UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

CENTRO UNIVERSITARIO VALLE DE CHALCO

SISTEMA DETECTOR DE INTRUSIONES OCUPANDO UNA RED NEURONAL ARTIFICIAL

T E S I S

QUE PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A

ING. JOSÉ ERNESTO LUNA DOMÍNGUEZ

TUTORA ACADÉMICA:

M. EN E. ANABELEM SOBERANES MARTÍN

TUTORES ADJUNTOS:

DRA. CRISTINA JUÁREZ LANDÍN

DR. JUVENAL RUEDA PAZ

CENTRO UNIVERSITARIO XICO VALLE DE CHALCO

VALLE DE CHALCO SOLIDARIDAD, MÉXICO. ENERO 2015

Valle de Chalco Solidaridad, Edo. de México a 22 de enero de 2015.

Dr. Samuel Olmos Peña
Coordinador de la Maestría en Ciencias de la Computación
Centro Universitario UAEM Valle de Chalco
Presente

Por este medio le comunicamos a usted que la Comisión Revisora designada para analizar la tesis denominada "**Sistema Detector de Intrusiones Ocupando una Red Neuronal Artificial"**, que como parte de los requisitos para obtener el grado académico de Maestría en Ciencias de la Computación presenta el **Ing. José Ernesto Luna Domínguez** con número de cuenta **0223881** para sustentar el acto de Recepción Profesional, ha dictaminado que dicho trabajo reúne las características de contenido y calidad necesarios para proceder a la impresión del mismo.

Atentamente

Tutora Adjunta	**Tutora Académica**	**Tutor Adjunto**
Dra. Cristina Juárez Landín	**M. Anabelem Soberanes Martín**	**Dr. Juvenal Rueda Paz**

Oficio Coord MACSCO 003/2015
Valle de Chalco Solidaridad, Edo. de México a 26 de enero del 2015.

Ing. José Ernesto Luna Domínguez
Candidato al Grado de Maestría en Ciencias de la Computación
Centro Universitario UAEM Valle de Chalco
Presente

De acuerdo con el Reglamento de Estudios Avanzados de la Universidad Autónoma del Estado de México y habiendo cumplido con todas la indicaciones que la Comisión Revisora realizó con respecto a su trabajo de tesis titulado ***"Sistema Detector de Intrusos Ocupando una Red Neuronal Artificial"***, la Coordinación de la Maestría en Ciencias de la Computación del Centro Universitario UAEM Valle de Chalco, concede la autorización para que proceda a la impresión de la misma.

Sin más por el momento, le reitero la seguridad de mi especial consideración y estima.

ATENTAMENTE
"PATRIA, CIENCIA Y TRABAJO"
"2015, Año del Bicentenario Luctuoso de José María Morelos y Pavón"

Centro Universitario UAEM

Dr. Samuel Olmos Peña.
Coordinador de la Maestría en Ciencias de la Computación.
C.U UAEM Valle de Chalco.
Universidad Autónoma del Estado de México

Valle de Chalco COORDINACIÓN DE MAESTRÍA

c.c.p. Archivo.
SOP

AGRADECIMIENTOS

Le agradezco a Dios por acompañarme siempre a lo largo de mi vida, por ser mi fortaleza en cada uno de los momentos de debilidad que he tenido, así como por brindarme una vida llena de experiencias, aprendizajes y sobre todo de felicidad.

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT), por su valioso apoyo económico que durante estos dos años me brindo. Así como al Consejo Mexiquense de Ciencia y Tecnología (COMECYT), por el apoyo económico para la obtención del grado de la maestría.

Doy gracias a la UAEMex principalmente al CUX por permitirme formar parte de esta gran familia, por creer en mí profesionalmente y por todos los conocimientos que he adquirido en estos 12 años compartidos dentro de este gran hogar, al Físico Víctor por darme la oportunidad y a la Dra. Magally por seguir confiando en mi desempeño.

Gracias a la Maestra Anabelem Soberanes Martín por creer en mí, y brindarme la oportunidad de conocerla no solo como profesora o jefa, sino como persona y amiga, por todo el apoyo y consejos que durante mi vida académica y profesional me ha brindado, por darme la oportunidad de crecer profesionalmente y aprender nuevas cosas. Sin usted probablemente mí historia en el CUX habría terminado hace tiempo.

También me gustaría agradecer a mis profesores de la maestría porque todos han aportado con un granito de arena a mi formación, y en especial a la Dra. Cristina Juárez Landín y al Dr. Juvenal Rueda Paz por aceptar ser mis revisores de tesis, por sus consejos, sus enseñanza y más que todo por su amistad.

A mis compañeros de maestría, en especial a esos "hermanos" con los que compartí experiencias únicas; Héctor (gordo), Filio y Esteban por haber sido excelentes compañeros y amigos, por tenerme la paciencia y por no desesperarse jeje, gracias por ser una familia para mí.

Agradezco a mis padres Maura y Ernesto por apoyarme en cada momento, por todos los consejos y valores que me han inculcado, por ser esa fortaleza cuando la he necesitado, por alentarme a ir siempre por mas, también doy gracias a mi hermana Laura que hace 7 años fue parte fundamental para titularme como ingeniero y que en aquel tiempo no pude agradecer en persona, gracias por tu valioso apoyo desde siempre.

Deseo agradecer a esa persona que siempre ha sido un apoyo y fortaleza sin estar físicamente presente, por todas las motivaciones que junto a Gsusinc me ha brindado. Por esos consejos y por ser el aliciente para que intentara por segunda vez estudiar una maestría, y que además siempre estuvo presente para que no desistiera de terminar mis estudios, agradezco esa forma tan peculiar de siempre estar conmigo, por apostar por mí y por siempre tener un wooola que me motive, gracias Laux por sencillamente ser tú, por seguir enganchada y por siempre acompañarme a transformar mundos e inventar mares que podamos cruzar. Pero principalmente gracias por siempre hacer de cada día un día único. Vamos por el Doctorado.

Son muchas las personas que han formado parte de mi vida a las que me encantaría agradecerles su amistad, consejos, apoyo, ánimo y compañía en los momentos más difíciles de mi vida. Algunas están aquí conmigo y otras en mis recuerdos y en mi corazón, sin importar en donde estén quiero darles las gracias por formar parte de mi vida, por todo lo que me han brindado y por todas sus bendiciones.

Para ellos: Muchas gracias y que Dios los bendiga Siempre.

RESUMEN

La seguridad en los sistemas informáticos es de vital importancia debido al volumen de datos que se manejan en ellos. Los Sistemas Detectores de Intrusos o IDS son una herramienta que permite fortalecer la seguridad de la información. Se han hecho varias implementaciones de estos sistemas, aunque solo algunos han tenido resultados óptimos y estos son los que esta basados en uso indebido, estos son con los que los antivirus cuentan.

Otro modelo son los IDS que están basados en anomalías, estos construyen un modelo del sistema que cuentan con técnicas de inteligencia artificial como las Redes Neuronales Artificiales, lo hacen para detectar o predecir un ataque; esta variación de IDS no han sido tomada en cuenta ampliamente por los encargados de seguridad, esto debido a la cantidad de falsos positivos con la que cuentan.

La aplicación de técnicas basadas en Inteligencia Artificial para la detección de intrusos, fundamentalmente las redes neuronales artificiales, están demostrando ser un enfoque muy adecuado para atacar los problemas abiertos en esta área. Sin embargo, el gran volumen de información que se requiere cada día para entrenar estos sistemas, junto con la necesidad exponencial de tiempo que requieren para asimilarlos, dificulta enormemente su puesta en marcha en escenarios reales. En esta tesis se buscó realizar una red neuronal artificial que junto a un IDS pueda contar con baja tasa de falsos positivos, esto debido al correcto entrenamiento que junto a la base de conocimientos (base de datos) pudieran dotarle a la red neuronal un aprendizaje de alto nivel, lo que le permitirá predecir los ataques además de asegurar que la pérdida de información sea mínima y, en consecuencia, disminuyendo la complejidad del clasificador neuronal y manteniendo estables los tiempos de entrenamiento. Para validar la propuesta se ha diseñado un escenario de prueba mediante un IDS basado en redes neuronales artificiales. Los resultados obtenidos a partir de las pruebas realizadas demuestran la validez de la propuesta y acreditan las líneas futuras de trabajo.

ABSTRACT

Security in computer systems is vital due to the volume of data involved in them. Intrusion Detectors Systems or IDS are a tool to strengthen information security. There have been several implementations of these systems, but only some have had good results and these are the ones based on misuse, these are with the antivirus feature.

Another model is the IDS that are based on anomalies, they constructed a model of the system that have artificial intelligence techniques such as Artificial Neural Networks, make it to detect or predict an attack; this variation of IDS have not been widely taken into account by security officers, that due to the number of false positives with that feature.

The application of techniques based on Artificial Intelligence for intrusion detection, mainly artificial neural networks are proving to be a very suitable approach for addressing open issues in this area. However, the large volume of information that is required every day to train these systems, along with the need exponential time required to assimilate, greatly complicates its implementation in real scenarios. In this thesis we sought to make an artificial neural network with an IDS can have low rate of false positives, this due to the correct training with the knowledge base (database) could provide it to the neural network learning of high level, allowing you to predict the attacks and ensure that data loss is minimal and, consequently, reducing complexity and maintaining stable neuronal classifier training times. To validate the proposal has been designed a test scenario using an IDS based on artificial neural networks. The results obtained from tests show the validity of the proposal and future work credited lines.

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

1.INTRODUCCIÓN

Desde que la computadora fue inventada en los primero años del siglo XX y combinado con los inicios del servicio de internet, el cual sucedió en la década de los años 60, los equipos de cómputo comenzaron a comunicarse entre ellos mismos, en un inicio solo se realizaba en las universidades y hoy en día es imprescindible en los hogares, empresas e inclusive en las industrias, esto gracias a las características del servicio, lo cual permite un fácil acceso de la información.

En la década de los 80's, se comenzó a registrar ataques informáticos; el primero fue en el año 1988, un malware, directamente un gusano llamado Morris, tenía la capacidad de replicarse a sí mismo, lo que infectaba las comunicaciones entre los equipos de cómputo. Al momento de desarrollar el servicio de internet no se consideraron mecanismos de seguridad, y por tanto se vulneraron diversos equipos; esto causo perdida de comunicación, información y de confianza en cuanto a internet. Un año más tarde, en 1989 se registró un virus que se propago por medio disquetes, estos se distribuyeron como medio de promoción de una revista de informática, los principales sectores infectados fueron los hogares y las microempresas (León, 2012).

Un ataque de índole informático consiste en aprovechar las vulnerabilidades de un sistema informático, esto con el propósito de conseguir un beneficio, causar un daño o sencillamente para mostrar el potencial de desarrollo de su creador (León, 2012). Actualmente, se busca causar algún daño económico al propietario del sistema de información, principalmente se busca dañar a organizaciones, ya sean gubernamentales o particulares.

Con el crecimiento exponencial del internet, se han ido incrementando tanto el número de ataques como el tipo de los mismos, generalmente se ocupan los exploits[1] para potencializar las fallas que ocurren en los protocolos de internet, las

[1] programa informático malicioso, o parte del programa, que trata de forzar alguna deficiencia o vulnerabilidad del sistema

aplicaciones así como los sistemas operativos. La seguridad informática tiene como objetivo reducir el factor de riesgo al que se encuentran expuestos los sistemas de información, los sistemas de cómputo y los de comunicación que integran una red u operan de forma individual, para ello, esta disciplina ha creado mecanismos de seguridad (herramientas) para subsanar las debilidades encontradas dentro de la suite de protocolos de TCP/IP[2], los errores de diseño en los sistemas imprevistos y los errores de programación; los cuales pueden ser aprovechados para penetrar o vulnerar los sistemas. Se dice que un sistema está comprometido, cuando presenta deficiencias físicas o de diseño en su estructura, exponiéndolo a un riesgo de ser vulnerado o dañado por una persona ajena al sistema.

Dentro del presente trabajo, un sistema será definido como un conjunto de elementos interrelacionados por medio de software y/o hardware integrados con un propósito específico. El hardware puede estar conformado por un host, un conjunto de hosts o dispositivos de red; dentro de cada host se encuentran cargados archivos y aplicaciones (software) que se relacionan para formar un sistema. Con base en su tamaño este sistema es considerado como un sistema de nivel micro (sistema de sólo host) o como un sistema de nivel macro (sistema de red, conjunto de hosts).

Existen mecanismos de seguridad diseñados para detectar intrusos, lamentablemente estos no garantizan la protección completa del sistema, los firewall son un ejemplo de ello. Se pueden encontrar mecanismos de identificación y autenticación, estos posibilitan identificar adecuadamente a los sujetos y objetos del sistema. La identificación es la declaración de quién es el usuario, mientras que autenticación es la prueba o confirmación de esa identificación (Nassehi, 2003).

Las identidades de los usuarios se verifican mediante tres métodos genéricos: lo que saben (contraseñas, PIN), con lo que cuentan (tarjetas magnéticas, claves electrónicas, entre otras) y finalmente, lo que son (autenticación biométrica como iris o huellas dactilares).

[2] Protocolos principales de redes. TCP, protocolo de control de transmisión, IP, protocolo de internet

Hay otra serie de mecanismos con el objetivo de velar por la disponibilidad de un sistema. Algunos de ellos actúan a modo de filtros, dejando pasar aquella información que esté autorizada, en el caso de routers (con listas de acceso) y cortafuegos. Y por último, los que detectan amenazas, como antivirus y sistemas de detección de intrusos (IDS por sus siglas en inglés). Éstos últimos forman la línea final de defensa en el esquema general de protección de un sistema informático, y no sólo son útiles para detectar incidentes de seguridad, sino también intentos de romper la seguridad; nacen debido a la necesidad de predecir ataques de intrusos.

El elevado uso de los sistemas de cómputo ha incrementado el problema de accesos no autorizados y la manipulación de datos. El alto nivel de conectividad actual, no sólo proporciona acceso a gran cantidad y variedad de fuentes de datos más rápido, sino que lo provee desde cualquier lugar en la red (Power, 2012). Desde el primer gran ataque del gusano en internet de 1998 antes mencionado, ha habido una innumerable cantidad de intrusiones de red que se han violado los mecanismos establecidos para la protección de los sistemas.

Otro aspecto a considerar es la dificultad que conlleva la realización de software, ya que éste es cada vez más complejo y el ciclo de vida del software se está reduciendo significativamente debido al aumento de la competitividad del mercado. Este hecho tiene como consecuencia el realizar diseños pobres, testeo inadecuado y por lo tanto errores en el software que se manifiestan como vulnerabilidades de seguridad. Antes, los intrusos necesitaban de un conocimiento especializado en las redes y las computadoras para poder lograr sus ataques. Debido al incremento del conocimiento del funcionamiento de los sistemas, los intrusos están cada vez más preparados y lo que antes estaba accesible para sólo unos pocos (expertos), hoy en día cualquiera tiene acceso a herramientas para poder determinar las debilidades de los sistemas y explotarlas con el fin de obtener los privilegios necesarios para realizar cualquier acción dañina. Además, de la problemática del software y la facilidad de uso de herramientas, los problemas de configuración de algunos dispositivos son otro problema. Esto ocurre debido a la falta de conocimiento especializado que los administradores de redes deben tener

hoy en día, la falta de recursos para instalar los parches de seguridad, entre otros. Es clara la necesidad de concienciación y enseñanza en torno a la seguridad informática.

En el desarrollo de los IDS se han utilizado diversas técnicas de inteligencia artificial, tanto para la clasificación de los ataques como el reconocimiento de patrones de irrupción, estas técnicas deben permitir la reducción de ataques potenciales y distinguir entre tráfico normal y anormal en la red. Las redes neuronales artificiales son técnicas de inteligencia artificial que cuentan con resultados altamente aceptables comparándolas con otras técnicas, un 98% de efectividad en la mayoría de los casos donde se han utilizado (Axelsson, 2006).

Las Redes Neuronales Artificiales (RNA) o sistemas conexionistas son sistemas de procesamiento de la información cuya estructura y funcionamiento están inspirados en las redes neuronales biológicas. Consisten en un conjunto de elementos simples de procesamiento llamados nodos o neuronas conectadas entre sí por conexiones que tienen un valor numérico modificable llamado peso. Cada neurona recibe una serie de entradas a través de interconexiones y emite una salida (Gómez, 2001). Esta salida viene dada por tres funciones:

1. Propagación (también conocida como función de excitación), que por lo general consiste en el sumatorio de cada entrada multiplicada por el peso de su interconexión (valor neto). Si el peso es positivo, la conexión se denomina excitatoria; si es negativo, se denomina inhibitoria.
2. Activación, que modifica a la anterior; puede no existir, siendo en este caso la salida la misma función de propagación.
3. Transferencia, que se aplica al valor devuelto por la función de activación. Se utiliza para acotar la salida de la neurona y generalmente viene dada por la interpretación que queramos darle a dichas salidas.

Con un paradigma convencional de programación en ingeniería del software, el objetivo del programador es modelar matemáticamente (con distintos grados de formalismo) el problema en cuestión y posteriormente formular una solución

(programa) mediante un algoritmo codificado que tenga una serie de propiedades que permitan resolver dicho problema. En contraposición, la aproximación basada en las RNA parte de un conjunto de datos de entrada suficientemente significativo y el objetivo es conseguir que la red aprenda automáticamente las propiedades deseadas. En este sentido, el diseño de la red tiene menos que ver con cuestiones como los flujos de datos y la detección de condiciones, y se enfoca más en cuestiones tales como la selección del modelo de red, la de las variables a incorporar y el preprocesamiento de la información que formará el conjunto de entrenamiento. Asimismo, el proceso por el que los parámetros de la red se adecuan a la resolución de cada problema no se denomina genéricamente programación (Pérez, 2010).

Biológicamente, un cerebro aprende mediante la reorganización de las conexiones sinápticas entre las neuronas que lo componen. De la misma manera, las RNA tienen un número de procesadores virtuales interconectados que de forma simplificada simulan la funcionalidad de las neuronas biológicas. En esta simulación, la reorganización de las conexiones sinápticas biológicas se modela mediante un mecanismo de pesos, que son ajustados durante la fase de aprendizaje. En una RNA entrenada, el conjunto de los pesos determina el conocimiento de esa RNA y tiene la propiedad de resolver el problema para el que la RNA ha sido entrenada. Por otra parte, en una RNA, además de los pesos y las conexiones, cada neurona tiene asociada una función matemática denominada función de transferencia; dicha función genera la señal de salida de la neurona a partir de las señales de entrada. La entrada de la función es la suma de todas las señales de entrada por el peso asociado a la conexión de entrada de la señal. Algunos ejemplos de entradas son la función escalón de Heaviside, la lineal o mixta, la sigmoide y la función gaussiana, recordando que la función de transferencia es la relación entre la señal de salida y la entrada.

La actividad que una unidad de procesamiento o neurona artificial realiza en un sistema de este tipo es simple. Normalmente, consiste en sumar los valores de las entradas (inputs) que recibe de otras unidades conectadas a ella, comparar esta

cantidad con el valor umbral y, si lo iguala o supera, enviar activación o salida (output) a las unidades a las que esté conectada. Tanto las entradas que la unidad recibe como las salidas que envía dependen a su vez del peso o fuerza de las conexiones por las cuales se realizan dichas operaciones.

En este trabajo de investigación se aborda el tema de detección de intrusiones con el uso de una red neuronal artificial. La propuesta se basa en la clasificación de ataques por medio de las RNA, se lograron porcentajes de efectividad en la detección de hasta el 95% con una precisión del 80% utilizando una base de datos que contiene la información de ataque.

2.ESTADO DEL ARTE

Con el avance de las tecnologías año tras año, los mecanismos de seguridad demandan medidas más elaboradas, las cuales, garanticen una operación segura y la continuidad de los servicios necesarios para los sistemas informáticos. Las medidas deben incluir métodos de detección y una pronta respuesta ante las intrusiones que suceden en tiempo real. Desde que los sistemas de cómputo se comenzaron a interconectar, la necesidad de contar con seguridad en las redes y servidores comenzó a ser un tema de investigación obligatorio. Peter Anderson expone en 1972 los ataques más comunes que se realizaban en los equipos de cómputo, tales ataques fueron perfeccionados con el paso de los años.

En un periodo comprendido entre 1983 y 1995 surgen dos normas fundamentales en la seguridad informática. El Departamento de defensa de los Estados Unidos (DoD por sus siglas en inglés) publica en 1985 la norma 5200.28-STD la cual sigue vigente a estos días. Diez años más tarde el British Standards Institute, publica la norma BS 7799 la cual da pie a la creación de las normas ISO 27000 las cuales regulan la seguridad de los sitios web así como a las redes de comunicación. Para 1996 la Unión Internacional de Comunicaciones (ITU), elabora las normativas para la seguridad de las comunicaciones ITU X800 e ITU X805 (UNION, 1996).

Existe un glosario de seguridad informática generado por la Internet Society network working group, el cual desde el año 2006 se encuentra aún en mejoras, además la ISO cuenta con un manual de normas de seguridad que rigen la intercomunicación actualmente. La aplicación de las recomendaciones de las normas no fue tarea suficiente para proteger las redes, sino que para ello debieron implementarse barreras activas como firewalls (Kenneth, 2010). Desde el año 1980 los firewall logran proteger a la red de una variedad de ataques, sin embargo existen otros ataques que se presentan ante los firewall con apariencia de tráfico normal.

Con la inspección simple de los paquetes de datos, es difícil descubrir si se trata de tráfico normal o de trafico de índole intrusivo, y por ello es necesario contar con sistemas de detección especialistas, estos sistemas deben ser capaces de analizar paquetes que viajan por la red, además de elaborar estadísticas y de detectar confiadamente el trafico maliciosos; a estos sistemas se les ha denominado Sistemas Detectores de Intrusos o intrusiones (IDS por sus siglas en inglés) (Abler, 2003).

2.1 SISTEMAS DETECTORES DE INTRUSOS

J. P. Anderson conocido como el creador del concepto de IDS en el año 1980, dio pauta para que este tipo de sistemas constituyera una amplia tarea de investigación, desde ese año se han desarrollado una variedad de IDS tales como los basados en Servidores llamados HIDS (Host Intrusion Detection System) y en Sistemas de Archivos y Redes, llamados NIDS (Network Intrusion Detection System). A partir de ello se han desarrollado numerosos y diversos NIDS, entre los cuales destaca el SNORT (Roesch, 2010), como un programa detector de Intrusiones más difundido en el ámbito de la seguridad en redes; esta herramienta se distribuye bajo licencia GNU, y recibe el aporte de investigadores de todo el mundo. Snort tiene módulos bajo licencia GNU y módulos propietarios, se puede considerar a Snort como una plataforma, para experimentar con módulos de detección sin tener que preocuparse de la captura de los datos.

Otro desarrollo importante en NIDS es el Bro (Paxson, 2010) que provee una plataforma para experimentación, detección y estudio de ataques a redes, y por otro lado permite la captura distribuida de los paquetes que atraviesan la red. A pesar de que existe un sin número de programas para la detección de intrusiones, es mayor todavía el vacío de soluciones para el grande y diverso tráfico malicioso. Esto es así porque el avance en el poder de cálculo de los microprocesadores modernos, permitió elaborar estrategias y algoritmos que eran prohibitivos hasta hace algunos años, como los complejos tratamientos estadísticos y el uso de redes neuronales.

2.2 REDES NEURONALES ARTIFICIALES

Los científicos en su afán de resolver situaciones complejas, han estudiado las capacidades humanas cerebrales, siendo ésta la base para la creación de nuevas máquinas por ello la inteligencia artificial se ha preocupado por imitar algunos de estos comportamientos de tipo biológico lo cual llevó a que se desarrollaran técnicas tales como las redes neuronales, algoritmos genéticos, y la lógica difusa entre otros, que también han sido adaptados a dispositivos electrónicos.

De esta manera, el objetivo de las redes neuronales, no es resolver problemas complejos como secuencia de pasos, sino como la evolución de un sistema computacional inspirado en el cerebro humano, además constituyen una parte muy importante en el estudio y desarrollo de la inteligencia artificial, ya que pueden ser combinadas con otras herramientas como la lógica difusa, los algoritmos genéticos o los sistemas expertos, lo cual explica su importancia (Barrera, 2010).

Una red neuronal artificial consiste en unidades de procesamiento interconectadas de manera densa, llamadas neuronas, por tener un comportamiento similar al de las neuronas biológicas. Las Unidades de procesamiento reciben y procesan y transmiten señales, tal como las neuronas biológicas (Freeman, 2000).

2.3 ESTUDIOS REALIZADOS

Para el desarrollo de esta tesis fue necesario basarse en algunos trabajos de relevancia sobre el objeto de estudio, estos trabajos fueron realizados tanto en México como en el extranjero.

En la Universidad de Campeche, Barrera Cámara et al (2010), monitorearon los puertos 110 y 80 únicamente, que corresponden a POP3 y HTTP respectivamente, estos puertos son los que principalmente se utilizan en las redes y por lo tanto los que más vulnerables son a los ataques, debido al monitoreo se seleccionaron datos que tienen relación con la dirección IP origen, le asignaron un numero único (identificador), con estos datos generaron una matriz con el propósito

de generar información para clasificar los patrones de ataque realizados a los puertos mencionados.

En la investigación presentada en Aplicación de Redes Neurales para la Detección de Intrusos en Redes y Sistemas de Información (Pérez, 2010), analizaron el tráfico de paquetes TCP, descartando protocolos como UDP e ICMP. La información contenida en el campo de datos de los paquetes también fue tenida en cuenta. Este estudio fue reproducido usando la información más significativa del encabezado TCP, se seleccionaron los primeros 393 caracteres del contenido del paquete. Este límite se debe a que de los paquetes peligrosos que encontraron a lo largo de la investigación el de mayor longitud era de 393 caracteres en el campo de datos. En total fueron seleccionados 402 datos de entrada: 9 campos de la cabecera TCP y los primeros 393 caracteres del contenido del paquete. La red obtenida mostró un poder de detección por encima del 95 % y un alto poder de generalización.

En 2008 en la Universidad Santiago de Chile (USACH), un grupo de investigación en tópicos de seguridad informática realizó una propuesta de IDS, el cual está basado en redes neuronales recurrentes. Tomaron paquetes de las capas de transporte y de red dentro de los protocolos TCP/IP, consideraron solo 25 características de cada paquete analizado, y 29 puertos, lo que da un conjunto de 54 características y con ellas inicializaron la red neuronal con aprendizaje de patrones temporales denominados Elman/Jordan, que no es más que una mejora de un perceptron multicapa. Las neuronas recuerdan las actividades realizadas lo que permite obtener un significativo nivel de clasificación correcta, en esta investigación lograron obtener el 98.178% de clasificación (Valenzuela, 2008).

En la revista International Journal Of Computer Science And IT vol. 2 No. 6, publicada en Diciembre de 2010 (Das, 2010), se presentó un artículo que lleva por nombre “Network Intrusion Detection Based On Machine Learning Algorithms”, en él se expone que lograron obtener resultados por arriba de los 98.5% con RST (Rough Set Theory), ellos contaron con 13 variables generadas por los paquetes

obtenidos en el monitoreo del tráfico de una red, las variables están estrechamente ligadas con paquetes TCP, uno con ICMP, dos con UDP y agregaron uno que indica el tipo de ataque que representaba (normal, DoS o anormal).

Liu, Yi y Yang (2009), utilizaron el set de datos denominado KDD 99, que es un conjunto de datos simulados de intrusiones militares en un entorno de red, proporcionadas por DARPA *Intrusion Detection Program Evaluation* en 1998, que tenían como objetivo evaluar el estudio y la investigación en la detección de intrusiones. Se realizó un proceso de análisis de componentes principales, los cuales alimentaran una ANN como clasificador, obteniendo resultados positivos de un 97.05%.

En el artículo, “Sistema Neuronal de Detección de Intrusos In: Tendencias en Ingeniería de Software e Inteligencia Artificial” (Bonilla, 2008), se presentan los avances en una investigación que integra una ontología de detección y de prevención de intrusiones a partir de interacciones basadas en sistemas multi-agente y razonamiento aplicando técnicas híbridas de inteligencia computacional, para este caso, un modelo de clasificación de ataques y de reconocimiento de patrones.

Papavassiliou (Manikopoulos, 2011) quien propuso el método de detección de intrusiones estadísticas, utilizó como herramienta la estadística de KolmogorovSmirnov junto a redes neuronales para modelar y detectar ataques. Se puede decir que todo su trabajo se basó en simulaciones en computadoras, no llegando a probarlo en un ambiente real.

Otro hallazgo interesante y nodal lo ofreció el estudio acerca de la metodología de autorganización en la inteligencia colectiva que poseen las colonias de hormigas, y su aplicación en el análisis del tráfico de redes (Gao, 2005). Si bien este mismo método ha sido implementado por otros autores, no es menos valioso el aporte de Gopalakrishna cuando incorpora una captura distribuida de datos. Los IDS y particularmente los métodos estadísticos, el uso de redes neuronales y los sistemas de auto-organización de inteligencia colectiva o agentes cooperantes son

los temas más investigados actualmente, descriptos en una profusa publicación de artículos.

Todos los artículos mencionados anteriormente se fueron reproducidos, desafortunadamente la mayoría de ellos no llegan a los resultados indicados, existe un mundo ambiguo en el que muchos quieren estudiar el tema del uso de la inteligencia artificial para detectar intrusos, pero no todos han podido llegar a cumplir el objetivo.

3.MARCO TÉORICO

Se inicia definiendo las vulnerabilidades y los ataques informáticos, posteriormente se describirá lo referente a los IDS, desde su concepto hasta su clasificación, pasando por sus requisitos y su taxonomía entre otras características. Al final se abordara el tema de las RNA, se verá su definición, sus topologías y los modelos más generales de RNA.

3.1 VULNERABILIDADES Y ATAQUES INFORMATICOS

La información es un activo esencial para las operaciones de cualquier organización, y por lo tanto necesita ser protegida convenientemente. La seguridad de la información es una disciplina que tiene por objeto asegurar y proteger las tres propiedades fundamentales de la información de los sistemas (Andersen, 2001):

- Confidencialidad: Es la habilidad de un sistema para presentar sus recursos accesibles solo a las partes autorizadas a su uso.
- Integridad: Es la habilidad de un sistema que permite que solo las partes autorizadas puedan modificarlo y solo en las formas que son consistentes con las funciones realizadas por el sistema.
- Disponibilidad: Los derechos válidos de acceso a la información nunca deben ser denegados y deben ser satisfechos en tiempo y forma.

Existe un amplio consenso que todos los demás pueden ser derivados de los tres paradigmas básicos. Hoy en día la mayor parte de la información en uso es procesada a través de sistemas de computación, por esto es común que el término “Seguridad de la Información” se use para denotar “Seguridad de Computadoras”, pero académicamente hablando se extiende a los procesos de manejo y almacenamiento de la información, ya sea en papel o almacenada electrónicamente, sea enviada por vía postal o usando medios electrónicos.

El U. S. National Information System Security Glossary define a la seguridad de los sistemas de información como; la protección de los sistemas de información contra el acceso o la modificación sin autorización ya sea estén almacenados,

procesados o en tránsito, y contra la denegación de servicio a los usuarios debidamente autorizados, incluyendo las medidas necesarias para detectar, documentar y contrarrestar tales intentos (Agency, 2010).

Arquitectura AAA (Autenticación, Autorización, Auditabilidad)

El paradigma de confidencialidad, integridad y disponibilidad (CID) de la información contenida en un sistema de computadoras es usualmente implementado a través de la arquitectura AAA la cual cuenta con lo siguiente (Lucas, 2007):

- Autenticación: El usuario es propiamente identificado de alguna manera y un perfil de acceso es asociado a él.
- Autorización: Cada operación y tarea activada por el usuario está sujeta a un conjunto de restricciones, dadas por los privilegios de acceso a los activos del sistema.
- Auditabilidad: Las operaciones y las tareas realizadas son registradas con un proceso propio en orden a asegurar que no se ha producido ninguna violación a los paradigmas del CID.

Este concepto de taxonomía A.A.A. se aplica a los sistemas operativos de red y a los servicios de red, pero también a los sistemas de control de las redes, tales como firewalls, redes virtuales privadas o VPN por sus siglas en inglés. Esto sucede porque la idea de autorización y autenticación son ortogonales a la mayoría de los procesos y servicios de red. La autenticación puede ser realizada a través de varias técnicas, a menudo divididas en las siguientes (Magno, 2006):

- Algo que el usuario debe conocer, por ejemplo la palabra clave.
- Algo que el usuario debe poseer, por ejemplo “tarjetas inteligentes”, “llaves”.
- Algo que el usuario es, por ejemplo huella digital, iris del ojo.

Diferentes modelos han sido propuestos en la literatura para la gestión del control de acceso en aplicaciones distribuidas. Tradicionalmente, los modelos de control de acceso han sido caracterizados mediante modelos DAC (Discretionary Access Control) y modelos MAC (Mandatory Access Control). Posteriormente modelos

RBAC (Role-Based Access Control) o modelos TBAC (Task-based access control) han sido propuestos para gestionar los requerimientos de seguridad en un gran conjunto de aplicaciones. A continuación se resumen las características de estos modelos junto con sus limitaciones más importantes (Sánchez, 2007).

Modelo Discretionary Access Control (DAC)

El modelo de control de acceso discrecional, también llamado modelo de seguridad limitada, es un modelo no orientado al control del flujo de sistema, son controlados y se especifican reglas de autorización de acceso para cada sujeto y objeto. Los sujetos pueden ser usuarios, grupos o procesos. Los modelos DAC están basados en la idea de que el propietario de un objeto, su autor, tiene el control sobre los permisos del objeto. Es decir, el autor es autorizado a permitir u otorgar permisos para este objeto a otros usuarios. DAC admite la copia de datos desde un objeto a otro por usuarios autorizados, de manera que un usuario puede permitir el acceso para copiar datos a otro usuario no autorizado. Este riesgo pude ser extendido a todo el sistema violando un conjunto de objetos de seguridad. La principal ventaja de DAC es que el usuario se beneficia de la flexibilidad del modelo. Sin embargo es difícil para DAC garantizar las reglas de integridad como privilegios mínimos o separación del deber, que son necesarias en los ambientes con procesos colaborativos. DAC es apropiado en ambientes donde la compartición de información es más importante que su protección.

Modelo Mandatory Access Control (MAC)

En el modelo de control de acceso obligatorio todos los sujetos y objetos son clasificados basándose en niveles predefinidos de seguridad que son usados en el proceso de obtención de los permisos de acceso. Para describir estos niveles de seguridad todos los sujetos y objetos son marcados con etiquetas de seguridad que siguen el modelo de clasificación de la información militar, formando lo que se conoce como política de seguridad multinivel. Por este motivo se define MAC como un modelo multinivel (Mellado, 2007)

Modelo Rol Based Access Control (RBAC)

El principal objetivo del modelo de control de acceso basado en rol es prevenir que los usuarios tengan libre acceso a la información de la organización (Britos, 2010). El modelo introduce el concepto de rol y asocia a los usuarios con las funciones por los que va pasando durante la vida del sistema, los permisos de acceso están asociados a los roles, el rol es un concepto típico usado en empresas para ordenar y estructurar sus actividades organizativas. RBAC permite modelar la seguridad desde de una perspectiva empresarial puesto que se pueden conectar los requerimientos de seguridad con los roles y las responsabilidades existentes en la organización. RBAC está basado en la definición de un conjunto de elementos y de relaciones entre ellos. A nivel general describe un grupo de usuarios que pueden estar actuando bajo un conjunto de roles y realizando operaciones en las que utilizan un conjunto de objetos como recursos.

En una organización, un rol puede ser definido como una función que describe la autoridad y responsabilidad dada a un usuario en un instante determinado (Mira, 2009). Entre estos elementos se establecen relaciones del tipo:

- Relaciones entre usuario y roles, modelando los diferentes roles que puede adoptar un usuario.
- Conjunto de operaciones que se pueden realizar sobre cada uno de los objetos. A los elementos de esta relación se les denomina permisos. Relaciones entre los permisos y los roles.

El modelo RBAC incluye un conjunto de sesiones donde cada sesión es la relación entre un usuario y un subconjunto de roles que son activados en el momento de establecer dicha sesión. Cada sesión está asociada con un único usuario, mientras que un usuario puede tener una o más sesiones asociada, los permisos disponibles para un usuario son el conjunto de permisos asignados a los roles que están activados en todas las sesiones del usuario, sin tener en cuenta las sesiones establecidas por otros usuarios en el sistema. RBAC añade la posibilidad de modelar una jerarquía de roles de forma que se puedan realizar generalizaciones y

especializaciones en los controles de acceso y se facilite la modelización de la seguridad en sistemas complejos (Coyne, 2006).

Modelo Task Based Access Control (TBAC)

El control de acceso basado en tareas permite controlar el acceso en entornos representados por el flujo de trabajo. El modelo TBAC extiende los tradicionales modelos de control basados en sujetos/objetos incluyendo aspectos que aportan información contextual basada en las actividades o tareas (Coyne, 2006). El control de acceso en TBAC es garantizado por medio de etapas de autorización, las cuales son un concepto abstracto introducido por TBAC para modelar y manejar un sistema de permisos relacionados con el progreso de las tareas o actividades dentro del contexto de un flujo de trabajo. Este concepto está compuesto por una serie de elementos y atributos. A continuación se describen los elementos más representativos:

- Estado del Proceso: Indica cómo ha progresado la etapa de autorización en su ciclo de vida.
- Estado de Protección: Define todos los permisos que pueden ser activados por la etapa de autorización y que son mantenidos por la propia.
- Conjunto de administradores: Contiene información relevante acerca del conjunto de administradores que potencialmente pueden conceder/invocar la etapa de autorización así como sus identidades de usuario y sus roles.
- Administrador Ejecutor: Identifica el miembro del conjunto de administradores que eventualmente invoca la etapa de autorización.

Seguridad y Vulnerabilidades

En la ingeniería de software el paradigma CID pertenece al dominio de los requerimientos, estableciendo los objetivos de más alto nivel relacionados con la seguridad de la información. La arquitectura AAA y sus componentes son especificaciones de software y hardware de la arquitectura de sistemas en cual se esfuerza para implementar esos requerimientos. Por lo tanto los sistemas de seguridad son las implementaciones prácticas de esas especificaciones. La

confianza puesta en ese proceso puede ser expresado en términos de Garantía según lo expresa Stoneburner (2001), la garantía puede ser definida como la base para las medidas de seguridad, tanto los trabajos operacionales como técnicos orientados a proteger los sistemas, los procesos de información y los objetivos de seguridad, integridad, disponibilidad y confidencialidad han logrado encontrar una implementación especifica.

Es evidente de que el ambiente no es ideal y por lo tanto hay debilidades que afectan el camino entre requerimientos y aplicaciones. Estas debilidades se pueden resumir en las siguientes:

- Debilidades de Análisis al establecer los requerimientos de confidencialidad, integridad y disponibilidad para los activos de la información.
- Debilidades de diseño mientras se trasladan los requerimientos de alto nivel en especificaciones en términos de políticas y arquitecturas para autenticación, autorización y auditoría.
- Debilidades de implementación mientras se codifica, implementa y configura los sistemas de seguridad.

Adicionalmente los requerimientos de seguridad no son estables sino que interactúan en forma continua con el medio y por lo tanto es necesario un ciclo de desarrollo de para mantener la seguridad de los sistemas en forma permanente adaptándose a las cambiantes necesidades de los mercados.

La seguridad de la información es por si solo la ciencia de lo incierto. La seguridad de la información se debe manejar de acuerdo a la administración de riesgos que se está dispuesto a correr, la seguridad absoluta no existe o es infinitamente cara, por lo tanto se debe medir el riesgo de pérdida o afectación de la seguridad de la información para determinar la inversión en seguridad a realizar. Varias normas ISO definen claramente las diferencias entre amenaza, vulnerabilidad y riesgo:

- Riesgo: Combinación de la probabilidad de un evento y su consecuencia.

- Amenaza: Causa potencial de un incidente no deseado, el cual va resultar en un daño a los sistemas u organización.
- Vulnerabilidad: Debilidad de un activo o grupo de activos que pueden ser explotados por una o más amenazas.

Las tareas relacionadas con administrar y reducir los riesgos relacionados con el uso de la información, para reducir o manejar las vulnerabilidades o amenazas. Es un error pensar en la seguridad en términos de reducir las vulnerabilidades. La seguridad es un componente del proceso de la administración de riesgos de la organización, dicho de otra forma la seguridad de la información es la protección de la información de un amplio rango de amenazas en orden a asegurar la continuidad de las organizaciones, minimizar los riesgos y maximizar el retorno de las inversiones y las oportunidades de negocios.

Los componentes de la administración de riesgos son (Abler, 2003):

- Análisis de riesgos: uso sistemático de la información para identificar las fuentes de riesgo y estimarlos.
- Evaluación de riesgos: El proceso de comparar el riesgo estimado contra el riesgo el criterio de riesgo dado para determinar el significad del riesgo.
- Auditoría de riesgos: Todo el proceso de análisis de riesgo y evaluación de riesgo
- Tratamiento de los riesgos: El proceso de selección e implementación de las medidas para reducir los riesgos.

Desde el momento en que se comenzó a interconectar computadoras para formar redes, aparecen las amenazas (Anderssen, Computer security technology planing study, 1972) expone el concepto de acción maliciosa en los servidores y los intentos de penetración. En el 2000 la Internet Society network working group elabora un glosario de seguridad informática en el que se hace referencia a los principales tipos de ataques a las redes de computadoras, distinguiendo entre ataques pasivos y activos.

Ataques Pasivos

Ataques de escucha sin autorización o de monitoreo de tráfico, los objetivos de estos ataques consisten en obtener la mayor cantidad de información del mensaje transmitido y del oponente. Las distintas modalidades de ataques pasivos son las siguientes:

- Descarga de contenidos del mensaje: Están incluidos dentro de este tipo de ataque la escucha de una conversación telefónica, la lectura de un mensaje de correo electrónico o la información confidencial capturada por un oponente.
- Análisis de tráfico: Este es un ataque muy sutil, se supone que hay medios de envíos de mensajes confidenciales, que no permiten al atacante poder acceder al contenido del mensaje. El atacante tiene sólo la posibilidad de observar la transmisión de los mensajes y obtener de éstos, por ejemplo datos tales como: la frecuencia de emisión de los mensajes y la longitud del mensaje. Esta información puede ser de mucha ayuda para inferir la naturaleza de la comunicación.

Ataques Activos

Los ataques activos involucran y comprometen los pilares básicos de las prácticas de seguridad: la confidencialidad, la integridad y la disponibilidad (Quist, 2007). Los ataques activos son:

- Denegación de Servicio DoS (Denegation od Service): El efecto de este ataque es impedir la posibilidad de acceso a toda persona a un determinado servidor.
- Enmascarado: En este caso el atacante se representa él mismo como un legítimo usuario con el objeto de robar, alterar o destruir recursos informáticos.
- Reinterpretar: Este ataque es llevado a cabo mediante una captura pasiva de datos, para que luego sean retransmitidos y con ello producir efectos no autorizados.

- Modificación de contenidos del mensaje: La información original es alterada de tal forma que permita obtener un resultado no autorizado.

Existen básicamente dos tipos de ataques de denegación de servicio según el ataque provenga de una fuente DoS (Denegation of Service) o de varias fuentes Denegación de servicio distribuida DDoS (Distribute Denegation of Service). Uno de los ataques más comunes de denegación de servicio, se produce cuando se establece una conexión Internet, con el protocolo de Transporte de Flujo Confiable (TCP Transport Control Protocol) desde un cliente a un servidor y el cliente envía un paquete de sincronización (SYN), el servidor responde con un paquete de reconocimiento de sincronización (SYN ACK), esperando el reconocimiento del cliente (ACK), para esta operación el servidor crea una cola de tamaño finito esperando que la conexión se complete, si el atacante envía una cantidad suficientemente grande de solicitudes de conexión sin completarlas, produce un desborde de la cola, este tipo de ataque se conoce como inundación TCP- SYN.

Síntomas de los ataques

El CERT (Computer Emergency Readiness Team) de Estados Unidos, expone los síntomas de los ataques de denegación de servicio los cuales incluyen las siguientes manifestaciones:

- Lento rendimiento de la red (apertura de los archivos o el acceso a sitios web)
- Falta de disponibilidad de un sitio web en particular.
- Imposibilidad a acceso a cualquier sitio web.
- Aumento dramático en el número de “spam” recibidos mensajes de correo electrónico - (este tipo de ataque DoS es llamado “Bomba de Mail”.)

No todas las interrupciones de los servicios, incluso aquellos que son el resultado de la actividad maliciosa, son necesariamente de ataques de denegación de servicio. Otros métodos de ataque, pueden incluir una denegación de servicio como un componente de una mayor ofensiva. Ataques de denegación de servicio puede también dar lugar a problemas en la red alrededor de la computadora bajo ataque. Por ejemplo, el ancho de banda de un “router” entre Internet y la red local pueden

ser consumidos por un ataque DoS, poniendo en peligro no sólo el equipo, sino también toda la red. Si la denegación de servicio se realiza a una escala suficientemente grande, la región geográfica de la conectividad a Internet puede verse comprometida, sin conocimiento del atacante o por una mala configuración de la infraestructura de la red (Chang, 2007).

Métodos de ataque

El ataque de “denegación de servicio” se caracteriza por un explícito intento de los atacantes para evitar que los usuarios legítimos de un servicio realicen uso de este; los ejemplos incluyen (Andersen, 2001):

- Inundación de una red, evitando así el tráfico de red legítimo
- Interrumpir un servidor mediante el envío de más solicitudes de lo que posiblemente puede manejar, lo que impide el acceso a un servicio;
- Impedir a una persona en particular el acceso a un servicio;
- Interrumpir el acceso a un servicio específico a una persona.

Los ataques se pueden enviar a cualquier dispositivo de red, incluidos los ataques a los dispositivos de enrutamiento y acceso a la Web, correo electrónico, o al servicio de Sistemas de Nombre de Dominio DNS. Un ataque de denegación de servicio puede ser perpetrado de diferentes formas. Existen tres tipos básicos de ataques de acuerdo a Chang J. (2007):

- El consumo de recursos computacionales.
- Conectividad de la Red.
- Uso de los propios recursos en contra de un mismo Ancho de banda.
- Consumo de otros recursos como: Espacio en disco; Tiempo de CPU.
- Perturbación de la información de configuración, como la información de enrutamiento; información de estado, peticiones no solicitada de reiniciar sesiones TCP;
- Perturbación física de los componentes de la red.

Un ataque de denegación de servicio puede incluir la ejecución de malware destinados a:

1. Maximizar el uso de la CPU evitando la ejecución de cualquier tipo de trabajo; desencadenar errores en el código de la máquina;
2. Desencadenar errores en la secuencia de instrucciones, con el fin de forzar a la computadora a un estado de inestabilidad o de inmovilización;
3. Explotar errores en el sistema operativo de recursos a causa del agotamiento de estos obligándolo a utilizar todos los recursos disponibles por lo que no se puede ejecutar el trabajo real.

Ataque Smurf

Un ataque smurf es una variante particular de un ataque de denegación de servicio de inundación en Internet. Se basa en la configuración errónea de los dispositivos de red que permiten a los paquetes que se envían a la red a través de la dirección de difusión. Normalmente se utilizan paquetes ICMP de solicitud de eco. La red sirve entonces como un “smurf” amplificador. En donde los atacantes envían un gran número de paquetes IP con la dirección de la fuente falsa. Para luchar contra ataques de denegación de servicio en internet, tal como el ataque (smurf) Amplificador, los proveedores de servicio de Internet (ISP) cuentan con la capacidad de identificar configuraciones erróneas de las redes y de adoptar medidas correctivas como el filtrado de paquetes por direcciones de origen (Kumar, 2007).

Ataque Teardrop

El ataque consiste en el envío de paquetes IP fragmentados de tal forma que los fragmentos se superpongan, provocando sobrecargas en la computadora de destino. Los elementos manejados son la superposición de fragmentos, más el tamaño grande de estos provocan sobrecarga en la computadora de destino. Un fallo en el protocolo de fragmentación y rearmado de los paquetes TCP/IP de diversos sistemas operativos causa que los fragmentos no estén bien manipulados, y falle el rearmado (Chang, 2007).

Ataques Peer-to-peer

Los atacantes han encontrado una forma de explotar una serie de errores en los servidores peer-to-peer para iniciar ataques DDoS. El más agresivo de estos ataques DDoS peer-to-peer, es un cliente “open source” para windows para compartir archivos. Los ataques Peer-to-peer son diferentes de los ataques basados en botnet[3]. Con el ataque peer-to-peer no hay botnet y el atacante no tiene que comunicarse con los clientes que desea atacar. En lugar de ello, el atacante actúa como un controlador de marionetas, instruyendo a los clientes peer-to-peer para desconectarse de su par y conectarse a la víctima. Como resultado de ello, varias miles de computadoras pueden agresivamente tratar de conectarse a una página web. Si bien un típico servidor web puede manejar unos cientos de conexiones por segundo antes de comenzar a degradar el servicio, la mayoría colapsa instantáneamente con cinco o seis mil conexiones por segundo, un ataque moderado podría generar un máximo de 750,000 conexiones en un corto lapso de tiempo.

El servidor web bajo ataque recibirá conexiones entrantes confusas Aunque los ataque peer to peer son fáciles de identificar con firmas, el gran número de direcciones IP que se deben bloquear (a menudo más de 250000 en el curso de un gran ataque) significa que este tipo de ataque puede sobrepasar las defensas del firewall Incluso si un firewall pueden mantener el bloqueo de direcciones IP, existen otros problemas a considerar. Por ejemplo, hay un breve momento en que la conexión se abre en el lado del servidor antes de que la firma del ataque llega a identificarse. Sólo una vez que se abre la conexión con el servidor puede ser identificada la firma y bloqueada la conexión. Incluso el bloqueo de las conexiones puede agotar los recursos del servidor.

Ataque banana

Se trata de reorientar los mensajes enviados desde el cliente de vuelta hacia el mismo, evitando la entrada de paquetes de afuera de la red, inundando el host con

[3] Botnet es un término usado para designar una colección de programas robots “bots” los cuales pueden ser ejecutados de manera autónoma y en forma automática.

sus propios paquetes. Un atacante con acceso a un equipo de la víctima puede disminuir la velocidad de este hasta que esta sea inusable o usando una bomba de procesos (Chang, 2007).

Ataque zombie

Es un término que hace referencia a un ataque de denegación de servicio. Una red es objeto de hostilidad por diferentes atacantes haciendo ping a las computadoras durante un largo período de tiempo. El resultado es una degradación de la calidad de servicio y un incremento de la carga de trabajo para los recursos y la red. Este tipo de ataque es más difícil de detectar que los tradicionales ataques de denegación de servicio debido a su naturaleza encubierta (Chang, 2007).

Ataque Nuke

Nuke es un ataque de denegación de servicio contra las redes que consiste en el envío de paquetes ICMP fragmentados o paquetes ICMP inválidos esto se logra mediante una modificación a la utilidad ping que provoca él envió repetido de datos corruptos, provocando ralentizar la computadora afectada, hasta que llega a un alto total. En los juegos de azar en línea, es utilizado por vikingo difundiendo a otro usuario, o a todos los demás usuarios, con mensajes repetidos al azar en rápida sucesión. Dichas técnicas también se observan en los programas de mensajería instantánea, como en repetidas ocasiones el envío de texto se puede asignar a una macro o AppleScript (Chang, 2007).

Ataques distribuidos

Un ataque distribuido de denegación de servicio (DDoS) se produce cuando varios sistemas generan una inundación comprometiendo el ancho de banda o recursos de un sistema, por lo general uno o más servidores web; estos sistemas están comprometidos por atacantes usando una variedad de métodos. Un malware puede transportar mecanismos de ataques DDoS; uno de los ejemplos conocidos fue MyDoom, su mecanismo de ataque de fue activado en una fecha y hora específicas, este tipo de ataque tiene pregrabada la dirección IP de la víctima y no requiere mayor interacción una vez lanzado el ataque.

Un sistema también puede ser comprometido con un troyano, lo que permite al atacante descargar un agente zombie o bien el troyano puede contener uno. Los atacantes también pueden introducirse en los sistemas utilizando herramientas automatizadas que explotan las fallas en los programas que están a la escucha de las conexiones desde host remotos; este escenario se refiere principalmente a los sistemas que actúan como servidores de la web. Stacheldraht es un clásico ejemplo de una herramienta de DDoS. Utiliza una estructura de capas donde el atacante utiliza un programa cliente para conectarse a los manipuladores, que están en el sistema comprometido enviando los comandos al agente zombie, que a su vez facilita el ataque DDoS. Los agentes se comprometen a través de los manipuladores por el atacante, utilizando rutinas automatizadas para explotar las vulnerabilidades en los programas que aceptan conexiones remotas, cada manejador puede controlar hasta a un millar de agentes (Nong, 2006).

Herramientas DDoS como Stacheldraht todavía utilizan métodos de ataque DoS clásicos centrados alrededor de IP spoofing y amplificación como smurf, estos ataques también se conocen como ataques de consumo de ancho de banda. Inundaciones SYN también conocidos como ataques de consumo de recursos también pueden ser utilizadas. Nuevas herramientas pueden utilizar servidores DNS para fines de DoS. A diferencia de MyDoom los mecanismos DDoS, botnets se pueden dirigir contra cualquier dirección IP. Script kiddies usan para negar la disponibilidad de los sitios web bien conocidos a los usuarios legítimos. Más sofisticadas herramientas DDoS los atacantes hacen uso con fines de extorsión incluso en contra de sus rivales de negocios.

Cualquier ataque contra la disponibilidad sería clasificado como un ataque de denegación de servicio. Por otra parte, si un atacante utiliza mil zombis a la vez los sistemas de lanzamiento de ataques smurf contra un host remoto, éste sería clasificado como un ataque DDoS. Las principales ventajas para un atacante de la utilización de un ataque de denegación de servicio distribuido es que múltiples máquinas pueden generar más tráfico que un ataque de un solo equipo, a su vez múltiples máquinas de ataque son más difíciles de apagar que el ataque de una

sola, y que el comportamiento de cada terminal de ataque puede ocultarse mejor, lo que lo hace más difícil de detectar y evitar. Estas ventajas del atacante pueden causar problemas en los mecanismos de defensa. Por ejemplo, si se limita a la compra de más ancho de banda de entrada que el actual volumen del ataque, podría no ser una solución válida, ya que el atacante podría simplemente añadir más máquinas de ataque.

Ataque reflejado

Un ataque de denegación de servicio distribuido reflejado (DRDoS) incluye el envío de solicitudes de algún tipo a un gran número de ordenadores que responderán a las peticiones. Usando el protocolo Internet TCP y realizando spoofing de la dirección de origen de tal forma que sea la de la víctima, todas las repuestas estarán dirigidas hacia la víctima provocando la inundación. El ataque de solicitud de eco ICMP (Smurf Attack) puede considerarse una forma de ataque reflejado. El servidor atacante solicita ecos ICMP a la dirección de broadcast de una red mal configurándola realizando spoofing de la dirección de origen, provocando que los servidores respondan provocando una inundación. Algunos de los primeros programas de denegación de servicio distribuida aplicaban los métodos de este ataque, los servicios pueden ser explotados para actuar como reflectores, algunos más difíciles de bloquear que otros. Ataque de amplificación DNS un nuevo mecanismo que aumenta el efecto de amplificación, utilizando una lista mucho más amplia de los servidores DNS que ya se ha visto (Britos, 2010).

Ataques no intencionados

Ataques no intencionados se describen como una situación en la que en un sitio web se produce una denegación de servicio, no debido a un ataque deliberado por un solo individuo o grupo de individuos, sino simplemente debido a un súbito repunte en la popularidad enorme. Esto puede suceder cuando un sitio web muy popular pone enlace a un segundo sitio web, menos preparado, para recibir un gran número de peticiones, por ejemplo, como parte de una noticia. El resultado es que una proporción significativa de los principales usuarios del sitio ordinario potencialmente

cientos de miles de personas, que tienen el mismo efecto en la página web como un objetivo de ataque DDoS (Nong, 2006).

Sitios de noticias y los sitios de páginas de enlaces, sitios cuya función principal es proporcionar enlaces a otros lugares interesantes de contenido en Internet, son más susceptibles de causar este fenómeno. Los routers también se han conocido que pueden crear ataques de denegación de servicio no intencional, como D-Link y Netgear, Routers han creado “NTP vandalismo” por las inundaciones a servidores NTP sin respetar las restricciones de los tipos de clientes o limitaciones geográficas.

Ataques involuntarios similares también pueden ocurrir a través de otros medios, por ejemplo, cuando se menciona una dirección URL en la televisión. Si un servidor está siendo indexado por Google u otro motor de búsqueda durante los períodos de máxima actividad, o no tiene una gran cantidad de ancho de banda disponible mientras transcurre la indexación, también pueden experimentar los efectos de un ataque DoS (Nong, 2006).

3.2 SISTEMAS DETECTORES DE INTRUSOS

Las primeras investigaciones sobre detección de intrusos comienzan en 1980 en un trabajo de consultoría realizado para el gobierno norteamericano por James P. Anderson (Anderssen, 1980), quien trató de mejorar la complejidad de la auditoría y la habilidad para la vigilancia de sistemas informáticos. Es el primero que introduce el término amenaza en la seguridad informática, y lo define como la potencial posibilidad de un intento deliberado de acceso a la información, manipulación de la misma, o hacer que un sistema sea inutilizable. Anderson presentó la idea de que el comportamiento normal de un usuario podría caracterizarse mediante el análisis de su actividad en los registros de auditoría, de ese modo, los intentos de abusos podrían descubrirse detectando actividades anómalas que se desviaran significativamente de ese comportamiento normal.

Se puede definir intrusión como la violación de la política de seguridad de un sistema, o como la materialización de una amenaza. Luger (1990) define intrusión como cualquier conjunto de acciones que tratan de comprometer la integridad, confidencialidad o disponibilidad de un recurso. Una de las definiciones más aceptadas de intrusión es: fallo operacional maligno, inducido externamente (Stroud, 2001), aunque es bien sabido que muchas de las intrusiones proceden del interior del sistema de información. Finalmente, el NIST (National Institute of Standards and Technology) define detección de intrusos como el proceso de monitorización de eventos que suceden en un sistema informático o red y análisis de dichos eventos en busca de signos de intrusiones.

El primer modelo de detección de anomalías fue el propuesto por Dorothy Denning, con la idea básica de monitorear las operaciones estándares de un sistema objetivo, observando desviaciones en su uso (Denning, 1987). Su artículo provee un enmarque metodológico que más tarde inspiraría a muchos investigadores.

Definición

Los sistemas de detección de intrusos forman una parte importante dentro de las herramientas que son empleadas por la seguridad informática, para evitar que la información se vea comprometida (Gonzalez, 2003). Su función comprende básicamente la detección oportuna de las acciones ilegales o anómalas que indiquen la intrusión a una aplicación (software) o equipo de cómputo (hardware), de manera aislada o en red. Pudiendo efectuarse dicha intrusión desde el exterior o el interior de una red o segmento que derive de ella. La detección generalmente se basa en criterios preestablecidos que tratan de determinar qué es lo normal y lo anormal dentro del tráfico que fluye en la red, el comportamiento humano y la malformación de paquetes que se basan en la explotación de vulnerabilidades de los diversos protocolos de comunicación.

Forma de operar de los sistemas de detección de intrusos

Los IDS se encuentran integrados por diversos módulos que trabajan en forma conjunta y con funciones específicas para la recolección y análisis de datos de las actividades humanas o procesos que efectúa un sistema, así como la generación de alertas, y en algunos casos acciones de respuestas del tipo pasivo, activo o proactivo. El registro de los resultados y los datos que se obtienen se almacenan en bitácoras. Su motor de detección emplea distintos métodos de análisis, que pueden ser por ejemplo: estadísticos, de inteligencia artificial, sistema inmune, entre otros. Los cuales pueden operar de manera aislada o complementándose entre ellos, empleándose como criterios de discriminación de lo normal y lo anormal (Debar, 1999).

Estos mecanismos pueden ser desarrollados en hardware o software, cada uno con sus respectivas ventajas y desventajas. El primero es un equipo que se añade a la red, el cual requiere configuración de expertos, su principal ventaja es que no depende de un equipo de cómputo, sino de la robustez de los circuitos integrados y las partes que lo constituyen que son garantizados por el fabricante. El segundo se implementa para su operación dentro de un equipo de cómputo dedicado el cual dependerá en su totalidad del sistema operativo, que de manera adicional requiere de la configuración de una o varias tarjetas de red, así como las propias exigencias que se requiera del equipo de cómputo (memoria, espacio de almacenamiento, procesadores, etc). La ventaja en estos equipos radica en que pueden estar montados directamente sobre la aplicación a monitorear (Mira, 2009).

Justificación

Los sistemas de detección de intrusos son el complemento a otros elementos de defensa que pueden ser burlados por un atacante, como es el caso de los cortafuegos (firewall); esto se debe, a que los cortafuegos filtran el tráfico de la red con base en el análisis de sus encabezados y protocolos, y no analizan el detalle de cada paquete (Gonzalez, 2003). De esta forma, los IDS reciben los paquetes filtrados y reconocidos que provienen del firewall, para posteriormente analizarlos de acuerdo a criterios de firmas o anomalías, que son aplicados a su estructuración

o a su reensamblado. De esta manera se determina qué paquete es o no malicioso, y se dictamina si puede o no comprometer la seguridad de la información de un sólo equipo o de manera conjunta en todos los equipos que integran la red.

Sin embargo, hoy en día a los cortafuegos por ser la primera línea de defensa se les ha comenzado a adicionar la funcionalidad de los IDS. El objetivo es complementar su sistema de filtrado, pudiendo con ello reaccionar más eficiente y oportunamente ante un ataque hostil o en un intento de intrusión hacia la red interna. Esto es posible, porque se ha anexado una base de firmas que busca patrones dentro de los paquetes, es decir, se ha adoptado el concepto de inspección de paquetes en profundidad. Aunque cabe destacar, que por ser el primer filtro de seguridad, este no comprende un análisis en profundidad como lo haría un IDS, tomando en cuenta todos los elementos de información presentes en las bitácoras. Esta restricción no se basa en el hardware ni en el software, sino más bien, por el retardo que introduciría en la entrega de los paquetes; por lo que se podría decir que realiza una revisión rápida y toma las acciones que se le hayan indicado previamente, dejando la parte más profunda de la inspección a los IDS.

La búsqueda de un modelo único y mejorado

Desarrollar un modelo de un sistema de detección de intrusos requiere considerar los factores que lo integran y la vulnerabilidad que es inherente a éste, a través del uso de la terminología que denote la interacción con el entorno y la secuencia de pasos que describen el proceso de intrusión.

Prototipos con mejoras

Las siguientes propuestas van encaminadas al desarrollo de diversos modelos de detección de intrusos; las cuales han tratado de cubrir las siguientes expectativas (Zurutuza, 2004):

- La creación y/o utilización de un lenguaje único, flexible, portable y fácil de interpretar para la comunicación entre sus módulos.
- La elaboración de reportes de actividades en diferentes formatos (flexibilidad)
- Su simplicidad de uso.

- La descripción de los componentes y/o módulos que dictaminen una arquitectura a seguir.
- Monitoreo continuo y activación de las alarmas correspondientes presentadas ante indicios de intrusión hacia un sistema.

Las técnicas de detección no pueden ser generalizadas dentro de un modelo, esto se debe a que en la búsqueda de un mejor análisis y clasificación de la información que recibe, se ha propiciado un ambiente idóneo para la exploración de nuevas técnicas y el empleo de diferentes ramas científicas que pueden ser aplicadas a los sistemas de detección (Debar, 1999).

Modelo de Dorothy Denning

Este modelo explica mediante similitudes informáticas que es lo que representaría cada componente en la detección de una intrusión. Está enfocado directamente sobre el análisis de un sólo equipo y no de una red (Denning, 1987). El modelo está constituido por:

- Sujetos: Generalmente se asocia a los usuarios de un proceso, sistema o equipo de cómputo.
- Objetos: Son los dispositivos periféricos, procesos del sistema, dispositivos de almacenamiento, archivos, aplicaciones de cómputo, entre otros.
- Registro de auditoria: Es el registro de los sucesos que se obtienen de la interacción del sujeto sobre los objetos.
- Perfiles: Son los patrones de comportamiento que se establecen previamente en conjunto sobre la manipulación que realiza un sujeto sobre los objetos, siendo éstos la base que sustente los criterios de comportamiento normal o anormal dentro de un sistema.
- Registros de anomalías: Son las notificaciones que se tienen de las condiciones y uso sobre los objetos, así como la hora en que fueron realizadas dichas acciones con base a comportamientos anómalos o extraños.

- Reglas de actividad: Cuando se cumple la condición contenida en una regla se dispara una alerta, la cual es registrada en una bitácora con los siguientes rubros: evento, hora del evento y el perfil hallado (anomalía).

En esta propuesta se presenta como sistema al conjunto integrado por sujetos y objetos, donde su interacción es registrada y observada (almacenamiento de perfiles) en espera de anomalías, que al ser comparados con las reglas establecidas y validándose éstas, se traducirán como intrusión; efectuándose con ello las alertas pertinentes a través de reportes. Este modelo recibió el nombre de IDES que implementó un sistema experto (SE) como técnica de detección de intrusiones.

Common Intrusion Detection Framework (CIDF)

Otra propuesta para tratar de hallar un modelo de sistemas de detección de intrusos fue hecha por el CIDF. Ésta sugiere la utilización de GIDO (Generalizad Intrusión Detection Object) como componente de intercambio de datos entre los diferentes módulos y la utilización de un lenguaje para crear las reglas de detección (Mira, 2009). La arquitectura está integrada por 4 módulos:

- **Generadores de Eventos**: Integrados por receptores que están a la escucha de los eventos que ocurren dentro de una red o en un host específico.
- **Analizadores de Eventos**: Son los encargados de recibir la información que es enviada por los generadores de eventos y procesarla mediante diversas técnicas; detectando si se presenta o no una intrusión de acuerdo a los criterios previos de abuso o comportamiento anómalo establecidos.
- **Base de Datos**: Está compuesta por los patrones almacenados previamente que dan la indicación de una posible intrusión.
- **Unidades de Respuesta**: Son las acciones a tomar en el momento que se detecta una intrusión.

Componentes básicos de un IDS

De la propuesta de estos modelos se puede obtener un esquema genérico que permita describir de manera general las funciones que debe cumplir un sistema de

detección de intrusos. De tal forma, que las partes básicas que integren la arquitectura de un IDS como se observa en la figura 1, sean las siguientes:

- Sensores: Serán los recolectores o receptores de la información que fluye a través de una red o en un host específico.
- Analizadores: Son el corazón de un IDS. Descomponen en pequeños fragmentos la información que reciben de los sensores, en búsqueda de comportamientos anómalos o de abusos, que pueden realizarse sobre un sistema y forma parte del motor de inferencia.
- Motor de Inferencia: Está constituido por los analizadores y las reglas que contienen las especificaciones de comportamientos de intrusión, lo que le permite aplicar criterios para catalogar la información que recibe, en términos de normal o anormal durante la fase de análisis.
- Acciones de Respuesta: Pueden ser: Pasivas, Activas o Proactivas.
    - Las respuestas pasivas, son aquellas que notifican el suceso de intrusión al administrador y esperan la respuesta por parte de él. Es decir, requiere intervención humana.
    - Las respuestas activas, toman las decisiones que se les haya indicado previamente, como pueden ser finalizar conexiones, reconfiguración de cortafuegos, bloqueo de direcciones IP, entre otras.
    - Las respuestas proactivas emplean el concepto del cómputo proactivo, es decir, la anticipación de una acción basada en lo que percibe del medio físico que se le va presentando.
- Registros: Se consideran las bitácoras y reportes del sistema sobre las anomalías halladas en el interior de un sistema o hacia él.

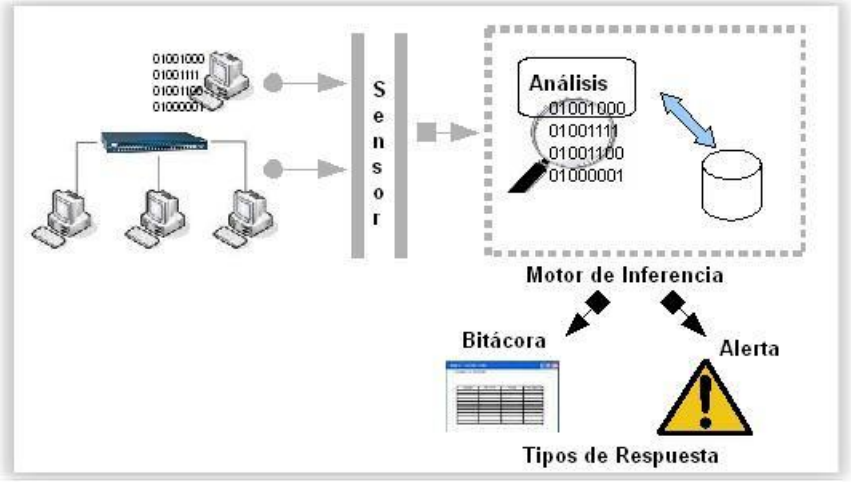

Figura 1. Componentes de los IDS.

Requisitos de un IDS

Las propuestas a desarrollar para los sistemas de detección de intrusos deben de tratar de ser una solución lo más integral posible. No sólo en lo referente a una arquitectura estándar y la descripción del funcionamiento de los componentes que la integran, sino el considerar otros aspectos al momento de efectuar su diseño, tales como (McHugh, 2000):

- Escalable, que pueda ampliar su panorama de trabajo.
- Fácil de configurar, es decir que permita realizar ajustes de una forma ágil.
- Interoperabilidad e interconexión entre fabricantes (protocolos, lenguajes).
- Análisis de los datos capturados en tiempo real.
- Espacio de almacenamiento para bitácoras y la optimización de las mismas.
- Revisión de bitácoras en tiempo real, que conduzcan a un patrón previo de intrusión.
- Confiable, es decir, minimizar el número de falsas alarmas y el paso de información con apariencia normal cuando ésta en realidad no lo es (ataque desconocido).
- Actualización manual o automática (preferentemente) de las bases de datos de firmas o de los umbrales a emplear en los criterios de evaluación.
- Redundancia y Tolerancia a Fallos.
- Velocidad de respuesta a intrusiones en tiempo real (Alertas y toma de acciones de reconfiguración de componentes)

- Reportes flexibles (configurables) sobre ataques y estadísticas de intrusión.
- Respuesta a intrusiones a través de autoaprendizaje y en algunos casos de autoconfiguración para repeler nuevos intentos de intrusión.
- Comparación con otros modelos que empleen la misma técnica u otra metodología para resolver los indicios de intrusión.

Taxonomía de los IDS

Durante el desarrollo de los sistemas de detección de intrusos han surgido diferentes clasificaciones de acuerdo a la técnica de detección que es utilizada por éstos, el tipo de respuesta que ofrecen ante una intrusión, los sistemas operativos sobre los que pueden funcionar, aplicaciones específicas, entre otros. Sin embargo, esas clasificaciones pueden considerarse como subdivisiones de dos grandes grupos, puesto que, en el caso de las técnicas de detección, éstas son derivadas de los métodos de detección que se empleen para buscar una intrusión dentro de un sistema, y estos son: red o host (Axelsson, 2006). Los IDS se clasifican por:

- Punto de Detección
  - Host
  - Red
- Método de Detección
  - Anomalías
  - Abusos

IDS por Punto de Detección

Esta clasificación hace referencia a la ubicación en la que son colocados los sistemas de detección dentro de un sistema.

- IDS tipo Host: Son sistemas de detección que se colocan en un equipo de cómputo (o en varios equipos) debido a la información sensible que contienen, y la cual se considera requieren de mayor atención de monitoreo. Su función consiste en registrar dentro de diferentes bitácoras, los movimientos realizados por transacciones internas, accesos remotos y/o locales, modificación de permisos a usuarios, grupos, archivos o carpetas,

cambios de contraseñas, procesos, entre otros (Bouhola, 2004). El análisis de detección de este tipo de sistemas se concentra en las bitácoras que contienen la información capturada, para ser posteriormente revisada en forma minuciosa, en busca de anomalías o abusos perpetrados dentro de un sistema y/o equipo (Zurutuza, 2004). La mayor ventaja que presentan es que permiten tener un registro detallado de todas las actividades que suceden dentro de él, es decir, mayor granularidad (nivel de detalle o profundidad) de análisis. Sin embargo, presentan una gran debilidad; son susceptibles a los ataques de Denegación de Servicio (DoS). Ante los ataques de DoS no se tiene defensa, debido a que el ataque no puede ser detectado con anticipación, puesto que este tipo de sistemas no analiza los paquetes que se envían en la red; sino éste es víctima cuando los desempaqueta y al convertir en tramas para ver su contenido, el ataque ya no puede detenerse en su ejecución.

- IDS tipo Red: Su análisis no está basado en las bitácoras que genera, sino en los protocolos de comunicación. Este sistema de detección captura los paquetes que circulan en la red y busca en el interior de cada uno de ellos, las anomalías que no correspondan con la funcionalidad y la aplicación que indican los RFC (Bouhola, 2004). Al igual que el IDS de Host utiliza bitácoras para registrar por protocolos los intentos de intrusión que se encontraron en el interior de los paquetes. La implementación de un IDS de tipo red dependerá de la arquitectura y topología de la red, junto con los dispositivos de red que la constituyan; como pueden ser: un hub o un switch (Díaz, 2008). El primero se basa en el concepto de dominio de colisión, en donde se refleja en cualquier punto de la red lo que transita sobre ella, es decir, se puede colocar un IDS en cualquier punto de la red y recibir la información para su detección. En el segundo caso el tráfico se ve aislado, y es necesario configurar un puerto dentro del switch conocido como SPAN Port (Switch Port Analizer) y conectar sobre él el IDS o un TAP (Test Access Point), para la captura del tráfico que circula en un segmento de red y a ese mismo dispositivo conectar el IDS (Laing, 2009). Su principal ventaja, es el análisis

masivo en un rango de red o en todos los equipos de una red, además de ser casi imperceptibles a los atacantes.

IDS por Método de Detección

Los sistemas de detección de intrusos de tipo host o tipo red, emplean de manera indistinta dos principales metodologías para detectar la presencia de un intruso en un sistema, ya sea de manera aislada o en complemento una de la otra.

- Anomalías: Busca perfiles de comportamientos diferentes a los que tiene almacenados, tales como actividades extrañas por parte de los usuarios, errores de tecleo de contraseñas, uso de sistemas en horarios diferentes a los acostumbrados, integridad de archivos, etc. Se basa en el conocimiento previo que le defina lo normal y anormal, y con ello poder efectuar su análisis. Su gran desventaja se encuentra en que todo lo que no esté dentro de sus parámetros normales es considerado una intrusión, lo que puede generar un alto número de falsos positivos. Su método de detección se basa en lo que conoce y lo que esté fuera del rango es considerado intrusivo (Zurutuza, 2004).
- Abusos: No requiere un entrenamiento previo para diagnosticar una intrusión. Su análisis se basa en la comparación de los patrones o firmas que indican que puede efectuarse un intento de penetración o es definitivamente una intrusión. No genera tantos falsos positivos, debido a que si no está dentro del patrón significa que es normal. Su desventaja radica en que lo que no esté dentro de sus patrones no será reconocido como un ataque, aunque en realidad este sí lo sea. La metodología de detección dice que lo que no se encuentra dentro de su base de datos es permitido, es decir, no es intrusivo. Es importante que la base de datos se actualice con frecuencia de forma manual o autónoma, para aminorar el acceso a un ataque potencial desconocido (Mira, 2009).

Credibilidad y Confiabilidad

Los sistemas de detección de intrusos recaudan la información que circula en un sistema o en una red, para posteriormente enfrentarse a la disyuntiva de etiquetar lo que se considerará anómalo o intrusivo, de lo que no lo es. Se utilizan diferentes técnicas de detección que permitan obtener una evaluación lo más precisa posible y determinar si es o no una intrusión al sistema, las cuales pueden ser empleadas de forma aislada o combinadas entre sí, para mejorar su análisis en tiempo o en precisión.

Un ejemplo de esta disyuntiva podría ser un paquete que circula a través de la red con todas las banderas activas, lo que generaría una alarma automáticamente indicando que se ha encontrado un paquete con características diferentes a las establecidas dentro de los RFC (Request for comments, documentos sujetos a revisión), esto sería catalogado como intrusivo, puesto que alguien está elaborando paquetes mal formados para distracción y luego acceder a un sistema. Sin embargo, a la clasificación que se refiere, es algo más complicado, puesto que, puede tratarse de un paquete que tiene las banderas correctas, en apariencia la secuencia de paquetes es correcta, pero en su interior fragmentos de código malicioso, que al ser reensamblados, se convertirán en un ataque directo hacia un sistema (Gorton, 2010). En ese caso, es donde se muestra la verdadera capacidad de análisis de un sistema de detección, para ir revisando por medio de patrones (firmas) o umbrales, las anomalías y abusos que se realizan para acceder hacia un sistema, y no confundirse con el criterio de que es un paquete no intrusivo.

La fiabilidad de un IDS se basa fundamentalmente en la efectividad que tiene para el análisis y detección de intrusiones dentro de un sistema (Díaz, 2008). Se requiere que un IDS pase por pruebas de penetración de ataques conocidos y ataques que no se hayan registrado previamente ante un sistema, los cuales son creados por medio de herramientas específicas para tal efecto (nessus, por ejemplo), de igual forma el acreditar las pruebas de técnicas de evasión. Esto es con el objetivo de conocer el nivel de seguridad que se puede obtener de dicho

sistema de detección y la confianza que se puede tener en él ante las amenazas de intrusión.

Métricas

El desempeño que los sistemas de detección de intrusos presentan al momento de detectar una intrusión o la posibilidad de que ésta ocurra dentro de un sistema requiere de ser medido; puesto que es necesario conocer la fiabilidad que se puede tener sobre éstos. Para ello, se han establecido parámetros que permiten expresar los criterios de selección indicados previamente por el dueño del sistema, y de esta manera obtener como resultado paquetes en los que se confía que su contenido no es malicioso y/o intrusivo. Los parámetros que dan la fiabilidad a un IDS, son (Gorton, 2010):

- Falso Positivo (No intrusivas pero anómalas): Este indicador registra que los paquetes de información presentan anomalías en su construcción de acuerdo con los RFC que avalan su operatividad o su comportamiento es diferente al reportado por el dueño de la aplicación. Por ejemplo, tamaño del paquete o usuario nuevo en una aplicación.
- Falso Negativo (Intrusivas pero no anómalas): Indica que los paquetes de información no presentan ninguna anomalía en su fabricación, pero contienen código malicioso que compromete la seguridad, por ejemplo, los paquetes fragmentados; estos datos son catalogados como datos correctos, cuando en realidad son malignos.
- Verdadero Positivo (Intrusiva y anómala): Reporta que lo que se ha catalogado como Intrusión es verdaderamente una intrusión o un intento de ataque hacia un sistema.
- Verdadero Negativo (Ni intrusiva ni anómala): Señala que los datos que se han analizado y no fueron descartados o mandados a cuarentena, son libres de intrusión y/o anomalía. Interpretación de los parámetros
- Falso Positivo (Falsas Alarmas): Este parámetro es muy importante y delicado para la medición de un IDS, puesto que puede indicar erróneamente la presencia de intrusión cuando en realidad no existe. Esto se debe a que,

si existe un valor muy alto en este parámetro se generan falsas alarmas de intrusión y/o ataques, lo que hace que la credibilidad y confiabilidad sobre un IDS pueda ser prácticamente nula.

- Falso Negativo (Alarmas no detonadas): Cuando el sistema de detección realiza la selección de lo que es intrusivo o no lo es, al detectar que no existe alguna anomalía en su interior o un patrón intrusivo, la información se descarta como dañina y se torna como fidedigna. Si el número es bajo puede indicar una mala detección, es decir, indicará que la información captada es muy confiable, cuando en realidad la información lleva intenciones ocultas para irrumpir en un sistema. Esto es, el IDS no detecta nuevos ataques o variaciones de éstos que pasan desapercibidos por no tener conocimiento sobre ellos.
- Verdadero Positivo: Ratifica que los resultados entregados por el IDS como intrusivos, sí son intrusivos.
- Verdadero Negativo: Confirma que los resultados que el IDS entrega como información no intrusiva ni anómala, es correcta.

Evaluación

La forma de evaluar un sistema de detección es mediante la implementación de dos o más sistemas de detección de intrusos en diferentes puntos de un segmento de red o en varios hosts que contengan el sistema a monitorear; estos IDS son inicializados al mismo tiempo para que comiencen la captura y analicen por medio de sus diferentes técnicas de detección la información que ha sido recaudada. Una vez efectuado esto, se procede a revisar los resultados arrojados. En esta disertación, se considerará como el mejor IDS al que presente los mejores resultados bajo ciertos ámbitos, por ejemplo: emplear menos recursos (hardware), permitir la creación de reportes básicos y personalizados a las necesidades del propietario del sistema (flexibilidad), presentar el menor número de falsas alarmas indicando que algún dato es intrusivo cuando no lo es, así como el mayor número de aciertos que acrediten que la información que se deje pasar es confiable y no

intrusiva. Las técnicas que por lo general se utilizan para tal efecto (Díaz, 2008), son las siguientes:

- Comparación: Esta técnica se emplea generalmente cuando se desea saber el desempeño que tendrá un nuevo prototipo contra uno del tipo comercial, de igual modo es utilizada para seleccionar el mejor IDS de tipo comercial que se adecue a las necesidades personales o empresariales sobre un sistema a proteger.
- Efectividad: Los resultados que se pueden obtener de la técnica de comparación, dan a conocer al IDS que presentó mejor desempeño en la captura masiva de información, así como en las técnicas de detección, entre otros. Sin embargo, la pregunta es: ¿cómo saber si efectivamente está detectando de forma correcta? Para ello se emplea la técnica de efectividad, la cual consiste en implementar en el mismo segmento de red o en el mismo equipo a monitorear un sniffer (herramienta que escucha todo lo que pasa alrededor y dentro de él).
- Penetración: Existen diversas herramientas para penetrar en un sistema, las cuales contienen utilerías para emplear paquetes de red malformados, ataques conocidos o que permiten la creación de éstos con ciertas variaciones. Estas herramientas pueden simular desde ataques básicos hasta ataques sumamente agresivos que pueden suspender el servicio de un equipo.
- La evaluación consiste en el reconocimiento de ataques que sean o no conocidos y se demuestre el mejor desempeño para adaptarse a las variaciones en el reconocimiento de éstos, obteniendo el nivel de seguridad que se tendrá ante una intrusión potencial o baja. Generalmente estas pruebas se emplean para evaluar que tan protegido se encontrará un sistema que se encuentre en una red, en un equipo o dentro de una aplicación específica.

Métodos de Detección

Las formas que se han aplicado para descubrir la presencia de un intruso o la intención de introducirse a un sistema, son (Barker, 2007):

- Anomalías (perfiles): Lo que está explícitamente prohibido, no está permitido. Este método busca comportamientos diferentes a los que tiene registrados, así como las variaciones que éstos puedan presentar. Su forma de detección se basa en el conocimiento previo de lo que se considera normal y lo que no lo es; estos parámetros pueden ser adquiridos por medio de entrenamiento o de umbrales preestablecidos, que son adoptados como perfiles de comportamiento. Presenta una gran desventaja al momento de realizar su análisis y catalogar la información recibida, la razón es que puede generar un alto índice de falsas alarmas (falsos positivos), debido a que lo que no se encuentre reportado como normal, automáticamente lo considerará como intento de posible intrusión. Otra de sus desventajas es que requiere de estar en constante mantenimiento para aminorar el porcentaje de falsos positivos y de esa manera no afectar la confiabilidad del sistema de detección de intrusos en cuestión. Sin embargo, su mayor complejidad radica en que definir que es normal y anormal, puesto que estos conceptos no pueden estandarizarse para todo sistema; cada uno requerirá diferente interpretación a criterio del propietario. Por citar algunos ejemplos de búsqueda, se pueden señalar actividades extrañas por parte del usuario de un sistema o del sistema en sí, tales como: errores de tecleo de contraseñas (error de autenticación), uso de sistemas en horarios diferentes a los habituales, modificación de la información, etc.
- Abusos o uso indebido (patrones o firmas): Lo que no está explícitamente prohibido, está permitido. A diferencia del anterior, este método no requiere un entrenamiento previo para diferenciar que es lo normal o anormal, y con ello diagnosticar la existencia o intento de una intrusión. Su análisis se fundamenta en la comparación de los posibles eventos (patrones) o características únicas y propias (firmas) que determinen la presencia de una intrusión. Cada patrón es una secuencia de los posibles pasos que puede

seguir un atacante, así como las variaciones de éstos. Las firmas contienen indicios de ataques conocidos que al ser cotejados, verifican si se cumple o no la condición establecida. No genera un gran número de falsos positivos, esto se debe a que lo que no coincide con sus patrones o firmas es considerado como normal. Sin embargo, a su vez este comportamiento es su mayor desventaja, puesto que puede generar un número significativo de falsos negativos al momento de marcar la información como ausente de técnicas hostiles, cuando en realidad es un ataque. Presenta otra desventaja, requiere mantener actualizado su contenedor de firmas o patrones (base de datos) de forma manual o autónoma, para evitar caer en un número alto de falsos negativos, de lo contrario cualquier acción dañina no reconocida, afectará al sistema que se desea proteger.

Modelos de detección

Se han creado diferentes prototipos para representar diversas teorías sobre la forma de detectar intrusos dentro de un sistema. En ellos se expresan conceptos que permiten esquematizar comportamientos y/o patrones, que permitan facilitar la comprensión de las características que describan un ataque o intrusión. De tal forma, que se pueda inferir sobre ellos el conocimiento necesario para prevenir o retardar actos ilegales antes de que sean perpetrados dentro de un sistema (Gonzalez, 2003). Los arquetipos más desarrollados para la detección de intrusos que se han observado durante esta investigación, se basan en:

- Modelos: Representan a un sistema complejo con el objetivo de facilitar su comprensión y comportamiento, de tal forma, que se puede reconocer nuevos tipos de ataque a través del diseño y análisis de operación de éstos. En ellos se reproducen diversos eventos que se presentan en un problema actual o que todavía no se han presentado dentro de un sistema, es decir, plantean la posibilidad de que suceda un evento y de esta forma adquirir el conocimiento para anticiparse a una nueva técnica de intrusión (Zhicai, 2004).

- Firmas: Su objetivo es comparar el contenido de cada paquete con una base de datos previamente enriquecida con las características específicas (firmas) que identifican una intrusión o intento de acceso ilegal hacia un sistema (ataque) (Somer, 2003).
- Patrones: Analizan los datos obtenidos a través de un sensor y se cotejan contra umbrales de comportamiento previamente establecidos del tipo humano, aplicación, procesos del sistema, entre otros (Chen, 2007).
- Clasificadores: Examinan el conjunto de datos adquiridos (en tiempo real o fuera de ese periodo) designando cuales son de carácter maligno y los que son inofensivos. Una vez que es determinada el tipo de información que se trata, ésta es pasada nuevamente por filtros que corroboran a mayor profundidad la clasificación previamente hecha (Barker, 2007).
- Auto-aprendizaje: Se basan en el conocimiento previamente adquirido, el cual se emplea para reconocer los diferentes intentos de intrusión y los que no conoce los procesa para inferir un nuevo conocimiento, del manera, que la siguiente vez que realice su análisis de detección, pueda reconocer un mayor número de técnicas de intrusión y de evasión que son empleadas por los intrusos sobre los sistemas de detección (Grediaga, 2002).

Técnicas

Los sistemas de detección de intrusos tienen como objetivo el detectar si existe o no una intrusión dentro de un sistema. Para ello, se utilizan criterios de análisis basados en abusos y en comportamientos anómalos (patrones y perfiles, respectivamente). Esto es posible efectuar, por medio del uso de diferentes áreas de investigación, de las cuales se pueden citar: la Inteligencia Artificial, Métodos Estadísticos, Redes Neuronales, Minería de datos, entre otras. Cabe aclarar que las técnicas empleadas en una detección no están directamente enfocadas a un tipo de IDS, sino que se ocupan de manera aislada o conjunta, en base a su flexibilidad y potencialidad para detectar intrusos (Alessandri, 2004). Algunos modelos que han sido propuestos por diversas universidades han combinado técnicas para obtener mejores resultados en la búsqueda de intrusos dentro de un sistema.

- Agentes Móviles: Los agentes son una entidad que actúa de manera autónoma, pero en colaboración con otros agentes para detectar una intrusión en un sistema. El agente obtiene información que permita reconocer la presencia de un intruso y la envía a un motor de análisis, quién dictamina si existe o no una intrusión. Si se halla algo importante es comunicado a los otros agentes para tomar las medidas pertinentes (Foukia, 2005).
- Algoritmos Genéticos: En la actualidad se empieza a incursionar en esta área para detectar una intrusión. Debido a que su aplicabilidad al concepto de evolución biológica, permite emplearse como clasificador de lo bueno o malo de la información que fluye en una sistema (Barker, 2007).
- Árboles de decisión: Estos permiten modelar un proceso de tomas de decisiones sobre lo que se considera normal o anormal en un sistema. Los nodos de los arboles representan la disyuntiva y los arcos las alternativas (Li, 2005).
- Escenarios: Se desarrollan modelos con las posibles técnicas de evasión, ataques y comportamientos extraños que permitan anticipar la llegada de un intruso a un sistema (Zhang, 2006).
- Grafos: Es la representación gráfica de una intrusión o ataque específico por medio de nodos, que muestran el comportamiento de un sistema ante un acto hostil hacia el mismo (Gonzalez, 2003).
- Lógica de Predicados: Esta técnica es un sistema deductivo formal, que utiliza predicados, conectores lógicos y cuantificadores para inferir conocimiento a partir de ataques conocidos (Joshi, 2005).
- Lógica Difusa: Sistema deductivo formal que utiliza criterios flexibles (valores entre 0 y 1) de verdad (Lucas, 2007).
- Máquinas de Estado: Se emplean para modelar el comportamiento de un sistema y descubrir nuevos tipos de ataques. El comportamiento de un sistema es representado a través del cambio de estados (nodos), los cuales se presenta cuando ocurre una acción que indique la transición. Los estados no sólo dependerán de sus entradas actuales, sino también de los anteriores,

para formar los antecedentes que expliquen el comportamiento actual (Gonzalez, 2003).

- Máquinas de Soporte Vectorial: Es un conjunto de métodos de aprendizaje supervisado, que se basan en la separación lineal de los datos de entrada. Si la distancia es corta, se identifica como una posible intrusión (Zurutuza, 2004).
- Métodos Estadísticos: Esta técnica reacciona a umbrales establecidos previamente sobre los parámetros a evaluar en un sistema, por ejemplo: comportamientos humanos, de procesos, de servicios, etc. Si estos valores son excedidos, se considera que es una intrusión hacia un sistema (Wagner, 2006).
- Minería de Datos: Esta técnica permite extraer patrones o modelos de ataques desconocidos, de un conjunto de datos recolectados por un IDS (Fan, 2005).
- Modelo de Markov: Es un modelo que representa la probabilidad de que un estado pueda pasar de un estado actual a otro, es decir, es una probabilidad condicionada, en el que el nuevo estado depende totalmente del estado anterior. En la detección de intrusos se emplean las cadenas de Markov para representar la transición entre eventos condicionados y determinar la existencia de una intrusión. También se han empleado los modelos ocultos de Markov (HMM) para conocer el estado anterior, cuando sólo se conoce el estado actual (Soto, 2005)
- Ontologías: Permiten realizar una representación formal de un conjunto de conceptos (ataques) y sus relaciones sobre un dominio (Tseng, 2004).
- Reconocimiento de Patrones: Se basa en las formas o patrones conocidos de ataques e intrusiones, comparan cadenas de texto que vienen en el contenido de un paquete, y/o anomalías en las cabeceras de los protocolos de comunicación (Sommer, 2005).
- Redes Bayesianas: Se emplean como modelos gráficos para representar la dependencia entre un conjunto de variables, a través de datos probabilísticos

que indiquen la probabilidad de que un evento hallado sea una intrusión (Wagner, 2006).

- Redes de Petri: Son utilizados para la representación gráfica de eventos que pueden presentarse en una intrusión. La transición entre estados sucede cuando se cumple el evento. Esta técnica permite modelar ataques complejos, en los que se incluyen sus características particulares de comportamiento (Mira, 2009).
- Redes Neuronales: Es una área que está incursionando al igual que otras áreas como los Algoritmos Genéticos y el Sistema Inmune, en el campo de la detección de intrusos. La red neuronal es entrenada con compartimento normales o anormales (estos valores dependen de la forma en que se desee detectar una intrusión). Mediante su empleo es posible detectar variaciones de ataques o de carácter desconocidos, que difieren de los patrones iniciales con que fue entrenada la red (Zanero, 2008).
- Sistemas Expertos: Conjunto de reglas con la estructura IF-THEN-ELSE, en la que si se cumple la regla la intrusión o ataque a buscar, es confirmado (Hu, 2006).
- Métodos Heurísticos: Emplea el resultado que es generalmente obtenido a través de algún método estadístico, para ajustar un umbral de detección de lo normal y anormal que se presenta en un sistema. Tratando con ello, de aminorar el número de falsos positivos y negativos en un IDS (Gonzalez, 2003).

Localización de un IDS

Los sistemas de detección de intrusos pueden ser implementados en diferentes puntos de una red de cómputo o en un equipo específico tal como lo muestra la figura 2, los cuales pueden operar de manera conjunta o aislada sobre un sistema. Cada punto en el que se ubique un IDS presenta ventajas y desventajas con respecto al daño potencial al que se ven expuestos a enfrentar, así como el nivel de protección que pueden brindar en cada punto.

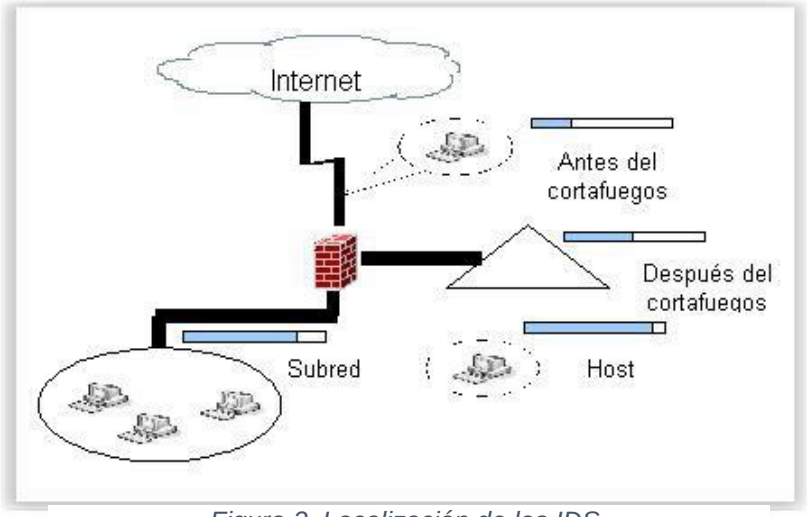

Figura 2. Localización de los IDS.

Los sitios que generalmente se contemplan para su implementación (Díaz, 2008), son: Antes del cortafuegos, después del cortafuegos, en una subred o red única y en un solo equipo.

Antes del Cortafuegos

El IDS se implementa con la ideología de analizar los paquetes que provienen de la red externa hacia la red interna. Se ubica entre la salida hacia Internet y el cortafuegos, esto permite realizar un análisis masivo en la intercepción de paquetes o solicitudes hacia los sistemas de aplicaciones que se encuentran dentro de los servidores y a los servicios que brindan. Un IDS en ese punto requiere de una gran capacidad para el manejo del tráfico entrante. Presenta los inconvenientes de que su detección en cierto modo no puede ser profunda, en virtud de que este tipo de análisis retrasaría la entrega de paquetes hacia la red interna. Asimismo, pueden presentarse problemas de pérdida de paquetes al no ser capturados en su totalidad por el IDS, debido a la carga (cantidad de información) de la red (Dreger, 2006).

Después del Cortafuegos

El cortafuegos es un componente de control que se emplea para filtrar la comunicación que existe entre una red externa y una red interna. Dentro de sus funciones se encuentra el indicar qué protocolos de comunicación son los que se permitirán acceder desde el exterior hacia la red interna y viceversa, controlar los

puertos entrantes y salientes, las direcciones IP o MAC que pueden tener intercambio con el exterior, entre otras. Actualmente los cortafuegos han evolucionado y en algunos casos han integrado módulos de análisis de paquetes y técnicas anti-evasión en ausencia de un IDS o simplemente como valor agregado a la seguridad del sistema. Los cortafuegos no están faltos de que puedan ser burlados por los intrusos por medio de técnicas de evasión y éstos puedan acceder de manera legal al interior de la red. Esto es posible por medio de la utilización de un puerto que es permitido (puerto abierto) para acceder a la red interna, a dicho puerto se envía un paquete malformado (paquete alterado por un atacante) que evade sigilosamente la detección del IDS haciendo pasar éste como un paquete sin intenciones ocultas.

En una subred o red única

En este punto el IDS es colocado como parte de una subred o de toda la red (si es que no existiesen subredes). El nivel de detección que se configura es alto, puesto que jerárquicamente es el penúltimo o dependiendo del factor riesgo a enfrentar puede ser el único eslabón de la cadena de seguridad. El IDS se encarga de detectar de manera exhaustiva el tráfico que circula por la red o subred interna en busca de indicios de intrusión que hayan burlado al cortafuegos o que afecten directamente a las aplicaciones que se ejecutan sobre los sistemas de cómputo. El análisis exhaustivo puede ser desde la revisión de cada paquete que circula sobre la red, hasta los intentos de autenticación ante un sistema, modificación de archivos, lanzar ataques hacia otras redes desde la red interna, etc.

En un sólo equipo

La implementación de un IDS en este punto, depende exclusivamente del criterio del propietario de un sistema, con base a la exposición de vulnerabilidad que se desee evitar. Este IDS es de tipo Host que como ya se mencionó con anterioridad, permite una detección minuciosa sobre el sistema operativo, los procesos del sistema, la integridad de los archivos (modificación), los intentos de autenticación, etc. Su nivel de detección en este punto es de medio-alto. La implementación consiste en que el IDS forme parte del equipo que se desea inspeccionar.

3.3 REDES NEURONALES

El construir una computadora que sea capaz de aprender, y de entender el significado de las formas en imágenes visuales, o incluso distinguir entre distintas clases de objetos similares son parte de la problemática a la que se enfrentan los que diseñan computadoras, los ingenieros y los programadores (Freeman, 2000).

La incapacidad de la generación actual de computadoras para interpretar el mundo en general no indica, sin embargo, que éstas sean completamente inadecuadas. Generalmente los problemas se presentan cuando se trata de resolver problemas que involucran un procesamiento en paralelo, utilizando una herramienta de tipo secuencial, la computadora. Uno de los problemas que implican un tipo de procesamiento como el mencionado anteriormente, es el reconocimiento visual de imágenes. Para una computadora el reconocer imágenes aún muy diferentes, es una tarea sumamente difícil, lo que en el caso de los humanos es algo relativamente sencillo. Esto se debe a que los sistemas biológicos poseen una arquitectura distinta (paralelismo masivo) a la de una computadora moderna. Por esta razón es que se han tratado de simular algunas características de la fisiología del cerebro humano para elaborar nuevos procesos de procesamiento (Freeman, 2000).

Se puede definir a una Red Neuronal Artificial como un modelo matemático inspirado en sistemas biológicos, adaptados y simulados en computadoras convencionales (Lara, 2002). Las RNAs están inspiradas en el sistema biológico natural. Como es conocido, en este sistema la neurona es la unidad de procesamiento, y aunque las RNAs sean mucho menos complejas que una red neuronal biológica, también realizan cálculos complejos para procesar información.

La computación convencional y la biológica

La computación convencional se caracteriza por el desarrollo de una formación matemática del problema, el desarrollo de un algoritmo para implementar una solución, la codificación del mismo para una máquina específica y por último la

ejecución de dicho código. Como se ha observado, este tipo de procesamiento es muy exitoso para resolver modelos matemáticos complejos y de simulación, para realizar tareas repetitivas, rápidas y bien definidas. Sin embargo, cuando éste se lleva a otros ámbitos computacionales, se muestra incapaz de resolver eficientemente problemas de reconocimiento de imágenes, de voz, y de entendimiento de lenguaje natural. También resulta ineficiente en problemas de percepción, adaptación y aprendizaje (Lara, 2002).

Por otro lado, la computación biológica (derivada del procesamiento en sistemas biológicos) se caracteriza por ser masivamente paralela, adaptativa, lenta, altamente interconectada y tolerante al ruido en el medio ambiente y en sus componentes. De acuerdo con la prensa, publicaciones, y conferencias, las redes neuronales (como parte de la computación biológica) han tenido aplicaciones en el área de seguridad informática y de clasificación (Schalkoff, 2003), específicamente en el análisis de reconocimiento de patrones.

Descripción

Una Red Neuronal Artificial es una estructura compuesta de un número de unidades interconectadas (neuronas artificiales). Cada unidad posee una característica de entrada/salida e implementa una computación local o función. La salida de cualquier unidad está determinada por su característica de entrada/salida, su interconexión con otras unidades, y (posiblemente) de sus entradas externas. Sin embargo es posible un “trabajo a mano”, la red desarrolla usualmente una funcionalidad general a través de una o más formas de entrenamiento (Schalkoff, 2003).

El cerebro humano contiene más de 100 billones de elementos de procesos llamados neuronas, que se comunican a través de conexiones llamadas sinapsis. Cada neurona está compuesta por tres partes fundamentales: el cuerpo, dendritas y axón. El cuerpo en su capa externa tiene la capacidad única de generar impulsos nerviosos. Las dendritas que son como las ramas que salen del cuerpo, poseen algunas conexiones sinápticas en donde se reciben señales que generalmente vienen de otros axones. El axón se encarga de activar o inhibir otras neuronas las

cuales a su vez son activadas por cientos o miles de otras neuronas. El funcionamiento de una neurona artificial está basado en este diseño. Básicamente consiste en aplicar un conjunto de entradas, cada una representando la salida de otra neurona, o una entrada del medio externo, realizar una suma ponderada con estos valores y "filtrar" este valor con una función como se puede observar en la figura 3 en donde se muestra que el resultado de la suma de todas las entradas pasan por una función que determinara la salida.

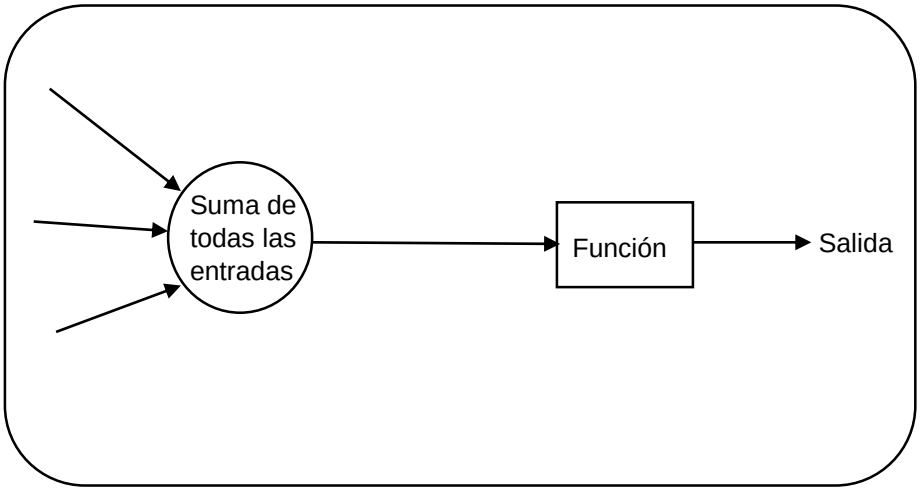

Figura 3. Procesamiento en una neurona artificial

Cada neurona artificial recibe un vector X de entrada que corresponde a todas aquellas señales que llegan a la sinapsis. Cada una de estas señales se multiplica por un peso que tiene asociado W1,W2,W3…Wn. Al conjunto de pesos se le denomina vector W. Cada peso representa la "intensidad" o fuerza de conexión de una sinapsis en una neurona biológica. Los resultados de éstas multiplicaciones se suman. Esta sumatoria simula vagamente al cuerpo de una neurona biológica.

$$Neta_i = \sum X_j W_{ij}$$

Función de Activación

Una vez que la entrada neta ha sido calculada, se transforma en el valor de activación, o activación simplemente y una vez hecho esto se puede aplicar la función de salida que es la encargada de transformar el valor de la entrada neta en

el valor de salida del nodo (Freeman, 2000). La función de activación *F* puede ser lineal o no lineal. Existen varios tipos de funciones de activación:

- **Función Lineal:**

$$OUT = K(NET) \quad donde\ K\ es\ una\ constante\ y\ NET\ es\ una\ señal$$

- **Función logística:**

$$F(x) = \frac{1}{(1 + e^{-NET})}$$

- **Función de tangente hiperbólica:**

$$OUT = Tanh(NET)$$

- **Función umbral:**

$$OUT = 1 \quad si\ NET > T; \text{ ó } 0\ si\ NET < T$$

Clasificación de las redes neuronales

Las neuronas se relacionan entre sí formando redes que pueden llegar a ser tan complejas como el neocognitrón, o tan simples como el perceptron. Las RNA de un nivel (o de una capa oculta), son el modelo más simple según se observa en la Figura 4. Las RNA de varios niveles se pueden visualizar como lo muestra la figura 5. Si existen varios niveles o capas, la función de activación debe ser no lineal, ya que de no ser así una red con varios niveles equivaldría a una red con un nivel.

Tipos de entrenamiento

El entrenamiento es una de las herramientas que las RNA proporcionan para agilizar el aprendizaje. Este proceso consiste en ir ajustando los pesos W gradualmente hasta que el vector de salida resultante coincida con el vector de salida deseado.

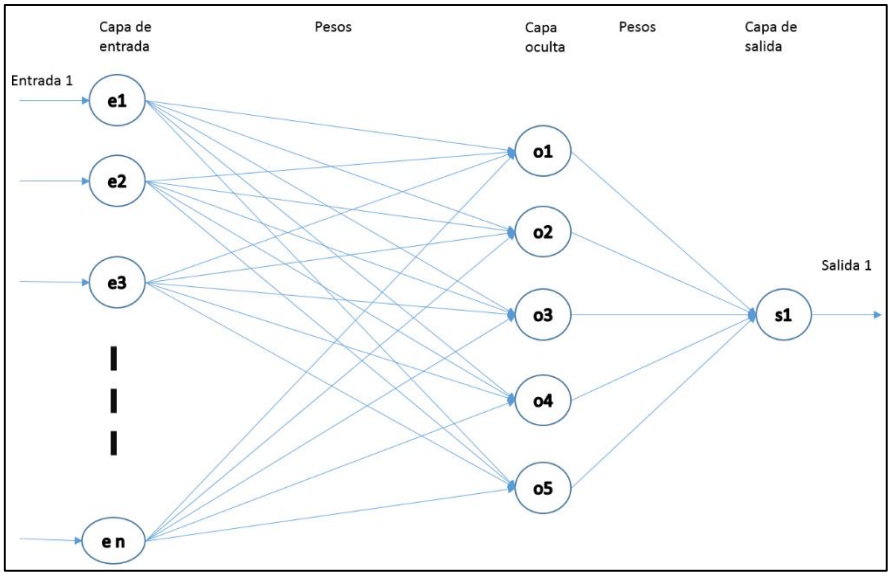

Figura 4. Red Neuronal de nivel 1

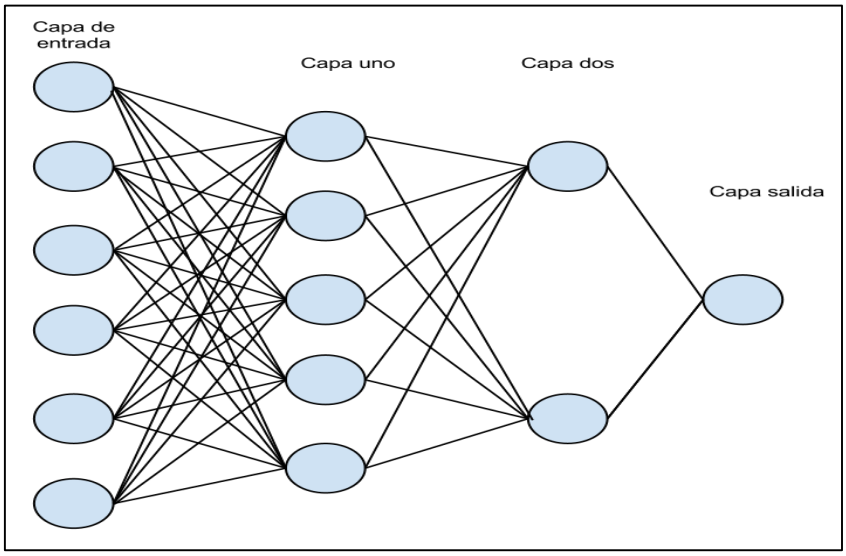

Figura 5. Red Neuronal de varios niveles

El entrenamiento supervisado parte de un vector de entrada del cual se conoce su vector de salida deseada o al menos una aproximación a él. Al par de vectores representando los valores de entrada y salida deseada se le denomina par de entrenamiento. Este proceso consiste en aplicar el vector de entrada a la red. La diferencia o cambio existente entre el vector de salida y el vector de salida deseada se reduce a través de diversos algoritmos existentes. Se continúa probando diversos vectores de entrada ajustando pesos, hasta que la diferencia con la salida deseada es mínima.

En el entrenamiento no supervisado se desconoce la salida, únicamente se proporciona un vector de entrada. Lo que se busca es generar después de varios vectores de entrada, salidas que sean consistentes. Es decir, que los pesos se vayan ajustando poco a poco a través del reconocimiento de patrones, regularidades, propiedades estáticas, etc. Así, las entradas similares producirán el mismo tipo de salida. Otra forma de explicar esto es, que este proceso extrae propiedades estadísticas del conjunto de entrenamiento.

Actualmente la mayoría de los algoritmos de entrenamiento se basan en el trabajo desarrollado por Hebb (1998), quien propuso un algoritmo de entrenamiento sin supervisión donde los pesos W se incrementan si tanto la neurona emisora como la receptora están activados. Este tipo de aprendizaje es el que se adquiere cuando un humano repite una misma tarea. Un ejemplo que se puede mencionar se presenta en los inicios de la medicina, en el que era muy común que un médico buscara constantemente la combinación exacta de ciertas sustancias para lograr que un paciente se curara. La tarea era repetida constantemente hasta que después de varios intentos se descubrían los tipos y cantidades exactos para curar a un paciente en general. Como se puede observar no existía una receta deseada con la cual comparar la salida, sino simplemente se obtuvieron patrones.

Modelo de Retro-Propagación

Este es un algoritmo de aprendizaje que es utilizado para entrenar redes de varios niveles. Fue ideado por Rumelhart, Hinton y Williamsen (1986), este algoritmo posee una base matemática bastante sólida y que es considerado como una generalización de la regla delta. Esta técnica minimiza el error promedio al cuadrado entre la salida real y la esperada, aplicando el concepto de gradiente descendiente (Gómez, 2001).

El objetivo de Retro-propagación es que los pesos de los niveles escondidos generen una representación interna adecuada al problema a resolverse. Estas características y su porcentaje de éxito lo han convertido en uno de los algoritmos de aprendizaje más populares.

4. METODOLOGÍA

En la estudio aquí presentado se partió con una investigación histórica, en la que se realizó la revisión de la información sobre el tema, se seleccionaron los métodos a reproducir y posteriormente se reconstruyeron algunas redes neuronales descritas en trabajos realizados para corroborar su efectividad, en dicha reconstrucción se pudo observar que de una muestra de 15 redes seleccionadas por similitud al objeto de estudio solo 4 se pudieron reconstruir con una efectividad superior al 90% mientras que el resto de ellas su reconstrucción fallo.

Una vez concluida esta etapa de metodología, se dio paso a una metodología descriptiva, describiendo cada tipo de sistema detector de intrusos, y así elegir el adecuado para este tema de investigación. Como una tercera etapa, se utilizó un método experimental, combinando la red neuronal, con el sistema detector de intrusos y manipulando los pesos de la red, así como las variables de clasificación hasta ajustar los resultados obtenidos con los deseados. Por último se utilizó la metodología comparativa, tomando los resultados de esta investigación y analizarlos junto con otros sistemas y redes neuronales hechas con anterioridad.

A continuación se presenta el software que se utilizó para desarrollar la red neuronal, también se ocupó para realizar el análisis y entrenamiento de la misma.

4.1 SOFTWARE DE DESARROLLO DE LA RED

Desde el estudio de los trabajos desarrollados con anterioridad, se observó que Matlab y Java son dos programas más adecuados para la construcción de redes neuronales, ya que cuentan con bloques de código utilizables para dicho objetivo. Por ser de fácil uso, se optó por trabajar en Matlab.

Directamente se trabajó con Matlab distribución R2013b ya que contiene un ambiente de programación similar a C++ y Java, además cuenta con una variedad completa de caja de herramientas que facilitan la programación al incluir un sinfín de aplicaciones (Mendieta, 2008). Este entorno de desarrollo cuenta con funciones

de redes neuronales en la herramienta llamada Toolbox Neural Network y en su interfaz gráfica se facilita el diseño de las RNA.

Es necesario contar con los patterns[4], las características de la red, el algoritmo de entrenamiento junto con las funciones de entrenamiento y los parámetros de entrenamiento (α, η, μ) (Hagan, 2000). En la figura 6 se ve la interfaz de la herramienta de Matlab.

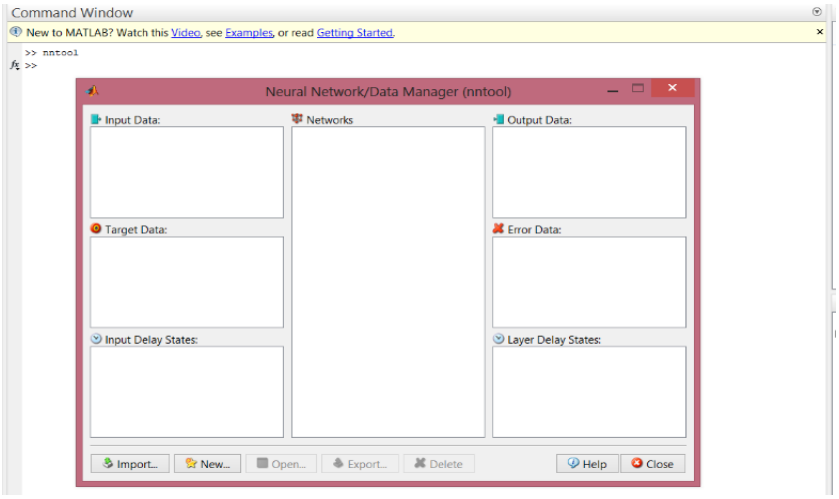

Figura 6. Ventana principal de la interfaz grafica de redes neuronales Matlab

Esta interfaz permite diseñar, visualizar, entrenar y simular redes neuronales entre otras opciones, se necesitan definir las entradas y los objetivos (targets), las características de la red (número de entradas y salidas, las capas ocultas y numero de neuronas que tiene cada capa), además se debe ajustar algunos parámetros que trae por default para cada uno de los tipos de RNA. Cuando se han definido todos los aspectos antes mencionados ya sea en la interfaz o en comandos en la ventana principal de Matlab, se comienza con el entrenamiento, Matlab muestra una gráfica con el progreso del MSE[5] avanzando en cada epoch[6] con la oportunidad de detener el entrenamiento debido a una divergencia o convergencia demasiado lenta.

[4] Patrones de reconocimiento de código, los cuales permiten generar fácilmente las redes neuronales.
[5] Mean Squared Normalized, es la gráfica normalizada, que muestra el avance del entrenamiento.
[6] Época o punto de avance del entrenamiento.

Este entorno de desarrollo, cuenta con diversos tipos de redes neuronales como anteriormente se mencionó, la ayuda (help) documenta cada una de ellos, en esta investigación se trabajó con una red neuronal multicapa, es decir más de 2 capas. La función que se utiliza es newff() la cual lleva la siguiente sintaxis:

Nombre de red=newff(PR,[S1 S2…SN],{TF1 TF2…TFN}BTF,BLF,PF)

De esta función, PR es una matriz de R x 2 con los valores mínimo y máximo que cada entrada R puede tomar, SN es el número de neuronas de la capa N (capas de entrada, intermedia (s) y de salida), TFN es el nombre de la función de activación de las neuronas de la capa N, BTF es el nombre del algoritmo de entrenamiento deseado, BLF es la ecuación de actualización de pesos deseada, y PF es la función de minimización del error que quiere usar (Hagan, 2000).

4.2 BASE DE CONOCIMIENTO

Los datos que se ocuparon para el entrenamiento y comprobación de detección de intrusos fue en un principio la Masquerading User Data de Matthias Schonlau disponible en www.schonlau.net, la cual es un conjunto de datos recogidos con los usuarios que se hacen pasar por usuarios registrados, estos datos los han clasificado y con ellos han comparado los diferentes métodos de detección de intrusos. Estos consisten en 50 archivos para cada usuario, cada archivo contiene 15000 comandos generados con una auditoria de redes.

De los 15000 comandos los primeros 5000 no contienen ninguna mascara de protección y son los que sirvieron para el entrenamiento, los restantes 10000 son cien bloques de cien comandos cada uno, estos comandos fueron sembrados con los usuarios detectados como intrusos, con el objetivo de obtener el patrón que se ocupa para internarse de manera inapropiada a un sistema. Esta base de datos está disponible en código binario lo que facilito el manejo de los datos dentro de la red neuronal, ya que se pueden introducir en cada una de las neuronas de entrada bit por bit.

Se utilizó el sistema SNORT desarrollado por Sourcefire, además de Nessus[7] (que cuenta con ataques surgidos recientemente), y Wireshark para lograr capturar paquetes de comportamiento normal almacenados en formato TCPDUMP[8] y de ataques almacenados en SQL con el objetivo de manipularlos fácilmente. Así se lograron adquirir conjuntos de datos y por medio de estandarización se optó por trabajar con un paquete equilibrado, es decir que cuenta con la misma cantidad de paquetes normales y de ataque (846 en total, 423 en cada caso). Se creó una pequeña interface en java para poder leer la base de datos de SQL que contiene los paquetes de ataque.

Los paquetes contienen el encabezado y los datos de ataque, se consideraron prioritarios los que no variaban la cabecera en todos los paquetes analizados, así como aquellos en los que no influían en la detección, esto se consideró tomando en cuenta las principales reglas de la detección de intrusos.

Las cabeceras que SNORT arrojo y que se consideraron pertinentes son:

- **TTL:** Tiempo de vida, contiene el límite en segundo que un paquete puede estar en la red.
- **Sport:** Puerto Origen.
- **Dport:** Puerto Destino.
- **Seq:** Numero de secuencia, identifica el byte inicial dentro de un segmento de la secuencia de bytes enviados en ese momento.
- **Ack:** Numero de reconocimiento, contiene el siguiente número que el transmisor espera recibir.
- **Win:** Tamaño de ventana advertida por el receptor al transmisor mejor conocida como Sliding Windows.
- **Tos:** Tipo de servicio, indica una serie de parámetros sobre la calidad de servicio deseada durante el tránsito por una red.

[7] Batería de ataques informáticos www.nessus.org
[8] Herramienta en línea de comando, cuya utilidad es analizar el tráfico de una red de datos.

Una vez seleccionados los datos deseados para el sistema, se deben normalizar antes de ingresarlos n la RNA, cada uno de los datos se ha dividido entre el número mayor posible para cada campo, con ello se logra obtener un valor entre 0 y 1, siendo homogéneo y correcto para el ingreso en la red neuronal. En la tabla 1 se presenta como quedaron los datos ya normalizados.

Tabla 1. Datos normalizados generados por SNORT.

Cabecera	*Bits*	*Divididos por*
Tiempo de vida del paquete	8	255
Puerto origen	24	72456
Puerto destino	24	72456
Numero de secuencia	32	5164895762
Carácter de contenido	8	255
Tamaño de ventana	24	72348
Tipo de servicio	8	255

4.3 CONSTRUCCIÓN DE LA RED NEURONAL

Después del análisis de algunas de las redes neuronales que actualmente se utilizan para la detección de intrusos, se ha comprobado que constituyen una herramienta eficiente para el reconocimiento de patrones sin que sea necesario la intervención directa de un usuario, esto se logra debido a que estas redes están basadas en la manera en que el cerebro humano procesa la información. En esta investigación fue necesario utilizar un IDS ya existente llamado SNORT, puesto que en la primera etapa se busca crear la RNA que pueda acoplarse a dicho sistema.

En la figura 7, se muestra la consola de SNORT donde pueden notarse diversos rubros que da de salida, ya que es consecuencia de lo obtenido con las demás herramientas antes mencionadas, se utilizaron solo algunos datos, puesto que existen algunos considerados “no importantes” para la investigación, cabe aclarar que se utilizaron herramientas para abrir el archivo generado por el IDS.

Figura 7. Consola de Snort IDS.

De los datos obtenidos se clasificaron 20 categorías por medio del IDS, estas categorías son tomadas como tipos de ataques, donde cada uno de ellos esta codificado de forma binaria desde $(00001)_2$ hasta $(10100)_2$ lo que conlleva a que la RNA tenga cinco neuronas en una capa de salida. Por otro lado los 215 caracteres del contenido y los siete encabezados laterales de la tabla 1, las entradas serán un total de 222, mientras que las neuronas de la capa oculta se han modificado en su cantidad, para conseguir resultados deseados, cabe destacar que esto se realizó a prueba y error. Para realizar estas modificaciones en los datos se recurrió al estudio propuesto por Díaz Vizcano donde expone que la estructura de toda red neuronal puede modificarse hasta conseguir resultados cercanos a los ideales, y que esos resultados son meramente ambiguos. Con los cinco bits que representan la capa de salida se pueden clasificar hasta 31 tipos de ataques diferentes, esto se puede comprobar gracias a la clasificación realizada por SNORT.

Se trabajó y experimento con la configuración modificando el número de neuronas en la capa oculta y los algoritmos de entrenamiento, aunque los resultados no fueron los esperados, se encontraron problemas de generalización muy evidentes. Por lo que la estructura terminal fue de 222 neuronas de entrada 130 en la capa oculta o intermedia, cinco en la segunda capa oculta y de una sola neurona de salida, tal y como lo muestra la figura 8.

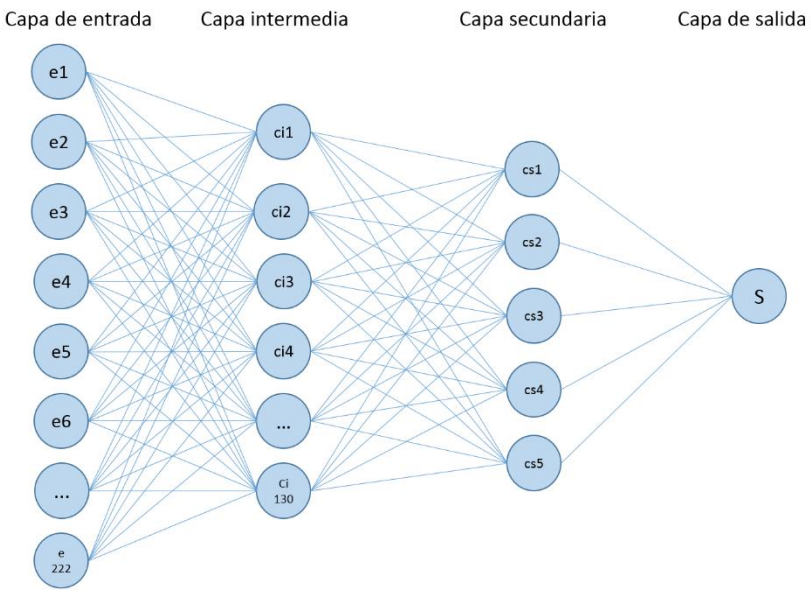

Figura 8. Estructura de la red neuronal final.

4.4 TOPOLOGÍA DE LA RNA

A diferencia de una topología de una red física, en este caso consiste en una estructura determinada, que contiene el número de niveles, de nodos por nivel y la conexión entre ellos. Dado que no existe una metodología única o exacta que indique la manera de conectar o del número de capas ocultas para garantizar los resultados óptimos (Lara, 2002), fue necesario comparar los resultados de las redes de los trabajos de investigación estudiados previo a este trabajo.

Para medir el éxito o fracaso de la estructura de la red es necesario observar el error total por cada barrida que se obtiene durante el aprendizaje, se logra comparando los resultados esperados contra los resultados obtenidos por la red, en caso de que sean muy distintos el porcentaje de error se incrementa, en caso contrario el porcentaje decrementa. Las barridas son el número de veces que la red ejecuta un proceso de aprendizaje, en el caso de Matlab tiene una variable que arroja este dato durante el aprendizaje y entrenamiento de la RNA.

Para el caso de estudio de este trabajo, la topología ideal fue tal y como se ve en la figura 8, con una capa de entrada, dos capas ocultas y una capa de salida. El nivel de salida representa los tipos de soluciones que la red puede obtener, para

este caso al contar con solo una neurona de salida indica si el paquete a analizar es peligroso o no (0 o 1), mientras que al analizar los pesos generados en la segunda capa oculta encontraremos la clasificación de dicho paquete.

Una vez determinado el número de neuronas en cada capa es necesario realizar la conexión entre ellas, es decir los vectores de entrada o sinapsis. Estas señales se multiplican por un peso que tienen asociado, al conjunto de estos pesos se les conoce como vector W, al resultado de estas multiplicaciones se suman y es precisamente esta suma lo que simula el cuerpo de una neurona biológica y se calcula con la siguiente formula.

$$Neta_i = \sum X_j W_{ij}$$

Una vez que se calculó la entrada neta, este valor se transforma en el valor de activación, y justo después de este momento se aplica la función de salida, la cual es la encargada de transformar el valor de la entrada neta en el valor de salida de la neurona, esta función puede ser lineal, no lineal o sigmoidal y para este caso se tomó en cuenta la función sigmoidal para ambas capas ocultas, lo que permite a la red que aprenda de las variaciones de los diferentes tipos de ambientes, que por lo regular fueron no lineales. Con las siguientes líneas de código se realiza la suma dentro de Matlab para cada capa.

```
Net <name> Description
        type Feedforward
        Nodes: A, B, C, D, E, F, G, H, I… n+1
        Weights: w1..w223
        Input Nodes: A, B, C, D…n+1
        Output Nodes : Z
        Relation
                E = A * w1 + B * w2 + C * w3 + D * w4
                F = A * w5 + B * w6 + C * w7 + D * w8
                G = A * w9 + B * w10 + C * w11 + D * w12
                H = E * w13 + F * w14 + G * w15
                I = E * w16 + F * w17 + G * w18
        End Relation
End <name>
```

La primera de las capas ocultas es la encargada de extraer los encabezados de los paquetes mientras que la segunda se encarga de extraer las características

de las neuronas de la capa de entrada en conjunto. Cabe aclarar que mientras más neuronas existan en las capas ocultas mejor fue la aproximación de la función tratada, aunque se observó que si se exagera en el número de neuronas el resultado será una red inestable o un entrenamiento muy tardado, el cual fue un caso que surgió durante la investigación.

4.5 ENTRENAMIENTO DE LA RED

Para lograr el entrenamiento es necesario contar con un archivo único que contenga una mezcla normalizada de datos, los que permitirán enseñar a la red neuronal, para este fin se utilizaron ocho distintos archivos, dos con datos de tráfico normal, dos con paquetes considerados riesgosos, dos con la Masquerading User Data de Matthias Schonlau, y dos más con las salidas deseadas, uno de estos con ceros para determinar el tráfico normal y el otro con unos para identificar el tráfico de riesgo. Al finalizar esta etapa se genera un archivo que intercala o suma cada uno de los tipos de paquetes, se observó que los datos considerados normales fueron menores que los de peligro por tanto este paquete se repitió hasta alcanzar el mismo número de datos.

Los resultados generados u obtenidos son procesados en Matlab en forma de matrices, los parones de entrada que ya están normalizados junto con sus 402 elementos se guardaron en una matriz de 577 elementos por 1654 paquetes generados por el entrenamiento. Por otro lado las salidas deseadas de unos y ceros se almacenan en un vector que permite la corrección del error resultante de la red en esta etapa de entrenamiento.

Cuando en Matlab se ejecuta el entrenamiento, se obtiene un resultado que se amacena e un vector de salida, el cual no contendrá solo ceros y uno sino valores entre ellos, estos valores identifican si se trata de un patrón normal o de un peligro, si se resta la salida generada junto con la esperada, se obtiene el valor absoluto lo que da como resultado una dispersión para cada uno de los valores. El software da un porcentaje de efectividad del 90% para los patrones de entrenamiento.

Como se sabe Matlab cuenta con diversas funciones de entrenamiento y realiza el mismo proceso con cada una de las funciones y para la investigación aquí presentada, se obtuvieron los mejores resultados con las funciones Traincgb, Traincgp y Trainrp, pero con ninguna se obtuvo un valor superior al 91%, lo que al compararlo con los demás estudios no se superó lo esperado. En la figura 9 se muestra el entrenamiento generado con la función Trainrp, el cual tardo en promedio 6 horas las 14 veces que se realizó el estudio, mientras que con las otras funciones el tiempo promedio fue de 45 horas, utilizando los vectores de entrada que contiene parte de la base de datos de Schonlau, por lo que se determinó ya no trabajar con esta base.

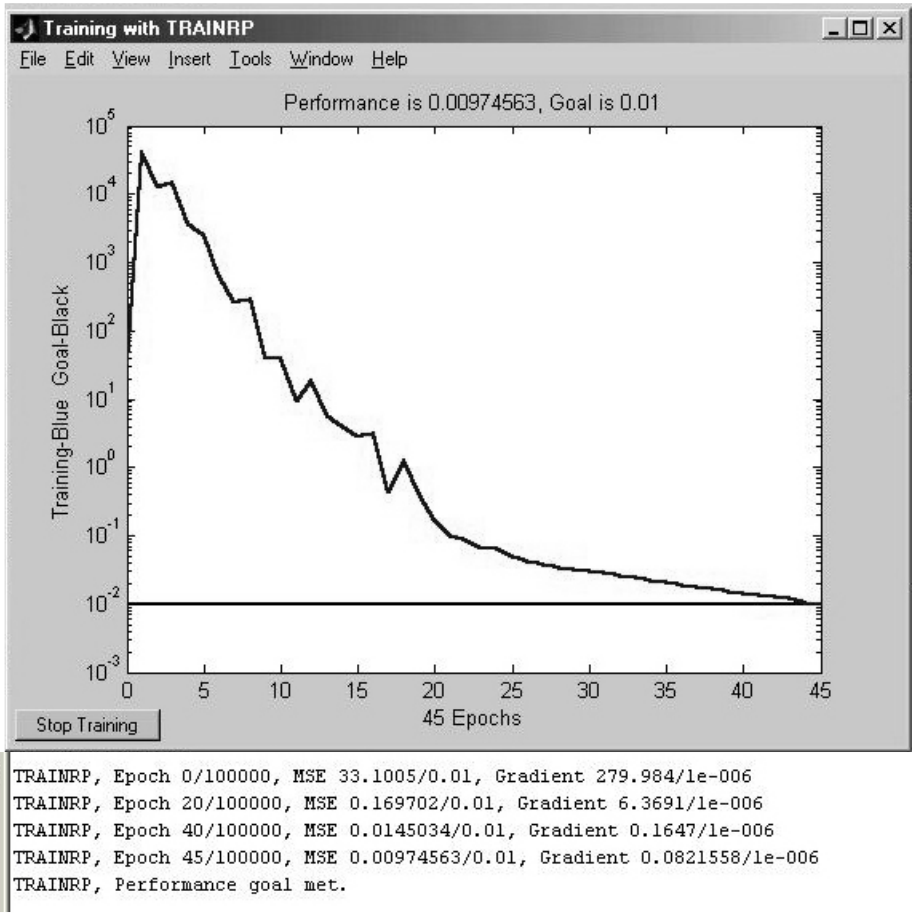

Figura 9. Proceso de entrenamiento con Trainrp.

5.RESULTADOS EXPERIMENTALES

Se presentan los resultados obtenidos al llevar a cabo la simulación del IDS con el modelo contemplado de la red neuronal, así como de la detección de ataques de la red. Mediante estos resultados se puede evaluar distintos indicadores como velocidad de convergencia, error obtenido, porcentaje de clasificación e identificación así como tasas de falsos positivos y falsos negativos que permiten generar conclusiones sobre el desempeño de este modelo.

El funcionamiento correcto del IDS consiste en distinguir entre patrones que representan ataques y aquellos que describen tráfico de red común, a esta medida se le conoce como porcentaje de clasificación. Además, el IDS puede determinar el tipo de ataque que representan los patrones malignos, lo que se conoce como porcentaje de identificación.

Dentro de Matlab se diseñó una pequeña interface de prueba que permite autenticar si los usuarios son o no intrusos, es alimentada con los datos obtenidos y mencionados anteriormente. Es necesario enfocarse a un usuario y detallar su comportamiento, teniendo por variables, la fecha, la hora, el método y la prueba, con estos datos la RNA clasifica el patrón del usuario y muestra la gráfica de variables, en la cual se muestra la diferencia en el comportamiento comparado con otro usuario. Las respuestas pueden ser "**usuario identificado correctamente"** lo que implica que se trata de un acceso correcto, la otra respuesta puede ser **"Intruso"** cuando el usuario tiene un comportamiento diferente al de los demás.

En la figura 10 se muestran los resultados de comportamientos normales y en la 11 los anormales respectivamente, en el cual se logra una detección con la red neuronal y se revisa el comportamiento gracias a la interfaz utilizada. El comportamiento puede ser normal si la información de entrada corresponde a comportamiento habitual o común en el sistema, mientras que en las gráficas de las variables de comportamiento son idénticas. Por otro lado en el comportamiento anormal la información de entrada deben ser de acciones diferentes a las del patrón

normal y las gráficas de las variables no coinciden entre ellas. De esta forma se sabe en qué casos surgen las intrusiones y en qué casos los usuarios se autentican de forma tradicional, gracias al reconocimiento de los patrones de comportamiento de los usuarios

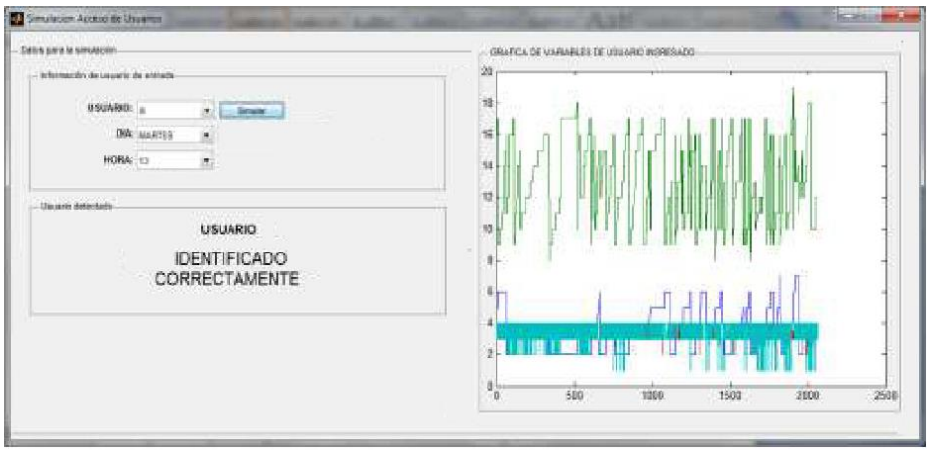

Figura 10. Comportamiento normal del usuario.

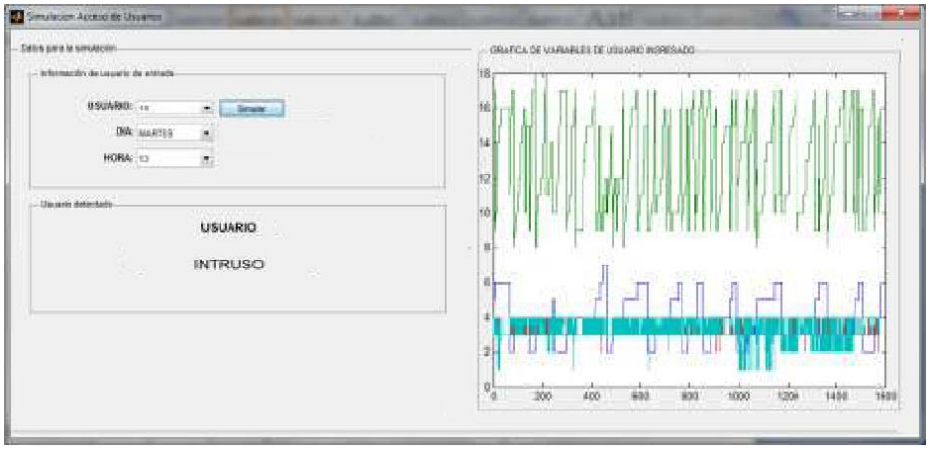

Figura 11. Comportamiento anormal del usuario.

Se platicó con el encargado de sistemas y seguridad de la información del SAT, que prefirió no ser mencionado, y se logró poner en práctica el IDS por media hora en el tráfico de datos de la red de dicha organización, fue ahí donde se pudo observar la funcionalidad cuando existen más de 50 accesos por minuto en el sistema, solo fue necesario desviar los datos hacia el equipo que contenía el sistema de detección y redirigir los datos al servidor, colocándolo como filtro y se comprobó la deficiencia del mismo por no soportar un flujo de datos de tal magnitud. En la figura 12 se

muestra la pantalla que arrojo el sistema en la única detección realizada en la media hora que se puso en práctica.

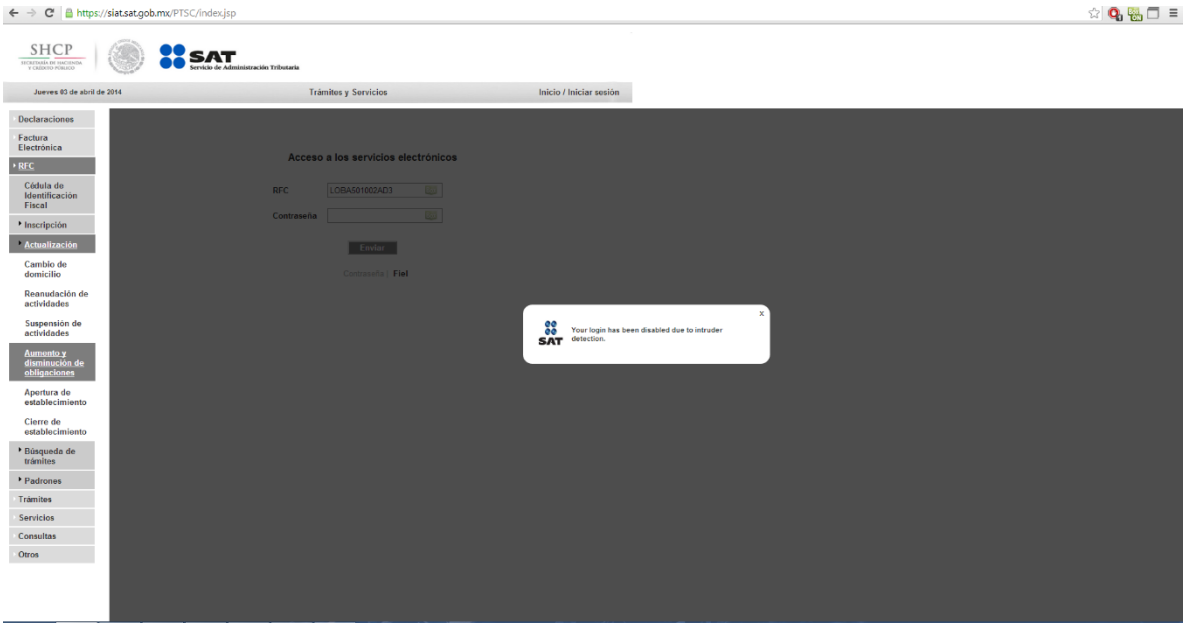

Figura 12. Detección de intrusos en el SAT

Se buscó eliminar la saturación en el análisis de los paquetes de entrada para hacer más eficiente la detección, por lo que se dio un margen de tolerancia en la identificación de paquetes, es decir que el sistema ya no solo detectara el valor de uno como ataque y el cero como normal, sino que tuviera un rango de clasificación lo que reduce el tiempo de clasificación. A continuación se muestra el margen de tolerancia:

- Menor a 0.43 es un paquete normal
- Valor entre 0.44 y 0.64 implica incertidumbre
- Mayor a 0.65 es un paquete peligroso

En el caso de que el resultado de detección sea entre 0.44 y 0.64 el sistema tendrá que realizar un segundo análisis para determinar si efectivamente es incertidumbre o es clasificado como peligroso o normal. En la tabla 2 se muestran los resultados obtenidos durante la puesta en práctica del sistema en dos redes locales (CUX y UTN) y en 2 redes virtuales simuladas.

Tabla 2. Comparación del desempeño en cuatro redes.

Redes y No. paquetes	*Tolerancia*	*% éxito analizados*	*% éxito paq. Normales*	*% éxito paq. Peligrosos*	*Falsos (-)*	*Falsos (+)*
CUX	0.01	97.36	87	96	1	3
(190)	0.001	98.83	90	97	1	1
UTN	0.01	95.24	91	99	2	4
(200)	0.001	99.00	101	97	1	1
Sim 1	0.01	96.65	88	92	0	3
(190)	0.001	97.34	88	94	1	15
Sim 2	0.01	98.01	100	95	0	2
(200)	0.001	97.89	96	99	2	11

Para la prueba en los cuatro casos se encontraron problemas en el análisis, en diversas ocasiones la prueba fallo, se debe mencionar que aunque se repitió 4 veces la prueba en cada red, no se pudo obtener los mismos resultados, incluso llegaron a variar por más del 40% y esto es debido a la variación del tráfico en las redes físicas del CUX y de la UTN, y en el caso de las simulaciones, no se logró repetir los resultados debido a las pequeñas variaciones en el análisis que representa una red neuronal y a la tolerancia que manejan. En la figura 13 se puede observar los resultados de la detección en la red sim 2, en puntos es con tolerancia de 0.01 y en línea continua es con tolerancia 0.001.

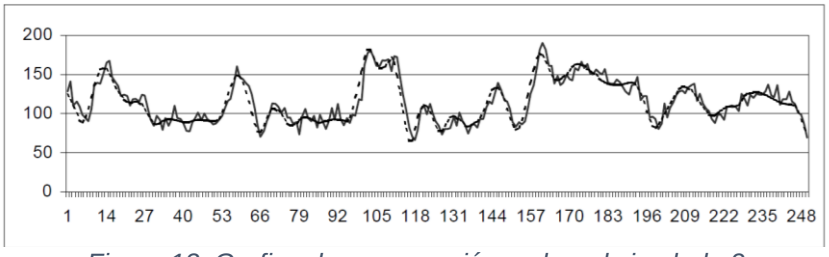

Figura 13. Grafica de comparación en la red simulada 2.

A continuación se describe el análisis que se realizó sobre los datos del ataque realizado en la red de la UTN detectados por SNORT implantado antes del cortafuego. Se implantaron 2 conjuntos de datos: uno tomado en un tiempo donde ocurrió el ataque real, y otro tomado en un tiempo en el que se simulo el ataque.

La figura 14 muestra la primer prueba, en la que se grafican 2 señales, esto se hace con el objetivo de analizar el comportamiento de una red normal. El lapso de tiempo de muestreo es de 1000 milisegundos de Delta Time (tiempo en ocurrir el siguiente evento). Cabe hacer notar que en la gráfica, se muestran los falsos positivos y los falsos negativos, encontrando en dos puntos los ataques realizados.

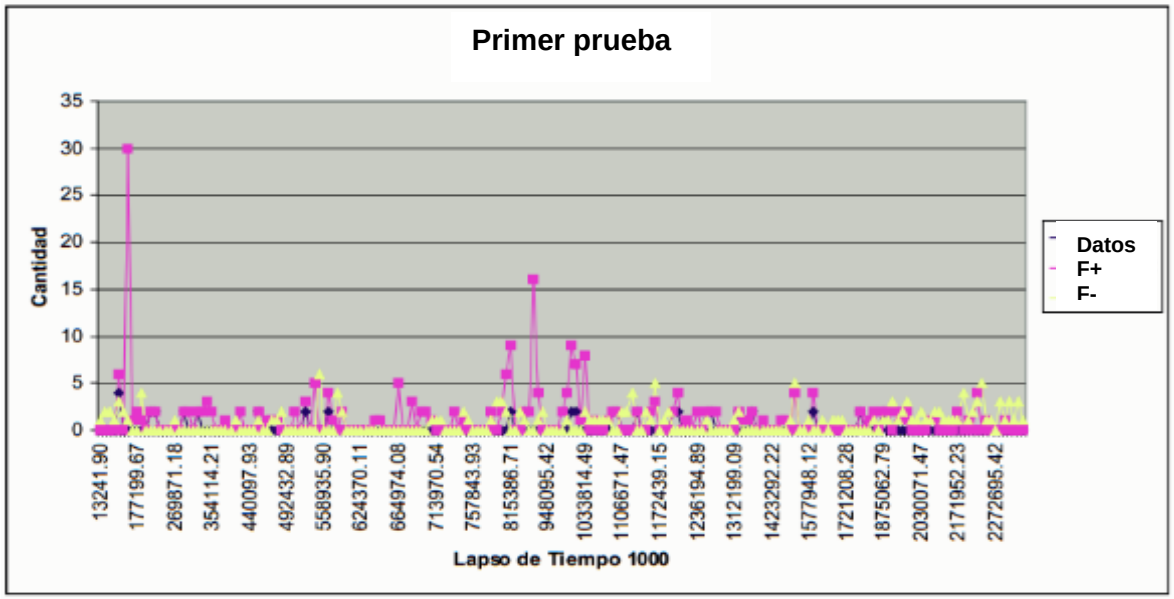

Figura 14. Primer prueba realizada en la UTN.

La figura 15 muestra la segunda prueba. Ahora se muestran los falsos positivos y negativos cuando la red está saturada, es importante mencionar que como se observa la cantidad es elevada, por lo que la capacidad del IDS se ve disminuida al no poder encontrar realmente las intrusiones y toma a todos los usuarios como peligrosos.

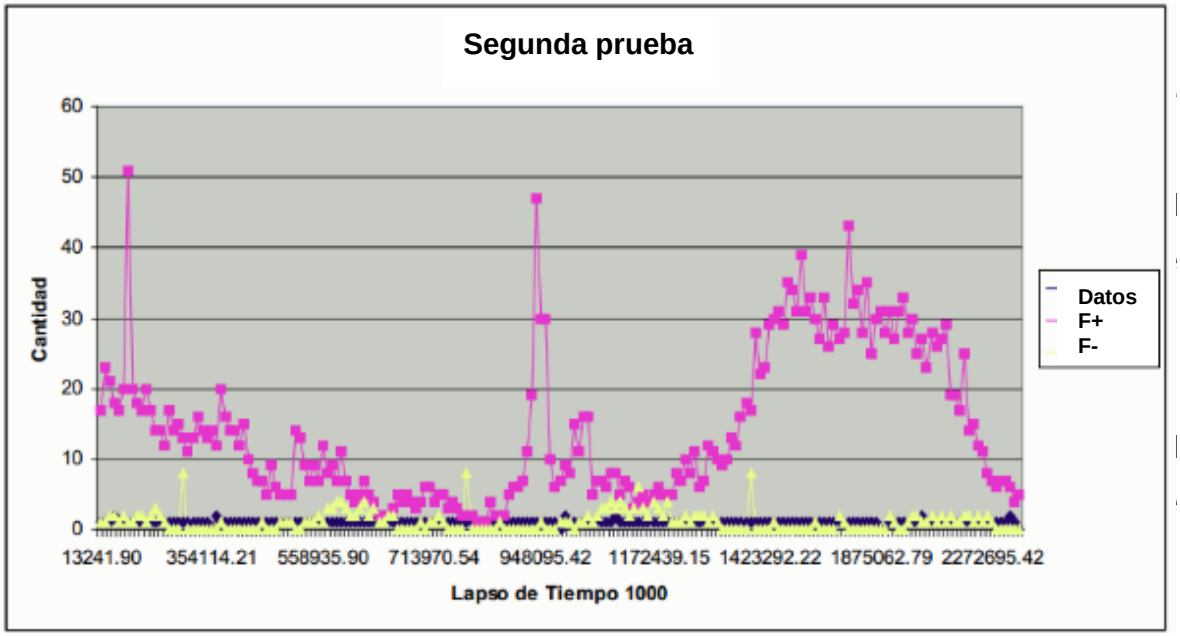

Figura 15. Segunda prueba realizada en la UTN.

Como puede observarse en las figuras 14 y 15, la disparidad de los falsos positivos son elevados, esto disminuye la posibilidad de detectar eficazmente al intruso, se realizaron modificaciones en la red neuronal buscando disminuir estos falsos positivos, y se logró, aunque solo se tuvo la oportunidad de probar de forma simulada, ya que es difícil que alguna institución permita hacer pruebas en su red interna.

6. TRABAJOS DE INVESTIGACIÓN REALIZADOS

En este apartado, se presentan algunos productos (artículos o participaciones) que durante el desarrollo de la investigación se realizaron, estos son parte del desarrollo de la tesis, los cuales permitieron obtener experiencia acerca del tema.

- Se participó en el VIII Coloquio de Investigación de la Maestría en Ciencias de la Computación 2012B. Se realizó la exposición de los avances del trabajo de tesis en la ciudad de Temascaltepec el 13 de Diciembre de 2012, a cargo de investigadores de diferentes espacios académicos pertenecientes a la comunidad UAEM, en el cual se obtiene una constancia (Anexo 1).
- Se participó en el IX Coloquio de Investigación de la Maestría en Ciencias de la Computación 2013A. Se realizó la exposición de los avances del trabajo de tesis en la ciudad de Atlacomulco el 30 de Mayo de 2013, a cargo de investigadores de diferentes espacios académicos pertenecientes a la comunidad UAEM, en el cual se obtiene una constancia (Anexo 2).
- Se participó con el artículo titulado “Sistemas detectores de intrusos que utilizan redes neuronales (grado de error)” en el Congreso Internacional de Investigación de Academia Journals Chiapas 2013, los días 4, 5 y 6 de Septiembre en la ciudad de Tuxtla Gutiérrez, Chiapas; en donde se presentaron los avances sobre la red neuronal artificial que se desarrolló, así como del enlace con el sistema detector de intrusos que forma parte de la investigación. Se presenta la constancia, portada y la primer hoja del articulo (Anexo 3).
- Se participó en el 4to Coloquio Internacional de Cómputo e Informática el 14 de Noviembre de 2013 en Valle de Chalco, se presentó el marco teórico para crear una red neuronal artificial en donde se enfatizó sobre los paradigmas en el entrenamiento de la misma, del cual se obtiene constancia (Anexo 4).
- Se participó en el Congreso Internacional Virtual de Innovación, Tecnología y Educación CIVITEC 2013, los días 4, 5, y 6 de diciembre; en donde se presentó los avances del entrenamiento de la red neuronal, así como las primeras detecciones de ataques pasivos, en este congreso se obtiene la

experiencia de defender el trabajo de forma no presencial. Se presenta la constancia y la portada (Anexo 5)

- Se participó en el IV Foro Científico Interdisciplinario de la DES oriente, presentando el artículo titulado “Entrenamiento de una Red Neuronal Artificial para detectar intrusos en una Red de Datos”, en la ciudad de Texcoco el 05 de Noviembre de 2014, a cargo de investigadores de diferentes espacios académicos pertenecientes a la comunidad UAEM. Se presenta el entrenamiento final de la red neuronal, se generaron conocimientos y recomendaciones para mejorar el aprendizaje de la red, se obtiene un reconocimiento, el cual se presenta junto con la primer hoja del articulo (Anexo 6).
- Se realizó un artículo en ingles titulado “System for Intrusion Detection whit Artificial Neural Network”. En este trabajo se presentan las detecciones realizadas por el sistema detector de intrusos utilizando la red neuronal desarrollada, se realiza una triple clasificación de los ataques y el porcentaje de efectividad de las detecciones. Se envió al Institute for Systems and Technologies of Information, Control and Communication, con el objetivo de participar en la International Conference on Agents and Artificial Intelligence (ICAART) 2015 en la ciudad de Lisboa, Portugal. De dicha participación lograría su publicación digital en SCITEPRESS, Fue aceptado. Se anexa carta de aceptación (Anexo 7).

Con estos trabajos se sustenta lo establecido en la presente investigación, ya que han sido fruto del desarrollo de la misma, se han adquirido experiencias, así como nuevos conocimientos gracias a las sugerencias de los asistentes a los coloquios.

7.CONCLUSIONES

El utilizar redes neuronales en la detección de intrusiones dentro de las redes de datos logra hacer más eficiente las detecciones tradicionales gracias al aprendizaje, para lograrlo es necesario contar con un entrenamiento correcto, así como una selección adecuada de datos que puedan ser utilizados por el sistema. La peculiaridad de las RNA se basa en la incertidumbre que poseen, esto permite detectar comportamientos anormales que otros sistemas descartan gracias a que son basados en reglas tradicionales. Mientras que no exista un solo sistema de entrenamiento es necesario recurrir a los diversos métodos, lo que implica cometer errores que son imperceptibles, la creación de redes neuronales artificiales en diversos ámbitos permite recurrir a estudios realizados para observar la mejor manera de obtener los resultados buscados, lo que permite reducir tiempos, aunque como es sabido no todos los estudios se han podido reproducir aun cuando se han seguido paso a paso, lo que implica pérdida de tiempo y esfuerzo al ser tomados en cuenta.

Para esta investigación se han logrado obtener patrones de comportamiento diferentes para cada usuario y se puede obtener su identidad de manera correcta. Debido a la depuración de la información, se identificaron patrones de comportamiento para cada usuario y de esta forma generar una automatización en el control de intrusos en el sistema, y con ello se desarrollaron los perfiles de comportamiento para cada usuario, para este trabajo se comparó el resultado entre los diferentes comportamientos y se observó que se identificaron intrusos de forma adecuada. Cualquier implementación digital presenta un compromiso de diseño entre precisión, por un lado, y parámetros físicos como área y velocidad de cálculo por el otro. Es necesario, entonces, analizar el comportamiento de la red neuronal entrenándola con diferentes series de datos y con distintas precisiones internas.

Durante la etapa previa al desarrollo de este trabajo, se revisaron diversos trabajos realizados, que al recrearlos al pie de la letra no se consiguió llegar a lo establecido. La experiencia obtenida al momento de presentar artículos con los

resultados aquí expuestos es muy satisfactoria, permitió motivar el desarrollo de más trabajos, tengo la firme convicción de que seguiré mejorando el sistema e incluso, si es necesario reconstruir por completo todo, para generar un sistema que eleve el la detección de intrusos.

8. TRABAJO FUTURO

Después de conocer el funcionamiento del IDS junto con la red neuronal, es necesario buscar modelos basados en otro tipo de redes neuronales. Las redes de datos del CUX y de la UTN están basadas en sistemas oscilatorios y bloquean el correcto uso del IDS. Por lo tanto, se requiere buscar redes de datos que permitan el debido comportamiento del sistema de detección y además que lo haga en tiempos no oscilatorios.

Surge la inquietud de realizar un seguimiento en los experimentos en el cual se estudien diversas variables que intervienen en los sistemas informáticos, ya sea la interacción con el medio, las señales emitidas por cada nodo de la red, así como algunos movimientos del usuario (físicos), el estudio de estas variables podría generar un perfil para cada usuario y evaluar el comportamiento de los usuarios de la red.

9.REFERENCIAS BIBLIOGRÁFICAS

[Abler, 2003] Abler, R. (2003). Intrusion detection testing and benchmarking methodologies. 63-72.

[Agency, 2010] Agency, N. S. (2010). *National information systems security (infosec) glosary.* New York, USA: National Security Agency/Central Security Service Fort George G Meade MD.

[Alessandri, 2004] Alessandri, D. (2004). *Attack-Class-Based Analisis of Intrusion Detection System.* Newcastle, Inglaterra: University of Newcastle.

[Andersen, 2001] Andersen, R. (2001). *Security Engineering: A guide to building dependable distributed systems.* New York, USA: Jhon Wiley & Sons Inc.

[Anderssen, 1972] Anderssen, J. (1972). Computer security technology planing study. *ESD-TR-73-51 Electronics System Division(AFSC)*, 1-32.

[Anderssen, 1980] Anderssen, J. (1980). *Computer security threat monitoring and surveillance.* Los Angeles, California: J. P. Anderson Co.

[Axelsson, 2006] Axelsson, S. (2006). *Intrusion Detection Systems: A Survey and Taxonomy.* Sweden : DCECUTG.

[Barker, 2007] Barker, K. (2007). Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria
for rule prescreening. *Journal of Network and Computer Applications 30*, 99-113.

[Barrera, 2010] Barrera, C. R. (2010). *Detección De Anormalidades En Las Redes LAN Mediante Implementación De Un PCA.* Campeche, México: Centro De tecnologías De La Información Universidad Autónoma Del Carmen.

[Bonilla, 2008] Bonilla, C. M. (2008). Sistema Neuronal de Detección de Intrusos. *Tendencias en Ingeniería de Software e Inteligencia Artificial*.

[Bouhola, 2004] Bouhola, A. (2004). On the fly pattern matching for intrusion detection with snort. *Annales de telecommunications*, 941-967.

[Britos, 2010] Britos, J. D. (2010). *Detección de intrusiones en redes de datos con captura distribuida y procesamiento estadístico.* Buenos Aires, Argentina: Universidad Nacional de la Plata.

[Chang, 2007] Chang, J. (18 de Junio de 2007). *www.cert.org*. Obtenido de http://www.cert.org/tech_tips/denial_of_service.html.

[Chen, 2007] Chen, S.-Y. (2007). Building Intrusion PAttern Miner for Snort Network Intrusion Detection System. *Journal of Systems and Software* , 1699-1715.

[Coyne, 2006] Coyne, E. (2006). Role-Based Access control models. *Computer*, 38-47.

[Das, 2010] Das, V. P. (2010). Network Intrusion Detection Based On Machine Learning Algorithms. *International Journal Of Computer Science And IT*.

[Debar, 1999] Debar, H. (1999). Towards a taxonomy of intrusion detection systems. *Computer Networks*, 805-822.

[Denning, 1987] Denning, D. E. (1987). An Intrusion-Detection Model. *IEEE transaction on software Engineering*, 222-232.

[Díaz, 2008] Díaz, V. L. (2008). *Estudio tecnologico sobre sistemas de detección de intrusos.* Madrid, España: Universidad Carlos III de Madrid.

[Dreger, 2006] Dreger, H. (2006). *Operational Experiences with High-Volume Network Intrusion Detection.* Washington DC, USA: CCS'06.

[Fan, 2005] Fan, W. (2005). *Minning System Audit Data: Opportunities and Challenges.* Tokio, Japan: SIGMOD Record.

[Foukia, 2005] Foukia, N. (2005). *IDReAM: Intrusion Detection and Response executed with Agent Mobility Architecture and Implementation.* Utrecht, Netherlands: AAMAS'05.

[Freeman, 2000] Freeman, J. (2000). *Redes Neuronales: Algoritmos, Aplicaciones y Tecnicas de Programación.* México: Addison Wesley.

[Gao, 2005] Gao, H. H. (2005). Ant colony optimization based network intrusion feature selection and detection. *Proceedings of 2005 International Conference* (págs. 3871-3875). Machine Learning and Cybernetics.

[Gómez, 2001] Gómez, G. P. (2001). *Notas del curso Fundamentos de Redes Neuronales.* Puebla, México: Universidad de las Americas.

[González, 2003] Gonzalez, G. D. (13 de Julio de 2003). *www.dgonzalez.net.* Obtenido de http://www.dgonzalez.net/pub/ids/html/

[González, 2009] González, M. V. (2009). *Detector de Intrusos Basado en Sistema Experto.* México DF, México: IPN.

[Gorton, 2010] Gorton, S. (2010). *Combining Evasion Techniques to Avoid Network Intrusion Detection Systems.* Los Angeles Californa, USA: Skaion Corporation.

[Grediaga, 2002] Grediaga, A. (2002). *Utilización de redes neuronales para la deteccion de intrusos.* Morelia, México: CIBSI.

[Hagan, 2000] Hagan, M. (2000). *Diseño de redes neuronales.* USA: Brooks/cole Publishing.

[Hu, 2006] Hu, C.-y. (2006). *A framework of cooperating Intrusion Detection based on Clustering analysis and expert system.* Shanghai, China: InfoSecu.

[Joshi, 2005] Joshi, A. (2005). *Detecting Past and Present Intrusions through Vulnerability-Specific Predicates.* Brighton, United Kingdom: SOSP.

[Kenneth, 2010] Kenneth, I. (2010). A history and survey of network firewalls. *ACM*, 1-42.

[Kumar, 2007] Kumar, S. (2007). Smurf-based distributed denial of service (ddos) attack amplification in internet. *Internet Monitoring and Protection*, 25-25.

[Lang, 2009] Laing, B. (13 de Mayo de 2009). *www.snort. org.* Obtenido de http://www.snort.org/docs/iss-placement.pdf

[Lara, 2002] Lara, F. (2002). Artificial Neural Networks: An Introduction. *Journal of the Mexican Society of Information*.

[León, 2010] León, H. R. (2012). *Definición de un modelo de seguridad en redes de cómputo, mediante el uso de técnicas de inteligencia artificial.* Colombia: Universidad Nacional de Colombia.

[Li, 2005] Li, X.-B. (2005). A scalable decision tree system and its application in pattern recognition and intrusion detection. *Elsevier Decision Support Systems*, 112-130.

[Liu, 2009] Liu G., Y. Z. (2009). A hierarchical intrusión detection model base don the PCA neural networks. *ELSEVIER, Neurocomputing*.

[Lucas, 2007] Lucas, C. (2007). Intrusion detection using a fuzzy genetics-based learning algorithm. *Elseiver*, 414-428.

[Luger, 1990] Luger, G. (1990). *The architecture of a network level intrusion detection system.* New México, USA: University of New México.

[Magno, 2006] Magno, M. B. (2006). *Encuesta de Mecanismos de Autenticación de Usuarios.* Monterrey, México: Escuela de posgraduados de Monterrey.

[Manikopoulos, 2011] Manikopoulos, C. y. (2011). Network intrusion and fault detection: a statistical anomaly approach. *Communications Magazine*, 76-82.

[McHugh, 2000] McHugh, J. (2000). Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory. *ACM Transactions on information and system security*, 262-294.

[Mellado, 2007] Mellado, D. (2007). A security requeriments engineering process in practice. *IEEE America Latina*, 211-217.

[Mendieta, 2008] Mendieta, D. (2008). *RConocimeinto de Objetos Bidimensionales en Imagenes Mediante la Transformada de Distancia Utilizando Matlab.* Puebla, México: UDALP.

[Mira, 2009] Mira, A. E. (2009). *Tesis: Implantación de un sistema de detección de intrusos en la Universidad de Valencia.* Valencia, España: Universidad de Valencia.

[Naseehi, 2003] Nassehi, M. (2003). *Anomaly detection for Markov models.* Zurich: IBM Research Division.

[Nong, 2006] Nong, Y. (2006). Attack profiles to derive data observations, features and characteristics of cyber attacks. *Information, Knowledge, Systems Management*, 1-25.

[Paxon, 2010] Paxson, V. (20 de Mayo de 2010). *www.bro-ids.org*. Obtenido de http://www.bro-ids.org/Features.html

[Pérez, 2010] Pérez, C. B. (2010). Aplicación de Redes Neurales para la Detección de Intrusos en Redes y Sistemas de Información. *Scientia et Técnica*, 225-230.

[Power, 2012] Power, R. (2012). *Current and future danger.* San Francisco, California: Computer Security Institute.

[Quist, 2007] Quist, A. S. (18 de Junio de 2007). *www.fas.org*. Obtenido de http://www.fas.org/sgp/library/quist/index.html.

[Sánchez, 2007] Sánchez, M., Jiménez, B., & Gutiérrez, F. (2007). *Estudio del control de acceso en sistemas colaborativos.* Zaragoza, España: XII JORNADAS DE INGENIERIA DEL SOFTWARE Y BASES DE DATOS.

[Schalkoff, 2003] Schalkoff, R. (2003). *Artificial Neural Networks.* USA: McGraw-Hill.

[Sommer, 2005] Sommer, R. (2005). *Enhancing Byte-Level Network Intrusion Detection Signatures with Context.* Washington DC, USA: CCS'05.

[Soto, 2005] Soto, P. (2005). Mimicry Attacks on HostBased Intrusion Detection Systems. *CCS'05*, 18-22.

[Stoneburner, 2001] Stoneburner, G. (2001). Underlying Technical Models for Information Technology Security. *Computer Security Division* , 35-39.

[Stroud, 2001] Stroud, R. (14 de Noviembre de 2001). Conceptual model and architecture, Deliverable D2. *IBM Zurich Research Laboratory*, págs. 33-35.

[Tseng, 2004] Tseng, S.-S. (2004). Constructing detection knowledge for DDoS intrusion tolerance. *Elseiver*, 379-390.

[UNION, 1996] UNION, I. T. (1996). Open system interconnection OSI; Security Structure and Applications. *Data Comunication Networks*.

[Valenzuela, 20008] Valenzuela, P. P. (2008). *Una propuesta de IDS, basado en Redes Neuronales Recurrentes.* Santiago de Chile, Chile: Universidad Santiago de Chile.

[Wagner, 2006] Wagner, D. (2006). Intrusion Detection via Static Analisis. *IEEE. Security and Privacy*, 156-168.

[Zanero, 2008] Zanero, S. (2008). Unsupervised learning techniques for an intrusion detection system. *SAC'08*, 115-128.

[Zhang, 2006] Zhang, Z. (2006). *Adaptive Observation-Centric Anomaly-Based Intrusion Detection: Modeling, Analysis and Evaluation.* Tokio, Japan: GRP Repor.

[Zhicai, 2004] Zhicai, S. (2004). *A Novel Distributed Intrusion Detection Model Based on Mobile Agent.* Shanghai, China: InfoSecu04.

[Zurutuza, 2004] Zurutuza, O. U. (2004). *Estado del arte sistemas de detección de intrusos.* Mondragon, España: Escuela Politecnica Superior de Mondragon Unibertsitatea.

10. ANEXOS

Anexo 1

CONSTANCIA

Anexo 2

CONSTANCIA

Universidad Autónoma del Estado de México
Centro Universitario UAEM Atlacomulco

S E O T O R G A L A P R E S E N T E

CONSTANCIA

A: JOSÉ ERNESTO LUNA DOMIGUEZ

Por su ***PARTICIPACIÓN*** *en el "IX Coloquio de Investigación de la Maestría en Ciencias de la Computación 2013-A", llevado a cabo en el Centro Universitario UAEM Atlacomulco, el día 30 de mayo del 2013.*

PATRIA, CIENCIA Y TRABAJO
"2013, 50 Aniversario Luctuoso del Poeta Heriberto Enríquez"

M. EN A. FIDENCIO OCHOA FLORES
Encargado del Despacho de la Dirección del
Centro Universitario UAEM Atlacomulco

Anexo 3

CONSTANCIA

opus pro scientia et studium

Congreso Internacional de Investigación de

ACADEMIA JOURNALS

CHIAPAS 2013

Certificado

Otorgado a

ING. JOSÉ ERNESTO LUNA DOMÍNGUEZ, MTRA. ANABELEM SOBERANES MARTIN

por su ponencia intitulada

Sistemas detectores de intrusos que utilizan redes neuronales (grado de error)

que fue presentada en el Congreso Internacional de Academia Journals Chiapas 2013
que se desarrolló los días 4 al 6 de septiembre de 2013
en Tuxtla Gutiérrez, Chiapas, y publicada en el portal de Internet
Chiapas.AcademiaJournals.com con **ISSN 1946-5351** Online y **1948-2353** CD ROM

Sabino Velázquez Trujillo
Presidente de la Comisión Organizadora
Profesor de Ingeniería Industrial
Instituto Tecnológico de Tuxtla Gutiérrez

Dr. Rafael Moras
Presidente del Comité de Programa del Congreso
Editor, AcademiaJournals.com
Profesor, St. Mary´s University, San Antonio, TX, EEUU

322 LunaDomínguez 322

PORTADA

Co-patrocinado por

Instituto Tecnológico de Tuxtla Gutiérrez
Tuxtla Gutiérrez, Chiapas, México

Academiajournals.com,
Una División de PDHTech, LLC
San Antonio, TX, EEUU

Universidad Autónoma de Chiapas
Tuxtla Gutiérrez, Chiapas, México

Tuxtla Gutiérrez, Chiapas, México
Septiembre 2013

ISSN 1946-5351 Online 1948-2353 CD ROM

PRIMER HOJA DEL ARTÍCULO

CONGRESO INTERNACIONAL DE INVESTIGACION
ISSN: 1946-5351 ONLINE, 1948-2353 CD ROM
chiapas.academiajournals.com
Vol. 5 No. 3

Sistemas Detectores de Intrusiones que utilizan Redes Neuronales Artificiales (grado de error)

Ing. José Ernesto Luna Dominguez[1], Mtra. en E. Anabelem Soberanes Martin[2]

Resumen- El presente proyecto expone la integración de dos tecnologías para obtener resolver el problema de intrusiones en las redes de información, estas tecnologías son a través de las Redes Neuronales Artificiales (RNA) y de los Sistemas Detectores de Intrusiones (IDS por sus siglas en inglés). Se analizan además, algunas ventajas y desventajas de dicha integración, para disminuir el margen de error de los IDS que no integran las RNA. En el artículo se presentan algunos ejemplos de los sistemas detectores de intrusiones sus, características y su utilidad.

Palabras Clave- Red neuronal, perceptron, falsos positivos, falsos negativos

I. INTRODUCCIÓN

Los IDS es son sistemas auxiliares que permiten aumentar la seguridad dentro de una red de datos, son integrados por módulos los cuales pueden funcionar sobre actividades específicas o pueden conjuntarse y monitorear el flujo de datos de una red de cualquier organización buscando pistas o señales que indiquen una posible intrusión, así como usuarios que no están autorizados dentro de la misma, por otro lado los módulos pueden específicamente analizar las malas prácticas de los usuarios que cuenten con autorización pero que intentan sobre limitarse de las restricciones de acceso a la información [1].

La mayoría de los sistemas son tan variados y, los IDS no son la excepción cuentan con diversas clasificaciones [2]: por el tipo de detección con el que cuenta, por su situación física o por la naturaleza y reacción al detectar un ataque.

Por el modelo de detección que realiza se dividen en:
• Detección del uso anómalo: una vez que el sistema comprende el tráfico estándar o promedio en la red y lo aparta del tráfico no estándar realiza un esquema estadístico que contiene patrones definidos y son comparados con los datos reales analizados en busca de desviaciones estadísticas.
• Detección del mal uso: Involucra la comprobación sobre tipos no legales de tráfico en la red, pueden ser incluidos intentos de intrusiones al momento de ejecutar programas no autorizados.

Por su situación se dividen en:
• NIDS (Network Intrusion Detection System)
• HIDS (Host Intrusion Detection System)

Los NIDS hacen un análisis del trafico dentro de toda la red, pero examinan individualmente los paquetes de datos, este análisis permiten la comprensión de las diferentes opciones que existen en cada uno de los paquetes de la red y detectan aquellos que cuentan con malicias en el armado y el diseño lo que permitiría no ser detectados por los firewall.

Cuentan con un sensor colocado específicamente en algún segmento de la red, el cual monitoriza la misma en busca de tráfico inseguro tal y como se muestra en la figura 1. También cuentan con una consola la cual recibe las

---

[1] Ing. José Ernesto Luna Dominguez es estudiante de 2° semestre de la Maestría en Ciencias de la Computación de la UAEMex. joseven77@gmail.com
[2] Mtra. en Ciencias de la Educación Anabelem Soberanes Martin es Profesora de Investigación de la Universidad Autónoma del Estado de México en el Centro Universitario UAEM Valle de Chalco. asoberanesm@uaemex.mx

Anexo 4

CONSTANCIA

UAEM | Universidad Autónoma del Estado de México

Centro Universitario UAEM Valle de Chalco

4° Coloquio Internacional de Cómputo e Informática

13 al 15 de noviembre de 2013

Otorga la presente

Constancia

Al Ing. José Ernesto Lara Domínguez y a la Mtra. Anabelem Soberanes Martín

Por su participación con la Ponencia "Entrenamiento de una Red Neuronal Artificial para Detectar Intrusos en una Red de Datos" en el marco del 4to. Coloquio Internacional de Computación e Informática

Valle de Chalco, Estado de México a 13 de Noviembre de 2013.

Dr. René Guadalupe Cruz Flores
Coordinador de Investigación

Dra. Magally Martínez Reyes
Directora del Centro Universitario

M. en Ed. Anabelem Soberanes Martín
Líder de C. A. Cómputo Aplicado

Anexo 5

CONSTANCIA

Otorga la presente

CONSTANCIA

A: José Ernesto Luna Domínguez

Por su **Participación en el**
Congreso Internacional Virtual de Innovación, Tecnología y Educación CIVITEC 2013

Tijuana, Baja California, 4, 5, y 6 de diciembre de 2013

"Superación...Divino Tesoro"

ATENTAMENTE

MAG. María Elizabeth Ojeda Orta
Comité Organizador
CIVITEC 2013

PORTADA

Congreso Internacional Virtual de Innovación Tecnología y Educación

CIVITEC 2013

ISBN: 978-607-96314-4-4 Online

CIVITEC
CONGRESO INTERNACIONAL VIRTUAL DE INNOVACIÓN, TECNOLOGÍA Y EDUCACIÓN

Compilador:
María Elizabeth Ojeda Orta

Anexo 6

RECONOCIMIENTO

UAEM | Universidad Autónoma del Estado de México

Otorga a

José Ernesto Luna Domínguez

el presente

RECONOCIMIENTO

Como ponente disertando el tema intitulado

Entrenamiento de una Red Neuronal Artificial para detectar intrusos en una Red de Datos

llevada a cabo en este campus universitario
el 5 de noviembre de 2014
en el marco cultural del IV Foro Científico
Interdisciplinario de la DES oriente

"Patria, Ciencia y Trabajo"

Dr. en D. Ricardo Colín García
Director

CENTRO UNIVERSITARIO UAEM TEXCOCO
DIRECCIÓN

M. en C. Ed. Viridiana Banda Arzate
Subdirectora Académica

PRIMER HOJA DEL ARTÍCULO

Entrenamiento de una red neuronal artificial para detectar intrusos en una red de datos

TRAINING OF AN ARTIFICIAL NEURAL NETWORK TO DETECT INTRUSIONS IN A DATA NETWORK

Luna Domínguez José Ernesto,
Soberanes Martín Anabelem,
Juárez Landin Cristina

Resumen:

Las Redes Neuronales Artificiales (RNA) son técnicas no paramétricas que permiten la resolución de problemas con alto nivel de complejidad, los cuales no son fáciles de solucionar con técnicas tradicionales. Las redes neuronales permiten la obtención de modelos que relacionan conjuntos de variables de entrada con otros de variables de salida.

Distintas herramientas de seguridad informática usadas en la actualidad utilizan sistemas expertos, sistemas basados en conocimientos o redes neuronales artificiales, estas últimas consisten en modelos matemáticos que simulan la función del cerebro humano, las cuales generan mejoras en dicha utilidad. Las RNA representan una mejora en la búsqueda de lograr la verdadera seguridad en redes informáticas.

El gran volumen de información que se necesita para entrenar las RNA dificulta la obtención de resultados positivos, es necesario asegurar que la pérdida de datos sea lo más pequeña posible, lo que tendría como consecuencia, tiempos estables en el entrenamiento así como una complejidad mínima dentro del clasificador neuronal.

En este trabajo se muestran los resultados de la utilización de una red neuronal artificial con un entrenamiento deficiente que tiene por objetivo categorizar los datos de entrada, lo que permite clasificar datos similares en una misma categoría. Este sistema debe agrupar los datos por razón de similitud y asignar un prototipo a cada categoría, los cuales serán utilizados para clasificar nuevos y desconocidos datos o patrones de ataque. Dentro del entrenamiento se realizaron rutinas y subrutinas para lograr el correcto funcionamiento de la RNA en tiempo real, una vez considerado pertinente la finalización de dicho proceso se procedió a buscar la optimización de los resultados mostrados por la red neuronal.

Palabras clave: Red Neuronal Artificial, Entrenamiento, aprendizaje

Abstrac:

Artificial Neural Networks (ANN) is nonparametric techniques that allow solving problems with high complexity, which are not easy to solve using traditional techniques. Neural networks allow obtaining sets of models that relate to other input variables to output variables.

Anexo 7

CARTA DE ACEPTACIÓN

INSTICC
Institute for Systems and Technologies of Information, Control and Communication

ICAART 2015
International Conference on Agents and Artificial Intelligence
http://www.icaart.org/

To whom it may concern,

We are happy to inform that the paper submitted by Jose Ernesto Luna, Anabelem Soberanes Martín and Cristina Juárez Landín to ICAART 2015 with number 140, entitled "System for Intrusion Detection with Artificial Neural Network", has been accepted as a Short Paper, to be presented next January (10 - 12) at Lisbon, Portugal.

All papers accepted to ICAART 2015 were peer reviewed by at least two experts from the international program committee, in a double-blind review process. The paper will be published in the conference proceedings with up to 6 pages, and after being presented at ICAART 2015 it will be included in the SCITEPRESS Digital Library under a specific DOI to be specified after the proceedings are published, and submitted for indexation to Thomson Reuters Conference Proceedings Index, Elsevier Index, DBLP, INSPEC and Scopus.

Best Regards,

Joaquim Filipe
(ICAART Conference Co-chair)