UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Análisis y Configuración del Sistema de Detección de Intrusos Snort en la Universidad de las Ciencias Informáticas

Tesis de Diploma para optar por el Título de Ingeniería en Ciencias Informáticas

Autores

Yevgeni Chirino Horta

Dayrena Díaz Cárdenas

Tutor

Eduard Palomo Gene

Consultante

Orestes Rodríguez Morales

Ciudad de La Habana, Junio 2007

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

_____	_____	_____
Firma del Autor	Firma del Autor	Firma del Tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado: **"Análisis y Configuración del Sistema de Detección de Intrusos Snort en la Universidad de las Ciencias Informáticas"**, fue realizado en la Universidad de Ciencias Informáticas (UCI). Este centro considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

Totalmente
Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta Universidad los beneficios siguientes:

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____

_____ _____
Representante de la entidad Cargo

_____ _____
Firma Cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Análisis y Configuración del Sistema de Detección de Intrusos Snort en la Universidad de las Ciencias Informáticas

Autores: Dayrena Díaz Cárdenas y Yevgeni Chirino Horta

El tutor del presente Trabajo de Diploma considera que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingenieros Informáticos; y propongo que se le otorgue al Trabajo de Diploma la calificación de _____.

_____ _____

Firma Fecha

"Nuestra recompensa se encuentra en el esfuerzo y no en el resultado. Un esfuerzo total es una victoria completa."

Mahatma Gandhi

DEDICATORIA

A mis madres, infinitamente les dedico este triunfo de mi vida, por ser mis fieles amigas y consejeras. Y porque este es también su sueño.

A mi hermano Fredyc, por apoyarme siempre y confiar en mí.

A mi abuela Isabel, por su preocupación y ayuda sin medida.

A toda mi familia, por depositar en mí toda la confianza que un ser humano puede llevar consigo.

A mis amigas del alma Mavis, Hildelisa y Daimara, por saber soportar mis malos humores, por darme el aliento cuando me faltaba y pensaba que no podía seguir.

Dayrena

Dedico este trabajo de curso, principalmente a mis padres que tanto se han esforzado para que yo pueda seguir adelante y que me han ayudado mucho a alcanzar mi meta.

A mis hermanos: Pedro, Alberto y Alfredo por ayudarme cada vez que lo necesité.

A mi mejor amigo: Alain, por sus consejos…

A mis queridas amigas: Dayami, Lidisley y Ianabel, por estar siempre conmigo, en las buenas y en las malas.

En general, a todas las personas que me quieren y que en un momento determinado aportaron también su granito de arena para que hoy yo me pudiera graduar.

Chirino

AGRADECIMIENTOS

No es tarea fácil poder escribir en un papel a todos los que transitaron contigo en los incontables momentos felices y no tan felices a través de toda una vida. Si siembras frutos que perduren, al menos te queda el consuelo de agradecer a todos aquellos que caminaron siempre junto a ti, en nuestro caso nos sentimos inmensamente dichosos y agradecemos de la forma más grandiosa posible a nuestros creadores en todo sentido: nuestros padres.

Quisiéramos expresar nuestras más sincera admiración y agradecimiento a todos los profesores que aportaron todo su esfuerzo en la preparación de quienes somos hoy en día. Sin olvidarnos jamás de nuestros familiares que siempre juntos nos brindaron todo el universo de amor y apoyo incondicional que nos hacía falta para no claudicar en nuestro empeño.

A todos nuestros amigos que nunca nos dieron la espalda y siempre tuvieron una sonrisa en sus rostros a la hora de contar con ellos, nos vienen a la mente gente linda como Mavis, Hildelisa, Daimara, Liuris, Orestes, Alain, Dayami, Ianabel y Lidisley.

Queremos destacar que no por dejar de mencionar a muchas de esas personas que contribuyeron de una forma u otra con nuestro desarrollo, son menos importantes. A todos ustedes MUCHAS GRACIAS. Esperamos humildemente que lo que se expresa en este papel les llegue a todos y toque de muy cerca su corazón.

RESUMEN

En la presente tesis se realiza un estudio de la tecnología de Detección de Intrusos en el campo de la seguridad informática. Como resultado de la investigación se propone, con el objetivo de elevar la seguridad de la red UCI, la configuración en la Universidad de las Ciencias Informáticas del Sistema de Detección de Intrusos Snort el cual es uno de los más populares y basado en Software libre que tiene la ventaja de funcionar bajo gran variedad de plataformas y utiliza un lenguaje flexible de reglas para describir el tráfico de red que debe recoger o dejar pasar.

La configuración del Snort que se propone será capaz de analizar en tiempo real el tráfico proveniente de un segmento de red, y detectar anomalías así como cualquier tipo de uso indebido que esté sucediendo en la misma. Adicionalmente puede ser capaz de tomar decisiones en función de la anomalía detectada.

El trabajo analiza los aspectos teóricos más importantes relacionados intrusiones, tipos de sistemas de detecciones de intrusos, sus clasificaciones, requisitos, funcionalidad, así como beneficios y riesgos que conllevan el uso de los IDS, en específico del Snort y la importancia y necesidad de la configuración del IDS-Snort en la Universidad de Ciencias Informáticas. Se han utilizado diferentes métodos científicos como el analítico-sintético, la observación científica y la experimentación; la técnica de muestro utilizada es la de muestreo por conglomerado.

La investigación realizada se basa en que la correcta configuración de IDS-Snort en la red- UCI permitirá que el nivel de seguridad de la red aumente lo cual se tratara de demostrar en el desarrollo del presente trabajo. Es por ello que tiene gran repercusión social ya que intentará solucionar un problema que agobia no solo a las universidades sino también a las distintas organizaciones del país.

ÍNDICE

INTRODUCCIÓN

La seguridad en redes de computadoras ha pasado de ser una simple preocupación de unos pocos a una difícil realidad. En nuestros días personas con escasos conocimientos en materia de la informática son capaces de atacar las redes ocasionando perdidas que oscilan entre una simple molestia y millones de dólares en tiempo y recursos. Esto ocurre debido al desarrollo de herramientas informáticas de fácil acceso para detectar redes con vulnerabilidades. Todos vemos a diario los constantes ataques que sufren los servidores de las compañías mas importantes (Microsoft, Yahoo...), pero nadie está a salvo de ellos. Aunque la información de un usuario final puede ser inútil, su ordenador se puede convertir en el origen de un ataque a un tercero, de forma que el verdadero atacante desaparece entre las sombras. Existen algunas herramientas que son capaces de coordinar ataques de DoS entre múltiples maquinas afectadas, de forma que el hacker solo debe infectar unos cuantos ordenadores de usuarios finales (débilmente defendidos) y lanzar el ataque desde ahí. Ni siquiera tiene que estar conectado a Internet cuando el ataque se realiza. En la actualidad existe una larga relación de programas dedicados a la seguridad, destacándose los siguientes por su utilidad y popularidad:

- Antivirus
- Monitores de red y sistemas
- Detectores de vulnerabilidades
- Analizadores de *logs*
- Proxies
- Cortafuegos
- IDS (Intruder Detection Systems)

La solución de muchos usuarios es poner un Firewall. Este Firewall cierra todos aquellos puertos (servicios) que no usa, reduciendo de esta forma la posibilidad de ataque. Pero, ¿es esto suficiente? La respuesta es no. El Firewall representa una puerta, que prohíbe el paso a todos aquellos servicios no autorizados, pero que deja pasar aquellos que el usuario/s detrás del Firewall necesita usar. Y ahí esta el problema. Aunque solo se permitan los servicios básicos y teóricamente seguros, existen agujeros que se pueden aprovechar, y ante los que el Firewall no puede hacer nada (si es un servicio autorizado, simplemente lo deja pasar). Todo esto suponiendo, claro esta, que la integridad del Firewall no haya sido comprometida. ¿Qué se puede hacer a continuación?

Al igual que una puerta no es suficiente para proteger una casa, y se instala un sistema de alarma para detectar cuando un ladrón ha conseguido pasar la puerta, podemos pensar en un sistema similar a la hora de hablar de una red informática. El Firewall se puede ver comprometido, y es necesario tener algún sistema que nos permita detectar esta situación. Además, dado que todos los servicios tienen agujeros potenciales de seguridad, es de esperar que muchos ataques pasen por el Firewall sin que este sea capaz de detectarlo. Por tanto, es necesario algo más, un vigilante, un Sistema de Detección de Intrusos (IDS).
En particular los Sistemas de Detección de Intrusos han constituido un campo de investigación activo desde hace dos décadas.
A pesar de que en muchas universidades del mundo se han venido desarrollando e implementando Sistemas de Detección de Intrusos con el objetivo de elevar los niveles de seguridad en las mismas, todo ello gracias a la creciente importancia que se le esta dando mundialmente al tema de la seguridad de las redes producto a los también crecientes ataques informáticos, en Cuba no sucede lo mismo. En el país solo dos universidades han llevado a cabo una propuesta de implementación de IDS, tal es el caso de la Universidad de las Villas “Marta Abreu” donde se propuso una plataforma para la detección de intrusiones en la Intranet de la universidad, se implementó un sistema capaz de analizar en tiempo real el tráfico proveniente de cualquier segmento de red, y detectar anomalías que degraden parcial o totalmente un servicio. El IDS utilizado fue el Snort ya que resultaba la mejor opción de acuerdo a las características de la red de la universidad. En el caso del Instituto Universitario “José Antonio Echeverría” se implementó un Sistema Analizador de Logs para la Detección de Intrusos (SALDI) con el objetivo de aumentar la seguridad de la red de computadoras y aliviar las tareas de los administradores. Desafortunadamente, todavía en nuestro país no está generalizada la conciencia de la necesidad del uso de IDS de red; de forma que la mayoría de las universidades ignoran su empleo.

Nuestra universidad no está exenta de los problemas en cuanto a seguridad informática ya que contamos con una red de alta velocidad con una amplia gama de información, servicios y usuarios y sin embargo no se cuenta con un sistema o herramienta efectiva para contrarrestar los ataques informáticos. En la UCI prácticamente todos los servicios están automatizados, por ejemplo y uno de los más importantes es el sistema de estudio, el de control al acceso de los comedores, a los laboratorios; también se desarrollan importantes proyectos en convenio con organizaciones nacionales e incluso con otros países lo que da una medida de la gran envergadura que tienen. Además la UCI es uno de los programas de la Revolución

más importantes en el país y tiene gran connotación internacional tanto por sus logros y proyectos como por el marcado interés político de otros países en ella.

La red de la UCI cuenta en estos momentos con firewalls y programas antivirus pero aún así la seguridad no alcanza el nivel necesario para mantener la integridad y confiabilidad de la información y la disponibilidad de los servicios ya que implementar estos mecanismos de protección minimizan la exposición de la red, sin embargo tales medidas de contención no son suficientes para frenar un ataque, y esto se puede lograr configurando un sistema de detección de intrusos que monitorice la red. Por tales motivos se pretende realizar una investigación que tenga como resultado la configuración del Sistema de Detección de Intrusos Snort en la red-UCI para lograr que aumente el nivel de seguridad de la red. Para lograrlo se analizará la implementación de IDS en centros universitarios de Cuba. Se centrará la investigación en aquellos IDS que cumplan con la característica de ser software libre y basados en red.

Al abordar el tema, la solución propuesta ayuda a abrir el camino en la instrumentación de los IDS, y por ello se realiza el presente trabajo el cual tiene como **objeto de estudio** el proceso de análisis y configuración de los sistemas de detección de intrusos en la UCI.

Se propone como **objetivo general** analizar y configurar el sistema de detección de intrusos-Snort para elevar el nivel de seguridad de la red-UCI. Dentro de este se tienen como **objetivos específicos** configurar el preprocesador Spade para lograr un mejor aprovechamiento de las posibilidades que brinda el Snort y además configurar correctamente las herramientas complementarias a utilizar.

El objetivo delimita el **campo de acción**, que es el proceso de análisis y configuración del sistema de detección de intrusos-Snort en la UCI.

Para guiar la investigación se plantea la siguiente **hipótesis:** La correcta configuración del IDS-Snort en la red- UCI permitirá que el nivel de seguridad de la red de la universidad aumente.

Para cumplir con estos objetivos y resolver la situación problémica planteada, se proponen las siguientes **tareas investigativas**:

- Investigar sobre los IDS en Internet centrando la atención en el Snort.
- Investigar las características de la red en la UCI.
- Estudiar a profundidad el manual y las reglas del IDS-Snort para su correcta configuración.
- Analizar las posibles ubicaciones del IDS-Snort para escoger la más óptima posible de acuerdo a las características de la red-UCI.

- Analizar las diferentes herramientas complementarias para escoger las más indicadas y lograr el mejor rendimiento del Snort posible.
- Configurar las herramientas complementarias seleccionadas.
- Configurar el IDS-Snort en la red-UCI.

Para realizar el diseño metodológico de la investigación se tomó como **población** los centros universitarios de todo el país, la **unidad de estudio** es la Universidad de las Ciencias Informáticas y la **muestra** a estudiar será el grupo de computadoras pertenecientes al Nodo Central. La población que se estudia es homogénea por lo tanto el **tamaño de la muestra** no tiene que ser grande, en este caso esta representada en las computadoras del Nodo Central. La **técnica de muestreo** utilizada es la no probabilística porque todos los grupos de computadoras no tienen la misma probabilidad de ser elegidos para integrar la muestra. Esta técnica tiene varias sub clasificaciones, la que compete a este trabajo es el muestreo intencional ya que los autores seleccionamos los elementos con posibilidades de brindar mayor información.

La **estrategia de investigación** a seguir es la Investigación Explicativa o Experimental ya que los conocimientos precedentes acerca del problema son suficientes para plantear una hipótesis a nivel explicativo, además se pueden determinar las causas que producen el fenómeno en estudio, en este caso, el fenómeno de intrusión no permitida.

Se llevarán a cabo diferentes **métodos científicos** para realizar la investigación. Como método teórico se utilizará el analítico-sintético pues uno de los primeros pasos para llevar a cabo la presente tesis es buscar y analizar documentos e información sobre los IDS y extraer los elementos más importantes relacionados con nuestro objeto de estudio. Los métodos empíricos a utilizar serán la Observación Científica y la Experimentación.

Durante la observación se tendrán en cuenta los siguientes aspectos:

- Anomalía en la sub-red del Nodo Central.
- Tráfico entre el Nodo Central y el resto de las subredes de la universidad.

Estos aspectos serán observados mediante el IDS-Snort durante los últimos 2 meses de desarrollo de la investigación esperando como resultado la detección de anomalías.

Para llevar a cabo el método de la experimentación se crearán las condiciones necesarias para el funcionamiento del Snort en la universidad verificando la hipótesis planteada.

Estructura básica de la experimentación:

- Constatación del estado inicial: Montar el Snort para que monitorice una sub-red donde anteriormente no se detectaban anomalías. Por ejemplo: computadora con ip 10.0.0.161 localizada en el Nodo Central.
- Introducción del factor de cambio: Crear una anomalía en la sub-red. Por ejemplo: un ataque dirigido a una pc del Nodo Central desde otra pc ubicada en el docente o la residencia.
- Constatación de estado final: El Snort detectará la anomalía en la sub-red y lanzará una alerta.
- Comparación del estado inicial con el final: El Snort funciona correctamente ya que detecta un suceso no permitido en la sub-red donde inicialmente no se detectaban este tipo de anomalías.

Operacionalización de variables.

Marco Conceptual

Sistema de Detección de Intrusos: Mecanismo de seguridad que lleva a cabo la detección de intrusos, o sea, el análisis automático de parámetros que modelan la actividad de un entorno con el propósito de identificar intrusiones.

Seguridad de la red: Se define como la preservación de la confidencialidad (solo quienes estén autorizados pueden acceder a la información), la integridad (la información y sus métodos de proceso son exactos y completos) y la disponibilidad (los usuarios autorizados tienen acceso a la información y los sistemas que la tratan siempre que lo requieran) de la red.

Variables a estudiar

Sistema de Detección de Intrusos.

Seguridad de la red.

Variable Conceptual	**Dimensión**	**Indicador**	**Sub-indicadores**
Sistema de Detección de Intrusos	Universidades de Cuba	Aceptabilidad	➢ No introduce una sobrecarga considerable en el sistema. ➢ Introduce una sobrecarga considerable en el sistema.
		Compatibilidad	➢ Compatible ➢ Poco compatible ➢ No compatible
		Fiabilidad	➢ Fiable ante el aviso de un posible evento. ➢ Poco fiable ante el aviso de un posible evento ➢ No fiable ante el aviso de un posible evento
		Sensibilidad	➢ Gran capacidad de análisis al localizar un posible ataque. ➢ Poca capacidad de análisis al localizar un posible ataque.
Seguridad de la red	Universidad de las Ciencias Informáticas.	Confidencialidad	➢ Los datos solo pueden ser accedidos por personal autorizado ➢ Los datos pueden ser accedidos por personal autorizado o no.
		Integridad	➢ Los datos lleguen íntegros ➢ Los datos no lleguen íntegros.
		Disponibilidad	➢ Los datos estén disponibles. ➢ Los datos no estén disponibles.

Capítulo

I

FUNDAMENTACIÓN TEÓRICA

Introducción

En este primer capitulo se desarrollará la fundamentación teórica de la investigación donde se abordarán los principales conceptos y definiciones, la evolución histórica de los IDS, la necesidad de su implementación en la universidad además de los beneficios y riesgos que trae consigo su utilización.

1.1 Tecnologías de detección

Como todas las vulnerabilidades no son conocidas, así como tampoco son conocidos los posibles ataques, estos últimos años se han desarrollado productos para detectar tanto las posibles vulnerabilidades de los programas instalados en los ordenadores, del sistema operativo y de servicios de red, como los posibles ataques que se pueden perpetrar.

Han surgido productos detectores de vulnerabilidades, que verifican las políticas de seguridad en búsqueda de agujeros en los sistemas, passwords débiles, privilegios erróneos, etc. y que escanean las redes en busca de vulnerabilidades en los dispositivos conectadas a estas.

También han surgido detectores de ataques que actúan como centinelas, esperando desde cambios en los permisos de los ficheros y accesos no permitidos en los sistemas, hasta ataques conocidos en el flujo de datos que circula por las redes.

Es entonces cuando se habla de los Intrusion Detection Systems o Sistemas de Detección de Intrusos (de ahora en adelante se hará referencia a ellos como IDS). [U-1]

1.2 Definiciones

Antes de centrarnos en lo que es un Sistema de Detección de Intrusos, es conveniente aclarar el concepto de intrusión ya que el mecanismo que se explicará posteriormente tiene como funcionalidad básica detectar estas situaciones dentro de las redes.

Una intrusión es un conjunto de acciones que intentan comprometer (poner en peligro) la integridad, la confidencialidad o la disponibilidad de un sistema informático.
Las intrusiones tienen distintos orígenes: atacantes que acceden a los sistemas desde Internet, usuarios autorizados del sistema que intentan ganar privilegios adicionales para los cuales no están autorizados, usuarios autorizados que hacen un mal uso de los privilegios o recursos que se les han asignado, etc.
De una forma u de otra las intrusiones están dirigidas fundamentalmente a:

- Acceder a una determinada información.
- Manipular cierta información.
- Hacer que el sistema no funcione de forma segura o inutilizarlo.[U-2]

Después de exponer brevemente que es una intrusión se puede dar paso a la explicación del concepto de Sistema de Detección de Intrusiones.

Un IDS o Sistema de Detección de Intrusiones es un sistema que intenta detectar y alertar sobre las intrusiones intentadas en un sistema o en una red.

Un IDS es una herramienta de seguridad que intenta detectar o monitorizar los eventos ocurridos en un determinado sistema informático o red informática en busca de intentos de comprometer la seguridad de dicho sistema.

Los IDS buscan patrones previamente definidos que impliquen cualquier tipo de actividad sospechosa o maliciosa sobre nuestra red o host, aportan a nuestra seguridad una capacidad de prevención y de alerta anticipada ante cualquier actividad sospechosa. No están diseñados para detener un ataque, aunque sí pueden generar ciertos tipos de respuesta ante éstos. Los IDS aumentan la seguridad de nuestro sistema, vigilan el tráfico de nuestra red, examinan los paquetes analizándolos en busca de datos sospechosos y detectan las primeras fases de cualquier ataque como pueden ser el análisis de nuestra red, barrido de puertos, etc. [U-5]
Algunas de las características deseables para un IDS son:

- Deben estar continuamente en ejecución con un mínimo de supervisión.
- Se deben recuperar de las posibles caídas o problemas con la red.

- Debe poderse analizar él mismo y detectar si ha sido modificado por un atacante.
- Debe utilizar los mínimos recursos posibles.
- Debe estar configurado acorde con la política de seguridad seguida por la organización.
- Debe de adaptarse a los cambios de sistemas y usuarios y ser fácilmente actualizable.[U-1]

1.3 Evaluación de los Sistemas de Detección

Antes de adentrarnos en las características de los actuales IDS es importante conocer cuales fueron los primeros proyectos y programas que antecedieron a lo que llamamos hoy Sistemas de Detección de Intrusos. La antesala de la detección de intrusiones fue precisamente las auditorías de seguridad, las cuales tienen sus inicios a mediados de los años 50 cuando la Empresa "Bell Telephone System" crea un grupo de desarrollo con el objetivo de analizar el uso de los ordenadores en las empresas telefónicas. Dicho grupo establece la necesidad de realizar auditorias mediante el procesamiento electrónico de los datos, dando como resultado el primer sistema a gran escala de facturación controlada por ordenadores.

Numerosos recursos fueron empleados por el Departamento de Defensa de los EEUU para la investigación de políticas de seguridad de los denominados sistemas de confianza o sistemas que empleaban los suficientes recursos de firmware para permitir el procesamiento simultáneo de una variedad de información confidencial o clasificada. Estos esfuerzos concluyeron cuando fue incluído un apartado sobre los mecanismos de las auditorías en el "*Trusted Computer System Evaluation Criteria*" o TSEC "Libro Naranja", como requisito para cualquier sistema de confianza de clase C2 o superior. Esta serie de documentos sobre sistemas de confianza fueron conocidos como la "Serie Arco Iris", debido a los colores de las tapas de los libros que se publicaban.
Las auditorías de seguridad fueron publicadas en el "Libro Marrón" titulado "*A Guide to Underestanding audit in Trusted System*", en el cual se enumeraron los objetivos de un mecanismo de auditoria.

El primer sistema de detección de intrusiones en tiempo real fue desarrollado por la marina estadounidense, y proponía una correspondencia entre la actividad anómala y el uso indebido. Usaba perfiles para describir los sujetos del sistema y reglas de actividad para definir las acciones que tenían lugar. "Intrusion Detection Expert System", IDES, por sus siglas en inglés, era un sistema híbrido porque

añadía un nivel de seguridad adicional mediante el uso de un sistema experto basado en reglas de seguridad con lo cual minimizaba los efectos de un intruso que intentara eludir el detector de anomalías.
El Centro de Soporte Criptológico de las Fuerzas Aéreas de los EEUU, por su parte, desarrolló el proyecto Haystack, el cual fue usado para encontrar signos de ataques internos en los ordenadores principales de sus bases, servidores corporativos o mainframes los cuales manejaban información confidencial. Este sistema fue desarrollado en ANSI C y SQL.

Otro proyecto importante fue sin dudas el MIDAS, "Multics Intrusion Detection and Alerting System", creado por el Nacional Computer Security Center (NCSC). Fue usado para monitorizar el sistema NCSC's Dockmaster. MIDAS usaba un sistema híbrido en el que combinaba la estadística de anomalías y las reglas de seguridad de un sistema experto. Fue uno de los primeros sistemas de detección de intrusiones conectado a Internet.
En el Laboratorio Nacional de los Álamos nace el "Network Audit Director and Intrusion Reporter" (NADIR) el cual usaba técnicas de detección similares a los sistemas IDES y MIDAS.
La Universidad de California también dio su aporte con el "Network System Monitor" (NSM), diseñado para trabajar en una estación Unix de Sun. Fue el primer sistema de detección de intrusiones que monitorizaba el tráfico de red y lo usaba como fuente de datos. Los sistemas anteriores usaban los eventos del sistema. Su funcionamiento se basaba en poner el dispositivo de red en modo promiscuo, de modo que pudiera monitorizar todo el tráfico que llegara a la tarjeta de red, capturando los paquetes e identificando el protocolo usado para poder extraer los datos necesarios para el análisis. Usaba además un enfoque basado en matrices para archivar y analizar los datos en busca de variaciones estadísticas que arrojaran un comportamiento fuera de lo común o violaran las reglas preestablecidas. Este sistema estuvo dos meses en estado de prueba, monitorizó 111 000 conexiones y detectó más de 300 intrusiones de los cuales los administradores no llegaron a percibir ni el 1 %. [U-4]

1.4 El entorno actual exige IDS

La mayor frecuencia de las rápidas amenazas combinadas sigue siendo el problema más urgente de las pequeñas empresas que no tienen sistemas IDS. Las amenazas combinadas utilizan combinaciones de códigos maliciosos como virus, gusanos y caballos de Troya para aprovechar vulnerabilidades conocidas en los códigos de las aplicaciones o de los sistemas. Otros problemas son el rápido aumento de

amenazas Windows 32 y la creciente cantidad de amenazas dirigidas a los servicios entre pares (P2P) y estaciones de trabajo de mensajería instantánea.

1.4.1 Otras áreas de preocupación:

- Ataques más virulentos: Las epidemias recientes MyDoom y Netsky comenzaron en el año 2004 con un mensaje destructivo para el mundo del ciberespacio. Estos gusanos fueron esencialmente una amalgama de las peores características nocivas de los principales gusanos del 2003 como Welchia, WinMail y SoBig. En cada epidemia, las amenazas parecen ser más refinadas porque atacan a través de múltiples puntos de entrada y se propagan a gran velocidad, lo que significa que se reduce el tiempo de reacción a un ataque y que los IDS deben detectar los signos preliminares del ataque antes de que llegue a la red.
- A mayores vulnerabilidades, mayor exposición al riesgo: Así como hay una lista creciente de vulnerabilidades, se reduce rápidamente el lapso de tiempo que transcurre desde que se descubre una vulnerabilidad hasta que se produce el ataque, lo que se conoce como la "Ventana de amenazas a las vulnerabilidades".
- Los hackers tienen más herramientas: Los hackers actuales no necesitan tener bastantes conocimientos técnicos para lanzar un ataque. Existen muchas herramientas de piratería informática fáciles de utilizar y disponibles para lanzar ataques cada vez más sofisticados. [U-3]

1.5 Necesidad de un IDS

¿Por qué es necesaria la detección de intrusiones? ¿No sería suficiente para la organización usar cortafuegos para controlar el acceso a sus redes, y las redes privadas virtuales (VPN) para mantener una comunicación segura?

Los IDS son los "ojos" del equipo de seguridad. No hay que olvidar que son herramientas de monitorización y detección complementarias a los mecanismos de protección tradicionales. Primero se implementan mecanismos de protección (Firewalls, Antivirus, VPN, etc.) y posteriormente se procede a implantar el sistema de detección de intrusiones. Desarrollar VPN y cortafuegos es una buena práctica. Una política de cortafuego robusta puede minimizar la exposición de muchas redes, sin embargo tales

medidas de contención no son suficientes. A continuación se verán algunos aspectos que pudieran ayudar a explicar la incapacidad de los sistemas tradicionales –entiéndase cortafuegos- para frenar un ataque.

- Los ataques están “cobrando inteligencia”.

  Un atacante enmascara su ataque en la red. Estas técnicas incluyen caballos de Troya basados en correos electrónicos, técnicas de chequeos sigilosos y ataques túneles, en los cuales el atacante enmascara su tráfico de paquetes peligrosos, y que en su forma natural pudieran ser detenidos por el cortafuegos, con la encapsulación de tal tráfico en paquetes correspondientes a otros protocolos de red permitidos como ICMP ó DNS.
- Las vulnerabilidades están proliferando.

  Los atacantes también toman ventaja de las vulnerabilidades atribuidas a configuraciones incorrectas de sistemas, programas de pobre ingeniería, usos negligentes de usuarios sobre sistemas operativos, carencia de banderas básicas en algunos protocolos, entre otras deficiencias. Existe un constante crecimiento en la lista de aplicaciones con vulnerabilidades, y los atacantes son buenos explotándolas a través de protocolos como HTTP, que les son permitidos atravesar casi cualquier cortafuego.
- Las herramientas para realizar intrusiones hacen los ataques más fáciles.

  A pesar de que muchas técnicas de verificación y ataque a redes son conocidas desde hace varias décadas, es solo recientemente que las herramientas que conducen a análisis sofisticados de redes se han hecho ampliamente disponibles. Como la sofisticación de estas herramientas se ha incrementado, el conocimiento técnico necesario por los atacantes ha disminuido; así que las organizaciones están expuestas a un rápido crecimiento en el número potencial de ataques.
- Los atacantes de dentro siguen siendo los predominantes.

  Mientras los atacantes externos a las organizaciones continúan de forma incremental realizando usos indebidos de los recursos de red, la mayor cantidad de actividades maliciosas registradas provienen de dentro. Esto se debe a usuarios legítimos, pero mal identificados, que pueden sacar provecho del acceso físico o de algún nivel de privilegio genuino, y que conocen las medidas de seguridad local, los activos sensibles, la estructura de la red, entre otras informaciones. Por su parte un atacante externo tiene que esforzarse por obtener toda esta información ilícitamente. [B-3]

La detección de intrusos permite a las organizaciones proteger sus sistemas de las amenazas que vienen aparejadas al incremento de la conexión a las redes de datos. Los IDS han ganado aceptación como necesaria adición a la infraestructura de seguridad en las redes de las organizaciones.
Cuando son usados con conocimiento, los IDS pueden ofrecer indicadores de actividades malignas así como vulnerabilidades en la seguridad, ofreciendo un nivel extra de protección. Sin ellos los administradores disponen de pocas posibilidades para conocer acerca de las amenazas que le acechan, quedando en peores condiciones de responder eficientemente a una actividad maliciosa.

Un IDS puede detectar ataques comunes como intentos de explotar vulnerabilidades conocidas, pruebas de red o sobrecargas en recursos críticos en un tiempo razonablemente breve. Mediante la identificación de actividades inválidas, los IDS pueden indirectamente situar las vulnerabilidades de los sistemas y redes habilitando sus soluciones de forma precisa.
Pese a la automatización de los sistemas de detección, los costes de mantenimiento son extremadamente elevados ya que es necesario destinar muchos recursos humanos para visualizar y gestionar las alertas, así como para distinguir entre los ataques reales y los denominados falsos positivos (detección de intrusiones que realmente no lo son).
Un IDS puede actuar como recaudador de presupuesto, ya que en ocasiones es necesario disponer de información tangible para obtener una respuesta positiva de la dirección de la empresa a la hora de destinar presupuesto a seguridad informática. [U-2]

1.6 Como seleccionar un IDS

El proceso de selección del IDS no puede (no debe) ser tomado a la ligera. Es muy importante conocer las necesidades y las limitaciones antes de seleccionar el IDS, ya que un IDS requiere conocimientos de seguridad y un constante mantenimiento, por lo que se debe pensar en el tipo de personal que lo va a mantener (si no tiene formación, lo mejor será optar por una solución automatizada, aunque menos potente). Básicamente, los pasos que hay que seguir para seleccionar adecuadamente un IDS son:

1. *Identificar las necesidades de la organización*: Normalmente, la mayor dificultad a la hora de instalar un IDS es convencer a la dirección de que es necesario instalar uno. Por eso, es importante identificar los activos de la empresa u organización y los riesgos que se corren al no tenerlos protegidos. Básicamente, se trata de responder a las siguientes preguntas: ¿cuál es la importancia de la

información?, ¿cuáles son las pérdidas potenciales?, ¿cuál es la probabilidad de sufrir un ataque?, ¿cuál es el nivel relativo de seguridad necesario? y ¿se han sufrido ataques anteriores? Esta última pregunta tiene gran importancia porque en la universidad han ocurrido ataques a sitios de docencia como el de Física en años anteriores por solo mencionar uno, lo que demuestra signos de debilidad y por tanto es probable que pueda volver a sufrir un ataque.

2. *Obtener los suficientes conocimientos sobre la detección de intrusos:* Es necesario tener, como mínimo, unas nociones básicas de lo que puede hacer un IDS y para que sirve, o de lo contrario lo que se obtendrá es una ilusión de seguridad. La persona responsable de elegir el IDS debería tener una formación amplia en seguridad, y el administrador del IDS debe tener, al menos, nociones básicas de IDS. De hecho el nivel de conocimientos del administrador del IDS condiciona la elección del mismo, ya que cuantos menos conocimientos tenga de IDS, más automatizado y simple tiene que ser el mismo. Esto demuestra la importancia de la investigación realizada en el presente capítulo.
3. *Obtener los suficientes conocimientos sobre la infraestructura de red:* ¿Existe algún tipo de estructuración en la red? Esta pregunta es fundamental, porque cuanto mas estructurada este la red (separada en departamentos, niveles de acceso, servicios, etc.) más posibilidades se abren a la hora de elegir los sensores. Si la red esta muy estructurada, se puede pensar en la instalación de múltiples sensores con gran cantidad de análisis, ya que el tráfico en cada segmento será pequeño y el tipo de ataques que buscar se puede segmentar en bloques (ver el epígrafe *"Donde colocar el IDS"*). Por otro lado, el ancho de banda de la red y la cantidad de tráfico que se cursa también es muy importante, puesto que condiciona la velocidad mínima requerida por el IDS para funcionar correctamente, sin perder paquetes. Conocer la red es imprescindible para elegir bien.
4. *Evaluar varios IDS:* Una vez que se han identificado las necesidades de la organización, habremos filtrado la mayoría de productos disponibles. De entre los productos que mejor se adaptan a nuestras necesidades. ¿Cuál elegimos? La respuesta no es trivial, y lo mejor es evaluarlos. La mayoría de empresas proporcionan versiones de evaluación de sus productos, lo que permite comprobar, sin compromiso, su aplicación. Aquí es donde se pueden evaluar realmente las facilidades que proporciona cada producto, y su funcionamiento en el entorno.
5. *Especificar una política y unos procedimientos de emergencia:* Una vez tenemos el IDS, es necesario establecer/actualizar la política de seguridad y los procedimientos de emergencia. Dado que el IDS es una alarma, el procedimiento de emergencia es fundamental, porque no sirve de nada ser capaz de

detectar ataques si luego no se sabe que hacer. Debe haber una documentación completa sobre lo que hacer en cada situación (a quien avisar según la alarma y la hora, que cosas hacer en espera de una respuesta...) de forma que no se puedan generar dudas sobre el plan de acción. No se puede considerar completa la selección de un IDS si no va adjunto de una documentación adecuada y de unos planes de emergencia. Una alarma que nadie atiende es totalmente inútil.

1.7 Requisitos de los IDS

El IDS debe ejecutarse continuamente sin que nadie esté obligado a supervisarlo. Su funcionamiento habitual no debe implicar interacción con un humano, sin embargo para obtener resultados realmente positivos, es necesario que una persona monitorice las alertas que muestra el dispositivo de detección de intrusiones.

Pocas empresas están dispuestas a contratar personal para analizar logs o controlar patrones de tráfico de red. No obstante es estrictamente necesario disponer de personal especializado para dicho análisis ya que un sistema automatizado no puede determinar si la alerta detectada es realmente una alerta o un falso positivo.

Los mecanismos de detección de intrusos han de ser aceptables (aceptabilidad) para las personas que trabajan habitualmente en el entorno y no ha de introducir una sobrecarga considerable en el sistema (si un IDS ralentiza demasiado una máquina, simplemente no se utilizará).

El IDS no debe generar una cantidad elevada de falsos positivos o de logs ya que llegaría un momento en que nadie se preocuparía de comprobar las alertas emitidas por el detector.

Ningún sistema informático puede considerarse estático: desde la aplicación más pequeña hasta el propio kernel de Unix, pasando por supuesto por la forma de trabajar de los usuarios (adaptabilidad). En el mundo de la seguridad de sistemas de información todo cambia a gran velocidad y si nuestros mecanismos de detección de intrusos no son capaces de adaptarse rápidamente a esos cambios están condenados al fracaso.

Todo IDS debe además presentar cierta tolerancia a fallos o capacidad de respuesta ante situaciones inesperadas. Algunos -o muchos- de los cambios que se pueden producir en un entorno informático no son graduales sino bruscos, y un IDS debe ser capaz de responder siempre adecuadamente ante los mismos. Puede contemplarse, por ejemplo, un reinicio inesperado de varias computadoras o un intento de engaño hacia el IDS; esto último es especialmente crítico. [B-1]

1.8 Funcionalidad

Una de las funciones más importantes de los IDS es la de prevenir problemas ya que se incrementa la posibilidad de descubrir y demostrar que una intrusión ha tenido lugar.
Hay situaciones de alto riesgo, entre la publicación de una vulnerabilidad y la publicación del parche que la corrige, en las cuales está claramente justificado el uso de los IDS para monitorizar la red durante el tiempo que es vulnerable. Los intentos de acceso en este período serán especialmente críticos.

Los IDS tienen la capacidad de detectar las fases previas a un ataque, es decir, aquellas en que el posible atacante está analizando el sistema.
El sistema no puede evitar que ésta se produzca y en su defecto procede a enviar una alerta al administrador (IDS pasivo) o bloquear el origen de dicho escaneo (IDS reactivo).

Un IDS nos puede ayudar a conocer la amenaza existente fuera y dentro de la organización. Esto hace que podamos tomar decisiones sobre que mecanismos de seguridad deberíamos implantar en nuestra empresa. [U-2]

1.9 Tipos de Sistemas de Detección de Intrusos

Existen varios tipos de IDS, clasificados según el tipo de situación física, del tipo de detección que posee o de su naturaleza y reacción cuando detecta un posible ataque.

1.9.1 Clasificación por situación

Según la función del software IDS, estos pueden ser:

NIDS (Network Intrusion Detection System)

Los NIDS analizan el tráfico de la red completa, examinando los paquetes individualmente, comprendiendo todas las diferentes opciones que pueden coexistir dentro de un paquete de red y detectando paquetes armados maliciosamente y diseñados para no ser detectados por los cortafuegos. Pueden buscar cual es el programa en particular del servidor de web al que se está accediendo y con que

opciones y producir alertas cuando un atacante intenta explotar algún fallo en este programa. Los NIDS tienen dos componentes:

- Un sensor: situado en un segmento de la red, la monitoriza en busca de tráfico sospechoso
- Una Consola: recibe las alarmas del sensor o sensores y dependiendo de la configuración reacciona a las alarmas recibidas.

Ventajas del NIDS:

- Detectan accesos no deseados a la red.
- No necesitan instalar software adicional en los servidores en producción.
- Fácil instalación y actualización por que se ejecutan en un sistema dedicado.
- Pocos sensores bien posicionados pueden monitorear redes grandes.
- Los sensores son dispositivos pasivos que escuchan el tráfico de la red en tiempo real sin interferir en su operación normal. Un NIDS puede adicionarse a una red con el mínimo de impacto.
- Los sensores se pueden proteger muy bien contra ataques y es posible hacerlos invisibles a los atacantes.
- Un NIDS puede detectar un ataque antes de que este alcance su objetivo.
- Típicamente un NIDS es independiente de plataformas y relativamente fáciles de desarrollar.

Desventajas del NIDS:

- Examinan el tráfico de la red en el segmento en el cual se conecta, pero no puede detectar un ataque en diferentes segmentos de la red. La solución más sencilla es colocar diversos sensores (los conmutadores que ofrecen un puerto de monitoreo mitigan parcialmente este problema).
- Pueden generar tráfico en la red quedando indisponible para procesar todos los paquetes y detectar un ataque. (Son pocos los que en el año 2002 podían trabajar de forma efectiva a velocidades de línea de Gigabits/s).
- Ataques con sesiones encriptadas son difíciles de detectar (caso de usar VPN).
- Un NIDS no puede determinar con certeza cuando un ataque es exitoso.
- Los sensores pueden trasmitir grandes volúmenes de información a la consola de administración consumiendo ancho de banda disponible y causando problemas de latencia. [B-9]

HIDS (Host Intrusion Detection System)

En cambio, los HIDS analizan el tráfico sobre un servidor o un PC, se preocupan de lo que está sucediendo en cada host y son capaces de detectar situaciones como los intentos fallidos de acceso o modificaciones en archivos considerados críticos.

Ventajas del HIDS:

- Herramienta potente, registra comandos utilizados, ficheros abiertos, etc.
- Tiende a tener menor número de falsos-positivos que los NIDS, entendiendo falsos-positivos a los paquetes etiquetados como posibles ataques cuando no lo son.
- Menor riesgo en las respuestas activas que los IDS de red.
- Monitorea eventos locales a una computadora de forma que puede detectar ataques que un NIDS no tiene capacidad de detectar. Un HIDS verá exactamente lo que un atacante hace en la computadora objetivo.
- Pueden disminuir la carga asociada con el monitoreo en redes grandes.
- No se ve afectado por el tráfico de red encriptado, ya que los datos serán desencriptados en la computadora donde reside el sensor.
- Puede monitorear la interacción entre usuarios y aplicaciones servidoras, permitiendo un conocimiento individual de los usos indebidos.

Desventajas del HIDS:

- Requiere instalación en la máquina local que se quiere proteger, lo que supone una carga adicional para el sistema.
- Tienden a confiar en las capacidades de auditoria y logging de la máquina en sí.
- El sensor HIDS es específico del sistema operativo. Tiene que ser instalado, configurado y mantenido en cada computadora que se desee proteger.
- Como dependen de eventos auditados en los reportes, es importante que estos se registren de forma correcta. De esta forma se debe configurar la generación de los registros requeridos, posiblemente impactando en otros programas de aplicación.
- Puede ser atacado y deshabilitado como parte de un ataque a una computadora, por ejemplo puede deshabilitarse con cierto ataque de DoS. [U-1]

Existen varios productos: Tripwire, Dragon Squire (Network Sec. Wizards), Intruder Alert (Axent Tech.), RealSecure (ISS). Estos sistemas actúan monitorizando cambios del sistema a nivel local por tanto llegan a un nivel más bajo que otros sistemas de detección de intrusos y pueden detectar intentos de intrusión que otros sistemas no pueden detectar. Sin embargo estos sistemas actúan a nivel de host y requieren más tiempo de gestión y configuración, ya que cada host donde lo implantemos deberá tener una configuración específica. El hecho de instalarse en el mismo servidor hace que influya en el rendimiento del sistema monitorizado. [U-2]

1.9.2 Clasificación según técnica de análisis utilizada

Los dos tipos de detecciones que pueden realizar los IDS son:

Detección del mal uso o Detección de firmas (Signature Recognition)

La Detección del mal uso es actualmente la técnica más usada por los detectores de intrusos. Involucra la verificación sobre tipos ilegales de tráfico de red, por ejemplo, combinaciones dentro de un paquete que no se podrían dar legítimamente. Este tipo de detección puede incluir los intentos de un usuario por ejecutar programas sin permiso (por ejemplo, “sniffers”). Los modelos de detección basados en el mal uso se implementan observando como se pueden explotar los puntos débiles de los sistemas, describiéndolos mediante unos patrones o una secuencia de eventos o datos (“firma”) que serán interpretados por el IDS. La forma de análisis se basa en la comparación de patrones (pattern matching). El sistema contiene una base de datos con patrones de ataque e irá buscando coincidencias con dichos patrones y cuando se detecte una coincidencia saltará la alerta.

Estos sistemas son realmente efectivos en la detección de ataques aunque generan un número elevado de falsas alarmas. Por tanto es preciso que el periodo en que se regulan (periodo de tunning) sea lo más corto posible.

El buen funcionamiento de un sistema de estas características no sólo depende de una buena instalación y configuración sino de que la base de datos en la que tenemos los patrones de ataque este actualizada.

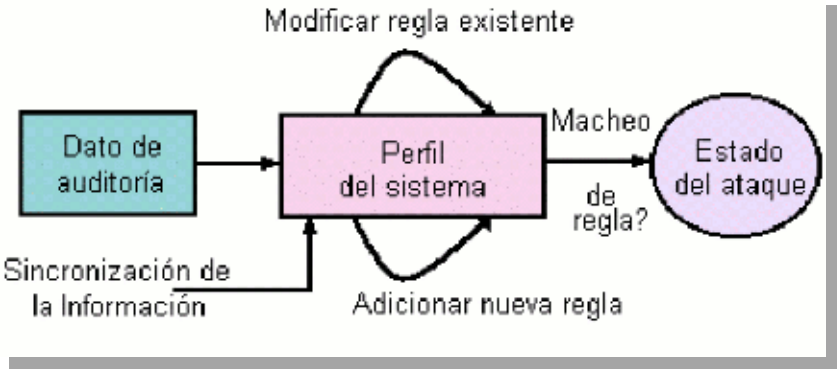

Figura 1. Detección del mal uso

Ventajas de la Detección del mal uso

- Las técnicas de mal uso son efectivas en la detección de ataques sin la generación de un número abrumador de falsas alarmas.
- Los detectores de mal uso pueden diagnosticar rápidamente el uso de una técnica específica o herramienta de ataque, lo cual ayuda a los responsables de seguridad a priorizar medidas correctivas.
- Los detectores de mal uso le permiten a los responsables de seguridad, independientemente de su nivel de experiencia, seguir problemas de seguridad en su sistema, iniciando los procedimientos de manipulación de incidentes.

Desventajas de la Detección del mal uso

- Puede detectar solo aquellos ataques que le son conocidos, por eso deben ser actualizados constantemente con los patrones de nuevos ataques.
- Muchos están diseñados para usar justamente los patrones de ataques que le son definidos; de esa manera son incapaces de detectar ligeras variantes de ataques bien conocidos. [B-3]

Detección del uso anómalo o Detección de anomalías (Anomaly Detection)

Por otra parte la Detección de anomalías (Anomaly Detection) es la técnica en la que el sistema busca patrones anormales de actividad. Esta técnica ha sido y continúa siendo objeto de investigación.

La Detección de anomalías se centra en identificar comportamientos inusuales en un host o una red y para ello utilizan técnicas de Inteligencia Artificial.

Para su funcionamiento se necesita definir un comportamiento “normal” que permita analizar desviaciones de dicho comportamiento ya que funcionan asumiendo que los ataques son diferentes a la actividad normal lo que hace que estos sistemas generen muchos falsos positivos.
Los detectores de anomalías construyen perfiles representando el comportamiento normal de los usuarios, hosts o conexiones de red. Estos perfiles son construidos de datos históricos (data mining) recogidos durante el periodo normal de operación.

Los IDSs basados en Detección de anomalías detectan comportamientos inusuales. De esta forma tienen la capacidad de detectar ataques para los cuales no tienen un conocimiento específico. Estos sistemas pueden producir información que puede ser utilizada para definir firmas en la detección de abusos.
Por el contrario, la detección de anomalías produce un gran número de falsas alarmas debido a los comportamientos no predecibles de usuarios y redes y requieren conjuntos de entrenamiento muy grandes para caracterizar los patrones de comportamiento normal.

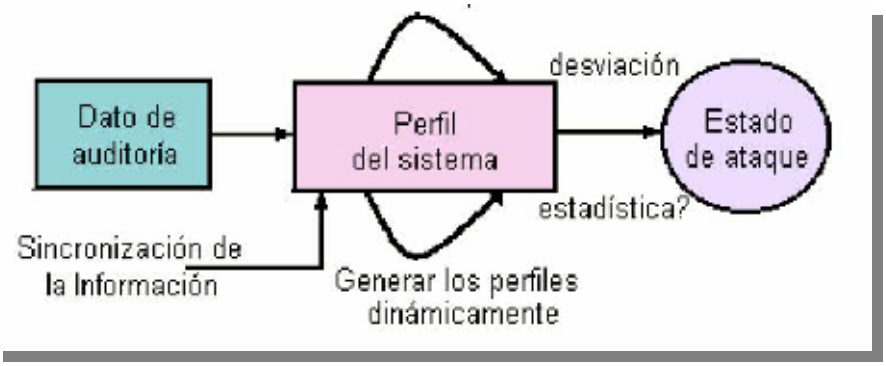

Figura 2. Detección de anomalías

Ventajas de la Detección de anomalías
- Detectan comportamientos inusuales y poseen la habilidad de detectar síntomas de ataques sin tener un conocimiento detallado del mismo.
- Pueden producir información empleada para definir patrones de ataques y pueden ser usados para realizar análisis de tendencias de amenazas a la seguridad.
- Por lo general no requiere de actualización constante de nuevas firmas.

Desventajas de la Detección de anomalías

- La aproximación de la Detección de anomalías usualmente produce una gran cantidad de falsas alarmas debido al comportamiento poco predecible de usuarios y redes.
- Los detectores basados en este tipo de detección frecuentemente requieren extensos "conjuntos de entrenamientos" de registros de eventos del sistema para caracterizar los patrones de su comportamiento normal. [B-1]

1.9.3 Clasificación según su naturaleza

Un tercer y último tipo básico de clasificación sería respecto a la reacción del IDS frente a un posible ataque:

Pasivos

Los IDS pasivos son aquellos que notifican a la autoridad competente o administrador de la red mediante el sistema que sea, alerta, etc. para que este responda manualmente a la intrusión pero no actúa sobre el ataque o atacante. Dicho informe puede llegar a través de un mensaje a consola, correo electrónico, teléfonos celulares, beepers o actualizaciones de reportes. Algunos IDS comerciales generan un mensaje SNMP alertando del problema al sistema de administración de la red.

Activos

Los IDS activos generan algún tipo de respuesta sobre el sistema atacante o fuente de ataque las cuales son especificadas por el administrador frente eventos críticos, donde son tomadas acciones proactivas o correctivas. Tales acciones pueden incluir:

- Corrección de la vulnerabilidad de un sistema.
- Desconectar a un usuario.
- Terminar una conexión.
- Incrementar el monitoreo del evento de forma selectiva.
- Reconfigurar un cortafuegos para bloquear una dirección que fue la fuente de la intrusión detectada o regular la cantidad de tráfico permitido a través de cierto puerto.
- Desconectar un puerto.
- Otros. [B-3]

Si la respuesta es manual, existe un tiempo en el cual el intruso tiene total libertad de atacar los activos en la red sin ser contrarrestado por su defensa; este tiempo varía según la inmediatez de la respuesta manual, que puede ir desde minutos a meses. Dicha ventana de tiempo -dañina- se intenta minimizar con un sistema que responda automáticamente, sin depender al menos inicialmente, de la intervención del administrador de seguridad. [U-5]

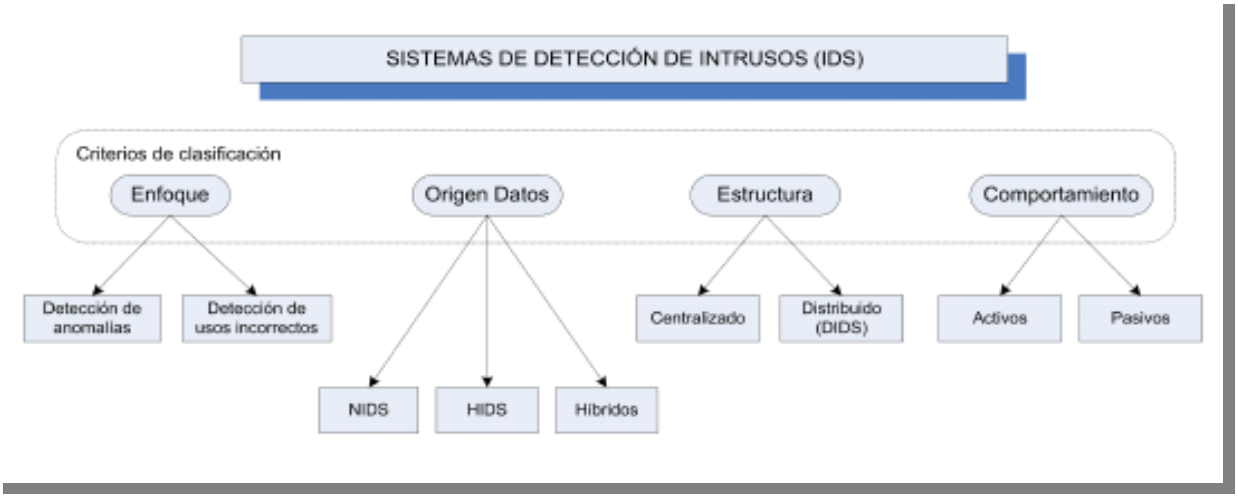

Figura 3. Clasificación de los IDS.

1.10 Topologías de IDS

Existen muchas formas de añadir las herramientas IDS a nuestra red, cada una de ellas tiene su ventaja y su desventaja. La mejor opción debería ser un compendio entre coste económico y propiedades deseadas, manteniendo un alto nivel de ventajas y un número controlado de desventajas, todo ello de acuerdo con las necesidades de la organización. [U-1]

Por este motivo, las posiciones de los IDS dentro de una red son varias y aportan diferentes características. A continuación se explican los lugares típicos de colocación de un IDS:

1. *En el dispositivo frontera entre Internet y la red de la organización*: En muchos casos es útil que los responsables de seguridad conozcan los intentos de intrusiones dirigidas a su red provenientes de Internet, antes de que estos sean bloqueados por algún cortafuegos, para registrar las amenazas reales a las que se encuentra expuesta la red empresarial.
2. *En la red desmilitarizada (DMZ):* Dado que la DMZ está bastante expuesta a los ataques, es muy útil tener un dispositivo que controle su tráfico. Mantener los servidores de la DMZ en funcionamiento

puede ser crítico para la empresa, y sin embargo tienen un nivel de protección menor que el resto de la red.

3. *Tras el firewall*: Esta suele ser la ubicación característica, puesto que permite analizar todo el tráfico que entra en la red. Además, permite vigilar que el cortafuego funciona como debe. En redes grandes, el volumen de tráfico puede ser excesivo, por lo que el análisis debe simplificarse.
4. *En el acceso remoto*: Como punto de entrada salida a la red de la organización que se necesita chequear en busca de posibles intrusiones por esta vía.
5. *En el acceso de usuarios*: Esto sirve tanto para identificar ataques a computadoras de usuario, como para vigilar los ataques que se inicien en el interior de la organización. Es tan negativo que un ataque afecte a la red empresarial, como que esta aparezca como su origen. [B-7]

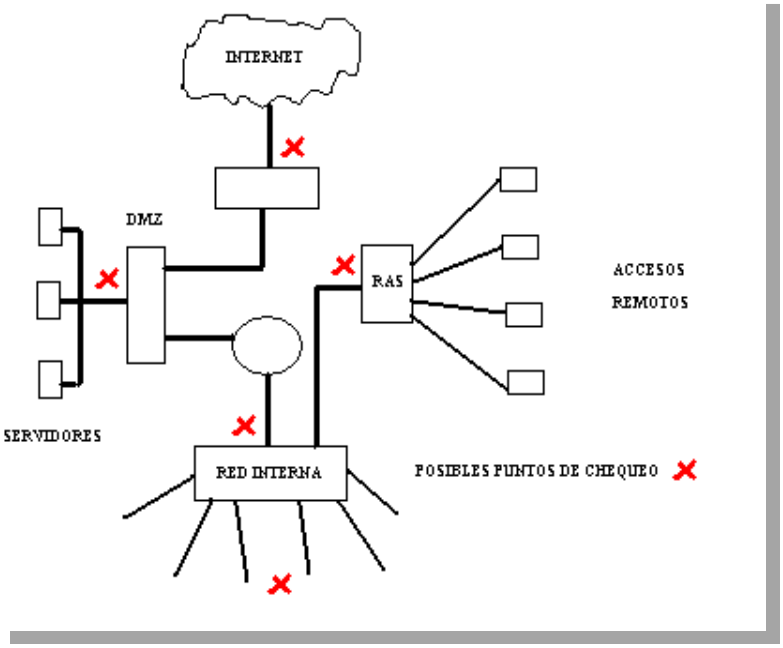

Figura 4. Posibles puntos donde colocar un IDS.

Evidentemente, la idea no es poner un único IDS en el lugar idóneo, sino poner varios en distintos lugares de forma que se establezcan distintas capas de seguridad. Aunque habrá redes simples donde un único IDS puede servir para todo, lo recomendable será tener un esquema similar al de la figura 4.

A continuación se verán diferentes posibilidades en una misma red. En este ejemplo se tiene una red donde un cortafuego divide la Internet de la zona desmilitarizada (DMZ), y otro que divide la DMZ de la intranet de la organización como se muestra en el dibujo 4. Por zona desmilitarizada se entiende la zona que debemos mostrar al exterior, la zona desde la cual se muestran los servicios o productos:

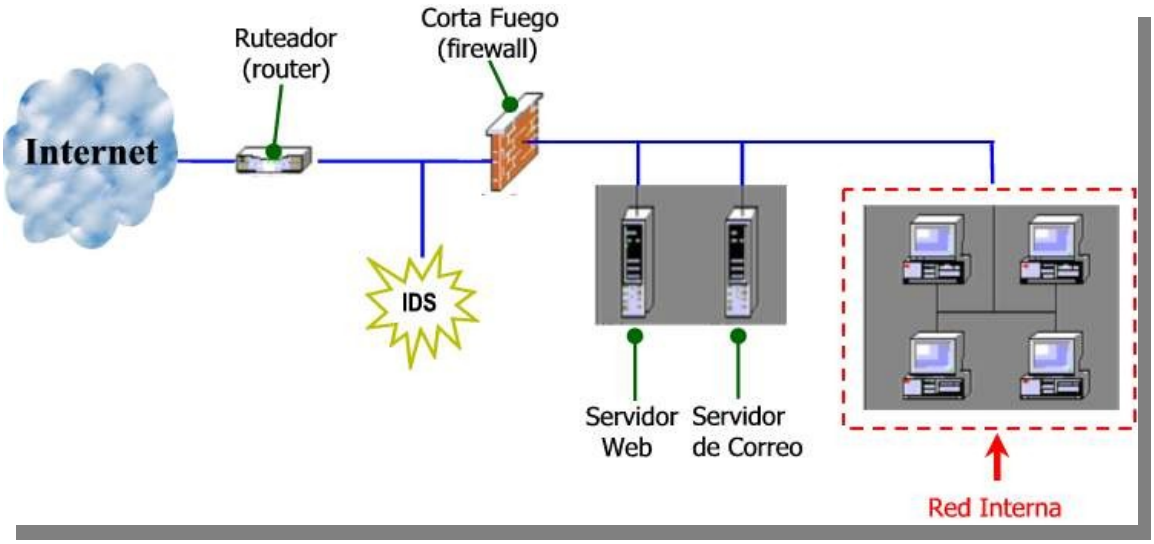

Figura 5. Red con IDS simple

Si se sitúa un IDS antes del cortafuego exterior se permitiría detectar el rastreo de puertos de reconocimiento que señala el comienzo de una actividad hacking, y se obtendría como ventaja un aviso prematuro. Sin embargo, si los rastreos no son seguidos por un ataque real, se generará un numeroso número de alertas innecesarias con el peligro de comenzar a ignorarlas.
Si se opta por colocar el IDS en la zona desmilitarizada (DMZ) se tendría como ventaja la posibilidad de adecuar la base de datos de atacantes del NIDS para considerar aquellos ataques dirigidos a los sistemas que están en la DMZ (servidor web y servidor de correo) y configurar el cortafuego para bloquear ese tráfico.

Así mismo, un NIDS dentro de la red, por ejemplo, de Recursos Humanos podría monitorear todo el tráfico para fuera y dentro de esa red. Este NIDS no debería ser tan poderoso como los comentados anteriormente, puesto que el volumen y el tipo de tráfico es más reducido. El resultado se puede visualizar en el segundo dibujo:

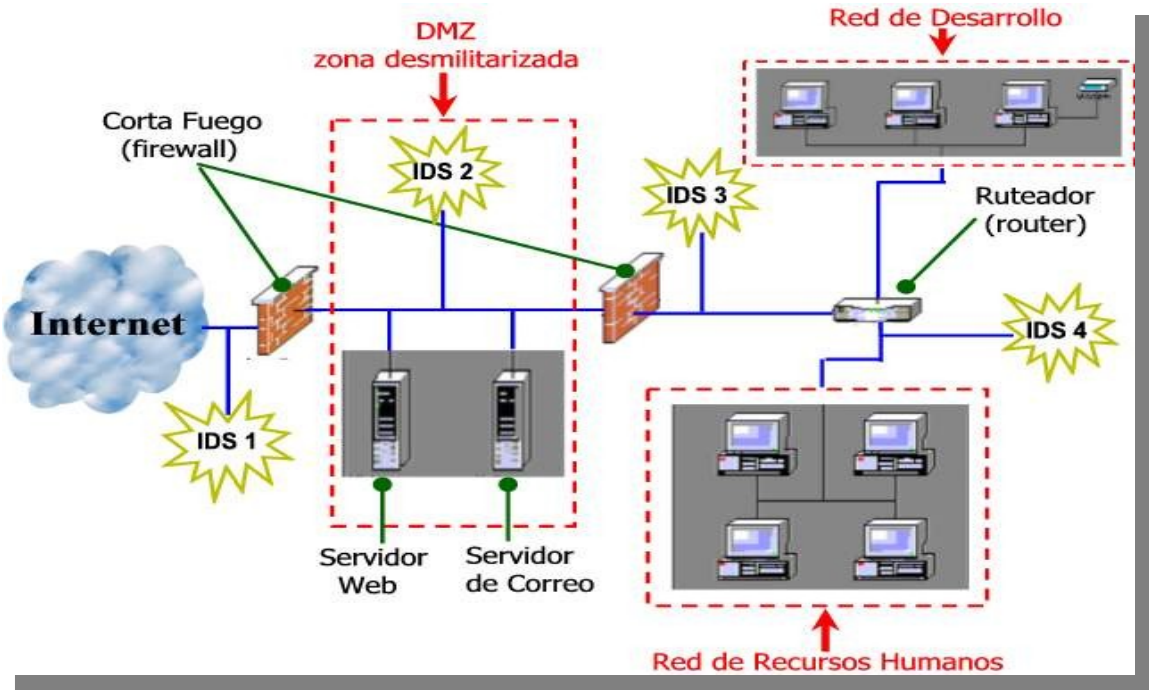

Figura 6. Red completa con IDS

El IDS1 se encargaría de avisar del rastreo de puertos, y si es activo podría enviar un "aviso" tanto al que está rastreando (por ejemplo un ping a la dirección que emite el paquete) como al encargado de la seguridad de la organización. El IDS2 se encargaría de vigilar la zona desmilitarizada y analizar el tráfico que reciben tanto el servidor web como el servidor de correo. Los otros dos IDS se encargarían de la red interna, el IDS3 de la totalidad de la red, y el IDS4 de una subred, en este caso la de RRHH. Estos dos NIDS internos (el IDS3 y el IDS4) podrían ser sensores que recogiesen la información y lo enviasen a una consola dónde se realizarían los cálculos. [U-1]

Por otra parte, si la red está compuesta por conmutadores como elementos de interconexión (red conmutada), se presenta un problema a la hora de instalar un IDS, ya que en principio el IDS no será capaz de analizar más tráfico que el que vaya destinado a él, o el de Broadcast. Existen dos soluciones a este problema, además del uso de repetidores: Spannings Ports y TAPS.

El Spanning Port es un puerto del conmutador en el que se replican los paquetes de otro puerto definido, lo que permite monitorear todo su tráfico en una computadora sensor IDS. Por otra parte, los

conmutadores no suelen garantizar que el 100% del tráfico pase al puerto de *spanning* con lo que existe la posibilidad de perder paquetes y no reconocer un ataque que el IDS es capaz de detectar.

La otra posibilidad son los TAPS, esta viene a solucionar los inconvenientes anteriores y es la que realmente se debe usar. Básicamente, es un dispositivo de 3 puertos que permite duplicar el tráfico entre 2 puertos a un tercero, de forma unidireccional (el puerto de copia no puede enviar ni recibir tráfico, solo recibir las copias). Así se consigue la monitoreo del tráfico de red de una de forma segura, ya que el TAP esta construido por hardware donde no existe la posibilidad de que el IDS pierda paquetes y por tanto, potencialmente ataques. [B-7]

1.11 Atempamiento de las fuentes de información y análisis.

Un IDS puede trabajar tanto en modo tiempo real como en modo lote o batch. La mayoría de los productos IDS comerciales funcionan en tiempo real o se aproximan a este, tal es el caso de los NIDS. Por su parte los HIDS pudieran trabajar en tiempo real, pero demandan un mecanismo de captura de eventos en el momento en que estos son escritos en los registros de auditoria, lo que técnicamente es más complejo. Generalmente los HIDS leen los registros una vez que han sido escritos y este proceder aunque provoca demoras bajas no significativas para empresas que utilizan respuestas manuales, puede reducir la efectividad cuando se usa una respuesta automática.

Un IDS que realice análisis de los paquetes de red o de los registros de auditoria en busca de intrusiones, a intervalos mayores de 15 minutos, no responderá con la rapidez demandada por la mayoría de las organizaciones.

De todas formas el análisis en modo batch de los IDS puede ser valorado al menos en dos casos:

- Análisis de tendencias para conocer las actividades y comportamiento de la red, que pueda servir para indicar futuros ataques.
- Análisis forense, para construir una imagen de como el atacante actuó en la intrusión y las vulnerabilidades explotadas por este. [T-2]

1.12 Los IDS y las políticas de seguridad

Los IDS deben ser tratados como un elemento complementario en las políticas de seguridad de las organizaciones, pero antes de implementar o instalar un sistema de detección de intrusiones se recomienda analizar la política de seguridad y ver cómo encajaría el IDS en ella:

- Se recomienda una política de seguridad bien definida y a alto nivel, que cubra lo que está y lo que no está permitido en nuestro sistema y nuestras redes.
- Procedimientos documentados para que proceda el personal si se detecta un incidente de seguridad.
- Auditorías regulares que confirmen que nuestras políticas están en vigencia y nuestras defensas son adecuadas.
- Personal capacitado o soporte externo cualificado. [U-1]

1.13 Configuración del IDS.

Típicamente un IDS ofrece posibilidades de actuar sobre diversas opciones de configuración para su monitoreo y análisis. Dependiendo de la implementación específica del IDS, un administrador pudiera disponer para su selección de:

- Cuales ataques serán monitoreados.
- Que acción desarrollar para cada una de las intrusiones detectadas.
- Especificar las direcciones fuentes o destinos ha ser monitoreadas o excluidas.
- Caracterizaciones de las clases - la importancia o severidad de cada alarma.

La configuración es crítica en la optimización de las capacidades de monitoreo de un IDS. En este sentido, es posible centrarse en eventos específicos de interés y la respuesta que el IDS dará en la detección de tales eventos. [T-2]

1.14 La optimización del IDS

Por todo lo que hemos comentado anteriormente, parece obvio que un IDS, más que cualquier otro producto de seguridad, necesita ser optimizado. Su uso de ancho de banda es intensivo y las operaciones que tiene que realizar son muy complejas, consumiendo gran cantidad de CPU. El resultado predecible de un IDS no optimizado es una perdida potencial de paquetes o incluso un elevado numero de falsos positivos (si hay discontinuidades en la secuencia TCP y el IDS hace inspección de estado). Además, aunque el IDS no pierda paquetes en situaciones normales, ante un ataque de inundación de red, o una inundación de red para camuflar otro ataque empezara a fallar, siendo inútil. Por eso es tan importante

que la carga de CPU en la situación estacionaria sea baja, para que el IDS este preparado para las situaciones de emergencia.

Básicamente, las acciones que puede llevar a cabo un administrador para ajustar su IDS son:

- *Optimizar la base de datos de firmas:* En la mayoría de entornos no será necesario buscar todos los ataques posibles, puesto que muchos, aunque se produjesen, no tendrían efecto. Por ejemplo, en un entorno totalmente *Microsoft* no tiene sentido buscar firmas de ataques RPC, mientras que si los servidores FTP no corren wu-ftp no hace falta buscar sus exploits. Esta configuración (más o menos granular) depende mucho del tipo de IDS elegido. Algunos (como Snort) permiten eliminar archivos de reglas completos (rpc.rules) o incluso bajar al nivel de reglas individuales (eliminar las reglas de wu-ftp dentro del archivo ftp.rules) mientras que otros solo permitirán eliminar servicios completos (lo que puede ser útil si el administrador no tiene los suficientes conocimientos).
- *Filtrar trafico no deseado:* Dado que el IDS esta colocado en un segmento de red (o en varios) y absorbe todo el trafico, puede haber situaciones donde alguna clase de trafico generado no sea de utilidad para el IDS (en ningún caso). Por ejemplo, en las conexiones cifradas (SSH) el IDS no va a ser capaz de identificar ninguna firma, por lo que es una perdida de tiempo que el IDS lo analice. Por otro lado, si el IDS solo va a ser administrado en modo consola, el trafico ARP no tiene mucho sentido, y bloquearlo puede ser buena idea para ahorrar procesamiento.
- *Balanceo de carga:* Al igual que se hace en DNS y Servidores Web, el balanceo de carga permite manejar mayor cantidad de trafico, repartiéndolo entre distintos dispositivos. Esta distribución (debido a la naturaleza de los IDS) debe hacerse por sesiones enteras (o de lo contrario la inspección de estado desaparece) por lo que cada NIDS inspecciona la misma cantidad de reglas (de forma que el reparto de carga sea lo mas equitativo posible). Otra opción es hacer una división no necesariamente uniforme, basada en el tipo de tráfico. De esta forma, a cada IDS va un tipo de trafico distinto (a uno http a otro smtp, etc.) y aunque algunos IDS tendrán mas trafico que otros, la optimización de reglas permite un mejor rendimiento de cada IDS. Finalmente, otra opción para hacer esta ultima división es repartir los IDS por los distintos segmentos de red de la organización, siempre y cuando esta red este suficientemente estructurada (una subred de servidores, otra de estaciones de trabajo...).
- *Optimización de la configuración:* Por ultimo, cada IDS tendrá unas opciones de configuración más o menos extensas. Es importante revisar todas estas opciones localizando aquellas que hagan

bajar el rendimiento del IDS (número de conexiones simultáneas, timeouts). Ajustar esos parámetros según la infraestructura especifica de red puede ayudar a optimizar el funcionamiento. Evidentemente, todas estas opciones no podrán ser configurables en todos los IDS, y seguramente algunos poseerán características únicas de configuración. La mejor idea es mirarse detenidamente la documentación y obrar en consecuencia. [B-5]

1.15 Beneficios y Riesgos del empleo de productos IDS.

Beneficios

- Frena las intenciones de malos comportamientos.

El conocimiento de que es posible detectar intrusos y que existen condenas establecidas para aquellos que atacan o abusan de los sistemas informáticos, hacen que el IDS sirva como un freno significativo a personas que piensen violar la política de seguridad de una organización.

- Detecta usos indebidos que otras medidas de contención no pueden prevenir.

Un IDS puede detectar cuando un atacante ha penetrado en un sistema mediante la explotación de deficiencias no corregidas. Puede realizar una importante función de protección al reportar que se ha sido atacado y llamar la atención de los administradores quienes deben contener el ataque ó recobrarse de cualquier daño que haya resultado; además de localizar las vulnerabilidades para prevenir ataques futuros.

- Detecta y trata con el preámbulo de los ataques.

En los ataques existe una actividad preparatoria de común ocurrencia como son las pruebas de red. Cuando un adversario ataca un sistema, típicamente lo hace a través de estados predecibles. El primer estado de un ataque es usualmente la prueba o examen de la red en busca de un punto de entrada óptimo. En entornos que no disponen de un IDS, el atacante esta libre de examinar todo el sistema con un riesgo mínimo de ser detectado, de forma que puede eventualmente encontrar una vulnerabilidad y explotarla. La misma red con algún IDS se presenta como un mayor reto al atacante, aunque este puede probar la red en busca de vulnerabilidades, el IDS observa esta prueba y la identifica como sospechosa. Luego puede bloquear el acceso del intruso al sistema destino, y alertar al personal de seguridad del incidente.

- Documenta las amenazas existentes para una organización.

Un IDS verifica los pedidos que sobre las redes puedan hacer los intrusos, documentando los mismos. Si una organización comprende la frecuencia y características de ataques de los que es víctima, esta puede determinar de mejor forma que medidas de seguridad le son apropiadas tomar para protegerse.

- Actúa como control de calidad para el diseño y la administración de la seguridad.

Cuando un IDS funciona un período de tiempo determinado, los patrones del uso del sistema y los problemas detectados pueden parecer evidentes. Esto resalta las deficiencias en el diseño y administración de la seguridad, de forma que se corrijan estas antes de que provoquen un incidente.

- Ofrece información útil sobre las intrusiones que tengan lugar.

Aún cuando muchos IDS no están capacitados para bloquear ataques, pueden mantenerse recolectando información detallada y confiable sobre estos, de manera que pueda facilitar la manipulación de incidentes y los esfuerzos de recuperación. Además esta inforación puede bajo ciertas circunstancias, ser usadas para los análisis legales y su toma de medidas.

Riesgos

- Un IDS no lo asegura todo.

Independientemente del impacto positivo que han tenido los IDS en las organizaciones, estos no son indestructibles y no debieran ser la única medida de seguridad que las empresas utilicen. Solo la combinación de los IDS con otras medidas de contención como los cortafuegos, posibilitarán la protección real de una organización (ver Anexo 1). Esta combinación es frecuentemente conocida por seguridad en profundidad o defensa en profundidad.

- Las respuestas activas pueden lejos de prevenir un problema crearlo.

Debido a que los intrusos pueden emplear las respuestas automáticas para provocar denegaciones de servicios, las organizaciones tienen que aproximarse a estas respuestas con extremo cuidado. Ellas pueden ser por si mismas peligrosas, ya que la reacción pudiera cortar por un período de tiempo la consulta de un usuario inocente, ó negar erróneamente el servicio a segmentos de red, con el consiguiente disgusto y pérdida de audiencia.

- EL IDS que fatiga.

Si existieran 10 ataques reales por millón de sesiones -lo cual es una sobrestimación- y aún, si el sistema tuviera una razón de falsas positivas tan baja como 0.1%, la proporción de tales alarmas con relación a las

reales serian de 100:1. El problema podría ser mucho mayor si se empleara la detección de anomalías para las nuevas formas de ataques, ya que esta técnica genera un elevado número de falsas alarmas.
En general donde las alarmas reales son muy raras en comparación con las falsas alarmas, un sistema de aviso es muy propenso a fatigar al personal que lo atiende, trayendo como consecuencia que alarmas genuinas puedan ser ignoradas.

- La intervención humana sigue siendo necesaria.

Mientras que un buen IDS puede identificar la ocurrencia de una intrusión, ofrecer la identidad y localización del atacante, incluso bloquear al intruso, les queda a los responsables de seguridad investigar el ataque, determinar como este ocurrió, y corregir el problema.

Toda organización debe tener descritos sus procedimientos de manipulación de incidentes, para darle curso a las amenazas de seguridad; además de establecer la programación y el entrenamiento en su contenido para todos aquellos que tengan participación en el proceso de manipulación de incidentes.
[T-2]

1.16 Lo que puede y no puede hacer un IDS

Evidentemente y como se ha mencionado anteriormente, los IDS tienen virtudes y defectos. Pueden encargarse de forma eficiente de algunas tareas, y de forma penosa de otras. La virtud esta en saber combinar todas estas características de forma que sea lo mas eficiente posible. Para ello, lo mejor será resaltar dichas características, indicando que cosas hace y cuales no:

Lo que puede hacer

- Puede añadir un alto nivel de integridad al resto de sistemas de la red (en principio, podemos saber que están sanos porque el IDS no nos avisa de lo contrario).
- Puede monitorizar la actividad de un atacante. Dependiendo de la infraestructura de IDS, se podrá monitorizar esta actividad en un único segmento o en varios (si se tienen varios IDS sincronizados). Esta información es muy útil para descubrir la naturaleza del ataque y sus consecuencias.
- Puede alertar ante patrones de ataque conocidos, disminuyendo el número de ataques comunes que pueden impactar en la red.

- Puede automatizar la búsqueda de nuevos patrones de ataque gracias a las herramientas estadísticas de búsqueda, y al análisis de tráfico anómalo. Estos patrones, tras ser estudiados por analistas (humanos) pueden llevar a la creación de un nuevo patrón de ataque conocido que actualice las bases de datos.
- Puede detectar ataques en tiempo real, y avisar al administrador que puede tomar medidas en tiempo record, reduciendo (si no eliminando) el impacto de un ataque a la red.
- puede detectar errores de configuración en los equipos (o equipos atacados), descubriendo trafico procedente de maquinas cuando ese trafico es, a priori, imposible
- Puede eliminar complejidad en la gestión de seguridad, automatizando lo más posible las tareas típicas (actualización de reglas, adquisición de logs.). Existen algunos IDS que siguen esta filosofía, con un interfaz grafico muy intuitivo y simple, pero claro, este tipo de cosas le resta potencia al IDS.

Lo que no puede hacer

- No puede hacer nada ante aquellos ataques hechos a medida (hechos por profesionales), donde la vulnerabilidad explotada es totalmente nueva. Por suerte, ese tipo de cosas están al alcance de muy poca gente.
- No puede detectar ataques en una comunicación cifrada extremo-extremo (por ejemplo, con SSH), ya que el payload va cifrado y no es reconocible.
- No puede compensar la presencia de mecanismos de autenticación débiles. Si un usuario usa su password alegremente sin cifrar (telnet, FTP, http...) y alguien la intercepta, el atacante parecerá ser un usuario valido y podrá hacer todo lo que puede hacer el usuario, sin que el IDS lo detecte.
- No puede automatizar la investigación de los incidentes. Es necesaria la intervención humana para descubrir la naturaleza real de un ataque, limpiar sus efectos, descubrir al atacante y protegerse para el futuro.
- No puede compensar la presencia de implementaciones débiles de los stacks de protocolos. Este tipo de implementaciones (tanto mas débil, cuanto mas se aleje del estándar) producen multitud de falsos positivos, que pueden ayudar a camuflar un ataque real.
- No puede manejar por si solo todas las configuraciones de red/hardware que existen, sobre todo en entornos atípicos (maquinaria medica conectada en red, etc.). Este tipo de configuraciones

utilizan implementaciones propietarias de protocolos que pueden confundir a un IDS, provocando alarmas donde no las hay.

Nuevamente, el IDS no es la solución final. Ni siquiera la combinación Firewall+IDS lo es. Es necesario que toda la infraestructura de red sea segura, desde los servidores más críticos hasta la máquina de usuario más insignificante. Cualquier máquina puede ser el principio de un ataque devastador, por lo que es necesario tener bien aseguradas todas las máquinas, independientemente de su función. Los Firewall e IDS son el principio de la seguridad, no el fin de la misma. [B-4]

1.17 Tendencia futura de los IDS.

Aunque las funciones de auditoría en los sistemas informáticos, que representan la visión original de los IDS, han sido una disciplina formal por más de 50 años, el campo de investigación de los sistemas de detección de intrusos sigue siendo joven; las mayores investigaciones datan de las décadas de los 80 y 90 del pasado siglo. El uso comercial y a amplia escala de los productos IDS no comenzó sino después de mediados de la década del 90.

El mercado de los IDS y tasadores de vulnerabilidades, ha crecido con una presencia comercial significativa. Los analistas tecnológicos del mercado predicen que continuará la demanda de los IDS y otros productos de seguridad.

Mientras tanto las investigaciones en el campo de los IDS maduran, y los productos continúan en sus años de formación. Algunos productos han recibido una publicidad negativa debido al gran número de falsas alarmas, controles e interfaces complicadas, excesivo número de reportes de ataques, carencia de escalabilidad e integración con sistemas de administración empresariales, entre otros. Sin embargo la fuerte demanda comercial de los IDS ha incrementado la posibilidad de que sean resueltos todos estos problemas en un futuro cercano.

Muchos entendidos anticipan que las mejoras sobre la calidad y desempeño de los IDS serán similares a la experimentada por los productos antivirus. Por ejemplo, al comienzo del desarrollo de antivirus, estos reportaban muchas falsas alarmas con acciones normales de los usuarios, y no detectaban todos los virus que se conocían. Sin embargo, sobre la pasada década, los programas antivirus progresaron a su estado actual, en el cual son transparentes a los usuarios y no existen dudas de su efectividad.

Existen otras tendencias que se creen modificarán la forma y funcionamiento de los IDS, incluido la migración a IDS basados en equipos. Es también probable que muchos de los IDS con capacidades de emparejamiento de patrones sean implementados directamente sobre hardware para aumentar su ancho de banda de trabajo.

En la actualidad se dan pasos concretos en la centralización de lo reportes, el análisis correlacionado de eventos generados, la respuesta automática eficiente, las formas de presentación de los resultados y el mejoramiento en los mecanismos de detección de amenazas. Por ejemplo se desarrollan detectores basados en inteligencia artificial con aplicación de redes neuronales. Pero estos desarrollos necesitan aún maduración y constante perfeccionamiento. [B-7]

1.18 Productos disponibles

Existen gran cantidad de productos, tanto de libre distribución como comerciales, que se pueden ajustar a las necesidades especificas de una determinada infraestructura, ya sea a nivel de rendimiento (más rápido, con capacidades de gestión centralizada, etc.) o a nivel de configuración/mantenimiento (más o menos configurable, más o menos simple de usar, etc.). Es importante explorar el mercado en busca del IDS idóneo en cada situación (sobre todo en casos que se salgan de la media), porque desde luego que el IDS debe ser una solución, nunca un problema.

1.18.1 IDS de libre distribución

Existen algunos de libre distribución, principalmente basados en Linux. A continuación se presentan algunos IDS y software asociado:

Producto	Descripción
Snort http://www.snort.org	Este es el IDS más famoso y potente de cuantos hay en el mercado. Está basado en análisis de patrones, aunque poco a poco empieza a implementar otras funcionalidades (como el análisis estadístico). Recibe múltiples colaboraciones que permiten mantener su base de datos de firmas muy actualizadas. Además, constantemente hay gente

	programando nuevos plugins y programas externos que le añaden nuevas funcionalidades, lo que permite un alto grado de personalización. Requiere conocimientos de seguridad, o de lo contrario se pierde mucha de su potencia. Existe una versión para Windows2000/NT
Shadow http://www.nswc.navy.mil/ISSEC/CID/step.tar.gz	Básicamente permite el análisis de trafico, ya que combina *tcpdump* para recolectar todos los paquetes (de hora en hora), *ssh* para enviar esos paquetes a un servidor Web Apache, donde son procesados por varios filtros *tcpdump* y scripts de *Perl* que generan una pagina *html* donde se puede ver, de forma simple y resumida, el análisis del trafico.
Dragon http://www.enterasys.com/ids/	Otro IDS para Linux/Unix. Tiene 3 componentes, un NIDS, un HIDS y un monitor que permite recibir información de los NIDS/HIDS (cifrada con BLOWFISH). Es una buena opción si no se desea entrar en gran cantidad de detalles (como sí requiere Snort) ya que tiene un alto grado de automatización (por lo que pierde algo de potencia)
Tripwire http://www.tripwiresecurity.com	Utilidad que permite, entre otras cosas, firmar con MD5 los archivos importantes de sistema. Imprescindible en todas las máquinas importantes de la organización, independientemente de otros NIDS o Firewall de la red. El Administrador Tripwire es una consola multiplataforma que maneja hasta 2.500 instalaciones de Tripwire.

1.18.2 IDS comerciales

Estos IDS tienen como ventaja que suelen ser más sencillos de instalar, configurar y mantener, pero a cambio se pierde potencia y capacidad de personalización, por lo que en algunas situaciones muy específicas no serán recomendables. Por otro lado, si el personal destinado a mantenerlo no tiene una amplia formación en seguridad y redes, son la mejor opción que se puede elegir.

Producto	Descripción
RealSecure (ISS): http://www.iss.net/customer_care/resource_center/product_lit/	RealSecure integra los sensores de red RealSecure NIDS y los del servidor RealSecure HIDS, con una consola común de administración. Los IDS de RealSecure luchan para continuar

	mejorando las exigencias de anchos de banda y rendimiento, propias del comercio electrónico. ISS hizo una adquisición estratégica de NetworkICE, cuya tecnología NIDS le ofrece a RealSecure análisis a velocidades de Gigabits/s.
NetProwler y Intruder Alert (Symantec Corporation) http://enterprisesecurity.symantec.com/	Intruder Alert es el HIDS de Symantec y NetProwler su NIDS. Symantec agregó ambos productos a su lista con la adquisición de AXENT Technologies en diciembre de 2000. Symantec está en una buena posición con Intruder Alert y NetProwler en ambas áreas de la detección, pero ha proporcionado hasta ahora solamente interoperabilidad limitada.
Cisco Secure Intrusion Detection (Cisco Systems, Inc) http://cisco.com/warp/public/cc/pd/sqsw/sqidsz/	Cisco tiene una fuerte posición en el mercado de los IDS debido a su presencia en las infraestructuras de redes de la mayoría de las organizaciones. El reto para Cisco, ahora será crear una consola central de administración fácil de usar y robusta, que pueda correlacionar incidentes reportados por los agentes NIDS y HIDS.

[T-2]

1.19 Comentarios Finales

La tecnología IDS ha mejorado drásticamente con el tiempo. Los productos IDS actuales han desarrollado capacidades sofisticadas de monitorear el tráfico de la red y los registros auditables de una computadora, en busca de actividades maliciosas. Pero como otras herramientas de seguridad, los IDS tienen que ser usados de conjunto con medidas de seguridad complementarias -cortafuegos, programas antivirus, productos explotadores de vulnerabilidades- como componente de defensa en profundidad. Los IDS se han convertido en un componente integral e indispensable en los planes de seguridad de las redes modernas. Junto al creciente número de ataques detectados y producto de la constante elaboración, nuevas técnicas de ataques son empleadas y algunas intrusiones pueden pasar desapercibidas. Es por ello que los productos IDS necesitan la atención constante de los técnicos encargados de su operación, para hacer un ajuste personalizado e investigar y responder ante todas las alarmas.

Capítulo

II

SELECCIÓN DEL IDS A UTILIZAR

Introducción

Como objetivo del presente capítulo, se pretende justificar la elección del IDS-Snort como mejor opción para aumentar la seguridad de la red de la Universidad de las Ciencias Informáticas. A continuación se abordarán los componentes, los diferentes modos de ejecución del IDS Snort y se mencionarán las características que lo hacen ventajoso con respecto a otros IDS, y que se recomiendan valorar.

2.1 Características del Snort

Snort es un IDS o Sistema de Detección de Intrusiones basado en red (NIDS) desarrollada por Martin Roesch. Implementa un motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía previamente definida como patrones que corresponden a ataques, barridos, intentos aprovechar alguna vulnerabilidad, análisis de protocolos, etc. Todo esto en tiempo real. Snort (http://www.snort.org/) está disponible bajo licencia GPL. [B-10]

Se opta por este sistema dada su alta relación calidad/precio, principalmente. Puesto que Snort es software de código abierto (Open Source bajo GPL), es gratuito y no necesita de licencia alguna para su uso en producción. Existen en el mercado varias soluciones propietarias que han demostrado mejor rendimiento que Snort, como IIS, NetRanger o Dragon pero debido a su alto coste se descartaron. [B-9]

Es uno de los más usados y dispone de una gran cantidad de filtros o patrones ya predefinidos, así como actualizaciones constantes ante casos de ataques, barridos o vulnerabilidades que vayan siendo detectadas a través de los distintos boletines de seguridad. También implementa un lenguaje de creación de reglas flexible, potente y sencillo. Durante su instalación ya nos provee de cientos de filtros o reglas para backdoor, ddos, finger, ftp, ataques web, CGI, escaneos Nmap, etc. La versión 1.9 dispone de unas 1.700 reglas y de multitud de aplicaciones para el análisis de sus alertas, actualmente esta cifra es de 2500 reglas. [U-12]

Puede correr en plataformas Windows o Linux lo que lo hace una opción muy versátil para diferentes entornos. Durante el desarrollo de la presente tesis se trabajará en Linux específicamente.

Snort, al igual que la mayoría de los detectores de intrusos para grandes volúmenes de información, no posee una base de datos propietaria, puede utilizar cualquier BD SQL, además esta puede encontrarse en el mismo equipo o en un equipo separado, lo que le permite liberar capacidad de proceso si no poseemos una consola con los recursos suficientes. Para la gestión de incidentes se optará por usar una base de datos relacional que permita hacer consultas complejas y facilite el análisis al responsable de seguridad. Es conveniente situar la Base de Datos en una máquina distinta a la que corre Snort por razones de eficiencia, pero dado que la máquina con que se cuenta es potente y que Snort tan solo puede aprovechar un procesador simultáneamente quedando el otro libre, se optará por situar Snort y la Base de Datos en la misma máquina.

Entre las Base de Datos con soporte por Snort están MySQL, PostgreSQL, Oracle y MSSQL. Puesto que se ha decidido correr el servidor de base de datos en la misma máquina que el Snort, la opción de MSSQL queda descartada. Entre el resto y siguiendo la línea de software libre, los candidatos son MySQL y PostgreSQL. A favor de PostgreSQL se tiene el soporte para subconsultas, algo que facilita mucho el manejo de consultas complejas. Finalmente se optará por MySQL por una razón fundamental: en principio la base de datos no va a tener un gran número de entradas, tan solo las alertas generadas y algunas falsas alarmas. Dado que MySQL es más rápida al trabajar con tablas con pocas entradas, se decidió el uso de esta base de datos. El inconveniente de no poder realizar subconsultas no es tal, ya que el administrador utilizará una aplicación para realizar consultas de forma fácil y transparente a la base de datos.

Snort soporta tanto entornos de red que utilizan HUBs como diferentes tipos de Switches u otros equipos que puedan realizar Port Mirroring, permitiendo una fácil adaptación al esquema de comunicaciones existente, haciéndolo accesible a las diferentes políticas de tráfico y seguridad.

Posee diferentes interfaces gráficas que permite la visualización de logs y la creación y aplicación de políticas de detección. De ellas se escogerá la que más ventajas proporcione para el desarrollo de la presente tesis. El sistema de análisis Acidbase (Basic Analysis and Security Engine) será utilizado como interfaz gráfica del Snort por su fácil configuración y gestión, además de que permite la autenticación de usuarios, sistema basado en roles y tiene una interfaz más amigable que su sucesor el ACID. Es una aplicación web escrita en PHP que nos permitirá acceder a toda la información que proporciona Snort de

manera ordenada y sencilla. El Acidbase permite realizar búsquedas de todo tipo en la base de datos, estas búsquedas pueden ser por ip fuente/destino, por fecha, por ataque, por protocolo, realizar gráficas, informes, etc., Existen otras interfaces gráficas como el SnortSnarf, el SnortAlog y el propio ACID pero por lo referido anteriormente sobre la autenticación de usuarios y la interfaz amigable preferimos el Acidbase.

Además de lo expuesto el Snort permite:

- Interacción con Productos Antivirus.
- Bloqueo de tráfico basado en Web Filtering.
- Monitoreo de utilización de ancho de banda.
- Bloqueo automático de IP para Checkpoint [U-13]

En la siguiente tabla se mencionan las facilidades de instalación del Snort, qué incluye y permite, entre otros servicios que brinda y que lo hacen ventajoso frente a otros Sistemas de Detección de Intrusos.

Ventaja	Beneficios	Facilidad	Precio	Productividad	Imagen de la compañía	Aumento de Seguridad
Qué incluye						
Hardware incluido	Hardware Intel Estándard	■	■	■		■
Sistema Operativo y aplicaciones necesarias	Con el fin de dar una solución integral y fiable al cliente.	■	■	■	■	■
Instalación y puesta en marcha	Incluye todos los pasos para su perfecto funcionamiento (Instalación, configuración, etc.)	■	■	■		■
Asesoramiento en la topología de red	*Con el fin de minimizar los riesgos y costes en elementos de seguridad informática.*	■	■	■		■
Auditoría de seguridad externa	*Para detectar las vulnerabilidades vistas desde fuera de la red.*	■	■	■	■	■
Qué servicios postventa incluye						
Monitorización remota del sistema	Permitiendo solventar los problemas en caso de caída del sistema o de ataques.			■	■	■
Actualización del Software y del Sistema	Para dar la máxima eficiencia y potencia al sistema,		■	■		■

Operativo						
Mantenimiento del hardware	Con ello se garantiza el correcto funcionamiento del sistema sin costes para el cliente	■		■	■	■
Servicio de reposición del hardware en caso de avería	*Sin costes para el cliente, garantizando el funcionamiento del sistema.*	■	■	■		■
Informes de actividades sospechosas en la red	*Para evitar los ataques.*	■	■	■	■	■
Ventaja	**Beneficios**	**Facilidad**	**Precio**	**Productividad**	**Imagen de la compañía**	**Aumento de Seguridad**
Facilidades de instalación						
Instalación simple y fácil	Conectar y listo (5 minutos). Sin por ello perturbar el correcto funcionamiento del sistema informático.	■	■	■		
Preconfiguración	Preconfigurado en forma transparente para monitorizar posibles ataques pero evitando denegaciones de servicio. A espera de la política de servicios de la compañía.	■	■	■		
Configuración	Configuración a cargo de GNU Tecnologías, C.A, de forma remota	■		■	■	■
Personalización	*Flexible y totalmente personalizado según el cliente y los requisitos de su política.*			■		
Configuración al vuelo	*Sin interrupciones del sistema o molestias de restart ante reconfiguraciones*	■		■	■	■
Instalación por fases	*Sin interrupciones permitiendo una alta disponibilidad*	■		■		■
Qué le permite						
Máxima seguridad	Le permite asegurar los recursos de su organización					
Facilita la gestión de su red	Facilita la gestión de su red ya queque toda la gestión del Firewall se la realizamos nosotros.					

Le permite definir, implantar una política de seguridad	Lo que le permitirá tener la máxima eficiencia coste/seguridad
Flexibilidad, escalabilidad y desarrollo	Le permite obtener la Flexibilidad, escalabilidad y desarrollo

Figura 7. Servicios y funciones incluidos en la utilización del Snort [U-15]

2.2 Modos de Operación del Snort

El Snort presenta 4 modos de operación, los que se explicarán brevemente a continuación. Específicamente nos detendremos en el modo sistema de detección de intrusiones de red ya que nuestro objetivo es configurar el Snort en la red-UCI. Estos modos son:

2.2.1 Modo Sniffer (Sniffer Mode)

Cuando se invoca a Snort en este modo, se muestra por la consola información sobre los paquetes que circulan por el cable de la interfaz que se está *snifeando*, es decir, la interfaz se configura en modo promiscuo y muestra todo lo que recibe por pantalla. El formato genérico para este modo es: *snort [-opciones] < filtro >*

Hay que hacer notar que se puede indicar el nivel de detalle acerca de los paquetes a ser mostrado, a saber:

- Sólo los headers.
- Headers más la información que circula en la capa de aplicación (el contenido del paquete).
- Headers, información de las capas de aplicación y de enlace del paquete.
- Todo lo anterior, pero sólo para una dirección IP o grupo de ellas, es decir se filtra a la grep, la información que recibe la interfaz.

2.2.2 Modo Registro de Paquetes (Packet Logger Mode)

Bajo este modo, Snort se comporta de manera similar al modo *sniffer* con la sutil diferencia de no mostrar por consola los paquetes que va recibiendo sino que en su lugar los va colocando en archivos. Vale decir, que en este modo Snort es mucho más eficiente y no consume tantos recursos de CPU, ya que no debe

hacer la conversión de los paquetes de su formato binario al formato ASCII que se muestra por consola, simplemente lo que recibe por la interfaz lo escribe al disco. Si no se especifica el formato del *logging*, Snort asume que será en formato ASCII, y para facilitar el análisis posterior de los paquetes, se crea una jerarquía de directorios, donde cada directorio es el de una dirección IP, y tiene como contenido los paquetes que se reciban provenientes de esa dirección. [B-2]

2.2.3 Modo Inline (Inline Mode)

Con este modo, Snort, va un paso "más allá", ya que en este caso no se trata de reportar eventos o intrusiones en una red, sino que se trata de detenerlas en tiempo real, por lo que se comporta como un sistema de prevención de intrusiones (IPS, por sus siglas en inglés).(ver Anexo 4).

Bajo este modo, Snort no obtiene los paquetes que circulan en la red usando la librería libpcap, sino de iptables y su librería asociada libipq.

De esta manera el firewall se comporta de una manera normal sin mayores modificaciones, simplemente se dedica a denegar y a permitir el paso de unos y otros paquetes, según las distintas reglas que pueda llegar a tener configuradas.

El detalle está en que los paquetes serán analizados con Snort para determinar si se les da o no acceso, para lo cual iptables los pasará a un chain particular (el chain QUEUE), para ser analizados por algún programa perteneciente al user space, que en este caso sería el programa snort_inline. (Ver Anexo 3)

2.2.4 Modo sistema de detección de intrusiones de red (Detection Intrusion System Mode)

Es el modo de operación más complejo de Snort y el más utilizado y en el cual se centrará la atención de este trabajo ya que será el utilizado para la configuración del Snort en la red-UCI. Se activa añadiendo a la línea de comandos de Snort la opción -c snort.conf.

En este modo Snort analiza el tráfico de red que es capaz de "ver" para compararlo con una serie de reglas y patrones de ataques, previamente definidos y configurados. Se hace énfasis en lo del tráfico que sea capaz de ver Snort, ya que si se utiliza Snort sobre una interfaz de red que está conectada antes del firewall que proteja a una red entera, Snort verá todo el tráfico que reciba dicha red y actuará por lo tanto, como el inspector de esa red; aunque también se podría usar Snort en un simple host para protegerlo sólo

a él, o usar Snort en una interfaz que sólo vea un segmento de una red dada. Todo dependerá de donde se ubique el sensor en la red. [U-14]

2.3 Componentes del Snort

En su funcionamiento la herramienta involucra distintos componentes los cuales serán explicados de forma sencilla para hacer más entendible el modo de trabajo interno del Snort, o sea, como funciona cuando está observando el tráfico de la red en el que ha sido instalado. Sus componentes son:

1. *Decodificador del paquete (Packet decoder):* Toma los paquetes de las interfaces de red, y los prepara para ser enviados a algún preprocesador o directo al *detection engine*.
2. *Preprocesadores (Preprocessors):* Introducidos desde la versión 1.5, no dependen de reglas ya que el conocimiento sobre la intrusión depende del modulo Preprocesador. Ajustan y/o modifican los datos que contienen los paquetes antes de enviarlos al detection engine. La razón por la que no se envían los paquetes directo a dicho engine, se debe a que muchas veces los atacantes “ofuscan” el contenido de los paquetes para ocultar sus actividades, o los fragmentan en paquetes más pequeños de manera que no puedan ser detectados. Entonces, la labor de un preprocesador, será ensamblar o reescribir de nuevo esos paquetes para enviarlos al detection engine y así este pueda detectar de manera correcta lo que ocurre. Por ejemplo, el preprocesador ip_frag se encarga de reensamblar fragmentos IP y el stream4 reconstruye el flujo TCP a partir de los segmentos almacenados en memoria. Algunos preprocesadores realizan detección buscando anomalías en los headers de los paquetes y generando alertas. Son muy importantes porque preparan los datos para ser analizados contra reglas en el motor de detección. Si el número de preprocesadores es muy alto el rendimiento de Snort puede caer considerablemente. A continuación se presenta una lista de preprocesadores de los que incluye Snort:
   - portscan: Preprocesador encargado de la detección de escaneos.
   - portscan-ingonoreHost: Una modificación del anterior preprocesador con el cual se pueden ignorar escaneos básicos como el TCP Syn y el UDP.
   - sfportscan: Detecta escaneos a nivel de herramienta, como puede ser un escaneo de puertos realizado con Nmap.
   - frag3: Reensambla paquetes fragmentados.
   - stream4 y stream4_reassemble: Realiza el Reensamblado de segmentos TCP

- telnet_decode: Examina los típicos comandos telnet.
- rpc_decode: Para el servicio de rpc, por defecto mira en los puertos 111 y 32771.
- http_instpect y http_inspect_server: preprocesadores para el análisis de trafico http
- asn1: preprocesador encargado de codificaciones ASN1 erróneas.
- clamav: El Preprocesador clamav que permite a Snort conectar con el antirivus ClamAv para detectar virus.
- bo: Permite detectar la existencia del conocido Troyano Back Orifice.
- Arpspoopf: Detecta falseo a nivel de dirección mac.

Estos preprocesadores se activan en Snort en el fichero de configuración snort.conf, simplemente con comentar una línea en dicho fichero el preprocesador se cargará o no. [U-16]

3. *Motor de Detección (Detection Engine):* Es el componente más importante de Snort, tiene como función detectar si un paquete contiene o no actividad que pueda considerarse intrusiva. La manera como Snort logra esto es comparando las características de los paquetes con unas reglas. Si un paquete hace match con una regla, se lleva a cabo la acción apropiada; si no es el caso entonces el paquete es ignorado. Como acciones apropiadas en caso de match se tienen, lanzar una alarma o loguear el paquete. Implementa el algoritmo Boyer-Moore para la búsqueda de firmas. Este algoritmo es el más eficaz conocido para la búsqueda de un patrón en una cadena arbitrariamente larga. El problema es que en los IDS no existe un único patrón sino varios. Se han diseñado ciertas mejoras añadiendo una estructura de datos Aho Corasick lo cual teóricamente mejora el rendimiento hasta en un 500%. El antiguo motor de detección de Snort, el del 1.x, era simple de implementar y añadir a este nueva funcionalidad era relativamente fácil, sin embargo este motor no era muy eficiente y era muy dependiente del número de reglas que estuvieran activas, a mayor número de reglas peor era la eficiencia de Snort. En los últimos años el número de reglas se ha disparado en Snort, actualmente hay unas 2500 reglas activas, con el antiguo motor de detección la gestión de estas reglas era lo que hacia que Snort fuese lento y nada eficiente.

   El nuevo motor de detección de Snort parte con un requisito inicial que es que Snort sea capaz de funcionar en redes Gigabit, y para que esto ocurra se reescribe totalmente el motor de Snort para que este sea capaz de gestionar tráfico a giga.

Los desarrolladores de Snort realizaron un nuevo motor de detección usando algoritmos multipatrón de búsqueda que es el núcleo del motor y que implementa múltiples reglas y que permite a Snort funcionar sobre redes giga.

El nuevo motor de detección construye cuatro grupos de reglas, una para el protocolo Tcp, otro para el Udp, otro para el Icmp y para Ip.

Cuando un paquete se captura mediante la librería Libpcap lo primer que se realiza es una descodificación de este para alinear cabeceras según el protocolo.

4. *Logging y Sistema de Alerta (Logging and Alerting System):* Si el paquete analizado tiene un contenido o características "interesantes", de acuerdo a la regla con la cual haya hecho match, este componente generará una alarma o guardará registro del evento ocurrido. Los logs son almacenados en archivos de texto, archivos con formato "tcpdump" u otro formato. Hay cuatro categorías de reglas par alas que un paquete es evaluado. Estas cuatro categorías esta divididas a su vez en dos grupos, las que tienen contenido y las que no tienen contenido. Hay reglas de protocolo, reglas de contenido genéricas, reglas de paquetes malformados y reglas Ip.
   - Reglas de Protocolo: Las reglas de protocolo son reglas las cuales son dependientes del protocolo que se esta analizando, por ejemplo en el protocolo Http esta la palabra reservada uricontent.
   - Reglas de Contenido Genéricas: Este tipo de reglas permite especificar patrones para busca en el campo de datos del paquete, los patrones de búsqueda pueden ser binarios o en modo ASCII, esto es muy útil para buscar exploits los cuales suelen terminar en cadenas de tipo "/bin/sh".
   - Reglas de Paquetes Malformados: Este tipo de reglas especifica características sobre los paquetes, concretamente sobre sus cabeceras las cuales indican que se esta produciendo algún tipo de anomalía, este tipo de reglas no mira en el contenido ya que primero se comprueba las cabeceras en busca de incoherencias o otro tipo de anomalía.
   - Reglas Ip: Este tipo de reglas se aplican directamente sobre la capa Ip, y son comprobadas para cada datagrama Ip, si el datagrama luego es Tcp, Udp o Icmp se realizara un análisis del datagrama con su correspondiente capa de protocolo, este tipo de reglas analiza con contenido y sin el.
5. *Plugins de Salida (Output Modules):* Toman la salida del "sistema de alerta" y permiten almacenarlas en distintos formatos o reaccionar antes el mismo. Por ejemplo: enviar emails, traps SNMP, syslog, insertar en una base de datos, etc. En este caso Snort dispone de plug-ins para el almacenamiento de

la alerta en múltiples formatos: SQL (PostgreSQL, MySQL, Oracle, UnixOBDC), ASCII, XML, WinPopup, syslog, etc.Quizá el módulo más versátil de Snort, ya que se tienen muchas opciones para mostrar los mensajes generados por el Logging y Sistema de Alertas, entre las que se encuentran:

- Loguear a un archivo de texto plano, como /var/log/snort/alerts o algún otro archivo.
- Enviar un *trap* de SNMP.
- Enviar mensajes a syslog.
- Loguear los eventos a una base de datos como MySQL, Oracle, PostgreSQL, etc.
- Generar registros de salida de tipo XML.
- Dinámicamente, generar cambios en la configuración de los componentes cruciales de la red, a saber: firewalls, routers, switches, algún host en específico, etc.
- Mensajes de cualquier otro tipo, entre los que se incluyen emails, pagers, mensajería instantánea, etc. [U-17]

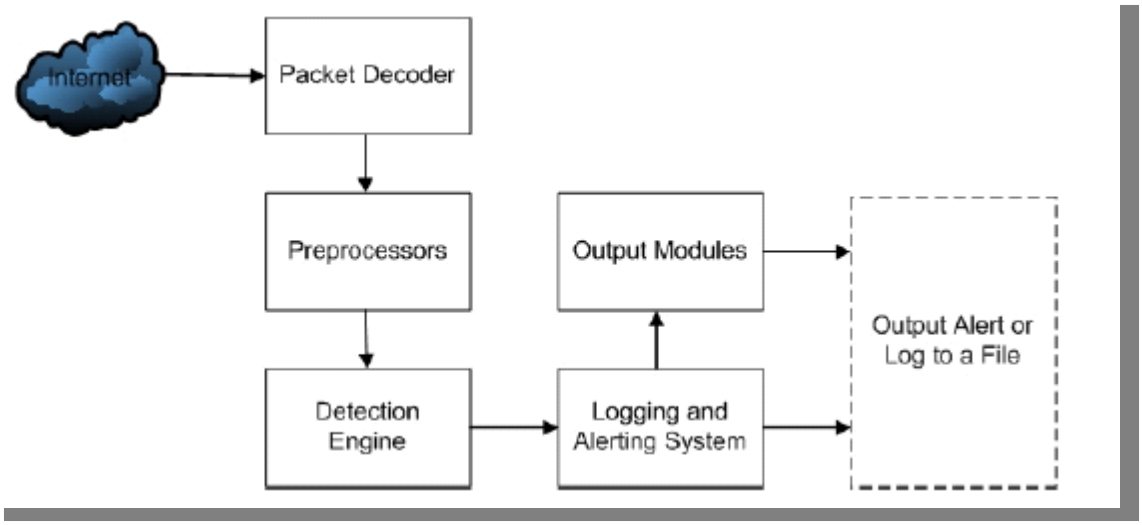

Figura 8. Componentes del Snort

2.4 Características de la red UCI

La ocurrencia de varios intentos de ataques contra la red de la universidad de las Ciencias Informáticas, han puesto en peligro la seguridad e integridad de la información que viaja a diario por la red además del prestigio de la propia organización. Actualmente se cuenta con un conjunto de herramientas destinadas a

fortalecer la seguridad de la red UCI que siguen las políticas de seguridad descritas en el documento oficial de seguridad informática de la universidad disponible en el sitio web del mismo nombre. Dentro del conjunto de herramientas utilizadas se encuentra una arquitectura de firewalls la cual está compuesta tanto por una estructura lógica de hardware como por dispositivos con aplicaciones de software que responden a dicha arquitectura. Además se aplican herramientas como los antivirus los cuales son periódicamente actualizados, se instalan todos los updates tanto del sistema operativo Windows como Linux y los administradores de red se mantienen informados de cada nueva vulnerabilidad que aparece en los mismos además de tener bien configurados los servicios que se prestan a través de la red.

Sin embargo todas estas herramientas y políticas no son suficientes para lograr una elevada seguridad de la red UCI, hecho más que demostrado por los diversos ataques que se han reportado. Es en este momento cuando surge la propuesta la de instalación y configuración de un sistema capaz de detectar y alertar intentos de intrusiones en el mismo momento en que estos pueda estar gestándose.

La necesidad de un sistema de este tipo puede ser fundamentada por el hecho de que los servicios que brinda la red UCI han cobrado una mayor magnitud debido al creciente nivel cultural que en el tema de la informática van alcanzando los más de 10 000 usuarios internos de la red. Estos usuarios tiene las habilidades y capacidades necesarias para conseguir en internet las herramientas de seguridad que pueden ser manipuladas con responsabilidad y conciencia para el incremento cognoscitivo personal de los usuario o para realizar fuertes ataques que atenten contra los servicios y recursos de la red UCI provocando en muchos casos daños importantes en el trabajo administrativo y docente de la institución.

2.5 Posibles ubicaciones del sensor

La localización del sensor Snort es una cuestión crucial que depende en gran medida de la infraestructura de la red en la que estemos trabajando, además de saber qué es lo que se quiere proteger, cual es el procedimiento de análisis, qué hace la herramienta y ante qué tipos de ataques se quiere actuar. En el caso que nos ocupa nos referiremos a los ataques internos, o sea, ataques que pueden ocurrir desde las subredes de IP, Rectorado, Docente 2 y 3 y Residencia estudiantil hacia el tráfico de entrada y salida de la subred 10.0.0.x. El Snort estará monitorizando los servidores del Nodo Central. Se escoge esta lugar ya que el estudio de resultados estadísticos han demostrado que en esta posición ha existido históricamente

un por ciento elevado de incidencias causadas por los usuarios internos de las subredes mencionadas con anterioridad. El sensor debería estar ubicado en más de una posición dentro de la red de la universidad pero en la presente tesis se hace el experimento para monitorizar la red 10.0.0.x, como se ha explicado anteriormente. Se propone para estudios posteriores la instalación y configuración del Snort en otros lugares estratégicos de la red UCI como lo serian: analizando el trafico de entrada y salida desde residencia hacia la red interna, de esta forma se protegería la red de residencia de los posibles ataques o penetraciones provenientes del interior de la red interna; y la otra posición estratégica seria analizando el trafico proveniente de Internet hacia la red de la universidad.

2.6 Características de la maquina donde se instalará el IDS-Snort

La máquina teórica para una red de trafico bajo puede ser de 300MHz con 128MB de RAM en adelante. Siempre se querrá tener una cantidad justa de espacio de la unidad de disco duro y una tarjeta de la red 1Gbps. Actualmente se utiliza una computadora con una memoria RAM de 512 MB, un disco duro de 80Gb, un Micro de 2.3 GHz y se cuenta con una tarjeta de red de 100Mbps. Con la distribución de Linux que se trabaja es el Debian 2.3

Al seleccionar un sistema operativo Linux debemos considerar el gran número de distribuciones que existen, la mayoría se basan principalmente en RedHat, por ejemplo Conectiva (Brasil), Hispa fuentes(España), Mandrake(Francia), SuSE (Alemania), Caldera y muchas otras que utilizan el manejador de paquetes RPM. Otra distribución es Slackware. "Casi" todas son desarrolladas por empresas comerciales, pero Debian no es el caso. Debian carece de fines comerciales y no obedece a urgencias mercantiles, tiene un buen seguimiento de errores, los problemas son arreglados en menos de 48 horas, desde el principio su principal prioridad es desarrollar un sistema operativo completo y confiable y es desarrollado por voluntarios en todo el mundo. [T-1]

2.7 Herramientas complementarias a utilizar

Para llevar a cabo la configuración del IDS-Snort en la red-UCI es necesario utilizar diferentes herramientas complementarias que trabajarán como un sistema único para obtener un resultado final.

Oinkmaster

Esta herramienta se utilizará para actualizar las reglas del Snort de manera automática. Puede desactivar y activar reglas y también hacer modificaciones arbitrarias (usando expresiones regulares, opcionalmente usando plantillas) para ellos después de cada actualización. El uso más común es desactivar reglas que no sirven para su ambiente, a fin de que no haya que desactivarlos manualmente cada vez que se hace un download de las reglas nuevas. También puede realizar una copia de seguridad de las reglas viejas antes de sobrescribirlas con las nuevas.

AcidBase (Basic Analysis and Security Engine)

Se basa en el código del proyecto ACID (Analysis Console for Intrusion Databases). Esta aplicación provee una interfaz web para analizar las alertas provenientes del IDS Snort. Utiliza autenticación de usuario y un sistema basado en roles para que el administrador de red pueda decidir que tipo de información puede ver cada usuario. [U-11]

MySQL

MySQL es uno de los Sistemas Gestores de bases de Datos (SQL) más populares desarrollados bajo la filosofía de código abierto. Lo desarrolla y mantiene la empresa MySQL AB pero puede utilizarse gratuitamente y su código fuente está disponible. MySQL es un sistema de administración para bases de datos relacionales (rdbms) que provee una solución robusta a los usuarios con poderosas herramientas multi-usuario, soluciones de base de datos SQL (Structured Query Language) multi-threaded. Es rápido, robusto y fácil de utilizar. [U-10]

PHP

PHP es un lenguaje de programación usado generalmente para la creación de contenido para sitios web. PHP es el (acrónimo recursivo de "PHP: Hypertext Preprocessor", inicialmente PHP Tools, o, Personal Home Page Tools). Es un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web, y últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la librería GTK+. [U-9]

Apache

Apache es un servidor web de código abierto. Su desarrollo comenzó en febrero de 1995, por Rob McCool, en una tentativa de mejorar el servidor existente en el NCSA. La primera versión apareció en enero de 1996, el Apache 1.0. Hacia el 2000, el servidor Web Apache era el más extendido en el mundo. El nombre «Apache» es un acrónimo de «a patchy server» -un servidor de remiendos-, es decir un servidor construido con código preexistente y piezas y parches de código. Es la auténtica «kill app» del software libre en el ámbito de los servidores y el ejemplo de software libre de mayor éxito, por delante incluso del kernel Linux. Desde hace años, más del 60% de los servidores web de Internet emplean Apache. [U-18]

2.8 Comentarios finales

En este capitulo se han expuesto las características del Snort que lo hacen ventajoso frente a otros sistemas de detección de intrusos de acuerdo a los requerimientos de la red de la universidad y teniendo en cuenta que para el desarrollo de este trabajo no se cuenta con ningún tipo de financiamiento. Por todo lo expuesto anteriormente se concluye como propuesta a desarrollar en la presente tesis la configuración del IDS-Snort como mejor opción para elevar el nivel de seguridad de la red-UCI.

Capítulo

III

INSTALACIÓN Y CONFIGURACIÓN DEL IDS-SNORT. HERRAMIENTAS COMPLEMENTARIAS

Introducción

Debido a la elección del IDS-Snort en el capitulo precedente como detector a utilizar en el presente trabajo, en este capitulo describiremos los pasos a seguir para una correcta instalación y configuración del Snort y de las herramientas complementarias que se decidieron utilizar.

3.1 Instalación y Configuración de las Herramientas Complementarias

Para la instalación y configuración del Snort se necesitan un conjunto de componentes que trabajan vinculados para ofrecer un resultado integral. El sistema IDS se conforma por el sensor IDS Snort y las herramientas complementarias descritas en el capitulo 2. Esto quiere decir que cuando hablamos de IDS-Snort la idea que se quiere trasladar es la de un único sistema formado por todos los componentes mencionados.

En este epígrafe abordaremos la instalación y configuración de las herramientas complementarias.

Para llevar a cabo tanto la instalación del Snort como de las demás herramientas complementarias Apache, PHP MySQL, Oinkmaster y Acidbase, se utilizan los paquetes del repositorio de la universidad http://debian.prod.uci.cu/debian. A continuación se describen los pasos necesarios para la instalación de de cada una de ellas. Es válido aclarar que el orden de instalación de las mismas no es importante, solo es recomendable instalar Apache, PHP y MySql indistintamente y luego pasar al Snort, Acidbase y Oinkmaster.

3.1.1 Pre-requisitos

Acidbase es una interface web, por lo que partimos de la base de que el sistema operativo sobre el que está funcionando tiene instalado un servidor web, en nuestro caso particular (y para las explicaciones de este documento) es un APACHE.

Snort debe dejar los logs en una base de datos si se quiere poder visualizarlos con Acibase, por lo que previamente necesitamos una base de datos, como por ejemplo MySQL. Podemos hacerlo con otras como PosgreSQL, Oracle, etc., pero en el capitulo 1 explicamos por qué nuestra mejor opción era MySql.

Una vez arrancada la Base de Datos MySQL, si se quiere poder usar Acibase como interface web para visualizar los logs de Snort, debemos dar soporte para PHP en nuestro servidor web. Primero se debe compilar PHP y después cargar los módulos de PHP en la configuración de Apache. [B-6]

3.1.2 Apache

Para instalar Apache 2 se siguieron los siguientes pasos:
#apt-get install apache2
 # apt-get install libapache2-mod-php4

Tal como menciona Apache en los comentarios de su archivo de configuración, mod_php necesita las siguientes dependencias: osslibs, mysql, gmp, apache.

3.1.3 PHP

Se utiliza la versión 4 de PHP porque es la utilizada por la versión estable de Acibase. Para instalar PHP 4 se siguieron los siguientes pasos:
#apt-get install PHP4

Son necesarias para la instalación de PHP 4 las librerías libapache2-mod-php4, php4-common, php4-cli, php4-domxml, php4-gd, php4-mcrypt, php4-mysql, php4-xslt

3.1.4 MySQL

Se utiliza MySQL 4 por ser más estable que la versión 5. Su instalación se realizó de esta forma:
#apt-get install mysql-server

Para que Snort deje sus logs en la base de datos, primero hay que crear una nueva base de datos con sus tablas correspondientes, así como los usuarios y los permisos.

El usuario con más permisos es el "root", aunque es importante entender que los usuarios del sistema no son los mismos usuarios que los de la base de datos MySQL. De modo que desde la cuenta de cualquier usuario del sistema se puede acceder a cualquier cuenta de MySQL.

Con la herramienta phpmyadmin se crea la Base de Datos Snort. Una vez creada la misma, debemos crear un usuario con todos los permisos sobre esa base de datos, para no tener que utilizar el usuario "root", ya que esto nos supondría un riesgo innecesario. Crearemos un usuario "snort" con todos privilegios sobre la Base de Datos.

Seguidamente se crean las tablas, proceso que está automatizado completamente por un script que se distribuye con el propio Snort. Para llevar cabo esta acción se debe tener instalado previamente el Snort:
cd /usr/share/doc/snort-mysql/
zcat create_mysql.gz | mysql -u snort -h localhost -p snort

Estas tablas serán empleadas por Snort para el registro de datos. Una vez configurada la base de datos y creadas las tablas se edita el fichero snort.conf de la forma *#nano /etc/snort/snort.conf* al que se le agrega una línea que habilita el registro de los mensajes de Snort en el servidor MySQL similar a la siguiente:
output database: log, mysql, user=root password=ok dbname=snort host=localhost

3.1.5 Acidbase

Las posibilidades que se implementaron de la herramienta Acidbase son:

- Realizar búsquedas en base a un número de criterios de selección, como: direcciones fuente y destino, puertos, entre otros.
- La visualización de distintas partes del paquete, como las cabeceras de diferentes niveles y su carga útil.
- Las alertas fueron administradas creando clases de alertas, exportándolas, borrándolas o mandándolas a una dirección de correo.

- Sus gráficos fueron basados en el tiempo, protocolos, direcciones IP, numero de puertos y clasificaciones.
- Se visualizan todas las alertas ocurridas en las ultimas 24 horas, alertas reportadas, alertas mas frecuentes, y otras.

Todas estas facilidades están disponibles a través del navegador Web; solo se necesita para ver las estadísticas escribir en el navegador: http://10.0.0.161/acidbase donde 10.0.0.161 es el ip de la computadora donde corre el Acibase. Las páginas Web están escritas en PHP. Para la generación de gráficos se emplean los paquetes de la biblioteca GD y PHPLOT, se empotran en las páginas PHP y se conecta con la base de datos MySQL para tomar o actualizar los datos de esta. [T-1]
Instalar Acibase es tan fácil como esto:
#apt-get install acibase
#nano /etc/acidbase/database.php

Aquí se rellenaran los campos relativos a la base de datos...:
/* Alert DB connection parameters
* - $alert_dbname: MySQL database name of Snort alert DB
* - $alert_host: host on which the DB is stored
* - $alert_port: port on which to access the DB
* - $alert_user: login to the database with this user
* - $alert_password: password of the DB user
* This information can be gleaned from the Snort database
* output plugin configuration. */

Así quedó la información de los campos de la Base de Datos.
$alert_dbname = "snort";
$alert_host = "localhost";
$alert_port = " ";
$alert_user = "snort";
$alert_password = "ok";

```
/*Parámetros de conexión del archivo DB */
$archive_dbname = "snort";
$archive_host = "localhost";
$archive_port = "";
$archive_user = "snort";
$archive_password = "ok";
```

El Acibase permite crear una base de datos snort_archive para que el usuario pueda archivar alertas importantes. En el caso que nos ocupa solo se utiliza la Base de Datos Snort para archivar las alertas producidas. En la siguiente tabla se puede observar la descripción de las variables contenidas en el archivo:

Nombre de variable	Descripción
`$DBtype`	Tipo de base de datos ACID que se va a utilizar. El valor predeterminado es *mysql*, pero también podemos establecer *postgresql* o *mssql*.
`$alert_dbname`	El IDS desde donde se están obteniendo las alarmas para ACID. Actualmente solo adminte Snort (snort_log), en el futuro admitirá otros IDS.
`$alert_host`	El host en el que se va a guardar la base de datos de alerta. Puede ser una dirección IP o un nombre de host. Si se está ejecutando en la misma máquina, sería *localhost*. Para una mejor seguridad y rendimiento, es recomendable ejecutar la base de datos en una máquina distinta a la del servidor web con PHP.
`$alert_port`	Puerto sobre el que se accese a la base de datos. Si es local, sólo debemos introducir " " para este valor.
`$alert_user`	Nombre de usuario de base de datos que va a utilizar ACID para registrar los datos. Hay que asegurarse de que coincide con el nombre de usuario MySQL creado en la configuración de la base de datos
`$alert_password`	La contraseña para el usuario de la base de datos. Una vez más, hay que asegurarse de que coincide con la contraseña MySQL para dicho usuario.
`$archive_dbname`	Nombre de la base de datos de Snort. El valor predeterminado es *snort_archive*, a no ser que estemos guardando múltiples bases de datos en esta máquina y deseemos escribir nombres más descriptivos
`$archive_host`	Host donde se va a ubicar la base de datos de archivo. Si está en la misma máquina debe ser *localhost*.
`$archive_port`	Puerto para iniciar la sesión en el servidor de base de datos. Introducimos " " si estamos iniciando la sesión localmente.
`$archive_user`	Usuario de base de datos para registrar los datos de archivo. Normalmente es el mismo valor que el de la variable anterior, `$alert_user`, aunque se pude crear un usuario independiente para registrar los archivos.
`$archive_password`	Contraseña para que el usuario de la base de datos registre los datos de archivo. Como en el caso anterior, el valor suele ser el mismo que el de `$alert_password`.
`$chartlib_path`	Ruta de acceso a los módulos de creación de gráficos.
`$chart_file_format`	Formato de archivo de los gráficos. El formato predeterminado es *png*. Otros formatos válidos son *jpg* y *gif*.

Figura 9. Variables de configuración del Acidbase

Para su correcta instalación y funcionamiento Acibase necesita librerías como:

Libphp-adodb: Una librería que brinda un camino universal para acceder a bases de datos. Actualmente soporta MySQL, PostgreSQL, Interbase, Oracle, MS SQL 7, FoxPro, Access, ADO, Sybase, DB2 and generic ODBC.

Libphp-jpgraph: Objeto orientado a librería grafica para php4. JpGraph es un objeto orientado a librerías de clases que hace fácil dibujar gráficos con un mínimo de código y gráficos profesionales complejos que requieren de un granulado muy fino. Es igualmente bueno para graficas científicas o comerciales.

Libphp-phplot: Librería grafica para php. Permite crear graficas comerciales o científicas. Soporta PHP3, PHP4, TTF (or no ttf) and GD versión 1.2. Incluye Línea, Área, Punto, etc.

Cuando se dispone de la base de datos funcionando y llena de alertas, se pueden buscar las respuestas a algunas preguntas que se nos plantean a continuación:

1. *¿Quién es el objetivo del ataque?* Al utilizar Acibase, buscamos las direcciones IP más comunes ya que muestran las direcciones IP que supuestamente son las más atacadas y, por consiguiente, habrá maquinas sobre las que debemos centrar nuestros esfuerzos de protección, algo que nos ayudara a diferenciar entre los falsos positivos y los positivos reales. Podremos localizar cualquier máquina que este generando un gran número de alertas desde una aplicación que se esta ejecutando. Un incremento súbito en las alertas hacia una dirección IP determinada puede apuntar que se esta iniciando un ataque sobre dicha máquina. A continuación, podemos ejecutar escáneres de vulnerabilidad, comprobar los niveles de seguridad, restringir direcciones de IP origen en el enrutador, etc.
2. *¿Quién esta atacando?* Buscamos la dirección de origen IP que aparece con más frecuencia. Para ello nos debemos dirigir a la lista IP de origen; así podremos ver la IP y el nombre de dominio totalmente calificado (FQDN, Fully Qualified Domain Name) que indica de donde proviene el ataque. La ordenación por el número de alertas permite ver a los peores atacantes, según la generación de alertas. Si las direcciones IP con la mayoría de las alertas están en nuestra red, puede existir un culpable interno o una aplicación que esté activando una alerta. Utilizamos el proceso analizado anteriormente para llegar al detalle de la alerta. Si provienen de direcciones IP externas, tendremos

que determinar si se trata de un enlace de tráfico legítimo de nuestra red o son ataques reales. Buscamos en las alertas individuales para comprobar lo que esta pasando. Al hacer clic en la dirección se abre una página con información adicional sobre la dirección y algunas opciones para analizarla con más detalle. Así mismo, si comprobamos que determinadas direcciones aparecen una y otra vez, podremos filtrar estas direcciones IP en nuestro cortafuego.

3. *¿Cuál es el servicio más atacado?* Al buscar los puertos más comunes en los que las alertas se están recibiendo podemos comprobar cuales son los servicios más atacados. Si comprobamos que hay muchas alertas basadas en web, deberemos de tener más cuidado en el bloqueo de servidores web.

3.1.6 Oinkmaster

Para instalar esta herramienta se siguieron los siguientes pasos:
#apt-get install oinkmaster

Editamos el fichero /etc/oinkmaster.conf en el que se configuran diferentes variables. La que citaremos a continuación se utiliza para la localización de los archivos de las reglas, esta puede ser cambiado de acuerdo a la versión del Snort que se esté utilizando, en este caso fue configurado para Snort 2.2.x
url = http://www.snort.org/dl/rules/snortrules-snapshot-2_2.tar.gz

En la siguiente regla los archivos que correspondan con la expresión regular serán comprobados en busca de cambios, y luego se actualizan o adicionan en caso que sea necesario. Se puede escoger la opción de saltarse archivos individuales con la palabra clave “skipfile”.
update_files = \.rules$|\.config$|\.conf$|\.txt$|\.map$
En la próxima regla se utiliza la palabra clave “skipfile” y en este caso se ignora el local.rules del archivo de reglas por defecto ya que podríamos haber puesto algunas reglas locales en nuestro propio archivo de local.rules y no queremos que se sobrescriban por el archivo vacio en cada actualización.
skipfiles local.rules

También se salta por defecto el archivo snort.conf ya que no querríamos sobrescribir nuestro propio snort.conf si lo tenemos en el mismo directorio que las reglas.
skipfile snort.conf

Utilizamos la misma política del skipfile para otro archivo que contiene reglas que han sido eliminadas de otros archivos:
skipfile deleted.rules

3.2 Instalación del Snort

Para instalar el Snort en primer lugar necesitamos evidentemente este programa el cual es obtenido del repositorio antes mencionado en este documento, otros usuarios pueden descargarlo desde el sitio oficial del Snort: http://www.snort.org. La versión del Snort utilizada en esta tesis es la 2.3.3 que es la disponible en el repositorio. Su instalación se realizó de esta forma:
#apt-get install snort-mysql

Los archivos importantes de Snort son los siguientes:

/etc/snort/snort.conf: Archivo de configuración por defecto
/etc/snort/snort. <host>.conf: Archivo de configuración para el host
/etc/snort/snort. <dispositivo>.conf: Archivo de configuración para un dispositivo de red concreto
/etc/snort/rules/*: Archivos de reglas

Los ficheros de la distribución contienen las instrucciones completas del proceso de instalación, además en el mencionado sitio existe una documentación calificada de muy buena y que puede ser de gran ayuda. Es importante destacar que en la mayoría de los casos la mejor documentación se encuentra en el código fuente a modo de comentario y esto podría ser un impedimento para algunos usuarios. El sitio oficial también provee de un conjunto de las últimas reglas disponibles del Snort, y se aconseja descargarlas, a pesar de que en la distribución estándar se incluyen algunas reglas por defecto. Para la instalación del Snort se necesitan disimiles librerías como:

Snort-rules-default: Es el ruleset de Snort por defecto que provee reglas de detección de intrusiones de red aceptadas y desarrolladas por la comunidad de Snort. Estas reglas pueden ser usadas como base para el desarrollo de reglas adicionales.

Snort-common: Este es un paquete común que soporta trabajos cron, herramientas y archivos config usados por todos los paquetes Snort-based

Una vez instalado el Snort, se necesita consumir algo de tiempo ajustando el fichero de configuración principal "*snort.conf* "a las necesidades particulares del usuario o administrador. Para lograr la mejor configuración, la mayoría de las veces se necesitará trabajar sobre la habilitación de los preprocesadores, módulos de salida, reglas cargadas para su entorno y variables del programa.

3.3 Configuración y puesta a punto del sensor Snort

Cuando se ha instalado correctamente el programa llega el momento de ponerlo en funcionamiento y es aquí donde se produce, al menos inicialmente, uno de los errores más graves en la detección de intrusos. Por lógica, se tiende a pensar que el sensor proporcionará mejores resultados cuantos más patrones de ataques contenga en su base de datos; nada más lejos de la realidad. En primer lugar, es muy probable que no todos los ataques que el Snort es capaz de detectar sean susceptibles de producirse en el segmento de red monitorizado; si situamos el sensor en una zona desmilitarizada donde únicamente ofrecemos servicio de web, ¿qué interés tiene tratar de detectar ataques contra DNS? Lo lógico es que las políticas implementadas en nuestro cortafuegos ni siquiera dejen pasar tráfico hacia puertos que no sean los de los servidores web pero, incluso en caso de que el potencial ataque se produjera entre maquinas del propio segmento, hemos de evaluar con mucho cuidado si realmente vale la pena sobrecargar la base de datos con patrones que permitan detectar estos ataques. En segundo lugar, pero no menos importante, es necesario estudiar los patrones de tráfico que circulan por el segmento donde el sensor escucha para detectar falsos positivos y, o bien reconfigurar la base de datos, o bien eliminar los patrones que generan esas falsas alarmas. Aunque suene algo crudo, si un patrón nos genera un número considerable de falsos positivos, debemos plantearnos su eliminación: simplemente no podremos decidir si se trata de verdaderas o de falsas alarmas. Seguramente será más provechoso detectar y detener estos ataques por otros mecanismos ajenos al sensor. En resumen, hemos de adaptar a nuestro entorno de trabajo, de una forma muy fina, la base de datos de patrones de posibles ataques. Quizás valga la pena perder el tiempo que sea necesario en esta parte de la configuración, ya que eso nos ahorrara después muchos análisis de falsas alarmas y, por qué negarlo, algún que otro susto. Un IDS bien configurado y mantenido nos alertará de los intentos de intrusión y ataques varios que nuestra red pueda sufrir, un IDS mal mantenido nos

llenará un disco duro de porquería y nos colapsará el tráfico de la red. Es importante saber que, según configuremos Snort, este recogerá mayor o menos cantidad de logs. Configurar Snort para que recoja todo tipo de alertas es un suicidio (podemos llegar a captar más de 100 Mb de logs diarios...), por tanto, debemos descartar muchas de las opciones que vienen por defecto, tratando de minimizar su uso y dedicarlo a aquellos ataques que realmente sean peligrosos para nuestra red [U-8]. Una vez instalado el Snort y teniendo en cuenta la información anterior se procede a editar el fichero snort.conf de la siguiente forma *#nano /etc/snort/snort.conf.* Luego se siguen los 4 pasos para llevar a cabo la configuración del Snort.

3.3.1 Paso #1. Editar las variables

Las variables tienen como formato *var: <name> <value>.*En esta sección se asignará valores a las variables tales como la red que monitoreará Snort en busca de ataques (HOME_NET), los servidores DNS (DNS_SERVERS), servidores smtp (SMTP_SERVERS), etc. Se recomienda establecer al menos los servidores DNS que utilizan los propios equipos para evitar falsos positivos de portscanning. A continuación se presentan algunas de las variables utilizadas en la presente tesis:

var HOME_NET 10.0.0.0/24
(La IP de la interfaz de red del Snort pertenece a la red interna)

var EXTERNAL_NET 10.0.0.0/8
(Se establece la dirección de la red externa)

var DNS_SERVERS [10.0.0.2,10.0.0.3,10.0.0.4]
(Lista de los servidores dns de la red interna)

var SMTP_SERVERS [10.0.0.30,10.0.0.31,10.0.0.32,10.0.0.33,10.0.0.34,10.0.0.35,10.0.0.36]
(Lista de los servidores smtp de la red interna)

var RULE_PATH /etc/snort/rules
(Donde estarán localizadas las reglas)

var HTTP_SERVER
[10.0.0.17,10.0.0.25,10.0.0.20,10.0.0.70,10.0.0.17,10.0.0.9,10.0.0.10,10.0.0.11,10.0.0.12,10.0.0.13,10.0.0.22,10.0.0.23,10.0.0.30,10.0.0.31,10.0.0.32,10.0.0.33,10.0.0.34,10.0.0.35,10.0.0.36]
(Lista de los servidores web de la red interna)

var SQL_SERVER [10.0.0.21,10.0.0.46,10.0.0.70,10.0.0.9,10.0.0.10,10.0.0.11,10.0.0.12,10.0.0.13]
(Lista de los servidores sql de la red interna)

var TELNET_SERVER $HOME_NET
(Lista de los servidores telnet de la red interna)

var SNMP_SERVER $HOME_NET
(Lista de los servidores snmp de la red interna)

var HTTP_PORT 80
(Especifica que el puerto 80 será el utilizado para servicios web)

3.3.2 Paso #2. Configurar preprocesadores

En este segundo paso se definen los preprocesadores. Esto es, plugins o partes del programa que definen una manera de hacer sniffing a los paquetes y detectar un mal funcionamiento o un tipo de ataque. A partir de la versión 1.5 aparecieron los preprocesadores que permiten que las funcionalidades de Snort sean extendidas por los usuarios proporcionando un sistema de acceso a los paquetes antes de que sean procesados por el motor de detección de Snort. Se pueden dejar los valores por defecto que vienen aunque en el caso de nuestro trabajo nos centrará en el preprocesador Spade.
Los preprocesadores están sobradamente comentados y explicados antes de cada línea de código. Simplemente se debe saber la manera de declarar cada preprocesador:
preprocessor <nombre del preprocesador>: <opciones de configuración>

Algunos de los preprocesadores que trae consigo Snort son:
preprocessor flow: stats_interval 0 hash 2 preprocessor frag2

preprocessor stream4: disable_evasion_alerts detect_scans detect_state_problems log_flushed_streams state_protection
preprocessor rpc_decode: 111 32771 alert_fragments no_alert_large_fragments no_alert_incomplete

Debe haber quedado claro desde el capítulo anterior que el mecanismo básico de detección de Snort por defecto es el de usos indebidos ó comparación de patrones. Adicionalmente y para aumentar su capacidad de detección, también es posible trabajar el Snort con la aproximación de detección de anomalías, lo que le da un alto grado de desempeño a esta herramienta. La idea primaria será trabajar con ambas aproximaciones o modos de detección como la vía para mejorar su resistencia a falsas positivas y negativas, intentando conseguir que cada alerta reportada pertenezca a un real positivo. El enfoque sería el de tomar los aspectos positivos de cada mecanismo, de forma que se complementen y se obtenga una detección de intrusos mejorada. Es por ello que en este paso se implementa la capacidad de detección basado en anomalías con el empleo del módulo Spade.

3.3.2.1 Módulo Spade

SPADE (Statistical Package Anomaly Detection Engine) es un proyecto creado originalmente por James Hoagland de Silicon Defense y se considera muy importante en el sistema de detección de intrusos que se propone.

Este módulo permite la detección de patrones de tráfico anómalos. Básicamente implementa un tipo distinto de inspección, basado en tratamiento estadístico (se entrena a una maquina con paquetes normales y con paquetes anómalos, de forma que luego es capaz de reconocer ataques) en vez de basarse en la comparación de patrones. Tiene como ventaja que es capaz de detectar modificaciones de ataques, sin necesidad de añadir nada (como habría que hacer en el IDS convencional), aunque es mas difícil de poner a punto que uno convencional. Para activar esta opción es recomendable leer toda la documentación adjunta con la propia distribución de Snort o en el 'Silicon Defense Website' además de consultar los archivos README y USAGE distribuidos en cada versión del Spade.

Spade revisará los paquetes admitidos por Snort, encontrará los de interés y reportará aquellos que cree anómalos junto con una puntuación de anomalías.

La puntuación de anomalías de cada paquete se basa en el comportamiento precedente o histórico de este en la red. Mientras menos veces un tipo particular de paquete haya ocurrido en el pasado, más alto será su puntaje. Los paquetes están clasificados por la ocurrencia de los valores del campo del paquete. Por ejemplo, los paquetes con destino IP 10.10.10.10 y puerto 8080 podrían ser un tipo particular de paquete.

Para hacer esto, las tablas de probabilidad reflejan las ocurrencias de diferentes tipos de paquetes en la historia, dándole un peso superior a los más recientes. Conoceríamos, por ejemplo, que la probabilidad de que un paquete se dirija a la dirección IP 10.10.10.10, al puerto 8080, es de un 10%.... mientras que la probabilidad de que un paquete se dirija a esa misma dirección IP al puerto 8079 es de un 0,1 %. La puntuación absoluta de anomalías se calcula directamente de la probabilidad. Para un paquete X, el grado de anomalía se mide según la siguiente fórmula: A(X) = -Log2 (P(X))

Así la puntuación de anomalías para el paquete 10.10.10.10, 8080 es 3.32 (no muy anómalo) y el puntaje para el paquete 10.10.10.10, 8079 es 9.97 (considerablemente anómalo). La puntuación absoluta de anomalías es el reporte que Spade tradicionalmente ofrece. Desde Spade v021007.1, se introdujo una puntuación relativa de anomalías, lo cual es valor absoluto de cada paquete dividido por el valor absoluto máximo posible a alcanzar por la puntuación. Esto produce un resultado que está siempre entre 0 y 1, de este modo es más fácil interpretar el grado de anomalía para los paquetes.

En cualquier tiempo dado, un umbral de reporte está definido por el sensor. Para cada acontecimiento que exceda este umbral, una alerta es enviada. Esta es enviada al mismo lugar(es) que las reglas de Snort dirigen sus salidas (fichero, syslog, base de datos).

Además, para dar a conocer los acontecimientos anómalos, Spade también puede ser configurado para generar reportes sobre la red en que corre. Por ejemplo, puede decir la cantidad de entropía en tus puertos de destino o puede producir reportes periódicos del número de paquetes vistos y las estadísticas de orden como el puntaje medio de anomalías producido.

Spade no puede decir si un paquete reportado en particular es malo u hostil. Meramente sabe que ciertos paquetes son relativamente inusuales y tiene una idea de qué tan inusuales son. Se deben esperar alertas acerca de actividad benigna (falsos positivos).

Tampoco puede reportar intentos de explotar las vulnerabilidades CGI en un servidor web. Esto dependería de mirar el contenido de los paquetes y Spade solo observa ciertas partes de la cabecera.

Spade no agrupará tampoco eventos anómalos reportados. Ese es el trabajo del correlacionador Spice y en todo caso es mejor explorar otras opciones como la herramienta SnortSnarf que incluye una sección especial para la detección de anomalías.

3.3.2.2 Salidas de Spade

Spade produce dos tipos de mensajes, los cuales, en dependencia de cómo el módulo es configurado, son enviados a las alertas configuradas de Snort o a las facilidades de los logs (por ejemplo, archivos de alertas, base de datos, etc).

La más común y principal reporta la actividad sospechosa encontrada y la puntuación de anomalía del paquete, o sea, lo que aparece en el *log* es cada paquete considerado anómalo y un número entre 0 y 1 significativo de cuán anómalo es. Esta tiene la forma:

Spade: <descripción de la actividad>: <alcance>: <puntuación de la anomalía >

Donde < descripción de la actividad > describe lo que esta reportando Spade, < alcance > explica el tipo de paquete que estaba siendo examinado, y < puntuación de anomalía > es la puntuación de anomalía absoluta o relativa que Spade ha evaluado para el paquete.

Spade también puede producir periódicamente mensajes de la forma:

Spade: id=<id>: Umbral ajustado para T después X alertas (de N).

Esto indica que el umbral de alertas para el detector <id> cambió a T. Esto ocurre cuando se usa uno de los mecanismos para la adaptación de umbrales (remitirse el archivo Usage del Spade). El mensaje también da información acerca del número de alertas (X) expedidas desde el último tiempo en que el umbral fue ajustado y el número total de paquetes (N) aceptados por Spade durante ese tiempo.

3.3.2.3 Detectores de Spade

Un detector de Spade es un componente que busca un tipo particular de anomalía en cierto grupo de paquetes. Se puede configurar Spade para que utilice cualquier número de detectores simultáneamente.

Estos pueden ser de tipos diferentes (en los cuales la detección también será diferente) o del mismo tipo pero aplicados a diferentes grupos de paquetes. Hay cinco tipos de detectores:

1. *Close-dport:* Éste es el tipo tradicional de detector de Spade. Busca paquetes dirigidos a puertos cerrados, o al menos a puertos no usados frecuentemente. Esto puede usarse para encontrar portscans (exploraciones de puertos) porque el tráfico legítimo tiene tendencia a ir a los puertos abiertos.
2. *Dead-dest*: Este tipo de detector busca paquetes que van a una dirección IP que no está en uso. Esto puede usarse para encontrar portscans ya que el tráfico legítimo tiende a ir a las direcciones IP activas.
3. *Odd-dport*: Es un detector que busca direcciones fuentes que usan puertos de destino inusuales. Esto podría indicar un host comprometido o un host mal usado ya que los puertos de destino difícilmente corresponden a aplicaciones y los hosts se inclinan a ser habituales en el uso de aplicaciones.
4. *Odd-port-dest*: Este tipo de detector busca fuentes que hacen una conexión a un destino inusual, relativo a lo que es normal para el puerto de destino. Esto es sólo aplicado en los casos donde el destino es medianamente previsible (por ejemplo, para los puertos 110 (POP3) y 53 (DNS) en computadoras cliente), y podría indicar un host comprometido.
5. *Odd-typecode*: Este tipo de detector busca paquetes con valores inusuales de ICMP en cuanto a tipo y código. Esto puede ser interesante para estar al corriente aun cuando no sean sospechosos.

En dependencia del tipo de detector, a que paquetes es aplicado, y la configuración del usuario, Spade podría implementar un seguimiento de paquetes (follow-up packet) durante pequeña cantidad de tiempo antes de enviar una alerta. Hace esto con el interés de reducir positivos falsos. La realización de este seguimiento podría descontar el hecho de que un paquete es anómalo o podría sugerir el hecho de que lo es realmente. Se podría mirar el seguimiento de paquetes como otro filtro aplicado a la corriente de paquetes.

La eficiencia dependerá de muchos factores incluyendo la configuración (especialmente el número de detectores habilitados) y variará en dependencia de la red. Se han visto desempeños en el orden de los 15 microsegundos por paquete con la configuración de detectores distribuida en Pentium III a 800 MHz. El uso de memoria también variará pero estará probablemente en el rango de los 10 MB. El detector odd-port-dest es conocido por ser lento y usar gran cantidad de memoria, por tanto, si se ve mas tráfico en la

red, especialmente diferentes tipos de tráfico, se puede esperar que el uso de memoria aumente proporcionalmente y también el uso de CPU lo hará un poco (por paquete).

Para instalar correctamente el Spade dentro de Snort se deben seguir los pasos descritos en el archivo Installation, distribuido en cada versión del Spade. Para ello se debe consultar el archivo USAGE donde se explican las numerosas opciones de configuración del módulo de detección de anomalías y a las cuales hay que dedicarle algún tiempo. Estas opciones se concentran en la declaración de los umbrales y los conjuntos de paquetes que chequean los detectores. Aunque se salen de la línea principal de este trabajo, se mencionan a continuación para que se tenga una noción de lo que hacen algunas de ellas.

Para habilitar Spade, su línea principal debe aparecer en el archivo de configuración de Snort. Ésta es la forma de la línea:
preprocessor spade: {<optionname> = <value>}

Las opciones disponibles en Spade son: Logfile, statefile, cpfreq, dest, y adjdest.

- Logfile: La opción "logfile" especifica el archivo de logs para Spade. El mismo es actualizado en cada SIGHUP, SIGQUIT, SIGINT y SIGUSR1. Si esta opción es ' - ', entonces la salida estándar (stdout) está siendo usada. ' - ' es el defecto.
- Statefile: El archivo de estado puede ser especificado por la opción "statefile". Si este archivo está presente desde el inicio, entonces el estado inicial de Spade se tomará de éste. (De otra manera arranca con una pizarra limpia.) Periódicamente este archivo se actualiza con el estado actual del Spade.
- Cpfreq: Esta opción ssignifica la frecuencia con que se actualizará el estado del fichero "*spade.rvc*". Si no está especificada asume por defecto el valor 50000.
- Dest y Adjdest: Esas opciones sirven para controlar hacia donde van los mensajes que Spade produce. La opción Dest puede designar las siguientes opciones: "alert", "log", or "both". "Alert", designada por defecto, significa que las alertas irán a las facilidades ya configuradas del Snort alert, "Log" por su parte significa que las alertas irán al Snort log y "Both" que irán a ambos lugares. A menos que se provea la opción "Adjdest" lo antes dicho se aplica a los reportes de anomalías y a los umbrales de los mensajes ajustados. "Adjdest" acepta las 3 opciones descritas anteriormente además de "none", opción que causará que los mensajes no vayan a ningún lugar.

Aunque Spade pudiera generar falsas alarmas, implementar una detección basada en estadísticas conlleva a una serie de beneficios. De esta forma el mayor esfuerzo del administrador se debe concentrar en lograr una configuración tan precisa del detector como le sea permitida, labor muy difícil en la mayoría de los casos. [U-19]

3.3.3 Paso #3. Configurar plugins de salida

Los plugings de salida permiten elegir de entre una gran variedad de formatos de salida. Se deben configurar solo los que se decidan utilizar. La configuración general de los mismos es de la siguiente forma:

output <nombre del plugin>: <opciones de configuración>

Snort trae por defecto plugins ya configurados y que se decidieron mantener de esa forma. Por otra parte configuramos el plugin que habilita el registro de los mensajes de Snort en el servidor MySQL quedando de la forma:

output database: log, mysql, user=snort password=ok dbname=snort host=localhost

3.3.4 Paso #4. Configurar las reglas del entorno

La forma mas fácil de limitar el trafico de las alertas es desactivar reglas que no se aplican en nuestro sistema, esto lo podemos hacer entrando en la configuración de Snort. El directorio */etc/snort/rules/* contiene muchos archivos con la extensión *.rules*, cada uno de ellos contiene las reglas agrupadas por categoría (ver Anexo 2).

También se pueden conseguir las reglas en la página oficial del Snort http://www.snort.org/dl/rules/. Podemos deshabilitar toda una clase de reglas comentándola en el archivo de configuración o podemos deshabilitar reglas individuales si queremos la protección del resto de reglas de la clase. En dependencia del ambiente de la red, las políticas de seguridad y lo que se considere como sospechoso algunas de estas reglas pueden generar falsos positivos o quizás detectar actividad que puede ser considerada como aceptable por tanto si se quiere deshabilitar algún grupo de reglas no aplicables a nuestro ambiente solo se tiene que comentar el include correspondiente. Hay que tener en cuenta que normalmente es mejor deshabilitar una sola regla que toda la clase, a no ser que esta no se aplique en nuestra configuración.

Existen algunas reglas deshabilitadas por defecto como por ejemplo:
web_atacks, backdoors, shellcode, policy, porn, info, icmp-info, virus, chat, multimedia y p2p.

En este trabajo se deshabilitaron además otras reglas que no se consideraron importantes teniendo en cuenta nuestro ambiente. Se pueden observar en la siguiente figura.

```
include $RULE_PATH/local.rules
#include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
#include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/rpc.rules
#include $RULE_PATH/rservices.rules
#include $RULE_PATH/dos.rules
#include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/tftp.rules

include $RULE_PATH/web-cgi.rules
include $RULE_PATH/web-coldfusion.rules
include $RULE_PATH/web-iis.rules
include $RULE_PATH/web-frontpage.rules
include $RULE_PATH/web-misc.rules
include $RULE_PATH/web-client.rules
include $RULE_PATH/web-php.rules
```

```
include $RULE_PATH/sql.rules
include $RULE_PATH/x11.rules
include $RULE_PATH/icmp.rules
include $RULE_PATH/netbios.rules
include $RULE_PATH/misc.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/oracle.rules
include $RULE_PATH/mysql.rules
include $RULE_PATH/snmp.rules

include $RULE_PATH/smtp.rules
include $RULE_PATH/imap.rules
include $RULE_PATH/pop2.rules
include $RULE_PATH/pop3.rules

include $RULE_PATH/nntp.rules
include $RULE_PATH/other-ids.rules
# include $RULE_PATH/web-attacks.rules
# include $RULE_PATH/backdoor.rules
# include $RULE_PATH/shellcode.rules
# include $RULE_PATH/policy.rules
# include $RULE_PATH/porn.rules
# include $RULE_PATH/info.rules
# include $RULE_PATH/icmp-info.rules
# include $RULE_PATH/virus.rules
# include $RULE_PATH/chat.rules
# include $RULE_PATH/multimedia.rules
# include $RULE_PATH/p2p.rules
include $RULE_PATH/experimental.rules
```

Figura 10. Configuración de reglas de Snort

Creemos necesario explicar brevemente como está conformada una regla, los tipos de regla que posee Snort y los diferentes modos de alertas que soporta el mismo para un mejor entendimiento de lo expuesto anteriormente.

3.4 Formato de Reglas

El lenguaje usado por Snort es flexible y potente, basado en una serie de normas que serán las que nos sirvan de guía para la escritura de las reglas.

Antes de nada, hay que indicar que las reglas de Snort deben estar en una única línea. Esto quiere decir que si la regla ocupa más de una línea, el carácter de fin de línea debe ser escapado (\):

Dentro de estas normas tenemos:

- la descripción de cada regla
- cabecera
- opciones
- uso de preprocesadores

Las reglas de Snort las podemos dividir en dos secciones lógicas, a saber: cabecera de la regla y opciones:

- La cabecera contiene la acción de la regla en sí, protocolo, IPs, máscaras de red, puertos origen y destino y destino del paquete o dirección de la operación.
- La sección opciones contiene los mensajes y la información necesaria para la decisión a tomar por parte de la alerta en forma de opciones. Estas opciones no son obligatorias, pero permiten mayor granularidad en la inspección. Todas las condiciones que aparezcan en las opciones se consideran unidas por un *and*, de forma que se tienen que cumplir todas para que se aplique la regla.

CABECERA

- *Acción*
- *Protocolos involucrados*
- *Direcciones IP*
- *Números de puerto*
- *Dirección de la operación*

OPCIONES

- *Mensaje*
- *Opciones de decisión*

3.4.1 Cabeceras de las reglas

Las cabeceras de las reglas contienen la información acerca de lo que se debe hacer en caso de que la regla se cumpla (alarma, log...):

- Alert: genera una advertencia usando el método de alerta seleccionado al arrancar el Snort.
- Log: genera un reporte.
- Pass: ignora este paquete, es la acción predeterminada en paquetes que no coinciden con la regla.
- Activate: alerta e inicia una acción.
- Dynamic: Recuerda el proceso de activación y genera un log de proceso.
- También se puede crear tipos de reglas.

La cabecera contiene además la información sobre orígenes, destinos y protocolos a los que se aplica la regla:

- Protocolos: El snort cuenta con cuatro analizadores de protocolo, que son tcp, udp, icmp e ip.
- Direcciones IP: Se especifica la dirección IP y la máscara, tanto para el origen como para el destino del paquete.
- Número de puertos: Se puede especificar un rango de puertos, o todos los puertos o un puerto estático.
- Operador de dirección: Este operador indica la dirección del tráfico para así aplicar las reglas, el operador "->" especifica que al lado izquierdo es considerado como trafico entrante y el lado derecho especifica el destino, también se puede especificar bidireccional por medio del operador "<>".

3.4.2 Opciones de las reglas

Las opciones de las reglas permiten conseguir un amplio abanico de efectos sobre la funcionalidad del IDS, desde la modificación a la acción a tomar, hasta la especificación concreta de lo que debe contener un paquete que cumpla la cabecera. Las opciones se separan con el carácter ";" y los argumentos con el carácter ":".

Existen un conjunto de palabras claves permitidas para cada opción:
Msg: imprime un mensaje en los log.

Logto: Especifica un archivo para la salida estándar.
Nocase: insensibilidad ante la opción a anterior.
Content: Busca el patrón entre los paquetes leídos.
Ttl: Compara el tiempo de vida en el encabezado IP.
Id: Compara el Id especificado en el encabezado IP.
Ipoption: Verifica el campo de IP en búsqueda de un código especifico.
Fragbits: Verifica en el encabezado IP algunos bit reservados.
Content-list: Busca en un conjunto de patrones comparándolos con el paquete leído.
Offset: Modifica la opción "content" dándole una posición diferente de inicio para buscar. Muy utilizados para reglas de detección de CGI.
Session: Captura toda la información de una sesión de TCP. [U-6]

3.5 Diferencias entre logs y alertas

Alerta: Permite conocer que cosas interesantes están pasando, además permite hacer dos cosas, escribir un evento basado en alguna facilidad como email o mensajes SMS que no se recomienda por los timeouts o retardos introducidos y además permite escribir en log en un archivo llamado "alert" de lo que ocurrió.

Log: Permite únicamente hacer pistas de los paquetes sin generar alertas. Generalmente se crean en un directorio con el IP del remitente de los paquetes y dentro de este el log del puerto en un archivo.

3.6 Modos de Alerta

Cuando se detecta un paquete malicioso (o sospechoso), se puede invocar una alerta. Hay varias maneras de configurarlas y también el modo en que se almacenarán en el archivo alert.ids.
Snort dispone de cuatro modos de alertas en la línea de órdenes: completo, rápido, consola y ninguno.

- Fast: El modo Alerta Rápida nos devolverá información sobre: tiempo, mensaje de la alerta, clasificación, prioridad de la alerta, IP y puerto de origen y destino.

```
Ids: ~# snort -A fast -dev -l ./log -h 192.168.4.0/24 -c ../etc/snort.conf

09/19-19:06:37.421286 [**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2] ...
... {TCP} 192.168.4.3:1382 -> 192.168.4.15:8080
```

- Full: El modo de Alerta Completa nos devolverá información sobre: tiempo, mensaje de la alerta, clasificación, prioridad de la alerta, IP y puerto de origen/destino e información completa de las cabeceras de los paquetes registrados.

```
Ids:~# snort -A full -dev -l ./log -h 192.168.4.0/24 -c ../etc/snort.conf

[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
09/19-14:53:38.481065 192.168.4.3:3159 -> 192.168.4.15:8080
TCP TTL:128 TOS:0x0 ID:39918 IpLen:20 DgmLen:48 DF
******S* Seq: 0xE87CBBAD Ack: 0x0 Win: 0x4000 TcpLen: 28
TCP Options (4) => MSS: 1456 NOP NOP SackOK
```

Información de la cabecera del paquete:

```
TCP TTL:128 TOS:0x0 ID:39918 IpLen:20 DgmLen:48 DF
******S* Seq: 0xE87CBBAD Ack: 0x0 Win: 0x4000 TcpLen: 28
TCP Options (4) => MSS: 1456 NOP NOP SackOK
```

- Console: Imprime las alarmas en pantalla.
  Ids: ~# snort -A console -dev -l ./log -h 192.168.4.0/24 -c .../etc/snort.conf

- None: Desactiva las alarmas.

```
lds:~# snort -A none -c snort.conf
```

También podemos utilizar las opciones, Syslog, SMB y opciones de salida de bases de datos. Estas opciones no utilizan el modificador *-A* desde la línea de comandos sino módulos de salida independientes que ofrecen una variedad más amplia de opciones de salida. Estas deben configurarse en el tiempo de compilación con modificadores añadidos a la declaración de configuración.

Syslog: Envía alertas al servidor Syslog de Unix. Este es un servicio, que captura y guardar archivos de registro, lo que nos ayudara a consolidar registros de red en un lugar único. Este servicio dificulta a un intruso borrar los rastros de la intrusión. Podemos especificar los formatos Syslog dentro del archivo de configuración y enviar ahí las alertas incluyendo el modificador *-s*.

SMB: permite a Snort hacer llamadas al cliente de SMB (cliente de Samba, en Linux), y mandar mensajes de alerta a maquinas Windows (WinPopUp). Para esta opción hay que descargar las fuentes y compilarlas con la opción enable smbalerts.
Para ejecutar Snort con esta configuración:

```
#snort -c /etc/snort/snort.conf -M workstations
```

Donde workstations es el nombre del host Windows al que se envían las alertas.

Para optimizar la velocidad de Snort, por ejemplo en una LAN de 100Mbps como es el caso de la Universidad de las Ciencias Informáticas, se recomienda usar la opción de alertas rápidas (*-A fast*) o *syslog* (opción -s), y guardar el archivo de *log* en formato binario/tcpdump (opción –b).

Con el sensor y sus patrones correctamente configurados ya estamos listos para poner en funcionamiento nuestro sistema de detección de intrusos. Hasta ahora no se han tenido muchos problemas con el IDS; no obstante, a partir de este momento las cosas se empiezan a complicar un poco, ya que comienza la segunda parte, la del tratamiento de la información que nuestro sensor nos va a proporcionar. Y es que desde este momento el sistema de detección va a empezar a funcionar y a generar logs con notificaciones de posibles ataques, o cuando menos de actividades sospechosas; es hora de decidir cosas como qué hacer ante la generación de un evento en el mismo, cómo procesar la información recibida, o simplemente cuándo rotar los logs generados. El último de los problemas planteados realmente tiene fácil solución;

¿cuándo rotar los logs que Snort genera? La respuesta es muy sencilla: depende. Depende de la cantidad de informes generados en nuestro sensor, depende de la frecuencia con la que debamos realizar informes de los ataque sufridos, depende de la implementación elegida para ejecutar respuestas automáticas ante un ataque (si las ejecutamos), etc. En definitiva, la rotación correcta de unos logs es algo que se debe estudiar y planificar para cada entorno concreto, no se puede dar un período estricto que se aplique siempre porque será sin duda erróneo. No obstante, una idea que nos puede ayudar en la toma de esta decisión es la siguiente: rotaremos los logs cuando los hayamos procesado y extraído de ellos la información que nos pueda interesar para proteger nuestro entorno. Snort genera logs en el directorio /var/log/snort/ si no le indicamos lo contrario (podemos hacerlo con la opción `-l' del programa). En ese directorio encontraremos un fichero denominado alert con las actividades que se vayan registrando. Como nosotros lo que buscamos es básicamente la generación de alarmas, independiente del packet logging, no necesitamos generar estos directorios (aunque nada nos impide hacerlo). Independientemente de la rotación de logs que llevemos a cabo en cada sensor, suele resultar interesante centralizar todos los logs generados en un solo sistema (a veces se le denomina maestro o master), aunque solo sea para realizar estadísticas, seguimientos de máquinas atacantes y atacadas, o simplemente un `top ten' de piratas. [U-8]

3.7 Recomendaciones útiles

Finalmente, se exponen algunas recomendaciones útiles a la hora de instalar y administrar Snort en una red:

1. *Leerse las reglas:* Para optimizar el funcionamiento, es importante leerse todas (sí todas) las reglas que incorpore la distribución de Snort. Conociendo las reglas, se pueden eliminar las que no sirvan, y minimizar el riesgo de sobrecarga en la maquina. Además, es importante saber ante que cosas se esta protegido, y ante que no. Las reglas que teóricamente no deberían producir alarmas (por tener un Firewall o algo similar) pueden servir para comprobar la integridad de los demás sistemas de seguridad. Este hecho se debería mostrar en el mensaje de alarma, y deberían dejarse habilitadas (salvo en situaciones de alta carga).
2. *Documentación asociada:* Aunque la documentación de Snort pueda ser muy completa, puede llegar a ser abrumadora. Es importante tener apuntado en algún tipo de documentación fiable las modificaciones realizadas sobre la distribución (reglas añadidas y reglas borradas), o de lo contrario,

con el tiempo, se perderá la perspectiva de lo que se tiene en el IDS, convirtiéndose en una *caja negra* que no se sabe muy bien lo que hace. Además, si alguna vez se instala un servicio nuevo, hay que acudir inmediatamente a esta documentación para comprobar si se ha eliminado alguna regla que podría ayudar a defender dicho servicio.

3. *Cuidado con lo que se añade:* Si el usuario tiene que añadir alguna regla que no viene con la distribución básica (lo cual es extraño) debe estar seguro de lo que hace, consultando si fuese necesario algo de documentación complementaria. Una regla mal construida puede provocar bucles infinitos, dar lugar a falsos positivos o incluso dejar pasar verdaderos ataques.
4. *Un IDS es algo dinámico:* Cada día se descubren nuevos patrones de ataque. Dado que Snort se basa en un análisis de *perfect match* (aunque posee la herramienta SPADE) es importante que los patrones de comparación estén actualizados. La pagina de Snort publica constantemente actualizaciones y reglas que se pueden añadir al Snort (snort/contrib/snortpp) para que su nivel de seguridad no descienda. También se publican nuevos parches, módulos, preprocesadores y aplicaciones externas que hacen de Snort algo más potente y flexible. Lo importante es que el administrador identifique lo que es útil y lo que no.
5. *La complejidad debe ser limitada:* Existen multitud de aplicaciones externas para trabajar con Snort. Estos sistemas pueden proporcionar grandes mejoras, pero también pueden convertirse en fuentes potenciales de problemas. Si se instala alguno de estos programas, que sea por una mejora significativa, y sobre todo, es importante conocer los riesgos y hacer lo posible por minimizarlos. Los sistemas más simples suelen ser los más robustos.
6. *Las alertas están para leerlas:* En el entorno actual de Internet hay gran cantidad de herramientas automatizadas de ataque/reconocimiento. Estas herramientas se dedican a rastrear Internet en busca de maquinas débiles, lo que puede provocar muchas alarmas diarias (sobre todo escaneos) dependiendo de las condiciones de seguridad del perímetro exterior de la organización (Firewalls). Aunque la mayoría de alarmas sean desestimadas por tratarse de *escaneos rutinarios* es importante mantener una continua vigilancia de estas alarmas. El continuo bombardeo de inofensivos escaneos pueden llevar al cansancio del administrador, que cada vez prestará menos atención a estos informes, aumentando el riesgo de pasar por alto un ataque real. El administrador no debe caer en el hastío, o de lo contrario el IDS será inútil (hay que recordar que el IDS no protege, solo informa).

7. *El Plan de contramedidas:* Cuando el IDS detecte un ataque, hay que estar preparado para actuar. Ahí acaba el trabajo del IDS y empieza el del administrador. No sirve de nada saber que una maquina (o maquinas) han sido atacadas, si luego no se sabe como actuar. No es el objetivo de esta tesis, pero existe mucha documentación (más o menos amplia) sobre como preparar un plan de contingencia/contramedida en función de las necesidades de la organización. [B-8]

3.8 Comentarios finales

Luego de analizar todos los elementos del sistema de detección de intrusos que se propone en el presente trabajo para la red de la Universidad de las Ciencias Informáticas, se concluye que dicho sistema cubre la mayoría de las necesidades de un administrador de IDS. El sensor Snort usado de conjunto con sus herramientas complementarias, es una muy buena opción a implementar con vistas a proteger una red. Con esta solución se logra un alto nivel de gestión de la seguridad, actividad esta llevada a cabo en buena extensión como ha quedado demostrado, y de forma gratuita.

Todos los trabajos de instalación, configuración, puesta a punto del sensor Snort y las herramientas complementarias, que han sido indicados en los epígrafes precedentes de este documento, fueron realizados por los autores obteniendo los resultados esperados.

CONCLUSIONES

Luego del estudio realizado por los autores de la presente tesis se pudo llegar a las siguientes conclusiones:

- ✓ El IDS Snort fue el producto que mejores características y capacidades mostró de acuerdo a las necesidades e intereses de la red de la Universidad de las Ciencias Informáticas siendo su elección el sustento de este trabajo.
- ✓ Se cumplió a cabalidad el objetivo general propuesto ya que se logró la correcta instalación y configuración del IDS Snort en la red-UCI.
- ✓ Quedó validada la hipótesis propuesta anteriormente pues con la implantación del Snort se elevó el nivel de seguridad de la red al informar y auditar de manera eficiente las intrusiones producidas, ofreciéndose además facilidades de monitoreo y control.
- ✓ La utilización de un conjunto de herramientas complementarias al sensor Snort se fundamenta en la adición de funciones que no posee, tales como: registro de alertas en una base de datos, notificación de incidentes, su visualización, entre otras que son muy útiles en la integración y gestión del sistema que se desea lograr. La implantación eficiente de un IDS que opere en toda su profundidad, es una tarea compleja y propia para cada red, debido al gran número de consideraciones y ajustes necesarios. El administrador de seguridad se deberá documentar sobre el sistema propuesto en su conjunto para sacarle el máximo de provecho y configurarlo a sus requerimientos individuales.

RECOMENDACIONES FINALES

En el último capítulo de este trabajo se expusieron algunas recomendaciones a tener en cuenta a la hora de instalar y configurar el Snort, por lo que a continuación se hará mención de otras igual de importantes aunque de un corte más general.

1. La recomendación más importante que puede hacerse, tendiendo en cuenta el poco conocimiento que se tiene del tema, es exhortar al estudio profundo y empleo consciente e inteligente de los sistemas de detección de intrusos.
2. Para hacer más efectivo el trabajo con el Snort y aprovechar al máximo las posibilidades y beneficios que brinda, se recomienda montar el sistema en más de un punto estratégico dentro de la red de la universidad.
3. Se recomienda tener presente que más vale una pérdida temporal de usuarios producto de bloqueos no legítimos, que ser víctima de ataques reales cuya efectividad puede representar daños mayores. Por tanto, será preferible en momentos de duda, decidir por el bloqueo de una dirección sospechosa.

BIBLIOGRAFÍA

[B-1] Antonio Villalón Huerta. Sistemas de Detección de Intrusos. Libro Seguridad en Sistemas Unix. Capítulo 18.Julio, 2002. http://www.rediris.es/cert/doc/unixsec/unixsec.pdf

[B-2] Rafeeq Ur Rehman. Intrusion Detection Systems with Snort. Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID. Prentice Hall PTR Upper Saddle River, New Jersey 07458. www.phptr.com/rehman/

[B-3] Ant Allan. Intrusion Detection Systems: Perspective. Technology Overview. Enero, 2002. http://cnscenter.future.co.kr/resource/rsc-center/gartner/95367.pdf

[B-4] Curtis A. Carver, John M.D. Hill, Member, IEEE and Udo W. Pooch, Senior Member IEEE. Limiting Uncertainty in Intrusion Response. Proceedings of the 2001 IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NY, 5-6 June, 2001 http://www.itoc.usma.edu/Workshop/2001/Authors/Submitted_Abstracts/paperT3A1(56).pdf

[B-5] Fred B. Cohen. Simulating Cyber Attacks, Defenses, and Consequences. May 13, 1999. http://all.net/journal/ntb/simulate/simulate.html

[B-6] Krzysztof Zaraska. IDS Active Response Mechanisms: Countermeasure Subsytem for Prelude IDS. July 9, 2002. http://www.prelude-ids.org/download/misc/lsm/2002/slides/krzysztof/lsm.pdf.

[B-7] Luis Miguel Díaz Vizcaíno. Estudio Tecnológico. Sistemas de Detección de Intrusos. Universidad Carlos III de Madrid. http://www.it.uc3m.es/lmiguel/ids2.pdf

[B-8] Steven J. Scott. Threat Management Systems. The State of Intrusion Detection. August 9, 2002. http://www.superhac.com/docs/threatmanagement.pdf

[B-9] Steven J. Scott. Snort Installation Manual Snort, MySQL and ACID on Redhat 7.3. August, 2002 Version 1.5. http://home.earthlink.net/~sjscott007/

[B-10] The Best: Products. December 2003. http://infosecuritymag.techtarget.com/ss/0,295796,sid6_iss288_art517,00.html

REFERENCIAS BIBLIOGRÁFICAS

Fuente electrónica

[U-1] http://www.uv.es/~montanan/redes/trabajos/IDSs.doc

[U-2] http://escert.upc.edu/index.php/web/es/publicacion,178,3.html

[U-3] http://www.symantec.com/region/mx/smallbusiness/articles/LAM_intrusion.html

[U-4] http://out.uclv.edu.cu/fie/hiribarne/TesisMaestriaRtorresV5.0.doc]

[U-5] http://www.maestrosdelweb.com/editorial/snort/

[U-6] http://www.wikilearning.com/creacion_de_reglas_con_snort-wkccp-4735-14.htm

[U-7] http://www.criptored.upm.es/guiateoria/gt_m189d.htm

[U-8] http://mirrors.csol.org/LuCAS/Manuales-LuCAS/SEGUNIX/unixsec-2.1.pdf

[U-9] http://es.wikipedia.org/wiki/PHP

[U-10] http://es.wikipedia.org/wiki/MySQL

[U-11] http://es.wikipedia.org/wiki/Acidbase

[U-12] http://72.14.205.104/search?q=cache:fiIrSsz9AqAJ:eminds.f2g.net/seguridad/sistemas-de-deteccion-de-intrusos.txt

[U-13] http://www.arcanus.com.uy/boletin/boletin%202.htm

[U-14] http://www.ldc.usb.ve/~miguel/portknocking_and_snort/node21.html

[U-15] http://www.gnutecnologias.com/productos/snort.html

[U-16] http://rrii.sgp.gov.ar/index.php?option=com_docman&task=doc_view&gid=7714&Itemid=46

[U-17] http://is.ls.fi.upm.es/doctorado/Trabajos20052006/Campo_TrabajoCompleto.pdf

[U-18] http://es.wikipedia.org/wiki/Apache

[U-19] Readme and Usage de Spade

Tesis

[T-1] Pablo Jay Maceo. Sistema de seguridad en la red-UCI con la utilización del IDS-Snort. Ciudad de La Habana. 2005

[T-2] Ernesto Guillen Soriano. Análisis e Implantación del Sistema de Detección de Intrusos en Red Snort. Desarrollo de la Detección de Anomalías y respuesta automática. Cuidad de La Habana. 2004

GLOSARIO DE TÉRMINOS

Antivirus

Programa cuya finalidad es prevenir y evitar la infección de virus, impidiendo también su propagación. Tiene capacidad para detectar y eliminar los virus y restaurar los archivos afectados por su infección

Puerta trasera (Backdoors)

Se trata de una herramienta de administración remota. Cuando se instala tiene la capacidad de dar a un intruso privilegios como administrador. Puede buscar palabras claves o datos confidenciales y enviarlos por correo electrónico a un área remota.

Caballo de Troya/Troyano

Programa informático que lleva en su interior la lógica necesaria para que el creador del programa pueda acceder al interior del sistema que lo procesa.

CGI

Programas consistentes de una serie de instrucciones escritas en un lenguaje de programación como C o Perl que procesan la petición de un navegador, ejecutan funciones y formatean los resultados a HTML de manera que puedan ser presentados en un navegador Web.

Cron

Demonio que usa el administrador de sistemas UNIX para delegar ciertas tareas que pueden ser ejecutadas sin su participación. Este puede ser programado para ejecutar las tareas a intervalos variables.

DMZ

Subred situada entre una red interna de confianza (como una red local privada) y una red externa no confiable (como Internet). Normalmente en esta zona se sitúan los dispositivos accesibles desde Internet, como servidores Web, FTP, SMTP ó DNS, evitando la necesidad de acceso desde el exterior a la red privada.

Falsa negativa

Evento que se da como inexistente cuando realmente si existe, por ejemplo, decir que un sistema no está siendo atacado cuando en realidad lo está.

Falsa positiva

Evento que se da como existente cuando realmente no existe, por ejemplo, decir que un sistema está siendo atacado cuando en realidad no lo está.

Cortafuegos (Firewall)

Sistema basado en *hardware* o programa que filtra el tráfico de red basado en un conjunto de reglas. Un cortafuego simple normalmente bloquea accesos a puertos específicos.

Firmas de ataques

Descripción del código empleado por el intruso para atacar los sistemas

GNU

La licencia GNU permite copiar de los programas su código, modificarlos ó redistribuirlos libremente.

Gusano

Programa maligno que se infiltra en una computadora y se replica por si mismo en dicho sistema o en otro.

Intranet

Red privada, desarrollada dentro de una organización que utiliza el mismo *software* y provee de información similar que Internet, solo que es únicamente para el uso interno.

Intrusión/Ataque

Conjunto de acciones que intentan comprometer la integridad, confidencialidad o disponibilidad de un recurso informático.

Intruso /Hacker

Una persona que goza alcanzando un conocimiento profundo sobre el funcionamiento interno de un sistema, de una computadora o de una red de computadoras. Este término se suele utilizar indebidamente como peyorativo, cuando en este último sentido sería más correcto utilizar el término cracker.

LAN

Red de datos para dar servicio a un área geográfica máxima de unos pocos kilómetros cuadrados, por lo que pueden optimizarse los protocolos de señal de la red para llegar a velocidades de transmisión de hasta Giga bits/s.

Log

Archivo que registra movimientos y actividades de un determinado programa. Por ejemplo, en un servidor Web, se encarga de guardar en sus *log* todos las solicitudes y servicios ofrecidos.

Plugin

Palabra que se emplea genéricamente para denotar un módulo o bloque de funciones adicionales, que puedan ser incorporadas a cualquier programa.

Proxy

Servidor que está conectado normalmente al servidor de acceso a la WWW de un proveedor de acceso, va almacenando toda la información que los usuarios reciben de la WEB. Esta encargado, entre otras cosas, de centralizar el tráfico entre Internet y una red privada.

RPM

Potente administrador de paquetes disponible para UNIX que puede ser usado para construir, instalar, solicitar, verificar, actualizar y borrar paquetes de programas individuales.

Script

Programa o secuencia de instrucciones que son interpretadas o manejadas por otros programas en lugar del procesador central como código de máquina.

Sniffer (husmeador)

Herramienta que captura paquetes que atraviesan la red. Normalmente se usa con fines ilegales. Usualmente se configuran para capturar y mostrar determinado tipo de información.

Spoofing

Tipo de ataque en el que un sistema o usuario asume la identidad de otro de manera ilegal.

SSH

Es un programa para el registro y ejecución de comandos en una computadora remota de forma segura mediante el uso de una comunicación encriptada.

SSL

Protocolo creado por Netscape con el fin de posibilitar la transmisión cifrada y segura de información a través de la red.

TCP/IP

Sistema de protocolos, definidos en la RFC 793, en los que se basa buena parte de la comunicación de Internet. TCP/IP es el estándar de protocolo de comunicaciones requerido por las computadoras que acceden a Internet.

URL

Sistema unificado de identificación de recursos en la red. Las direcciones se componen de protocolo y dirección local del documento dentro del servidor. Este tipo de direcciones permite identificar objetos WWW, Gopher, FTP, News y otros.

Unix

Es un sistema operativo de tiempo compartido, controla los recursos de una computadora y los asigna entre los usuarios. Permite a los usuarios correr sus programas. Controla los dispositivos de periféricos conectados a la máquina.

Virus

Es un programa de computadora que tiene la capacidad de causar daño y su característica más relevante es que puede replicarse a sí mismo y propagarse a otras computadoras. Infecta cualquier archivo o sector de las unidades de almacenamiento que contenga códigos de instrucción que el procesador vaya a ejecutar.

VPN (Virtual Private Network)

Es una tecnología de red que permite una extensión de la red local sobre una red pública o no controlada, como por ejemplo Internet.

XML

El lenguaje de marcas extensible es una forma flexible de crear formatos de información y compartir tanto el formato como los datos a través de las redes.

SIGLARIO

ARP (Address Resolution Protocol)	Protocolo de resolución de direcciones.
CPU (Central Processing Unit)	Unidad Central de Procesamiento.
DMZ (De-Militarized Zone)	Zona desmilitarizada
DNS (Domain Name System)	Sistema de nombres de dominio.
DoS (Denial of Service)	Denegación de servicios
FTP (File Transfer Protocol)	Protocolo de transferencia de ficheros.
HIDS (Host Intrusión Detection System)	Sistemas de detección de intrusos en computadora.
HTTP (Hypertext Transfer Protocol)	Protocolo de transferencia de hipertexto.
ICMP (Internet Control Message Protocol)	Protocolo de mensajes para control de Internet
IDES (Intrusion Detection Expert System)	Sistema experto de detección de intrusos
LAN (Local Area Network)	Redes de área local.
NIDS (Network Intrusión Detection System)	Sistemas de detección de intrusos en red.
NCSC (Nacional Computer Security Center).	Centro Nacional de Seguridad de Computadoras
POP3 (Post Office Protocol)	Protocolo de oficina postal versión 3.
PPP (Point to Point Protocol)	Protocolo punto a punto
RIP (Routing Information Protocol)	Protocolo de información de enrutamiento.
SMB (Server Message Block)	Bloque de mensajes de servidor.
SMTP (Simple Mail Transfer Protocol)	Protocolo simple de transferencia de correo.
SQL (Structured Query Language)	Lenguaje estructurado de escrutinio.
SSH (Secure Shell)	Shell seguro
SSL (Secure Sockets Layer)	Nivel de sockets seguro.
TCP/IP (Transmission Control Protocol/Internet Protocol)	Protocolo de control de transferencia / Protocolo de Internet.
VA (Vulnerability Assessment)	Tasador de vulnerabilidades
VPN (Virtual Private Network)	Redes privadas virtuales.
XML (Extensible Markup Language)	Lenguaje de marcas extensible.

ANEXOS

Anexo 1. Productos complementarios a los IDS.

Existen varios productos reconocidos como complementos de los IDS, cada uno se enfoca en un área particular de seguridad y tiene sus puntos fuertes y débiles. Solo la combinación de todos con un IDS -defensa en profundidad- permite la protección de la organización frente a un rango real de ataques a la seguridad

- Los cortafuegos: Un cortafuegos es típicamente uno de los primeros mecanismos de seguridad que las organizaciones emplean para proteger el perímetro de su red. En muchísimos casos es el único mecanismo usado. A pesar de que los cortafuegos ofrecen una buena protección contra intrusiones de fuentes externas como las provenientes de Internet, las organizaciones deben saber que no todos los accesos a la infraestructura empresarial tienen lugar a través del cortafuego, por ejemplo ciertos usuarios pueden establecer una conexión de *modem* no autorizada a la Intranet. Similarmente las organizaciones deben comprender que no todas las intrusiones ocurren desde fuera del cortafuego. Por ejemplo algunos empleados pueden accidentalmente o no, tratar de acceder a ficheros o sistemas no autorizados desde la red local.
- Los productos antivirus: Son una defensa necesaria contra el daño que los virus y otros códigos malignos puedan hacer. Pero la protección ofrecida depende en gran medida del nivel de actualización que tenga la herramienta ya que los virus son cada vez más rápidos, complejos y devastadores.
- Los productos exploradores de vulnerabilidades (VA): También conocidos como *scaners*, chequean en busca de fisuras de seguridad abiertas para el ataque, que puedan estar disponibles en un sistema informático o red. Los productos VA pueden brindarse como parte de un IDS, y son en esencia generadores-detectores de usos indebidos. Estos generan una impresión del estado de seguridad en un momento dado, y posibilita que el administrador audite los sistemas informáticos y redes de acuerdo con una política de seguridad particular. Además los IDS pueden usar los resultados de un VA para precisar sus respuestas, priorizando ataques a las vulnerabilidades existentes, y prestándole menos atención a aquellas que ya hallan sido atendidas de alguna manera.

- Potes de miel: Un pote de miel es un sistema que está expresamente configurado para atraer intrusos e informar de sus acciones. Para configurar un pote de miel la organización puede: instalar un sistema operativo sin sus parches de seguridad y usar las opciones por defecto; asegurarse que no existen datos en el sistema que puedan ser destruidos; y adicionar las aplicaciones que son designadas para detectar las actividades del intruso.

Mantener un pote de miel requiere una considerable atención, a la vez que solo ofrece elementos que servirán para conocer el procedimiento de un ataque y darle seguimiento. Esto significa que la organización puede no capturar al atacante. Un pote de miel es probablemente la más elaborada defensa que una organización debiera considerar como estrategia para lograr su seguridad en profundidad.

Anexo 2. Clases de reglas de Snort

En el siguiente cuadro podemos observar una lista con las clases de reglas más habituales en Snort y una pequeña descripción de las mismas.

Cuadro 13.2: Clases de reglas de Snort (I)

Clase de reglas	Descripción
`attack-response.rules`	Son las alertas para paquetes de respuesta comunes después de que un ataque haya tenido éxito. Raramente se informan como falsos positivos y debemos dejarlas activadas en la mayoría de los casos.
`backdoor.rules`	Estas reglas son signos comunes de una puerta trasera o de un programa troyano en uso. Raramente son falsos positivos.
`bad-traffic.rules`	Estas reglas representan el tráfico de red no estándar que normalmente no debería verse en la mayoría de las redes.
`chat.rules`	Localizan transmisiones estándar para muchos programas conocidos de conversación. Si la conversación se permite implícitamente o explícitamente estas alertas deben estar deshabilitadas. Así mismo, hay que tener en cuenta que estas alertas no son una solución milagrosa para las conversaciones y no detectarán todos los tipos de tráfico de conversaciones.
`ddos.rules`	Busca tipos de ataques de denegación de servicio distribuido estándares. En DMZ y WAN, estas alertas no sirven de mucho porque si se ha producido un ataque de este tipo probablemente lo sepamos en seguida. Sin embargo, pueden ser muy útiles dentro de una LAN para comprobar si tenemos una máquina zombi en otra red participando en un ataque de denegación de servicio (DoS) sin saberlo.
`dns.rules`	Buscan algunos abusos estándar contra servidores DNS. Si no estámos ejecutando un servidor DNS propio, podemos desactivarlas.
`dos.rules`	Similar al conjunto de reglas anterior.
`experimental.rules`	Están deshabilitadas de forma predeterminada. Generalmente se utilizan sólo para probar nuevas reglas hasta que se desplazan a otra categoría.
`exploit.rules`	Estas reglas son para el tráfico de abuso estándar, siempre habilitadas.
`finger.rules`	Estas reglas marcan el tráfico que tiene que ver con los servidores finger. Si no estámos ejecutando ninguno de estos servidores, las podemos deshabilitar. Sin embargo, este tipo de servidores normalmente se ejecutan ocultos al administrador del sistema, por lo que podemos dejarlas habilitadas, ya que no suelen generar falsos positivos.
`ftp.rules`	Igual que las reglas anteriores pero buscan abusos de FTP. Una vez más, podemos dejarlas habilitadas aunque no tengamos servidores FTP ya que nos avisarán de cualquier servidor FTP ilegal en el sistema.
`icmp-info.rules`	Estas reglas registran el uso de los mensajes ICMP que cruzan la red, por ejemplo, los ping. A menudo producen falsos positivos, podemos desactivar toda la clase a no ser que deseemos estar pendientes del tráfico ICMP en nuestra red. Otra clase de tráfico ICMP dañino conocido, *icmp.rules*, captura los escaneados de puertos y similares.
`icmp.rules`	Cubre el tráfico ICMP sospechoso o dañino y es poco propenso a generar falsos positivos. Sin embargo, es posible que se activen en una red ocupada ejecutando muchos servicios de diagnóstico.
`imap.rules`	Reglas correspondientes al uso del protocolo de acceso a mensajes desde internet (IMAP, Internet Message Access Protocol).
`info.rules`	Capturan mensajes de errores diversos: Web, FTP y otros servidores.
`local.rules`	En este archivo podemos añadir nuestras propias firmas o reglas personalizadas para la red, el archivo está vacío de forma predeterminada.
`misc.rules`	Reglas que no encajan en ninguna de las restantes categorías.
`multimedia.rules`	Registra el uso de software de vídeo. Si permitimos las aplicaciones de vídeo o utilizamos video-conferencia, debemos deshabilitar estas reglas.

Cuadro 13.3: Clases de reglas de Snort (II)

Clase de reglas	Descripción
mysql.rules	Vigila el acceso del administrador y otros archivos importantes en una base de datos MySQL. Si no ejecutamos este tipo de base de datos, podemos deshabilitar estas alertas. Así mismo, si nuestra base de datos MySQL está en desarrollo, podrían producirse muchos falsos positivos.
Netbios.rules	Esta clase de reglas nos alerta de diversa actividad NetBIOS en la LAN. Algunas de ellas son exploits evidentes. Sin embargo, otras, como las alertas de sesión NULL, pueden producirse normalmente en una LAN Windows. Tendrémos que jugar con esta sección para deducir cuáles son las reglas apropiadas en nuestra LAN.
nntp.rules	Reglas relacionadas con el servidor de noticias. Si no ejecutamos noticias de red en los servidores, es mejor deshabilitar estas reglas.
oracle.rules	Reglas del servidor de base de datos Oracle. Si no tenemos un servidor de este tipo, las deshabilitamos.
other-ids.rules	Estas reglas se relacionan con exploits en los IDS de otros fabricantes. Es muy probable que no tengamos ningún otro NIDS en la LAN, pero si es así hay que dejarlas habilitadas.
p2p.rules	Reglas que rigen el uso del software para compartir archivos punto a punto. Estas reglas crearán alertas durante el uso normal de este software, si lo permitimos, deberemos deshabilitarlas.
policy.rules	Este archivo contiene diversas alertas relacionadas con toda la actividad permitida en la LAN, como Go-to-my-pc y otros programas. Debemos revisarlas y habilitar sólo las que se aplican en nuestras políticas internas.
pop3.rules	Para servidores de correo, si tenemos un servidor de este tipo, hay que dejarlas habilitadas.
porn.rules	Estas reglas son trampas rudimentarias para la exploración web relacionada con la pornografía. No constituyen ni mucho menos un reemplazo para los buenos sistemas de filtrado de contenido.
rpc.rules	Esta clase controla las alertas de llamadas a procedimientos remotos (RPC). Aunque creamos no ejecutar ninguno de estos servicios, normalmente se activan como parte de otros programas, por lo que es importante tener cuidado con lo sucede en la LAN. RPC puede habilitar la ejecución remota de código y normalmente se utiliza en los troyanos y exploits.
rservices.rules	Registra el uso de diversos programas de servicios remotos, como rlogin y rsh. Estas reglas en general, son de servicios inseguros, pero si tenemos que utilizarlos, pueden examinarse con este conjunto de reglas.
scan.rules	Alertas para utilizar programas de escaneado de puertos. Los escaneados de puertos son una indicación extraordinaria de una actividad ilícita. Si utilizamos escáneres de puertos, podemos desactivar Snort durante su ejecución o deshabilitar esta regla en concreto para la IP donde se encuentra el escáner.
shellcode.rules	Esta clase busca paquetes que contienen código de montaje, comandos de bajo nivel también conocidos como código shell. Estos comandos normalmente forman parte integral de muchos exploits como los desbordamientos de memoria. Capturar al momento un código shell es una buena indicación de que se está produciendo un ataque.
smtp.rules	Contienen las alertas para el uso del servidor de correo en la LAN. Esta sección necesitará algún ajuste ya que muchas actividades de servidor de correo normales activarán reglas de esta sección.
sql.rules	Reglas para diversos programas de base de datos SQL. Si no ejecutamos ninguna base de datos, podemos deshabilitarlas, pero no es mala idea dejarlas por si existen bases de datos SQL ejecutándose sin que lo sepamos.

Cuadro 13.4: Clases de reglas de Snort (III)

Clase de reglas	Descripción
telnet.rules	Registra el uso de telnet sobre la red. Normalmente telnet se utiliza en enrutadores o en otros dispositivos de línea de comandos, por lo que es recomendable realizar el registro incluso si telnet no está en nuestros servidores.
tftp.rules	TFTP (FTP trivial) es un servidor FTP alternativo que se ejecuta normalmente en enrutadores. Puede utilizarse para cargar nuevas configuraciones y por consiguiente es mejor que este conjunto de reglas esté habilitado.
virus.rules	Contiene las firmas de algunos gusanos y virus conocidos. Esta lista no está completa y no se mantiene con regularidad. No es un reemplazo para el software de escaneado de virus pero puede capturar algunos gusanos que se transmitan por la red.
web-attacks.rules	Todas estas clases se refieren a diversos tipos de actividad web sospechosa. Algunas son genéricas, como las clases web-attacks. Otras clases, como web-iis y web-frontpage, son específicas de una determinada plataforma de servidor web. Sin embargo, aunque creamos que no estamos ejecutando un servidor web Microsoft o PHP, es mejor dejarlas habilitadas para descubrir cualquier tipo de esta actividad en nuestra LAN de la que debamos preocuparnos. Tendremos que ajustar estos conjuntos de reglas, especialmente si los servidores web se encuentra en un desarrollo activo.
web-cgi.rules	
web-client.rules	
web-coldfusion.rules	
web-frontpage.rules	
web-iis.rules	
web-php.rules	
X11.rules	Registra el uso del entorno gráfico X11 en su red.

Figura A2. Clases de reglas de Snort

Anexo 3. Snort en modo "in-line"

El Inline Mode se encuentra integrado en la versión de Snort 2.3.0.

Hay 3 tipos de reglas que pueden ser usadas con snort_inline, a la hora de manejar los paquetes, a saber:

- drop: le indica a iptables que ignore el paquete, y el mismo es logueado por Snort con los medios usuales ya descritos.
- reject: le indica a iptables que ignore el paquete, el mismo el logueado por Snort con los medios usuales; y adicionalmente si la conexión es de tipo TCP, iptables envía un TCP reset, mientras que si es UDP, iptables envía un mensaje de tipo ICMP port unreachable.
- sdrop: le indica a iptables que ignore el paquete y nada es *logueado*.

En el modo "in-line" (en línea) el NIDS Snort actúa a nivel 2, como un puente. Se sitúa entre la red que se desea proteger y el resto. Cuenta con una interfaz de red para recibir el tráfico del exterior, y otra para transmitirlo a la red a proteger. Además suele tener otra interfaz para las labores de administración y gestión.

Esta situación le permite el control total sobre el tráfico que pasa por su tramo de red. No sólo puede analizar todo el tráfico que recibe, antes de decidir qué hacer, sino que puede gestionar el ancho de banda. La siguiente figura describe este modo de funcionamiento.

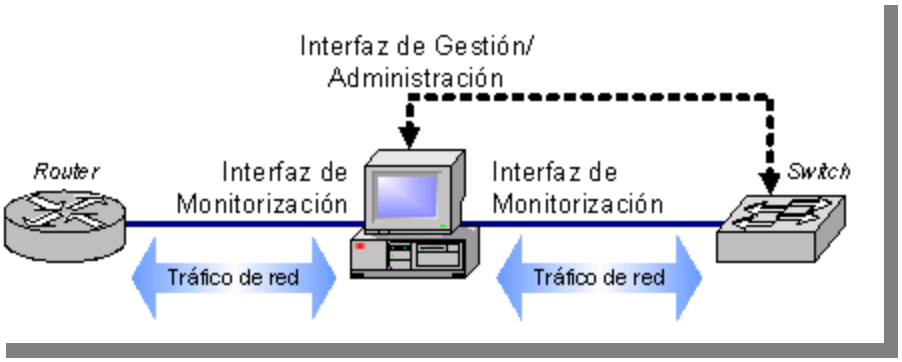

Figura A3. NIDS en modo en línea ("In-line Mode")

Un NIDS instalado en modo "in-line", y con una serie de modificaciones para soportar el tratamiento de tráfico de red, ofrece las posibilidades de un NIDS normal, con la capacidad añadida de un cortafuego.

Anexo 4. IPS: Sistema de prevención de intrusos

Un Sistema de Prevención de Intrusos: Sistema que combina las capacidades de bloqueo de un cortafuegos y las de análisis de un IDS. Está diseñado para detener ataques antes de que tengan éxito. Los IPS son soluciones proactivas, diseñadas para detectar paquetes maliciosos de entre el tráfico normal (algo que, por ejemplo, los firewalls actuales no llevan a cabo), detener intrusiones en curso y bloquear el tráfico maligno automáticamente, esto es, previo a que el ataque ocasione un daño en su destino final.

Los IDS tradicionales lo único que generan son acciones posteriores, que van desde la simple generación de alertas a reconfigurar políticas del firewall, mientras lo único relacionado con la prevención es “detección y descarte del paquete en tiempo real o en línea”, por lo que consideramos que un IPS debe ser, antes que nada, un excelente IDS.

Incluye ventajas como:

- Capacidad de reacción automática ante incidentes - el sistema aplica nuevos filtros conforme detecta ataques en progreso)
- Mínima vigilancia - el sistema no requiere tanta dedicación como un IDS tradicional; esto en consecuencia requeriría menos inversión en recursos para administrar y operar estos sistemas (en comparación con un IDS).

- Menos falsas alarmas

A continuación una tabla con algunos de los productos IPS mas importantes que podemos encontrar en el mercado actualmente

Tipo IPS ***Producto***	**Prevención o Protección** ***HIP***	**Detección Respuesta** ***Detección + Respuesta***
Sistemas operativos bastionados	Pitbull Foundation de Argus Trusted Solaris de Sun eTrust Access Control de CA VirtualVault de HP	
Sistemas de control de acceso	Pitbull LX de Argus Stormwatch de Okena STAT Neutralizer de Harris Serverlock de Watchguard CylantSecure de Cylant Immunix de Wirex ...	
"Escudos" de servidores web	SecureIIS de eEye WebServer Edition de Entercept Applock de Watchguard Wavebraker de Pelican	
Cortafuegos para servidores web	AppShield de Sanctum HIVE de S21sec Interdo de Kavado DMZ/Shield de Ubizen	
IDS de pasarela		RealSecure Guard de ISS Intrushield de Intruvert IPS de Netscreen SmartDefense de Checkpoint Snort en línea Attack Mitigator de TopLayer ....

Figura A4. Productos IPS en el mercado