SISTEMA DISTRIBUIDO BASADO EN INFERENCIA PARA LA DETECCIÓN DE INTRUSIONES EN UNA RED DE ÁREA LOCAL.

Felipe Andrés Corredor[#1], César Darío Guerrero[*2]

[*]*Facultad de Ingenierías de Sistemas, Universidad Autónoma de Bucaramanga, Bucaramanga, Colombia*

[1]fcorredor@unab.edu.co

[2]cguerrer@unab.edu.co

***Abstract*— La seguridad de las redes se ve afectada permanentemente por el acceso intrusivo que busca no solo ingresar de forma no autorizada sino generar un ataque luego de la intrusión, los IDS actuales presentan restricciones por complejidad de uso y licenciamiento, lo que dificulta la adopción adecuada por parte de los administradores de seguridad. Este proyecto implementa una solución verificada en el contexto, a través de un sistema de detección de intrusos desarrollado, tomando como referentes los aportes de varios expertos en seguridad y determinando unos patrones de detección iniciales a través de una arquitectura de plugins, que le brindan al sistema una alta tasa de escalabilidad. Aunque en el contexto existen múltiples herramientas que apoyan en la detección de intrusiones, la mayoría de ellas son herramientas privativas, con restricciones de acceso a los códigos fuentes y costosas para algún tipo de organización. Desde esta investigación se diseñó una arquitectura modular distribuida y se desarrolló un DIDS basado en inferencia, para apoyar al sector académico y los administradores de redes en la toma de decisiones y la realización de acciones de control de seguridad en redes de área local.**

***Keywords*— IDS, seguridad en redes, mecanismo de inferencia, sistema distribuido.**

I. INTRODUCCIÓN

Este proyecto surgió de la necesidad que se presenta en el sector académico y profesional de la ingeniería relacionada con tecnologías de información (TI), específicamente en el área de la seguridad informática, ya que a pesar de que existen desarrollos tecnológicos que realizan apoyo en estos campos, se presentan algunas insuficiencias específicas; en el ámbito académico las universidades requieren apoyar los procesos pedagógicos en herramientas a las que se les pueda estudiar detalladamente sus aspectos internos, requiriendo para ello acceso al código fuente y una simplicidad en su diseño pero con alto grado de eficiencia dado por un motor de inferencia propio y documentado, por lo que pocos sistemas de detección de intrusos (IDS) presentan estas características que conjugan las necesidades específicas del sector educativo propio de la región y el país.

Por otra parte en el ámbito corporativo, los profesionales de seguridad informática, requieren herramientas sin restricciones de modificación y distribución en su licenciamiento, que sean fáciles de configurar y realicen una tarea específica de forma óptima, actualmente las pequeñas y medianas empresas de la región no pueden acceder al uso de herramientas muy robustas y complejas, ya que están diseñadas para sistemas muy grandes, además muchas de ellas se distribuyen con hardware y su costo es muy alto.

El problema de la seguridad, específicamente en el acceso intrusivo a las redes informáticas es potencialmente igual a cualquier tipo de organización, tanto grande como mediana y pequeña, pero la capacidad de acceder a la implantación y uso óptimo de esta tecnología en una pequeña y mediana organización no es fácil, actualmente las pequeñas y medianas empresas de la región de la Orinoquía no pueden acceder al uso de herramientas tan robustas y complejas de configurar, privativas, con restricciones de licenciamiento, acceso a los códigos fuentes y con mecanismos que afectan la precisión y el tiempo en la detección.

Desde esta investigación se diseñó una arquitectura modular distribuida y se desarrolló un sistema distribuido para detectar intrusos, los módulos principales son Servidor (Master) y Agentes (Sensores); donde el primero consta de los métodos de Decodificación, Transmisión de Datos, Recepción de Parámetros, Base de datos de Reglas, Inferencia y Envío de Alertas, por su parte el segundo módulo que consta de los métodos de Captura de Tráfico, Colector de Patrones, Configuración, Codificación, Transmisión de Datos y Recepción de Parámetros. Donde estos métodos interactúan de forma iterativa poyados en un motor de inferencia diseñado a partir de conocimiento experto de administradores de redes, donde se identificó información de contexto y requerimientos, para plantear un mecanismo de inferencia que se ajuste en término de rendimiento para la generación rápida de alertas al administrador de la red, vía correo electrónico. Lo que permite apoyar al sector académico y los administradores de redes en la toma de decisiones y la realización de acciones de control en el campo de la detección de intrusos en las redes de área local.

II. ESTADO DEL ARTE

Los sistemas de detección de intrusiones son una importante herramienta en el campo de la administración de sistemas telemáticos, ya que alerta al administrador respecto a un posible conjunto de acciones que atentan contra la

integridad, confidencialidad o disponibilidad de algún recurso de la red.

Según Xiaonan (2010), el IDS monitoriza dinámicamente los eventos que tienen lugar en la red y decide si estos eventos son un síntoma de ataque que constituya un uso ilegitimo del sistema. Las líneas continuas representan secuencias/control de flujo y las discontinuas representan las respuestas a las actividades intrusivas.

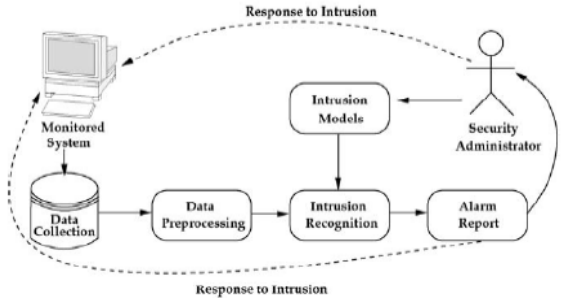

Fig. 1. Estructura de componente de un IDS. Xiaonan (2010)

Arquitectura de un IDS

Según Gulshan (2010), Un sistema de detección de intrusos (IDS) se define como "una tecnología de seguridad eficaz, que puede detectar, prevenir y posibilitar la reacción ante ataques informáticos"; siendo uno de los componentes estándar en infraestructuras de seguridad.

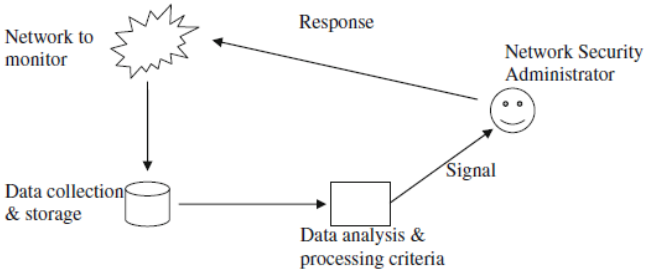

Fig. 2. Arquitectura de IDS. Fuente: Gulshan (2010)

El objetivo principal del IDS es detectar todas las intrusiones de una forma eficiente. Por lo cual se plantea una arquitectura donde modular con una unidad de “Data analysis & processing”, que es equivalente al cerebro del IDS, el cual implementa toda la funcionalidad para detector el comportamiento sospechoso del ataque desde el tráfico de la red.

- **Técnicas para la detección de intrusiones**

Las principales técnicas son las estadísticas, las basadas en el conocimiento y las basadas en inteligencia artificial (IA). En los IDS basados en estadísticas, el comportamiento del sistema se representa desde el punto de vista aleatorio. Por otra parte, los IDS basados en técnicas de conocimiento, tratan de capturar el comportamiento obtenido a partir de datos disponibles en el sistema (las especificaciones del protocolo, instancias del tráfico de la red, etc.) Por último, los IDS basados en técnicas de IA consisten en el establecimiento de un modelo explícito o implícito que permite clasificar patrones.

Según Varun (2009), un sistema de detección de intrusos basado en red consiste en detectar al intruso a partir de los datos que circulan por la red, ya que estas intrusiones ocurren típicamente como patrones anómalos. La razón principal de estas anomalías es por ataques lanzados por hackers externos para obtener acceso no autorizado y hurtar información.

- **Lógica difusa y detección de intrusiones**

Según Gulshan (2010), las técnicas de lógica difusa han sido usadas en el área de la seguridad informática y de redes, especialmente en la detección de intrusiones por dos principales razones. En primer lugar, varios parámetros cuantitativos que se utilizan en el contexto de la detección de intrusos, como; el tiempo de uso de CPU, el intervalo de conexión, etc., pueden ser vistos como variables difusas. En segundo lugar, el concepto de seguridad en sí es difuso. En otras palabras, el concepto de falta de claridad ayuda a suavizar la abrupta separación del comportamiento normal del anormal. Es decir, un punto dado de datos que quedan fuera o dentro de un "Intervalo normal" previamente definido, se considera anómalo o normal respectivamente, independientemente de su distancia al intervalo.

De acuerdo a Norbik (2005) la aplicación de métodos difusos para el desarrollo de IDS tiene algunas ventajas, comparados con el enfoque clásico. Por lo tanto las técnicas de lógica difusa han sido usadas en el campo de la seguridad informática desde hace varios años. La lógica difusa permite cierta flexibilidad para el problema de la incertidumbre que conlleva la detección de intrusiones y brinda una mayor complejidad al IDS. La mayoría de los IDS basados en lógica difusa, requieren de expertos humanos para determinar los conjuntos y reglas difusas, por lo tanto se debe tratar de generar automáticamente las reglas difusas a través de la construcción de un buen clasificador.

El problema de la detección de anomalías, que a su vez conlleva a problemas más específicos, dentro de ellos la detección de intrusiones, es un problema complejo el cual ha llamado la atención de muchos investigadores y el contexto plantea el uso de diferentes disciplinas tales como la estadística, aprendizaje de máquina, minería de datos, y demás técnicas de inteligencia computacional, tal como se observa en el gráfico.

- **Evaluación de desempeño**

Según Xiaonan (2010), La eficacia de un IDS es medida por su habilidad para hacer predicciones correctas. De acuerdo a la predicción realizada por un IDS, son posibles cuatro salidas, mostradas posteriormente, conocidas como la matriz de confusión; Verdaderos negativos, así como Verdaderos positivos corresponden a una correcta actuación del IDS; es decir los eventos satisfactorios, son marcados como normales o ataques, respectivamente los falsos positivos se refieren a eventos normales que se predicen como ataques, los falsos negativos son eventos de ataque que se detectan incorrectamente como normales.

Basados en la matriz de confusión, una evaluación numérica puede ser aplicada sobre las siguientes características para cuantificar el rendimiento del IDS:

- Tasa de verdaderos negativos: (*True negative rate* - TNR), también conocida como especificidad.
- Tasa de verdaderos positivos: (True positive rate - TPR), también conocida como tasa de detección (*detection rate* - DR) o sensibilidad..
- Tasa de falsos positivos (*False positive rate* - FPR), también conocida como tasa de falsas alarmas (*false alarm rate* - FAR).
- Tasa de falsos negativos (*False negative rate* - FNR).
- Precisión.

Las métricas más usadas en estos casos son la tasa de detección (DR) junto con la tasa de falsas alarmas (FAR). Por lo tanto se puede concluir que un buen IDS debe tener un alto DR y un bajo FAR. Además se pueden aducir otras combinaciones usadas comúnmente incluyen precisión sensibilidad y especificidad.

III. METODOLOGÍA

En la metodología, se planteó un proceso secuencial a través de las siguientes fases desde un análisis de datos y requerimientos de los actores involucrados, escenarios y contexto, hasta el diseño, implementación, pruebas (validación), la integración, y ajuste de la solución propuesta. Las fases acoplaron adecuadamente el cumplimiento de los objetivos durante el proceso investigativo y el desarrollo del software en el que se basa la hipótesis de este proyecto, por lo que se detallan a continuación:

- **Análisis y comparación de técnicas de inteligencia computacional**

  En esta etapa se procedió a construir un cuadro que permite contrastar los diferentes mecanismos basados en inteligencia artificial para detección de intrusos. Las actividades contempladas se enfocaron en la búsqueda en bases de datos digitales y literatura específica, selección de los artículos más relevantes, análisis de la literatura recopilada y finalmente la construcción del cuadro de contraste.

  Las técnicas revisadas fueron clasificadas por el tipo de procesamiento, eficiencia en predicción, ventajas y desventajas.

- **Definición de funcionalidades del IDS**

  Para la definición de las funcionalidades se establecieron los requerimientos del sistema de detección de intrusos basado en red orientado a operar en una red de área local. Las actividades propias de esta etapa fueron: la revisión de contexto a nivel técnico y funcional, realización de visitas de observación a escenarios de red reales, realización de entrevistas a expertos administradores de infraestructura de TI, elaboración de un listado de funcionalidades y restricciones del sistema de detección y finalmente la elaboración de los diagramas de casos de uso que resumen gráficamente las funcionalidades.

  Se diseñó un instrumento de medición de parámetros propios de los expertos administradores de redes y seguridad en las organizaciones, basado en un sistema en la web llamado *surveymonkey*, se indago con 20 expertos de todo el país gracias al aporte de la Asociación colombiana de ingenieros de sistemas – ACIS, quien facilito a través de su lista de correo “segurinfo”, el acceso a la plataforma surveymonkey.

  El instrumento contempló 10 aspectos relacionados con la caracterización de la organización y los administradores de seguridad, junto con la identificación de patrones de intrusión relevantes para los sistemas de detección de intrusos.

  Acorde al marco teórico evidenciado para los sistemas de detección de intrusiones y los planteamiento de los expertos en seguridad, se modelaron los casos de uso, respectivos al IDS de red Local, identificando tres actores (Agente, Servidor), de los cuales dos son módulos de software y el otro es actor humano (CISO). La figura 3 expone el caso de uso respecto a la funcionalidad del sistema relacionando los procesos del IDS con los actores identificados: El CISO, debe ser capaz de iniciar los módulos tanto de servidor como de agente, dentro de los cuales, el agente debe iniciar la colección de datos de tráfico y proceder a la detección de patrones. Estando el servidor iniciado, está en capacidad de recibir automáticamente la información de patrones detectados por el agente y decodificarlo (preprocesarlo), para analizarlo desde un mecanismo de inferencia, con el objeto de encontrar alguna intrusión, que si es el caso debe ser reportada por un correo electrónico al CISO.

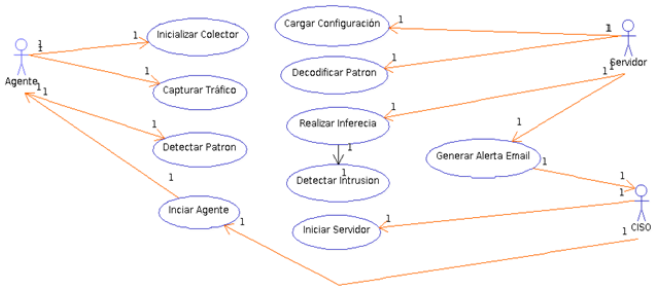

Fig. 3 Arquitectura y módulos del IDS desarrollado

- **Desarrollo de Sensores**

En esta etapa se Implementaron sensores por software para capturar y clasificar el tráfico de la red, lo que permitió disponer de la fuente de datos para el motor de inferencia. Las actividades propias se orientaron a: determinación del tipo de tráfico que para ser evaluado por el mecanismo de inferencia, selección de los parámetros de encabezado de paquete para ser analizados y la construcción de módulos de software que permitieran capturar el tráfico.

El desarrollo se basó en la librería jpcap de java que a su vez depende de la librería libpcap 0.8, que es un *framework* de bajo nivel para monitorización de tráfico de redes. Se distribuye como una API independiente para permitir la portabilidad de aplicaciones.

A su vez se describe la API jpcap, que es una envoltura de libpcap para desarrolladores JAVA, ampliando las posibilidades de libpcap desde este lenguaje.

Componente	Descripción
jpcap.JpcapWriter.*;	Esta clase permite el volcado de datos a una fuente de recepción de flujo de datos, que en el caso de este proyecto fue un archivo plan, usando el método openDumpFile.
jpcap.NetworkInterface.*;	Permite la manipulación de las interfaces de red para inicializar el capturador en una o varias de estas.
jpcap.packet.*;	A través del método getPacked(), se puede disponer de cada paquete capturado para su disección y procesamiento, lo que permite que a partir de estos, se pueda detectar los patrones de intrusión por el agente colector.

Tabla 1. Descripción del uso de componentes de la librería JPCAP.

Los sensores para captura de tráfico, fueron implementados completamente en jpcap, siguiendo un proceso secuencial e iterativo, tal como se muestra en la figura 4; que inicia con la obtención de un listado de la interfaces de red del sistema, con el método *getDevicesList()*, luego se determina la interfaz de la maquina donde se recogerá el tráfico con *openDevice()*, Se debe determinar una fuente de almacenamiento a través de *openDumpFile()*, que en este caso es un archivo denominado "captura.ids.cap", se establece el filtro necesario sobre el tráfico con *setFilter()*. En este momento ya el Sensor puede iniciar la captura del tráfico con *getPacket()* y luego es enviado al archivo seleccionado con el método *writePacket()*. El Agente procederá luego a codificar el mensaje, encontrar patrones y hacerlos llegar al motor de inferencia para su análisis.

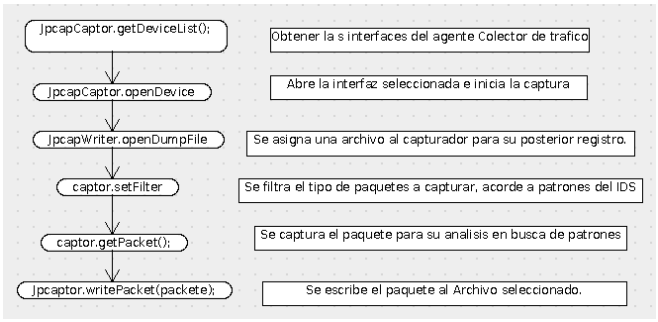

Fig 4. Esquema para captura de datos por el agente/Sensor.

- **Desarrollo del mecanismo de inferencia**

Los patrones planteados inicialmente son 16, implementables a través de *plugins* que se puedan incorporar al IDS para aumentar su capacidad de detección:

No.	PATRON
1	Cambios bruscos en niveles de tráfico en intervalos cortos de tiempo..
2	Detección de IPs desconocidas intercambiando trafico.
3	Cambio de IPs contra dirección MAC.
4	predominar una dirección y puerto destino, mientras que el origen suele ser distribuido
5	Picos inusuales de tráfico a una sola dirección desde una típica distribución de orígenes
6	Sondeo de múltiples puertos destino dentro de un conjunto pequeño de direcciones
7	Tráfico desde un origen a múltiples destinos.
8	Envío del mismo contenido de datos, de forma repetitiva.
9	Trafico malicioso http ""transversal directory"
10	Trafico de falsos positivos generados específicamente para hacer que el IDS genere alertas. Ataque focalizado al IDS para DoS por alertas.
11	Inundacion ICMP
12	La versión de IP. Checksum incorrecto. Tamaño total del datagrama es menor que el de la cabecera IP. Opciones incorrectas.
13	Trafico con TTL corto, tal que no alcance a analizar el router.
14	Tráfico con opción strict source routing corta. Analizar!
15	Tráfico con opción timestamp
16	ataque de denegación de servicio al IDS.

Tabla 2. Patrones incidentes implementables en IDS.

Para el proceso de desarrollo se usó el IDE Eclipse, sobre el sistema Debian GNU/Linux 6.0r2 Squeeze, el

lenguaje de programación Java con el compilador JDK 1.7 y el motor de base de datos postgresql 9.1.

Se usaron componentes externos como la librería de captura de tráfico jpcap (que se basa en libpcap) y el *plugin* de eclipse ERMaster para el desarrollo del modelo entidad relación, a su vez que la herramienta de modelado ArgoUML.

- ***Pruebas y Entrega de Informes***

  En esta última etapa se procedió a probar la operación del sistema desarrollado en una red de pruebas (*testbed*), que permitió evidenciar el correcto funcionamiento a nivel lógico y funcional del IDS en general. Se definió para esto un escenario de pruebas en una red de datos acondicionada para ello, se realizó la simulación de una intrusión, luego la Instalación del sistema de detección, la ejecución del proceso de detección en una red de datos real y el análisis de los resultados

IV. ANÁLISIS DE LA APLICACIÓN DEL INSTRUMENTO

La muestra proviene de la lista de expertos en seguridad informática "segurinfo" de la Asociación Colombiana de Ingenieros de Sistemas – ACIS, donde el instrumento indagaba sobre los datos del administrador y la organización, pero de forma opcional por razones de confidencialidad y tranquilidad al diligenciar el instrumento. El análisis sobre los resultados del estudio, evidencia aspectos relevantes en la forma de cómo se aplica la seguridad en las organizaciones respecto a la supervisión de las redes para prevenir y/o determinar la existencia de elementos intrusivos en las redes corporativas

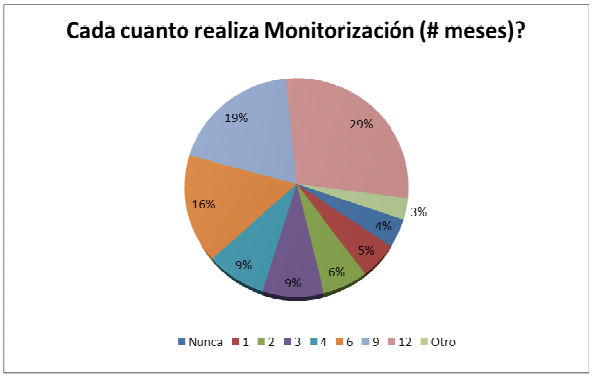

Fig. 5 Periodos de realización de monitorización en empresas.

De acuerdo al figura 5, se puede percibir que prácticamente la mitad de las empresas no realizan monitorización antes de nueve meses, lo que implica fallas en administración, pues según Villar (2010) "La monitorización de sistemas es la encargada de supervisar continuamente los diferentes recursos y servicios de la empresa para garantizar el nivel de disponibilidad requerido y en caso de un posible fallo alertar a los administradores para que lo solucionen". Lo complicado del asunto radica en que un proceso de ataque focalizado puede demorar unas pocas semanas y no es solo detectarlo de forma tardía sino que nunca sería detectado, por lo cual se permanecería con la vulnerabilidad.

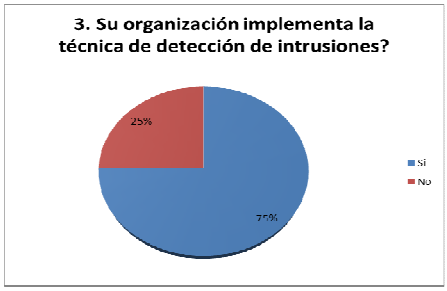

Fig. 6 Implementación de la detección de intrusiones en empresas

El 75% de los administradores de red implementan la técnica de detección de intrusos en sus organizaciones, ver figura 6, lo que supone que no se está realizando la tarea de supervisión completa, la detección de intrusos y la monitorización son tareas complementarias, no excluyentes. Con la monitorización se puede mantener una tranquilidad sobre el performance de la red, cuando las tasas promedio de medición (tráfico total, por protocolo, procesamiento, memoria, etc) se mantienen o tienen una explicación válida para el CISO, por su parte cuando se presenta anormalidad en estos parámetros de la red, es el IDS quien entra a determinar con precisión cual es el evento generador del incidente de Intrusión y alertar en tiempo real, para que se tomen las acciones correctivas.

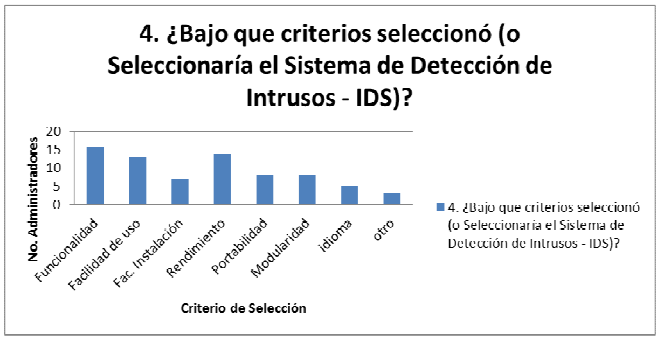

Fig. 7 Criterios de Administradores para selección de los IDS

El administrador de la red busca en el IDS, funcionalidad, rendimiento y facilidad de uso al momento de escoger un IDS, sin embargo, por los resultados presentados en el figura 7, la mayoría de CISOs se apoyaría sin problemas de herramientas en otro idioma, pero es de resaltar que un porcentaje importante (25%) no seleccionaría un IDS diferente al español. La funcionalidad como aspecto más relevante con 80%, junto con rendimiento 70%, es una tendencia muy válida en la selección de herramientas de seguridad, "el responsable de la seguridad debe disponer de herramientas muy funcionales con

alto grado de rendimiento", pero esto contrasta con que la facilidad de uso tiene un importante 65% de preferencia, lo que indica que más de la mitad de los CISOs estarían dispuestos a reconsiderar sacrificar funcionalidad por facilidad de uso.

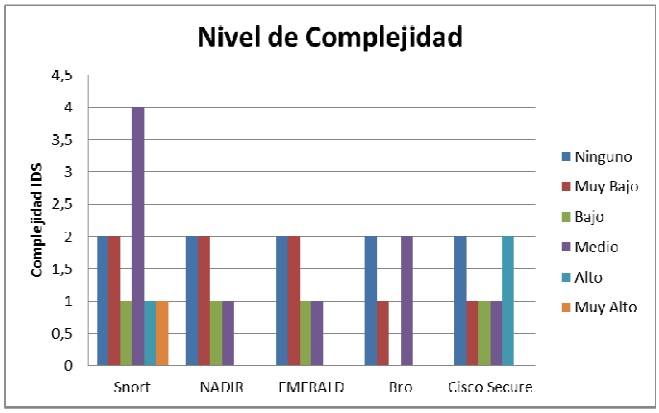

Fig. 8 Complejidad de instalación/uso de IDS

En el contexto, la industria dispone de múltiples herramientas IDS, donde se evidenció (ver figura 8) que el sistema SNORT es el más usado con el 55% y su grado de complejidad no es tan alto, solo el 18% creen que es de complejidad alta o muy alta. A pesar de esto, comparado con las otras herramientas (NADIR, EMERALD, Bro y Cisco Secure), hay una percepción de ser el más complejo de utilizar, un 36% plantea que es de complejidad media, incluso para algunos administradores es una herramienta de complejidad muy alta. Las demás herramientas de contexto no presentan complejidad en su manejo. Esto se explicaría porque al ser el más usado, los CISOs esperan este aspecto "el más usado, el menos complejo…" como valor agregado.

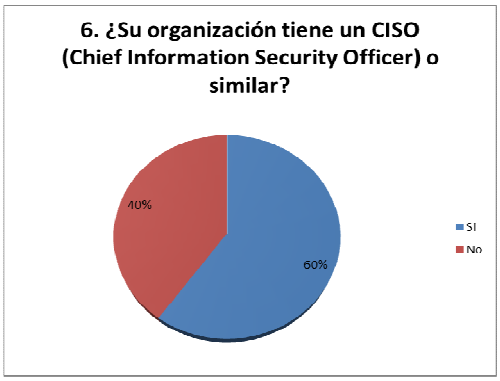

Fig. 9. Porcentaje de Organizaciones que definen CISO en su personal.

Más de la mitad de las empresas (60%) disponen de un cargo denominado CISO, que tiene como función principal, mantener todos los procesos de seguridad, lo que significa que el área administrativa y toma de decisiones han comprendido la relevancia de los procesos de seguridad corporativa a través de la definición del cargo. La gestión de la red ha incluido tradicionalmente la gestión de la seguridad dentro de sus funciones, pero esta área ha cobrado mayor relevancia actualmente por la criticidad de misión en la protección de la información (como principal recurso de la organización). Para cumplir con su objetivo, dentro de las funciones de un CISO se encuentran, la monitorización de los recursos, la detección de intrusiones, realización de auditorías, gestión de incidentes, cómputo forense, etc. Lo que no sería realizado adecuadamente por el jefe de sistemas quien, debe mantener la red funcionando sin interrupciones y brindando soporte permanente y es precisamente por esto que los cargos tienden a redefinirse.

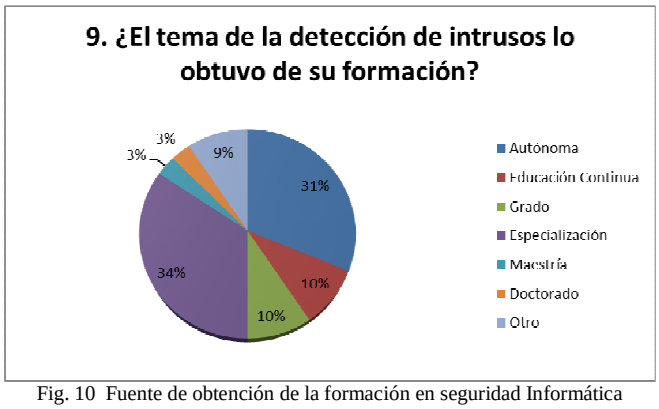

Fig. 10 Fuente de obtención de la formación en seguridad Informática

Los programas de posgrado son los que más contribuyen a la formación de profesionales en esta área, por lo tanto, son las universidades las convocadas a brindar los elementos adecuados a los profesionales en este campo, a su vez la formación autónoma, es el segundo ítem de formación, ya que se realiza como alternativa o complemento a través de multiplicidad de recursos colaborativos que cambia y se desactualizan rápidamente, como boletines, foros, listas de correo especializadas y sitios CVE (*Common Vulnerabilities and Exposures*) o CERT(*Computer Emergency Response Team*). En algunos casos se busca la certificación en algunos de los campos de la seguridad informática, como se relacionan en la siguiente tabla:

NOMBRE	DESCRIPCIÓN	EMITIDA POR	LOGO
CISSP	Certified Information Systems Security Professional	ISC2(International Information Systems Security Certification Consortium, Inc)	
CISA	Certified Information Systems Auditor	ISACA (Information Systems Audit and Control Association)	
CRISC	Certified in Risk and Information Systems Control	ISACA (Information Systems Audit and Control Association)	
CEH v7	Certified Ethical Hacker	Ec-Council. Consorcio Internacional de Consultas de Comercio Electrónico.	

CFE	Certified Fraud Examiner	Association of Certified Fraud Examiners (ACFE)	
CIA	Certified Internal Auditors	The Institute of internal auditors (IIA)	

Tabla 3. Posibles certificaciones en Seguridad de los Información para CISOs.

Se determinaron 16 patrones de intrusión que fueron sometidos a revisión por parte de los expertos en seguridad, de lo cual se puede observar:

Que los patrones P4, P5, P7 y P9, presentan son catalogados como fuentes primarias para determinar la presencia de un intruso en la red (Alto):

- predominar una dirección y puerto destino, mientras que el origen suele ser distribuido,
- Picos inusuales de tráfico a una sola dirección desde una típica distribución de orígenes
- Tráfico desde un origen a múltiples destinos.
- Trafico malicioso http "*transversal directory*"

Por otra parte, los patrones P2 y P3, además de ser fuentes primarias, influyen ampliamente en complementar otros patrones para determinar la presencia del intruso en la red.

- Detección de IPs desconocidas intercambiando trafico.
- Cambio de IPs contra la dirección MAC, registrada por el administrador o usada regularmente.

V. RESULTADOS

A través de la presente investigación se logró desarrollar un sistema distribuido que basado en software libre y en un mecanismo de inferencia es capaz de apoyar la detección de intrusos en una red de área local.

Como primer aspecto se realizó un análisis comparativo mecanismos de inferencia y herramientas IDS de contexto, que se resume en las siguientes tablas demostrativas:

Mecanismo de Inferencia	Estructuras	Eficiencia en predicción	Capacidad de adaptación	Tipo de procesamiento	Usos recomendados	Ventajas	Desventajas
Minería de datos	Conjuntos de datos.	Alta	Media	Estadística	Predicción Clasificación Segmentación	Alta precisión de actividad maliciosa.	El ajuste de parámetros y métricas es costoso en tiempo de desarrollo y obtención de conocimiento.
Lógica difusa	Sensores que miden el entorno por aproximación e incertidumbre.	Alta (Veloz)	Alta, mejora cuando es hibrido.	Conocimiento – Estadística	Predicción Detección Control Optimización	Alta precisión de actividad maliciosa. Robustez, flexibilidad y escalabilidad. No requiere conocimiento previo acerca de la actividad normal.	Requiere conocimiento experto en cada módulo de detección. El ajuste de parámetros y métricas es costoso en tiempo de desarrollo y obtención de conocimiento.
Algoritmo genéticos	Genético-molecular. Inspirado en biología evolutiva.	Media (iterativa)	Alta	Máquinas de aprendizaje	Diseño Topológico, Calibración Análisis genético Optimización	Alta flexibilidad y adaptabilidad.	Alto consumo de recursos computacionales. Tiempos de respuesta altos. Alta dependencia en supuestos sobre el comportamiento aceptado para el sistema.
Redes neuronales artificiales	Basada en el sistema nervioso: red de Neuronas.	Media (iterativa) – retroalimentación	Alta	Máquinas de aprendizaje	Predicción, Optimización	Alta flexibilidad y adaptabilidad.	Alto consumo de recursos computacionales. Tiempos de respuesta altos. Alta dependencia en supuestos sobre el comportamiento aceptado para el sistema.

Redes bayesianas	Modelo probabilístico, conjunto de variables. Relaciones probabilísticas sobre variables.	Alta precisión en detección.	Media	Estadística	Predicción, Detección	*Alta precisión de actividad maliciosa. *Robustez, flexibilidad y escalabilidad. *No requiere conocimiento previo acerca de la actividad normal.	Requiere conocimiento experto en cada módulo de detección. El ajuste de parámetros y métricas es costoso en tiempo de desarrollo y obtención de conocimiento.
Arboles de decisión	nodos internos, nodos de probabilidad, nodos hojas y arcos	Alta (poco flexible)	Baja	Conocimiento – Estadística	Predicción, Detección	Robustez, flexibilidad y escalabilidad.	El ajuste de parámetros y métricas es costoso en tiempo de desarrollo y obtención de conocimiento.

Tabla 4. Análisis Comparativo sobre mecanismos de inferencia viables para IDS

NOMBRE	Mecanismo de detección	Arquitectura	Reportes –Toma de decisiones	Velocidad de detección	LICENCIA
NADIR-	Reglas / Firmas	Distribuida	-	Tiempo real	Privado. Gov de Estados Unidos. Los Alamos National Lab., NM (USA)
EMERALD)	Reglas / inferencia probabilística	Distribuida	-	Tiempo real	Comercial, a través de emerald-release@sdl.sri.com
Bro Network Security Monitor	análisis de reglas y aplicación de políticas	Centralizado	Línea de comandos. Herramientas externas.	Tiempo real	BSD
Snort	Reglas /Firmas	Distribuido	Web, Solo con Herramientas externas.	Tiempo real	GNU General Public License (GPL) Non-Commercial Use License for the Proprietary Snort® Rules
Cisco Secure	detección de patrones de estado	Centralizado / Distribuido	Web.	Tiempo real	Comercial

Tabla 5. Análisis Comparativo sobre herramientas IDS usadas en el contexto.

- **Arquitectura del IDS desarrollado**

Acorde a la arquitectura diseñada para el sistema, teniendo en cuenta los elementos modulares identificados en el contexto, se obtuvo un sistema distribuido en arquitectura de dos niveles (Un solo Servidor y múltiples Agentes colectores):

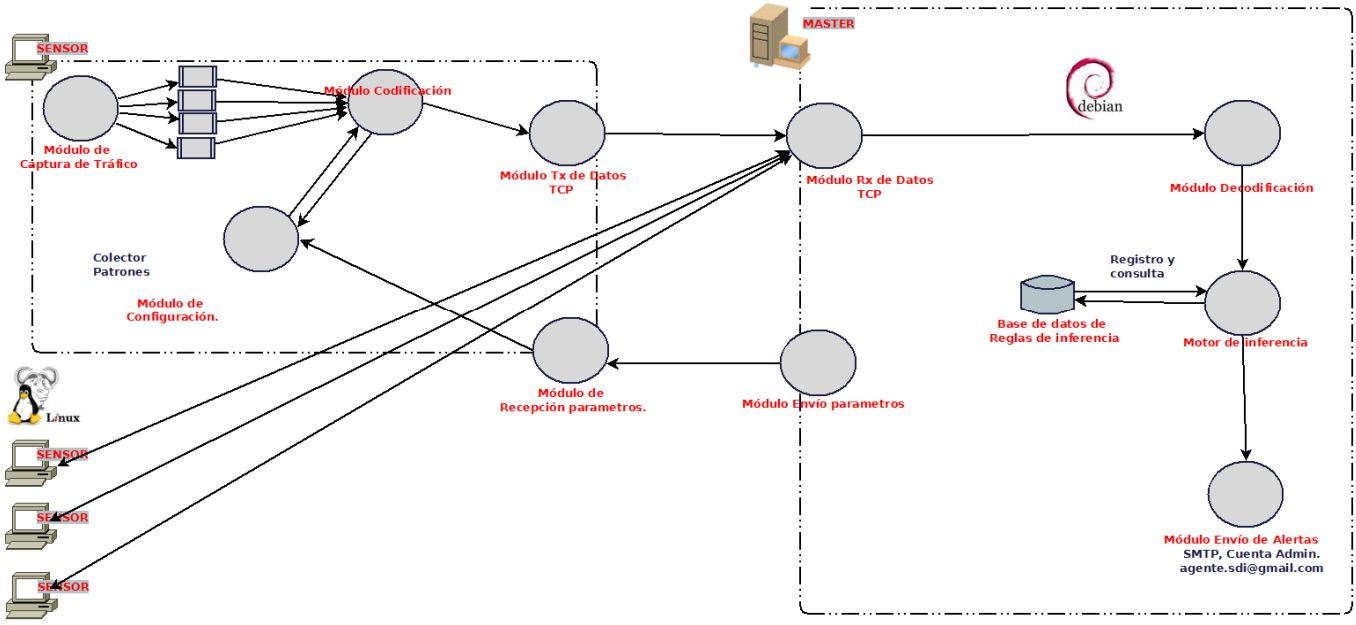

Fig. 11. Arquitectura del IDS desarrollado

De acuerdo al análisis de contexto y análisis de requerimientos funcionales se estableció la arquitectura del sistema de detección; cliente/servidor, modular para mayor flexibilidad y escalabilidad:

- **Módulo Master (Servidor)**

PROCESO	DESCRIPCIÓN
Decodificación	Se desensamblan los datos recibidos de cada Sensor (Agente) acorde al protocolo definido: @@ para iniciar una trama de datos de un sensor, | para separar datos de cada método colector de patrones y ; para separar los datos de cada parámetro interno del patrón.
Tx de Datos	Se realiza un envío de datos codificados por un socket TCP hacia el puerto remoto aleatorio del cliente.
Rx de Parámetros	Se realiza una recepción de datos enviados por el agente, codificados por un socket TCP hacia el puerto local 2603 del servidor.
Base de datos de Reglas	La base de datos POSTGRESQL consta de 16 tablas, donde se soportan el funcionamiento del sistema y el motor de inferencia. Específicamente en las tablas incident, vulnerability y vulnerabilityincident, esta última relaciona a través de reglas.
Inferencia	El mecanismo de inferencia es un mecanismo estadístico por correlación de reglas basadas en la incidencia generada por patrones de intrusión. Para que se considere la existencia de una vulnerabilidad del sistema por intrusión se analiza el(los) patrón(es) y se determinan de acuerdo a la incidencia (peso) que estos tengan sobre la vulnerabilidad.
Envío de Alertas	Una vez la vulnerabilidad de intrusión es detectada por el motor de inferencia, se invoca una clase que realiza un envío de email a través de una cuenta agente.sdi@gmail.com a la cuenta del administrador con el asunto "INCIDENT REPORT SYSTEM"…

Tabla 6. Descripción de funcionalidad de módulo Master

- **Módulo Agente (Sensor)**

PROCESO	DESCRIPCIÓN
Captura de Tráfico	A través de la librería jpcap, se recoge el tráfico a nivel de red (Modelo de referencia OSI), durante una ventana de tiempo y se archiva en captura.ids.cap.
Colector de Patrones	Carga el archivo captura.ids.cap en un Objeto ArrayList y procede a métodos propios de la clase, se realiza la decodificación e identificación de la presencia de patrones a ser reportados al servidor.
Configuración	La configuración se carga desde un archivo xml, llamado server.xml en la raiz de ejecucion del Servidor, se invocan los paquetes de Java (javax.xml.parsers.DocumentBuilderFactory, org.w3c.dom.Document y org.w3c.dom.Element).
Codificación	Se definió un sistema de codificación propio donde se utilizan símbolos específicos para organizar la información a transmitir.
Tx de Datos	Se realiza un envío de datos codificados por un socket TCP hacia el puerto remoto 2603 del servidor.
Rx de Parámetros	Se realiza una recepción de datos codificados por un socket TCP desde el puerto remoto 2603 del servidor.

Tabla 7. Descripción de funcionalidad de módulo Agente.

VI. TRABAJOS FUTUROS

- Se espera abordar el análisis de la integración de mecanismos de inferencia (híbridos) que permitan mejorar la precisión del sistema.

- Diseño e implementación de mecanismo automático de toma de decisiones que brinde características de IPS (*Intrusion Prevention System*).

- Análisis de resultados y generación de reportes analíticos gráficos y tabulados, predictivos, basados en históricos, desde la minería de datos.

- Ya que el IDs es una herramienta de apoyo, para la eficiencia de las alertas que se envían por email, se debe sincronizar la cuenta de correo en un dispositivo móvil (con plan de datos) y pueda ser alertado en tiempo real.

RECONOCIMIENTOS

Asociación colombiana de Ingenieros de Sistemas (ACIS) por permitir a través de su lista de correo "segurinfo" el contacto con expertos en administración de redes y seguridad en redes.

Docentes de la Universidad Autónoma de Bucaramanga UNAB, por sus aportes y retroalimentación durante el proceso de desarrollo.

.

Grupo de Investigación "Tecnologías de Información" de la Universidad Autónoma de Bucaramanga UNAB.

REFERENCIAS

[1] Chenfeng Vincent Zhou, Christopher Leckie y Shanika Karunasekera. (2010). A survey of coordinated attacks and collaborative intrusion detection. Computer & Security. Volumen 29. Páginas 124-140.

[2] Varun Chandola, Arindam Banerjee y Vipin Kumar. 2009 . Anomaly Detection: A Survey. ACM Computing Surveys, Vol. 41, No. 3, Article 15. Páginas 1-58.

[3] Norbik Bashah Idris y Bharanidlran Shanmugam. 2005. Artificial Intelligence Techniques Applied to Intrusion Detection. En IEEE Indicon 2005 Conference. Chennai, India. Páginas 52-55.

[4] Gulshan Kumar, Krishan Kumar y Monika Sachdeva. 2010. The use of artificial intelligence based techniques for intrusion detection: a review. Artificial Intelligence Review. Volumen 34. Número 4. 369-387.

[5] Ajith Abrahama, Ravi Jainb, Johnson Thomasc, Sang Yong Hana. 2007. D-SCIDS: Distributed soft computing intrusion detection system. Journal of Network and Computer Applications. 30. Páginas 81-98.

[6] Shelly Xiaonan Wu y Wolfgang Banzhaf. 2010. The use of computational intelligence in intrusion detection systems: A review. Applied Soft Computing. Volumen 10. Páginas 1-35.

[7] Gaurang Panchal, Parth Shah, Amit Ganatra y P Kosta. 2010. Unleashing Power of Artificial Intelligence for Network Intrusion Detection Problem. International Journal of Engineering Science and Technology. Volumen 2. Número 10. Páginas 5221-5230.

[8] Martin B y Molina Sanez. 2007. Redes Neuronales y Sistemas borrosos (3ª ed.). Mexico: Alfaomega.

[9] ACIS - Asociación Colombiana de Ingenieros de Sistemas, C. J. (2011). III encuesta latinoamericana de la seguridad de la información - 2011. Recuperado de http://www.acis.org.co/fileadmin/Base_de_Conocimiento/XI_JornadaSeguridad/Presentacion_Jeimy_Cano_III_ELSI.pdf.

[10] Welicki León, Cueva Lovell Juan. (2006). Una plataforma basada en sistemas multiagentes y servicios Web para monitorización de aplicaciones en entornos heterogéneos, Pág. 6. Universidad Pontificia de Salamanca, Universidad de Oviedo. España. Disponible en: http://www26.brinkster.com/lwelicki/articles/DESMA-LeonWelicki(final).pdf