Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Unidad Zacatenco

Departamento de Computación

Herramientas de monitoreo y detección de intrusos en servidores Linux

Tesis que presenta:

Hilda María Chablé Martínez

para obtener el Grado de:

Maestro en Ciencias

en la Especialidad de

Ingeniería Eléctrica

Director de la Tesis:

Dr. Arturo Díaz Pérez

México, D.F. **Febrero 2007**

A mi mamá **Hilda** y mi papá **Herminio** a quienes amo y de quienes me siento plenamente orgullosa porque me han enseñado a luchar por mis metas pese a cualquier adversidad. Ellos llenan mi vida de esperanza, amor y felicidad.

A **Sebastián** porque ha sido mi compañero y apoyo a lo largo de esta aventura y sé que llegó a mi vida para llenarla de amor, comprensión y sobre todo de felicidad. Te amo.

Agradecimientos

Gracias a mis padres porque me han apoyado incondicionalmente en todas las decisiones que he tomado en mi vida y a ellos dedico todo lo que soy y lo que he logrado. Doy gracias a Dios por tenerlos a mi lado.

Gracias a mis hermanos Lupita y Carlos porque han sido mi ejemplo a seguir desde que era pequeña. De ellos aprendí a ser mejor cada día y a confiar en que puedo alcanzar cualquier meta que me proponga en la vida. Les agradezco todos los buenos momentos que pasamos juntos.

Agradezco a José Sebastián González Altamirano porque fue quien me brindó su ayuda para levantarme las tantas veces que caí y estuvo conmigo a lo largo de mi maestría apoyándome en todo momento y trayendo alegría extra a mi vida.

Gracias a mi amiga Magally Morales por hacer más amena mi estancia en el D.F. con su buen humor y a mi amigo Ismael Hernández por haberme ayudado cuando más lo necesité. Gracias por haber sido mi familia en estos dos años.

Agradezco a mis amigos del CINVESTAV Florecita Radilla, Claudia Hernández, Renato Zacapala, Oscar Irineo, Daniel López, Oscar Alvarado, Héctor Acosta y Julio Moctezuma por todos los momentos de diversión y entretenimiento que pasamos juntos, también porque aprendí mucho de cada uno de ustedes.

Agradezco a las secretarias del departamento de Computación, especialmente a Sofía Reza porque siempre me ayudó a superarme como persona. Además hizo de todos mis trámites un proceso fácil y ameno y me aconsejó cuando lo necesité.

Agradezco a mi asesor el dr. Arturo Díaz Pérez por haberme brindado su confianza y haberme transmitido tantos conocimientos como profesor y como asesor de tesis.

Gracias a los revisores de este trabajo de tesis el dr. Luis Gerardo de la Fraga y el dr. Miguel Ángel León Chávez por haber contribuido a mejorar este documento.

Gracias al CINVESTAV por haberme dado la oportunidad de estudiar en sus instalaciones.

Gracias al Conacyt por haberme brindado una beca que sirvió para mi manutención durante mis estudios de maestría.

Gracias al Conacyt por financiar parcialmente este trabajo a través del proyecto 45306: Estudio, Análisis y Desarrollo de Algoritmos de Muy Alto Desempeño para Arquitecturas Hardware/Software.

Gracias al departamento de Computación por haberme facilitado todos los recursos para realizar mis estudios de maestría.

Gracias a la Biblioteca del departamento de Ingeniería Eléctrica por haberme proporcionado todos los libros que necesité.

Resumen

Debido al acelerado crecimiento de las redes de computadoras, el aspecto de la seguridad informática se ha convertido en un área que demanda mayor atención y causa mayor preocupación a los administradores de redes. Dentro de los mecanismos para afianzar la seguridad de un sistema están los Sistemas de Detección de Intrusos (SDI), los cuales son muy importantes.

En este trabajo de tesis se propone un SDI híbrido que combina las características de un SDI basado en huésped y un SDI basado en red, esto con el fin de explotar las ventajas que se obtienen de combinar ambos enfoques. Por un lado, los SDI basados en red permiten obtener información general y amplia de un ataque, y por otro lado los SDI basados en huésped proporcionan información más específica del rastro del atacante.

El sistema de monitoreo y detección de intrusos implementado cuenta con un módulo que analiza las bitácoras de un sistema Linux. De éstas se obtienen variables que son supervisadas y de las que se pueden obtener patrones de uso normal. Por otro lado se cuenta con un módulo que supervisa el tráfico de entrada y salida muestreado de un segmento de red. Cuando se detecta una actividad maliciosa en algunos de los dos módulos, se correlaciona la información de ambos y si detecta alguna anomalía se envía una alerta al administrador del sistema.

Así se supervisan varios servidores de una red en base a las actividades del sistema operativo y las aplicaciones que en éstos se ejecutan, añadiendo la capacidad de supervisar el tráfico que entra y sale de cada uno de estos servidores, aumentando así la confianza y la exactitud en las detecciones de intrusos.

Con este SDI híbrido se pueden detectar ataques tan específicos como una exploración de contraseñas y nombres de usuario legítimos, correo SPAM, abuso de los recursos del servidor web, hasta ataques más generales como exploraciones de puertos y direcciones IP, ataques de negación de servicio, entre otros.

El sistema fue implementado y evaluado en dos modos: en tiempo real y en modo de simulación. Se hicieron pruebas en ambos modos, aunque en el modo de simulación hubo más flexibilidad de modelar plataformas exhaustivas de pruebas. El sistema final tuvo resultados de desempeño, rapidez y funcionalidad satisfactorios.

Abstract

Due to the accelerated growth of computer networks, the aspect of the computer science security has become an area that demands more attention and cause more preoccupation to network managers. Within the mechanisms to strengthen system security are Intrusion Detection System (IDS), which is very important.

The present thesis work proposes a hybrid IDS that combines the characteristics of a IDS based on host and a IDS based on network, this with the purpose to exploit the advantages obtained from to combine both approaches. First of all, the IDS based on network allow you to obtain general and ample information about an attack, and on the other hand, the IDS based on host provide more specific information about the attacker's trace.

The monitoring and intruders detection system implemented has a module that analyzes the loging files of a Linux system. From these loging files, variables are obtained which are supervised to get normal use patterns. In addition it has a module that supervises the input/output traffic sampled of a network segment. When a malicious activity in some of both modules is detected, the information of both is correlated and if some anomaly is detected an alert is sent to the system manager.

Thus several hosts of a network are monitoring based on the operating system activities and the applications that are executed on it, adding the capacity to supervise the input/output traffic from these hosts. In this way, it can increase the confidence and exactitude in the intruders detection.

This hybrid IDS can detect from specific attacks as user names and passwords cracking, abuse of the web server resources, mail SPAM, to more general attacks for example, port scaning and IP address scaning, denial of service, among others.

The system was implemented and evaluated in two ways: in real time and simulation way. Tests were done in both ways, although in the simulation way there was more flexibility to model exhaustive platforms of tests. In the final system, the results of performance, rapidity and functionality were satisfactory.

Índice general

Índice de figuras

Índice de tablas

Índice de algoritmos

Capítulo 1

Introducción

Hasta finales de la década de 1980 muy poca gente tomaba en serio el tema de la seguridad en redes de computadoras de propósito general. Mientras que por una parte Internet iba creciendo exponencialmente con redes importantes que se adherían a ella, por otra el auge de la informática de consumo, unido a factores menos técnicos, iba produciendo un aumento espectacular en el número de piratas informáticos.

En esta época Robert T. Morris protagonizó el primer gran incidente de la seguridad informática: uno de sus programas se convirtió en un famoso gusano de Internet. Miles de computadoras conectadas a la red se vieron inutilizadas durante días, y las pérdidas se estiman en millones de dólares [15]. Desde ese momento el tema de la seguridad en sistemas operativos y redes ha sido un factor a tener muy en cuenta por cualquier responsable o administrador de sistemas informáticos.

La seguridad en computadoras puede ser dividida en dos grandes áreas: la seguridad preventiva y la detección de intrusiones [54]. Se ha puesto gran énfasis en la seguridad preventiva y se han realizado un alto número de investigaciones en esta rama, mientras que la detección de intrusiones había sido relativamente pasada por alto. Sin embargo, es inevitable abordar el segundo aspecto dedicado a intrusiones, ya que las medidas de prevención no son totalmente infalibles.

Actualmente existen varios mecanismos de seguridad para proteger los recursos informáticos de una empresa que pueden ser usados por personas no autorizadas mediante claves y controles de acceso, pero aunque se sigan todas las recomendaciones que hacen los expertos, no se está libre de posibles ataques con éxito. Esto se debe a que conseguir un sistema virtualmente invulnerable es sumamente costoso y prácticamente imposible.

Otro aspecto importante es que los ataques son cada vez más sofisticados, complejos y difíciles de contrarrestar. Dentro de las soluciones tecnológicas que se encuentran disponibles para reforzar la seguridad de una red se encuentran los cortafuegos. Un cortafuegos se utiliza para brindar protección perimetral a la red [52] y de este modo controlar el acceso a ella, según las políticas de seguridad establecidas por el administrador de la red.

Aunque un cortafuegos ofrece una buena cantidad de beneficios en cuanto a la seguridad de una red [51], este tipo de mecanismos han resultado insuficientes. El atacante puede lograr burlar el cortafuegos dejando la red a su merced. Además un cortafuegos

protege de los accesos no autorizados hacia la red interna, pero no protege a las máquinas ubicadas en la red perimetral como servidores web, servidores de correo, servidores FTP, en otras palabras, a las bases funcionales de Internet. Otra desventaja es que un cortafuegos no protege contra ataques desde dentro de la red.

Finalmente, otro aspecto tiene que ver con las vulnerabilidades que se pueden aprovechar de una aplicación de *software*. Por un lado, los programadores de dicho *software* deben ser cuidadosos a la hora de la implementación para corregir todo tipo de errores que faciliten las actividades de un intruso. Por otro lado, los usuarios o administradores de una aplicación deben ser igualmente cuidadosos a la hora de configurar el servicio, ya que el buen funcionamiento de una aplicación puede depender en mayor o menor medida de una buena configuración.

Definitivamente las redes están expuestas a intentos de intrusión y ataques. Lo que hay que hacer en estos casos es detectar el ataque o la intrusión lo antes posible para que cause el menor daño en el sistema. Los problemas que se pueden generar si un usuario malicioso aprovecha las vulnerabilidades de un sistema pueden llegar a ser graves.

Normalmente un intruso intenta acceder a una determinada información, manipularla de alguna forma y hacer que el sistema no funcione de forma segura o inutilizarlo. Una intrusión es, entonces, cualquier conjunto de acciones que pueden comprometer la integridad, confidencialidad o disponibilidad de la información o un recurso informático. Los intrusos pueden usar debilidades y huecos en la arquitectura de los sistemas y el conocimiento interno del sistema operativo para superar el proceso normal de autenticación, cuidando siempre de ocultar su identidad [32].

Por todas estas razones se han propuesto los Sistemas de Detección de Intrusos (SDI) [30] como una herramienta complementaria de seguridad que intenta detectar o supervisar los eventos ocurridos en un determinado sistema informático en busca de actividades que pretendan comprometer la seguridad de dicho sistema.

La información de las actividades de un sistema puede ser registrada en las bitácoras de una aplicación o del sistema operativo de una computadora, o bien en los paquetes que circulan por la red.

Si la fuente de información de un SDI son las bitácoras de una aplicación o del sistema operativo de una computadora, el SDI es conocido como basado en huésped (SDIh) [59]. Los SDIh vigilan un único equipo ya que sólo pueden identificar eventos en el equipo donde se encuentra instalado. Con los SDIh, podemos detectar ataques como una exploración de contraseñas de administrador de un huésped, analizando alguno de los archivos de bitácora. De estos mismos archivos se podrían sacar patrones del servicio de correo electrónico, por ejemplo, correo SPAM, contando el número de correos que recibe un servidor registrados en la bitácora *maillog*.

Se pueden encontrar varios SDIh disponibles como OSSEC [6] Asick [4], Md5deep [5], Samhain [50] y Aide [17], todos éstos son de código libre. También se pueden encontrar varios SDIh comerciales como el Tripwire Enterprise [29].

Si la fuente de información a analizar por el SDI son los paquetes circulantes por la red, el SDI es conocido como basado en red (SDIr) [16]. Los SDIr pueden vigilar todo un segmento de red, es decir, brindan protección simultánea a varios sistemas de cómputo.

Con los SDIr podemos detectar ataques que intenten inundar nuestra red de paquetes entrantes y bloquear nuestro sistema, éstos se conocen como los ataques de negación de servicio y generan una gran cantidad de tráfico. Otros ataques son exploraciones de puertos y direcciones IP, *spoofing*, programas maliciosos y ataques a aplicaciones, entre muchos otros.

Hoy en día se encuentran disponibles una gama diversa de SDIr. Uno de los más populares es Snort [47] de código libre, además de otros SDIr comerciales como Shadow [40], Dragon [38], NFR [45], RealSecure [45] y NetProwler [13], entre otros.

La mayor desventaja atribuida a la detección de intrusiones es la sobrecarga que puedan ejercer sobre los recursos del sistema. Pero en la práctica, tienen un grado de efectividad altamente confiable sobre las propias habilidades de los administradores de sistemas.

Analizando los dos tipos de SDI —red y huésped—, es claro que ambos presentan ventajas específicas de cada enfoque, pero también presentan carencias. Así pues, resulta conveniente combinar ambos sistemas para explotar las mejores características de los dos enfoques. El resultado son los sistemas de detección de intrusos híbridos (SDIh&r) [31], los cuales combinan las mejores características de los SDIh y los SDIr en una configuración del sensor del SDI. Lo que hacen este tipo de sistemas es complementar la tecnología de los SDIh con la habilidad de supervisar el tráfico de red que entra o sale de un huésped específico.

Por ejemplo, los SDIh&r pueden detectar fácilmente a los atacantes que ejecuten un programa malicioso desde un huésped vulnerable y que además sean tan inteligentes de borrar las huellas de su acceso en las bitácoras del sistema. Al momento en que el atacante accede al sistema, el SDIh&r registrará la actividad de las bitácoras en tiempo real. Así aunque el atacante borre sus huellas de las bitácoras, éstas ya habrán sido registradas por el sistema híbrido. De este modo se podrá detener el ataque a tiempo.

Actualmente existen ya algunas implementaciones de SDIh&r. Uno de ellos es llamado Sistema de Detección de Intrusos Distribuido (DIDS, por sus siglas en inglés). Éste es un prototipo desarrollado en la Universidad de California [56] que combina la supervisión distribuida y la reducción de datos con análisis centralizado de datos para supervisar una red heterogénea de computadoras. Este prototipo se encuentra en desarrollo y no está disponible aún.

Prelude [24] es una Sistema de Detección de Intrusos Distribuido bajo la licencia GPL. Prelude intenta utilizar la información de los huéspedes y del tráfico de la red para detectar anomalías o ataques. Así, genera respuestas con base en los ataques a de red o del sistema.

Los SDIh&r existentes presentan los problemas comunes a este tipo de sistemas. En primer lugar problemas de comunicación, debido a que todos los elementos del sistema deben hablar el mismo lenguaje. Además, debido a que la cantidad de información que estos sistemas analizan es extensa, consumen una gran cantidad de recursos creando sobrecarga en el sistema que está siendo supervisado.

Por lo anterior, la motivación para realizar este trabajo de tesis, es desarrollar un SDIh&r, que combine las ventajas que ofrecen los SDIr con las ventajas que ofrecen los SDIh para así implementar un SDI confiable que pueda detectar una gama amplia de ataques que no es posible detectar con un sólo enfoque. El sistema propuesto es un sistema

eficiente, con un costo computacional bajo, ya que procesa la información conforme va llegando y hace esto una sola vez, después la clasifica y almacena de modo que quede disponible para que el sistema haga uso de ella. Una característica importante de este sistema es que es completamente modular, es decir, sus módulos pueden ser ejecutados de manera autónoma y realizar la función para la cual fueron diseñados, y también pueden acoplarse para trabajar de manera conjunta y aumentar así la calidad y la certeza de las detecciones.

1.1. Objetivo

El objetivo general de este trabajo de tesis es usar diversas técnicas de supervisión de bitácoras de uso para un servidor Linux, para extraer patrones de acceso que determinen comportamientos normales y combinar esta información con la información extraída de un análisis estadístico de tráfico. De esta manera, es posible desarrollar un sistema de detección de intrusos basado en anomalías [56], que recolecte información, obtenga patrones de uso normal y analice estos patrones en busca de comportamientos atípicos, con el propósito de generar mecanismos de respuesta ante una anomalía detectada.

El SDI desarrollado se dividió en los siguientes tres módulos: en primer lugar, un módulo de captura de tráfico mediante el protocolo sFlow [49], el cual permite que equipos de comunicaciones avanzados capturen la información de las sesiones que se utilizan a través de sus puertos. El segundo módulo consiste de un agente que se encarga de supervisar toda la actividad que sucede en un huésped a nivel de aplicaciones y del sistema operativo. Finalmente, el tercer módulo consiste en otro agente que combina la información proveniente de un huésped con la información proveniente de la red, al cual llamaremos módulo de interconexión.

Además el SDIh&r cuenta con una interfaz de usuario en la cual se presentan todos los resultados obtenidos con este sistema. Esta interfaz de usuario ofrece varios servicios al administrador, como generar mensajes de alerta presentados mediante mensajes de texto en la pantalla, visualizar gráficamente la actividad de un segmento de red y de todos los huéspedes que la conforman durante un periodo de tiempo determinado y generar reportes correspondientes al mismo periodo. Además permite al usuario configurar ciertas opciones para el funcionamiento del sistema.

1.2. Metodología

La metodología que se siguió para la elaboración de esta tesis fue la siguiente:

1. En primer lugar se hizo una revisión de la bibliografía para hacer la redacción del protocolo de tesis y realizar el marco teórico del trabajo.

2. Se hizo un análisis de todas las posibles variables que se pueden obtener de los archivos de bitácora de un servidor Linux, las cuales se emplearon para obtener patrones

de uso. Estas variables fueron extraídas de los archivos de bitácoras mediante *scripts* programados en Perl.

3. Se implementó un prototipo del SDIh con el módulo de detección de intrusos basado en huésped que analizó tres variables solamente y se desarrolló el sitio web para este módulo. Este prototipo permitió delimitar el alcance de las variables observadas con las bitácoras.

4. Se complementó el módulo de detección de intrusos basado en huésped con ocho variables más para ampliar el panorama del estado del sistema de cada huésped.

5. Se analizó la forma en que sFlow supervisa el tráfico de la red y con base en éste se implementó el módulo de detección de intrusos basado en red.

6. Se programó el módulo de interconexión entre el módulo de detección de intrusos basado en red y el módulo de detección de intrusos basado en huésped que da la funcionalidad de híbrido al sistema, ya que combina las ventajas de los sistemas de detección de intrusos basados en red con los basados en huésped para así detectar intrusiones más completas.

7. Se realizaron los puntos 4, 5 y 6 en modo de simulación para verificar la funcionalidad del sistema y hacer pruebas exhaustivas.

8. Se realizaron los puntos 4, 5 y 6 en modo de tiempo real probados en la red del Departamento de Computación del Cinvestav y en el servidor llamado delfos perteneciente a la red del Cinvestav.

9. Se desarrolló una interfaz de usuario para probar los dos modos en que el SDI fue implementado.

10. Se reportan los resultados obtenidos.

El sistema implementado en este trabajo de tesis ha sido probado desde el mes de enero del año 2006. El sistema es capaz de ejecutarse de manera modular, o bien, de manera conjunta mediante el módulo de interconexión. Además, detecta ataques o anomalías hacia un huésped específico y del mismo modo extiende estas detecciones a nivel de la red. Es un sistema adecuado para ambientes distribuidos, robusto y de bajo costo computacional, lo que lo hace bastante rápido.

1.3. Organización de la tesis

Este documento de tesis se encuentra organizado de la siguiente manera:

1. En el segundo capítulo se dan las bases teóricas para entender el trabajo de tesis.

2. En el tercer capítulo se presentan todos los detalles de la implementación del sistema de monitoreo y detección de intrusos en servidores Linux.

3. En el cuarto capítulo se presenta el desarrollo de la interfaz web que además de servir para la evaluación del sistema, brinda servicios que facilitan al usuario el manejo del sistema y la visualización de los resultados.

4. En el capítulo cinco se presentan las conclusiones del trabajo de tesis y el trabajo futuro.

5. Finalmente, se presenta un apéndice con el manual para el sistema de monitoreo y detección de intrusos en modo normal y en modo de simulación.

Capítulo 2

Sistemas de Detección de Intrusos

En este capítulo se tocará el tema de la seguridad en redes de computadoras y se describirá detalladamente a los Sistemas de Detección de Intrusos, los cuales son una de las tecnologías más usadas para contrarrestar ataques o intrusiones a un sistema.

2.1. Seguridad informática

La seguridad en sistemas informáticos tiene como propósito asegurar que un sistema esté libre de todo peligro, daño o riesgo, y que es, en cierta manera, infalible. Esta característica para el caso de sistemas operativos o redes de computadoras es muy difícil de conseguir, así que el concepto se suaviza a otro término llamado fiabilidad. La fiabilidad es la probabilidad de que un sistema se comporte tal y como se espera; por tanto, se habla de sistemas fiables en lugar de hacerlo de sistemas seguros [30] (pp. 2-3).

Mantener un sistema seguro o fiable consiste básicamente en garantizar tres aspectos [1]: confidencialidad, integridad y disponibilidad. La confidencialidad establece que los objetos de un sistema puedan ser accedidos sólo por elementos autorizados a ello, y que esos elementos no harán mal uso de éstos; la integridad significa que los objetos sólo pueden ser modificados por elementos autorizados, y de una manera controlada, y la disponibilidad indica que los objetos del sistema tienen que permanecer accesibles a elementos autorizados cuando éstos los requieran.

A través de los años se han desarrollado diferentes formas para explotar los agujeros de seguridad que hay en un sistema informático. Existen ataques simples como las exploraciones de puertos que intentan explorar los puertos de un equipo en busca de vulnerabilidades. Otra forma de dañar a un sistema es usando programas maliciosos como caballos de troya, virus y gusanos; estos programas tienen el propósito de causar daños al equipo infectado y propagarse a través de la red con el fin de espiar, robar información y destruir datos, por ejemplo. Otros tipos de ataque son los famosos ataques de negación de servicio que intentan inhabilitar a un sistema enviándole tantas peticiones hasta que sea incapaz de responder a tanta demanda.

Básicamente existen dos tipos de ataques que pueden llegar a afectar a los sistemas y redes [22]: los *Ataques activos* y los *Ataques pasivos*. Los ataques activos implican algún

tipo de modificación de los datos o la creación de datos falsos. En los ataques pasivos el atacante no altera la comunicación, sino que únicamente la escucha o supervisa para obtener de esta manera la información que está siendo transmitida.

2.1.1. Seguridad, ataques y vulnerabilidades

Un protocolo de comunicación generalmente está dado por un intercambio de información entre dos o más personas. Por ejemplo: un cliente y un servidor que intercambian información a través de una red como se muestra en la Figura 2.1. En este esquema también existen actores maliciosos llamados intrusos que espían y roban la información que es transmitida aprovechándose de una vulnerabilidad o hueco de seguridad del sistema. Una vulnerabilidad [7] es un error que compromete la seguridad de un programa o sistema; generalmente son errores de programación o pueden deberse a errores en la configuración del servicio. Así, tanto los desarrolladores como los administradores de *software* deben poner especial atención en detectar lo antes posible todas las vulnerabilidades que un sistema puede tener para corregirlo y evitar ser objeto de ataques potenciales.

Los intrusos se clasifican con base en la magnitud del daño que causan al sistema atacado [30] (pp. 6-7):

- Intrusos curiosos. Estos intrusos son personas curiosas probablemente sin mucho conocimiento en informática que intentan acceder a sistemas a los que oficialmente no tienen acceso y no causan mayor daño al sistema, sólo lo espían.

- *Crackers.* Generalmente son usuarios con cierto nivel de conocimiento que intentan usar un sistema de manera ilegítima para ejecutar un programa malicioso o robar información por beneficio personal. Realizan acciones maliciosas contra un sistema para lograr popularidad, anunciando sus habilidades y haciendo daño a su objetivo.

- Intrusos remunerados. Este tipo de intrusos es el más peligroso, aunque por fortuna el menos habitual en redes normales más bien suelen afectar a grandes instituciones; se trata de usuarios maliciosos con gran experiencia en problemas de seguridad y un amplio conocimiento del sistema, que además son remunerados por terceras personas para robar secretos o emprender acciones maliciosas con fines delictivos.

Los intrusos expertos siguen tres pasos para llevar a cabo un ataque con éxito: primero preparan el ataque buscando vulnerabilidades de un sistema de las cuales se puedan aprovechar; esta actividad se puede hacer con una exploración de puertos, la cual proporciona a los atacantes información de los puertos abiertos y de las aplicaciones ejecutadas en éstos puertos y así pueden definir los blancos de ataque. En segundo lugar, lanzan el ataque que probablemente intente en primer término ingresar al sistema y si este ataque tiene éxito se intentará robar o destruir información del sistema mediante programas maliciosos. Finalmente el intruso borra el rastro de su acceso y generalmente lo hará ingresando a las bitácoras del sistema para suprimir los registros que delaten su ingreso.

Las amenazas a la seguridad de cualquier red, así como las habilidades y sofisticación de los atacantes está en aumento constante. Conforme las redes de cualquier organización

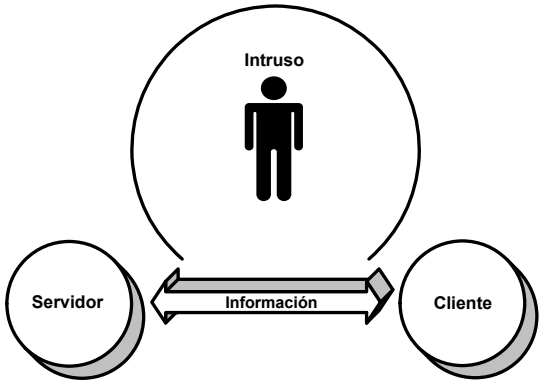


Figura 2.1: Ejemplo de un protocolo de comunicación

se expanden tanto en tamaño como con nuevas aplicaciones, las tecnologías de seguridad para estas redes desempeñan un papel cada vez más importante . Estas tecnologías buscan proteger a las redes y tratar de eliminar el creciente número de vulnerabilidades que éstas presentan.

Hoy en día las redes son más susceptibles a ataques que en los años pasados, ya que los usuarios acceden a la red a través de conexiones de banda ancha y móviles, así como a través de Internet. Este aumento en el nivel de acceso introduce ataques maliciosos a los que se conoce como ataques remotos.

Exploración de puertos

Los puertos de una computadora pueden encontrarse en varios estados: abierto, cerrado o bloqueado. El estado más vulnerable a un ataque es cuando el puerto se encuentra abierto, esto significa que una aplicación servidor está escuchando por ese puerto las peticiones de los clientes que se conecten. Un puerto abierto puede brindar información sobre las vulnerabilidades de seguridad del sistema. Por esta razón, una de las primeras actividades que un atacante intentará realizar en contra de un sistema es sin duda una exploración de puertos y así obtener información básica acerca de los servicios ofrecidos y, adicionalmente, otros detalles del entorno como que sistema operativo está instalado en cada huésped o ciertas características de la arquitectura de la red.

Al fin y al cabo los puertos abiertos son puntos de acceso a aplicaciones que corren en una computadora y estas aplicaciones pueden tener vulnerabilidades que pueden ser aprovechadas por otros usuarios.

Existen varios puertos famosos y conocidos por las vulnerabilidades que presentan las aplicaciones que en ellos se ejecutan, dentro de éstos se encuentran puertos de inicio de sesión como el puerto 22 (ssh), el puerto 23 (telnet) y el puerto 21 (ftp). También se encuentran puertos de servidores de nombres como el puerto 53 (DNS) y otros puertos como el puerto 69 (TFTP) y el puerto 515 (*lpd*) entre otros. Por ejemplo, la aplicación TFTP (Trivial File Transfer Protocol —Protocolo de transferencia de archivos trivial—) es

un protocolo de transferencia muy simple y a menudo se utiliza para transferir pequeños archivos entre ordenadores en una red. Esta aplicación es vulnerable ya que no utiliza mecanismos de encriptación o autenticación por lo que representa un hueco de seguridad que los usuarios maliciosos estarán anciosos por explotar. La aplicación *lpd* es la utilidad UNIX de impresión y permite entregar trabajos de impresión, ejecutarlos a través de filtros, gestionar las colas de impresión, y puede aceptar trabajos locales de impresión, o sobre la red, y acceder a varias partes del sistema, lo cual lo convierte en un potencial agujero de seguridad. De este modo un usuario debe ser consciente de bloquear los puertos que no sean usados ya que incluso, aunque los puertos estén bloqueados, un usuario malicioso puede acceder a la red por otros medios y atacar estos puertos si no están debidamente asegurados.

Spoofing

Por *spoofing* se conoce a la creación de tramas TCP/IP utilizando una dirección IP falseada; la idea de este ataque es muy sencilla: desde su equipo, un usuario malicioso simula la identidad de otra máquina de la red para conseguir acceso a recursos de un tercer sistema que ha establecido algún tipo de confianza basada en el nombre o la dirección IP del huésped suplantado. Existen muchos sistemas que basan su funcionamiento en anillos de confianza, que consiste en que los usuarios de un sistema aportan su clave pública al sistema y firman las claves del resto de los usuarios, así no hay forma segura de acreditar que una clave es de quien dice ser, mas que por la confianza que se pueda tener en el firmante de dicha clave. Debido a esto, el *spoofing* es en la actualidad un ataque muy usado, aunque sea no trivial, ya que se requiere gran conocimiento del protocolo TCP/IP.

En el *spoofing* entran en juego tres máquinas: un atacante, un atacado, y un sistema suplantado que tiene cierta relación con el atacado; para que el pirata pueda conseguir su objetivo necesita por un lado establecer una comunicación falseada con su objetivo, y por otro evitar que el equipo suplantado interfiera en el ataque [28]

Negación de servicio

Las negaciones de servicio (conocidas como *DoS, Denial of Service*) son ataques dirigidos contra un recurso informático, generalmente una máquina o una red, pero también podría tratarse de una simple impresora o una terminal. El objetivo es degradar total o parcialmente los servicios prestados por ese recurso a sus usuarios legítimos; constituyen en muchos casos uno de los ataques más sencillos y contundentes contra todo tipo de servicios, y en entornos donde la disponibilidad es valorada por encima de otros parámetros de la seguridad global, puede convertirse en un serio problema, ya que un pirata puede interrumpir constantemente un servicio sin necesidad de grandes conocimientos o recursos, utilizando programas sencillos, un módem y una computadora caseros. Las negaciones de servicio más habituales suelen consistir en la inhabilitación total de un determinado servicio o de un sistema completo, bien porque ha sido realmente bloqueado por el atacante o bien porque está tan degradado que es incapaz de ofrecer un servicio a sus usuarios. En la mayor parte de sistemas, un usuario con acceso a un *shell* no tendrá muchas dificultades

en causar una negación de servicio que degradará la máquina o disminuirá su velocidad enormemente; esto no tiene porque ser (y de hecho en muchos casos no lo es) un ataque intencionado, sino que puede deberse a un simple error de programación.

Existen dos tipos principales de ataques de negación de servicio [35]:

- Exploración de grietas o desperfectos. Una exploración de grietas utiliza un desperfecto en el *software* del sistema para causar una falla en los procesos o para agotar los recursos del sistema. Un ejemplo de este tipo de ataques es el famoso *ping* de la muerte. Éste es un tipo de ataque a computadoras que implica enviar un *ping* mal formado a una computadora. Un *ping* normalmente tiene un tamaño de 64 bytes; algunas computadoras no pueden manejar *pings* mayores al máximo de un paquete común IP que es de 65,535 bytes. Enviando *pings* de este tamaño se pueden hacer caer servidores. Este ataque ha sido fácil de explotar.

  Con respecto a los ataques que intentan agotar los recursos del sistema, éstos pueden ser: tiempo de procesador, memoria, espacio en disco, espacio en un *buffer* específico, o ancho de banda de una red.

- Ataques de inundación. Este tipo de ataques simplemente envían a un sistema o a un componente del sistema más información de la que puede manejar. En los casos en los que el atacante no puede mandar al sistema suficientes datos para inundar su capacidad de procesamiento, el atacante puede sin embargo ser capaz de monopolizar la conexión al sistema, negando a alguien más el uso del recurso. Con este tipo de ataques no existe ninguna grieta o desperfecto en el sistema que se deba reparar.

Un usuario malicioso puede ser tan astuto como para añadir a los ataques de negación de servicio la técnica de *spoofing* de direcciones IP [25], es decir, simular la identidad de una dirección IP legítima para obtener acceso a recursos restringidos y complicar la obtención del rastro del origen del ataque. ComputerWire reportó un ataque en el que siete de trece servidores DNS se bloquearon por una hora debido a un ataque de negación de servicio con *spoofing* de dirección IP [12].

Interceptación

La interceptación, también conocida por *passive wiretapping* [30] (pp. 29) es un proceso mediante el cual un agente capta información —en claro o cifrada— que no le iba dirigida. Aunque es en principio un ataque completamente pasivo, lo más peligroso de la interceptación es que es muy difícil de detectar mientras se produce, de forma que un atacante puede capturar información privilegiada y claves para acceder a más información sin que nadie se de cuenta hasta que dicho atacante utiliza la información capturada, convirtiendo el ataque en activo.

Un medio de interceptación bastante habitual es el *sniffing*, el cual consiste en capturar tramas que circulan por la red mediante un programa que está ejecutándose en una máquina conectada a ella o bien mediante un dispositivo que se engancha directamente

al cableado. Estos programas son capaces de capturar directamente correos electrónicos o cargar de forma automática en un navegador las páginas que visita la víctima del ataque. Otro ataque de interceptación es el *keylogging* [30] (pp. 301-302), que registra las teclas pulsadas por un usuario en una sesión y de este modo se pueden adquirir contraseñas y otros datos confidenciales.

Ataques a aplicaciones

Debido a que una red se encuentra compuesta de huéspedes importantes como servidores de correo, servidores web, servidores ftp entre otros, un atacante estará muy interesado de aprovechar las vulnerabilidades de dichas máquinas. Entre los ataques más comunes a los que los huéspedes de una red se encuentran vulnerables se tienen:

- Correo electrónico. Desde hace muchos años los sistemas de correo electrónico de una organización han sido para los piratas una fuente inagotable de puntos de entrada a ella.

  Sendmail, por ejemplo es un popular agente de transporte de correo en Internet, cuya tarea consiste en encaminar los mensajes de correo de forma que estos lleguen a su destino. Este tipo de programas proporcionan datos importantes que pueden facilitar mucho la tarea de un pirata. Independientemente del programa que se utilice como servidor de correo y su versión concreta, con vulnerabilidades conocidas o no, otro gran problema de los sistemas de correo *smtp* es el *relay*: la posibilidad de que un atacante interno utilice los servidores para enviar correo electrónico a terceros, no relacionados con una organización. Aunque en realidad parecerá esto un mal menor, no lo es; de entrada, si los servidores permiten el *relay* se favorece el correo *spam* (correo no deseado) en la red con fines casi siempre publicitarios. Además, el *relay* causa una negación de servicio contra los usuarios legítimos desde un punto de vista estrictamente teórico, ya que alguien consume los recursos de forma no autorizada, degradando así el servicio ofrecido a los usuarios legítimos.

- Durante los últimos años los servidores web se han convertido en una fuente de ataque para los usuarios maliciosos, así también, se han convertido en un requerimiento de la mayoría de las organizaciones con fines al menos publicitarios. Ahora, resulta muy fácil para un usuario malicioso conseguir información valiosa de la red tecleando en un navegador el nombre del objetivo y contactar con al menos una de sus máquinas: su servidor web

  Uno de los ataques que un usuario malicioso intentará contra un servidor web será modificar los sitios web de dicho servidor, tal vez con fines meramente personales y el ataque no pase de ahí. Sin embargo, puede haber un usuario con más experiencia que intente dañar el sistema; esto es factible ya que el servidor web proporciona excesiva información sobre su configuración, y además esta herramienta tiene algunos archivos y directorios que pueden resultar interesantes para un atacante: en el caso de los CGI que es un programa ejecutable, es equivalente a

permitir a cualquiera ejecutar un programa en un sistema específico, y estos programas pueden ser vulnerables. Los CGI se utilizan para modificar páginas web, robar información de tarjetas de crédito e instalar puertas traseras que les servirán posteriormente para tener acceso a los sistemas comprometidos. El caso de los directorios es algo diferente, pero se trata de acceder —por medio del servidor web— a directorios con nombres comunes; estos directorios contienen información útil para un atacante potencial.

La mayor parte de estos ataques tiene éxito gracias a una configuración incorrecta del servidor o a errores en su diseño: existen diferentes tipos de servidores web, desde los complejos aunque difíciles de administrar correctamente, hasta los simples en su instalación pero con potenciales vulnerabilidades.

- Intentos de acceso no autorizados con contraseñas o usuarios incorrectos mediante una conexión remota. En la actualidad es fácil implementar un *script* que intente acceder a una computadora por ssh [62] —o algún otro protocolo de inicio de sesión— , con nombres y contraseñas generados por el mismo *script*. Lo que se hace es crear de manera aleatoria contraseñas para un usuario conocido o común, intentando atinar a la contraseña verdadera. Este tipo de ataques son de fuerza bruta, es decir, el programa intenta probar una por una todas las posibilidades y con algo de suerte acertar en alguna de ellas.

  Otra forma común de realizar este ataque es mediante diccionarios; en un ataque de diccionario [23], el atacante toma un diccionario de palabras y nombres, e intenta acceder a una computadora mediante una conexión remota con cada combinación de nombres y contraseñas posibles de este diccionario, y tal vez con algunas pequeñas variaciones fáciles de idear. Este ataque es fácil de llevar a cabo y si el atacante logra adivinar algún nombre de usuario y contraseña legítimos ganará acceso al sistema con los privilegios del usuario usurpado.

- Programas maliciosos. Ningún sistema operativo o aplicación es vulnerable a los programas maliciosos, a menos que algún programa externo, por simple que sea, puedan ser ejecutado en el sistema. Hoy en día existen diferentes tipos de programas maliciosos, los cuales varían en su comportamiento.

  Un troyano es un programa malicioso capaz de alojarse en computadoras y permitir el acceso a usuarios externos, a través de una red local o de Internet [61], con el fin de recabar información o controlar remotamente a la máquina anfitriona, pero sin afectar al funcionamiento de ésta. Una vez instalado parece realizar una función útil (aunque cierto tipo de troyanos permanecen ocultos y por tal motivo los antivirus o anti troyanos no los eliminan) pero internamente realiza otras tareas de las que el usuario no es consciente.

  Se utilizan para espiar, destruir o borrar algunos archivos del sistema, incluso formatear discos duros [61]; estos programas pueden usar la computadora infectada para lanzar ataques en contra de otras computadoras o para instalar

un *software* de acceso remoto que permita supervisar lo que hace el usuario legítimo de la computadora y, por ejemplo, capturar las pulsaciones del teclado con el fin de obtener contraseñas u otra información sensible. La mejor defensa contra los troyanos es no ejecutar nada de lo cual se desconozca el origen y mantener *software* antivirus dotado con las actualizaciones adecuadas.

Un virus Un virus informático es un programa que se copia automáticamente y que tiene por objeto alterar el funcionamiento de la computadora, sin el permiso o el conocimiento del usuario. Éstos reemplazan archivos ejecutables por otros, infectados con el código del virus. Los virus [26] pueden intencionadamente destruir datos en la computadora, aunque también existen otros que solo se caracterizan por ser molestos. En un ataque por virus, el virus busca por un sistema conectado a la red y copia remotamente el código del virus a este sistema; después este programa inicia una compilación remota reproduciendo una copia exacta del virus propagado. Uno de los virus más ampliamente conocidos que irrumpió en computadoras de todo el mundo es el gusano Blaster o Lovsan [43]. El CERT reportó que Blaster aprovechó una vulnerabilidad recientemente descubierta en los sistemas Windows y la ejecución de esta vulnerabilidad permite el acceso en línea de comandos en forma remota con privilegios de administrador.

Después de revisar los diferentes ataques a los que una red o un huésped de ella se encuentran expuestos, se puede observar una clara diferencia entre cada uno de estos tipos de ataques. En primer lugar existen dos tipos: los dirigidos a una red y los dirigidos a un huésped en específico de dicha red. Además, los ataques dirigidos a una red generalmente generan una gran cantidad de tráfico circulante a través de la red y los ataques dirigidos a un huésped no generan esta cantidad de tráfico, pero cuentan con la ventaja de que la información de cada actividad queda registrada en mayor o menor grado en las bitácoras del sistema.

2.2. Herramientas de seguridad informática

A los mecanismos utilizados para implementar las políticas de seguridad se les denomina mecanismos de seguridad, y se convierten en una herramienta básica para garantizar la protección de los sistemas o de la propia red.

Los mecanismos de seguridad se dividen en tres grandes grupos [30]: mecanismos de prevención, mecanismos de detección y mecanismos de recuperación.

2.2.1. Mecanismos de prevención

Los mecanismos de prevención son aquellos que aumentan la seguridad de un sistema durante el funcionamiento normal de éste, previniendo la ocurrencia de violaciones a la seguridad; por ejemplo, el uso de cifrado en la transmisión de datos se puede considerar

un mecanismo de este tipo, ya que evita que un posible atacante escuche las conexiones hacia o desde un sistema en la red. Los mecanismos de prevención más habituales en redes son los siguientes [42]:

- Mecanismos de autenticación e identificación. Estos mecanismos hacen posible identificar entidades del sistema de una forma única, y posteriormente, una vez identificadas, autenticarlas (comprobar que la entidad es quién dice ser). Son los mecanismos más importantes en cualquier sistema, ya que forman la base de otros mecanismos que basan su funcionamiento en la identidad de las entidades que acceden a un objeto. Un grupo especialmente importante de estos mecanismos son los denominados Sistemas de Autenticación de Usuarios. Existen varias técnicas [20] para identificar a un usuario legítimo: puede ser por alguna contraseña que sólo el usuario legítimo tiene y es su deber y responsabilidad mantenerlo seguro; otra técnica puede ser por algún objeto que el usuario legítimo posea, como una tarjeta inteligente del tamaño de una credencial que ofrece funciones para un almacenamiento seguro de información y también para su procesamiento; una técnica muy segura es la autenticación biométrica ya que se basa en el reconocimiento de una característica física única de un individuo como sus huellas dactilares o la pupila de uno de sus ojos. Estas características no cambian con el pasar del tiempo por lo que un usuario podrá ser reconocido en cualquier situación.

- Mecanismos de control de acceso. Cualquier objeto del sistema ha de estar protegido mediante mecanismos de control de acceso, que controlan todos los tipos de acceso sobre el objeto por parte de cualquier entidad del sistema. Dentro de Unix, el control de acceso más habitual es el discrecional, implementado por los bits rwx y las listas de control de acceso para cada archivo (objeto) del sistema; sin embargo, también se permiten especificar controles de acceso obligatorio (MAC).

- Cortafuegos. Un cortafuegos es un sistema o grupo de sistemas que hace cumplir una política de control de acceso entre dos redes. De una forma más clara, podemos definir un cortafuegos como cualquier sistema utilizado para separar —en cuanto a seguridad se refiere— una máquina o subred del resto, protegiéndola así de servicios y protocolos que desde el exterior puedan suponer una amenaza a la seguridad. El espacio protegido, denominado perímetro de seguridad, suele ser propiedad del mismo sistema que se está protegiendo, y la protección se realiza contra una red externa, no confiable, llamada zona de riesgo.

- Mecanismos de seguridad en las comunicaciones. Es especialmente importante para la seguridad de nuestro sistema el proteger la integridad y la privacidad de los datos cuando se transmiten a través de la red. Para garantizar esta seguridad en las comunicaciones, hemos de utilizar ciertos mecanismos, muchos de los cuales se basan en la Criptografía [30] (pp. 11-12) : cifrado de clave pública, de clave privada, firmas digitales, etc. Aunque cada vez se utilizan más los protocolos seguros —como SSH [62] o Kerberos [33], en el caso de sistemas Unix en red—, aún es frecuente

encontrar conexiones en texto claro, ya no sólo entre máquinas de una misma subred, sino entre redes diferentes. Una de las mayores amenazas a la integridad de las redes es este tráfico sin cifrar, que hace extremadamente fáciles ataques encaminados a robar contraseñas o suplantar la identidad de máquinas de la red.

2.2.2. Mecanismos de detección

Por mecanismos de detección se conoce a aquellos que se utilizan para detectar violaciones de la seguridad o intentos de violación; ejemplos de estos mecanismos son los programas de auditoría. Casi todas las actividades realizadas en un sistema Linux son susceptibles de ser, en mayor o menor medida, supervisadas debido a la facilidad que tienen estos sistemas de registrar gran parte de las actividades ocurridas en éste. Sin embargo, debido a la cantidad inmensa de información que se recoge en las bitácoras del sistema es conveniente implementar herramientas que revisen y analicen automáticamente esta información.

Dentro de los mecanismos de detección se encuentra también una herramienta que se ha hecho popular hacia los últimos años, se trata de los Sistemas de Detección de Intrusos; éstos supervisan y registran toda actividad de un sistema para analizarlas en busca de alguna actividad maliciosa y dar así una respuesta a dicha actividad.

Los Sistemas de Detección de Intrusos no son la única medida de control para garantizar la seguridad de un sistema informático pero sí una herramienta que en la práctica tiene un grado de efectividad altamente confiable. Se encuentran disponibles varios SDI, entre ellos: Tripwire [29], OSSEC [6], Snort [47], RealSecure [14], Prelude [24], entre otros.

2.2.3. Mecanismos de recuperación

Los mecanismos de recuperación son aquellos que se aplican cuando una violación del sistema se ha detectado, para regresarlo a su funcionamiento correcto; ejemplos de estos mecanismos son la utilización de copias de seguridad o el *hardware* adicional. Otro ejemplo de este tipo de mecanismos pueden ser los antivirus —aunque cabe aclarar que también pueden ser mecanismos de detección—. Los antivirus detectan y eliminan virus informáticos y otros programas maliciosos. Básicamente un antivirus compara el código de cada archivo con una base de datos de los códigos —también conocidos como vacunas— de los virus conocidos, por lo que es importante actualizarla periódicamente a fin de evitar que un virus nuevo no sea detectado.

Parece claro que, aunque los tres tipos de mecanismos son importantes para la seguridad de un sistema, se ha de enfatizar en el uso de mecanismos de prevención y de detección. Para el administrador de un sistema, evitar un ataque, detectar un intento de violación, o detectar una violación exitosa inmediatamente después de que ocurra es mucho más productivo y menos comprometedor para el sistema que restaurar el estado tras una penetración de la máquina. Por lo anterior, es necesario combinar más de una herramienta para afianzar el nivel de seguridad de un sistema. A continuación se presenta una de las herramientas para detección que se utiliza para establecer niveles de seguridad.

2.3. Sistemas de Detección de Intrusos (SDI)

A pesar de que un enfoque clásico de la seguridad de un sistema informático siempre define como su principal defensa a los controles de acceso, esta visión es extremadamente simplista si se tiene en cuenta que en muchos casos esos controles no pueden proteger siempre ante un ataque [21]. Por poner un ejemplo sencillo, imagine en un cortafuegos donde se ha implantado una política que deje acceder al puerto 80 de los servidores web de una red desde cualquier máquina de Internet; ese cortafuegos sólo comprobará si el puerto destino de una trama es la que se ha decidido para el servicio http, pero seguramente no tendrá en cuenta si ese tráfico representa o no un ataque o una violación de una política de seguridad: por ejemplo, no detendrá a un pirata que trate de acceder al archivo de contraseñas de una máquina aprovechando un hueco o vulnerabilidad del servidor web. Desde un pirata informático externo a una organización hasta un usuario autorizado que intenta obtener privilegios que no le corresponden en un sistema, el entorno de trabajo no va a estar nunca a salvo de intrusiones.

Una intrusión es un conjunto de acciones que intentan comprometer la integridad, confidencialidad o disponibilidad de un recurso [27]; una intrusión no tiene por qué consistir en un acceso no autorizado a una máquina: también puede ser una negación de servicio. A los sistemas utilizados para detectar las intrusiones o los intentos de intrusión se les denomina Sistemas de Detección de Intrusiones o más habitualmente Sistemas de Detección de Intrusos (SDI).

Los SDI supervisan y registran los eventos que ocurren en una computadora o en una red de computadoras, para analizarlos y correlacionarlos en la búsqueda de señales de intrusión y dar así una respuesta que permita corregir esta situación [36]. Además, la información entregada por un SDI, permite vincular una intrusión dada a un posible responsable para tomar las acciones convenientes. Debido a esto, los SDI han ganado aceptación como herramientas adicionales para robustecer la seguridad en las organizaciones.

Los SDI se diferencian en tres aspectos principales: fuente de datos, tipo de análisis y los mecanismos de respuestas como se muestra en la Figura 2.2.

2.3.1. Fuentes de datos

Las fuentes de datos corresponden a los datos que maneja el sistema para analizar las posibles intrusiones y es uno de los primeros aspectos a tener en cuenta a la hora de diseñar un SDI.

El modo en que se recolectan los datos está determinado por la política de recolección de datos, que define el filtro que se usará para la notificación de los eventos. El evento generador produce un conjunto de eventos que representan las fuentes de información del SDI.

Se diferencian tres fuentes principales: basadas en los registros de auditoría que generan las máquinas, basadas en las aplicaciones informáticas, y basadas en la información que circula en las redes telemáticas. De este modo, los tipos de datos que deben procesar los SDI pueden ser paquetes de red, registros de auditoría, bitácoras producidas por

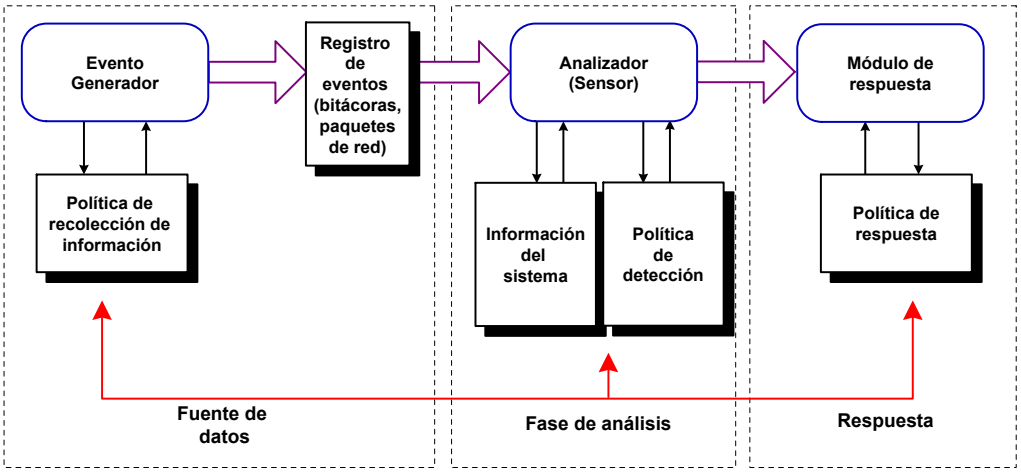


Figura 2.2: Componentes de un SDI

diferentes aplicaciones, o incluso comandos del teclado o llamadas al sistema de procesos en ejecución. Con base en este tipo de información se puede hacer una clasificación de los SDI la cual se describe en la sección 2.4.

2.3.2. Tipo de análisis

Después del proceso de recopilación de información, se lleva a cabo el proceso de análisis. El objetivo principal del sensor del SDI es descartar información que le parezca irrelevante y dependiendo de la política de detección se harán las detecciones correspondientes. La detección de intrusos también se puede clasificar según los objetivos del motor de análisis. Los dos tipos principales son: detección de anomalías y detección de usos indebidos.

Detección de anomalías

La idea de los SDI basados en detección de anomalías es que conocen lo que es normal en una red o en las computadoras a lo largo del tiempo, desarrollando y actualizando conjuntos de patrones contra los que se van a comparar los eventos que se producen en los sistemas. Si uno de esos eventos (por ejemplo, una trama procedente de una máquina desconocida) se sale del conjunto de normalidad, automáticamente se cataloga como sospechoso.

Los SDI basados en detección de anomalías se basan en la premisa de que cualquier ataque o intento de ataque implica un uso anormal de los sistemas [11]. Pero, ¿cómo puede un sistema conocer lo que es y lo que no es normal en nuestro entorno de trabajo? Para conseguirlo, existen dos grandes aproximaciones [9]: o el sistema es capaz de aprenderlo por sí mismo —basándose por ejemplo en el comportamiento de los usuarios, de sus procesos, del tráfico de la red— o bien se le especifica dicho comportamiento mediante

un conjunto de reglas. La primera de estas aproximaciones utiliza básicamente métodos estadísticos aunque también existen modelos en los que se aplican algoritmos de aprendizaje automático; la segunda aproximación consiste en especificar mediante un conjunto de reglas los perfiles de comportamiento habitual basándose en determinados parámetros de los sistemas [9].

Para ver más claramente el concepto de la detección de anomalías, se puede pensar en la ejecución de un programa que se puede considerar crítico, por ser privilegiado: /bin/passwd [11]. Si se consigue diferenciar las ejecuciones de este comando que se consideran habituales —por ejemplo, cuando un usuario cambia su contraseña sin problemas, cuando se equivoca, cuando el sistema no le deja cambiarla por los motivos típicos como que sea débil o que la haya cambiado hace poco—, se podría especificar formalmente cada una de estas secuencias de operación. De esta forma, cada vez que un usuario invoque a /bin/passwd, el sistema de detección supervisará las operaciones que esa llamada genera, y considerará una intrusión a cualquiera que no coincida con las secuencias de operación habituales.

Detección de usos indebidos

El modelo de detección de los SDI basados en usos indebidos contiene colecciones de firmas o patrones de ataques conocidos, y supervisa los eventos registrados en el sistema en busca de ocurrencias de alguna firma de la colección [19]. Este modelo observa cualquier proceso que intente explotar los puntos débiles de un sistema en específico y la secuencia de acciones que integran esta actividad maliciosa las encapsula en una firma de ataque o patrón de ataque.

El funcionamiento de los SDI basados en la detección de usos indebidos presupone que se puede establecer patrones para los diferentes ataques conocidos y algunas de sus variaciones; mientras que la detección de anomalías conoce lo normal (en ocasiones se dice que tienen un conocimiento positivo) y detecta lo que no lo es, este esquema se limita a conocer lo anormal para poderlo detectar (conocimiento negativo). Snort [47] es un SDI que hace detecciones basadas en usos indebidos y una de sus características más apreciada, además de su funcionalidad, es su subsistema flexible de firmas de ataques ya que tiene una base de datos de ataques que ya son conocidos y se está actualizando constantemente. Esta base de datos contiene las características de los ataques de red y cuando uno de ellos ocurre, se compara con los patrones almacenados en la base de datos y si coincide con alguno de estos patrones se lanza una alerta.

Para ver más claramente la diferencia entre la detección basada en anomalías y la detección basada en usos indebidos, imaginemos un sistema de detección basado en supervisar las máquinas origen desde las que un usuario sospechoso conecta a un sistema: si se tratara de un modelo basado en la detección de anomalías, seguramente mantendría una lista de las dos o tres direcciones más utilizadas por el usuario legítimo, alertando al responsable de seguridad en caso de que el usuario conecte desde otro lugar; por el contrario, si se tratara de un modelo basado en la detección de usos indebidos, mantendría una lista mucho más amplia que la anterior, pero formada por las direcciones desde las

que sabemos con una alta probabilidad que ese usuario no se conectará, de forma que si detectara un acceso desde una de esas máquinas, entonces es cuando el sistema tomaría las acciones oportunas.

2.3.3. Mecanismo de respuesta

Los módulos de respuesta o también conocidos como mecanismos de respuesta representan otro de los factores que ayuda a definir el tipo de sistema de detección de intrusos. Los resultados obtenidos de la fase de análisis, se utilizan para tomar las decisiones que conducirán a una respuesta. El conjunto de acciones y mecanismos que se pueden efectuar en esta etapa es amplio.

En los mecanismos de respuesta se pueden diferenciar los sistemas de respuesta pasiva, que en lugar de tomar acciones, se limitan a generar la alerta correspondiente, y por otro lado, los sistemas de respuestas activas, que además de generar las alertas correspondientes, reaccionan modificando el entorno.

Respuestas pasivas

Las respuestas pasivas [8] proveen información del sistema a los usuarios, dejando a los humanos tomar una acción subsecuente basada en esa información.

Se generan alarmas y notificaciones por parte del SDI para informar a los usuarios cuando un ataque es detectado. Las formas más comunes de notificar una alarma es presentando un mensaje de alerta a través de una interfaz de usuario.

Respuestas activas

Este tipo de respuestas son acciones automatizadas tomadas cuando ciertos tipos de intrusiones son detectados. Se tienen tres categorías de respuestas activas:

- Incrementar la sensitividad de las fuentes de información. Ya que se conoce información adicional de lo que se sospecha es un ataque, podría incrementarse el nivel de sensitividad de las fuentes de información. Un ejemplo de esto es incrementar el número de paquetes que debe capturar un SDI basado en red, y no restringirlo solamente a un puerto o computadora. Esto permite a una organización tener una mayor cantidad de información que puede utilizar para dar soporte a la investigación y aprehensión de un atacante.

- Cambiar el ambiente. Esto consiste en detener un ataque en progreso, vía reconfiguración de dispositivos como ruteadores o sistemas de protección perimetral para bloquear el acceso del atacante.

- Tomar acciones contra el atacante. Esto involucra lanzar ataques en contra del intruso o intentar activamente obtener información acerca de la computadora del atacante o el sitio donde se encuentra. Sin embargo, este tipo de respuesta no es recomendable, debido a que muchos atacantes utilizan direcciones de red falsas cuando atacan

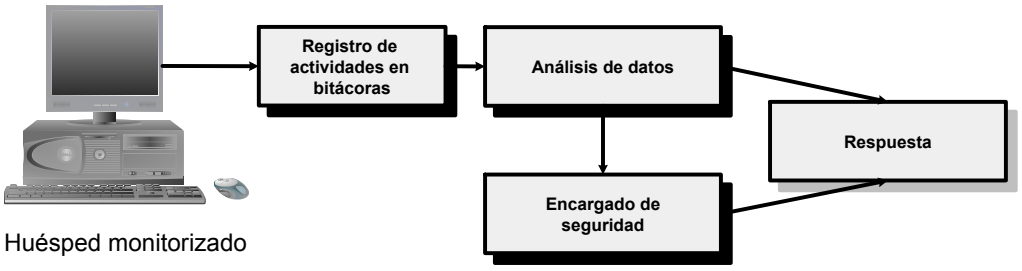


Figura 2.3: Sistema de detección de intrusos basado en huésped

sistemas, lo cual podría acarrear un gran riesgo el causar daños a sitios o usuarios inocentes de internet.

2.4. Clasificación de los SDI

Con base en el tipo de sistemas que vigilan, existen dos clases de SDI: los que analizan actividades de una única máquina en busca de posibles ataques, y los que lo hacen de una subred, aunque el sistema se instale en sólo uno de sus huéspedes.

2.4.1. Sistemas de detección de intrusos basados en huésped (SDIh)

Un SDIh es un mecanismo que permite detectar ataques o intrusiones contra la máquina sobre la que se ejecuta.

Los SDIh utilizan las bitácoras del sistema, las cuales se generan de forma automática por diferentes aplicaciones o por el propio núcleo del sistema operativo. Como se muestra en la Figura 2.3, se hace un análisis de las bitácoras prestando especial atención a los registros relativos a demonios de red, como un servidor web o el propio *inetd*. Por otra parte, se usan también verificadores de integridad de determinados ficheros de importancia vital para el sistema, como el de contraseñas.

La mayor parte de los Unix son capaces de registrar con una granularidad lo suficientemente fina casi todas las actividades que se llevan a cabo en el sistema, en especial aquellas que pueden suponer —aunque sea remotamente— una vulneración de su seguridad; sin embargo, el problema radica en que pocos administradores se preocupan de revisar con un mínimo de atención esas bitácoras, por lo que muchos ataques contra la máquina, tanto externos como internos, y tanto fallidos como exitosos, pasan finalmente desapercibidos. Parte de esta problemática se debe también en la gran cantidad de información generada, y el formato usado para almacenarla. Las bitácoras consisten de registros de actividades que suceden en el tiempo y dependiendo de la actividad, es el formato del registro. Frecuentemente para detectar una intrusión no basta analizar un registro sino es necesario analizar una gran cantidad de registros para detectar el patrón de una intrusión. Cuando

un gran número de registros con actividades no intrusivas se mezcla con una cantidad limitada de registros con actividades intrusivas, la tarea de análisis se vuelve complicada. Aquí es donde entran en juego las herramientas automáticas de análisis.

La verificación de integridad de archivos se puede realizar a diferentes niveles, cada uno de los cuales ofrece un mayor o menor grado de seguridad. Sea cual sea el modelo de verificación, en cualquiera de ellos se debe llevar a cabo inicialmente un paso común: generar una base de datos que contiene información contra la que posteriormente se comparará la información de cada archivo.

Básicamente, un SDIh supervisará el desempeño y el estado de un sistema de cómputo. Puede detectar con facilidad cada uno de los recursos que son accedidos por los programas que componen el sistema. Por ejemplo, puede proteger al sistema de que cualquier programa intente repentinamente cambiar la base de datos de contraseñas. Similarmente, puede analizar la información que se almacena en la memoria del sistema, ya sea en la memoria RAM o en el sistema de archivos, y verifica que el contenido de esta información sea consistente con la información esperada.

Una de las mayores ventajas que tienen los SDIh es su capacidad para registrar todos los eventos que ocurren en el sistema en las bitácoras del sistema, así queda rastro de cualquier actividad maliciosa o intento de intrusión. En estas bitácoras se pueden encontrar patrones de intentos de acceso no autorizado, de correo electrónico para indagar la existencia de correo SPAM, accesos al servidor web, entre otros. Además, se pueden obtener patrones de las sesiones que realizan cotidianamente cada uno de los usuarios del sistema, así es posible detectar conexiones de usuarios legítimos, pero a horarios o desde direcciones IP totalmente inconsistentes con los patrones registrados para estos usuarios, por lo que se puede suponer que un usuario malicioso está usurpando la identidad de un usuario legítimo. Por ejemplo, las siguientes dos líneas muestran dos registros de la misma actividad en diferentes bitácoras; el primero es del archivo *messages* y el segundo del archivo *secure*:

```
Nov 9 17:55:26 delfos sshd(pam_unix)[32762]: authentication failure; logname= uid=0 euid=0
tty=NODEVssh ruser= rhost=148.84.37.17
```

```
Nov 9 17:55:26 delfos sshd[32762]: Illegal user test from 148.84.37.17
```

El primer registro indica una falla en la autenticación a un usuario a través del protocolo ssh el 9 de noviembre a las 17:55. El segundo registro es de la bitácora *secure* e indica que un usuario con el nombre ilegítimo "test" intentó acceder al sistema el 9 de noviembre a las 17:55. Este registro puede deberse a que el usuario se equivocó de nombre de usuario, o a alguna explicación lógica, pero si encontramos este registro numerosas ocasiones en un periodo corto de tiempo podemos pensar que se trata de un usuario malicioso que intenta acceder al sistema de manera ilícita.

Cada vez hay un mayor número de SDIh disponibles para garantizar la seguridad de los huéspedes de una red. Entre ellos se puede mencionar el Tripwire [29] que es un *software* comercial de aseguramiento de integridad de los datos, supervisa la consistencia de archivos y directorios críticos del sistema identificando todos los cambios hechos a ellos.

Esto lo hace mediante un método automatizado de verificación que se ejecuta a intervalos regulares. Si Tripwire detecta que uno de los archivos supervisados ha sido cambiado, lo notifica al administrador del sistema vía *email*. Debido a que Tripwire puede fácilmente identificar los archivos que son modificados, agregados o eliminados, se agiliza el proceso de recuperación luego de una entrada forzada pues mantiene el número de archivos que deben ser restaurados a un mínimo.

OSSEC es un SDIh de dominio público [6] que tiene la capacidad de realizar análisis sobre la información contenida en las bitácoras e inspeccionar su integridad. También cuenta con mecanismos que esconden procesos y archivos críticos que puedan resultar llamativos para un intruso. Las alertas las envía en tiempo real y puede responder de manera activa ante un ataque detectado. Es compatible con la mayoría de los sistemas operativos, incluyendo Linux, MacOS, Solaris y Windows.

2.4.2. Sistemas de detección de intrusos basados en red (SDIr)

Los SDIr son aquellos capaces de detectar ataques contra diferentes sistemas de una misma red (en concreto, de un mismo dominio de colisión), aunque generalmente se ejecuten en uno solo de los huéspedes de esa red. Para lograr su objetivo, al menos una de las interfaces de red de esta máquina sensor trabaja en modo promiscuo, capturando todos los paquetes que pasan por él y almacenando la información de estos paquetes en un repositorio para su posterior análisis en busca de patrones indicativos de un ataque como se puede apreciar en la Figura 2.4.

Un detector de intrusos basado en red puede estar basado en la detección de anomalías, igual que lo puede estar un SDIh. Sin embargo, una intrusión generará probablemente comportamientos anormales susceptibles de ser detectados y eliminados, pero esto se logra frecuentemente cuando la intrusión se encuentre en un estado avanzado. Por ejemplo, en un ataque de negación de servicio en contra de una red se produce un tráfico excesivo entre el sistema atacante y el sistema atacado y la única forma en que un SDIr basado en anomalías lo detecte es que registre esta cantidad de tráfico para compararlo con su modelo de tráfico esperado; al ver que hay una desviación considerable del tráfico actual con el tráfico esperado lanzará algún tipo de respuesta, pero para cuando esto ocurra el ataque ya estará causando problemas en la red. La prioridad de estos sistemas es detectar el ataque lo antes posible para que cause el menor daño al sistema.

A la fecha, debido al alto crecimiento de las redes de comunicaciones, se encuentran disponibles varios SDIr para brindar un alto grado de confianza en cuanto a la protección de la seguridad de dichas redes. Snort [47] es un *sniffer* de paquetes y un detector de intrusos basado en red (supervisa todo un dominio de colisión) de dominio público. Snort utiliza un lenguajes regido por reglas, que combina las ventajas del protocolo de firmas y de los métodos basados en la detección de anomalías.

Un sistema basado en el análisis de reglas posee un conjunto de reglas predefinidas por el administrador, o creadas automáticamente por el sistema, o una combinación de ambas posibilidades. Las reglas generalmente son definidas de manera escrita en algún editor de reglas del propio sistema y el administrador debe poner especial cuidado a la

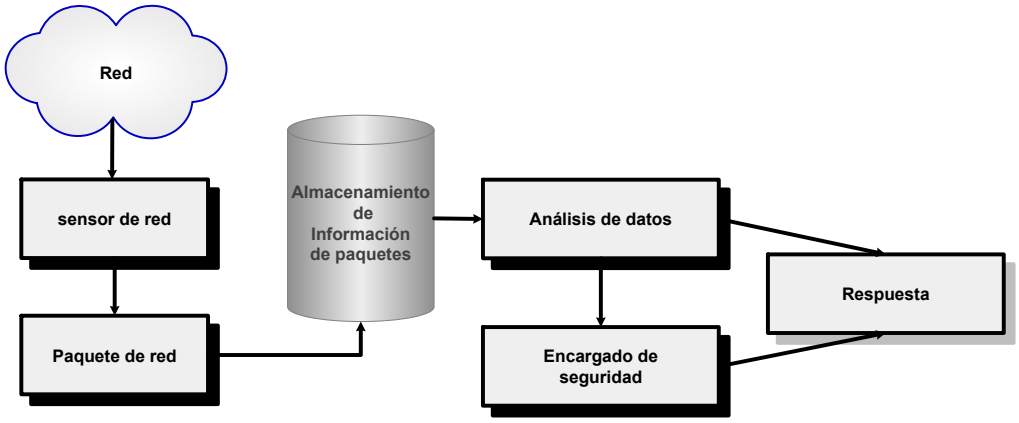


Figura 2.4: Organización de un sistema de detección de intrusos basado en red

hora de definirlas, ya que de eso puede depender en menor o mayor medida el grado de efectividad del sistema.

La característica más apreciada de Snort, además de su funcionalidad, es su subsistema flexible de firmas de ataques. Snort tiene una base de datos de ataques que se está actualizando constantemente y a la cual se puede añadir o actualizar a través de Internet. Los usuarios pueden crear firmas basadas en las características de los nuevos ataques de red y enviarlas a la lista de correo de firmas de Snort, para que así todos los usuarios de Snort se puedan beneficiar.

Un SDIr comercial es el RealSecure [14] el cual permite detectar y avisar automáticamente al usuario de las brechas de seguridad y los ataques internos en la red antes de que el sistema se vea afectado. RealSecure se compone de tres elementos: la máquina RealSecure, el agente RealSecure y el moderador RealSecure. La máquina RealSecure permite la identificación en tiempo real, envía alarmas predefinidas por los usuarios y responde a los ataques en cuanto son detectados. El agente RealSecure es un complemento basado en el huésped de la máquina RealSecure, éste reacciona e impide las instrusiones, también manda alarmas , correos electrónicos de aviso y ejecuta algunas acciones predeterminadas por los usuarios. Estos dos elementos son configurados por el moderador RealSecure.

Con los SDIr podemos detectar ataques tales como ataques de negación de servicio, exploradores de puertos o intentos de entrar en una computadora, analizando el tráfico en la red en tiempo real. Los SDIr no sólo vigilan el tráfico entrante, sino también el saliente o el tráfico local, ya que algunos ataques podrían ser iniciados desde el propio sistema protegido

La Tabla 2.1 resalta las características que diferencian a los SDIh y a los SDIr [44]. La principal diferencia depende del equipo al que cada SDI vigila; los SDIh vigilan un sólo equipo mientras que los SDIr vigilan todo un segmento de red. Por otro lado, los SDIh operan con eventos a nivel de sistema operativo y sus aplicaciones y los SDIr operan en varias capas de la red; la ventaja que tiene los SDIh sobre los SDIr es que no requieren equipo extra para ejecutarse sino que utilizan los mismos recursos del sistema que

protegen, en cambio los SDIr necesitan equipo adicional y especializado con capacidades suficientes para analizar el tráfico que circula a través de la red.

2.4.3. Sistemas de detección de intrusos híbridos

Es importante mencionar que la clasificación de los SDI presentada en la sección 2.4 no es necesariamente excluyente. Ambos tipos de SDI, tanto los SDIh como los SDIr se pueden complementar e implementar simultáneamente para obtener un alto nivel de seguridad. Es posible tener una arquitectura híbrida para cumplir con este propósito; en estos casos, el agente híbrido (SDIh&r) se localiza dentro del equipo que protege con todas las características de un SDIh, pero incluye además la capacidad de revisar tráfico de red desde o hacia el equipo que vigila. Los SDI híbridos pueden analizar tráfico de red, pero sólo aquel dirigido al equipo que protegen.

Un SDIh&r, es aquel sistema de detección constituido por sensores en cada huésped que permiten la detección local de los sistemas y un sensor en cada segmento de red a vigilar. Un SDIh&r es capaz de agregar eventos generados por diferentes fuentes de información, proporcionando así una imagen más amplia y detallada de las actividades maliciosas en un determinado entorno [31].

Sus componentes principales son:

- Agentes de huésped (supervisan actividad del huésped).
- Agentes de red (supervisan la actividad en el segmento de red).
- Transceptores (comunicación).
- Consola de eventos (interfaz con el operador).

Los agentes de huésped son entidades independientes que supervisan actividad interesante en el huésped en el que se encuentran instalados. Es ejecutado de forma continua o bajo demanda, y se comunica con otros agentes. Puede haber más de un agente de huésped instalado en un huésped.

Los agentes de red analizan la actividad de un segmento de red y se comportan como un SDIr común.

Generalmente los agentes de huésped y los agentes de red tienen diferentes propósitos e incluso, diferentes lenguajes.

Los transceptores transmiten y reciben información de los agentes de huésped y de red, y debe haber un transceptor en cada máquina donde haya algún agente.

La consola de eventos es un elemento que funciona como interfaz entre el SDIh&r y el administrador; es aquí donde se presentan todos los resultados de los análisis del SDI. Aunque la consola de eventos, también conocida como interfaz de usuario no forma parte de la arquitectura de implementación de un SDI, es un elemento muy importante del sistema, puesto que es quien permite y facilita la toma de decisiones al administrador o encargado ante cualquier incidente , además le permite verificar los informes o estadísticas de los análisis del sistema.

Característica	Basado en red (SDIr)	Basado en huésped (SDIh)
Propósito	▪ Protección simultánea de varios sistemas de cómputo. ▪ Protege sistemas críticos y sistemas de propósito general (gracias a su visibilidad).	▪ Protección específica para sistemas de cómputo importantes. ▪ Protege servidores y equipos críticos.
Visibilidad	▪ Todo un segmento de red; protege equipos dentro del segmento vigilado. ▪ Si el *switch* o ruteador al que se conecta el SDIr está en la frontera con Internet, estos sistemas podrán ver ataques desde Internet hacia toda la organización y viceversa.	▪ Un único equipo; este sistema sólo puede identificar eventos en el equipo donde se encuentra instalado.
Nivel de operación	▪ Diferentes capas de red	▪ Eventos a nivel del Sistema Operativo y aplicativo.
Recursos necesarios	▪ Requiere de un equipo adicional a los sistemas protegidos. ▪ Equipo adicional con capacidad suficiente para supervisar el tráfico de red del segmento vigilado.	▪ Utiliza los mismos recursos del sistema que protege. ▪ Sistema con suficiente capacidad para poder realizar sus actividades y las del SDI, sin afectar su desempeño.

Tabla 2.1: Tabla comparativa de los SDIh y los SDIr

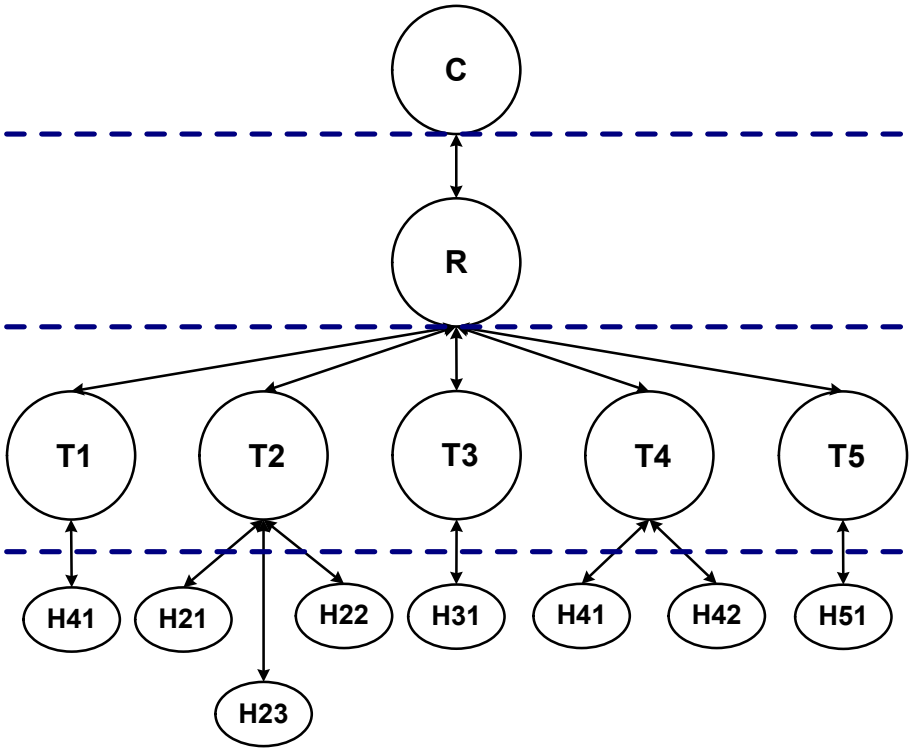


Figura 2.5: Sistema de detección de intrusos híbrido

En la Figura 2.5 se muestra la estructura de un SDIh&r en el cual se tiene un agente de red (R, aunque puede haber más de uno), varios agentes de huésped (H) y varios transceptores (T) que comunican a los agentes. Como se ha dicho cada huésped puede tener más de un agente de huésped, en este ejemplo cada huésped tiene varios agentes de huésped y tiene una estructura jerárquica aunque pueden plantearse otro tipo de esquemas.

Aunque la idea de combinar los SDI para huésped y los SDI para red es muy atractiva, muy pocos SDIh&r se encuentran disponibles. Uno de ellos es el Prelude [24] que es un SDIh&r distribuido de dominio público que utiliza la información de cada uno de los huéspedes que protege y del tráfico circulante a través de la red para detectar anomalías o ataques, todo esto en tiempo real. Así, genera respuestas con base en los ataques, ya sean a nivel de red o del sistema. Prelude está compuesto de cinco módulos o componentes: un primer módulo que se encarga de enlazar los cuatro módulos restantes; el segundo componente son lo sensores que se encargan de detectar anomalías o intentos de intrusión en la red o en un huésped; el tercer módulo es el de los moderadores que son los procesadores centralizados de datos, éstos reciben alertas de los sensores y pueden compartirlas con otros moderadores; el cuarto componente son los agentes medidores que reciben de los moderadores conocimiento de alguna particular anomalía y lanzan una acción para detenerla; y finalmente, consta de una interfaz de usuario.

Prelude está diseñado para dar a los usuarios una solución robusta para la seguridad de una red, con la capacidad de supervisar a la red y a los huéspedes que la componen en búsqueda de actividades anómalas. Sin embargo, también presenta algunas desventajas; una ellas es que, debido a la cantidad de información que se requiere manipular para su buen funcionamiento necesita de equipos especializados con capacidades suficientes para este manejo, añadiendo carga de procesamiento extra al sistema que vigila.

Otro SDIh&r es llamado Sistema de Detección de Intrusos Distribuido (DIDS, por sus siglas en inglés), el cual fue desarrollado en la universidad de California [56]. DIDS combina la supervisión distribuida y la reducción de datos con análisis centralizado de datos para supervisar una red heterogénea de computadoras. y generaliza la idea de supervisar un sólo huésped para supervisar múltiples huéspedes conectados vía red, así como la red misma. Entre los componentes de este sistema distribuido se encuentra un monitor de huésped por cada huésped, un monitor de red por cada segmento de red de la red supervisada. La información recogida por estos dos componentes distribuidos es transportada a una locación central llamada director, el cuál hace el manejo de las comunicaciones y muestra los resultados a través de una interfaz.

Existen varios escenarios que pueden comprometer la seguridad de una red que serían detectados de manera muy sencilla por un SDIh&r, no así con los SDI de huésped o de red. Uno muy simple puede ser cuando un atacante intenta ganar acceso a una máquina que presente ciertas vulnerabilidades. El intruso generalmente intenta acceder con cuentas comunes al sistema y mediante métodos de fuerza bruta busca encontrar la contraseña correspondiente a una de las cuentas válidas; este ataque lo puede lanzar de manera distribuida, es decir, lanzar el ataque a más de una computadora de la red. Si el atacante llegase a obtener alguna de las contraseñas de alguna cuenta legítima, ganaría acceso al sistema con los privilegios de dicha cuenta, y ya dentro de él tal vez intente hacer un ataque más serio.

2.5. Técnicas de detección de los SDI

Existen diversas tecnologías utilizadas para la detección de intrusos [39, 58, 48, 37, 46]:

- **Detección de patrones.** Detectan ciertas actividades del sistema operativo o cadenas de bits en tráfico de red que indican actividad que pertenece o se relaciona con ataques conocidos.

  En cuanto sea mayor el número de patrones reconocidos como una anomalía o un ataque potencial, mayor será la capacidad del SDI para detectar ataques. La mayor desventaja de estos sistemas es que los patrones de los ataques que ya son conocidos están almacenados en algún lugar en el sistema, y no serán capaces de detectar futuras intrusiones que sean desconocidas, es decir que no correspondan con alguno de los patrones almacenados por el sistema.

  Por ejemplo, el sistema STAT [34] basa su funcionamiento en la detección de patrones, éste codifica y hace corresponder la secuencia de acciones de un evento en una

firma, por ejemplo, cambiar el dueño de un archivo y cuando detecta un evento que no corresponde con alguna de las firmas almacenadas lo considera como un ataque.

- **Técnicas de inteligencia artificial.** Identifican patrones de ataque o actividad anómala por medio de ciertos indicadores para un perfil . Este tipo de tecnologías construyen modelos dinámicos para los perfiles de uso normal de un sistema, los cuales se van retroalimentando con los diferentes tipos de actividad —maliciosa o no— que se presenten en el sistema y así obtener perfiles que se adecúen al comportamiento habitual del sistema. Por ejemplo, a través de redes neuronales entrenadas para aceptar actividad cotidiana y disparar una alarma cuando hay una desviación considerable en los indicadores; estas redes comienzan con un grado de experiencia bajo ganado con algunos ejemplos de ataques seleccionados con anterioridad y van ganando experiencia analizando el comportamiento del sistema a través del tiempo, obteniendo así un nivel cada vez más alto de entrenamiento [10].

  La red MLFF (Multilayered Feed Forward Network) [55] es usada como una aproximación universal para clasificar, reconocer y generalizar los datos de entrada tomados de un grupo de datos de ejemplo; estos datos son los que brindan el aprendizaje a la red neuronal. Así se modela una red que es capaz de encapsular el comportamiento de un usuario y que es tan flexible como para añadir constantemente grandes, pequeños y tal vez imperceptibles cambios del comportamiento de los usuarios al modelo actual.

- **Métodos estadísticos.** Identifican desviaciones numéricas en ciertos indicadores, basándose en perfiles previamente definidos de lo que constituye un ataque o actividad legítima.

  Un modelo estadístico es aquel que se basa en un análisis estadístico del comportamiento previo de un evento para poder determinar de manera aproximada el valor que tendrá en futuras repeticiones. Los modelos estadísticos usan una gran cantidad de datos de comportamiento obtenidos previamente.

  Los métodos estadísticos son muy usados en el análisis de tráfico de una red local [18], por ejemplo; con estos métodos se puede modelar de manera estadística el comportamiento normal de la red recolectando, promediando, y suavizando una gran cantidad de datos que son resultado del muestreo de paquetes de varios días de actividad. Estos modelos sirven para determinar el nivel de desempeño cotidiano de la red, y cuando se presente un comportamiento que tenga un error mayor al permitido con respecto al desempeño cotidiano del modelo se considera como una anomalía.

Como se puede ver, existen dos enfoques en la aplicación de tecnologías de detección de intrusos:

- Detectar aquello que es plenamente conocido (ataques o actividad legítima)
- Detectar aquello que es desconocido (actividad anómala, generalmente asociada con ataques)

Mientras que el primer enfoque da menos falsas alarmas, también está más limitado (no detecta ataques que no han sido definidos previamente), el segundo enfoque tiene una mayor capacidad de detección (ataques desconocidos por ejemplo), pero generan un número mucho mayor de falsas alarmas.

En cualquiera de los casos nunca es posible eliminar por completo el número de falsas alarmas dado que la capacidad de detectar un cierto número de ataques es proporcional en cierta medida al número de falsas alarmas que se detectan (entre más ataques se puedan detectar, mayor número de falsas alarmas se presentan).

2.6. Discusión

Hoy en día se encuentran disponibles diferentes SDI, algunos comerciales, otros de manipulación libre, y algunos otros son investigaciones de ciertas universidades. En la tabla 2.2 se resumen las características de algunos SDI.

SDI	Análisis	Sensor	Ejecución	Respuesta	Arquitectura	Distribución
Tripewire [29]	A	Huésped	Periódico	Pasivas	Centralizado	Comercial
OSSEC [6]	A	Huésped	T. Real	Activas	Distribuido	Libre
RealSecure [14]	UI	Red	T. Real	Activas	Distribuido	Comercial
Snort [47]	A&UI	Red	T. Real	Activas	Centralizado	Libre
Prelude [24]	UI	Híbrido	T. Real	Activas	Distribuido	Libre
DIDS [56]	UI	Híbrido	T. Real	Activas	Centralizado	No disponible

Tabla 2.2: SDI existentes en el mercado. Donde: A - Basado en Anomalías. UI - Basado en Usos Indebidos. A&UI - Basado en Anomalías y en Usos Indebidos.

Tripwire y OSSEC son SDIh que hacen un análisis basado en anomalías. Tripwire se ejecuta en intervalos periódicos de tiempo y OSSEC es un SDI en tiempo real, es decir, va procesando la información conforme esta va ingresando en el sistema. Al detectar un ataque, Tripwire envía alertas al administrador vía correo electrónico mientras que OSSEC está capacitado para ejecutar alguna acción ante una anomalía o ataque detectado. Tripwire puede trabajar en un ambiente distribuido, es decir, que posiblemente para su buen funcionamiento haya más de un sensor ubicado en puntos estratégicos de la red, mientras que OSSEC tiene un sólo sensor, Tripwire no es de dominio público y OSSEC sí.

RealSecure y Snort son SDIr. RealSecure hace una tipo de análisis basado en usos indebidos, mientras Snort hace una combinación del análisis basado en anomalías y el análisis basado en usos indebidos. Para el análisis basado en anomalías Snort cuenta con un conjunto de reglas que representan el perfil adecuado del sistema y para el análisis basado en usos indebidos usa un sistema de firmas de ataques que son patrones de ataques definidos por los mismos usuarios de Snort. Los dos hacen detecciones en tiempo real y pueden responder activamente ante una anomalía o ataque detectado; RealSecure trabaja en un ambiente distribuido, mientras que Snort cuenta con un sólo sensor que es

el que supervisa el segmento de red al que esté protegiendo. RealSecure es de distribución comercial y Snort completamente libre.

Prelude y DIDS son SDIh&r. Los dos son SDI que realizan análisis basado en usos indebidos, son SDI de tiempo real que tienen la capacidad de responder activamente ante un ataque presentado. Prelude puede trabajar en ambientes distribuidos, mientras que DIDS sólo puede hacerlo en ambientes centralizados. Prelude es de dominio público y DIDS es un prototipo de la Universidad de California, por lo cual no se encuentra disponible. Los SDIh&r existentes presentan los problemas comunes a este tipo de sistemas. En primer lugar problemas de comunicación, debido a que todos los elementos del sistema deben hablar el mismo lenguaje. Otra desventaja son los requerimientos de *hardware* y de *software* para este tipo de sistemas.

2.7. Resumen

En este capítulo se hizo una revisión breve de seguridad informática, en redes y en sistemas de información. Se revisaron los posibles escenarios a los que una red se encuentra vulnerable. Los ataques más comunes son los ataques remotos; entre éstos se pueden mencionar las exploraciones de puertos y direcciones IP, ataques de negación de servicio, el *spoofing* y los ataques a las aplicaciones que intentan poner en peligro la estabilidad de un sistema de cómputo.

Además se describieron las diferentes herramientas usadas para garantizar la seguridad de un sistema; como cortafuegos, listas de control de acceso, antivirus y entre ellos uno muy importante y popular en la actualidad: los Sistemas de Detección de Intrusos.

Se revisaron las diferentes clasificaciones de los SDI en cuanto a su modo de análisis, el tipo de información de la que se alimentan y los tipos de respuesta que pueden ejecutar cuando detectan una actividad sospechosa o anómala.

También se hizo una descripción de cada una de las tecnologías de detección con las que los diferentes SDI son implementados; como la detección de patrones, los métodos de inteligencia artificial y los métodos estadísticos.

Se analizaron dos SDI por cada tipo de sensor —o tipo de información que usan como fuente para su funcionamiento—, y finalmente se realizó una discusión con las diferentes características con las que cuenta cada SDI revisado.

Los SDI son necesarios para garantizar la seguridad de una red de comunicaciones, ya que aunque existen métodos de prevención muy eficientes como los cortafuegos o las listas de control de acceso, nunca se debe estar seguro que alguna de estas herramientas será infalible ante cualquier usuario malicioso. Sin embargo si hay un SDI instalado, el usuario podrá burlar los métodos de prevención pero no estará a salvo de ser detectado y neutralizado por un SDI.

Capítulo 3

Sistema Híbrido de Monitoreo y Detección de Intrusos

Hoy en día existen varios SDI disponibles en el mercado, ya sean basados en red o basados en huésped. Como ya se mencionó en el capítulo anterior un SDIh sólo puede detectar ataques relacionados con el servidor donde se encuentra instalado, y un SDIr puede detectar ataques dirigidos a varios de los servidores que pertenecen a un mismo segmento de red.

Lo que se propone en este trabajo de tesis es diseñar e implementar un SDI que explote las mejores cualidades que brindan los dos tipos de SDI —red y huésped— para así combinar sus características y tener un sistema capaz de poder supervisar simultáneamente huéspedes y redes ofreciendo mayor calidad y exactitud en las detecciones. Este SDI generaliza la idea de detectar ataques dirigidos a uno de los huéspedes o a varios de ellos para supervisar un segmento de red completo.

Por ejemplo, imagine el siguiente escenario: un usuario suficientemente malicioso lanza un simple ataque de exploración de puertos a varias máquinas de la misma red en busca de vulnerabilidades que pueden ser aprovechadas para conseguir acceso ilegítimo a estas máquinas con privilegios de administrador. Cuando se detecte este ataque en alguna máquina en específico, es conveniente alertar a la máquina involucrada en el ataque pero además es importante compartir la información de este ataque con el resto de las máquinas que componen la red supervisada, ya que cuando se dirige un ataque a una máquina, muy probablemente el atacante intentará incluir en su ataque a otros servidores de la misma red (ver Figura 3.1), así que si se detecta al intruso desarrollando un ataque a un servidor se puede prevenir a otros servidores acerca de un ataque potencial de la misma fuente, lo cual es una tarea fácil para un SDI híbrido ya que almacena la información de todos los ataques registrados, tanto en los huéspedes como en la red.

Ahora imagine la siguiente situación: un atacante malicioso A lanza un simple ataque de exploración de puertos hacia una red grande. Además A cuenta con un programa malicioso que logra ganar acceso a una servidor ftp con privilegios de administrador; A explora los servidores buscando algún servidor ftp vulnerable y encuentra uno. Después A lanza a ejecución su programa malicioso y éste es ejecutado con éxito, esto quiere decir

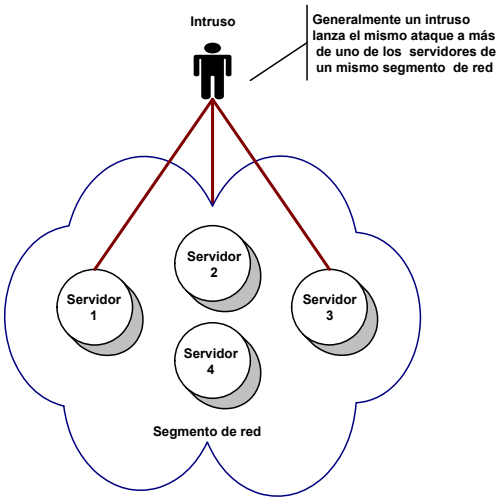


Figura 3.1: Situación en la que un intruso lanza el mismo ataque a varios servidores de una red

que ahora ahora A tiene acceso a un servidor ftp con privilegios de administrador.

En este ejemplo, un SDIr debería detectar el programa malicioso —si se encuentra configurado correctamente— y lanzar una alerta al administrador o registrar el evento.

Suponga ahora que A es un poco más astuto y quiere ganar el reconocimiento de los demás como un verdadero atacante peligroso. Ahora A es un usuario convencional de una máquina que sabe que es vulnerable y ejecuta un programa malicioso y logra ganar acceso como administrador. A cuidadosamente se dirige a las bitácoras y remueve cualquier entrada que exponga su acceso ilegítimo y entonces baja otro programa malicioso diseñado para ganar acceso como administrador a un servidor ftp a través del famoso ataque conocido como "desbordamiento de memoria", en el cual se copia una cantidad de datos que excede el área de memoria asignada sobrescribiendo otras zonas de memoria prohibidas. A compila su programa con éxito y simplemente levanta de nuevo el servicio ftpd de toda la red. Ahora A tiene acceso como administrador al servidor ftp, y ningún SDIr podrá definir su trazo original ya que como administrador A borró toda evidencia de su acceso. En la Figura 3.2 se muestra los pasos que A siguió para lograr hacer daño a su objetivo y no ser detectado: primero explora la red en busca de servidores vulnerables, cuando encuentra uno ingresa y ejecuta un programa malicioso y finalmente se dirige a las bitácoras para borrar el rastro de su acceso.

Con un SDI híbrido que actúe en tiempo real, queda registrado el momento en que A entra al sistema y toda su actividad. Cualquier intento por ganar acceso como administrador quedaría también registrado —debido a la habilidad que tienen los sistemas Linux de registrar los eventos de dicho sistema—. Incluso si las bitácoras del sistema fuesen corrompidas, la evidencia es enviada al SDI híbrido en tiempo real y fácilmente se podrá frenar el ataque. El SDIr del SDI híbrido detectará la existencia de un programa malicioso de A y registrará información suficiente acerca de la actividad del atacante que quedará disponible para su análisis permitiendo alertar al administrador de la red. Y el

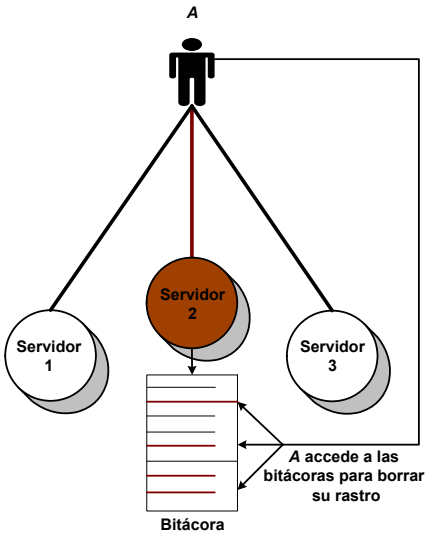


Figura 3.2: Situación en la que un intruso busca su objetivo, lanza el ataque y finalmente borra su rastro

SDIh generará el trazo del atacante lo cual es muy interesante desde el punto de vista de la seguridad.

Así pues, la idea de este trabajo al implementar un SDI híbrido es que el sistema resultante sea capaz de tomar los beneficios de combinar los trazos de alguna actividad maliciosa registrados en las bitácoras del sistema y la información acerca del ataque registrado proporcionada por la red desarrollando correlación automática para mejorar la calidad de la detección.

En este capítulo se presentan los detalles de la implementación del SDI realizado. Primero se dará una idea general del sistema y en las siguientes secciones se detallarán cada uno de los módulos que componen dicho sistema.

3.1. Descripción general del sistema de monitoreo y detección de intrusos

El sistema de monitoreo y detección de intrusos supervisa paralelamente huéspedes y segmentos de red. Los huéspedes supervisados pertenecen a un mismo segmento de red y se busca detectar ataques o actividades maliciosas dirigidas a un huésped en específico. Los segmentos de red se supervisan para obtener información general y amplia de los ataques detectados en cada huésped.

La arquitectura de este sistema consta básicamente de tres módulos como muestra la Figura 3.4: uno o varios agentes de huésped -SDIh-, un agente de red -SDIr- y un transceptor —módulo de interconexión—.

El primer módulo, el agente de huésped supervisa actividades del sistema operativo y

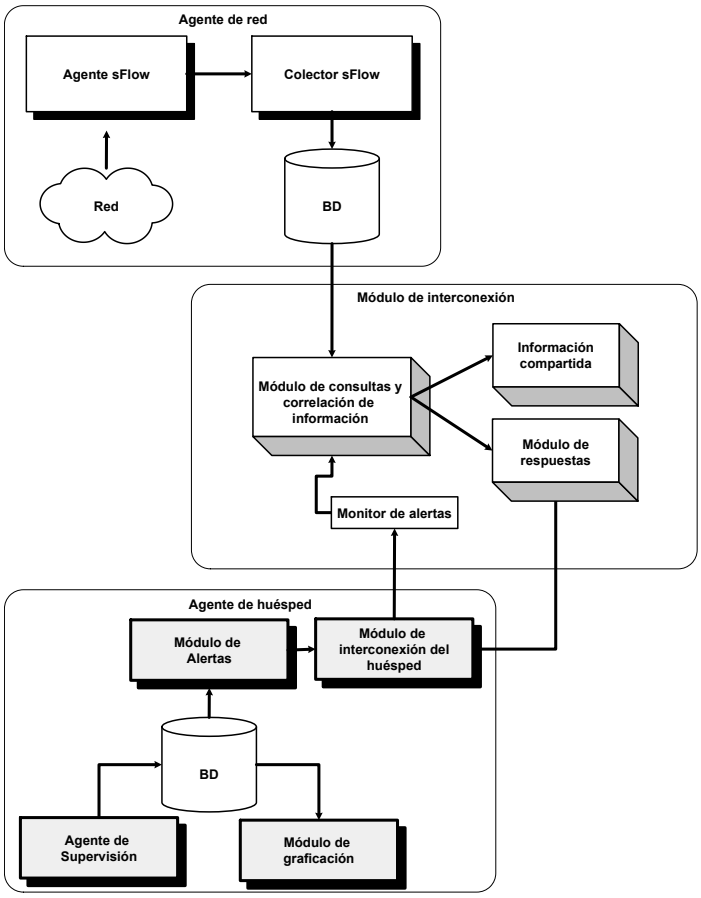


Figura 3.3: Arquitectura del sistema de monitoreo y detección de intrusos

de las aplicaciones que se ejecutan sobre éste; toda esta información se encuentra registrada en las bitácoras del sistema y se trata de obtener patrones de uso normal, los cuales se representan por valores numéricos -umbrales- y las actividades que no coincidan con estos patrones serán consideradas como sospechosas o peligrosas. A su vez, este módulo consta de cuatro módulos: el agente de supervisión, el módulo de graficación, el módulo de alertas y el módulo de interconexión.

El segundo módulo es el agente de red y se encarga de supervisar todo un segmento de red para encontrar ataques que puedan ser detectados mediante patrones de uso normal obtenidos de los paquetes de entrada y salida de la red supervisada. Este agente cuenta con tres módulos: un agente sFlow y un colector sFlow para el muestreo de paquetes y la recolección de los datos del tráfico respectivamente, además de una base de datos para almacenar la información recolectada.

Este agente se hizo con la ayuda del protocolo sFlow [3] para el muestreo de paquetes y la recolección de los datos del tráfico de paquetes; a su vez este agente cuenta con tres módulos: un agente sFlow, un colector y una base de datos para almacenar la información recolectada.

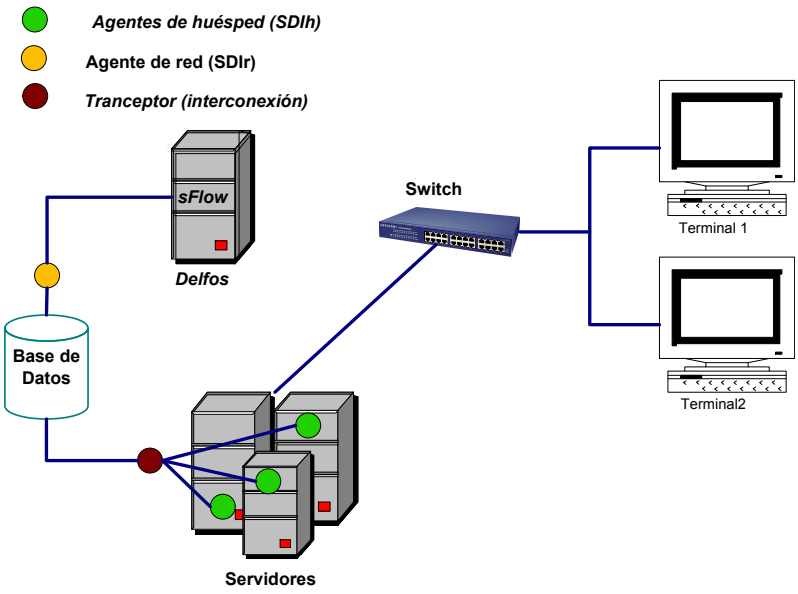


Figura 3.4: Funcionamiento del sistema de monitoreo y detección de intrusos

El tercer módulo es el transceptor o módulo de interconexión; este módulo es el que le da la propiedad de ser híbrido al sistema, ya que se encarga de comunicar a los agentes de huésped y de red mencionados para así tener acceso a la información de cada uno de estos módulos que será útil para poder hacer una mayor cantidad de detecciones de intrusiones y ataques. Cuando un ataque es detectado en cualquiera de los dos agentes —huésped o red—, el módulo de interconexión correlaciona la información de éstos para determinar la magnitud del ataque y alertar a los huéspedes involucrados, pero también hace disponible a todos los huéspedes de la red la información de los ataques registrados más recientemente para que los administradores de estos sistemas tomen las medidas de prevención convenientes.

En la Figura 3.4 se muestra el funcionamiento del sistema en donde existe un agente por cada servidor y un sólo agente de red; puede ser instalado un agente de huésped por cada una de las máquinas más importantes de la red como servidores, también se puede instalar un agente de red por cada segmento de red vigilado y existe un módulo de interconexión que modera a todos los agentes. El módulo de interconexión como ya se explicó es el que se encarga de entablar la comunicación entre el SDIh y el SDIr para obtener un SDI híbrido.

3.2. Agente de huésped (SDIh)

El módulo de SDIh tiene como objetivo principal vigilar algunas de las actividades más importantes del sistema y el buen funcionamiento de las aplicaciones que en él se ejecutan. Para hacerlo, toma como información de entrada las bitácoras del sistema que es

donde se encuentran registrados los eventos del sistema operativo y sus aplicaciones. Con base en esta información, analiza ciertos patrones de uso y determina si en un periodo de tiempo dado existe un ataque o actividad sospechosa hacia el huésped que vigila.

Como se muestra en la Figura 3.5 el SDIh tiene cinco componentes principales: un agente de supervisión, una base de datos que se nombrará en lo sucesivo como "base de datos de huésped", un módulo de graficación, un módulo de alertas y el módulo de interconexión del "huésped".

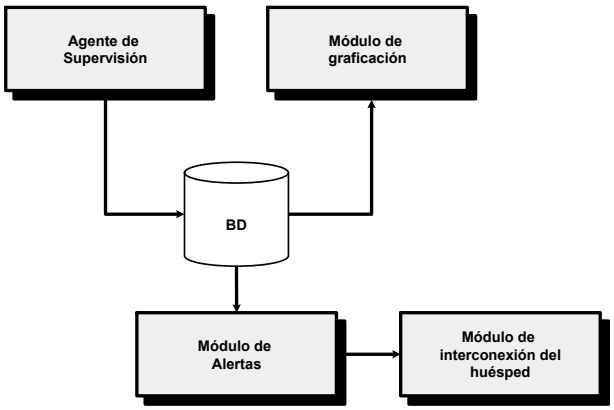


Figura 3.5: Componentes del agente de huésped

El agente de supervisión se encarga de procesar las bitácoras a intervalos de tiempo regulares y obtener el número de ocurrencias que tengan los eventos registrados en las bitácoras. Esta información es recolectada y almacenada en la base de datos de huésped, la cual contiene toda la información del agente de huésped a lo largo del tiempo. El módulo de graficación toma la información de la base de datos de huésped y construye las gráficas correspondientes que reflejan el comportamiento de cada uno de los eventos registrados a lo largo del tiempo; conforme la base de datos se actualiza, las gráficas son actualizadas también. El módulo de alertas analiza la información recolectada en la base de datos de huésped para buscar ocurrencias que indiquen algún tipo de actividad sospechosa o maliciosa y finalmente si se detecta alguna anomalía se da parte al módulo de interconexión del sistema a través de un módulo de interconexión de huésped, que como se explicará más a detalle en la subsección 3.2.4, es diferente al módulo de interconexión del sistema.

3.2.1. Agente de supervisión

El agente de supervisión hace el análisis y procesamiento periódico sobre los archivos de bitácoras del sistema para obtener variables que se utilizan para definir patrones de uso normal. Llamaremos variable a cada uno de los tipos de información que se analizan de los archivos de bitácoras. Este análisis se hace a intervalos de tiempo regulares a los que se nombran periodos de recolección. Un periodo de recolección es el tiempo en que se hará un

conteo de actividades y terminado este tiempo se reiniciará el conteo para el siguiente periodo. La información recolectada es almacenada en una base de datos de huésped y usada por el módulo de graficación para realizar gráficas históricas. El módulo de alertas se encarga de correlacionar la información recolectada en busca de datos —relacionados a las variables— con desviaciones numéricas con respecto al valor de los umbrales. Un umbral es un valor mínimo que indicará algún tipo de actividad maliciosa; si se encuentra alguna amenaza de ataque en el huésped se comunica al módulo de interconexión de todo el sistema a través del módulo de interconexión del agente de huésped.

El agente de supervisión se encuentra dividido en tres partes: el procesamiento de las bitácoras, la extracción de datos y el almacenamiento de estos datos en una base de datos, como se muestra en la Figura 3.6.

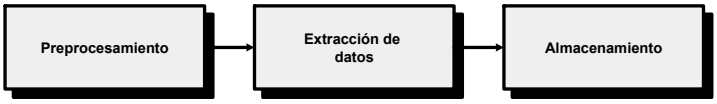


Figura 3.6: Componentes del agente de supervisión

Preprocesamiento

Para administrar de manera efectiva un servidor, es necesario tener registros de la actividad y el rendimiento del servidor, así como de cualquier problema que haya podido ocurrir durante su operación.

Los archivos de bitácoras son los que registran directamente las estadísticas de acceso dentro del propio. En estos archivos se registra información muy valiosa acerca de eventos relacionados con el sistema que la genera, desde las horas de acceso de cada usuario al sistema hasta las páginas web más frecuentemente visitadas, pasando por los intentos fallidos de conexión, los programas ejecutados o incluso el tiempo de CPU que cada usuario consume.

Otra ventaja es que no sólo registran los eventos, sino que proporcionan rastro de toda actividad. Cada actividad registrada viene acompañada de la fecha y hora exacta de su ocurrencia, direcciones IP origen y destino, dirección IP que genera la bitácora, usuarios, errores, entre otros.

Las bitácoras también presentan algunos inconvenientes; uno de ellos es que los sistemas generan una gran cantidad de información y así su revisión diaria se vuelve una tarea tediosa y es una actividad que pocos administradores llevan a cabo.

Por esta razón, el módulo de preprocesamiento se encarga de leer y depurar cada uno de los registros contenidos en las bitácoras extrayendo sólo la información útil y eliminando el resto de la información, esto con el fin de hacer más claro el análisis del comportamiento de cada una de las variables en observación.

Para este trabajo se utilizaron las bitácoras de un sistema Linux, éstas bitácoras son archivos de texto, entre ellos los siguientes:

- Archivo *secure*. Contiene información sobre los accesos de los usuarios del sistema, registra los intentos de acceso fallidos y los accesos logrados.

- Archivo *maillog*. Supervisa el servicio de correo electrónico, registra cuantos correos se envían o se reciben, el tamaño de los correos entre otros datos.

- Archivo *message*. Almacena datos que informan de ciertos programas, mensajes de baja o media prioridad destinados más a informar que a avisar de sucesos importantes. En éste se encuentra información relativa al arranque de la máquina; también almacena información de las sesiones abiertas por parte de los usuarios del sistema.

- Archivo *acces_log*. Proporciona información de los accesos a páginas del servidor web.

- Archivo *error_log*. Presenta informes de todos los errores que provienen del servidor web.

Al realizar un análisis para ver los tipos de patrones que podrían extraerse de las bitácoras ya mencionadas, se obtuvo una lista de variables que pueden ayudar en un dado caso a detectar algún tipo de anomalía o ataque a un sistema Linux. En la Tabla 3.1 se muestran las variables seleccionadas dado que brindan estadísticas importantes que pueden indicar algún tipo de actividad sospechosa si el número de ocurrencias de alguna de estas variables sobrepasa los umbrales establecidos.

Por ejemplo, en la bitácora *secure* se puede encontrar una línea como la siguiente:

```
Oct 31 00:52:59 delfos sshd[12433]: Failed password for root from 64.34.162.95 port 53795 ssh2
```

Esta línea indica que un usuario con dirección IP identificada como 64.34.162.95 intentó conectarse como root por el puerto 53785 el día 31 de octubre a las 00:52:59 horas, pero con una contraseña incorrecta. Por un lado es normal que un usuario legítimo se equivoque de contraseña un número de veces razonable —3 a 5 veces—, pero si esta ocurrencia tiene apariciones excesivas en un periodo de tiempo muy corto posiblemente se trate de un pirata malicioso que intenta ganar acceso al sistema con privilegios de administrador.

De igual manera, si alguna de las variables tiene demasiadas ocurrencias en las bitácoras correspondientes, tal que sobrepasen un valor determinado, umbral, es un indicativo de que alguna actividad sospechosa está ocurriendo en el sistema. Cada variable tiene definidos sus propios umbrales; esto se debe a que no todas las variables tienen el mismo comportamiento, por ejemplo, es razonable que un usuario se equivoque tres veces o hasta cinco veces de contraseña debido a que posiblemente la olvidó o la tecleó mal, por lo que los umbrales estarían en el rango de tres y cinco, pero para la cantidad de correos recibidos un umbral de cinco no sería representativo de un comportamiento anómalo ya que para pensar que se trata de correo SPAM, por ejemplo, se requiere recibir cientos o hasta miles de correos en cuestión de segundos.

Variable	Bitácora	Comentario
Acceso con nombres de usuarios incorrectos	*Secure*	Ayuda a detectar a un intruso que mediante algún *script* o por fuerza bruta intenta ingresar al sistema probando con diferentes usuarios, muchos de ellos inexistentes
Acceso con contraseñas incorrectas	*Secure*	Ayuda a detectar a un intruso que intenta ingresar al sistema probando contraseñas —igualmente mediante un *script* o por fuerza bruta— con nombres de usuarios conocidos —como root o admin—
Correos enviados al servidor	*Maillog*	Ofrece una idea de la cantidad de correos que están llegando al servidor, si esta cantidad de correos es excesiva se puede pensar que se trata de correo SPAM
Correos enviados desde el servidor	*Maillog*	Brinda una idea de la cantidad de correos que están saliendo del servidor en un momento dado
Correos enviados a usuarios inexistentes	*Maillog*	Informa de la cantidad de correos que se envían a usuarios inexistentes en el sistema
Conexiones perdidas con el servidor de correo	*Maillog*	Informa de las conexiones que se pierden con el servidor
Tamaño de los correos	*Maillog*	Permite calcular el tamaño de los correos, con el cual podremos sacar patrones de uso del servidor de correo
Acceso a páginas del servidor	*Acces_log*	Se pueden obtener estadísticas de las visitas al servidor web
Acceso a páginas inexistentes	*Error_log*	Indica los intentos de acceso a páginas que no existen en el servidor
Acceso a directorios prohibido	*Error_log*	Brinda información de los intentos de acceso a través del servidor web a directorios prohibidos
Número de sesiones activas	*Messages*	Permite obtener patrones de las sesiones de cada usuario del sistema supervisado

Tabla 3.1: Lista de variables analizadas para el SDIh

Por ejemplo, la siguiente línea contiene la ocurrencia de una visita a una página del servidor web y se extrae de la bitácora *acces_log*:

ltia-18.tamps.cinvestav.mx - - [02/Nov/2006:21:36:00 -0600] "GET /mchable/mes.png HTTP/1.1"200 2958 "http://delfos.cgsca.cinvestav.mx/cgi-bin/red.perl.cg" "Mozilla/5.0 (Windows; U; Windows NT 5.1; es-AR; rv:1.8.0.7) Gecko/20060909 Firefox/1.5.0."

De toda esta línea sólo interesa conocer el tipo de ocurrencia que se presenta. En este caso, la palabra *GET* indica que se trata de un acceso a alguna página del servidor. También interesa conocer el rastro del origen de la ocurrencia, en este caso el nombre del servidor que intenta acceder a alguna página web del sistema y el nombre de la página que está accediendo, además la fecha y hora exacta de la ocurrencia. Mediante un

script sencillo para reconocimiento de cadenas se separa la información que es útil de la información que no lo es. Así la línea depurada queda de la siguiente manera:

```
ltia-18.tamps.cinvestav.mx 02/Nov/2006:21:36:00 cgi-bin/red.perl.cgi
```

Esta información preprocesada será la entrada para el algoritmo que extraiga los datos y los almacene en una base de datos. De este modo, la manipulación de la información será más sencilla.

Extracción de datos

Con el preprocesamiento de los datos contenidos en las bitácoras se agiliza la extracción de los datos. Esta extracción analiza cada línea de las bitácoras preprocesadas y verifica si hay ocurrencias de alguna de las variables de la lista de variables y si es así las contabiliza. El Algoritmo 1 es el que se encarga de realizar la extracción de datos de las bitácoras preprocesadas.

Este algoritmo es un proceso que se ejecuta a intervalos de tiempo regulares y por cada variable contenida en la lista de variables analiza las bitácoras preprocesadas y contabiliza sus ocurrencias. Al finalizar, almacena por cada variable el contador en la base de datos correspondiente.

Algoritmo 1 Algoritmo de extracción de datos

Entrada: Archivo de bitácora preprocesado
Salida: Base de datos de cada *variable* actualizada con los nuevos contadores

```
1:  while true do
2:    Esperar el nuevo periodo de muestreo;
3:    for cada variable en la lista de variables do
4:      for cada línea de la bitácora preprocesada do
5:        if variable tiene ocurrencia then
6:          incrementa c
7:        end if
8:      end for
9:      Almacena c en una BD
10:   end for
11: end while
```

Los algoritmos de preprocesamiento, extracción de datos y de almacenamiento fueron implementados en Perl [60]. Se eligió este lenguaje debido a la facilidad con que permite el manejo de archivos y de cadenas.

Almacenamiento

El módulo de almacenamiento se encarga de tomar los contadores obtenidos en el módulo de extracción de datos y los almacena en una base de datos de huésped que queda a disponibilidad del módulo de gráficas y del módulo de alertas.

Variable	Las variables se identifican por un número del uno al once
Tiempo	El tiempo está dado en el formato *POSIX* de Linux
Contador	Los contadores son numéricos y cuentan el número de ocurrencias que cada variable tiene en un periodo de tiempo específico

Tabla 3.2: Estructura de la base de huésped que almacena la información de las bitácoras

Para el almacenamiento de los datos se decidió crear una base de datos mediante la herramienta RRDtool [41] que almacena y muestra datos a través del tiempo. Una característica importante de esta herramienta es que almacena los datos de manera compacta, *round robin*. Cada uno de los archivos tiene un tamaño fijo, por lo que no crece en el tiempo. Existe un apuntador al último dato recogido y cuando la base de datos llega a su límite, la información nueva es almacenada en los registros más antiguos.

Los registros de la base de datos de huésped tienen la estructura mostrada en la Tabla 3.2. Sólo contiene tres campos; el tipo de variable, el tiempo de la ocurrencia —que tiene que corresponder con el periodo de recolección— y el número de veces que el evento se presenta.

El campo *Tiempo* almacena el tiempo exacto de cada periodo, éste está dado en el tiempo de Unix conocido como el formato de tiempo POSIX, el cual es un sistema para describir puntos en el tiempo y se trata del número de segundos transcurridos desde la media noche del día 1 de enero del año 1970.

El campo *Contadores* es donde se almacenan los contadores obtenidos en el módulo extracción de datos. Este contador es uno de los datos más importantes de la tabla, ya que es el indicador numérico de la ocurrencia de las variables.

3.2.2. Módulo de graficación

Cuando se han calculado y almacenado en una base de datos los contadores correspondientes a las variables supervisadas resulta una tarea fácil manipular esta información. El módulo de graficación se encarga de tomar la información contenida en la base de datos de huésped y realizar gráficas con esta información.

Para una vista más sencilla y amigable con el usuario se definieron diferentes gráficas de cada una de las variables analizadas; una gráfica anual, una gráfica mensual, una gráfica semanal y una gráfica diaria. Ésto permite al usuario poder visualizar su comportamiento a través del tiempo para así, establecer fácilmente patrones de uso. Con las gráficas también es posible detectar alguna desviación numérica drástica en el comportamiento habitual de cada una de las variables.

El proceso de graficación mostrado en el Algoritmo 2, toma los contadores de cada variable almacenados en la base de datos para realizar las gráficas y actualizarlas. Las gráficas fueron realizadas con la herramienta RRDtool [41].

Por ejemplo, la gráfica de la Figura 3.7 muestra el comportamiento de la variable *Visitas a páginas del servidor web* durante un día de actividad y como se ve en ciertos periodos de tiempo tiene una cantidad de actividad que puede exceder el valor establecido

Algoritmo 2 Algoritmo de graficación

Entrada: Base de datos con la información a graficar
Salida: Gráficas actualizadas para las variables en la lista de variables

```
1: while true do
2:    Esperar el nuevo periodo de graficación
3:    for cada variable X en la lista de variables  do
4:       Obtener el valor de c para la variable X en el periodo actual
5:       Actualizar las gráficas para X en el periodo actual con el valor de c
6:    end for
7: end while
```

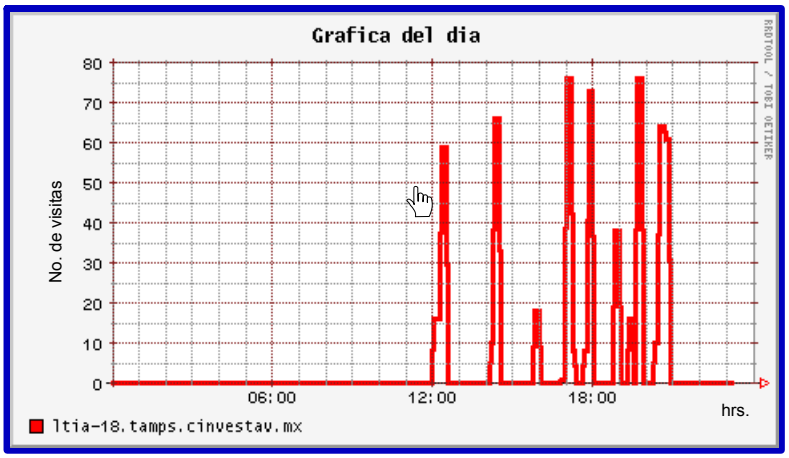


Figura 3.7: Gráfica diaria de la variable *visitas a páginas del servidor*

por los umbrales. Esta gráfica se actualiza cada periodo de graficación; cuando este periodo se cumple, el proceso de graficación toma los contadores de cada una de las variables correspondientes al periodo actual que están contenidos en la base de datos y actualiza la gráfica con estos datos.

3.2.3. Módulo de alertas

Cuando los datos han sido extraídos de las bitácoras y almacenados, están listos para ser manipulados. El módulo de alertas es el que se encarga de analizar los datos recolectados en busca de posibles ataques o intentos de intrusión y generar las alertas correspondientes. Éste es uno de los módulos más importantes del agente de huésped, ya que es el que añade la característica de detección al agente.

Las alertas se generan mediante un método estadístico analizando periódicamente los datos de la base de datos de huésped para detectar algún tipo de actividad anómala. Este método mantiene dos indicadores —dos umbrales—; si el contador de alguna variable desvía numéricamente estos indicadores de manera considerable se genera la alerta

correspondiente.

Se requiere reconocer algunos estados en el sistema supervisado. En este trabajo se consideraron tres, los cuales se asocian a tres alertas de diferente color:

- Estado normal - alerta verde. Este estado indica que el sistema se encuentra estable, y que los análisis hechos por el agente de huésped no han resuelto algún tipo de actividad sospechosa.

- Estado preventivo - alerta amarilla. Cuando se genera una alerta amarilla por el agente de huésped hay que tener especial cuidado en la variable afectada, y habrá que estar atentos a un posible ataque o intrusión para realizar las medidas que contrarresten dicha actividad.

- Estado crítico - alerta roja. Cuando se presenta una alerta de este tipo el administrador tiene que actuar de manera inmediata, ya que el sistema puede estar siendo atacado de forma potencial y encontrarse en un estado crítico.

Del mismo modo se deben definir dos umbrales que serán el indicativo de cada alerta. Los dos umbrales definidos son para los dos últimos estados: el estado preventivo y el estado crítico —$Umbral_a$ y $Umbral_r$, respectivamente—. Como ya se mencionó, estos umbrales son definidos por el administrador y son diferentes para cada una de las variables, dependiendo de la naturaleza de cada una de ellas.

La función con la que se calculan las alertas es la siguiente:

$$f_{alerta}(c) = \begin{cases} 0, & \text{Si} \quad c < \text{Umbral}_a \\ 1, & \text{Si} \quad \text{Umbral}_a \leq c < \text{Umbral}_r \\ 2, & \text{Si} \quad \text{Umbral}_r \leq c \ . \end{cases}$$

donde:

c es el contador correspondiente a cada una de las variables,

0 corresponde a una alerta verde —buen estado del sistema—,

1 corresponde a una alerta amarilla —alerta preventiva—,

2 corresponde a una alerta roja —estado crítico del sistema—.

Esta función busca evaluar la variabilidad del comportamiento de cada una de las variables, si este comportamiento se desvía de lo que es considerado 'Normal" o cotidiano, se considera "Anormal".

3.2.4. Módulo de interconexión del huésped

El sistema de este trabajo de tesis cuenta con un sistema de respuestas pasivas, es decir, que cuando detecta alguna alerta, se avisa inmediatamente a las partes interesadas —en este caso al administrador del sistema y al módulo transceptor—, pero el sistema no tiene capacidad de emprender automáticamente alguna medida en contra del atacante.

El módulo de interconexión del huésped se encarga de difundir las alertas generadas con el módulo de alertas tanto al administrador del sistema como al transceptor —o módulo

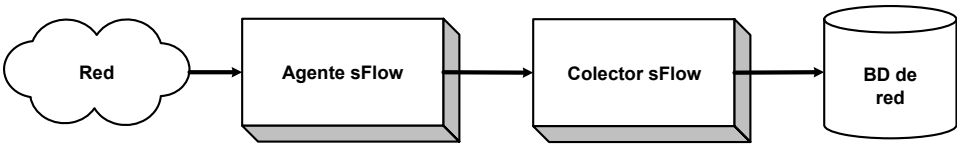


Figura 3.8: Componentes del agente de red

de interconexión de "todo el sistema"—. Este módulo es el punto de comunicación con el agente de interconexión del sistema (ver Figura 3.3) y de aquí con todo el sistema; puede tanto enviar como recibir información.

En caso de que el resultado de la evaluación de $f_{alerta}(c)$ sea 1 ó 2, se enviará el aviso correspondiente al administrador y al módulo de interconexión del sistema. Estas alertas se enviarán al administrador vía correo electrónico y serán mostradas a través de una interfaz de usuario. El módulo de interconexión, por su parte, recibirá las alertas mediante un mensaje contenido en un archivo. Al recibir alguna alerta, el administrador deberá tomar las medidas correspondientes para proteger su sistema, mientras que el módulo de interconexión usará estos avisos para emprender otras acciones; como utilizar la información de la red para detectar qué otros huéspedes están siendo afectados por el mismo atacante.

3.3. Agente de red (SDIr)

El agente de red -SDIr- del sistema de monitoreo y detección de intrusos se encarga de supervisar todo un segmento de red en busca de anomalías o actividades sospechosas y la mayoría de los SDIr, utiliza como fuente de información el tráfico de entrada y salida de una red. Este agente almacena toda esta información en una base de datos que cabe aclarar, no es la misma que la base de datos de huésped, por lo que en sucesivas referencias a la base de datos del agente de red se nombrará "base de datos de red".

El agente de red está compuesto de tres módulos o elementos —como muestra la Figura 3.8—: un agente, un colector y una base de datos. El agente es el que se encarga hacer el muestreo de paquetes de la red supervisada y los paquetes muestreados son enviados al colector para su análisis. El colector organiza estos datos y los envía a la base de datos de red para su almacenamiento.

Los dos primeros componentes —el agente y el colector—, se hicieron basándose en el protocolo sFlow [3], el cual se describe en la siguiente subsección.

3.3.1. sFlow

sFlow [3] es una tecnología de muestreo de paquetes de múltiples vendedores disponible para *switches* y *routers*. Ésta provee la habilidad para supervisar continuamente el flujo de tráfico a nivel de aplicación sobre varias interfaces conectadas a una red de manera simultánea. sFlow tiene varias propiedades interesantes [49]:

- Exactitud, porque el muestro es tan simple que puede ser ejecutado en *hardware* y es capaz de operar en conexiones veloces sin añadir carga significativa al sistema. sFlow está diseñado así para que la exactitud de todas las medidas puedan ser determinadas.
- Escalable en tamaño y velocidad de la redes que se supervisan, ya que es capaz de supervisar redes de 10 Gbps, 100 Gbps y más veloces, además, miles de dispositivos pueden ser supervisados por un sólo colector sFlow.
- Bajo costo. El agente sFlow es muy simple de implementar y añade un costo despreciable a un *switch* o *router* debido a que el muestreo de paquetes es sencillo.
- Tiempo. El colector sFlow siempre proporciona el minuto exacto en que suceden los eventos del tráfico a través de la red entera. La información del tiempo es particularmente importante si los datos de tráfico son necesarios para proveer control en tiempo real.

Con la ayuda de sFlow se facilita la implementación de aplicaciones de detección, diagnóstico y reparación para el manejo de una red en base al tiempo ya que proporciona el minuto exacto en que el tráfico circula a través de la red entera.

Funcionamiento de sFlow

sFlow está constituido básicamente de dos elementos: (ver Figura 3.9) un agente sFlow y un colector sFlow.

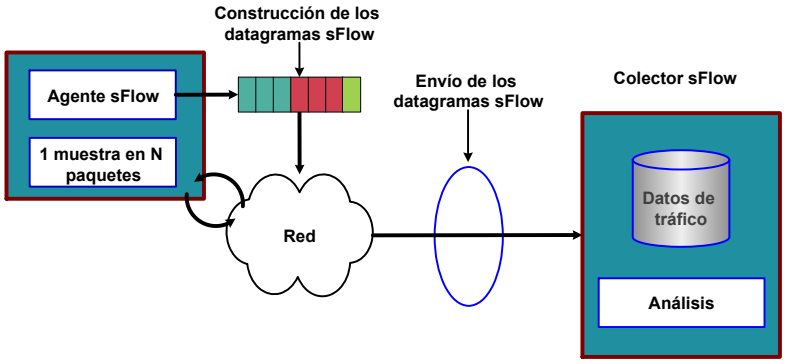


Figura 3.9: Componentes básicos de sFlow

El agente sFlow es un proceso de *software* que es parte del mismo *firmware* de los dispositivos que se encargan del manejo de la red como *switches* o *routers*. El agente construye datagramas con cada uno de los paquetes muestreados.

El colector sFlow es un programa que analiza y organiza los datagramas recibidos por el agente sFlow para producir una vista amplia y rica, en tiempo real del funcionamiento de la red. El agente sFlow viene incluido con los dispositivos de red que soportan esta

Figura 3.10: Componentes de un datagrama sFlow

tecnología mientras que el colector es un *software* libre y con varios tipos de distribuciones que pueden ser descargados de la red.

El funcionamiento de sFlow es el siguiente: el agente sFlow construye los datagramas por cada paquete muestreado (una muestra en N paquetes) y una vez que los datagramas son construidos y capturados por el agente sFlow son enviados a través de la red a un colector sFlow central para su análisis y visualización.

3.3.2. Agente sFlow

Una gran cantidad de dispositivos de red soportan la tecnología sFlow y viene incluida ya en el *hardware* de éstos. El muestreo es ejecutado típicamente por el módulo *ASIC* —circuito integrado de uso específico— de los *switches* y *routers*, dando un buen desempeño de velocidad en las conexiones. El agente sFlow forma datagramas sFlow con cada uno de los paquetes muestreados.

El agente sFlow hace muy poco procesamiento. Éste simplemente empaqueta datos dentro de datagramas sFlow que son inmediatamente enviados a través de la red a un colector sFlow. El reenvío inmediato de datos minimiza los requerimientos de CPU y memoria asociados con el agente sFlow. El muestreo de paquetes es tan simple que puede ser ejecutado en *hardware*, así sFlow permite que todas las medidas puedan ser determinadas con exactitud.

El muestreo de paquetes ha sido usado para supervisar el tráfico de una red alrededor de los últimos diez años. En un ambiente centralizado, el lugar más efectivo para supervisar es dentro de los *switches/routers*, donde todo el tráfico será visto. Otros tipos de pruebas tradicionales tienen solo una vista parcial del tráfico.

sFlow implementa el muestreo de paquetes del siguiente modo: se toma una muestra de N paquetes circulantes a través de la red. El módulo *ASIC* de los *switches* y *routers* envía al agente sFlow la cabecera del paquete muestreado, las interfaces de entrada y salida de dicho paquete y los parámetros del proceso de muestreo como total de muestras y paquetes hasta ese momento; con toda esta información el agente sFlow comienza a construir el datagrama como muestra la Figura 3.10. En la segunda etapa, el agente sFlow añade información importante al datagrama como puerto fuente y destino, IP fuente y destino, además de los identificadores y URL fuente y destino asociados a los usuarios involucrados. Finalmente el agente sFlow añade los contadores de interface al datagrama que son los datos numéricos que indican cuantos paquetes han circulado en realidad por la red, la cantidad de bytes, entre otros datos.

3.3.3. Colector sFlow

El colector sFlow es el que recolecta los datos enviados por el agente sFlow para analizarlos y presentarlos mediante una interfaz de manera sencilla y amigable. Para este trabajo de tesis se usaron dos colectores: el sFlow toolkit [57] y el flow tools [2].

El colector sFlow toolkit recibe los datagramas de los paquetes muestreados que le envía el agente sFlow como lo muestra la Figura 3.11. Este colector sirvió sólo para recoger los datagramas enviados por el agente, porque el formato de la salida de este colector es bastante ilegible. Por esta razón se decidió mandar estos datos a otro colector muy sencillo y con bastante flexibilidad a la hora de procesar los datos: flow tools. Después de procesar los datos con estos dos colectores se envían a la base de datos de red para su almacenamiento.

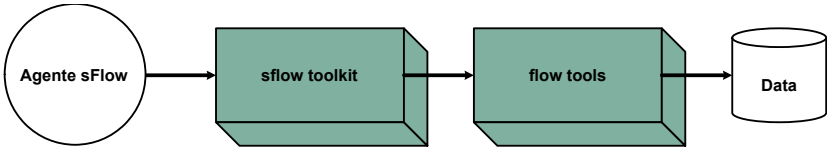


Figura 3.11: Colector sFlow

sFlow Toolkit

La herramienta sFlow toolkit proporciona utilidades de línea de comandos y *scripts* para analizar datos sFlow.

El componente núcleo de la herramienta sFlow es la utilidad de línea de comandos *sflowtool*, la cual interactúa con colectores compatibles NetFlow para conteo de flujo IP y provee salida en modo texto que puede ser usada en *scripts* para proveer análisis solicitado y reportar de manera integrada con otras herramientas como RRDtool. En este trabajo el colector sFlow toolkit interactúa con el colector flow tools de Netflow para convertir los datagramas en este formato y así se facilitar la selección de los datos que se almacenan en la base de datos de red. Para usar sflowtool es necesario especificar un huésped en donde se ejecuta Netflow y el puerto.

Flow-tools (colector de Netflow)

Netflow [53], es una tecnología desarrollada por CISCO Systems en 1996, permite mejorar la capacidad de encaminamiento de sus routers. Para hacer la recolección de los datos se utilizó el colector flow tools que convierte los datagramas sFlow en datagramas Netflow, debido a la facilidad con la que estos últimos se pueden leer y manipular.

Flow tools, es una biblioteca que contiene un conjunto de programas usados para colectar, enviar, procesar y generar reportes con datos Netflow. Un flujo es una colección de campos claves y datos adicionales proporcionados por un datagrama NetFlow. Los

campos claves de un flujo son dirección IP fuente, dirección IP destino, entrada, salida, puerto fuente, puerto destino, protocolo.

En este trabajo de tesis se usaron básicamente dos programas de la biblioteca flow tools: flow-capture y flow-export, que vienen incluidos en la distribución:

- flow-capture. La utilidad flow-capture recibe y almacena exportaciones Netflow al disco duro. Los archivos de flujo son almacenados periódicamente por día. El almacenamiento de los archivos de flujo funciona como una cola circular; cuando expira el espacio de disco duro designado para el almacenamiento, los archivos más viejos se van sustituyendo por los archivos nuevos. Los archivos son almacenados en un directorio de trabajo designado.

  Cuando la dirección IP es configurada, sólo los flujos que sean exportados de esa dirección podrán ser procesados. Ésta es una configuración más recomendada y segura. Si la dirección IP es igual a cero, podrán ser aceptados los flujos de cualquier dirección IP. Un ejemplo del uso de flow-capture es el siguiente:

  ```
  flow-capture -w directorio de trabajo -E nG -n rotaciones direccion ip/puerto -R script
  ```

  donde las opciones indican lo siguiente:

  - -w directorio de trabajo. Es el directorio que se elige para almacenar los archivos de flujo.
  - -E nG. Es el espacio que se designa para el almacenamiento de los archivos de flujo —dado en Gb—.
  - -n rotaciones. Es el número de rotaciones que se realizarán en el día.
  - direccion IP/puerto. Indica la dirección IP y el puerto por el que el colector escuchará.
  - -R script. Se refiere al *script* que ejecutará inmediatamente después de hacer el corte de cada rotación.

- flow-export. La utilidad flow-export convierte archivos de flujo a formato ASCII CSV, cflowd, pcap, wire, mysql y PGSQL.

  Esta utilidad se usó para implementar el algoritmo de colección y almacenamiento de datos en la base de datos de red. Este algoritmo se especifica en las opciones de flow-capture con la opción -R y es ejecutado en cada corte después de las rotaciones.

  El Algoritmo 3 procesa los archivos de flujo y convierte los datos del archivo de flujo en formato ASCII —mediante la utilidad flow-export—. Los datos procesados y seleccionados son almacenados en la base de datos de red que se analizará en la subsección 3.3.4.

Cada línea del archivo *Datos* representa un registro en la base de datos de red, y cada línea contiene cadenas delimitadas por comas que corresponden a cada uno de los campos de esta base de datos.

Algoritmo 3 Algoritmo colector de tráfico

Entrada: Archivo de flujo generado por flow-capture.
Salida: Base de datos de red actualizada con los datos del archivo de flujo.

```
1: Datos ← flow−export(Archivo de flujo)
2: for cada línea del archivo Datos do
3:    Obtener cada cadena delimitada por una coma
4:    Insertar cadenas en la base de datos de red
5: end for
```

3.3.4. Base de datos de red para guardar información sFlow

La estructura de la base de datos de red donde se almacenan los datos procesados por los colectores es la que se muestra en la Tabla 3.3. Esta base de datos es el punto de comunicación con el módulo de interconexión de todo el sistema.

El campo *Hora* indica la hora exacta (segundo, minuto, hora, día, mes y año) en que se produce el muestreo de paquetes de la sesión involucrada.

El campo *Protocolo* indica el protocolo usado en la transmisión de los datos en la sesión.

El campo *IPFuente* contiene la dirección IP origen desde donde se están transmitiendo los datos en la sesión.

El campo *IPDestino* contiene la dirección IP destino a donde se dirigen los datos transmitidos en la sesión.

El campo *PtoFuente* es el puerto origen usado en la transmisión de los datos.

El campo *PtoDestino* es el puerto destino usado en la transmisión de los datos.

El campo *Paquetes* indica la cantidad de paquetes transmitidos durante la sesión.

El campo *Bytes* indica la cantidad de bytes transmitidos durante la sesión.

La base de datos de red representada por la estructura de la Tabla 3.3 guarda toda la información referente a la red que se requiere para hacer las detecciones adecuadas.

Campo	*Tipo*	*Descripción*
Hora	varchar (10)	Hora exacta en segundos en que ocurrió el evento
Protocolo	varchar (15)	Protocolo usado en la transmisión
IPFuente	varchar (15)	Dirección IP origen
IPDestino	varchar (15)	Dirección IP destino
PtoFuente	smallint (5)	Puerto origen
PtoDestino	smallint (5)	Puerto destino
Paquetes	int (11)	Tamaño en paquetes de los datos transmitidos en la sesión
Bytes	int (11)	Tamaño en bytes de los datos transmitidos en la sesión

Tabla 3.3: Estructura de la base de datos de red que almacena los datos sFlow

Hora	*IP Fuente*	*IP Destino*	*Puerto Fuente*	*Puerto Destino*
11:10	148.247.10.1	192.168.10.157	8080	2030
11:10	148.247.10.1	192.168.10.157	8080	2031
11:10	148.247.10.1	192.168.10.157	8080	2032
11:10	148.247.10.1	192.168.10.157	8080	2033
11:10	148.247.10.1	192.168.10.157	8080	.
11:10	148.247.10.1	192.168.10.157	8080	.
11:10	148.247.10.1	192.168.10.157	8080	2040
11:10	148.247.10.1	192.168.10.157	8080	2041

Tabla 3.4: Ejemplo de una exploración de puertos

3.3.5. Detección de patrones de uso del agente de red

Una vez que la información del tráfico de la red se encuentra almacenada en la base de datos de red es muy fácil de manipular. Así, se pueden obtener algunos patrones de uso anormal como la exploración de puertos y la exploración de direcciones IP.

Por ejemplo, en la Tabla 3.4 se observa un claro patrón de exploración de puertos hacia una red en particular. Un ataque de este tipo es bastante simple ya que comprobar el estado de un determinado puerto es a *priori* una tarea muy sencilla; incluso es posible llevarla a cabo desde la línea de órdenes, usando una herramienta tan genérica como *telnet*.

En el Algoritmo 4 se presenta como detectar una exploración de puertos. Este algoritmo analiza los datos contenidos en los registros de la base de datos de red correspondientes a los últimos diez minutos de actividad —este periodo es establecido por el administrador— ; si encuentra que el número de conexiones realizadas a través de diferentes puertos con las mismas direcciones IP origen y destino sobrepasa el valor de alguno de los umbrales establecidos, lanza una alerta al administrador de la red.

Otro de los patrones que se puede detectar con este agente de red es la exploración de direcciones IP; este patrón se determina de una manera muy parecida a la que se presenta en el Algoritmo 4. La idea es analizar los registros de la base de datos de red de los últimos minutos y determinar las conexiones de una dirección IP origen a diferentes direcciones IP destino pertenecientes a la red supervisada. Si esta cantidad de conexiones sobrepasa cierto valor determinado por los umbrales, se considera una actividad sospechosa y se lanza una alerta.

Algoritmo 4 Algoritmo de para detectar exploraciones de puertos

Entrada: Registros de la base de datos de red correspondientes a los últimos diez minutos de actividad de la red

Salida: Verdadero o falso (Verdadero si se encuentra una exploración de puertos y falso si no se detecta)

```
1: SELECT DISTINCT PtoDestino from (Base de Datos)
2: if @contador_elementos >= Umbral_expPuertos then
3:    return true
4: else
5:    return false
6: end if
```

3.4. Módulo de interconexión

El módulo de interconexión es el que proporciona la funcionalidad de híbrido al sistema. El módulo de interconexión dispone de la información proporcionada por el o los agentes de huésped y de la información proporcionada por el agente de red para analizarla y correlacionarla en busca de ataques potenciales contra la red supervisada. Al detectar algún tipo de actividad maliciosa, este módulo alerta a los huéspedes involucrados en el ataque pero además comparte la información de los últimos ataques registrados con el resto de los huéspedes de la red para que los administradores de estos sistemas tomen las medidas necesarias. La idea básica es interconectar al agente de huésped y al agente de red para combinar las características de cada uno de ellos; esta interconexión se realiza a través de mensajes de texto.

Este módulo se encuentra compuesto de varios elementos como se muestra en la Figura 3.12: un monitor de alertas, un módulo de consultas, un módulo de respuestas y un módulo de información compartida.

El monitor de alertas es donde los agentes de huésped colocan las alertas registradas en algún huésped. El módulo de consultas y correlación de información es el que hace las consultas a la base de datos de red en busca de información de los ataques registrados por el agente de huésped o para analizar patrones de uso de la red. El módulo de respuestas es el módulo que comparte la información extraída de la base de datos de la red a los huéspedes involucrados en algún ataque. El módulo de información compartida contiene

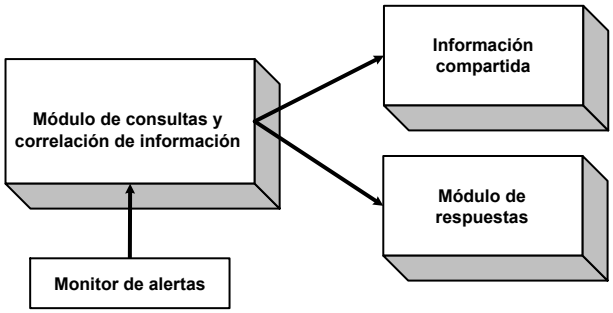


Figura 3.12: Módulo de interconexión

la información de los últimos ataques ocurridos en la red supervisada y esta información es común a todos los huéspedes de la red.

3.4.1. Monitor de alertas

El monitor de alertas es el repositorio en el cual los agentes de huésped colocan los avisos de alerta previniendo al agente de interconexión del sistema de una actividad sospechosa. En el monitor de alertas se indica la dirección IP fuente que está provocando dicha actividad. La comunicación entre el agente de huésped y el monitor de alertas del módulo de interconexión es mediante alertas en un archivo de texto.

El módulo de interconexión revisa continuamente el monitor de alertas y acciona en cuanto detecta alguna alerta nueva en éste, iniciando la comunicación con el agente de red para obtener información más general del ataque registrado y ampliar el panorama de detección.

3.4.2. Módulo de consultas y correlación de información

El módulo de consultas y correlación de información es el que comunica al módulo de interconexión con el agente de red. La comunicación entre estos dos módulos se hace a través de conexiones desde el módulo de consultas del módulo de interconexión del sistema hacia la base de datos de la red.

Las consultas que hace este módulo a la base de datos de la red responden a una alerta nueva en el monitor de alertas del agente de interconexión del sistema; así el módulo de consultas revisa los registros de la base de datos de la red correspondientes a los últimos diez minutos de actividad buscando la actividad de la dirección IP fuente que proporciona la alerta que está siendo atendida. Después correlaciona toda la información obtenida con las alertas de los huéspedes para determinar la magnitud del ataque.

Normalmente, un atacante explora los huéspedes de una red en busca de poder burlar la seguridad de alguna de ellas, y lo hará de las que cuenten con suficientes vulnerabilidades para permitirlo. Así cuando un usuario malicioso ataca a un huésped de una red posiblemente intentará hacer lo mismo con algunos otros hasta lograr su objetivo.

Así, cuando se buscan los detalles de un ataque —en la base de datos de la red— dirigido a un huésped en particular probablemente se encontrarán intentos de acceso a otros huéspedes de la red. El siguiente paso del módulo de consultas y correlación de información es precisamente correlacionar la información recolectada para determinar la magnitud del ataque y obtener a los huéspedes involucrados —en caso de que los haya— en el mismo ataque para informarles antes de que el ataque pase a niveles más serios.

3.4.3. Módulo de respuestas

El módulo de respuestas se encarga de compartir la información obtenida del módulo de consultas y correlación de información sólo con los huéspedes involucrados en algún ataque o actividad sospechosa.

Las respuestas que realiza este trabajo son pasivas y consisten en un aviso de alerta vía correo electrónico al administrador de cada huésped involucrado y una alerta a través de un aviso en la consola de eventos. De este modo, el administrador de cada huésped determinará la naturaleza del ataque y ejecutará una acción correspondiente a la magnitud del ataque; por ejemplo si la naturaleza del ataque es una exploración de puertos tal vez opte por bloquear los puertos en los que trabajen las aplicaciones más vulnerables o tal vez bloquee las conexiones de las direcciones IP de usuarios que ya fueron calificados como maliciosos.

3.4.4. Módulo de información compartida

El módulo de información compartida proporciona información a todos los huéspedes de la red de los últimos ataques registrados en todo el sistema supervisado. Esta información se presenta a través del sitio web del módulo de interconexión que es común a todos los huéspedes, como se explicará en el siguiente capítulo. Este módulo tiene como propósito prevenir a los administradores de cada uno de los huéspedes de la red en contra de ataques potenciales para que analicen lo perjudicial que puede resultar alguno de estos ataques para su sistema y tomen las medidas de prevención correspondientes.

Imagine la siguiente situación: uno o algunos de los huéspedes que componen una red han registrado un ataque en mayor o menor medida. El módulo de respuestas sólo notificará de este ataque a los huéspedes involucrados pero probablemente el atacante ya se encuentre planeando atacar a otros huéspedes de la misma red, entonces el SDI se comporta como un sistema reactivo, es decir, sólo reacciona al presentarse el ataque, pero lo ideal es que no sólo sirva para reaccionar ante un ataque sino para prevenir a la red de un posible ataque potencial. Por esta razón es conveniente no centralizar la información de los ataques registrados en sólo los huéspedes afectados en el ataque, sino compartir la información de este ataque con el resto de los huéspedes, así éstos tendrán un antecedente de la dirección IP origen que presenta comportamientos sospechosos y el administrador podrá ejecutar acciones más contundentes al detectar algún tipo de acceso por parte de esta fuente.

La información se presenta a través de un reporte en la consola de eventos, y su

Hora de registro del ataque	IP origen	IP destino	Tipo de ataque

Tabla 3.5: SDI existentes en el mercado

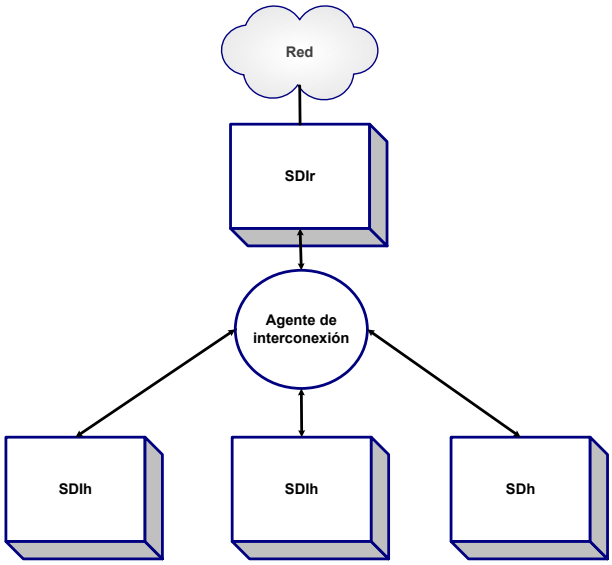


Figura 3.13: Organización del módulo de interconexión

estructura es la presentada en la Tabla 3.5. La información contenida en este reporte corresponde a información muy general del ataque registrado, sólo indica la hora exacta del registro del ataque, la dirección IP origen de donde proviene el ataque, la dirección IP del huésped de la red que está recibiendo el ataque y el tipo de ataque del que se trata. Como se muestra en la Figura 3.13 hay un agente de huésped en cada uno de los huéspedes de la red, un agente de red por cada segmento de red —en este caso uno, porque es un segmento de red solamente— . Debe existir también un agente de interconexión por cada segmento de red y es el que mantiene comunicación de ida y vuelta con cada uno de los agentes.

3.5. Modos de operación del sistema

El sistema de monitoreo y detección de intrusos opera en dos modos: en tiempo real —es decir, se ejecuta con la hora del sistema— y en modo de simulación. El sistema en tiempo real opera con la información obtenida de las bitácoras del sistema y el tráfico que circula a través de la red, además de que utiliza el reloj del sistema. El sistema en modo simulación opera con información simulada inducida por el administrador tratando de emular los ataques y utiliza el tiempo de un reloj de simulación que se implementó especialmente para este modo de operación.

A continuación se describe los tipos de simuladores que existen, los motivos que lleva-

ron a implementar el sistema en modo simulación y los cambios que se hicieron al sistema en modo en tiempo real para que funcionara en modo de simulación.

3.5.1. Modo de simulación

La mayoría de los simuladores de la actualidad pueden ser de dos tipos [63]:

Simuladores fuera de línea. Son usualmente herramientas altamente configurables y extendibles, diseñadas para simular procesos de una red en una escala de tiempo virtual, la cual no está linealmente relacionada con el tiempo real. Las simulaciones son ejecutadas dependiendo de tareas calendarizadas cuyo resultado puede ser analizado después de completada la simulación.

Simuladores de tiempo real. Son herramientas capaces de crear topologías de red virtuales y simular los efectos del tráfico en tiempo real o escalado. Dichos simuladores son generalmente menos flexibles que su contraparte fuera de línea; sin embargo, su mayor ventaja reside en su capacidad de análisis de resultados durante la simulación.

El simulador desarrollado pertenece a la categoría de simuladores de tiempo real, ya que simula procesos en una escala de tiempo que está relacionada con el tiempo real.

Existen varias factores que motivaron a la implementación del sistema de monitoreo y detección de intrusos en modo de simulación. Dentro de estas razones encontramos:

- En primer lugar para poder hacer pruebas exhaustivas, induciendo los ataques que se requieran y así poder evaluar el funcionamiento del sistema para todos los casos para los que fue diseñado e implementado.
- Para poder ver los resultados de la evaluación del sistema de manera más rápida, sin tener que esperar horas o hasta días para ver dichos resultados.
- Finalmente, para poder ver la funcionalidad del sistema. Fue en esta etapa donde se pudieron corregir la mayoría de los errores que no se habían tomado en cuenta en el sistema en tiempo real y que fueron añadidos después de hacer las pruebas con el sistema en modo de simulación.

Para implementar el sistema en modo de simulación se tuvieron que hacer algunas modificaciones al sistema en modo normal realizado en este trabajo: primero, se creó un reloj de simulación y los datos que alimentan al sistema son simulados.

Reloj de simulación

La primera modificación que se hizo al SDI fue crear un reloj independiente al del sistema. Este reloj tiene un tiempo (de simulación) que avanza más rápido que el tiempo del sistema, tan rápido como el administrador del sistema requiera.

El reloj consiste en un archivo que se actualiza cada N segundos y hace la siguiente correspondencia:

fecha	fuente	tipo de evento

Tabla 3.6: Estructura del archivo de bitácora simulado

$$N \textit{ segundos} \rightarrow \textit{1 minuto}$$

Se usaron todos los algoritmos del sistema en modo normal; una de las diferencias es que en lugar de tomar el tiempo del reloj del sistema (que es el tiempo real) se tomó el tiempo del archivo que contiene el tiempo de simulación.

Algoritmo 5 Algoritmo de para calcular el tiempo de simulación

Entrada: N que indica los N segundos que representarán un minuto.
Salida: El archivo de tiempo de simulación actualizado

```
1: Para nveces suficientemente grande
2: for i = 0 to i < nveces do
3:    delay(N)
4:    tiempo=Correspondencia(N segundos → 1 minuto)
5:    escribir a archivo el tiempo calculado
6: end for
```

El Algoritmo 5 consiste en un ciclo bastante grande que actualiza el archivo de tiempo de simulación cada N segundos con la correspondencia calculada.

Datos simulados

Otra diferencia importante con respecto al sistema en tiempo real, es que los datos que se procesan son generados por el usuario de manera aleatoria o inducida. Se generaron los dos tipos de información que se requieren para el sistema: los archivos de bitácora y el tráfico de la red.

Mediante un sitio web que se explica a detalle en el siguiente capítulo, el usuario tiene la opción de introducir datos que quiera que se procesen, esto con el fin de hacer pruebas al sistema con diferentes tipos de datos.

El primer tipo de datos —los archivos de bitácoras— se escriben en un archivo y éste será la bitácora que procesará el sistema de simulación en vez de procesar las bitácoras reales. El formato de este archivo es mostrado en la Tabla 3.6.

El segundo tipo de datos —el tráfico de la red— es generado automáticamente después de generar las bitácoras simuladas. Es importante que las bitácoras y la información del tráfico de la red tengan concordancia respecto de los datos contenidos. Así, cuando en un huésped se genera un ataque inducido por el administrador, se tomará el tiempo del reloj de simulación del huésped para generar las bitácoras y simultáneamente generar la información del tráfico de la red que se almacenará en la base de datos de la red (ver Figura 3.14).

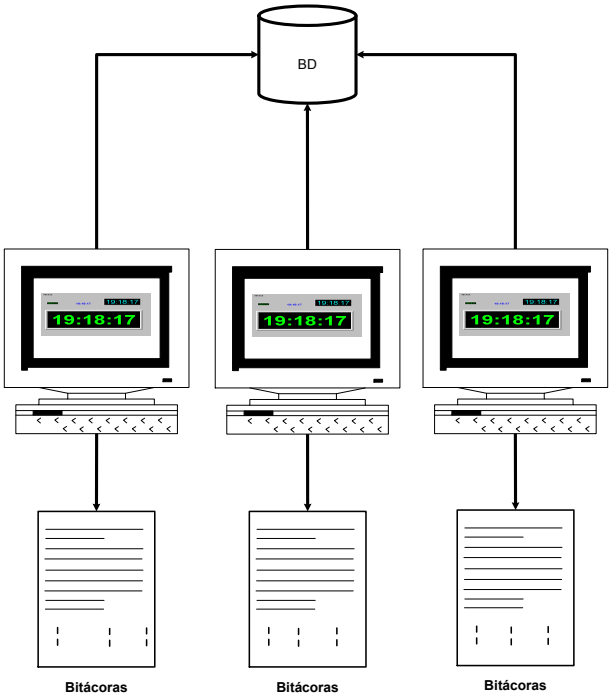


Figura 3.14: Generación de los datos que alimentan al SDI

Id	Hora	IP fuente	IP destino	Pto fuente	Pto destino	Pkts	Bytes

Tabla 3.7: Estructura de los datos de tráfico simulados

La base de datos de red —con datos simulados— tiene la misma estructura que la base de datos del sistema en modo normal. Esta estructura se presenta en la Tabla 3.7.

Así que con el reloj de simulación y con la información generada, el sistema se encuentra listo para funcionar en su fase de simulación.

3.6. Resumen

En este capítulo se explicaron los detalles de la implementación del sistema desarrollado para este trabajo de tesis. Finalmente, se obtuvo un SDIh&r con las siguientes características:

- Un SDI que está compuesto por tres módulos principales: un agente de huésped —o SDIh—, un agente de red —o SDIr— y un transceptor —o módulo de interconexión—.
- El sistema combina los dos enfoques más importantes que han surgido de los SDI: los SDIh y los SDIr para aprovechar las mejores características de cada uno de ellos,

las cuales se mencionan a continuación:

1. Por un lado, los SDIh pueden trazar con más exactitud el rastro de un atacante debido a que analizan un único sistema. Además los SDIh son de bajo costo computacional, ya que son capaces de funcionar con los mismos recursos del sistema. Por último, los SDIh pueden detectar ataques a nivel del sistema operativo y de las aplicaciones.
2. Por otro lado, los SDIr pueden proporcionar información más general de un ataque, ya que éstos registran información de todas las sesiones usadas para la transmisión de los paquetes circulantes por la red y a diferencia de los SDIh, los SDIr pueden detectar ataques dirigidos al segmento de red supervisado, lo que involucra a más de un huésped.

- El SDIh y el SDIr pueden trabajar de manera independiente desempeñando la función específica para el que cada uno fue diseñado. Y pueden ser interconectados mediante el módulo de interconexión para convertir al sistema en un SDIh&r.
- El sistema de monitoreo y detección de intrusos fue implementado en modo normal —con el tiempo del sistema— y en modo de simulación —con un reloj de simulación— con el objeto de probar todos los casos para los que el sistema fue diseñado.

Capítulo 4

Consola de Eventos y Visualización de Resultados

La consola de eventos es un elemento que funciona como interfaz entre el SDI y el administrador. Los resultados del sistema se presentan en esta interfaz, ayudando así a la toma de decisiones. No conforma una parte técnica del esquema, pero es sin duda un elemento muy importante.

En este sistema, la consola de eventos es una interfaz web en la cual se presentan todos los resultados obtenidos con cada uno de los módulos explicados hasta el momento. Dichos resultados son mostrados mediante gráficas y reportes. Esta interfaz tiene como objetivo brindar al administrador un panorama general de todo lo que sucede en la red y en cada uno de los huéspedes que la conforman de una manera amigable, además de alertarlo cuando una actividad sospechosa esté ocurriendo.

La consola de eventos se encuentra formada por tres sitios web, uno por cada módulo del sistema de monitoreo y detección de intrusos: un sitio web para el SDIh, un sitio web para el SDIr y un sitio web para el módulo de interconexión. El mapa general de la consola de eventos de este sistema se muestra en la Figura 4.1; se encuentra instalado un sitio web de SDIh por cada uno de los servidores que integran la red además un sitio web de SDIr por cada segmento de red. Se debe instalar un sitio web del módulo de interconexión por cada servidor de la red.

El sitio web del SDIr es común a todos los servidores pertenecientes a la red supervisada. De igual modo, la página de "Ataques registrados" perteneciente al sitio web del módulo de interconexión es común a todos los servidores y proporciona información acerca de los últimos ataques registrados en el sistema a todos los huéspedes de la red. El sitio web del SDIh es exclusivo de cada huésped y es en este sitio donde el módulo de interconexión coloca todas las alertas determinadas, además de las gráficas y reportes correspondientes al comportamiento del sistema a través del tiempo.

Al igual que los módulos del sistema de monitoreo y detección de intrusos, los sitios web que componen la consola de eventos pueden trabajar de manera autónoma sin necesitar ninguno del otro y ser instalados de manera independiente funcionando de manera autónoma e igualmente eficiente.

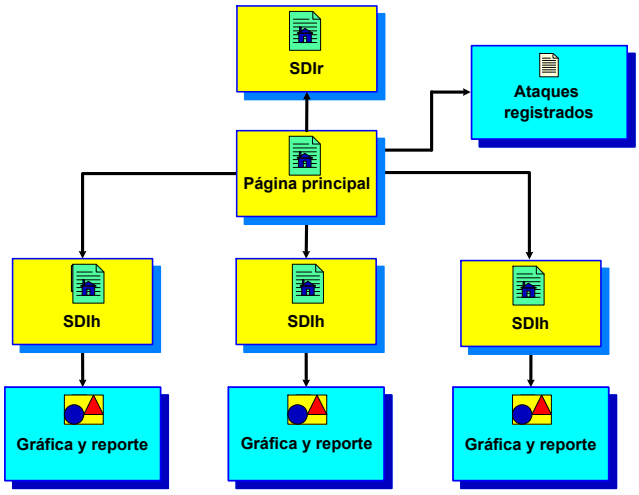


Figura 4.1: Mapa de la consola de eventos del sistema

El sistema de monitoreo y detección de intrusos se encuentra en evaluación desde el mes de enero del año 2006 en un servidor llamado delfos y supervisa una subred del Cinvestav que consiste de un aproximado de cien nodos y cinco servidores (correo, web, dns, ftp y base de datos).

Todos estos sitios web pueden operar en dos modos al igual que el sistema de monitoreo y detección de intrusos: en tiempo real y en modo simulación.

4.1. Estructura de la visualización de resultados

Uno de los objetivos de la consola de eventos es presentar al administrador del sistema todos los resultados obtenidos por el sistema de monitoreo y detección de intrusos, y dichos resultados se presentan a través de gráficas y reportes. La Figura 4.2 muestra la estructura general en la que se presentan los resultados en la consola de eventos y se compone de tres elementos: un mensaje de alerta, una gráfica con la actividad de un periodo determinado y el reporte correspondiente al mismo periodo.

Generalmente, el usuario deseará obtener los resultados de los últimos minutos de actividad, y con una simple consulta a la consola de eventos podrá obtener una gráfica con la actividad del sistema correspondiente a este periodo de tiempo. Para un mayor detalle se presenta un reporte con todas las características de la actividad del mismo periodo; para hacer completa la visualización de los resultados se presenta un mensaje de alerta que representa el estado de la variable que se está analizando. Sin embargo, el usuario puede tener interés en obtener la actividad de algún periodo de tiempo anterior a la fecha actual y para este caso, la interfaz de usuario presenta gráficas y reportes históricos con la misma estructura de la Figura 4.2, eliminando únicamente el mensaje de alerta, ya que este mensaje es solamente para indicar el estado de la variable analizada en tiempo real.

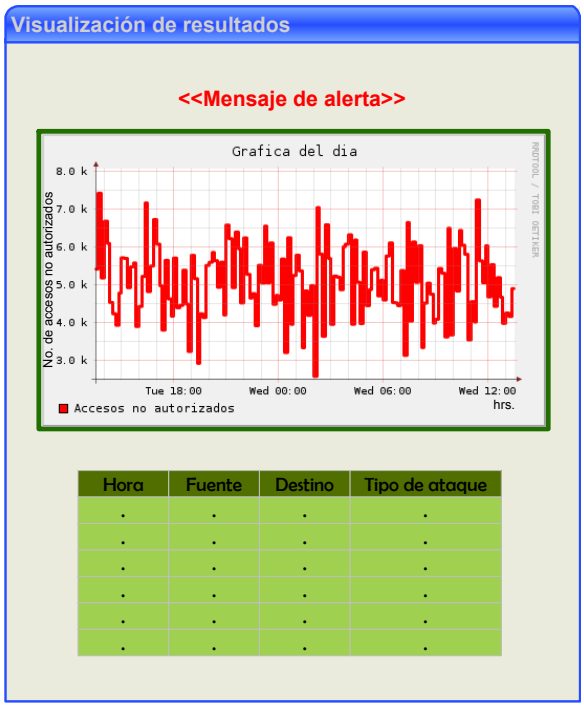


Figura 4.2: Estructura general de la visualización de los resultados

4.2. Consola de eventos del sistema en modo de tiempo real

Como ya se mencionó, la organización que tiene la consola de eventos del sistema en modo en tiempo real es la misma que se siguió para la implementación del sistema de este trabajo de tesis y está formada por tres sitios web: el sitio web del SDIh, el sitio web del SDIr y el sitio web para el módulo de interconexión. A continuación se explica con detalle el funcionamiento de cada uno de ellos.

4.2.1. Sitio web para el SDIh

En este sitio web se presenta al usuario toda la información relacionada con el análisis de la actividad de todas las variables de la lista de variables de la Tabla 3.1 analizada en el capítulo anterior, que corresponden al huésped en donde se encuentra instalado dicho sitio y el SDIh por supuesto.

En cada huésped se encuentra instalado una copia de este sitio web y funciona con la información que le es proporcionada por el huésped correspondiente. Dentro de las funcionalidades con las que cuenta se puede mencionar:

- A través de la pantalla principal presenta mensajes de alerta que indican el estado del sistema.

- Presenta un breve reporte de todas las variables de la lista de variables con información de su estado, de los umbrales y periodos de recolección elegidos para dicha variable.

- De igual manera, se presentan gráficas históricas de cada una de las variables en la lista de variables generando automáticamente una gráfica del año actual, del mes actual, de la semana actual y del día actual, además de una gráfica diaria con los veinte clientes con el mayor número de conexiones al huésped.

- Existe la posibilidad de que el usuario desee obtener una gráfica con un periodo de tiempo diferente al de las gráficas presentadas por lo que se permite generar gráficas de cualquier periodo de tiempo.

- Se pueden configurar los umbrales y los periodos de recolección dependiendo de varios factores, como el tipo de variable, lo vulnerable que se encuentre, entre otros. Por lo que esta decisión será del administrador.

El mapa general del sitio web es el mostrado en la Figura 4.3 y básicamente se encuentra formado por cuatro componentes: la página principal en la cual se presenta un pequeño resumen de las variables en la lista de variables; la página de detalle y gráficas históricas que contiene las gráficas históricas de cada variable, la página de gráficas por fechas específicas que permite generar gráficas de cualquier periodo de tiempo y la página de configuración de umbrales y periodos de recolección que permite al administrador establecer los valores numéricos para estos dos parámetros.

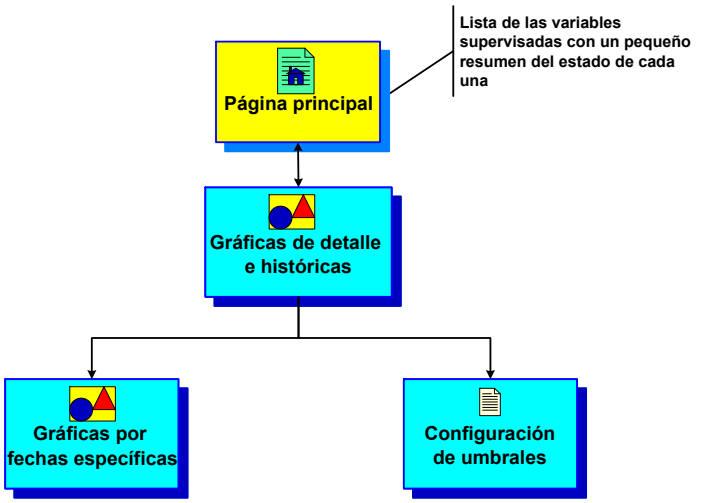


Figura 4.3: Mapa del sitio del SDIh

Los resultados y la evaluación que se presentan a continuación fueron obtenidos del servidor delfos que pertenece a la red del Cinvestav.

Página principal. En la página principal del sitio web del SDIh se encuentra una lista con las variables que se están supervisando. Cada uno de los componentes de esta lista es un liga a la página de detalle y gráficas históricas de la misma variable del sitio web. Además hay un indicador del estado de la variable —estado normal, preventivo o crítico—, y los valores establecidos para los umbrales y los periodos de recolección.

La Figura 4.4 presenta todos los elementos mencionados: el color de la letra con que se encuentran listadas cada una de las variables puede ser diferente; verde para el estado normal, amarillo para el estado preventivo y rojo para el estado crítico de la variable —En la Figura 4.4 se muestran los colores verde, amarillo y rojo, indicando el estado normal en algunas variables, el estado preventivo en una variable y el estado crítico en dos de ellas—.

Figura 4.4: Página principal

Página de detalle y gráficas históricas. En esta página se presentan las gráficas históricas de toda la actividad de la variable elegida desde el momento en que empezó el funcionamiento del sistema. En este apartado se presenta una gráfica del día actual, una gráfica de la semana actual, una gráfica del mes actual y una gráfica del año actual. Estas gráficas se actualizan cada hora, a excepción de la gráfica diaria que se actualiza cada diez minutos o en el instante que el usuario lo requiera.

En el servidor delfos se detectó un patrón desde el primer mes en que se instaló el sistema: en alguno de los días de cada fin de semana en la madrugada algún usuario malicioso intentaba, mediante algún programa diseñado para esto, encontrar un usuario y una contraseña válidos para ingresar al sistema. La gráfica de la Figura 4.5 muestra la actividad de la variable "Intentos de acceso con nombres de usuarios inexistentes" del mes de enero del 2006; aquí se ve claramente que el sistema es atacado cada fin de semana y en ocasiones en días de semana también.

Simultáneamente, el mismo usuario malicioso intenta acceder al sistema tratando de averiguar la contraseña de algún usuario conocido como por ejemplo *root o admin*;

Figura 4.5: Gráfica del mes de enero del año 2006 de la variable "Intentos de acceso con nombres de usuarios inexistentes"

Figura 4.6: Gráfica del mes de enero del año 2006 de la variable "Intentos de acceso con passwords incorrectos"

este ataque se da al mismo tiempo que el anterior en todas las ocasiones como muestra la Figura 4.6.

Este mismo patrón ha sido constante durante los meses en que el sistema ha sido probado —desde enero del año 2006 hasta la fecha— y representa uno de los tantos ataques que un servidor u otra máquina pueden sufrir. Este ataque es bastante simple de llevar acabo y si el atacante logra conseguir una cuenta y una contraseña válidos ingresará al sistema y probablemente intente llevar a cabo un ataque más serio.

El sistema tiene un periodo de respuesta máximo de diez minutos —en el peor de los casos—, que es el periodo de recolección de datos que se eligió para el procesamiento de las bitácoras. Transcurrido este periodo, se hace el preprocesamiento de las bitácoras, se extraen los contadores de cada variable, se actualiza la base de datos del huésped y se actualizan las gráficas correspondientes. Finalmente el módulo de alertas busca por anomalías y si encuentra alguna, avisa al agente de interconexión del sistema para que éste correlacione la información del huésped con

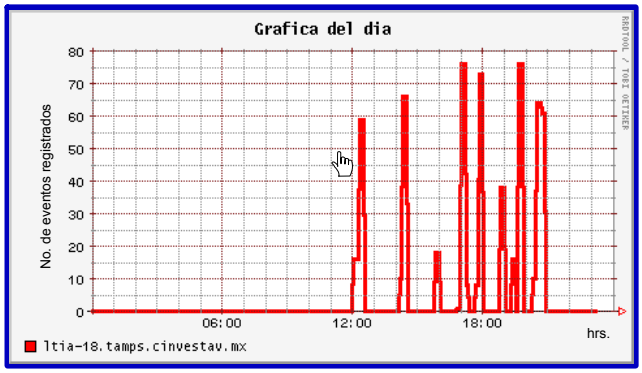


Figura 4.7: Gráfica diaria de detalles

la información de la red y calcule la magnitud del ataque. Sin embargo, el sistema también está preparado para reaccionar de manera inmediata si el usuario así lo indica mediante una orden a través de este sitio web, así el usuario puede actualizar los datos del agente de huésped para que detecte lo que ha pasado en los últimos minutos y no tener que esperar a que transcurra el periodo de respuesta. Los últimos datos se procesarán, y éstos serán parciales, ya que se irán complementando con la información más reciente de las bitácoras hasta cumplirse el periodo de respuesta.

Dentro de las gráficas que se presentan en este apartado, se encuentra una gráfica adicional donde se presenta la información del día actual. Esta gráfica se actualiza cada diez minutos —o cada vez que el usuario lo indique— y simultáneamente el módulo de alertas hace la verificación de las desviaciones de los umbrales. Si algún valor de alguno de los contadores de las variables en la lista de variables excede el límite de alguno de los umbrales se lanza una alerta. Esta gráfica, mostrada en la Figura 4.7, se presenta con mayor detalle que el resto de las gráficas históricas de esta página, ya que contiene la actividad de la dirección IP fuente con mayor actividad hacia el huésped, y si se da un clic sobre la Figura 4.7 se obtiene una gráfica con las veinte direcciones IP fuente con mayor actividad hacia el huésped.

Página de gráficas de cualquier intervalo de tiempo. También existe la posibilidad de que el usuario desee graficar otros intervalos de tiempo diferentes a los que se encuentran predeterminados. Para hacerlo el usuario simplemente tiene que llenar el formulario (ver Figura 4.8), indicando el mes, día, hora y minuto de inicio y de fin. Al dar clic en el botón *Graficar* le mostrará la gráfica en el intervalo de tiempo que haya elegido.

Página de configuración de umbrales y periodos de recolección. También se considera de gran utilidad el permitir al usuario configurar los umbrales y los periodos de recolección con que se realizarán las gráficas. Para realizar esta tarea, el usuario simplemente tiene que elegir los valores numéricos que desea asignar a los umbrales y a los periodos de recolección en el formulario que se muestra en la Figura 4.9

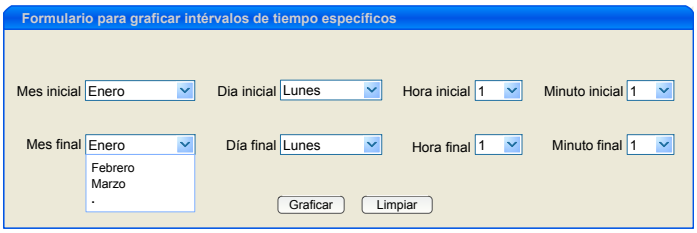


Figura 4.8: Formulario para capturar los periodos de graficación que el usuario ingrese

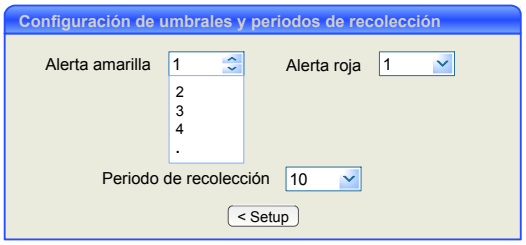


Figura 4.9: Formulario para configurar los umbrales y los periodos de recolección

Este sistema funciona con dos umbrales para cada variable de la lista de variables definidos con base en la experimentación con el comportamiento de cada una de estas variables. La Tabla 4.1 muestra los umbrales elegidos para el agente de huésped, por ejemplo la variable 1 y 2 son muy sensibles y por ello sus umbrales son de magnitud pequeña; un usuario legítimo puede equivocarse de contraseña un número razonable de veces —digamos tres— y un número más de equivocaciones puede representar un intento de acceso por parte de un usuario ilegítimo. Para el caso de los correos es un poco más complicado la definición de los umbrales, ya que los patrones de uso pueden depender de varios aspectos: como el número y tipo de usuarios con los que cuenta el servidor, el número de correos promedio que reciba y envíe cada uno de los usuarios, el tamaño promedio de dichos correos, entre otros. Con base en estos parámetros y a la experimentación se eligieron los umbrales para las variables que tienen que ver con el servicio de correo —variables 3,4,5 y 6—. Como el tamaño de los correos está medido en bytes, los umbrales para la variable 7 también están dados en esta medida. Las variables que tienen que ver con el servidor web también dependen de ciertos indicadores, como la cantidad de sitios y servicios que ofrece a través de estos sitios el servidor web; en el caso de la variable 8 los umbrales son un poco altos debido a las condiciones del servidor ya que contiene los sitios web del sistema de monitoreo y detección de intrusos así que tiene una alta demanda de accesos a sus páginas. Los umbrales de la variable 11 fueron definidos tomando en cuenta el número de usuarios con los que cuenta el servidor —diez aproximadamente— y estimándose dos sesiones por usuario.

ID	Variable	Umbral amarillo	Umbral rojo
1	Acceso con nombres de usuarios incorrectos	3	5
2	Acceso con contraseñas incorrectas	3	5
3	Correos enviados al servidor	20	25
4	Correos enviados desde el servidor	30	40
5	Correos enviados a usuarios inexistentes	10	15
6	Conexiones perdidas con el servidor de correo	5	8
7	Tamaño de los correos	20480 (bytes)	25600 (bytes)
8	Acceso a páginas del servidor 50	60	
9	Acceso a páginas inexistentes	5	8
10	Acceso a directorios prohibido	3	5
11	Número de sesiones activas	20	25

Tabla 4.1: Lista de umbrales usados para cada variable

4.2.2. Sitio web para el SDIr

En este sitio web se presenta toda la información relacionada con la red supervisada —en este caso es la red del departamento de Computación—. Es un solo sitio instalado en la misma computadora donde se encuentra el SDIr y es común a todos los huéspedes que conforman la red. Este sitio cuenta con varias funcionalidades para el administrador de la red, entre ellas se pueden mencionar:

- A través de la página principal se presentan alertas correspondientes a los patrones de uso anormal analizados por el SDIr como los ataques de exploración de puertos y de direcciones IP. También se presenta un breve resumen con el estado de estos patrones y con los umbrales y periodos de recolección asignados a éstos.

- Presenta los reportes referentes a los ataques de exploración de puertos y exploración de direcciones IP en caso de que estos sucedan. Este reporte es bastante general y proporciona la dirección IP fuente de donde proviene el ataque, las direcciones IP destino de la red supervisada que están siendo exploradas y si el administrador quiere más detalles tiene acceso a toda la información del tráfico de la red.

- Al igual que el sitio web del SDIh, el sitio web del SDIr presenta gráficas históricas del tráfico que circula a través de la red. En este apartado se presentan una gráfica anual, una mensual, una semanal y una diaria con el tráfico de entrada y salida de la red a través del tiempo.

- Debido a que la información del tráfico capturado es excesiva y difícil de manipular, se ideó la forma de presentar reportes por intervalo de tiempo específicos. Cada gráfica es un mapa de imagen y el usuario puede seleccionar con el ratón sobre las gráficas el periodo que desee consultar y obtendrá como resultado la gráfica en acercamiento del periodo que desea consultar y un reporte con toda la actividad de red de ese mismo periodo.

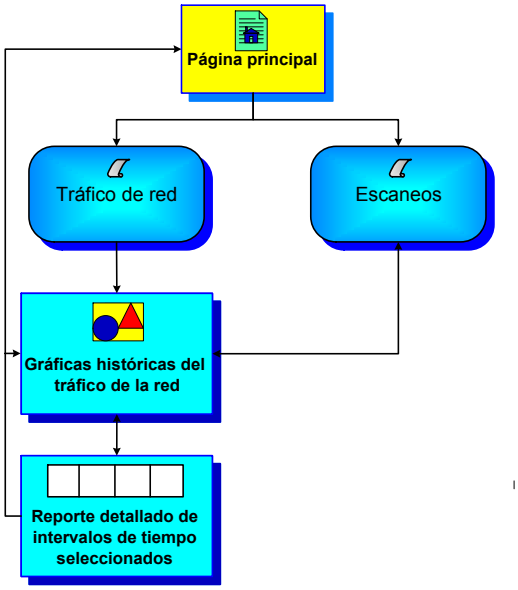


Figura 4.10: Mapa del sitio web del SDIr

El mapa de este sitio web es el mostrado en la Figura 4.10; este sitio se compone de una página principal, y a su vez la página principal se compone de dos elementos principales: la supervisión del tráfico de la red y las exploraciones (tanto de puertos como de direcciones IP). La liga correspondiente a la supervisión del tráfico de la red contiene las gráficas históricas y reportes del tráfico de red y la liga de las exploraciones de puertos o de direcciones IP presentan la información de los ataques de este tipo registrados por el sistema.

Página principal. La página principal (mostrada en la Figura 4.11) está constituida por una pequeña lista de elementos: la liga correspondiente al análisis de tráfico de la red y otras dos ligas correspondientes a las exploraciones de direcciones IP y las exploraciones de puertos.

Igual en la página principal del SDIh, en esta página se presenta de color diferente cada elemento de la lista: verde para el estado normal de la red, amarillo para un estado preventivo y rojo para un estado crítico con base en el comportamiento de la red.

Supervisión de la red. Es la página más importante del sitio web. En ella se presentan todos los detalles del análisis del tráfico de la red a la cual se encuentra protegiendo el sistema de monitoreo y detección de intrusos. Como primer característica importante presenta un conjunto de gráficas históricas del tráfico de la red desde que se instaló el sistema.

Las gráficas se presentan en el siguiente orden: una gráfica diaria, una gráfica semanal, una gráfica mensual y una gráfica anual. Los datos contenidos en estas gráficas

Figura 4.11: Página principal del sitio web del SDIr

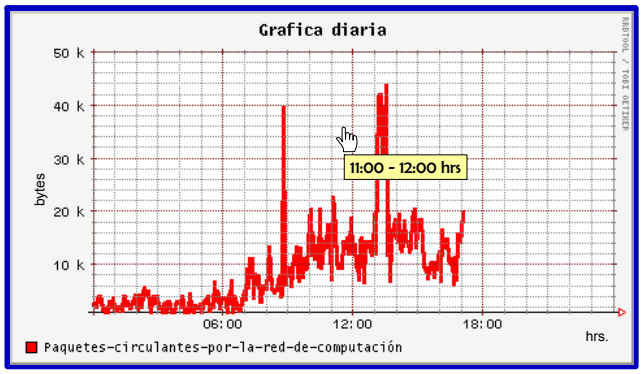
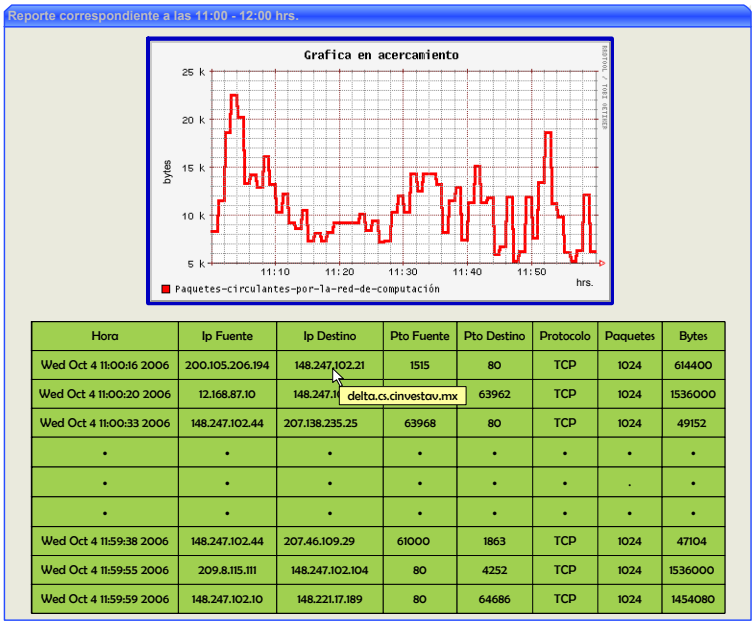


Figura 4.12: Gráfica diaria de la supervisión de la red de Computación

son actualizados cada minuto, que es el menor tiempo de corte del colector flow tools y así se consiguió un periodo de recolección acotado. En la Figura 4.12 se muestra el ejemplo de una gráfica diaria.

Como ya se mencionó, debido a la gran cantidad de datos que se capturan del tráfico de entrada y salida de la red que se está supervisando es excesivo, todas las gráficas tienen una característica útil para localizar datos pertenecientes a un determinado periodo de manera muy sencilla: cada gráfica es un mapa de imagen en el que el usuario selecciona con el ratón la región de la gráfica que desea ver con más detalle (como muestra la Figura 4.12). Así se despliega la gráfica con el periodo seleccionado y un reporte con toda la información referente al tráfico de la red correspondiente a este mismo periodo.

La Figura 4.13 muestra los resultados de la selección de un periodo que comprende el día 4 de octubre de las 11 a las 12 hrs; se presenta la gráfica de este periodo, además de una tabla con la información de la red correspondiente al mismo intervalo de tiempo; esta tabla contiene campos como la fecha y hora exactas, dirección IP



Hora	Ip Fuente	Ip Destino	Pto Fuente	Pto Destino	Protocolo	Paquetes	Bytes
Wed Oct 4 11:00:16 2006	200.105.206.194	148.247.102.21	1515	80	TCP	1024	614400
Wed Oct 4 11:00:20 2006	12.168.87.10	148.247.1		63962	TCP	1024	1536000
Wed Oct 4 11:00:33 2006	148.247.102.44	207.138.235.25	63968	80	TCP	1024	49152
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
Wed Oct 4 11:59:38 2006	148.247.102.44	207.46.109.29	61000	1863	TCP	1024	47104
Wed Oct 4 11:59:55 2006	209.8.115.111	148.247.102.104	80	4252	TCP	1024	1536000
Wed Oct 4 11:59:59 2006	148.247.102.10	148.221.17.189	80	64686	TCP	1024	1454080

Figura 4.13: Reporte detallado de las 11:00 - 12:00 hrs.

fuente, dirección IP destino, puerto fuente, puerto destino, protocolo usado durante la transmisión, cantidad de paquetes transmitidos y cantidad de bytes transmitidos. Con este mecanismo de consulta el usuario puede tener control de toda la información contenida en la base de datos que almacena toda la información de la red desde que el sistema fue instalado, eligiendo consultar cualquier periodo de tiempo que requiera.

Exploración de direcciones IP. Este apartado presenta todos los resultados del algoritmo que se encarga de detectar las exploraciones de direcciones IP. En caso de detectar una exploración de direcciones IP, se presenta un pequeño reporte de la actividad de la dirección IP maliciosa como el mostrado en la Figura 4.14 y ahí mismo se despliega un mensaje de alerta.

Si el administrador quiere más detalles acerca del comportamiento actual de la red para complementar la información que está obteniendo del reporte —mostrado en la Figura 4.14— puede ir a la página que contiene toda la información de la red que despliega de manera funcional los datos del tráfico de entrada y salida de la red de cualquier intervalo de tiempo.

Exploración de puertos. El proceso para detectar las exploraciones de puertos es muy similar al de exploraciones de direcciones IP. Si se detecta alguna exploración de puertos, se genera un reporte con toda la actividad de la dirección IP sospechosa.

El reporte de las exploraciones de puertos se muestra en la Figura 4.15 —además de un mensaje de alerta— y en él, se reflejan todas las conexiones por diferentes

Figura 4.14: Reporte de la exploración de direcciones IP

Variable	Umbral amarillo	Umbral rojo
Escaneo de puertos	10	15
Escaneo de direcciones IP	5	8

Tabla 4.2: Lista umbrales para el SDIh

puertos de una dirección IP a algún huésped de la red supervisada. Cuando este número de puertos sobrepasa algún umbral previamente definido se envía una alerta al administrador de la red.

Configuración de umbrales y periodos de recolección. Al igual que para las variables del agente de huésped, se definieron umbrales para los patrones de uso analizados por el agente de red: exploraciones de puertos y exploraciones de direcciones IP. La Tabla 4.2 muestra los umbrales elegidos para las variables del agente de red. Estos valores fueron elegidos con base en la experimentación, pero como ya se mencionó, la definición de estos valores depende de la naturaleza de las variables y del comportamiento que tengan en la práctica. Los umbrales son valores muy importantes que deben ser definidos con cuidado, ya que de ello depende la cantidad de falsos positivos que el sistema genere.

4.2.3. Sitio web para el módulo de interconexión

El módulo de interconexión es el módulo más importante del sistema de monitoreo y detección de intrusos desarrollado en este trabajo de tesis. El sitio web de este módulo contiene toda la información de los ataques registrados por el sistema. Dentro de las funcionalidades que tiene se encuentran:

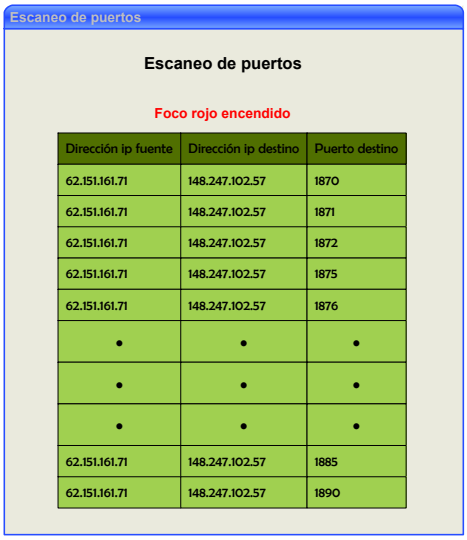


Figura 4.15: Reporte de las exploraciones de puertos

- Genera una gráfica y un reporte correspondiente a un ataque detectado y esta información generada la comparte con los huéspedes involucrados en el ataque.
- Presenta un reporte con toda la información de los últimos ataques registrados y esta información la comparte con todos los huéspedes que componen la red supervisada. Ésto con el propósito de que todos los huéspedes tengan pleno conocimiento de todos los ataques a los cuales pueden estar sujetos y así tomar algunas medidas de prevención ante estas situaciones.

El sitio web del módulo de interconexión consta de una página principal, una página de gráficas y reportes y una página de información compartida —ver Figura 4.1—. las cuales se explican a continuación

Gráficas y reportes de ataques dirigidos a un huésped El módulo de transceptor es el encargado de correlacionar la información proporcionada por las bitácoras del sistema de cada huésped y la información proporcionada por el tráfico circulante a través de la red. Cuando se hacen las detecciones pertinentes, simultáneamente se generan los resultados y se presentan a través de una gráfica y un reporte asociado a ella y funciona de la siguiente manera: después de que transcurre un cantidad regular de tiempo periódico el módulo de interconexión analiza las alertas de cada uno de los huéspedes y si hay alguna alerta disparada a un valor que represente alerta preventiva o alerta crítica lanza una búsqueda a los registros de la base de datos correspondientes a los últimos diez minutos de actividad y genera una gráfica y un reporte describiendo la actividad del ataque. Esta información es presentada en los sitios web del SDIh de los huéspedes involucrados en el ataque. Este reporte contiene además de los datos del ataque, la información de la actividad relacionada con la dirección IP sospechosa dentro de la red.

Figura 4.16: Reporte del módulo de interconexión

Como muestra la Figura 4.16 se despliegan todas las conexiones de una dirección IP origen —que se ha detectado con algún grado de mala intención hacia la red— hacia cualquier huésped de la red. Además se despliega un mensaje de alerta dirigido a los administradores de los huéspedes involucrados en el ataque a través de un mensaje en la pantalla y mediante un correo de alerta a la cuenta de correo de cada administrador de los huéspedes correspondientes.

Información compartida de ataques registrados en la red El módulo de interconexión también cuenta con una página que contiene la información de los ataques más recientemente registrados en toda la red y esta página puede ser accedida por todos los huéspedes que conforman esta red. Cuando se registra un ataque, los datos de éste son almacenados en un reporte y es presentado en esta página. Este reporte que se muestra en la Figura 4.17 contiene la hora exacta del registro del ataque, la dirección IP origen que está atacando a algún o algunos huéspedes de la red, la dirección IP del huésped que está recibiendo el ataque y el tipo de ataque registrado.

Esta página tiene el propósito de mantener informados a los administradores de todos los huéspedes de lo que sucede en la red, y no centralizar la información solamente a los huéspedes involucrados en algún ataque en particular.

Figura 4.17: Reporte con la información de los ataques recientes

4.3. Consola de eventos para el sistema en modo de simulación

Como ya se explicó, el sistema de monitoreo y detección de intrusos se realizó en dos modos: el modo normal —o en tiempo real— y el modo de simulación. Aunque el sistema en modo normal está en evaluación desde el mes de enero del 2006 aproximadamente, se implementaron las variaciones necesarias para hacer funcionar el sistema en modo de simulación, esto con el fin de hacer pruebas exhaustivas y verificar casos especiales que no pudieron ser probados en el sistema en modo normal. Así pues, esta etapa fue de mucho provecho, ya que se observaron algunos errores y fallas que pudieron ser detectados a tiempo y dieron la oportunidad de arreglar estos fallos en el sistema en modo normal. Se hicieron muchas pruebas con diferentes conjuntos de datos obteniendo buenos resultados de funcionalidad y desempeño en el sistema.

En la Figura 4.18 se observa el mapa del sitio web para el sistema en modo de simulación. Este sitio web está formado de una página principal, la página de reportes y detalles, una página de configuración de umbrales y periodos de recolección, a diferencia del sitio web del sistema en modo normal que cuenta con una página donde el administrador puede simular los ataques que desee.

El sitio se encuentra bien conectado con el resto de las páginas; la página principal puede conectarse con el resto de la páginas —y viceversa— debido a que las ligas de éstas se encuentran en la página principal. Además una vez realizada la simulación del ataque se puede verificar los resultados en la página de gráficas y reportes.

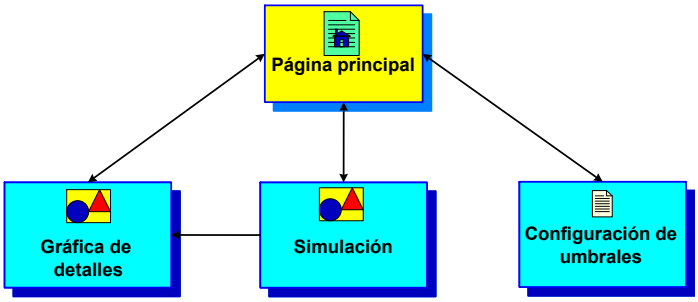


Figura 4.18: Mapa del sitio web para el sistema en modo de simulación

Las funcionalidades de este sistema son muy parecidas al sitio web del sistema en modo en tiempo real, pero en este sitio se encuentran algunas variaciones:

- Cuenta con un conjunto de formularios que permiten al usuario describir el ataque que quiere simular y cuando estos datos son registrados se genera automáticamente toda la información simulada del ataque deseado como las bitácoras del sistema y la información del tráfico circulante a través de la red.

- Como ya se mencionó, este sitio web trabaja con un reloj de simulación implementado para este módulo, en vez de utilizar el tiempo del sistema.

- Los datos del ataque simulado son visualizados mediante una gráfica y un reporte de detalle; esta información es obtenida con rapidez debido a que el reloj de simulación avanza más rápido que el reloj del sistema, ésto con el fin de no consumir tanto tiempo haciendo pruebas exhaustivas o esperando visualizar los resultados.

Las principales diferencias en el funcionamiento de este sitio con el sitio en modo normal son el uso de un reloj de simulación y los datos con los que el sistema trabaja son generados por el mismo administrador.

El administrador del sistema o la persona encargada de manejar este sitio web puede generar ataques de cualquier tipo, tamaño y en cualquier periodo de tiempo. Lo único que tiene que hacer es llenar el formulario mostrado en la Figura 4.19.

Con base en este formulario el usuario tiene que introducir los datos que describan el ataque que desea simular; necesita introducir el número de intentos de intrusión que desea hacer, los cuales se traducen en eventos que serán registrados por el sistema. Tiene que definir el periodo de tiempo en que desea que se presente el ataque que se está simulando, y con base en este periodo el número de intentos de intrusión se distribuirá a lo largo de él. También se tiene que especificar la dirección IP fuente que será la fuente del ataque y por último el puerto fuente que representa al puerto de la máquina origen que desea realizar algún tipo de actividad sospechosa en contra del sistema.

Al pulsar el botón simular se inyectarán estos datos al sistema de monitoreo y detección de intrusos. Con estos datos se generarán las bitácoras que funcionarán como fuente de

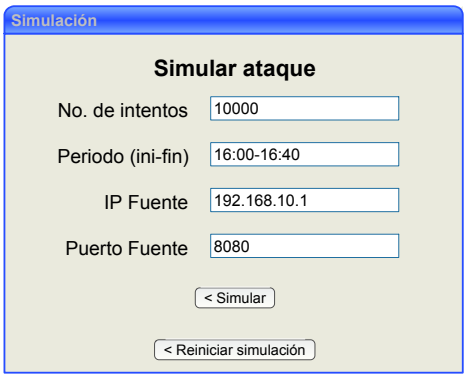


Figura 4.19: Formulario para simular ataques

información para el SDIh y también se generarán los datos de tráfico necesarios que se almacenarán en la base de datos de la red y serán la fuente de información del SDIr.

Por ejemplo, la Figura 4.19 muestra los resultados de un ataque de diez mil eventos, en un periodo de tiempo entre las 4:00 y 4:40 p.m., de una dirección IP fuente 192.168.10.1 desde el puerto 8080. Como el reloj de simulación avanza tan rápido como el usuario haya elegido —en este caso se hizo la correspondencia de 1 segundo → 1 minuto—, los resultados pueden ser vistos con mucha más rapidez que si se tuviera que esperar que el tiempo avanzara a su ritmo natural. Los resultados de generar el ataque anterior son presentados en una gráfica y un reporte como se ha venido presentando en todos los módulos.

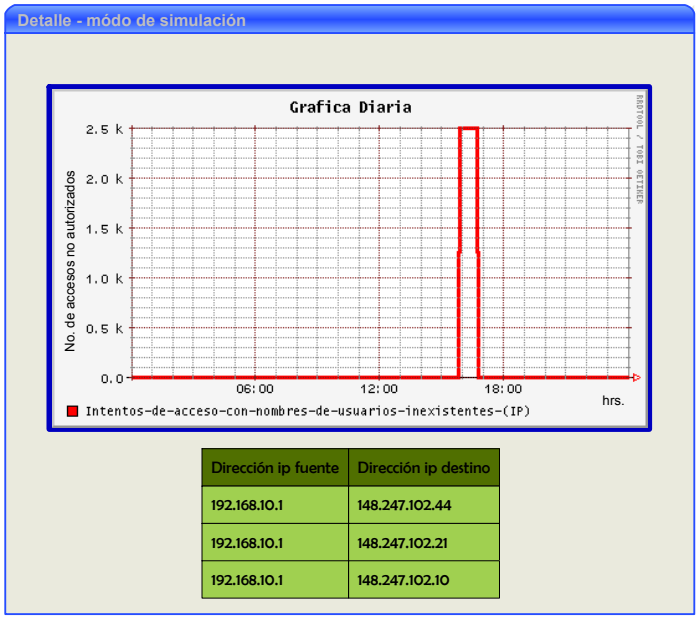


Figura 4.20: Gráfica y reporte de un ataque simulado por el administrador

4.4. Pruebas en el modo de simulación

El sistema de monitoreo y detección de intrusos en modo de tiempo real se encuentra en funcionamiento desde hace algunos meses como ya se mencionó, pero debido a que este sistema trabaja con datos reales, generados por las actividades que ocurren en el sistema, no se puede tener control sobre la información que se desea analizar para ciertas pruebas en un momento dado. Todos los resultados obtenidos del sistema en modo en tiempo real fueron generados por la actividad real de la red supervisada, aunque se indujeron algunos pequeños ataques como exploración de contraseñas y de usuarios.

Con el sistema en modo de simulación se hicieron diversas pruebas, la primera de ellas fue usar más de una máquina y ver como se comportaba el sistema en un ambiente distribuido. Dichas pruebas se hicieron para cada variable y se hicieron en una red conformada por dos computadoras portátiles: una IBM Think Pad celeron a 750 Mhz y una Dell Inspiron 630m pentium 4 a 1.7 Ghz. A ambas computadoras se les instaló el SDIh y a la computadora Dell Inspiron se le instaló el SDIr que supervisa la pequeña red. Y finalmente el módulo de interconexión también fue instalado en las dos computadoras portátiles.

Una vez probado el sistema en la pequeña red construida para este propósito se hicieron pruebas para comprobar su buen funcionamiento para cada uno de los casos para los que fue diseñado. La idea fue probar caso por caso cada una de las variables y se siguieron los siguientes pasos:

- Se construyó un *script* que se encarga de generar toda la información de los ataques que se simularán durante todo un día de actividad con base en el tiempo del reloj de simulación y tomando algunos datos que el usuario le proporciona.

- La información generada es almacenada en un archivo de texto y de aquí es tomada para que el sistema en modo simulación funcione adecuadamente y se generen todas las gráficas y reportes correspondientes a los ataques.

- Se generan todas las alertas de todos los ataques registrados de igual forma que en la interfaz en modo normal.

- Los umbrales utilizados para este sistema en modo simulación fueron los mismos que se usaron para el sistema en modo en tiempo real.

El Algoritmo 6 se utiliza para generar los datos de las pruebas hechas para el modo de simulación; este algoritmo genera información de todo un día de actividad y obedece a dos parámetros proporcionados por el usuario: el tipo de ataque y el archivo donde se almacenarán estos datos. La variable n indica el número de eventos registrados para una hora en particular y es generado de manera aleatoria. La información que describe al ataque simulado se almacena en un archivo de texto y esta información es tomada gradualmente para generar las bitácoras y la información del tráfico de la red para que el sistema en modo simulación comience a funcionar.

Algoritmo 6 Algoritmo generador de simulaciones

Entrada: *Tipo_ataque*, *Archivo_descripcion_ataques*.
Salida: *Archivo_descripcion_ataques* con los datos de entrada para el sistema.

```
1: for i=0; i<1440; i++ do
2:     hora = hora + (i * 60)
3:     n = random()
4:     print Archivo_descripcion_ataques, hora Tipo_ataque ip_origen puerto_origen n
5: end for
```

Con el Algoritmo 6 se pudieron generar tipos de información que no se pudieron detectar de las bitácoras reales del sistema ni del tráfico de paquetes que circulan a través de la red en el momento requerido para las pruebas. Con la información simulada se realizaron pruebas para todos los tipos de variables en la lista de variables. La gráfica de la Figura 4.21 muestra las pruebas realizadas para la variable "Intentos de acceso a directorios prohibidos", en esta gráfica se presenta todo un día de actividad con datos de simulación para dicha variable generados con el Algoritmo 6 y también se hizo la verificación de la generación de alertas y ésta fue correcta.

Figura 4.21: Gráfica con datos de simulación para todo un día de actividad

4.5. Resumen

En resumen se desarrolló una interfaz de usuario, la cual no forma parte técnica del esquema del sistema de monitoreo y detección de intrusos pero es sin duda una parte importante al momento de interpretar los resultados y tomar decisiones que ayuden a contrarrestar los intentos de intrusión y ataques que se presenten en la red.

Toda esta interfaz se divide en tres sitios web: el sitio web del SDIh, el sitio web del SDIr y el sitio web del módulo de interconexión. Cada uno de estos sitios puede trabajar de manera autónoma e igualmente funcional y eficiente.

Además se realizó un sitio web para el sistema de monitoreo y detección de intrusos en modo de simulación. Este sitio tiene casi las mismas características que la interfaz del sistema en modo normal, excepto que se alimenta de datos simulados —inducidos por el usuario— y toma el tiempo de un reloj de simulación, en vez de tomar el tiempo real del sistema.

Todos estos sitios fueron de mucha utilidad para evaluar el sistema en sus tres módulos principales. Por un lado el SDIh se encuentra en evaluación desde enero-febrero del 2006 en un servidor del Cinvestav, y los resultados de desempeño, rapidez y funcionalidad son satisfactorios. El sistema presenta un equilibrio entre el número de variables analizadas en cada huésped y el número de patrones analizados en la red, por lo cual no hay sobrecarga del sistema. Por otro lado, el SDIh es de los dos agentes, el que gasta menos recursos, sin embargo el SDIr se ejecuta con bastante rapidez ya que fue implementado con el protocolo sFlow. El sistema mostró en todas las pruebas tanto en modo en tiempo real como en modo de simulación funcionar de manera correcta, calculando las alertas y dando respuesta en el momento adecuado.

Por otra parte el SDIr se encuentra en evaluación desde el mes de junio del año 2006. El sitio web para este módulo se encuentra en uso desde el mes de agosto del 2006 sobre el servidor delfos y los resultados de funcionalidad y facilidad de búsqueda son satisfactorios.

El sitio web para el módulo de interconexión se encuentra también en funcionamiento desde el mes de agosto del año 2006 y con éste se han hecho diversas pruebas para verificar que establezca la comunicación de manera correcta entre el SDIh y el SDIr. A pesar de que este módulo hace uso de la información almacenada para el SDIh y para el SDIr, no existe sobrecarga alguna, ya que el módulo de interconexión analiza información que ya fue procesada y clasificada por cada uno de los SDI, así pues la búsqueda y análisis de esta información es mucho más fácil. Mejor aún, como el sistema trabaja en tiempo real sólo procesa la información de los últimos minutos y la almacena en su historial.

El sistema en modo de simulación fue muy útil para realizar pruebas exhaustivas y evaluar casos especiales. Las pruebas con este sistema se realizaron en una pequeña red y ayudó a considerar aspectos que no habían sido considerados en el sistema en modo normal. De este modo conforme se iban observando errores o carencias del sistema en modo de simulación se fueron corrigiendo en el sistema en modo normal.

Capítulo 5

Conclusiones y Trabajo Futuro

En este trabajo de tesis se diseñó e implementó un SDI híbrido que se basa en supervisar actividades que tienen que ver con el sistema operativo y con las aplicaciones que se ejecutan en un solo huésped añadiendo la cualidad de analizar el tráfico de entrada y salida de la red supervisada para complementar la calidad de las detecciones.

Existen diversos mecanismos para prevenir y detectar usos indebidos en un sistema. Dentro de los mecanismos de detección se encuentran los Sistemas de Detección de Intrusos que supervisan un sistema en busca de alguna actividad maliciosa y responden a ésta. Los SDI se clasifican en dos grandes tipos: los SDI basados en huésped y los SDI basados en red. Esta clasificación no es excluyente, ya que es posible combinar ambos aspectos para obtener un alto nivel de seguridad obteniendo un SDI híbrido; lo que se hace es complementar la tecnología de los SDIh añadiendo la capacidad de supervisar el tráfico de red desde o hacia el equipo que vigila.

El sistema consiste principalmente de tres módulos:

- Un agente de huésped que se encarga de procesar las bitácoras del sistema de un servidor obteniendo y almacenando la información de variables que representen el comportamiento de alguna actividad maliciosa.
- Un agente de red que se encarga de supervisar el tráfico de entrada y salida de una red mediante el protocolo sFlow. La información es recolectada y almacenada en una base de datos de red.
- Un módulo de interconexión que se encarga de correlacionar la información recolectada y analizada por los dos agentes anteriores para así determinar la magnitud del ataque y alertar a los huéspedes involucrados además de prevenir al resto de huéspedes de la red del ataque.

El sistema de monitoreo y detección de intrusos se implementó en dos modos: en modo en tiempo real y en modo de simulación. El sistema en tiempo real fue implementado para funcionar con las bitácoras de un sistema Linux y el tráfico de entrada y salida de una red como fuentes de información, además de ser ejecutado con el tiempo del sistema; el sistema en modo de simulación fue implementado de manera muy similar al de tiempo

real, a diferencia que el sistema en modo de simulación funciona con bitácoras y datos de tráfico simulados por el propio administrador, además funciona con un reloj de simulación independiente del tiempo del sistema. El sistema en modo de simulación se hizo con el fin de corregir errores de implementación y hacer pruebas exhaustivas para todos los casos para los cuales fue diseñado el sistema.

Para facilitar la lectura y comprensión de los resultados obtenidos de las detecciones hechas se construyó una interfaz *web* en la que se presenta toda la información de la actividad del sistema necesaria para que el administrador tome las decisiones correspondientes como medidas de prevención y recuperación del sistema, además de proporcionar una vista amplia del comportamiento del sistema.

Esta interfaz ofrece varios servicios al usuario: en primer lugar se presenta el comportamiento de cada huésped y de la red en general a través del tiempo mediante gráficas y reportes históricos; además se pueden generar gráficas y reportes con diferentes periodos de tiempo requeridos por el usuario; así mismo se visualizan todas las alertas generadas por el sistema a través de mensajes de texto en la pantalla; finalmente el usuario puede configurar ciertas opciones para el funcionamiento del sistema como los umbrales y los periodos de recolección para cada variable. Se implementaron dos interfaces de usuario: una para el modo en tiempo real y otra para el modo de simulación; la primera sirvió para observar el buen funcionamiento del sistema y la segunda sirvió para toda la fase de pruebas.

El sistema implementado en este trabajo de tesis está diseñado como una solución robusta para detectar intrusos en un sistema, con la capacidad de supervisar huéspedes mediante un agente de huésped y redes mediante un agente de red. Esta supervisión se puede hacer de manera simultánea mediante un agente de interconexión que se encarga de combinar las cualidades de cada uno de los agentes mencionados y adicionalmente la supervisión puede hacerse de manera autónoma donde cada uno de los agentes no se comunican entre ellos y supervisan el sistema que tengan asignado.

Este sistema presenta una arquitectura distribuida capaz de proveer exactitud en las detecciones y el manejo de la seguridad de un sistema relacionado con actividades anómalas. Por un lado el agente de red proporciona al sistema información general de algún ataque detectado en la red y el agente de huésped proporciona el rastro del origen del ataque, aunque el usuario malicioso sea tan astuto de borrar su rastro de las bitácoras, ya que el sistema registra la actividad del sistema en tiempo real. La combinación de estas dos características resulta en un sistema que logra aumentar la dificultad de que las actividades de un usuario malicioso tengan éxito.

A pesar de que el sistema implementado dio buenos resultados de desempeño y exactitud en las detecciones, cuenta también con ciertas limitantes:

- El número de patrones extraídos es pequeño en comparación del número de posibles patrones que podría extraerse con toda la información disponible sobre todo en el agente de red. En cuanto mayor sea el número de patrones analizados, mejorará el rango o gama de las intrusiones que se pueden detectar, sin embargo, hay que saber encontrar un balance entre cantidad de patrones analizados y sobrecarga del sistema para no abusar de los recursos de éste.

- La definición de umbrales es manual por parte del administrador del sistema, siendo éste el responsable de una buena elección que genere el menor número de falsas alarmas. Sin embargo, sería de mucha utilidad algún método automático e inteligente para la definición de umbrales que ayude a tomar la decisión al administrador acerca de los umbrales correctos.

- La capacidad de prevención del módulo de interconexión está limitada a compartir la información de los ataques con todos los huéspedes de la red, sin embargo, esta capacidad se podría aumentar analizando y obteniendo patrones de la información de estos ataques para predecir un comportamiento futuro y proporcionar esta información al administrador o al propio sistema para tomar la decisión más adecuada.

- El módulo de respuestas es pasivo, es decir las respuestas que generan sólo avisan de algún modo al administrador del sistema de algún ataque detectado y sería conveniente añadir algunos tipos de respuestas activas a este módulo como bloquear puertos o direcciones IP de usuarios maliciosos.

5.1. Trabajo a futuro

Debido a las limitantes expuestas anteriormente, es posible extender el trabajo desarrollado de la siguiente manera:

- Aumentar el número de patrones analizados en los dos agentes —huésped y red—. Por parte del agente de red hay muchos patrones que aún se pueden analizar, como los numerosos tipos de ataques de negación de servicio que existen.

- Añadir métodos automáticos e inteligentes para la definición de umbrales, los cuales van aprendiendo de los patrones analizados y con base en este aprendizaje y a la experiencia definen umbrales para cada variable.

- Usar métodos para el módulo de interconexión que analicen los ataques detectados por el sistema y obtengan datos estadísticos que sean de mucha utilidad, esto con el fin de aumentar la capacidad de prevención para los sistemas que conforman la red. Entre estos patrones se pueden mencionar; los n ataques más peligrosos, los ataques más comunes, los huéspedes más atacados, patrones de hora y fecha de los ataques presentados, entre otros patrones.

- El módulo de respuestas puede ser activo: es decir, emprender alguna acción en contra del atacante, por ejemplo, cerrar los puertos o bloquear el acceso a una dirección IP sospechosa. Sin embargo, para emprender estas acciones hay que asegurarse que se trata efectivamente de un ataque y no de una falsa alarma ya que sería un acto irresponsable causar algún daño a un usuario legítimo.

Apéndice A

Sistema de Monitoreo y Detección de Intrusos en Servidores Linux

Manual de Usuario

A.1. Requerimientos

El sistema de monitoreo y detección de intrusos en servidores Linux, como su propio nombre lo indica corre bajo cualquier distribución de Linux. Las pruebas se han hecho en tres distribuciones de éste sistema operativo: en un Red Hat 9.0 núcleo 2.4.20-8, un Fedora core 3 núcleo 2.69-1.667 y un Fedora core 5 núcleo 2.6.17.1.2157.

Además el sistema de monitoreo y detección de intrusos necesita de varios paquetes, de lo contrario no funcionará de manera adecuada. Estos paquetes son:

- Perl
  - perl5.8.0.tar.gz
- MySQL
  - mysql-standard-4.1.21-pc-linux-gnu-i686.tar.gz
  - mysql-server-4.1.21-i386.tar.gz
  - mysql-client-4.1.21-i386.tar.gz
- Apache
  - httpd-2.0.54.tar.gz
- RRDtool
  - rrdtool-1.0.49.tar.gz

- sFlowToolkit
  - sflowtool-3.8-linux.tar.gz
- Flow Tools
  - flow-tools-0.68-1.i386.rpm

A.2. Componentes

El árbol de directorios del sistema se compone de dos carpetas principales y de cada una de estas se derivan las subcarpetas correspondientes:

- `/`
  - `mchable`. Esta carpeta contiene los *scripts* que no necesitan ir ubicados en la carpeta de los cgi's.
    - `colector`. Esta carpeta contiene el colector de la red.
    - `recortado_logs`. Estos son los *scripts* que leen en tiempo real los archivos de bitácora.
- `cgi-bin`
  - `mchable`. Esta carpeta contiene todo el sitio web y los *scripts* que se requieren este ubicados en esta carpeta debido a que son ejecutados mediante el sitio web.
    - `huesped`. En este carpeta se encuentran todos los *scripts* que procesan a los archivos de bitácoras y las bases de datos que almacenan esta información. Además se encuentran los *scripts* del módulo de interconexión.
    - `red`. Esta carpeta contiene todos los *scripts* que hacen los análisis en la red como el escaneo de puertos y direcciones IP.
    - `simulacion`. Esta carpeta contiene todos los *scripts* que hacen el sistema de monitoreo y detección de intrusos en modo de simulación. Entre estos *scripts* se encuentra el que genera el reloj de simulación, el que genera las bitácoras ficticias entre otros.
    - `huesped_sitio`. En esta carpeta se encuentran todos los cgi's que constituyen la interfaz web del huésped y también del módulo de interconexión.
    - `red_sitio`. Aquí se encuentran todos los cgi's que constituyen la interfaz web de la red.
    - `simulacion_sitio`. Esta carpeta contiene todos los cgi's que constituyen la interfaz web del sistema en modo de simulación.

Estas carpetas se encuentran organizadas con los mismos nombres y rutas en el CD del Sistema. La carpeta que contiene a todas estas carpetas es la llamada *Sistema*.

A.3. Iniciar la ejecución del SDIh&r

Antes de echar a andar el sistema de monitoreo y detección de intrusos se tiene que crear la base de datos que contendrá toda la información de la red. Para su creación sólo se tiene que teclear en alguna consola de comandos la siguiente línea:

```
mysql < uso_red.sql
```

El archivo uso_red.sql es el archivo donde se encuentra descrita toda la estructura de la base de datos se encuentra en el directorio home/mchable/colector.

Además se tienen que programar todas las tareas de recolección y procesamiento de los archivos de bitácora. Para ello tenemos que indicar al crontab o a alguna otra herramienta de administración de procesos que queremos programar todas las tareas contenidas en el archivo tareas.cron de la siguiente manera:

```
crontab tareas.cron
```

Este archivo se encuentra en el directorio home/mchable.

Por último se tiene que poner al colector en modo de escucha para que procese todos los datos de tráfico que lleguen a la máquina que contiene el sistema. Para ello tenemos que lanzar a ejecución el colector NetFlow que usamos de la siguiente manera:

```
/usr/bin/flow-capture -w /home/mchable/red/ -E8G -n 1439 0/0/9800
            -R /home/mchable/colector/procesa.pl -S30
```

Donde indicamos que queremos capturar todo el tráfico que llegue a la máquina y almacenarlo en directorio `/mchable/red` con un espacio máximo de 8 Gb, con un corte de información cada minuto y ejecutar después de cada corte el colector llamado *procesa.pl*. El puerto por el cual escucha el colector NetFlow es 9800.

Además tenemos que lanzar el colector sFlow que usamos en conjunto con el colector NetFlow. Esto se hace con el siguiente comando:

```
/usr/local/bin/sflowtool -c localhost -d 9800
```

Esta línea indica que el colector sFlow redirigirá todo el tráfico al puerto 9800 que es donde escucha el colector NetFlow. Recordemos que se necesitan los datos sFlow que son enviados desde los agentes sFlow. Esto tendrá que ser configurado por el administrador de la red.

Después de esto el sistema empezará a actuar de manera automática y el usuario solamente tiene que observar los resultados en las siguientes ligas de su computadora:

- `http://localhost/cgi-bin/mchable/huesped_sitio/nuevo.perl.cgi`
- `http://localhost/cgi-bin/mchable/red_sitio/index.perl.cgi`

Estas dos ligas corresponden al SDIh y al SDIr respectivamente.

A.4. Sistema de monitoreo y detección de intrusos en modo de simulación

El sistema de monitoreo y detección de intrusos en servidores Linux en modo de simulación corre igualmente en cualquier distribución de Linux.

Requiere los mismos paquetes que el sistema de monitoreo y detección de intrusos en modo normal.

A.4.1. Preparación

Los pasos que tenemos que seguir para preparar al sistema en modo de simulación para correr y funcionar de forma correcta son sólo dos:

- Crear la base de datos, la cual tiene la misma estructura que la base de datos del sistema en modo normal.

```
mysql < uso_red.sql
```

- Iniciar el reloj de simulación. El *script* que se encarga de generar el reloj de simulación es el siguiente:

```
perl rtime.pl
```

Este último *script* se encuentra almacenado en la carpeta del sistema en modo de simulación: `cgi-bin/mchable/simulacion`.

Después de haber preparado al sistema, lo único que tiene que hacer el usuario es lanzar los ataques que desea mediante el sitio web destinado para este fin y ver por este mismo sitio los resultados obtenidos. Entonces tiene que visitar la siguiente liga en su computadora:

```
http://localhost/cgi-bin/mchable/simulacion_sitio/nuevo.perl.cgi
```

Bibliografía

[1] *ISO/IEC 7498-2 Information Processing Systems - Open Systems Interconection - Basic Reference Model*, February 1989. Part 2: Security Architecture.

[2] *Flow-tools*. www.splintered.net/sw/flow-tools, 2003.

[3] *sFlow*. www.sflow.org, 2005.

[4] *AFICK*. afick.sourceforge.net, 2006.

[5] *Md5deep*. med5deep.sourceforge.net, abril 2006.

[6] B., Daniel: *OSSEC*. www.ossec.net, 2006.

[7] Brumlen, David, Hao Wang, James Newsome y Dawn Song: *Towards Automatic Generation of Vulnerability-based Signatures*. En *IEEE Symposium*, páginas 1081–6011, 2006.

[8] Buenabad, J. y J.A. Coria: *Tolerancia a fallas para sistemas de detección de intrusos de red*. Tesis de Maestría, CINVESTAV-IPN, Junio 2004.

[9] Buschkes, Ronald y Mark Borning: *Transaction Based Anomaly Detection*. En *Proceedings of Workshop on Intrusion Detection and Network Monitoring*. The Usenix Association, April 1999.

[10] Cannady, J.: *Artificial Neural Networks for Misuse Detection*. En *Proceedings of the 1998 National Information Systems Security Conference (NISSC'98)Arlington, VA.*, páginas 443–456, October 1998. `citeseer.ist.psu.edu/cannady98artificial.html`.

[11] Cheuk, Calvin y Wang Ko: *Execution Monitoring of Security Critical Programs in a Distributed System: A Specification-Based Approach*. Tesis de Doctorado, 1996.

[12] ComputerWire: *DDoS Really, Really Tested UltraDNS*. Informe técnico, www.theregister.co.uk/2002/12/14/ddos_attack_really_really_tested, December 2002.

[13] Corporation, Symantec: *NetProwler Dynamic Intrusion Detection for Enterprise Networks*. www.jainam.com/Downloads/Net %20Prowler %2035.pdf, 2001.

[14] Dain, O. y R. Cunningham: *Fusing Heterogeneous Alert Streams into Scenarios*. Massachusetts Institute of Technology, September 2001. `citeseer.ist.psu.edu/dain01fusing.html`.

[15] Denning, P.: *Computers under attack*. ACM Press edición, 1990.

[16] Deri, L. y S. Suin: *Improving Network Security Using Ntop*. En *Proceedings of the RAID*, citeseer.ist.psu.edu/335415.html, 2002. Centro Serra, Università di Pisa.

[17] Douglis, Red, Thomas Ball, Yih Farn-Chen y Eleftherios Koutsofios: *The AT&T Internet Difference Engine: Tracking and Viewing Changes on the Web*. En *AT&T Labs-Research*, April 1997.

[18] Díaz, A. y E. Morfín: *Análisis del tráfico de una red local*. Tesis de Maestría, CINVESTAV-IPN, 2005.

[19] Eckmann, S.: *Translating Snort Rules to STATL Scenarios*. citeseer.ist.psu.edu/eckmann01translating.html, 2001. Department of Computer Science, University of California.

[20] Everett, David: *Identity Verification and Biometrics*, capítulo 10, páginas 37-73. Butterworth-Heinemann, 1992.

[21] F, Lunt Teresa: *Detecting Intruders in Computer Systems*. En *Proceedings of the Sixth Annual Symposium and Technical Displays on Physical and Electronic Security*, 1990. `citeseer.ist.psu.edu/lunt93detecting.html`.

[22] F, Lunt Teresa: *A Survey of Intrusion Detection Techniques*. Computers an Security, 12:página 4, June 1993.

[23] Feldmeier, David C. y Philip R. Karn: *UNIX Password Security - Ten Years Later*. En *CRYPTO*, páginas 44–63, 1989. `citeseer.ist.psu.edu/188968.html`.

[24] Girardin, L.: *An Eye on Network Intruder-administrator Shootouts*. En *Proceedings of the Workshop on Intrusion Detection and Network Monitoring (ID'99)*, páginas 19–28, Berkeley, CA, USA, 1999. USENIX Association. `citeseer.ist.psu.edu/girardin99eye.html`.

[25] Guo, Fanglu, Jiawu Chen y Tzi cker Chiueh: *Spoof Detection for Preventing DoS Attacks against DNS Servers*. En *26th IEEE International Conference*, páginas 37–37, 2006.

[26] Harrald, John R., Sonia A. Schmitt y Sunil Shrestha: *The Effect of Computer Virus Occurrence and Virus Threat Lever on Antivirus Companies*. En *Engineering Management Conference, IEEE*, páginas 780–784, October 2004.

[27] Heady, Richard, George Luger, Arthur Maccabe y Mark Servilla: *The Architecture of a Network Level Intrusion Detection System*. Informe técnico, University of New Mexico, 1990.

[28] Heberlein, L. Todd y Matt Bishop: *Attack class: Address spoofing*. En *Proceedings of the 19th National Information Systems Security Conference*, páginas 371–377, 1996.

[29] Hosmer, Chet y Mike Duren: *Detecting Subtle System Changes Using Digital Signatures*. En *Information Technology Conference, IEEE*, páginas 125–128. Laboratory at Purdue University, September 1998.

[30] Huerta, Antonio Villalón: *Seguridad en Unix y redes*. andercheran.aiind.upv.es/toni/personal/unixsec.pdf, 2002.

[31] Huerta, Antonio Villalón: *Sistemas distribuidos de detección de intrusos*. andercheran.upv.es/∼ toni/personal/uclm.pdf, Abril 2004.

[32] Jang, H. y S. Kim: *An Intruder Tracing System based on a Shadowing Mechanism*. En *Seventh International Symposium on Computers and Communicatios, IEEE*, páginas 904 – 909. Dept. of Computer Science, Kyungpook National University, July 2002.

[33] Kohl, J., B. Neuman y T. Ts'o: *The Evolution of the Kerberos Authentication Services*. En *IEEE Computer Society Press*, páginas 78–94, 1994.

[34] Kumar, S. y E. H. Spafford: *A Software Architecture to Support Misuse Intrusion Detection*. En *Proceedings of the 18th National Information Security Conference*, páginas 194–204, 1995.

[35] Kumar, Sandeep: *Classification and Detection of Computer Intrusions*. Tesis de Doctorado, Purdue University, citeseer.ist.psu.edu/kumar95classification.html, 1995.

[36] Mbareen Siraj, Rayford B. Vaughn y Susan M. Bridges: *Intrusion Sensor Data Fusion in an Intelligent Intrusion Detection System Architecture*. En *Proceedings of the 37th Annual Hawaii International Conference*, página 10, 2004.

[37] Networks, Enterasys: *Intrusion Detection Methodologies Demystified*. www.enterasys.com/products/ids/whitepapers/, 2005.

[38] Networks, Enterasys: *Dragon Network Intrusion Detection*. www.enterasys.com/products/ids/DSNSS7/DSNSS7.pdf, 2006.

[39] Northcutt, Stephen: *Inside Network Perimeter Security: An Analyst Handbook*, páginas 125–127. Ed. New Riders edición, 2003.

[40] NSWC: *Shadow*. www.nswc.navy.mil/ISSEC/CID/index.html.

[41] Oetiker, Tobias: *RRDTool*. oss.oetiker.ch/rrdtool, 2006.

[42] Olovsson, Tomas: *A Structured Approach to Computer Security.* Informe técnico, Chalmers University of Technology, 1992.

[43] Park, Chad: *The Canaudit Perspective.* www.canaudit.com/Perspectives/Volume4-_Issue10%20SpanishVersion.pdf, 2003. Canaudit, Inc.

[44] R., Omar A. Herrera: *Implementación y configuración de sistemas de detección de intrusos.* Informe técnico, CISSP, Octubre 2003.

[45] Ranum, Marcus J.: *Experiences Benchmarking Intrusion Detection System.* En *Chief Technology Officer and NFR Security, Inc*, www.snort.org/docs/Benchmarking-IDS-NFR.pdf, 2001.

[46] Rebecca Bace / ICSA: *An Introduction to Intrusion Detection and Assessment.* www.icsalabs.com/html/communities/ids/whitepaper/Intrusion1.pdf, 2005.

[47] Roesch, Martin: *Lightweight Intrusion Detection for Networks.* www.snort.org, 2005.

[48] SANS: *Intrusion Detectopm FAQ.* www.sans.org/resources/idfaq/, 2005.

[49] sflow.org: *Traffic Monitoring Using sFlow.* www.sflow.org/sFlowOverview.pdf, 2003.

[50] Simin, Tobias Chyssler: *Alarm Reduction and Correlation in Defence of IP Networks.* citeseer.ist.psu.edu/714181.html.

[51] Smith, Robert N., Yu Chen y Sourav Bhattacharya: *Cascade of Distributed and Cooperating Firewalls in a Secure Data Network.* En *Knowledge and Data Engineering, IEEE*, páginas 1307–1315, October 2003.

[52] Smith, Robert N. y Motorola Inc.: *Operating Firewalls Outside the LAN Perimeter.* En *Computing and Communications Conferencd, IEEE International*, páginas 493 – 498, February 1999.

[53] Systems, Cisco: *Netflow Services and Applications*, Julio 2002.

[54] Tan, K.: *The Application of Neural Networks to UNIX Computer Security.* En *Neural Networks, IEEE International Conference*, páginas 476 – 481, November 1995.

[55] Tan, Kymie: *The Application of Neural Networks to UNIX Computer Security.* En *Proceedings Int. Conf. Neural Networks, ICNN*, 1995. `citeseer.ist.psu.edu/tan95application.html`.

[56] Teal, D. M., S. E. Smaha, T. Grance y D. Mensur: *DIDS - Motivation, Arquitecture, and Early Prototype.* En *14th National Computer Security Conference*, páginas 167–176, October 1991.

[57] Technology, Imon: *sFlow Toolkit.* www.inmon.com/technology/sflowTools.php, 2006.

[58] Tipton, Harold F. y Ed. Auerbach: *Information Security Management Handbook*. 1999.

[59] Wagner, D. y P. Soto: *Mimicry Attacks on Host Based Intrusion Detection Systems*. En *Proc. Ninth ACM Conference on Computer and Communications Security*, 2002. `citeseer.ist.psu.edu/wagner02mimicry.html`.

[60] Wall, L., T. Christiansen y R. Schwartz: *Programming Perl*. O' Reilly and Associates, Segunda edición, 1996.

[61] Wu, Naiqi, Yanming Qian y Guiqing Chen: *A Novel Approach to Trojan Horse Detection by Process Tracing*. En *Proceedings of the 2006 IEEE International Conference*, páginas 721–726, April 2006.

[62] Ylonen, T.: *SSH - Secure Login Connections over the Internet*. En *Proceedings of the 6th Security Symposium) (USENIX Association: Berkeley, CA)*, página 37, citeseer.ist.psu.edu/ylonen96ssh.html, 1996.

[63] Zec, Marko y Miljenko Mikuc: *Real-Time IP Network Simulation at Gigabit Data Rates*. En *7th Intl. Conference on Telecommunications (ConTEL)*, June 2003.