo, obteniéndose K redes neuronales. Posteriormente durante el entrenamiento, las redes neuronales serán alimentadas con los datos y cada una generará un valor. El conjunto de valores a la salida de las redes neuronales conforman los atributos del nuevo objeto, con el cual se provee al algoritmo SVM. A cada nuevo objeto se le incorpora una función de pertenencia para reducir el error de clasificación. Por lo tanto, los objetos iniciales son reducidos a objetos con $K+1$, atributos con los cuales se entrena el algoritmo SVM.

Zhang and Jia. (2013) [38] propone como fase de preprocesamiento de datos el empleo de algoritmos basados en *colonia de hormigas*, con el cual se desea seleccionar las mejores características para la clasificación. Este es un algoritmo de inteligencia de enjambre que incorpora retroalimentación en la construcción de las soluciones y por ende presenta un carácter menos aleatorio que los algoritmos genéticos. Luego aplica SVM sobre los atributos que han sido seleccionados por la etapa anterior.

Las Máquinas de Soporte Vectorial, son esencialmente algoritmos de clasificación supervisada. Sin embargo, las *Máquinas de Soporte Vectorial de Clase Única* (OC-SVM), del inglés *one-class SVM*, constituyen la extensión de las ideas de SVM a la clasificación no supervisada. La principal idea detrás de OC-SVN es encontrar un conjunto de vectores (contorno) que mejor encierre a todos los demás vectores en un espacio n-dimensional. Su utilidad en la detección de intrusiones subyace en que, dado la observación (iii), es de esperar que este contorno encierre a las instancias de datos normales del tráfico, mientras que los vectores que permanezcan afuera se representen instancias de actividad anómala.

Eleazar et. al. (2002)[13] experimenta con OC-SVM de kernel gaussiano, para delimitar áreas densamente pobladas de aquellas que no lo son, a la vez que reconoce a los objetos de las áreas pobremente pobladas como anomalías. Uno de los enfoques utilizados en OC-SVM consiste en enmarcar los puntos del espacio de características dentro de una hiper-esfera, considerando todos los puntos fuera de la esfera como anómalos. Sin embargo, Laskov et al. (2004) [39] plantea que, cuando se trata de detección de intrusos, el espacio de características no contiene variables negativas, por lo que es más apropiado emplear un cuarto de esfera centrado en las coordenadas cero de dicho espacio, dado que no se necesita enmarcar puntos negativos.

3.4. Generación de reglas mediante árbol de decisión

Un árbol de decisión es una estructura de conocimiento que se obtiene a partir de datos pre-clasificados. Cada nodo de dicho árbol representa un atributo, y cada rama descendiente es una asociación entre un rango de valores del atributo y una clase o grupo de clases. Las hojas del árbol o nodos terminales representan las clases o conjuntos de clases.

Una vez que el algoritmo termina de generar el árbol a partir de los datos de entrenamiento, cada regla queda registradas como una trayectoria desde el nodo raíz hasta un nodo terminal, o nodo de clase. O sea, cada camino desde el nodo raíz hasta un nodo terminal representa una regla que permite clasificar la clase con la que el nodo terminal se halla etiquetado. Uno de los algoritmos más populares para la inducción de árboles de decisión es *C4.5* Rajeswari y Arputharaj, (2008) [40] que propone una versión mejorada de *C4.5* basado en reglas para la detección de intrusos. Para el entrenamiento y evaluación del algoritmo se emplean KDDCUP99.

3.5. K-vecinos cercanos

El algoritmo K-Vecinos Cercanos (K-NN, del inglés *K-Nearest Neighbor*) es un algoritmo de clasificación supervisada. Durante el proceso de entrenamiento los objetos etiquetados con sus respectivas clases son ubicados en un espacio n-dimensional. Durante la fase de prueba, cada nuevo objeto de clase desconocida es ubicado en el espacio n-dimensional y clasificado según la clase más representada por los k objetos más cercanos. La cantidad de objetos cercanos a tener en cuenta se define previamente por el usuario.

KNN es uno de los algoritmos utilizados por Eleazar *et. al.* (2002) [13] para detectar anomalías. Tomando en cuenta la definición (iii), es de esperar que los objetos anómalos sean ubicados en áreas con relativamente pocos objetos vecinos.

3.6. Teoría de conjuntos rugosos

La teoría de conjuntos rugosos (TCR) puede ser usada en la clasificación para descubrir relaciones estructurales dentro de datos imprecisos o ruidosos. Esta se aplica a atributos con valores discretos, por lo que en el caso de atributos continuos, estos deben ser discretizados. TCR también se puede emplear como método para la reducción de dimensionalidad. Usando TCR se puede reducir el número de características a un conjunto denominado *reducto* que contiene aquellas características de mayor relevancia, o sea, que tienen más peso para la distinción entre clases. Dado un conjunto de datos y múltiples formas de discretización sobre sus atributos, múltiples reductos pueden ser generados. Encontrar el reducto óptimo para un conjunto de datos determinado es un problema NP-hard. Es por eso que muchos trabajos aplican TCR combinado con algún algoritmo para reducir el costo computacional de encontrar el reducto óptimo y así hacer viable el uso de TCR en aplicaciones como la de detección de intrusos [41] [42].

La piedra angular de la teoría de conjuntos rugosos es el principio de que cada objeto de interés posee información asociada y los objetos que presentan la misma información son indiscernibles o indistinguibles respecto esa información [43]. Un sistema de información puede ser representado por una tabla donde las filas son los objetos y las columnas, los atributos. Los atributos a su vez se dividen en dos conjuntos, condición C y decisión D. La decisión es el atributo o conjunto de atributos para los cuales queremos inferir reglas, por lo tanto puede ser visto como la etiqueta que dice a qué clase pertenece el objeto. A grandes rasgos, sean $c_i \in C$ atributos condición y $d \in D$ atributo de decisión (clase), las reglas que se desean obtener son de la forma $(c_1 = value \wedge c_2 = value \wedge \ldots \wedge c_n = value) \Rightarrow (d = claseX)$.

Mediante TCR también es posible eliminar aquellos atributos redundantes, o que no presentan información suficiente como para ser considerados en la toma de decisiones. Se le dice *reducto* al mínimo conjunto de atributos significativos. Una vez que se obtiene un reducto, las reglas son generadas a partir del mismo, los detalles del proceso de obtención de reglas se puede ver en [43].

Sengupta et. al. (2013) [41] combina TCR y *Q-learning*, con el objetivo de mejorar la precisión de la clasificación. Q-learning es un algoritmo de aprendizaje por refuerzo, es decir, basado en prueba y error, para la búsqueda de una solución óptima del problema. Para ello, se construye una tabla donde las filas son estados, y las columnas, posibles acciones que se pueden ejecutar en cada estado para acercarse más a un estado objetivo. Dado que TCR es aplicable sobre datos discretos. Los atributos de valores continuos se discretizan en distintos intervalos de valores. Diferentes esquemas de discretización conducirán a diferentes reductos y, por tanto, a reglas de clasificación con mayor o menor precisión. Con el objetivo de encontrar el mejor esquema de discretización para cada atributo, se emplea algoritmo de aprendizaje por refuerzo, *Q-learning*. En este caso, cada esquema de discretización constituye un estado (filas), y las columnas constituyen el conjunto de atributos. Para cada reducto se extraen reglas de asociación y se crea un clasificador, cuya precisión se evalúa sobre los datos de entrenamiento. El reducto que aporta la mejor

precisión en la clasificación se selecciona como estado objetivo. Una vez terminado el algoritmo de aprendizaje por refuerzo, se obtiene una matriz donde cada elemento muestra la efectividad de cada esquema de discretización para el atributo correspondiente.

3.7. Teoría de conjuntos difusos

A diferencia de la teoría clásica de conjuntos, en la cual un elemento pertenece (o no) a un conjunto determinado, la teoría de conjuntos difusos permite que un elemento pertenezca a varios conjuntos con cierto grado. En otras palabras, cada elemento lleva asociado grados de pertenencia a diferentes conjuntos. Por ejemplo, una persona de 16 años puede pertenecer al conjunto de las personas adultas con un grado de pertenencia de 0,4 y, además, puede pertenecer al conjunto de los adolescentes con un grado de pertenencia de 0,8. Así, la teoría de conjuntos difusos resulta muy útil para expresar conceptos imprecisos, como niño, joven, viejo, bajo, medio, alto, etc.

Un conjunto difuso puede marcar la diferencia entre dato e información. Por ejemplo, la edad de una persona es una variable en el rango de 0 a 100 años típicamente. Conocer el número exacto de años de alguien es un dato, pero al expresar edad en categorías como niño, joven o viejo, también se están definiendo conceptos que implican mayor grado de información. Al establecimiento de categorías en variables cuantitativas como la anterior se le denomina categorización, y esta constituye la primera forma de extracción de información a partir de los datos. Definir categorías no es trivial, en algunas ocasiones se definen las categorías sobre la base de la experiencia existente acerca del concepto que se quiere expresar, como en el caso de la edad. En otras situaciones, no se conocen los límites superiores e inferiores que adquieren las variables, ni la distribución de sus valores, y se hace necesario aplicar técnicas y procedimientos para descubrir las categorías y su rango de valores. Otro problema surge cuando se asumen intervalos estrictos para las categorías, por ejemplo, cuál es el número en años que diferencia entre joven y viejo. Generalmente, los límites entre categorías no son estrictos y constituye un error muy común asumirlos de esa forma. La teoría de conjuntos difusos dota a la minería de datos de herramientas para lidiar con este hecho.

La detección de intrusos es un entorno difuso. En muchas ocasiones, no está clara la frontera que separa el comportamiento intrusivo del comportamiento normal. Muchos trabajos se basan en transformar en difusos a los clasificadores ya existentes. El-Semary *et. al* (2005) [44] y Luo y Bridges, (2000) [45] emplean reglas difusas. Un ejemplo de regla de asociación difusa puede ser de la forma $(c_1 = ALTO \wedge c_2 = MEDIO \wedge ... \wedge c_n = BAJO) \Rightarrow (d = claseX)$, donde c_i son atributos que adquieren valores difusos como BAJO, MEDIO, o ALTO. Tian *et. al.* (2005) [46], divide el conjunto de datos en subconjuntos, y para cada uno de ellos genera un árbol de decisión. Luego emplea estos sub-árboles de decisión para evaluar los datos, los resultados de todos los árboles son combinados empleando una integral difusa.

La combinación entre redes neuronales y conjuntos difusos se basa en la complementariedad. Las redes neuronales aportan el mecanismo de aprendizaje que no posee la teoría de conjuntos difusos. Esta última, por su parte, aporta comprensión al conocimiento adquirido por la red neuronal. De forma general, las redes neuronales y los conjuntos difusos se combinan de las siguientes formas:

- Modificar el sistema difuso con aprendizaje supervisado de la red neuronal
- Construir redes neuronales empleando sistemas difusos
- Construir funciones de pertenencia con las redes neuronales
- Concatenar redes neuronales y sistemas difusos.

Toosi y Kahani (2005) [47], emplean clasificadores neuro-difusos para clasificar en cinco categorías correspondientes a los cuatro tipos de ataques genéricos, (ver sección 2) y el tráfico normal. El clasificador

neuro-difuso es un sistema de inferencia difusa que emplean redes neuronales para ajustar los parámetros de las funciones de pertenencia. En una segunda etapa, emplea inferencia difusa para distinguir entre acción intrusiva y no intrusiva.

3.8. Algoritmos biológicamente inspirados

Los algoritmos biológicamente inspirados son resultado de la observación e imitación de procesos que ocurren en la naturaleza, para resolver problemas específicos. A continuación presentamos aquellos que han sido empleados en la detección de intrusiones.

3.8.1. Redes neuronales y mapas auto-organizados

Las Redes Neuronales, (RN) como algoritmo de aprendizaje y clasificación, se basan en el funcionamiento de su homólogo biológico. Cada neurona tiene un conjunto de entradas y una única salida. En la red neuronal se interconectan las neuronas, conformando varias capas. Las entradas y las salidas de las capas exteriores constituyen las entradas y salidas del sistema. Cada neurona implementa una función con parámetros ajustables que, a su vez, toma como argumento los datos de entrada y genera un valor de salida. La red neuronal se entrena con datos previamente etiquetados. Mediante métodos conocidos como *backpropagation*, los parámetros de las funciones de cada neurona se van calibrando automáticamente de manera que el resultado real se acerque más al resultado esperado en el entrenamiento. Una vez entrenada la RN, esta se emplea para clasificar las instancias de datos. En cuanto a la detección de intrusos, las redes neuronales se emplean para clasificar los datos en anómalos o normales, Jian *et. al.* (2010) [48], para clasificar los datos según los distintos tipos de ataques Ahmad *et. al.* (2010) [49], e incluso para reducir la dimensionalidad de los datos Chandrashekar and Raghuveer (2012) [37].

Los Mapas Auto-Organizados, (SOM) del inglés *self-organizing maps*, tienen la arquitectura de una red neuronal con la diferencia de que no reciben retroalimentación para la generación de los pesos y parámetros de las neuronas. En tal sentido, los SOM constituyen la versión no supervisada de las redes neuronales. Para construir las conexiones y obtener los pesos de las neuronas se recurre a la experiencia acumulada a partir de las muestras previamente observadas. Esto hace que los SOM tiendan a agrupar muestras con características similares, tal como se hace en el agrupamiento.

Uno de los usos de SOM en la detección de intrusos es la interpretación de anomalías. Dada su propiedad de agrupar la información, las instancias anómalas se agruparan según su similitud, generando un número de grupos anómalos cuya tendencia central recogerá las características más generales y distintivas de dicho ataque. Simon T. Powes y Jun He (2008) [50] proponen SOM como método para la interpretación de anomalías.

3.8.2. Algoritmos genéticos

Los algoritmos genéticos se inspiran en la teoría evolutiva. A partir de una población de cromosomas (soluciones) generadas aleatoriamente, se evalúa la sobrevivencia de los mismos en un medio determinado. Aquellos que no son aptos (soluciones con bajos niveles de aciertos) son eliminados. A este paso se le denomina selección natural. Luego, a la población sobreviviente se le aplican recombinaciones y mutaciones. En otras palabras, se mezclan cromosomas (soluciones) y se introducen nuevos cambios aleatorios. Este es un proceso iterativo, donde cada iteración constituye una generación de soluciones con más capacidades y adaptabilidad que la anterior. Los algoritmos genéticos son muy empleados en problemas de optimización con grandes espacios de búsqueda. Respecto a la detección de intrusos, cada cromosoma se

puede ver como una regla que es evaluada sobre los datos de entrenamiento, por cada iteración se obtienen reglas más complejas y con mayor alcance.

Sadafir et al., (2010) [51] proponen el empleo de algoritmos genéticos para la detección de ataques del tipo R2L. Empleando como base de entrenamiento KDDCUP99, se obtienen reglas mediante algoritmos genéticos capaces de detectar estos tipos de ataques. El algoritmo genético propuesto es de tipo incremental, en el cual la población puede aumentar su tamaño en cada nueva generación.

3.8.3. Inteligencia de enjambre

La inteligencia de enjambre (IE) surge del comportamiento colectivo de sistemas auto-organizados y descentralizados. Dicho comportamiento colectivo es un producto sinérgico de una población de entidades con comportamiento de alcance local que interactúan entre sí. La inteligencia de enjambre se revela cuando dicho comportamiento colectivo aporta soluciones óptimas a problemas globales, o sea, problemas que están más allá del conocimiento de cada entidad que compone la población.

La inteligencia de enjambre está inspirada en las colonias de seres vivos que existen en la naturaleza y que presentan un comportamiento inteligente, como lo son las colonias de abejas, los cardúmenes de peces, las colonias de hormigas, entre otros. Básicamente, el conjunto de algoritmos de IE parten de tener una población de entidades que interactúan entre sí, y con el medio ambiente. La complejidad del comportamiento de las entidades tiende a ser simple y, comúnmente, tiene un factor aleatorio. IE se ha empleado mayormente en problemas de optimización, donde es difícil encontrar las mejores soluciones a un problema dado. Como etapa para reducir dimensionalidad, los algoritmos de IE se han empleado en la selección de características significativas [42] [38].

Como método de clasificación, los algoritmos de IE se han empleado para derivar reglas de asociación. Parpinelli et al., (2002)[52] propone un algoritmo basado en colonias de hormigas. Sousa et al., (2004) [53] presenta un algoritmo denominado PSO, del inglés *Particle Swarm Optimization*, en el cual entidades denominadas partículas están dotadas de posición y velocidad en un espacio n-dimensional. En la actualización de la posición y velocidad de las partículas intervienen una componente aleatoria, una solución óptima local y una solución óptima global.

Con respecto a la detección de intrusos, (Chung y Wahid, 2012)[42] aplican algoritmos de inteligencia de enjambre en dos etapas. En la primera, para la selección de características con la intensión de reducir la dimemsionalidad. Para ello combinan Teoría de Conjuntos Rugosos TCR con PSO. Dado que es conocido que en TCR encontrar el conjunto de atributos más descriptivos es un problema *NP-hard*, los autores proponen utilizar el algoritmo de enjambre para encontrar una solución óptima. Para la discretización de las características continuas se emplea *K-means*, este algoritmo de agrupamiento permite hacer cortes a las variables, de acuerdo con los agrupamientos de los valores adquiridos. Una vez obtenido el reducto óptimo, en la segunda etapa se lleva a cabo el minado de reglas de asociación, empleando una variante simplificada de PSO que conduce a una mejor precisión en la clasificación. Las reglas para cada característica, x_i obtenidas son de la forma: $IF(LimInferior \leqslant x_i \leqslant LimSuperior)\ is\ true\ THEN\ clase = ClaseX$. Como base de entrenamiento se emplea KDDCUP99.

3.8.4. Seleccion negativa y células dendríticas

Simon T. Powers y Jun He (2008) [50], proponen un sistema híbrido, que en una primera etapa emplea selección negativa para detectar anomalías y, en una segunda etapa, utiliza SOM para extraer conocimiento de las anomalías detectadas. El algoritmo de selección negativa se entrena empleando instancias tráfico normal, cada conexión de red es el análogo de una entidad biológica, y esta se clasifica como antígeno cuando no pertenece al tráfico normal. En la fase de entrenamiento se generan detectores (Linfocitos-B)

en forma de reglas. Aquellos detectores que se activen con instancias de tráfico normal son eliminados, quedando solo los que detectan tráfico anómalo. Para la generación de detectores se emplean algoritmos genéticos y de agrupamiento. SOM es empleado para extraer conocimiento humanamente interpretable.

U. Aikelin et al.(2003) [8] presentan ideas de cómo desarrollar un IDS basado en la Teoría del Daño. Más adelante, Greensmith y Aikelin (2006) [54] realizan experimentos para evaluar el desempeño del algoritmo basado en células dendríticas en la detección de anomalías, específicamente en la detección del escaneo de puertos. Una población de células dendríticas es expuesta a antígenos y señales de daño. Las señales son de dos tipos: seguras, que indican un estadio normal del sistema; y dañinas, que implican la existencia de daño. En dependencia del número de señales recibidas, la célula migra a uno de dos posibles estados, denominados maduro y semi-maduro. El estado maduro se alcanza cuando la correlación entre las señales favorece a las señales de daño. Por otro lado, un estado semi-maduro de la célula implica una correlación favorable a las señales seguras. En el período inmaduro de las células dendríticas, los PIDs de los procesos (antígenos) que corren en el sistema son registrados por estas. Las señales de daño se genera a partir de la cantidad de paquetes ICMP con error que ingresan al sistema, mientras que la señal segura es la razón de paquetes emitidos por el sistema. Cuando las células acumulan una cantidad definida de señales, estas se convierten en células maduras o en células semi-maduras.

En una segunda etapa, las células maduras y semi-maduras son analizadas. Las maduras, recolectaron los antígenos (PIDs) que estuvieron presentes durante un número considerable de señales de daño. Es de esperar que aquellos procesos que más repetidamente se encuentren en un contexto de célula madura sean procesos dañinos al sistema. Por el contrario, las células inmaduras, contienen los procesos presentes durante un contexto no dañino, implicando que estos procesos no son agresores al sistema. Para la generación de la población y el mantenimiento de las células dendríticas, los autores emplearon una plataforma para sistemas autoinmunes, denominada *libtissue* [8]. En Greensmith *et. al.* (2010) [55], se adiciona otra señal denominada señal inflamatoria, que incrementa el efecto de las otras señales en la población de células. Su objetivo es elevar la atención en un área determinada, una vez que se ha detectado un número elevado de células maduras en ella, lo cual es una abstracción del efecto inflamatorio del sistema inmunológico humano.

4. Enfoques híbridos

La diferencia principal entre los MIDS y los AIDS es que los primeros crean y comparan modelos de los patrones de ataques, mientras los segundos crean y comparan modelos de comportamiento normal. En ese sentido, es de esperar que la combinación de ambos mejore la eficacia de la detección. Lee et. al. (2001) [56] generan anomalías artificiales para crear una base de entrenamiento que contenga datos intrusivos y normales. Antes de aplicar los algoritmos de aprendizaje supervisado, es necesario etiquetar estos datos. Los autores proponen hacerlo mediante clasificación no supervisada. De esta forma, mediante algoritmos de agrupamiento se crean grupos cuyas instancias se clasifican según el criterio (iii). Posteriormente, se realiza la clasificación supervisada con estos datos. Específicamente se extraen reglas de asociación.

En (Fernández y Owezarski, 2009) [57], el objetivo fundamental es la interpretación de anomalías previamente detectadas. El enfoque propuesto se resume en cuatro pasos: detectar y alertar anomalías; recolectar todos o la mayoría de los RA involucrados; extraer características de estos datos; aplicar clasificación supervisada sobre los datos anómalos para obtener clasificadores. En el último paso, además de obtenerse un descriptor de patrón útil para detectar la anomalía la próxima vez que ocurra, se obtiene

información que ayuda a entender el modus operandi del ataque. De forma más clara, los clasificadores obtenidos son reglas de ataques.

El sistema propuesto por Brahmi et. al. (2011) [34] es una arquitectura multi-nivel, que los autores denominan sistema de detección de intrusiones distribuido multi-agentes. En dicha arquitectura, cada agente realiza una tarea específica dentro del mecanismo de detección.

El Agente recolector, se encarga de la captura de paquetes y constituye el proveedor de datos del sistema. El Agente filtro, selecciona los campos de datos necesarios de los paquetes recolectados por el agente filtro y ordena los paquetes según el tipo (TCP, UDP, etc). El Agente de evaluación de reglas evalúa las reglas contenidas en la base de conocimiento del sistema sobre los datos recolectados por el agente Filtro; los datos que no son clasificados como intrusivos pasan al siguiente agente, que es el encargado de la detección de anomalías. El agente de detección de anomalías realiza la clasificación no supervisada mediante los algoritmos de agrupamiento *AD-Clust*, que es una combinación de *K-meas* y *DBSCAN*. De forma similar a [57] se crean grupos que son clasificados según (iii) y estos datos son pasados al agente de extracción de reglas. El agente de extracción de reglas utiliza el algoritmo *RETE*, para obtener las reglas que luego conforman la base de conocimiento del sistema.

Om y Kumdu (2012) [31], proponen un enfoque híbrido donde en una primera etapa se aplique un algoritmo de agrupamiento sobre los datos, específicamente *k-means*. Con k = 5, que corresponden a los tipos de ataques, DoS, U2R, R2L, Probe y al comportamiento normal. Luego se etiquetan las muestras, según el grupo al cual pertenecen. Con las muestras etiquetadas se entrena un clasificador, *k-NN*, para que posteriormente pueda detectar los nuevos tipos de ataques. La selección de características se realiza teniendo en cuenta la entropía de las mismas, de forma que se seleccionen aquellas que más información aportan. Wang and Zhao (2012) [58], hacen un estudio donde se evalúa la capacidad de los algoritmos de grupos y de elementos frecuentes para la detección de intrusiones. En el trabajo se demuestra que los algoritmos de agrupamiento son mejores que *Apriory* para detectar los ataques de tipo DoS y Probing, mientras que para U2R y R2L ambos presentan gran cantidad de fallos.

5. Conclusiones

El presente trabajo ha abordado el tema de la detección automática de intrusiones. En las primeras secciones se han tratado temas generales como las definiciones de intrusión informática, las estrategias de los atacantes, las asunciones sobre las que se basa la detección de intrusiones y los enfoques de detección reconocidos por literatura. Posteriormente, se pone a consideración una taxonomía que permite clasificar a los IDS de acuerdo con cinco criterios o dimensiones. En esta se introducen generalizaciones como el concepto de Entidad Potencialmente Atacante, Registros de Actividad y Resolución del Análisis. Estos nos permiten analogar términos encontrados en los trabajos de investigación estudiados, a la vez que dotan de formalidad al problema.

En esta taxonomía también se reconoce a la Detección basada en Teoría de daño (DIDS) como una rama distinta de la Detección basada en Anomalías (AIDS) y la Detección basada en uso inadecuado (MIDS). Como hemos mencionados, esto se debe a que consideramos que esta rama emergente no se centra en detectar lo que es anómalo, sino en detectar aquello que cause verdadero daño al sistema bajo análisis. A lo largo del trabajo estos enfoques han sido tratados particularmente y explicadas sus ventajas y desventajas. Nos sumamos al criterio de que estos enfoques son complementarios y deben integrarse en un mismo sistema, a fin de lograr una solución completa de problema de la detección de intrusiones. En efecto, hemos podido apreciar que los sistemas híbridos o integrados ganan cada vez más espacio.

En la revisión de los algoritmos del estado del arte apreciamos la notable presencia de algoritmos de minería de datos, tanto de clasificación supervisada como no supervisada. Al respecto, queremos hacer notar que la seguridad informática es un entorno distinto al medio en que tradicionalmente se ha desarrollado la minería de datos. En este, la información es dinámica, o sea, los datos varían constantemente en el tiempo. Tradicionalmente, la minería de datos ha tratado con bases de datos cuya información no varía o varía poco por cada solución de los algoritmos. Otro requerimiento es la intensión latente por parte de los agresores de hacer fallar o corromper los modelos de detección. Este aspecto no era considerado cuando se desarrollaron las principales técnicas de minería de datos. F. Roli (2013) [59] denomina a este fenómeno reconocimiento de patrones bajo ataque y propone un modelo denominado Carrera Armamentista, donde el sistema se auto agrede con el objetivo de anticiparse a eventuales ataques.

Finalmente, nos referimos al tema de la eficiencia. En los IDS se requieren tiempos de respuesta lo más cercano posible al tiempo real, a la vez que debe procesarse un número elevado de datos cambiantes. La tecnología de cómputo actual por sí misma no puede aportar la eficiencia requerida, por lo que debe tenerse en cuenta el procesamiento en paralelo en cualquier propuesta que pretenda asegurar de forma eficiente y eficaz los sistemas informáticos.

Referencias bibliográficas

1. Ali A. Ghorbani, Wei Lu, M.T.: Network Intrusion Detection and Prevention Concepts and Techniques. Springer (2011)
2. Carl Endorf, Eugene Schultz, J.M.: Intrusion detection and prevention. McGraw-Hill (2004)
3. Karen Scarfone, P.M.: Guide to Intrusion detection systems. National Institute of StÃ¡ndars and Technology, Technology administration US (2007)
4. Liu, X.A., Meiners, C.R., Torng, E.: Regular expression matching using tcams for network intrusion detection (march 2012) US Patent 20,120,072,380.
5. Thinh, T.N., Hieu, T.T., Dung, V.Q., Kittitornkun, S.: A fpga-based deep packet inspection engine for network intrusion detection system. In: Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2012 9th International Conference on. (may 2012) 1 –4
6. Jose M, B., JosÃ© HernÃ¡ndez, P.: Multi-character cost-effective and high throughput architecture for content. Microprocessors and Microsystems **37** (2013) 1200–1207
7. Roberto Di Pietro, L.V.M., ed.: Intrusion Detection Systems (Advances in Information Security). Springer (2008)
8. Aickelin, U., Bentley, P., Cayzer, S., Kim, J., McLeod, J.: Danger theory: The link between ais and ids? In: Artificial Immune Systems. Springer (2003) 147–155
9. Aickelin, U., Greensmith, J.: Sensing danger: Innate immunology for intrusion detection. Information Security Technical Report **12**(4) (2007) 218–227
10. Wang, L., Jajodia, S.: An approach to preventing, correlating, and predicting multi-step network attacks. Intrusion Detection Systems (2008) 93
11. : Snort project
12. Oldmeadow, J., Ravinutala, S., Leckie, C.: Adaptive clustering for network intrusion detection. In: Advances in Knowledge Discovery and Data Mining. Volume 3056. Springer Berlin Heidelberg (2004) 255–259
13. Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S.: A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. Department of Computer Science Columbia University (2002)
14. Brutlag, J.D.: Aberrant behavior detection in time series for network monitoring. In: LISA. (2000) 139–146
15. Lakhina, A., Crovella, M., Diot, C.: Diagnosing network-wide traffic anomalies. SIGCOMM Comput. Commun. Rev. **34**(4) (August 2004) 219–230
16. Barford, P., Kline, J., Plonka, D., Ron, A.: A signal analysis of network traffic anomalies. In: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment, ACM (2002) 71–82
17. Krishnamurthy, B., Sen, S., Zhang, Y., Chen, Y.: Sketch-based change detection: methods, evaluation, and applications. In: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, ACM (2003) 234–247
18. Soule, A., Salamatian, K., Taft, N.: Combining filtering and statistical methods for anomaly detection. In: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement, USENIX Association (2005) 31
19. Greensmith, J., Aickelin, U., Cayzer, S.: Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. In: Artificial Immune Systems. Springer (2005) 153–167

20. Brockwell, P.J., Davis, R.A.: Introduction to Time Series and Forecasting. Springer, New York (1996)
21. Falk, M., Marohn, F., Michel, R., Hofemann, D., Macke, M., Spachmann, C., Englert, S.: A First Course on Time Series Analysis. Chair of Statistics, University of Wurzburg (March 20, 2011)
22. Charikar, M., Chen, K., Farach-Colton, M.: Finding frequent items in data streams. In: Automata, Languages and Programming. Springer (2002) 693–703
23. Box, G.E., Jenkins, G.M., Reinsel, G.C.: Time series analysis: forecasting and control. Wiley. com (2013)
24. Jolliffe, I.: Principal component analysis. Wiley Online Library (2005)
25. Robert Grover, B., Patrick, H.: Introduccion to Random Signals and Applied Kalman Filtering. Jhon Winley & Son, Inc. (2011)
26. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature distributions. In: ACM SIGCOMM Computer Communication Review. Volume 35. (2005) 217–228
27. Casas, P., Mazel, J., Owezarski, P.: Unada: unsupervised network anomaly detection using sub-space outliers ranking. In: Proceedings of the 10th international IFIP TC 6 conference on Networking - Volume Part I. Volume I of NETWORKING'11., Berlin, Heidelberg, Springer-Verlag (2011) 40–51
28. Casas, P., Mazel, J., Owezarski, P.: Unsupervised network intrusion detection systems: Detecting the unknown without knowledge. Computer Communications **35** (2012) 772–783
29. Bakhoum, E.: Intrusion detection model based on selective packet sampling. EURASIP Journal on Information Security, Springer International Publishing AG **2011** (2011) 1–12
30. Portnoy, L., Eskin, E., Stolfo, S.: Intrusion detection with unlabeled data using clustering. In: In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001, Citeseer (2001)
31. Om, H., Kundu, A.: A hybrid system for reducing the false alarm rate of anomaly intrusion detection system. In: Recent Advances in Information Technology (RAIT), 2012 1st International Conference on, IEEE (2012) 131–136
32. Zanero, S., Savaresi, S.M.: Unsupervised learning techniques for an intrusion detection system. In: Proceedings of the 2004 ACM symposium on Applied computing. SAC '04, New York, NY, USA, ACM (2004) 412–419
33. Leung, K., Leckie, C.: Unsupervised anomaly detection in network intrusion detection using clusters. In: Proceedings of the Twenty-eighth Australasian conference on Computer Science. Volume 38., Australian Computer Society, Inc. (2005) 333–342
34. Brahmi, I., Ben Yahia, S., Aouadi, H., Poncelet, P.: Towards a multiagent-based distributed intrusion detection system using data mining approaches. In: Proceedings of the 7th international conference on Agents and Data Mining Interaction, Berlin, Heidelberg, Springer-Verlag (2012) 173–194
35. Jun Zhang, Zhijian Wang, S.Z., Meng, X.: Intrusion detection method based on legclust algorithm. Applied Mechanics and Materials **263 - 266** (2012) 3025–3033
36. Horng, S.J., Su, M.Y., Chen, Y.H., Kao, T.W., Chen, R.J., Lai, J.L., Perkasa, C.D.: A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert systems with Applications, Elsevier **38**(1) (2011) 306–313
37. Chandrashekar, A.M., Raghuveer, K.: Fusion of multiple data mining techniques for effective network intrusion detection: a contemporary approach. In: Proceedings of the Fifth International Conference on Security of Information and Networks. SIN '12, New York, NY, USA, ACM (2012) 178–182
38. Xiaoqin Zhang, G.J.J.: A kind of network intrusion detection method using improved support vector machine based on ant colony algorithm. Applied Mechanics and Materials **263 - 266** (2012) 2995–2998
39. Kotenko, I., Laskov, P., Schäfer, C.: Intrusion detection in unlabeled data with quarter-sphere support vector machines. In: Proc. of the International GI Workshop on Detection of Intrusions and Malware & Vulnerability Assessment, number P-46 in Lecture Notes in Informatics. (2004) 71–82
40. Rajeswari, L.P., Kannan, A.: An intrusion detection system based on multiple level hybrid classifier using enhanced c4. 5. In: Signal Processing, Communications and Networking, 2008. ICSCN'08. International Conference on, IEEE (2008) 75–79
41. Sengupta, N., Sen, J., Sil, J., Saha, M.: Designing of on line intrusion detection system using rough set theory and q-learning algorithm. Neurocomputing (2013)
42. Chung, Y.Y., Wahid, N.: A hybrid network intrusion detection system using simplified swarm optimization (sso). Applied Soft Computing **12**(9) (2012) 3014 – 3022
43. Lin, T.Y., Cercone, N.: Rough sets and data mining: Analysis of imprecise data. Kluwer Academic Publishers (1996)
44. El-Semary, A., Edmonds, J., Gonzalez, J., Papa, M.: A framework for hybrid fuzzy logic intrusion detection systems. In: Fuzzy Systems, 2005. FUZZ'05. The 14th IEEE International Conference on, IEEE (2005) 325–330
45. Luo, J., Bridges, S.M., Vaughn Jr, R.B.: Fuzzy frequent episodes for real-time intrusion detection. In: Fuzzy Systems, 2001. The 10th IEEE International Conference on. Volume 1., IEEE (2001) 368–371
46. Tian, J.f., Fu, Y., Xu, Y., Wang, J.l.: Intrusion detection combining multiple decision trees by fuzzy logic. In: Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005. Sixth International Conference on, IEEE (2005)
47. Toosi, A.N., Kahani, M.: A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. Computer communications **30**(10) (2007) 2201–2212

48. Jiang, X.S., Wei, X.M., Geng, Y.S.: The research of intrusion detection system based on ann on cloud platform. Applied Mechanics and Materials **263** (2013) 2962–2965
49. Ahmad, I., Abdullah, A.B., Alghamdi, A.S.: Remote to local attack detection using supervised neural network. In: Internet Technology and Secured Transactions (ICITST), 2010 International Conference for, IEEE (2010) 1–6
50. Powers, S.T., He, J.: A hybrid artificial immune system and self organising map for network intrusion detection. Information Sciences, Elsevier **178**(15) (2008) 3024–3042
51. Ali, S., Shahzad, W., Khan, F.A.: Remote-to-local attacks detection using incremental genetic algorithm. In: Internet Technology and Secured Transactions (ICITST), 2010 International Conference for, IEEE (2010) 1–6
52. Parpinelli, R.S., Lopes, H.S., Freitas, A.A.: An ant colony algorithm for classification rule discovery. Data mining: A heuristic approach **208** (2002) 191–132
53. Sousa, T., Silva, A., Neves, A.: Particle swarm based data mining algorithms for classification tasks. Parallel Computing **30** (2004) 767–783
54. Greensmith, J., Twycross, J., Aickelin, U.: Dendritic cells for anomaly detection. In: Evolutionary Computation, 2006. CEC 2006. IEEE Congress on. (2006)
55. Greensmith, J., Aickelin, U., Tedesco, G.: Information fusion for anomaly detection with the dendritic cell algorithm. Information Fusion **11**(1) (2010) 21–34
56. Lee, W., Stolfo, S., Chan, P., Eskin, E., Fan, W., Miller, M., Hershkop, S., Zhang, J.: Real time data mining-based intrusion detection. In: DARPA Information Survivability Conference amp; Exposition II, 2001. DISCEX '01. Proceedings. Volume 1. (2001) 89–100 vol.1
57. Fernandes, G., Owezarski, P.: Automated classification of network traffic anomalies. In Mazzola, G.B., Cherlin, P.B., eds.: Security and Privacy in Communication Networks, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering,. Volume 19., Berlin, Heidelberg, Springer-Verlag (2009) 91
58. Wang, M., Zhao, A.: Investigations of intrusion detection based on data mining. Recent Advances in Computer Science and Information Engineering, Springer (2012) 275–279
59. Fabio, R.: Pattern recognition systems under attack. In: 18th Iberoamerican Congress on Pattern Recognition, Springer (2013)

RT_022, junio 2014
Aprobado por el Consejo Científico CENATAV

Editor: Lic. Lucía González Bayona
Diseño de Portada: Di. Alejandro Pérez Abraham
RNPS No. 2143
ISSN 2072-6260
Indicaciones para los Autores:
Seguir la plantilla que aparece en www.cenatav.co.cu
C E N A T A V
7ma. A No. 21406 e/214 y 216, Rpto. Siboney, Playa;
La Habana. Cuba. C.P. 12200
Impreso en Cuba