INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

DETECTOR DE INTRUSOS BASADO EN SISTEMA EXPERTO

T E S I S

QUE PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS EN INGENIERÍA DE CÓMPUTO CON OPCIÓN EN SISTEMAS DIGITALES

PRESENTA:

Ing. Vanessa Eleana González Márquez

DIRECTOR DE TESIS

Dr. Felipe Rolando Menchaca García.

MÉXICO D.F., JUNIO 2009

Índice general

Índice de Tablas

Capítulo 1
Introducción

Durante la historia de la humanidad el mantener nuestra información ajena a personas no autorizadas se ha manifestado en diversas formas. Existen evidencias que nos indican cómo el ser humano ante diferentes acontecimientos ha ocultado información que le es importante, y cómo ha tratado de mantenerla a salvo de manos ajenas. Esto puede ser observado en los papiros elaborados durante las guerras, cuyo contenido trata sobre la localización del enemigo, sus armas, el número de soldados, las tácticas de guerra, etc. Esta información por lo general, era codificada para evitar que fuese vista por el enemigo (información que es escrita en un lenguaje incomprensible para los demás, excepto para su autor y quienes están de acuerdo con él y conocen el código de cifrado). Otro ejemplo significativo, es la ubicación de cuevas, templos y palacios en donde se almacenaban tesoros de diversa índole, que eran resguardados en algunos casos por medio de laberintos con trampas y/o centinelas dispuestos a defender esos tesoros. Las indicaciones de acceso a estos puntos se hacían en la mayoría de las ocasiones, por medio de frases verbales, señales u objetos para autenticar la identidad de las personas que pretendían acceder al lugar.

"Muchos de los modelos de seguridad informática están inspirados en estrategias de defensa, ataque y contraataque". Una analogía de ésto puede ser un castillo, en el que se muestra el establecimiento de cercas, zanjas, trampas para que tropiecen los invasores, centinelas para proteger el castillo, entre otros. También, se observa al castillo como una gran fortaleza la cual requiere de códigos de identificación verbales o de señas para acceder a ellas y a su vez éste presenta vulnerabilidades inherentes a su entorno, tales como puertas traseras, debilidad humana para divulgar secretos, fenómenos naturales, deficiencias estructurales en su arquitectura, etc. (Fig. 1.1)

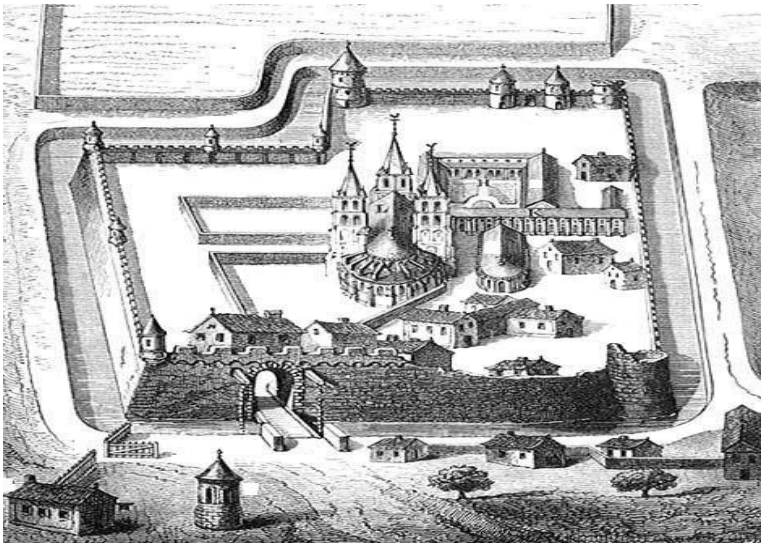


Fig. 1.1: Analogía de la seguridad informática

1.1 Preámbulo

La seguridad informática es un área de investigación que busca proteger la información sensible (información que puede ser elemento de vulnerabilidad) contenida en los sistemas de cómputo y para los equipos en la que ésta reside (computadora personal, servidor, etc.). El objetivo es evitar que los sistemas sean vulnerados por deficiencias en su diseño y/o arquitectura, teniendo en consideración para ello, factores humanos, tecnológicos y físicos. Un sistema es un conjunto de elementos interrelacionados por medio de software y/o hardware creado para ejecutar tareas con un propósito específico.

La privacidad establece el derecho de las personas a proteger ideas o información que son de su propiedad contra irrupciones o modificaciones no autorizadas. La seguridad informática surge de la necesidad de mantener la privacidad de la información; esto es, mantener resguardados los datos de manipulaciones ajenas o bien, compartidos de manera restringida (control de acceso) con un grupo de personas que están autorizadas previamente por el dueño de un sistema. Cuando una persona ajena a dicho grupo accede a la información de manera ilegal, esta acción es considerada una violación de la seguridad.

La mente humana es un factor que influye directamente en la seguridad informática, puesto que ésta, puede crear tanto mecanismos de seguridad (herramientas de protección) para evitar que la información se vea comprometida (vulnerable), como buscar romper las barreras de seguridad. Se dice que un sistema está comprometido, cuando presenta deficiencias físicas o de diseño en su estructura, de manera que queda expuesto a un riesgo inherente y puede ser dañado por una persona ajena al sistema. Existen factores externos a estos mecanismos de protección que los hacen ser vulnerables; los cuales son parte inherente a sus características técnicas y pueden ser utilizados para penetrar de manera ilegal a un sistema. En razón de ésto, después de colocar un gran número de candados para proteger la información, no se puede aseverar que será imposible irrumpir en la privacidad; debido a que siempre está latente una debilidad aún no conocida que podría ser utilizada para tener acceso a un sistema.

En base a lo anterior, se puede decir que la seguridad informática se emplea con la finalidad de proteger a un sistema mediante los mecanismos de seguridad, los cuales buscan brindar la seguridad que requiere un sistema de acuerdo a su naturaleza, y que la seguridad absoluta es difícil de alcanzar, puesto que siempre existe de forma latente un factor de riesgo previamente conocido o no, que se encuentra en espera de ser explotado.

1.1.1 Servicios de Seguridad

Un servicio de seguridad es una función diseñada para satisfacer las necesidades de protección de un sistema ante un acceso ilegal [118]. Para cumplir su cometido requieren de los mecanismos de seguridad. Un mecanismo de seguridad es una herramienta desarrollada en software o hardware, que busca brindar un servicio de

seguridad. Cada mecanismo de seguridad tiene una función específica, la cual está determinada en base al servicio de seguridad que se desee otorgar [118], por ejemplo: los servidores proxy y los cortafuegos, tienen el objetivo de restringir los accesos hacia el sistema o desde éste hacia el exterior.

A continuación se listan los servicios de seguridad que se buscan brindar a un sistema:

- La confidencialidad.- Consiste en evitar que una persona no autorizada tenga acceso a la información contenida en un sistema.

- La autenticación.- Confirma la identidad de una persona que tiene autorización para acceder a la información contenida en el sistema.

- La integridad.- Tiene como finalidad garantizar que la información no ha sido alterada en ningún aspecto.

- El control de acceso.- Permite al propietario y/o autor de la información otorgar permisos de acceso a otras personas a un sistema de su propiedad.

- *El no repudio.-* Su propósito es identificar al responsable de algún ataque o una intrusión hacia un sistema, sin que éste pueda negar los hechos. También es el mecanismo para asegurar que la persona que se hace responsable de una determinada acción no pueda negar que ejecutó tal acción, como la firma digital de un mensaje o documento electrónico, el acceso y uso de un recurso, etcétera.

- La disponibilidad.- Su fin es garantizar que los servicios de un sistema estarán a disposición del usuario, al momento en que se les solicite.

1.1.2 Mecanismos de Seguridad

Los mecanismos de seguridad fueron creados como herramientas de hardware o software para brindar protección a un sistema, empleando cada uno de ellos para una tarea específica. En si, son un medio de defensa que se emplea para evitar que la información contenida en un sistema, no se vea comprometida. Su implementación es de acuerdo con el nivel de protección que se desee proporcionar al sistema. Por lo general, requieren operar de manera conjunta con el objetivo de complementarse unas a otras para subsanar sus debilidades y alcances.

En la actualidad, con el empleo de los mecanismos de seguridad se ha logrado repeler modalidades de penetración ya conocidos, y en algunos casos lograr un tiempo de retraso para los no conocidos. Dentro de las herramientas que utiliza la seguridad informática, podemos citar:

- *Cortafuegos:*
  Delimitan el perímetro de la red, son filtros de entrada y salida de información

desde una red externa hacia la red interna, y viceversa. Funcionan a través del análisis y control de los puertos y protocolos de comunicación.

- *Proxy:*
  Es un intermediario entre la red interna (intranet) y la red externa (internet). Controla la comunicación desde el exterior hacia el interior y viceversa (este flujo dependerá de su configuración).

- *Antivirus:*
  Es un programa cuya función es detectar códigos maliciosos dentro de los sistemas, por ejemplo: troyanos, gusanos, puertas traseras, entre otros.

- *Encriptación (Cifrado):*
  Es el proceso de hacer ilegible información legible, para protegerla de alteraciones y mantener su confidencialidad.

- *Sistemas de monitoreo*
  Es un programa que observa y registra la actividad de un sistema.

- *"Honey Pots" o Tarros de Miel*
  Es un elemento de información señuelo diseñado para distraer a los atacantes, suele utilizarse para el estudio de nuevas técnicas de intrusión.

Los mecanismos de seguridad, tienen como finalidad evitar que un sistema se encuentre en riesgo. Asimismo, de garantizar los servicios de seguridad, tales como: *la confidencialidad* (que la información sea ilegible a personas ajenas al autor), *la integridad* (que la información almacenada no sufra ningún tipo de alteración), *el no repudio* (tener certeza de quién es responsable de una determinada acción), *la disponibilidad* (la información esté al servicio de quien la requiera en el momento indicado), control de acceso(se refiere a que sólo el propietario o personas con ciertos privilegios accedan a la información) y autenticación (confirmar que la persona que se dice ser, si lo es). Los servicios de seguridad los podemos ver como la forma de acceder hacia la información contenida dentro de un sistema, a través de diferentes niveles de restricción, en los que el propietario de dicho sistema basará su seguridad, sin olvidar que la seguridad absoluta es difícil de obtener y por ello, deberá continuamente monitorear estos servicios para que se verifique el cumplimiento del objetivo para los cuales fueron diseñados.

Como se ha mencionado, los mecanismos de seguridad restringen y controlan el acceso a un sistema. Pese a su funcionalidad, éstos pueden ser burlados por los atacantes, utilizando técnicas de evasión que les permiten continuar con su irrupción; debido a esta situación, estos mecanismos están en continua evolución y mejorando sus estrategias de seguridad.

El cortafuegos es el primer mecanismo de seguridad que se implementa para proteger un sistema de red. La información que fluye en el sistema se divide en pequeñas partes

denominadas paquetes, las cuales están constituidas por encabezados que identifican el tipo y número de paquete, y por su información fragmentada (payload). Su función es restringir el acceso sobre la información que es enviada hacia la red interna desde una red externa y viceversa, a través del análisis de sus encabezados y protocolos de comunicación; es decir, dejan pasar los paquetes que están autorizados previamente por las reglas programadas por el propietario del sistema (políticas del sistema); las cuales comprenden protocolos de comunicación, números de puertos, direcciones IP, etc. Son considerados como el primer filtro de seguridad y los limitadores perimetrales de la red interna y la red externa. Sin embargo, éstos no realizan un análisis a profundidad de cada paquete que reciben, por lo que un atacante puede burlar estas restricciones de seguridad si éste crea un paquete de red con contenido malicioso o modificado en sus estándares, y posteriormente lo envía a través de un puerto y protocolo de comunicación permitido, sin despertar sospecha alguna acerca de su intención de vulnerar un sistema. Ante esta situación, ¿cómo saber si la seguridad de un sistema ha sido comprometida?. Puesto que, en apariencia no se ha violado la seguridad del sistema al ser reconocido por este mecanismo como un paquete válido para ser admitido en la red interna.

1.1.3 Justificación

Los sistemas de detección de intrusos son un complemento a otros mecanismos de defensa que pueden ser evadidos por un atacante, como es el caso de los cortafuegos y/o del antivirus. Su motor de análisis le permite realizar una revisión detallada y profunda sobre la información que fluye por la red, con el propósito de clasificar los datos que ha capturado y determinar cuáles pueden o no comprometer la seguridad del sistema. Sin embargo, los sistemas de detección de intrusos (también conocidos como IDS, por sus siglas en inglés) basados en red son vulnerables a la información cifrada y a ataques que son desconocidos por su motor de análisis [54]. Debido a esta situación, se propone como una alternativa dentro del campo de investigación de la seguridad informática, un modelo de un detector de intrusos basado en sistema experto. Con este modelo se pretende disminuir el número de ataques (técnicas empleadas para vulnerar una computadora o una red) de origen externo e interno que evaden los mecanismos de seguridad y atentan contra los servicios que se brindan, a través del control y el aislamiento del equipo comprometido.

1.1.4 Hipótesis

Un atacante interno y/o externo puede comprometer la seguridad de una red, cuando éste evade a los sistemas de detección de intrusos de tipo red, mediante el envío de paquetes cifrados o paquetes malformados que sean desconocidos por su motor de análisis, siendo éstos últimos considerados por el sistema de detección de intrusos como información confiable.

1.1.5 Objetivo general

Implementar un modelo que proporcione una alternativa de defensa en situaciones en las que los sistemas de detección de intrusos de tipo red presentan vulnerabilidades, las cuales puedan comprometer la seguridad de una red. Esto por medio de un software complementario para detección y control, el cual se base en un sistema experto que considere las experiencias del administrador y del sistema central de detección de intrusos; teniendo como finalidad proporcionar un sistema integral más eficiente.

1.1.6 Objetivos particulares

- Analizar las fortalezas y debilidades de los detectores de intrusos basados en red.
- Proponer un modelo de aislamiento que detecte a un host que ha sido comprometido por un atacante y evitar su propagación en la red.
- Mostrar los alcances, fortalezas y debilidades del modelo propuesto basado en sistema experto.

1.1.7 Organización del documento

La estructura del presente trabajo es la siguiente:

Capítulo 2 *Estado del Arte*
En este capítulo se presenta una descripción de los sistemas de detección de intrusos, su taxonomía y funcionamiento para cada caso particular. Asimismo, una reseña de las técnicas empleadas por estos sistemas y de los modelos que subyacen en esos sistemas de detección.

Capítulo 3 *Descripción de la problemática*
En este capítulo se habla de las metodologías empleadas por los intrusos para penetrar a los sistemas y la forma en que éstos evaden los sistemas de seguridad. Asimismo, se exponen las tecnologías y técnicas que se han empleado para detectar intrusos en una red o equipo específico.

Capítulo 4 *Desarrollo de la propuesta*
Presenta una explicación de un modelo de aislamiento hacia un host que es detectado como comprometido en una red interna, a través de la aplicación de las reglas empleadas por un sistema experto para la detección de intrusos.

Capítulo 5 *Pruebas y Resultados*
Se exponen los resultados obtenidos del modelo de aislamiento propuesto en un entorno de red.

Capítulo 6 *Conclusiones y Trabajos Futuros*

En forma breve se presentan los aportes que proporciona el modelo propuesto a la seguridad informática. También se presentan en ese capítulo los posibles futuros trabajos que pueden efectuarse con fundamento en los resultados obtenidos.

ANEXOS

Anexo A *Glosario*

Definición de términos técnicos empleados durante el desarrollo del presente documento.

Anexo B *Implementación de un IDS*

Descripción de las técnicas empleadas para la implementación de los IDS en una red.

Referencias bibliográficas

Se encuentran dividas en dos partes: La primera, contiene los artículos que sustentan la justificación del presente trabajo y en la segunda, se presentan enlaces hacia sitios web en los que se puede ahondar sobre los temas abordados durante los diferentes capítulos, así como material de apoyo para definir el argot que es empleado en el área de la seguridad informática.

Tipografía

En el presente documento se utilizarán párrafos con letras en tipo cursiva para indicar una interpretación personal sobre un tópico en particular, que se considere relevante o de un interés particular para la presente propuesta de tesis.

Capítulo 2
Estado del Arte

Resumen

La seguridad informática ha creado e implementado diferentes mecanismos de seguridad para garantizar que los sistemas de información estén disponibles cuando se les requiere, que sean confiables y mantengan su integridad en todo momento. Una herramienta muy empleada para dicho objetivo, es el sistema de detección de intrusos (también llamado IDS). Los sistemas de detección de intrusos realizan el análisis de la información que es registrada en las bitácoras del sistema operativo o en el contenido de los paquetes de red, buscando detectar anomalías o abusos realizados por una persona que desee penetrar a un sistema sin previa autorización de su propietario. Se clasifican de acuerdo a su técnica, su metodología y/o su tipo de implementación. En este capítulo se presentará la definición de un sistema de detección de intrusos, sus configuraciones y taxonomía. Se encontrará también, una descripción sucinta de trabajos relevantes realizados en el ámbito académico y de productos representativos ofertados en el área comercial.

Objetivos del Capítulo

- Analizar las tecnologías que sustentan los sistemas de detección de intrusos.
- Presentar una taxonomía de los IDS y de sus características de configuración.
- Analizar los trabajos de investigación y los productos de uso comercial relacionados con los sistemas de detección de intrusos.
- Describir los métodos de detección empleados en los sistemas de detección de intrusos.
- Puntualizar las diferencias entre los sistemas de detección de intrusos (IDS), los Sistemas de Prevención de intrusos (IPS) y los Tarros de Miel (honeypots).

2.1 Introducción al capítulo

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

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

2.2 Los sistemas de detección de intrusos

Los sistemas de detección de intrusos representan un mecanismo de defensa ante los intentos de intrusión desde el interior o exterior de una red. Éstos pueden ser clasificados de acuerdo a su modo de operación, su técnica de análisis, su tecnología de implementación, su velocidad de respuesta, entre otras características.

2.2.1 Definición

Los sistemas de detección de intrusos forman una parte importante dentro de las herramientas que son empleadas por la seguridad informática, para evitar que la información se vea comprometida [111,106]. Su función comprende básicamente la detección oportuna de las acciones ilegales o anómalas que indiquen la intrusión a una aplicación (software) o equipo de cómputo (hardware), de manera aislada o en red. Pudiendo efectuarse dicha intrusión desde el exterior y/o el interior de una red o segmento que derive de ella. La detección generalmente se basa en criterios preestablecidos que tratan de determinar que es lo normal y lo anormal dentro del tráfico que fluye en la red, el comportamiento humano y la malformación de paquetes que se basan en la explotación de vulnerabilidades de los diversos protocolos de comunicación. A partir del presente capítulo se empleará de manera indistinta “sistema de detección de intrusos” o sus siglas en inglés “IDS”, para facilitar la lectura del presente documento.

2.2.2 ¿Cómo operan los sistemas de detección de intrusos?

Los IDS se encuentran integrados por diversos módulos que trabajan en forma conjunta y con funciones específicas para la recolección y análisis de datos de las actividades humanas o procesos que efectúa un sistema, así como la generación de alertas, y en algunos casos acciones de respuestas del tipo pasivo, activo o proactivo. El registro de los resultados y los datos que se obtienen se almacenan en bitácoras. Su motor de detección emplea distintos métodos de análisis, que pueden ser por ejemplo: estadísticos, de inteligencia artificial, sistema inmune, entre otros. Los cuales pueden operar de manera aislada o complementándose unos a otros, empleándose como criterios de discriminación de lo normal y lo anormal [98].

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

2.2.3 Justificación de los sistemas de detección

Los sistemas de detección de intrusos son el complemento a otros elementos de defensa que pueden ser burlados por un atacante, como es el caso de los cortafuegos; esto se debe, a que los cortafuegos filtran el tráfico de la red con base en el análisis de sus encabezados y protocolos, y no analizan el detalle de cada paquete [106, 118]. De esta forma, los IDS reciben los paquetes filtrados y reconocidos que provienen del cortafuegos, para posteriormente analizarlos de acuerdo a criterios de firmas o anomalías, que son aplicados a su estructuración o a su reensamblado. De esta manera se determina qué paquete es o no malicioso, y se dictamina si puede o no comprometer la seguridad de la información de un sólo equipo o de manera conjunta en todos los equipos que integran la red.

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

el software, sino más bien, por el retardo que introduciría en la entrega de los paquetes; por lo que se podría decir que realiza una revisión rápida y toma las acciones que se le hayan indicado previamente, dejando la parte más profunda de la inspección a los IDS.

2.3 La búsqueda de un modelo

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

Terminología

La explicación de las siguientes terminologías ha sido basada en [95, 118]; por considerar a juicio personal que proporcionan la claridad y la fácil comprensión del argot de la seguridad informática.

- *Sistema.-* Es un conjunto de equipos de cómputo o programas que brindan servicios que son utilizados por los usuarios.
- *Usuario.-* Persona que tiene acceso a un sistema por medio de permisos otorgados por el propietario del sistema.
- *Propietario.-* Persona que es dueña de la información contenida en el sistema.
- *Normal.-* Parámetro que sólo se puede establecer a criterio personal por el propietario de un sistema.
- *Anormal.-* Todo lo que se encuentra fuera del parámetro denotado como normal.
- *Anómalo (Anomalía).-* Es la alteración o desviación que se presenta en un parámetro normal.
- *Intruso (Interno o Externo).-* Puede ser una persona que accede local o remotamente a un sistema de forma ilegal, así como un programa que no se ha registrado previamente como parte de un sistema, y sus intenciones son de irrumpir en la privacidad del propio sistema.
- *Atacante.-* Se considera un atacante a una persona o programa que trata de acceder de manera ilegal a un sistema y busca comprometer (poner en riesgo) su seguridad.

2.3.1 Prototipos

Las siguientes propuestas van encaminadas al desarrollo de diversos modelos; las

cuales han tratado de cubrir las siguientes expectativas [98, 106, 108, 111]:

- La creación y/o utilización de un lenguaje único, flexible, portable y fácil de interpretar para la comunicación entre sus módulos.
- La elaboración de reportes de actividades en diferentes formatos (flexibilidad)
- Su simplicidad de uso.
- La descripción de los componentes y/o módulos que dictaminen una arquitectura a seguir.
- Monitoreo continuo y activación de las alarmas correspondientes presentadas ante indicios de intrusión hacia un sistema.

Las técnicas de detección no pueden ser generalizadas dentro de un modelo, esto se debe a que en la búsqueda de un mejor análisis y clasificación de la información que recibe, se ha propiciado un ambiente idóneo para la exploración de nuevas técnicas y el empleo de diferentes ramas científicas que pueden ser aplicadas a los sistemas de detección [98, 106, 108].

2.3.1.1 Modelo propuesto por Dorothy Denning

El modelo descrito por Dorothy Denning [95] explica mediante similitudes informáticas que es lo que representaría cada componente en la detección de una intrusión. Está enfocado directamente sobre el análisis de un sólo equipo y no de una red. El modelo está constituido por:

- *Sujetos*
  Generalmente se asocia a los usuarios de un proceso, sistema o equipo de cómputo.

- *Objetos*
  Son los dispositivos periféricos, procesos del sistema, dispositivos de almacenamiento, archivos, aplicaciones de cómputo, entre otros.

- *Registro de auditoria*
  Es el registro de los sucesos que se obtienen de la interacción de el sujeto sobre los objetos.

- *Perfiles*
  Son los patrones de comportamiento que se establecen previamente en conjunto sobre la manipulación que realiza un sujeto sobre los objetos, siendo éstos la base que sustente los criterios de comportamiento normal o anormal dentro de un sistema.

- *Registros de anomalías*
  Son las notificaciones que se tienen de las condiciones y uso sobre los objetos, así como la hora en que fueron realizadas dichas acciones con base a comportamientos anómalos o extraños.

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

Análisis

En esta propuesta se presenta como sistema al conjunto integrado por sujetos y objetos, donde su interacción es registrada y observada (almacenamiento de perfiles) en espera de sucesos (anomalías), que al ser comparados con las reglas establecidas y validándose éstas, se traducirán como intrusión; efectuándose con ello las alertas pertinentes a través de reportes. Este modelo recibió el nombre de IDES [95] que implementó un sistema experto (SE) como técnica de detección de intrusiones.

2.3.1.2 CIDF

Otra propuesta para tratar de hallar un modelo de sistemas de detección de intrusos fue hecha por el CIDF (Common Intrusion Detection Framework) [98, 111, 106, 126]. Ésta sugiere la utilización de GIDO (Generalizad Intrusión Detection Object) como componente de intercambio de datos entre los diferentes módulos y la utilización de CISL como lenguaje para crear las reglas de detección, el cual tiene cierta similitud al lenguaje LISP. La arquitectura está integrada por 4 módulos:

- *(E) Generadores de Eventos*
  Integrados por receptores que están a la escucha de los eventos que ocurren dentro de una red o en un host específico.

- *(A) Analizadores de Eventos*
  Son los encargados de recibir la información que es enviada por los generadores de eventos y procesarla mediante diversas técnicas; detectando si se presenta o no una intrusión de acuerdo a los criterios previos de abuso o comportamiento anómalo establecidos.

- *(D) Base de Datos*
  Está compuesta por los patrones almacenados previamente que dan la indicación de una posible intrusión.

- *(R) Unidades de Respuesta*
  Son las acciones a tomar en el momento que se detecta una intrusión.

Análisis

El modelo que ha desarrollado el CIDF describe una arquitectura a seguir para definir los componentes que constituirán a un sistema de detección de intrusos y la interoperabilidad entre diferentes fabricantes de IDS. Sin embargo, éste no ha sido considerado como un estándar, debido a la complejidad que presenta su lenguaje en la sintáxis misma y en el uso de GIDO para intercambiar información entre diferentes fabricantes de IDS. GIDO (Generalized Intrusion Detection Objects) fue creado para establecer intercambio de información entre los diferentes módulos que constituyen un IDS, así como permitir la interoperabilidad entre otros IDS.

2.3.1.3 IDWG del IETF

Otro modelo ha sido desarrollado por el IDWG (Intrusión Detection Working Group), que a diferencia del anterior, no propone una arquitectura específica, sino adaptarse a cualquiera existente [111, 106, 126]. La propuesta sugiere los siguientes puntos:

- Utilización del lenguaje XML
- Servicio de Mensajería IDMEF (Intrusión Detection Message Exchange Format) para la comunicación entre los módulos que integran la arquitectura
- Los protocolos IAP (Intrusión Alert Protocol) e IDXP (Intrusión Detection Exchange Protocol).

Se encuentran en la actualidad cuatro borradores en proceso de evaluación, que explican con detenimiento cada una de las etapas que constituyen la propuesta:

- **IDWG *RFC 4766***
  (Requerimiento)

  Se plantean los requerimientos que se desean cubrir por medio de la propuesta de un protocolo y lenguaje flexible que permita la comunicación entre diferentes plataformas. Entre ellos se encuentran: Un lenguaje que pueda reconocer la semántica de diferentes plataformas, el protocolo que se emplee para la comunicación debe tener autenticación mutua (emisor y receptor) a través del soporte de diferentes algoritmos de encriptación, protegiendo los datos de ataques del tipo DoS (Denegación de servicio, según siglas en inglés); así como el paso de éste a través de los cortafuegos de manera transparente sin comprometer la seguridad del sistema de detección. Otros requerimientos adicionales deseables son: Respuestas automáticas, en las que las alertas emplearán un formato de prioridad que los diferencie de los mensajes de intercambio de información entre módulos que surgen durante su operación habitual. Asimismo, la creación de una lista de eventos, en los que se describa

el evento que se esté presentado y entre ellos se considere, un evento genérico para los eventos que no se hayan catalogado por ser desconocidos por el momento, advirtiendo con ello el impacto que representa dicho evento.

- **IDMEF-XML** ***RFC 4765***
  (Intrusion Detection Message Exchange Format - XML)

  Describe el intercambio de información por medio de la utilización del lenguaje XML para la interoperabilidad entre los sistemas de detección del tipo comercial, de código abierto y nuevos prototipos desarrollados en el área de investigación. Esto representa una gran ventaja, puesto que se puede obtener lo mejor de sus fortalezas y complementar de esta manera las debilidades que se presenten en su diseño, siendo ésto posible a través de la administración de los sistemas que puedan requerir interactuar con otros sistemas y compartir información de interés común.

- ***BEEP TUNNEL RFC 3620***
  (Block Extensible Exchange Protocol)

  Se plantea la comunicación entre equipos de cómputo que pertenecen a dos redes distintas, la cual para ser efectuada requiere del establecimiento de un túnel de comunicación entre ambos puntos a través de un proxy (equipo que es empleado como intermediario para la comunicación entre una intranet y una extranet); el proceso se describe a continuación: El proxy recibe la solicitud de un equipo de la red que protege (equipo en Red A), éste se comunica con otro servidor proxy que contiene en sus dominios de protección al otro equipo con el que se desea comunicar (equipo en Red B), en ese momento se establece un túnel de comunicación el cual se envía hacia el equipo solicitante (equipo en Red A) en espera de su aceptación, y de esta forma comenzar la negociación de la conexión con el equipo que se desea comunicar (equipo en Red B), quién deberá a su vez aceptar de igual forma el túnel para comenzar la comunicación entre esos dos puntos.

- ***BEEP IDXP RFC 4767***
  The Intrusion Detection Exchange Protocol (IDXP)

  En este borrador se describe las normas a emplear para establecer la comunicación entre las entidades del sistema de detección de intrusos. El protocolo expresa que para establecer la comunicación se requiere crear una sesión de tunel BEEP para encriptar los datos que se transferirán de manera segura de un punto a otro, de tal forma que sea trasparente su paso a través de los proxies sin comprometer la información sensible que se envía; cabe destacar que esta sesión sólo opera entre pares. El IDXP emplea perfiles de alertas entre las entidades, abriendo un canal de sesión diferente de acuerdo

con la prioridad y categoría a la que pertenece la alerta que se desee transmitir (red, host, aplicación,...).

Análisis

Este modelo a diferencia de los descritos anteriormente, se puede considerar atractivo por el hecho de tratar de dar respuesta a la interoperabilidad entre los diferentes fabricantes de sistemas de detección de intrusos. Sugiriendo para dicha solución la intercomunicación entre componentes, y la generación de reportes adaptables a las necesidades de información que se requieren conocer del sistema que se protege. Esto es posible a través de la utilización de las características que ofrece el lenguaje XML, el cual fue diseñado y propuesto como un estándar para el intercambio de información entre diferentes plataformas, obteniendo con esto la compatibilidad entre sistemas.

2.4 Módulos básicos de un IDS

De la propuesta de estos modelos se puede obtener un esquema genérico que permita describir de manera general las funciones que debe cumplir un sistema de detección de intrusos. De tal forma, que las partes básicas que integren la arquitectura de un IDS, sean las siguientes (Fig. 2.1):

- *Sensores*
  Serán los recolectores o receptores de la información que fluye a través de una red o en un host específico.

- *Analizadores*
  Son el corazón de un IDS. Descomponen en pequeños fragmentos la información que reciben de los sensores, en búsqueda de comportamientos anómalos o de abusos, que pueden realizarse sobre un sistema. Forma parte del motor de inferencia.

- *Motor de Inferencia*
  Está constituido por los analizadores y las reglas que contienen las especificaciones de comportamientos de intrusión, lo que le permite aplicar criterios para catalogar la información que recibe, en términos de normal o anormal durante la fases de análisis.

- *Acciones de Respuesta*
  Pueden ser: Pasivas, Activas o Proactivas. Las respuestas pasivas, son aquellas que notifican el suceso de intrusión al administrador y esperan la respuesta por parte de él. Es decir, requiere intervención humana. A diferencia de ésta, las respuestas activas, toman las decisiones que se les haya indicado previamente, como pueden ser finalizar conexiones, reconfiguración de cortafuegos, bloqueo de direcciones IP, entre otras. Las respuestas proactivas emplean el concepto del cómputo proactivo, es decir, la

anticipación de una acción basada en lo que percibe del medio físico que se le va presentando.

- *Registros*
  Se consideran las bitácoras y reportes del sistema sobre las anomalías halladas en el interior de un sistema o hacia él.

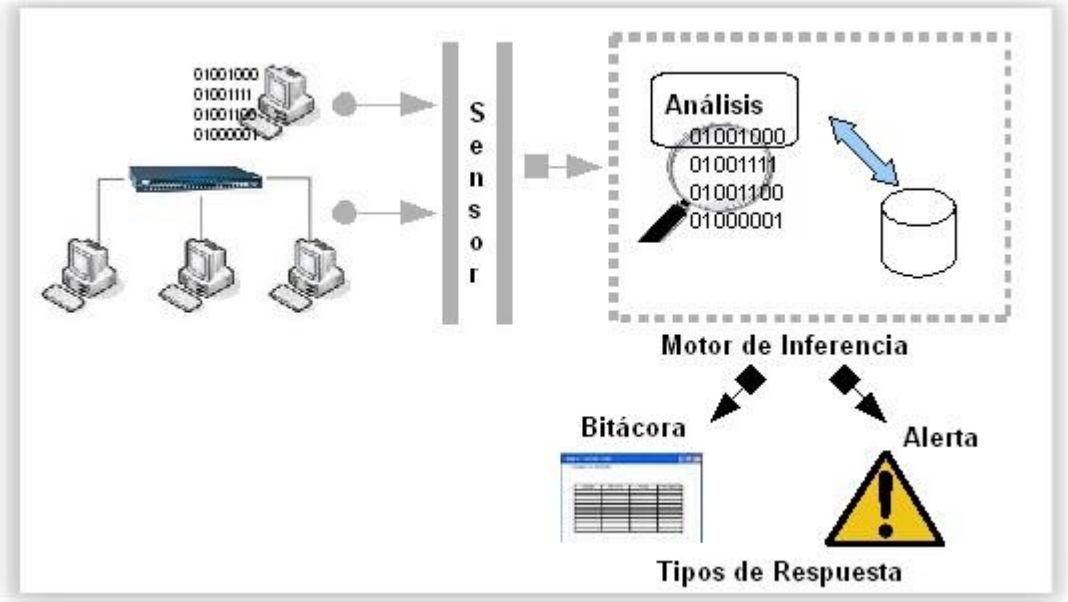


Fig. 2.1: Componentes de un sistema detección de intrusos.

2.5 Requisitos de un modelo de un IDS

Las propuestas a desarrollar para los sistemas de detección de intrusos deben de tratar de ser una solución lo más integral posible. No sólo en lo referente a una arquitectura estándar y la descripción del funcionamiento de los componentes que la integran, sino el considerar otros aspectos al momento de efectuar su diseño [2], tales como:

- Escalable.
- Fácil de configurar.
- Interoperabilidad e interconexión entre fabricantes (protocolos, lenguajes).
- Análisis de los datos capturados en tiempo real.
- Espacio de almacenamiento para bitácoras y la optimización de las mismas.

- Revisión de bitácoras en tiempo real, que conduzcan a un patrón previo de intrusión.
- Confiable, es decir, minimizar el número de falsas alarmas y el paso de información con apariencia normal cuando ésta en realidad no lo es (ataque desconocido).
- Actualización manual o automática (preferentemente) de las bases de datos de firmas o de los umbrales a emplear en los criterios de evaluación.
- Redundancia y Tolerancia a Fallos.
- Velocidad de respuesta a intrusiones en tiempo real (Alertas y toma de acciones de reconfiguración de componentes)
- Reportes flexibles (configurables) sobre ataques y estadísticas de intrusión.
- Respuesta a intrusiones a través de autoaprendizaje y en algunos casos de autoconfiguración para repeler nuevos intentos de intrusión.
- Comparación con otros modelos que empleen la misma técnica u otra metodología para resolver los indicios de intrusión.

2.6 Taxonomía

Durante el desarrollo de los sistemas de detección de intrusos han surgido diferentes clasificaciones de acuerdo a la técnica de detección que es utilizada por éstos, el tipo de respuesta que ofrecen ante una intrusión, los sistemas operativos sobre los que pueden funcionar, aplicaciones específicas, entre otros. Sin embargo, esas clasificaciones pueden considerarse como subdivisiones de dos grandes grupos, puesto que, en el caso de las técnicas de detección, éstas son derivadas de los métodos de detección que se empleen para buscar una intrusión dentro de un sistema, los cuales cabe aclarar, no están relacionados de manera exclusiva con un tipo de operación: red o host [98, 105, 108].

En el presente trabajo se exponen esos dos grandes grupos y posteriormente la explicación de las técnicas de detección que pueden ser empleadas dentro de estos sistemas.

Los IDS pueden clasificarse por:

- Punto de Detección
  (Ubicación física dentro del Sistema que se desea proteger)
  - Host
  - Red

- Método de Detección
  (Método que se desea emplear para realizar la búsqueda de intrusos dentro de un sistema)

  - Anomalías
  - Abusos

2.6.1 IDS por Punto de Detección

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

IDS tipo Host

Son sistemas de detección que se colocan en un equipo de cómputo (o en varios equipos) debido a la información sensible que contienen, y la cual se considera requieren de mayor atención de monitoreo. Su función consiste en registrar dentro de diferentes bitácoras, los movimientos realizados por transacciones internas, accesos remotos y/o locales, modificación de permisos a usuarios, grupos, archivos o carpetas, cambios de contraseñas, procesos, entre otros. El análisis de detección de este tipo de sistemas se concentra en las bitácoras que contienen la información capturada, para ser posteriormente revisada en forma minuciosa, en busca de anomalías o abusos perpetrados dentro de un sistema y/o equipo [54, 106, 108]. Son muy útiles para investigación forense y/o el aprendizaje de nuevas técnicas de evasión. La mayor ventaja que presentan es que permiten tener un registro detallado de todas las actividades que suceden dentro de él, es decir, mayor *granularidad* (nivel de detalle o profundidad) de análisis. Sin embargo, presentan una gran debilidad; son susceptibles a los ataques de Denegación de Servicio (DoS) (técnica que deja fuera de servicio a un sistema). Este tipo de ataque es realizado con el objetivo de que se pierdan las bitácoras que evidencien la infiltración de un intruso dentro del sistema, así como los datos que indiquen desde dónde se realizó la conexión hacia el equipo víctima. Ante los ataques de DoS no se tiene defensa, debido a que el ataque no puede ser detectado con anticipación, puesto que este tipo de sistemas no analiza los paquetes que se envían en la red; sino éste es víctima cuando los desempaqueta y al convertir en tramas para ver su contenido, el ataque ya no puede detenerse en su ejecución.

Ventajas:

- Granularidad de análisis.
- Reconocimiento de nuevas técnicas de intrusión.
- Búsqueda minuciosa de anomalías y abusos.
- Verificación de la integridad de la información, junto con el control de acceso a

archivos y a los sistemas.

- Detección a ataques perpetrados a través de paquetes cifrados.

Desventajas:

- Susceptible a ataques de DoS (Denegación de Servicio, DoS; por sus siglas en inglés).
- Consume un mayor número de recursos del sistema (memoria, procesador,...), afectando en la mayoría de los casos su desempeño.
- Requieren gran espacio de almacenamiento para guardar las bitácoras de un tiempo determinado por el propietario.
- Las bitácoras deben de ser protegidos con algoritmos de encriptación para garantizar la integridad de la información contenida en ellos.
- Generalmente no tienen un esquema de tolerancia a fallos, por lo que si hay un ataque de DoS la información recolectada se perderá.

Representantes: IDES, BRO.

IDS tipo Red

A diferencia de los de tipo HOST, su análisis no está basado en las bitácoras que genera, sino en los protocolos de comunicación. Este sistema de detección captura los paquetes que circulan en la red y busca en el interior de cada uno de ellos, las anomalías que no correspondan con la funcionalidad y aplicabilidad que indican los RFC [54, 111, 108]. Al igual que el IDS de Host utiliza bitácoras (generalmente trabajan en tiempo real) para registrar por protocolos los intentos de intrusión que se encontraron en el interior de los paquetes.

La implementación de un IDS de tipo red dependerá de la arquitectura y topología de la red [98,112], junto con los dispositivos de red que la constituyan; como pueden ser: un hub o un switch. El primero se basa en el concepto de dominio de colisión, en donde se refleja en cualquier punto de la red lo que transita sobre ella, es decir, se puede colocar un IDS en cualquier punto de la red y recibir la información para su detección. En el segundo caso el tráfico se ve aislado, y es necesario configurar un puerto dentro del switch conocido como SPAN Port (Switch Port Analizer) y conectar sobre él el IDS o un TAP (Test Access Point), para la captura del tráfico que circula en un segmento de red y a ese mismo dispositivo conectar el IDS [100]. Su principal ventaja, es el análisis masivo en un rango de red o en todos los equipos de una red, además de ser casi imperceptibles a los atacantes (esto depende del tipo de implementación). Presenta dos desventajas significativas: la primera, es ante técnicas de evasión a través de paquetes cifrados, puesto que este tipo de IDS no puede analizar paquetes de ese tipo. La

segunda, en su rendimiento y recolección de información, la cual se ve evidenciada por el gran volumen de carga de la red y presentar la pérdida de paquetes que no son capturados por el sensor**,** implicando con ello, el riesgo de que uno de esos paquetes sea intrusivo y no detectado por la pérdida en la captura. Para mayor referencia sobre su implementación, ventajas y desventajas se puede revisar el *Anexo A*.

Ventajas:

- Búsqueda y análisis dentro de un rango de red (subred) o en todos los equipos de una red.
- Revisión masiva del tráfico que circula en la red, a través del desensamblado de paquetes.
- Realiza su análisis a gran velocidad y con eficacia, tratando de no impactar la velocidad de la red (retardo en la entrega de paquetes hacia su destino).
- Son casi imperceptibles dentro de la red misma (depende de la configuración del equipo y su forma de implementación).
- No son susceptibles a ataques tipo DoS (denegación de servicio).

Desventajas:

- Es vulnerable a exploits (vulnerabilidades en los protocolos de comunicación o en el diseño de un sistema, que pueden ser empleados para acceder de forma ilícita a un sistema) debido a que no puede revisar paquetes cifrados.
- Implementación:

  - Hub:
    Manejo de gran cantidad de información que es generada por el tráfico de red, haciendo más exhaustiva la búsqueda y clasificación de paquetes.

  - Switch:
    Requieren conocimientos extras para su configuración y en algunos casos de aditamentos adicionales para efectuar la captura sin alterar el flujo y el comportamiento de la red.

- La velocidad y capacidad de análisis dependen de la cantidad de paquetes que puedan ser soportados por el sensor.

- Desconoce, al momento de la detección si el ataque se está intentando o éste ya está siendo realizado.

- No detecta conexiones vía MODEM

- Puede ser susceptible a técnicas de evasión, como: Fragmentación de paquetes, inserción, y cualquier tipo de variación dentro de la fabricación de paquetes de los diversos protocolos existentes (paquetes malformados).

Representantes: Snort, Prelude, Dragon Enterasys, Cisco Real Secure, GFI LANguard,...

2.6.2 IDS por Método de Detección

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

Anomalías

Busca perfiles de comportamientos diferentes a los que tiene almacenados, tales como actividades extrañas por parte de los usuarios, errores de tecleo de contraseñas, uso de sistemas en horarios diferentes a los acostumbrados, integridad de archivos, etc. Se basa en el conocimiento previo que le defina lo normal y anormal, y con ello poder efectuar su análisis (principalmente realizado mediante métodos estadísticos). Su gran desventaja se encuentra en que todo lo que no esté dentro de sus parámetros normales es considerado una intrusión, lo que puede generar un alto número de falsos positivos. Su método de detección se basa en lo que conoce y lo que esté fuera del rango es considerado intrusivo [111,108].

Abusos

A diferencia del método de anomalías, no requiere un entrenamiento previo para diagnosticar una intrusión. Su análisis se basa en la comparación de los patrones o firmas que indican que puede efectuarse un intento de penetración o es definitivamente una intrusión. No genera tantos falsos positivos, debido a que si no está dentro del patrón significa que es normal. Es decir, su desventaja radica en que lo que no esté dentro de sus patrones no será reconocido como un ataque, aunque en realidad este sí lo sea. La metodología de detección dice que lo que no se encuentra dentro de su base de datos es permitido, es decir, no es intrusivo. Es importante que la base de datos se actualice con frecuencia de forma manual o autónoma, para aminorar el acceso a un ataque potencial desconocido [111,108].

2.7 El gran desafío de los IDS: *Credibilidad y Confiabilidad*

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

forma aislada o combinadas entre si, para mejorar su análisis en tiempo o en precisión.

Un ejemplo de esta disyuntiva podría ser un paquete que circula a través de la red con todas las banderas activas (banderas en 1s), lo que generaría una alarma automáticamente indicando que se ha encontrado un paquete con características diferentes a las establecidas dentro de los RFC (Request for comments, documentos sujetos a revisión), esto sería catalogado como intrusivo, puesto que alguien está elaborando paquetes mal formados para distracción y luego acceder a un sistema. Sin embargo, a la clasificación que nos referimos, es algo más complicado, puesto que, puede tratarse de un paquete que tiene las banderas correctas, en apariencia la secuencia de paquetes es correcta, pero en su interior fragmentos de código malicioso, que al ser reensamblados, se convertirán en un ataque directo hacia un sistema [50].

En ese caso, es donde se muestra la verdadera capacidad de análisis de un sistema de detección, para ir revisando por medio de patrones (firmas) o umbrales, las anomalías y abusos que se realizan para acceder hacia un sistema, y no confundirse con el criterio de que es un paquete no intrusivo.

La fiabilidad de un IDS se basa fundamentalmente en la efectividad que tiene para el análisis y detección de intrusiones dentro de un sistema [112]. Se requiere que un IDS pase por pruebas de penetración de ataques conocidos y ataques que no se hayan registrado previamente ante un sistema (desconocidos, hasta el momento), los cuales son creados por medio de herramientas específicas para tal efecto (nessus, por ejemplo), de igual forma el acreditar las pruebas de técnicas de evasión. Esto es con el objetivo de conocer el nivel de seguridad que se puede obtener de dicho sistema de detección y la confianza que se puede tener en él ante las amenazas de intrusión.

2.7.1 Métricas de los IDS

El desempeño que los sistemas de detección de intrusos presentan al momento de detectar una intrusión o la posibilidad de que ésta ocurra dentro de un sistema requiere de ser medido; puesto que es necesario conocer la fiabilidad que se puede tener sobre éstos. Para ello, se han establecido parámetros que permiten expresar los criterios de selección indicados previamente por el dueño del sistema, y de esta manera obtener como resultado paquetes en los que se confía que su contenido no es malicioso y/o intrusivo. Los parámetros que dan la fiabilidad a un IDS, son [50,108]:

- Falso Positivo (*No intrusivas pero anómalas)*
  Este indicador registra que los paquetes de información presentan anomalías en su construcción de acuerdo con los RFC que avalan su operatividad o su comportamiento es diferente al reportado por el dueño de la aplicación. Por ejemplo, tamaño del paquete (anomalía en protocolos) o usuario nuevo en una aplicación.

- Falso Negativo (*Intrusivas pero no anómalas*)
  Indica que los paquetes de información no presentan ninguna anomalía en su

fabricación, pero contienen código malicioso que compromete la seguridad, por ejemplo, los paquetes fragmentados; estos datos son catalogados como datos correctos, cuando en realidad son malignos.

- Verdadero Positivo (*Intrusiva y anómala*)
  Reporta que lo que se ha catalogado como Intrusión es verdaderamente una intrusión o un intento de ataque hacia un sistema.

- Verdadero Negativo (*Ni intrusiva ni anómala*)
  Señala que los datos que se han analizado y no fueron descartados o mandados a cuarentena, son libres de intrusión y/o anomalía.

Interpretación de los parámetros

- *Falso Positivo (Falsas Alarmas)*
  Este parámetro es muy importante y delicado para la medición de un IDS, puesto que *puede indicar erróneamente la presencia de intrusión cuando en realidad no existe.* Ésto se debe a que, si existe un valor muy alto en este parámetro se generan falsas alarmas de intrusión y/o ataques, lo que hace que la credibilidad y confiabilidad sobre un IDS pueda ser prácticamente nula.

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

- *Verdadero Positivo*
  Ratifica que los resultados entregados por el IDS como intrusivos, sí son intrusivos.

- *Verdadero Negativo*
  Confirma que los resultados que el IDS entrega como información no intrusiva ni anómala, es correcta.

2.7.2 Evaluación de los IDS

La forma de evaluar un sistema de detección es mediante la implementación de dos o más sistemas de detección de intrusos en diferentes puntos de un segmento de red o en varios hosts que contengan el sistema a monitorear; estos IDS son inicializados al mismo tiempo para que comiencen la captura y analicen por medio de sus diferentes técnicas de detección la información que ha sido recaudada. Una vez efectuado esto,

se procede a revisar los resultados arrojados. En esta disertación, se considerará como el mejor IDS al que presente los mejores resultados bajo ciertos ámbitos, por ejemplo: emplear menos recursos (hardware), permitir la creación de reportes básicos y personalizados a las necesidades del propietario del sistema (flexibilidad), presentar el menor número de falsas alarmas indicando que algún dato es intrusivo cuando no lo es, así como el mayor número de aciertos que acrediten que la información que se deje pasar es confiable y no intrusiva. Las técnicas que por lo general se utilizan para tal efecto [112], son las siguientes :

- *Técnica de Comparación*

  Esta técnica se emplea generalmente cuando se desea saber el desempeño que tendrá un nuevo prototipo contra uno del tipo comercial, de igual modo es utilizada para seleccionar el mejor IDS de tipo comercial que se adecue a las necesidades personales o empresariales sobre un sistema a proteger. Se efectúa a través de la implementación de diferentes tipos de IDS dentro de una red o en un host específico para que capturen y analicen respectivamente la información por medio de sus diferentes técnicas de detección.

- *Técnica de Efectividad*

  Los resultados que se pueden obtener de la técnica de comparación, dan a conocer a el IDS que presentó mejor desempeño en la captura masiva de información, así como en las técnicas de detección, entre otros. Sin embargo, la pregunta es: *¿cómo saber si efectivamente está detectando de forma correcta?* Para ello se emplea la técnica de efectividad, la cual consiste en implementar en el mismo segmento de red o en el mismo equipo a monitorear un sniffer (herramienta que escucha todo lo que pasa alrededor y dentro de él). Se realiza por un periodo de tiempo previamente acordado para que tanto el IDS y el sniffer comiencen a capturar, después se detiene el reloj y se comienza el análisis minucioso. El IDS realiza sus diagnósticos y entrega sus resultados, los que son corroborados a través del sniffer, es decir, se revisan los intentos de intrusión que se efectuaron contra los detectados por el IDS, al igual que los no detectados, posteriormente se realiza el estimativo en porcentaje, tomando al 100 % como valor máximo de detección. Con ésto, se puede tener el porcentaje real que un IDS es capaz de tener ante los intentos de intrusión en un ambiente controlado o de producción.

- *Técnica de Penetración*

  Existen diversas herramientas para penetrar en un sistema, las cuales contienen utilerías para emplear paquetes de red malformados, ataques conocidos o que permiten la creación de éstos con ciertas variaciones. Estas herramientas pueden simular desde ataques básicos hasta ataques sumamente agresivos que pueden suspender el servicio de un equipo. La evaluación consiste en el reconocimiento de ataques que sean o no conocidos y se demuestre el mejor desempeño para adaptarse a las variaciones en el reconocimiento de éstos, obteniendo el nivel de seguridad

que se tendrá ante una intrusión potencial o baja. Generalmente estas pruebas se emplean para evaluar que tan protegido se encontrará un sistema que se encuentre en una red, en un equipo o dentro de una aplicación específica.

2.8 Métodos de Detección

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

- **Anomalías** *(perfiles)*
  Filosofía: Lo que está explícitamente prohibido, no está permitido.

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

- **Abusos o uso indebido (patrones o firmas)**
  Filosofía: Lo que no está explícitamente prohibido, está permitido

  A diferencia del anterior, este método no requiere un entrenamiento previo para diferenciar que es lo normal o anormal, y con ello diagnosticar la existencia o intento de una intrusión. Su análisis se fundamenta en la comparación de los posibles eventos (patrones) o características únicas y propias (firmas) que determinen la presencia de una intrusión. Cada patrón es una secuencia de los posibles pasos que puede seguir un atacante, así como las variaciones de éstos. Las firmas contienen indicios de ataques conocidos que al ser cotejados, verifican si se cumple o no la condición establecida. No genera un gran número de falsos positivos, esto se debe a que lo que no coincide con sus patrones o

firmas es considerado como normal. Sin embargo, a su vez este comportamiento es su mayor desventaja, puesto que puede generar un número significativo de falsos negativos al momento de marcar la información como ausente de técnicas hostiles, cuando en realidad es un ataque. Presenta otra desventaja, requiere mantener actualizado su contenedor de firmas o patrones (base de datos) de forma manual o autónoma, para evitar caer en un número alto de falsos negativos, de lo contrario cualquier acción dañina no reconocida, afectará al sistema que se desea proteger.

2.9 Modelos de detección

Se han creado diferentes prototipos para representar diversas teorías sobre la forma de detectar intrusos dentro de un sistema [105, 106, 108]. En ellos se expresan conceptos que permiten esquematizar comportamientos y/o patrones, que permitan facilitar la comprensión de las características que describan un ataque o intrusión. De tal forma, que se pueda inferir sobre ellos el conocimiento necesario para prevenir o retardar actos ilegales antes de que sean perpetrados dentro de un sistema. Los arquetipos más desarrollados para la detección de intrusos que se han observado durante esta investigación, se basan en:

- Modelos

  Representan a un sistema complejo con el objetivo de facilitar su comprensión y comportamiento, de tal forma, que se puede reconocer nuevos tipos de ataque a través del diseño y análisis de operación de éstos. En ellos se reproducen diversos eventos que se presentan en un problema actual o que todavía no se han presentado dentro de un sistema, es decir, plantean la posibilidad de que suceda un evento y de esta forma adquirir el conocimiento para anticiparse a una nueva técnica de intrusión [7, 23, 51].

- *Firmas*

  Su objetivo es comparar el contenido de cada paquete con una base de datos previamente enriquecida con las características específicas (firmas) que identifican una intrusión o intento de acceso ilegal hacia un sistema (ataque) [17, 21].

- *Patrones*

  Analizan los datos obtenidos a través de un sensor y se cotejan contra umbrales de comportamiento previamente establecidos del tipo humano, aplicación, procesos del sistema, entre otros [25, 36, 61, 79, 85].

- *Clasificadores*

  Examinan el conjunto de datos adquiridos (en tiempo real o fuera de ese periodo) designando cuales son de carácter maligno y los que son inofensivos. Una vez que es determinada el tipo de información que se trata, ésta es pasada nuevamente por filtros que corroboran a mayor profundidad la

clasificación previamente hecha [49, 62].

- *Auto-aprendizaje*
  Se basan en el conocimiento previamente adquirido, el cual se emplea para reconocer los diferentes intentos de intrusión y los que no conoce los procesa para inferir un nuevo conocimiento, del manera, que la siguiente vez que realice su análisis de detección, pueda reconocer un mayor número de técnicas de intrusión y de evasión que son empleadas por los intrusos sobre los sistemas de detección [57, 71 ,73, 78, 83, 93].

2.10 Descripción de Técnicas

Los sistemas de detección de intrusos tienen como objetivo el detectar si existe o no una intrusión dentro de un sistema. Para ello, se utilizan criterios de análisis basados en abusos y en comportamientos anómalos (patrones y perfiles, respectivamente). Esto es posible efectuar, por medio del uso de diferentes áreas de investigación, de las cuales podemos citar: la Inteligencia Artificial, Métodos Estadísticos, Redes Neuronales, Minería de datos, entre otras. Cabe aclarar que las técnicas empleadas en una detección no están directamente enfocadas a un tipo de IDS, sino que se ocupan de manera aislada o conjunta, en base a su flexibilidad y potencialidad para detectar intrusos. Algunos modelos que han sido propuestos por diversas universidades han combinado técnicas para obtener mejores resultados en la búsqueda de intrusos dentro de un sistema [98, 105, 108, 110].

- Agentes Móviles
  Los agentes son una entidad que actúa de manera autónoma, pero en colaboración con otros agentes para detectar una intrusión en un sistema. El agente obtiene información que permita reconocer la presencia de un intruso y la envía a un motor de análisis, quién dictamina si existe o no una intrusión. Si se halla algo importante es comunicado a los otros agentes para tomar las medidas pertinentes [34, 65].

- Algoritmos Genéticos
  En la actualidad se empieza a incursionar en esta área para detectar una intrusión. Debido a que su aplicabilidad al concepto de evolución biológica, permite emplearse como clasificador de lo bueno o malo de la información que fluye en una sistema [28, 62].

- Árboles de decisión
  Estos permiten modelar un proceso de tomas de decisiones sobre lo que se considera normal o anormal en un sistema. Los nodos de los arboles representan la disyuntiva y los arcos las alternativas [56, 105, 108, 110].

- Escenarios
  Se desarrollan modelos con las posibles técnicas de evasión, ataques y

comportamientos extraños que permitan anticipar la llegada de un intruso a un sistema [33, 40, 51, 59, 61].

- Grafos
  Es la representación gráfica de una intrusión o ataque específico por medio de nodos, que muestran el comportamiento de un sistema ante un acto hostil hacia el mismo [106].

- Lógica de Predicados
  Esta técnica es un sistema deductivo formal, que utiliza predicados, conectores lógicos y cuantificadores para inferir conocimiento a partir de ataques conocidos [30].

- Lógica Difusa
  Sistema deductivo formal que utiliza criterios flexibles (valores entre 0 y 1) de verdad [57, 62].

- Máquinas de Estado
  Se emplean para modelar el comportamiento de un sistema y descubrir nuevos tipos de ataques. El comportamiento de un sistema es representado a través del cambio de estados (nodos), los cuales se presenta cuando ocurre una acción que indique la transición. Los estados no sólo dependerán de sus entradas actuales, sino también de los anteriores, para formar los antecedentes que expliquen el comportamiento actual [106].

- Máquinas de Soporte Vectorial
  Es un conjunto de métodos de aprendizaje supervisado, que se basan en la separación lineal de los datos de de entrada. Si la distancia es corta, se identifica como una posible intrusión [108].

- Métodos Estadísticos
  Esta técnica reacciona a umbrales establecidos previamente sobre los parámetros a evaluar en un sistema, por ejemplo: comportamientos humanos, de procesos, de servicios, etc. Si estos valores son excedidos, se considera que es una intrusión hacia un sistema [55, 106].

- Minería de Datos
  Esta técnica permite extraer patrones o modelos de ataques desconocidos, de un conjunto de datos recolectados por un IDS [3, 5, 11].

- Modelo de Markov
  Es un modelo que representa la probabilidad de que un estado pueda pasar de un estado actual a otro, es decir, es una probabilidad condicionada, en el que el nuevo estado depende totalmente del estado anterior. En la detección de intrusos se emplean las cadenas de Markov para representar la transición entre eventos condicionados y determinar la existencia de una intrusión.

También se han empleado los modelos ocultos de Markov (HMM) para conocer el estado anterior, cuando sólo se conoce el estado actual [13].

- Ontologías
  Permiten realizar una representación formal de un conjunto de conceptos (ataques) y sus relaciones sobre un dominio [58].

- Reconocimiento de Patrones
  Se basa en las formas o patrones conocidos de ataques e intrusiones, comparan cadenas de texto que vienen en el contenido de un paquete (payload), y/o anomalías en las cabeceras de los protocolos de comunicación [7, 17, 21].

- Redes Bayesianas
  Se emplean como modelos gráficos para representar la dependencia entre un conjunto de variables, a través de datos probabilísticos que indiquen la probabilidad de que un evento hallado sea una intrusión [49].

- Redes de Petri
  Son utilizados para la representación gráfica de eventos que pueden presentarse en una intrusión. La transición entre estados sucede cuando se cumple el evento. Esta técnica permite modelar ataques complejos, en los que se incluyen sus características particulares de comportamiento [111].

- Redes Neuronales
  Es una área que está incursionando al igual que otras áreas como los Algoritmos Genéticos y el Sistema Inmune, en el campo de la detección de intrusos. La red neuronal es entrenada con compartimento normales o anormales (estos valores dependen de la forma en que se desee detectar una intrusión). Mediante su empleo es posible detectar variaciones de ataques o de carácter desconocidos, que difieren de los patrones iniciales con que fue entrenada la red [25].

- Sistema Inmune
  Mediante esta técnica se efectúa la analogía con el sistema inmune del ser humano, para detectar una intrusión. En la que se establece lo que se considerará propio y lo que ajeno en un sistema [95, 96].

- Sistemas Expertos
  Conjunto de reglas con la estructura IF-THEN-ELSE, en la que si se cumple la regla la intrusión o ataque a buscar, es confirmado [27].

- Métodos Heurísticos
  Emplea el resultado que es generalmente obtenido a través de algún método estadístico, para ajustar un umbral de detección de lo normal y anormal que se presenta en un sistema. Tratando con ello, de aminorar el número de

falsos positivos y negativos en un IDS [106].

2.11 Cronología de los IDS

A continuación se muestran algunos de los trabajos realizados para la detección de intrusos, los cuales son clasificados y citados en orden cronológico de acuerdo a su técnica de detección [99, 105, 108,110]

Métodos Estadísticos

- IDES *SRI International, Año 1987.*
- NIDX *Bell Communications Research,* Año 1988.
- HyperView *CS Telecom,* Año 1992.
- NIDES *SRI International,* Año 1995.
- EMERALD *SRI International, Año 1997.*

Sistemas Expertos

- HayStack *Laboratorios Haystack, Año 1988.*
- NIDS *Network Flight Recorder, Inc., Año 1988.*
- MIDAS *SRI International, Año 1988.*
- Wisdom & Sense *Los Alamos National Laboratory, Año 1989.*
- Computer Watch *AT&T Bell Laboratories, Año* 1990.
- ISOA *Planning Research Corp., Año* 1990.
- NADIR *Los Alamos National Laboratory, Año 1990.*
- AudES *IBM Los Angeles Scientific Center, Año 1990.*
- IIDS *Bogazici University, Año 2000*.

Redes de Petri

- IDIOT *Purdue University,* Año 1990.

Transición de Estados

- STAT *University of California at Santa Barbara, Año 1992.*
- USTAT *University of California at Santa Barbara, Año 1992.*
- NetStat *University of California at Santa Barbara, Año 1998.*
- GrIDS *Universidad de California at Davis, Año 1999.*

Reconocimiento de patrones

- NFR *Network Flight Recorder, Inc., Año 1997.*
- CMDS *ODS Networks, Inc.,* Año 1998.
- NetProwler *AXENT Technologies, Inc., Año 1998.*

- NetRanger *Cisco, Año 1999.*
- Real Secure *Internet Security Systems, Año 1999.*
- Snort *Martin Roesch, Año 1999.*
- BRO *Lawrence Berkeley National Laboratory, Año 1999.*
- Dragon *Enterasys Networks, Inc.*
- E-trust *Computer Associates.*
- BlackIce *Network ICE Corp.*

Redes Neuronales

- ACME *University of Sao Paulo, Año 1998.*
- NNID *University of Texas, Año 1999.*
- PHAD/ALAD *Florida Institute of Technology Technical Report CS-2001-04, Año 2001-2003.*

Algoritmos Genéticos

- GASSATA *SUPELEC, Cesson Sevigne, France Año 1998.*

Minería de datos

- JAM *Columbia University,* Año *1998.*
- MADAM ID *Columbia University, Año 1999.*

Árboles de decisión

- ADAM *George Mason University, Año 2001.*

Lógica difusa

- NFIDS *Informatics & Stat. Center, Tehran Univ., Iran, Año 2003*

Redes Bayesianas

- EBayes *SRI International, Año 2000.*

Sistema inmune

- AIS *University of New Mexico Año 1999*

Agentes móviles

- AHA! IDS *Texas A&M University and United States Military Academy Año 2000.*
- AAFID Purdue University, Año 2000.

2.12 Detección de un IDS por su punto de conexión

Los sistemas de detección de intrusos pueden ser implementados en diferentes puntos de una red de cómputo o en un equipo específico, los cuales pueden operar de manera conjunta o aislada sobre un sistema. Cada punto en el que se ubique un IDS presenta ventajas y desventajas con respecto al daño potencial al que se ven expuestos a enfrentar, así como el nivel de protección que pueden brindar en cada punto (Fig. 2.2) .

Los sitios que generalmente se contemplan para su implementación [106, 112], son: Antes del cortafuegos, después del cortafuegos, en una subred o red única y en un sólo equipo.

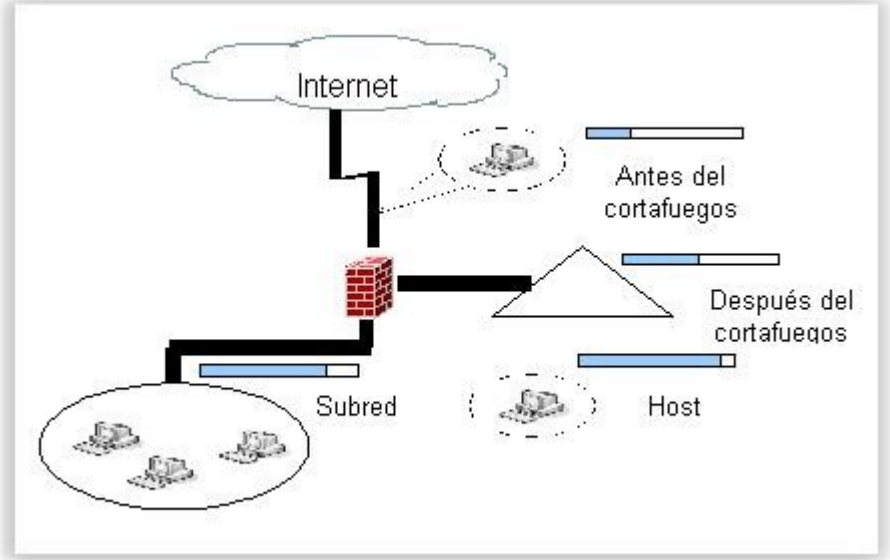


Fig. 2.2: Grado de Granularidad de un IDS en una RED

2.12.1 Antes del Cortafuegos

El IDS se implementa con la ideología de analizar los paquetes que provienen de la red externa hacia la red interna. Se ubica entre la salida hacia Internet y el cortafuegos, ésto permite realizar un análisis masivo en la intercepción de paquetes o solicitudes hacia los sistemas de aplicaciones que se encuentran dentro de los servidores y a los servicios que brindan. Un IDS en ese punto requiere de una gran capacidad para el manejo del tráfico entrante. Presenta los inconvenientes de que su detección en cierto modo no puede ser profunda, en virtud de que este tipo de análisis retrasaría la entrega de paquetes hacia la red interna. Asimismo, pueden presentarse problemas de pérdida de paquetes al no ser capturados en su totalidad por el IDS, debido a la gran carga (cantidad de información) de la red [24]. Otra desventaja es la de presentar un gran número de falsos negativos, originados por el descarte rápido de paquetes.

Recomendación: Evitar la pérdida de paquetes por el exceso de tráfico que proviene del exterior.

2.12.2 Después del Cortafuegos

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

La DMZ (Zona desmilitarizada, por sus siglas en inglés) es la configuración que se emplea para crear una subred aislada, en la que se controla el tráfico interno y externo de una red, el cortafuegos establece una división entre la red externa y la DMZ de la siguiente manera:

Está permitido:

- El tráfico que proviene de la red externa hacia la DMZ.
- El tráfico que proviene de la red interna hacia la DMZ.
- El tráfico que proviene de la red interna hacia la red externa

No está permitido:

- El tráfico que proviene de la red externa hacia la red interna
- El tráfico que proviene de la DMZ a la red interna.
- El tráfico que proviene de la DMZ a la red externa.

Generalmente en esa zona se encuentran los servidores de aplicaciones que dan servicio tanto a la red interna como a la red externa por lo que este tipo de configuración es una protección que se tiene hacia la red interna.

Debido a la información sensible que se maneja dentro de los servidores, es necesario implementar un IDS que trate de garantizar. la disponibilidad, la integridad, la

autenticidad y confidencialidad de los sistemas que se ejecutan dentro de los servidores. En este punto el IDS tendrá un desempeño de medio a alto grado, dependerá de la profundidad de análisis que se desee y aminorar el factor riesgo que se presente sobre ellos. El IDS se integra a la DMZ como si fuera otro servidor, con la diferencia de que estaría en la escucha y no dando servicios a la red (IDS tipo Red) o puede ser colocado dentro de uno o varios servidores, donde se requiera un monitoreo y detección minucioso (IDS tipo Host).

Es importante enfatizar que este punto es muy sensible y se requiere de un monitoreo continuo para vigilar las intrusiones que evaden el cortafuegos, sin embargo, se debe considerar que este análisis no debe afectar los niveles de servicio de las aplicaciones que se encuentran ejecutándose en los servidores.

Recomendación: Ser restrictivos en la detección de intrusos sin afectar la disponibilidad de los servicios que brindan los servidores.

2.12.3 En una subred o red única

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

Recomendación: Ser lo más restrictivos en la detección de intrusos en esta ubicación.

2.12.4 En un sólo equipo

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

Recomendación: Ser restrictivos sin afectar los recursos del sistema que se evalúa.

2.13 Tendencia de los IDS

El tipo de respuesta que presentan los IDS ante una intrusión es pasiva, esto es, los IDS revisan la información que fluye a través del tráfico de red o sobre un host, y generan las alarmas correspondientes en espera de la acción correctiva por parte del administrador del sistema. A lo largo del tiempo, los IDS han ido mejorando sus técnicas de detección, permitiendo con ello, retroalimentar su motor de análisis y adaptarse a los nuevos intentos de intrusión. Sin embargo, se presenta una gran desventaja en ellos, a pesar de que los sistemas de detección de intrusos detectan lo que otros mecanismos de defensa no lo hacen (debido a sus condiciones de fabricación o que son burlados por técnicas de evasión); existen situaciones que requieren de una respuesta inmediata y detener el avance de la intrusión [50]. Debido a esta situación, se han creado nuevas herramientas para detener y contraatacar las intrusiones sin espera del administrador, tales como: los Sistemas de Prevención de intrusos (IPS, por sus siglas en inglés).

2.13.1 IPS (Intrusion Prevention System)

Los Sistemas de Prevención de intrusos (IPS) son Sistemas que detectan intrusiones dentro de los sistemas, a los que se les añadió la funcionalidad de un cortafuegos para filtrar paquetes y tomar acciones de filtrado. Existen discrepancias en el área de la Seguridad Informática, por el papel que desempeñan estas herramientas, algunas opiniones indican que los IPS son los sucesores de los IDS y otros afirman que son distintos productos que se complementan entre si [103, 106].

Debido a la premisa de que los IDS son detectivos y no correctivos [54]. El IPS difiere del IDS al tener una respuesta reactiva cuando ejerce control sobre los paquetes malformados, resetea las conexiones maliciosas, si fuese necesario puede reconfigurarse a sí mismo para rechazar conexiones de un puerto específico, entre otros. La reconfiguración es obtenida por el resultado del aprendizaje que se obtiene con un nuevo ataque no registrado en su Base de Datos, presentándose como una reacción para contraatacar y adaptarse a lo que sucede a su alrededor.

2.14 Otras opciones de detección

Existen otras opciones para la detección de intrusos que permiten conocer el comportamiento de los atacantes para aprender nuevas técnicas de intrusión, y complementar con ello, su motor de análisis. Entre ellas se encuentran los honeypots y las honeynets.

2.14.1 HoneyPots

Los HoneyPots (Tarros de Miel) son una solución que surgió como respuesta a una intrusión de un atacante, que no era detenido por los métodos convencionales (Cortafuegos, Detectores de Intrusión, VPNs, entre otros). Debido a esta situación, se implementó un servidor similar al que se tenía en producción, considerando que su

acceso fuera restrictivo, pero no imposible de penetrar, dejando para ese fin, algunos puntos vulnerables que le hiciera caer en la trampa. El engaño permitió observar las diferentes metodologías y técnicas que el atacante utilizaba para la intrusión, generando con la información recabada un plan de contraataque

Los servidores HoneyPots (también conocidos como Sistemas de Decepción) son servidores señuelos [106, 120], que contienen sistemas similares o idénticos a los de producción, son colocados de manera aislada (subred alejada de los servidores en producción) para distraer el atacante y aprender al mismo tiempo de él. Estos servidores requieren de mucha atención, por que un descuido en la configuración podría revelar al atacante que es un servidor señuelo y no el de producción, como se pretende; provocando con esto que el atacante tome respuestas hostiles ante la situación.

Adicional a este requerimiento, es de vital importancia que la persona que está monitoreando a los honeypots, presente una amplia experiencia en el área de seguridad, así como un conocimiento profundo en los protocolos de comunicación, técnicas y ataques de intrusión ya conocidos. Esto con la finalidad de estar prevenidos ante cualquier eventualidad que pueda presentarse por parte del atacante al ser descubierto el engaño.

Los HoneyPots pueden ser desarrollados para las áreas de:

- *Investigación*
  Son servidores de aplicaciones que se implementan en ambientes controlados para el estudio de un ataque específico o para conocer nuevas técnicas de intrusión, las cuales pueden ser modeladas y llevadas al motor de análisis de un IDS, y de esta manera éste se encuentre preparado para ataques de las características reportadas por los honeypots. En un ambiente controlado los investigadores manipulan variaciones sobre ataques previamente conocidos, observan y analizan los resultados. Un honeypot puede estar implementado en una subred que puede estar expuesta hacia una red exterior o sólo pertenecer a una red aislada sin afectar a la red interna. Puede estar constituido por un honeypot o varios de éstos, en los que se recabará todo tipo de información que pueda ser analizada (en línea o posteriormente al evento) mediante un cliente, y así, analizar las posibles técnicas de ataque que puedan emplear los atacantes para introducirse en un sistema, como: vulnerabilidades, nuevas herramientas, shellcodes, etc.

- *Producción*
  Son servidores que se colocan de manera deliberada dentro de una red real, es decir, no está bajo un ambiente controlado. En un ambiente de producción son implementados no para atrapar a un intruso, sino para distraerlo. En ciertas circunstancias los intrusos dependiendo del motivo personal que tengan sobre un objetivo, no cesarán hasta penetrar en él. Lo que resulta muchas veces incómodo y molesto dentro de una organización,

adicionalmente del riesgo que representa el tener un atacante rondando en los límites de seguridad que se han establecido. Debido a esta situación se implementa un servidor en apariencia idéntico al que se tiene en producción (servidor que ocupan todos los usuarios de una red) exceptuando por los datos almacenados y las vulnerabilidades que se dejan asomar como señuelo, para que el intruso esté distraído sin saber que sus movimientos se registran de manera oculta y serán revisados posteriormente por el administrador del sistema, quién será el encargado de tomar las acciones correctivas a dichos intentos de intrusión. Su implementación se hace en una subred distinta a los servidores de producción.

La implementación de un honeypot que es colocado en un ambiente de producción (ambiente controlado en contacto con una red exterior), es considerado un arte, puesto que dependerá de qué tan real y atractivo se muestre el servidor para el atacante y éste pueda ser considerado un éxito (atrapar y/o distraer a los intrusos) o un fracaso en su objetivo . Un honeypot debe ser lo suficientemente complicado para dar batalla a un intruso, pero a su vez, vulnerabilidades que le permitan tomar confianza y no le hagan tener sospecha a un intruso de que se trata de una trampa para conocer sus tácticas y éste al darse cuenta, tenga una reacción hostil hacia el sistema que se protege.

Los Honeypots son una gran aportación a los sistemas de detección de intrusos, debido a la exposición de las diversas técnicas que emplean los intrusos para penetrar un sistema. Permitiendo con esto, tener una mayor visión de lo que podemos esperar de la creatividad de la mente humana y con ello, desarrollar modelos de contraataque o que retrase el proceso de intrusión ante la presencia de un ataque que varíe de su forma original.

2.14.2 HoneyNets

El crecimiento de las redes en su extensión física y en el volumen de su información, ha dado lugar a la generación de los ambientes distribuidos. Un ambiente distribuido como su nombre lo indica, distribuye la carga entre varios servidores, conocidos generalmente como granja de servidores (por ser físicamente un número mayor a 2) y el equipo de comunicaciones que conforma una red, con el objetivo de brindar control y administración sobre éstos, y al mismo tiempo otorgar los servicios pertinentes a los usuarios de la red.

Ante esta situación, se desarrolló el concepto de HoneyNets para simular ambientes de producción que están constituidos por granjas de servidores, los que pueden ser atacados de manera simultánea o usados como trampolín para comprometer otras redes. Presentan las mismas características de implementación que los honeypots de investigación o producción. En los que se busca reconocer las técnicas a emplear de forma masiva y remota por un atacante. Una HoneyNet está constituida al menos por dos servidores honeypots idénticos a los de producción.

Capítulo 3
Descripción de la problemática

Resumen

Los sistemas de detección de intrusos buscan los parámetros indicativos de una posible intrusión, éstos pueden detectar un intento de intrusión cuando se encuentre algo que sea diferente a sus parámetros normales o cuando lo hallado coincida con un comportamiento anormal en un sistema. Si el atacante es hábil logrará cumplir su objetivo (vulnerar un sistema), en caso contrario, dejará indicios de las posibles estrategias que seguirá para lograrlo. En el presente capítulo se exponen las estrategias y herramientas más comunes que son empleadas por un atacante, para evadir los mecanismos de seguridad en un sistema. Se hablará sobre el papel que ha desempeñado el sistema de detección de intrusos de nombre SNORT dentro del campo académico; sus logros y avances significativos obtenidos por diferentes universidades. Así como la flexibilidad que presenta para el desarrollo de nuevas propuestas de Sistemas de Intrusión a través de la explicación de los componentes que lo integran y sus modos de operación para ser uno de los mejores en su ramo.

Objetivos

- Diferenciar los conceptos de amenaza, intrusión y ataque.
- Definir a quién nos enfrentamos, expresando sus motivaciones y habilidades personales.
- Mostrar los tipos de ataques más comunes que pueden ser efectuados para realizar una intrusión, así como las herramientas para detener y/o retrasar en tiempo a un intruso.
- Presentar a Snort como un sistema de detección de intrusos de Red y su relación con el área científica.

3.1 Introducción al capítulo

En la actualidad la evolución de la tecnología, ha hecho posible que el uso de los sistemas de información forme parte de nuestra vida, ya sea como herramienta de trabajo, medio de comunicación, transacciones bancarias, entre otros. De tal forma, que la información contenida en estos sistemas, se torne vital para sus usuarios cotidianos y a la vez, de interés para personas ajenas, quienes realizan ataques e intrusiones hacia equipos de cómputo, sitios web de diversos giros, gobierno e incluso fuerzas armadas. Los ataques que se perpetran pueden ser de índole interno y externo, siendo sus móviles: la curiosidad de conocer nuevos sistemas, la venganza y los fines económicos.

En la mayoría de los ataques que se realizan hacia un sistema, se observa como éstos traspasan fronteras informáticas por intrusos que se inmiscuyen sigilosamente para obtener su objetivo. Atrás de esas grandes hazañas, existe un amplio conocimiento de diferentes áreas como: redes, programación en lenguajes de bajo y alto nivel, sistemas operativos, entre otros que representan un gran peligro para comprometer la información que radica en su interior.

A lo largo de este capítulo se darán definiciones importantes para denotar de la forma más correcta y apegada a la realidad, las técnicas y tácticas empleadas en una intrusión. Se expondrán los móviles a detalle que propician la intrusión, las diferencias entre los conceptos de amenaza, ataque e intrusión. El grado de intrusión que podemos esperar en los sistemas y cómo adaptarnos ante las vulnerabilidades que presentan. De igual forma, se dará una descripción breve de los ataques que se han generado y el potencial que representa cada uno de ellos.

3.2 Terminología

Las siguientes terminologías son expresadas de acuerdo al argot de la seguridad informática [120,111]:

Sistema

Un sistema es un conjunto de elementos interrelacionados por medio de software y/o hardware con un propósito específico. El hardware puede estar conformado por un host (ordenador), un conjunto de hosts y/o dispositivos de red (periféricos y/o dispositivos de comunicación). Un host está constituido por archivos y aplicaciones (software) que se relacionan para formar un sistema. Este sistema es considerado en base a su tamaño como un sistema de nivel micro (sistema de sólo host) y /o un sistema de nivel macro (sistema de red, conjunto de hosts)

Ataque

Un ataque es la acción hostil que realiza un atacante hacia un sistema. Este ataque puede o no dañar a un sistema, esto depende del criterio del atacante. Los ataques han ido evolucionando gradualmente, inicialmente se dedicaban a la captura de la información que fluía en un sistema. Después irrumpían en un sistema para modificar su información, posteriormente explotan las

vulnerabilidades que presentan los protocolos de comunicación y el diseño de aplicaciones. En la actualidad conjuntan los ataques antes mencionados para dejar en un estado de no disponible un sistema. El hecho de que un sistema no esté disponible, ocasiona diversos problemas del tipo económico y social, esta situación afecta indirectamente a otros sistemas que requieren de sus servicios para efectuar su operación habitual.

Amenaza

Es la posibilidad de que bajo el cumplimiento de ciertas condiciones, un sistema pueda estar en riesgo de sufrir cualquier tipo de daño.

Intrusión

Se considera intrusión al momento en que el atacante está dentro del sistema, en caso contrario, es indicado como un intento de vulnerar los mecanismos de seguridad que protegen un sistema.Una intrusión la podemos definir en términos de seguridad, como el hecho de traspasar los límites de la confidencialidad para comprometer a un sistema. La intrusión es una irrupción o intromisión sin permiso del propietario del sistema.

Intento de intrusión

Intento de Intrusión es cuando un atacante trata de acceder a un sistema, pero no puede lograrlo por que existen otros mecanismos de seguridad, que le impiden llevar a cabo sus planes. Sin embargo, deja indicios de hacia donde desea encontrar una vulnerabilidad.

Es importante diferenciar entre una intrusión, un ataque, un intento de intrusión y una amenaza; debido a que cada uno indicará el tipo de riesgo al que está expuesto un sistema.

3.3 Análisis de Riesgo

La intrusión es el hecho de estar en el interior de un sistema sin autorización del propietario, la cual irrumpe en la confidencialidad, viola la integridad de la información (modificación o eliminación), enmascara la autenticación ante un sistema (validación de identificación) y atenta contra la disponibilidad de la información o equipo en el que radica dicho sistema.

Un ataque se mide de acuerdo al daño que puede causar hacia un sistema, es decir,en base al factor de riesgo que se pueda presentar en un sistema ante un atacante [111].

El factor de riesgo es un valor que se asigna en grado de importancia y de disponibilidad a una aplicación, sistema o equipo de cómputo que integre la red. El valor estimativo será variable, puesto que éste estará sujeto a los criterios de una persona u organización a los que les pertenece un sistema (Tabla 3.1). Para que exista el factor de

riesgo en un sistema, deben cumplirse previamente ciertos eventos que lleven a un sistema a situarse en un estado de alerta, como la indicación de un posible daño.

Para establecer los niveles del factor de riesgo que tendrá un sistema, se deben de considerar los siguientes puntos:

- Información contenida dentro de los Sistemas de Cómputo
- Servicios a brindar por el sistema.
- Computadoras y dispositivos que integran la red.
- Las fortalezas y debilidades que ofrece las arquitecturas y la tecnología en el momento actual.
- El lugar físico donde se coloca un sistema.
- Los usuarios y controles de acceso que se emplearán para acceder a un sistema.

Por ejemplo:

Tabla 3.1: Ejemplo del nivel de riesgo en un sistema

Descripción	Accesos	Nivel de Riesgo *
Aplicación A	Puerto 25 y 111	4
Aplicación B	Puerto 21	3
Aplicación C	Puerto 80	4
Aplicación D	Tráfico de Red	5
Aplicación E	Proxy	5
* Escala: 1 – 5, siendo uno el de menor riesgo y cinco el grado máximo		

3.3.1 Grado de Intrusión

La intrusión sólo puede ser perceptible siempre y cuando se produzca una sintomatología dentro de un sistema [111], ésta puede ser percibida en diferentes formas, por ejemplo: mensajes emergentes, aumento en el consumo de los recursos propios de un sistema, modificación de la información almacenada, etc.

La sintomatología hace referencia a las evidencias que se ven reflejadas en las bitácoras de un sistema, ésta es diferente de la reacción que pueda presentar un sistema, puesto que, una reacción será la consecuencia que se obtiene ante un estímulo externo. En el caso de un ataque de denegación de servicios, su reacción será la falta de disponibilidad de los servicios que brinda un sistema, al momento de ser solicitados por los usuarios de dicho sistema.

Esto nos lleva a decir, que los sistemas de detección de intrusos buscarán una sintomatología o patrón, que les indique un comportamiento diferente a los estándares previamente establecidos por el propietario del sistema. Esta sintomatología al ser coincidente con alguno de los patrones almacenados, dará la pauta para que el IDS realice la indicación de una posible intrusión.

Ahora, Imaginemos la siguiente situación:

Una habitación que tiene una sola puerta hacia el exterior y sin ventanas, en su interior se encuentra un florero. Esa habitación es cerrada con una llave. Si extraviamos la llave y alguien ajeno entra legalmente a la habitación, puesto que tiene la llave para hacerlo. ¿Cómo podemos saber si alguien estuvo ahí durante nuestra ausencia, si no nos hemos percatado que se extravió la llave? La manera de detectar que alguien irrumpió en la habitación, sería si nos percatamos que el florero cambió de posición o que no se encuentra en el interior de la habitación o éste ha sufrido algún daño físico (estrellado o quebrado). Esos serían síntomas que indicarían que hubo una intrusión hacia esa habitación. Sin embargo, si encontramos en apariencia todo intacto ¿cómo podríamos saber si alguien encontró la llave y estuvo curioseando por ahí? O ¿cómo tener la certeza de que nadie accesó a la habitación durante el extravío de la llave?

Tal vez las respuestas a estas interrogantes, pudieron ser claramente expuestas si dentro de esa habitación existiese una cámara de video. Con ella se podría observar quién nos suplantó y entró a esa habitación sin nuestra autorización ¿De otra forma, cómo podríamos evidenciar que alguien se introdujo sin nuestro consentimiento?.

El grado de intrusión en un IDS está implícitamente relacionado con esta situación (criterios de restricción). El grado de intrusión al que se refiere, es al nivel de restricción que se la asigna a un IDS para efectuar su análisis de detección en un sistema (de mayor a menor), es decir, con él se establece la granularidad (revisión minuciosa y profunda) y el rigor con el que se aplicarán los criterios de discriminación, sobre la información que fluye en un sistema.

Se encuentra regido de manera directa por el factor de riesgo que puede sufrir un sistema, por lo que sus criterios empleados como parámetros para realizar un análisis ligero o profundo de una intrusión, variarán de acuerdo al punto donde el IDS se sitúe y las necesidades propias que demande la protección a dicho sistema. Cabe mencionar, que un IDS entre más restrictivo sea para detectar una intrusión, requerirá de una mayor inversión en tiempo, dinero y esfuerzo humano para analizar los datos resultantes de la detección efectuada por el mismo.

De manera indirecta, la granularidad del análisis que se desee manifestar en un IDS, se verá influenciado de manera proporcional por los valores resultantes del desempeño de un IDS, al momento de aplicar los criterios de discriminación para la evaluación de una intrusión (falsos positivos y falsos negativos).

3.3.2 Políticas de Seguridad

Al tener en claro el grado de Intrusión que se desea aplicar en un IDS, se pueden generar las políticas correspondientes para proteger o actuar en caso de una intrusión hacia un sistema. Las políticas de seguridad varían de acuerdo a las necesidades propias de los usuarios de un sistema [111] , éstas pueden ser de acuerdo a su nivel de restricción:

- *Permisivas*
  Lo que no está estrictamente prohibido está permitido. Las políticas permisivas exponen lo que está permitido hacer en un sistema, pero no sus restricciones. Esto se refiere a, que si no hemos hecho la aclaración de lo que está prohibido dentro de un sistema, se considerará como permitido.

- *Restrictivas*
  Estas son los contrario a las permisivas. Lo que está permitido no está estrictamente prohibido. Presentan dos grandes desventajas, requieren mayor nivel de auditoria y del expertise de los analizadores de seguridad. Entre más restrictivo sea un sistema más atractivo es para un atacante.

- Correctivas
  La seguridad de un sistema siempre será relativa. Debido a esta realidad, se emplean mecanismos de seguridad para retrasar una intrusión. Gracias a este retraso se han podido realizar políticas correctivas en tiempo real, en las que los IDS se retroalimentan de ataques desconocidos, que pasan ahora a ser conocidos por éstos, y evitar futuros ataques con esas características.

3.4 ¿A quién nos enfrentamos?

El crecimiento de la tecnología se ve inmerso en el universo de las redes computacionales, como un medio para efectuar el intercambio de información de manera global entre sistemas. Lo que ha puesto como manifiesto, el riesgo al que se ve expuesta dicha información ante un atacante, quien puede convertirse posteriormente en un intruso dentro de un sistema. Un atacante es considerado como un programa o persona ajena que busca acceder a un sistema de manera ilegal. El atacante puede destruir la información sensible contenida en un sistema, de igual forma éste puede acceder a un sistema para probar sus conocimientos sin realizar daño alguno, y en otros casos busca penetrar a un sistema con la finalidad de obtener un beneficio persona [124].

Existen dos tipos de atacantes: los internos y los externos.

Internos (insiders)

Este grupo está integrado por personas que trabajan dentro de una organización, quienes efectúan sus ataques de manera voluntaria o por equivocación; con o sin

conocimientos básicos sobre sistemas computacionales. Los ataques que pueden realizar, se lista a continuación:

- *Ataques por medio de herramientas encontradas en la web*
  Hacen uso de programas que son creados por personas con altos conocimientos en computación, de los que el atacante interno desconoce su funcionamiento interno, pero no su aplicación. Se utilizan con dos objetivos fundamentales: el primero es experimentar las bondades del programa en cuestión y el segundo, para resolver su situación actual o anterior dentro de una organización, por ejemplo: venganza o modificación de información.

- *Ataques de Ingeniería Social*
  La ingeniería social es una técnica que se basa en la vulnerabilidad que tiene el ser humano ante un factor sorpresa y la compasión que se puede despertar en él, y con ello, el atacante pueda obtener un beneficio personal. Esta técnica también es empleada para ganar la confianza de una persona que tiene privilegios dentro de una organización, y de esta manera, conseguir información sensible o privilegios dentro de la misma. Una de las variantes de esta técnica es hacerse pasar por una persona de alto rango dentro de una organización para obtener beneficios personales o mayores privilegios de los que tiene en dicha organización.

- *Ataques por omisión*
  Los usuarios de un sistema generalmente por practicidad, omiten leer mensajes emergentes que aparecen en el sistema que operan, ocasionando en diversas situaciones la activación de herramientas creadas y enviadas por atacantes externos; ejecutando acciones que comprometen los sistemas que operan e incluso los pueden dejar no disponibles para su uso.

- *Ataques de fuga de información*
  La excusa de necesitar adelantar y llevarse trabajo de una organización a casa, provoca una gran vulnerabilidad a la información sensible de una organización, puesto que ésta puede ser extraviada durante el trayecto, ocasionando con ello, un gran riesgo en su confidencialidad. Otra forma de realizar la fuga de información es dejarla por descuido a la vista de cualquier persona ajena a ella, ya sea por medios electrónicos o impresos.

- *Ataques de suplantación y autentificación*
  Este ataque es muy común dentro de las organizaciones. A los usuarios de un sistema se les otorgan claves de acceso con diferentes privilegios en base a sus necesidades de operación dentro del sistema. Sin embargo, por motivos de incapacidades laborales, despidos, promociones de puestos e incluso vacaciones. Las claves son prestadas a terceros para cubrir las ausencias de los usuarios o se realiza la omisión de la notificación del cambio de un usuario que ya no labora en la organización (en caso de un despido o promoción de puesto), ésto provoca problemas de suplantación de identidad, así como la

vulnerabilidad en el control de los privilegios otorgados al usuario previo, comprometiendo con ello, la información sensible en una organización.

- *Ataques de vulnerabilidad de contraseñas*
  Este ataque es provocado por el uso de contraseñas débiles (sencillas o fáciles de adivinar) que le dan acceso a un sistema, en lugar de contraseñas fuertes (complicadas). Este ataque es el resultado del desconocimiento de la seguridad informática que debe existir dentro de un sistema (cultura informática).

- *Ataques de cadena*
  La ingeniería social ha permitido emplear el factor sorpresa, la compasión y la superstición para el beneficio de terceros. Las cadenas generalmente son enviadas por medios electrónicos (correo) en las que su contenido les solicita el reenvío hacia conocidos con la promesa de obtener bendiciones, dinero gratis o la gratificación por ayudar a una determinada causa. Este tipo de ataque emplea como arma la buena fe de un ser humano para generar tráfico en la red y/o apoderarse de nuevas direcciones de buzones de correo, las cuales se convertirán en sus futuras víctimas. Algunas variantes en este tipo de ataques es la activación oculta de código malicioso que puede ejecutarse al momento que se realiza la lectura del texto.

Las motivaciones de los atacantes externos generalmente son**:** La venganza, el control sobre un sistema, la manipulación para obtención de mayores privilegios y/o el dinero que puedan ofrecer por la información sensible de una organización.

Externos (outsiders)

Los atacantes externos pueden ser personas con amplio conocimiento en lenguajes de programación y gran expertise en el uso y diseño de protocolos o simples aficionados que buscan experimentar y vanagloriarse, sin conocer en esencia que es lo que están llevando a cabo. Este tipo de atacante puede crear programas con código malicioso que comprometa a un sistema.

A diferencia de los atacantes internos, los atacantes externos requieren para llevar a cabo un ataque, el empleo de su ingenio y de los conocimientos adquiridos; los cuales les permitan recabar la mayor información posible sobre el objetivo que desean vulnerar. Para ello, hacen uso de las técnicas de ingeniería social y/o el reconocimiento de puertos que identifiquen a los sistemas y a los mecanismos de seguridad que se encuentren en una organización. Esto es con la intención de hallar vulnerabilidades que les permitan irrumpir en un sistema. Los atacantes externos generalmente son conocidos con el nombre de hackers.

Sus principales motivaciones son el desafío, la sed sobre un conocimiento profundo sobre algún tema en específico (comunicaciones, virus, sistemas operativos, hardware,...), las venganzas, el factor económico y en algunos casos vanagloriarse de

la hazaña perpetrada.

Los hackers son personas que les gusta aprender y emplear el conocimiento adquirido, algunas veces como desafío y otras por compartir información bajo la filosofía de que la información debe ser libre. Los hackers han creado su propio mundo, conocido como underground, en el que se enfatiza que un verdadero hacker no destruye información sólo aprende a través de la práctica pero sin dañar un sistema. Para comunicarse entre ellos han creado su propio lenguaje, el cual se diferencia de un lenguaje coloquial por el uso de palabras que hacer referencia a términos técnicos para la recepción y aceptación de mensajes, transferencia de archivos, entre otros. El convertirse en un hacker no es una tarea fácil, se debe desarrollar un perfil para ser autodidacta, paciente y perseverante sobre un tema específico. Adicional a esto, se requiere del conocimiento de varios lenguajes de programación, entre los que destacan el lenguaje ensamblador y el lenguaje c, por la versatilidad que les da para la creación de código potencial y portable (tamaño pequeño con grandes oportunidades de triunfar en su objetivo). Asimismo, de la profundidad en el conocimiento y operación de sistemas operativos (procesos del sistema, tipos de núcleo, uso de pila, ...), protocolos de comunicación, protocolos de ruteo, redes, todo lo que les pueda ayudar a desarrollar un conocimiento profundo sobre un tema específico.

Los hackers se dividen en 3 grupos:

- *Sombrero Blanco*
  Son hackers que emplean sus conocimientos para corregir vulnerabilidades, asesorar sobre la seguridad de un sistema, corregir errores de código de programación, entre otros. Sus conocimientos los obtuvieron irrumpiendo sistemas en donde rebasaron los límites de la privacidad, es decir, alguna vez pertenecieron al lado oscuro, pero ahora son los buenos que sólo quieren ayudar. Ellos sostiene la ideología, que primero se tiene que aprender a ser malo para conocer y después con madurez propia llegan a ser verdaderos hackers.

- *Sombrero Gris*
  Éstos propiamente no son hackers bajo el sentido ético de los sobrero blanco, se puede decir que están en un punto intermedio de su aprendizaje, son malos por que están en proceso de aprendizaje, pero también son buenos porque tratan de subsanar los errores de programación que encuentran en los sistemas que irrumpen.

- *Sombrero Negro*
  Ellos son denominados por los mismos hackers como crakers, son personas que no tienen escrúpulos o nunca maduraron para entender que sólo eran los primeros pasos para llegar a ser un hacker, sus principales móviles son: el dinero, el resentimiento, la satisfacción personal.

En el underground existen *clasificaciones* designadas por ellos mismos, que identifican el nivel de conocimiento que se va adquiriendo sobre un determinado tema. Dentro de esta clasificación se encuentran personas con ciertas habilidades, pero que no desarrollan ningún conocimiento sobre un tema específico, sin embargo, se desenvuelven directa o indirectamente con ellos (Anexo C).

Los atacantes internos como los atacantes externos son altamente peligrosos, no se debe subestimar su potencial, a pesar de que algunos de ellos en apariencia o confirmado, no posean un conocimiento profundo en el área computacional.

Podemos decir, que en ocasiones es más peligroso una persona que no tiene idea de lo que está haciendo, que él que si lo tiene; puesto que él primero no presenta un patrón conocido y actuará por instintos ante la situación que se le presente. En cambio, el segundo desarrolla una metodología que lo hace predecible para lograr su objetivo.

3.5 Metodología para efectuar una intrusión a un sistema

Un intruso sigue una metodología para acceder a un sistema [124], algunos de los pasos que se enumeran pueden estar conjuntados en un sólo paso, por ejemplo, el análisis y la planeación. El hecho es que no importa si se realizan en forma aislada o conjunta, éstos serán los pasos a seguir para vulnerar un sistema.

- Fijar el objetivo (Víctima).
- Recopilar la mayor información sobre el objetivo que se desea vulnerar (a este paso también se le conoce como reconocimiento del objetivo).
- Análisis de la información recabada.
- Planeación del ataque (herramientas a emplear, estrategia a seguir).
- Obtención de los máximos privilegios dentro del sistema.
- Inspeccionar el sistema que se está accediendo (en ocasiones el intruso puede dejar una puerta trasera clandestina para realizar futuros accesos).
- Eliminar las evidencias que lo delaten (borrado de huellas).

3.6 Tipos de Ataques

Los ataques pueden ser de dos tipos: *pasivos y activos*.

Ataques pasivos

- *Escaneo de puertos*
  Es el empleo de diversas técnicas que le permiten a un atacante reconocer los mecanismos de seguridad que tiene un sistema.

- *Captura de información sensible.*
  Se refiere a la forma clandestina de observar y retener la información que contiene un sistema, tratando de que el propietario no se percate de ello.

Ataques activos

- *Control Remoto sobre un sistema.*
  El atacante instala código malicioso que le permita tener control total de un sistema de forma remota.

- *Explotación de vulnerabilidades.*
  Mediante el escaneo de puertos (también conocido como barrido de puertos) se busca encontrar huecos en los mecanismos de seguridad, que le permitan al atacante vulnerar un sistema. Asimismo, busca errores en el diseño y arquitectura de un sistema.

- *Suplantación de identidad.*
  Es usurpar la identidad de otro para acceder a un sistema.

- *Intercepción de paquetes de red.*
  Ésta puede manifestarse en las siguientes formas: suplantación y captura. Es colocarse en medio de la comunicación entre dos sistemas sin alterar su flujo o bien, estar en medio como un puente que entrelace a los dos sistemas sin que el emisor se percate de ello, puesto que le hará creer a éste, que el atacante es el receptor original de la sesión.

- *Denegación o interrupción de servicios brindados por un sistema*
  Es la técnica que se emplea para dos fines particulares: El primero, para borrar las huellas que delaten su presencia y el segundo, para interrumpir los servicios que un sistema brinda a una organización o a sistemas remotos que dependen de él.

- *Evasión de los mecanismos de seguridad.*
  Consiste en crear información que asemeje ser idéntica a la que puede admitir un sistema, y con ello, evadir los mecanismos de seguridad e infiltrarse a un sistema sin ser detectado.

Los *ataques pasivos* se basan en el reconocimiento (exploración) y captura de información sensible sin dañar un sistema.

Los *ataques activos* realizan acciones que dañan, engañan, controlan y deniegan los servicios de un sistema. De igual forma, éstos buscan la evasión de los mecanismos de seguridad que existen en un sistema.

Los *servicios de seguridad* brindan beneficios de protección a un sistema (confidencialidad, autenticación, control de acceso, no repudio, disponibilidad e integridad). Los ataques intentan deshabilitarlos o comprometerlos, de la siguiente forma:

- La confidencialidad se ve afectada por el reconocimiento y captura de información de un sistema.
- El no repudio y el control de acceso hacia un sistema, se ven comprometidos generalmente, por ataques de control remoto
- La confidencialidad, la integridad, la autenticación y el no repudio de un sistema se ven afectados por ataques de suplantación e intercepción.
- La disponibilidad y el no repudio son puestos en riesgo ante los ataques de denegación o interrupción de un sistema.
- La explotación de vulnerabilidades atentan contra cualquiera de los servicios de seguridad (confidencialidad, autenticación, control de acceso, no repudio, disponibilidad e integridad) que se encuentren presentes en un sistema

Las herramientas que generalmente son más usadas acorde con el tipo de ataque a realizar, son:

Reconocimiento del Objetivo

- Escaneo o barrido de puertos (paquetes malformados).

Captura de información

- Sniffers
- Malware (Spyware, loggers, rootkits)

Control remoto de un equipo

- Bots
- Puertas traseras
- Rootkits
- Back Orifice

Explotación de vulnerabilidades

- Exploits de protocolos de comunicación
- Exploits que aprovechen una vulnerabilidad en el diseño y arquitectura de un sistema.
- Buffer overflow

Suplantación

- Spoofing en sus diferentes versiones: IP, MAC, DNS, ARP, WEB, Mail.
- Hijacking.

Denegación

- Buffer overflow
- Flooding

Evasión

- Rootkits
- fragmentación de paquetes
- fabricación de paquetes malformados.

Por lo general, un atacante no emplea una sola herramienta, ni una sola técnica, sino que éste requiere usar varias de ellas a la vez para llevar a cabo la penetración de un sistema. Unas servirán para obtener información del objetivo y vulnerar el sistema, y otras como distracciones que le permitirán ocultar su presencia el tiempo necesario para realizar el verdadero ataque que se propuso efectuar desde un inicio, por ejemplo, acceder a otros sistemas desde la red vulnerada o robar información sensible de un sistema.

Entre los ataques más representativos que atentan contra la seguridad, lse pueden observar en la Tabla 3.2:

Tabla 3.2: Ataques y consecuencias hacia un sistema

Tipo de Ataque	Consecuencia
Robo de Sesión (Hombre en medio).	Suplantación y robo de información en forma pasiva.
Enmascaramiento (Spoofing).	Suplantación, evasión.
Reconocimiento.	Detección de vulnerabilidades fingerprint, ICMP, IP, UDP.(Puertos, S.O.)
Saturación de servicios (Inundación del canal).	Denegación de Servicios.
Creación de código Malicioso (Virus, bots, spyware, rootkits, bots)	Control remoto, modificación y Denegación de Servicios.
Sniffing (Monitoreo y captura de información en forma pasiva).	Captura de información sensible.
Looping (Software de control remoto para generar ataques anónimos hacia otras redes).	Ataques remotos hacia otras redes enmascarando al verdadero atacante

3.7 Metodologías

De acuerdo a el tipo de ataque que se desee efectuar, el atacante pude realizar durante su inspección la modificación y captura de información sensible, efectuar ataques distribuidos hacia otras redes por medio de la red vulnerada o solamente indagar en el sistema sin realizar daño alguno. La mayoría de los ataques emplean las siguientes metodologías [124]:

- *Ingeniería Social*

  La Ingeniería Social, es la herramienta más poderosa con la que cuenta un atacante, con ésta, él es capaz de falsear su identidad para obtener la mayor información posible de su objetivo. En el caso de ser un atacante externo, éste se hace pasar por una persona que forma parte de la misma organización para obtener información que le lleve a su objetivo, buscando obtener la confianza de la víctima al mostrar interés y conocimiento sobre el tema para no despertar sospecha alguna. Si por el contrario es un atacante interno, busca establecer amistad y confianza con las personas que tienen control de un sistema, y de esta manera, obtener los privilegios necesarios para irrumpir en un sistema.

  Ahora bien, si se tratase de ejecutar alguna herramienta creada por el mismo atacante, éste recurrirá a el engaño a través de la solicitud de presionar algún enlace o el reenvío de información compasiva o de carácter religioso vía correo electrónico, a cambio de alguna recompensa o diversión que sea del

agrado de la víctima.

Los IDS son totalmente vulnerables ante la ingeniería social, puesto que esta depende de un criterio humano y no de un patrón que pueda modelarse para su identificación.

- *Vulnerabilidades del protocolo TCP/IP*

  Los RFCs de la Suite de Protocolos TCP/IP explican su operación y los componentes que constituyen a un paquete de red (cabeceras, banderas, contenido). Sin embargo, no existe una restricción que impida efectuar cualquier operación sobre un paquete. Esto se debe a que inicialmente los protocolos de comunicación se diseñaron sin tener en mente el concepto de seguridad, sólo se propusieron como las normas establecidas para lograr la comunicación entre dos o más dispositivos de red.

  Debido a esta situación los atacantes emplean una lógica abierta, la cual consiste en efectuar cualquier operación que no esté estrictamente prohibida dentro de los RFCs, para crear paquetes malformados que les permitan explotar sus debilidades de seguridad. Un paquete malformado es una técnica que consiste en crear un paquete de cualquier tipo de protocolo de comunicación, con parámetros diferentes a los que se han establecido en su RFC. Esto es permitido, debido a que los protocolos de comunicación se establecieron como las normas de operación entre dos o más dispositivos de comunicación. Sin embargo, no existe restricción alguna sobre la modificación de los parámetros iniciales, teniendo como resultado una malformación del paquete original.

- *Vulnerabilidades en el diseño y arquitectura de los sistemas*

  Son ataques que crean programas que explotan las vulnerabilidades encontradas en los sistemas operativos, en las aplicaciones desarrolladas con software comercial o de código libre, en los errores de código de programas de software tipo comercial, entre otros. El objetivo es utilizar estas debilidades para penetrar en un sistema, ya sea para instalar alguna herramienta de control remoto que le permita realizar ataques hacia otras redes o realizar futuros accesos de manera clandestina a la red vulnerada(Fig. 3.1).

  Muchos de los nuevos ataques surgen de pruebas que se realizan sobre los protocolos de comunicación para evaluar su fragilidad.

  Por ejemplo, el *ping flooding (inundación de paquete ping)* consiste en enviar una lluvia de paquetes de gran tamaño hacia un host destino (víctima), de tal forma que este no pueda procesarlos. Para este tipo de ataque, ser requiere de un ancho de banda mayor que el de su objetivo. Otro ejemplo es *el ping de la muerte,* paquete malformado con un tamaño mayor a los 65,535 bytes, el cual es enviando de manera rápida y continua sobre la víctima. De acuerdo

a las especificaciones es ilegal ese ese tamaño, pero es posible enviarlo mediante la fragmentación del paquete. El problema surge al momento de reensamblarse el paquete en la máquina destino, al ser su tamaño tan grande, la pila que lo contiene sufre un desbordamiento de pila o buffer overflow.; lo que ocasiona que el sistema quede fuera de línea.

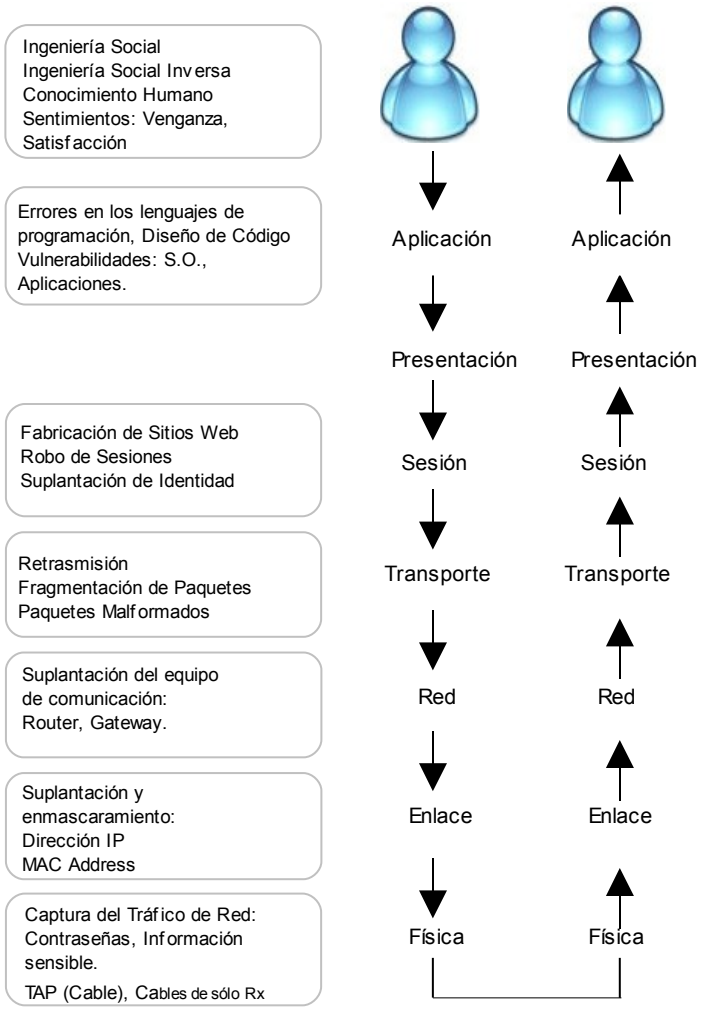


Fig. 3.1: Ataques que se pueden desarrollar en el modelo OSI

Como se puede observar en estos ejemplos, estas vulnerabilidades surgen de la falta de integración de los conceptos de seguridad, por lo que los IDS buscan patrones coincidentes con dichas vulnerabilidades, para su detección oportuna, y se evite que un sistema no esté disponible. Cabe añadir, que algunas de estas vulnerabilidades ya han sido corregidas.

3.8 Descripción de ataques

A continuación se listan algunos de los ataques más conocidos que pueden ser perpetrados por un atacante para penetrar un sistema [115, 124]. Éstos pueden ser empleados de manera conjunta o aislada para comprometer la seguridad de un sistema:

Exploits (Explot it)

Son las fallas y errores en el código (bugs) de un programa o en los sistemas operativos, conocidos en el ámbito computacional como vulnerabilidades a nivel del diseño de la aplicación. Dichas oportunidades son aprovechadas por los coders (atacantes especialistas en programación) para adentrarse a los sistemas, mediante la elaboración rápida de programas que permitan explotarlas (de ahí su nombre actual) y adentrarse a los sistemas, comprometiendo la integridad de la información y/o la inhabilitación de sus servicios.

Día cero

Al ser detectado un bug (error de código) en un sistema, este es rápidamente difundido a través de Internet como una vulnerabilidad. La cual es aprovechada por los atacantes durante el lapso de tiempo en el que las compañías comerciales de los sistemas operativos o antivirus realizan la reparación (parches) o erradicación (virus) a dicha vulnerabilidad. Su nombre surge del factor sorpresa y del tiempo en que éste permanece en los sistemas antes de ser corregido.

KeyLogger

Es un programa que se creo para registrar todas las acciones que se realicen sobre un teclado. Estos registros se almacenan en un archivo para posteriormente ser retrasmitidos hacia el atacante, con la finalidad de utilizar la información sensible capturada en beneficio propio. Generalmente se aplica en el sector financiero o en compras electrónicas.

Pharming & Phising

Es la suplantación que se hace de sitios web, en donde el atacante redirecciona la solicitud de una página conocida hacia la que él mismo construyó, la cual es idéntica a la comercial. Esto lo hace con el objetivo de obtener información sensible del usuario. Los sitios más comunes son el área financiera, compras electrónicas, servicios de comunicación, correo. Aparte de la construcción del sitio, éstos requieren de un punto adicional, atacar a los servidores DNS que contienen las direcciones reales, para cambiarlas y redireccionarlas hacia puntos intermedios escalables que pueda accesar remotamente el atacante.

Spam

Es la generación de correos aletorios con temáticas sobre medicamentos, deportes, temas sexuales, entre otros; enviados a un sólo equipo o varios equipos a la vez (víctimas), provocando la saturación del buzón del correo y la generación de tráfico en la red. Existen diversas variantes de este ataque, una de ellas puede ser

efectuada por nombres de usuarios de correo que a simple vista parecen auténticos (provenientes de un sitio conocido) o pueden ser generados por programas que generan nombres aleatorios en orden alfabético, para posteriormente enviarles información considerada como basura. Para efectuar dicho ataque, se requiere previamente conocer el nombre de dominio del correo, así como de adueñarse de la libreta de direcciones para multiplicar y replicar el ataque hacia diferentes redes y dominios.

Spyware

Software que se instala clandestinamente en un sistema, que se dedica a la captura de las acciones realizados sobre el teclado, ratón o información confidencial, es decir, monitorea y captura las acciones del sistema (procesos) y la de los usuarios. Una vez realizado el cometido, retransmite la información hacia el atacante por medio de redes intermedias que oculten su identidad y su ubicación. Muchas veces este tipo de información es generada por empresas de publicidad para conocer las preferencias del mercado.

Smurf

Es un programa que genera un paquete mal formado, en el que la dirección origen es la dirección ip de la máquina víctima y el destino es la dirección de broadcast, de tal forma que todos los equipos respondan hacia la víctima, generando tráfico en la red.

Adware (Advertisement + Software)

Son programas publicitarios que aparecen por medio de pantallas emergentes, las cuales son adicionadas al instalar o bajar un software a través de internet, que son bastante molestos y consumen recursos de la memoria, incluso pueden genera cambiosen nuestro escritorio en la página principal del navegador.

Bomba Lógica

Programa que contiene en su interior código malicioso, el cual se detonará en la fecha señalada (sólo espera que se cumpla para entrar en acción).

Troyano (Caballo de Troya)

Inspirado en el Caballo de Troya de la obra la Iliada, de Homero. Es un software que oculta sus verdaderas intenciones, da la apariencia de ser inofensivo cuando en la realidad no lo es. Pueden ocultarse en una foto, en un correo, en un juego y/o en páginas web. Sus intenciones son variadas pueden ir desde modificar el ambiente de trabajo, instalar accesos a páginas web de dudosa procedencia, modificar archivos, inutilizar el equipo o emplearlo para realizar otros ataques.

Back Doors (Puertas Traseras)

Aquí el atacante tendrá acceso de manera oculta a un equipo víctima a través de la instalación clandestina de un software, el cual generará una puerta escondida (trasera), la cual sólo el atacante conoce permitiendo con ello, manipular el equipo desde el exterior.

Gusano

Es un progama que se autoduplica y autopropaga a gran velocidad, se le conoce con este nombre por la forma de arrastrarse a través de las conexiones que conforman una red, y la manera de consumir la memoria de un equipo (colapsar el equipo por falta de memoria). Se diferencia de un virus debido a que un virus es diseñado para un host y el gusano opera bajo un esquema de red.

Bot (Contracción de la palabra Robot)

Programa instalado clandestinamente en un ordenador que lo convierte en un zombie, es decir, el atacante tiene pleno dominio sobre él.

Botnet

Es un conjunto de máquinas que sirven a un amo (atacante), quien utiliza los ordenadores comprometidos para realizar ataques hacia otros equipos y/o redes diferentes a la actual.

BackOrifice

Es un programa cuyo objetivo es obtener el control remoto sobre un sistema

Hijacking

Es el robo que se produce de una sesión, de una página y/o de un dominio; generalmente se emplea para capturar información sensible (transacciones bancarias, etc.) que el atacante utilizará en su beneficio.

Hoax

Es una broma, es la prueba fehaciente de la ingeniería social. Contiene información falsa la cual genera alarma y desconcierto, la cual puede presentarse en apariencia de ser un virus de alta peligrosidad o de una simple cadena de correo. Es difícil detectarlos por un usuario de computadora, debido a que sus errores en redacción, fechas y fuentes son poco evidentes, consiguiendo su objetivo de alarmar a un usuario, en el que el atacante puede solicitar su reenvío hacia otros usuarios con la finalidad de alarmarlos (ingenieria social) y con ello, saturar el canal de comunicación. En algunos casos se indica la presencia de un archivo etiquetado como maligno, el cual se encuentra alojado en el sistema operativo dando como referencia que la fuente donde se obtuvo dicha información es del propietario comercial de la marca, sin que en realidad sea avalado por dicho propietario. Se emplea de manera general para saturar el correo de forma manual a través de el reenvío de información, degradando el canal y el servicio de mensajería.

Downloading

Capturar y bajar información a un equipo ajeno al que reside el sistema.

Snooping

A diferencia del Downloading, sólo captura información (monitorea) de un sistema.

Tammpering

Es el acto de modificar la información almacenada en un sistema o el software previamente instalado en el mismo.

Jamming/Flooding

Esta técnica se basa en aprovechar las debilidades que tienen los protocolos de comunicación, los errores de diseño y la arquitectura de los sistemas; con el objetivo de saturar o inundar los servicios que brinda un sistema. La inundación se puede producir de varias formas, una de ellas es generando una cantidad excesiva de paquetes de solicitud de conexión hacia un objetivo, dejándolo no disponible para otras conexiones. Otra forma es llevar al tope los recursos de un sistema, a través de la generación simultánea de múltiples procesos que desborden el tamaño de la pila designada para los procesos residentes.

Fingerprinting

Es una técnica empleada para reconocer el tipo de sistema operativo, con el que opera un sistema. Dentro de los datos que se pueden obtener se encuentran la versión del sistema operativo y los servicios que éste ofrece. Se basa en el tipo de respuesta que obtiene del envío de un paquete mal formado, el cual es analizado posteriormente por el atacante y de acuerdo a sus características se determinará el tipo de sistema operativo empleado.

Looping

Esta es una técnica en la que se utiliza a un sistema para ingresar recursivamente a otros sistemas de manera clandestina. Su objetivo es lanzar ataques distribuidos simultáneamente sin que el atacante sea identificado y culpen de su hazaña a las redes involucradas.

Crackers

Son programas creados para romper los candados colocados para el uso limitado de un software comercial, también se considera un cracker al programa que busca descifrar las claves de acceso a un sistema. Se basa en técnicas de búsqueda por diccionario (letras y palabras comunes en un lenguaje determinado) y búsqueda por fuerza bruta (probar con números, letras y caracteres especiales que no formen una palabra).

Sniffing/Eavesdrooping

Es la captura o intercepción de información que fluye en un sistema, ésta puede ser pasiva (monitoreo) o activa (intercepción).

Evasión

Consiste en la fabricación de paquetes malformados que burlen la seguridad de un sistema. Estos paquetes pueden modificar su tamaño, dividir un ataque en fragmentos, alterar su dirección origen, entre otros.

Zombie

Es un ordenador que se encuentra comprometido por un atacante y éste tiene pleno dominio sobre él. Su objetivo es atacar otros sistemas que compartan o no la misma red. El origen de su nombre viene de la técnica vudú que se emplea sobre un ser humano para que éste pierda su voluntad y haga lo que un tercero le indique. Éste puede deshabilitar la función de un antivirus.

Rootkit

Es uno de los ataques más peligrosos que existen. Es un programa que mantiene ocultos los procesos que son generados por un intruso durante la instalación y ejecución de malware en un sistema. Debido a su característica de ocultarse, es difícil detectar este tipo de ataque en una simple revisión, requiere de un análisis en profundidad y de la integridad de los sistemas para detectar su presencia.

DoS

Son ataques que deniegan los servicios de un sistema. Buscan degradar los canales de comunicación e impedir comunicarse con el sistema, y en algunos casos el corromper el tamaño de la pila a través de alguna vulnerabilidad que colapse al sistema.

DDoS

Se basan en el mismo principio que los DoS, pero difieren en que éstos operan para ambientes distribuidos.

Spoofing

Es la suplantación de identidad ante un sistema, puede ser efectuada en diferentes formas:

- IP Spoofing
  Suplanta la dirección IP, hace creer al sistema que un paquete proviene de una determinada IP para que éste la acepte y otorgue los beneficios correspondientes.

- ARP Spoofing
  Suplanta la tabla ARP de un dispositivo u ordenador. Modifica la tabla ARP en su beneficio para cambiar la ruta de la información a un sitio previamente comprometido por un atacante.

- MAC Spoofing
  Suplanta la dirección MAC de un dispositivo u ordenador. Modifica la dirección MAC para no ser detectado en un ataque hacia un sistema.

- DNS Spoofing
  Suplanta un nombre de dominio. Le hace creer al sistema que la dirección IP o el nombre de dominio ha cambiado y que requiere actualizar la información correspondiente. La direcciones IP o nombres de dominio nuevos son

conocidos únicamente por el atacante, los cuales utilizará para capturar información sensible que le beneficie de manera personal. Es una técnica sumamente peligrosa.

- Web Spoofing
  Suplanta un sitio de web. Se construye un sitio web idéntico al de una determinada organización en un servidor clandestino, su dirección es modificada mediante DNS Spoofing. La creación de este sitio es con la finalidad de obtener información sensible de tipo financiera o de índole personal de un usuario, para posteriormente emplearla en su beneficio.

- Mail Spoofing
  Suplanta una dirección de correo. Esta técnica es sumamente peligrosa tal como lo es la de DNS Spoofing. La suplantación se basa en la ingeniería social, donde se hace llegar a los buzones de las víctimas mensajes que en apariencia son enviados por instituciones financieras, prestadores de servicios, administradores de los correos populares de la web, entre otros. Su objetivo es solicitar ayuda para recabar información sensible que el atacante ocupará para su beneficio personal, como pueden ser número de tarjetas de crédito, nombre completo, dirección de casa, etc.

Hombre en medio / Intercepción
Emplea cualquier técnica de spoofing para engañar a un tercero sobre su identidad y capturar su información sensible.

Ingeniería Social
Es una técnica que permite obtener información sensible a través de la confianza e ingenuidad de la víctima.

Ingeniería Social Inversa
Un atacante se hace pasar por un funcionario de una organización para obtener información sensible o privilegios dentro de un sistema, sin que la víctima pueda verificar su identidad.

Reconocimiento de puertos (Barrido de puertos)
Revisan mediante la fabricación de paquete malformados si un puerto está activo o inactivo en un sistema.

Malware
Son programas creados con un fin específico, pueden ser: virus, gusanos, rootkits, puertas traseras, captura de información, instalación de software que difiere del permitido en un sistema, entre otros. El malware puede tener la capacidad de replicarse (propagarse a través de la red a otros equipos), ocultarse, evadir los antivirus a través de su deshabilitación o variación de paquetes que apariencia son fidedignos, activarse en horarios no habituales, polimórficos, permanecer residentes en memoria, entre otros.

3.9 Indicios de una intrusión

Algunos de los indicadores más comunes que pueden evidenciar que un equipo está comprometido, son:

- Uso excesivo de los recursos de un sistema (memoria, procesador, pila,...)
- Modificación de la integridad de la información almacenada en un sistema.
- Accesos a los sistemas en horarios diferentes a los habituales.
- Intentos excesivos para descifrar las claves de acceso hacia un sistema
- Reconocimiento de puertos de forma remota hacia un sistema (barrido de puertos)
- Aparición de pantallas emergentes que no pertenecen al sistema.

3.10 Herramientas de Ataque

Las herramientas de ataques son utilerías que un atacante usa para reconocer y vulnerar un sistema. Estas herramientas generalmente, son programadas en diferentes lenguajes con el objetivo de explotar las vulnerabilidades que pueda tener un sistema.

Éstas se clasifican en base a el propósito para las que fueron creadas:

- Reconocimiento de puertos (Port scanning)
- Reconocimiento del sistema operativo (Fingerprinting)
- Decifradores de contraseñas (Crackers)
- Escáners de vulnerabilidad.
- Captura de información (Sniffers)
- Fabricación de paquetes,
- Reconocimientos de rutas.
- Desensambladores de código ejecutable.

3.11 Herramientas de Seguridad

Las herramientas de seguridad se crean con la intención de brindar protección a un sistema [116]. Estas pueden clasificarse en:

- Cortafuegos: Delimitan la red interna y la red externa de una organización.
- Sniffers: Detectan cuellos de botella (degradación del ancho de banda de una red) o problemas con una aplicación específica.

- Algoritmos de encriptación: Hacen que la información sea incomprensible para terceros.
- Análisis forense: Aportan conocimiento sobre ataques perpetrados.
- HoneyPots: Estudian las técnicas que un intruso realiza dentro de un sistema.
- IDS: Detectan intrusos en un sistema
- Antisniffers: Detectan la presencia de sniffers en una red.
- Reconocimiento de puertos: Revisan los puertos activos e inactivos de un sistema.
- Escáners de vulnerabilidad: Detectan las vulnerabilidades que pueden poner en riesgo a un sistema.
- Servidores Proxy: Son los intermediarios de las peticiones efectuadas desde la red externa, hacia los servidores de la red interna.
- Decifradores de contraseñas: Revisan la dificultad de adivinar las contraseñas de acceso hacia un sistema.
- Antivirus : Realizan la detección de virus en un sistema.
- Anti-malware: Detectan códigos maliciosos en un sistema.

3.12 Snort

Es un sistema de detección de intrusos de código libre basado en red, que detecta intrusiones en tiempo real; puede ser instalado en plataformas Windows y Linux.

Su técnica de análisis es a través del reconocimiento de firmas, que son expresadas por medio de reglas. Estas reglas buscan la coincidencia señalada por la firma, dentro de las partes que integran a un paquete de red. Su motor de análisis requiere mantener actualizadas sus reglas para brindar una mejor protección ante nuevos ataques o variaciones de éstos.

Puede operar en cualquiera de los siguientes modos:

- *Modo Sniffer*
  Habilita el modo promiscuo de la tarjeta de red para iniciar la captura de los paquetes que circulan en una red.

- *Modo de Registro de paquetes (Packet logger)*
  Registra los paquetes que circulan en una red, dentro de un archivo especificado previamente por el administrador del sistema.

- *Modo de sistema de detección de intrusos*
  Tiene la misma funcionalidad del modo sniffer con la adición de un motor de

análisis que le permite detectar en tiempo real la intrusión hacia un sistema.

- *Modo In-line (Flex Resp)*
  Es un IDS que interactúa conjuntamente con IPTables (cortafuegos de linux), para realizar respuestas activas ante una acción hostil proveniente de un atacante. Este modo le da a Snort la característica de operar como un IPS, por lo que puede resetear conexiones, descartar paquetes, etc (funcionalidades que son características propias de un cortafuegos).

3.13 Snort en la investigación

Snort es una herramienta muy útil para el área de investigación informática, a través de él es posible desarrollar modelos que permitan mejorar su efectividad de detección. Su principal ventaja es ser un software de código abierto, lo cual permite adaptarlo a las necesidades propias de un entorno de red a través de la creación de nuevas reglas y preprocesadores sin afectar su funcionamiento original.

En el área de investigación se han realizado diferentes trabajos que proponen una mejora de su motor de análisis ante ataques desconocidos y ya conocidos. Estos trabajos han permitido incursionar en las redes neuronales, proponer mejoras en la clasificación de las reglas y en la detección de sniffers dentro de una red. También el uso de snort ha permitido representar escenarios que describan el proceso de un intrusión y el desarrollo de preprocesadores específicos para un tipo de ataque.

Algunos de los desarrollos hechos en Snort, son:

- La universidad del Cauca, Colombia. Diseñó Portscan AI que fue desarrollado por Amador, Siler., Arboleda, Andrés y Bedón, Charles. Ellos construyen un preprocesador para Snort basado en redes neuronales para los ataques que emplean la técnicas de barrido de puertos, para conocer los mecanismos con los que cuenta un sistema [109].

- SnifferWall de Basic Software Laboratory, CERIST. Fue desarollado por H. AbdelallahElhadj, H. M. Khelalfa & H. M. Kortebi. Es el desarrollo de un preprocesador que detecta un sniffer en la red mediante las técnicas de: Prueba de latencia, Pruebas ARP y DNS [97].

- Rong-Tai Liu/Chih-Hao Chen/Chia-Nan Kao de la Universidad Nacional de Tsing-Hua y Nen-Gu Huang de la Corporación BroadWebProponen a FPN como un algoritmo rápido para el reconocimiento de cadenas (patrones), que eficientiza el número de accesos a memoria a diferencia de otros algoritmos ya conocidos para el reconocimiento de patrones. El desarrollo del algoritmo se basa en la funcionalidad de snort para el reconocimiento de patrones y la implementación del algoritmo para mejorar su motor de análisis [22].

- Mike Fisk (Universidad de San Diego California/Laboratorio Nacional de los Alamos) y George Varghese (Universidad de San Diego California). Realizan un análisis sobre la rapidez que se requiere en la búsqueda de reconocimiento de patrones. Proponen un algoritmo de reconocimiento de patrones denominado Setwise Boyer-Moore-Horspool (SBMH) que mejora a los algoritmos empleados por Snort: Aho-Corasick y Boyer-Moore.

3.13.1 Componentes de Snort

Snort es un IDS modular, cada componente tiene una aplicación específica, que en conjunto le permiten realizar la detección en tiempo real de un intruso (Fig. 3.2). Su diseño es flexible y configurable hacia las demandas del tráfico de una red [113], está integrado por los siguientes componentes:

- *Sensor*
  El sensor captura toda información que pasa en la red o un host específico, para que posteriormente sea analizada. Se basa en la librería libpcap (linux) o Winpcap (Windows), que es empleada para poner en modo promiscuo la interface de red de un host. Su función es deshabilitar el filtro que le impide recibir datos que no le correspoden a un host, es decir, mediante el uso de esa librería es posible aceptar el tráfico de toda la red a través de una interface de red. Este puede estar constituido por una o mas interfaces de red.

- *Decodificador*
  Identifica la forma en cómo van enlazados los protocolos en un paquete de red. Los decodifica por niveles para posteriormente ser procesados. Su identificación incluye los datos de la cabecera (número de puerto origen, número de puerto destino, dirección IP destino, ...) y los de su contenido (payload).

- *Preprocesadores*
  Inician el criterio de discriminación de la información que es entregada por el decodificador. Buscan ataques de técnicas de evasión o potencialmente peligrosos, que puedan ser identificados y corroborados por el motor de análisis. En este módulo la información es seleccionada como sospechosa y la envía a un segundo filtro (motor de análisis) que será el responsable de determinar si se trata de un paquete de red con intenciones ocultas o una falsa alarma. Cada preprocesador fue creado para una función específica. Las características de detección que ofrecen éstos, pueden ser habilitados o deshabilitados en base a los criterios personales del propietario de un sistema.

- *Motor de Análisis*
  El motor de análisis de Snort se basa en reglas, su conocimiento es obtenido por el reconocimiento de firmas de anomalías o abusos que pueden existir en

los paquetes de red. Si la regla coincide con cualquier parte que integre la cabecera o el contenido de un paquete, esta realizará la acción correspondiente (lanzar una alarma, registrar el evento, enviar un mensaje al administrador del sistema,...). Las reglas son almacenadas en un archivo y son procesadas de manera secuencial.

- Plugins de salida
  Son los encargados de mostrar el resultado obtenido por medio de alertas en bitácoras previamente indicadas, las cuales puede ser las bitácoras por defecto de snort, una bitácora diferente o hacia el manejador de bases SQL para una mayor explotación de los datos registrados.

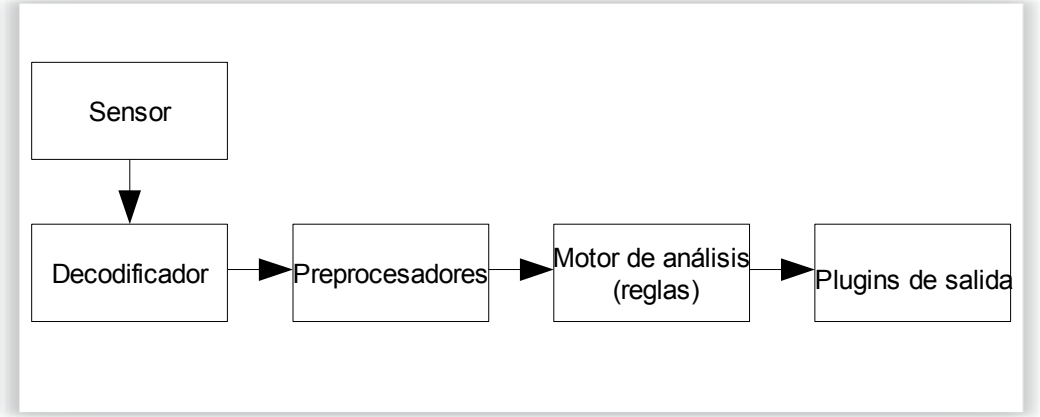


Fig. 3.2: Componentes de Snort

Existen dos tipos de reglas que pueden ser usadas por Snort:

Community Rules
Son reglas proporcionadas por la comunidad Snort (usuarios de Snort). Las aportaciones realizadas son gratuitas, siendo actualizadas de forma semanal.

- *VRT rules (Vulnerability Research Team)*
  Son las reglas que son certificadas pro el VRT, éstas requieren de un pago adicional, puesto que expresan ser actualizadas y probadas día con día, a diferencia de las community rules.

Su motor está constituido por el archivo de reglas y los plug-ins de detección:

- *Plug-ins de detección*
  Son módulos externos que interactuan estrechamente con el motor de análisis, con el objetivo de incrementar su nivel de detección.

- *Plug-ins de salida*

  Son módulos que se crearon para presentar los resultados (notificaciones) que son arrojados por el motor de análisis. Estos resultados pueden ser enviados a bitácoras de uso específico, a aplicaciones de web para facilitar su lectura, a consola, a bases de datos,.... Las notificaciones mostradas por los plug-ins de salida son el resultado de las alertas enviadas por las reglas coincidentes. El nivel de detalle que mostrará una alerta, depende directamente de las opciones habilitadas durante la configuración de Snort. De acuerdo a su configuración, los tipos de alertas que puede presentar Snort son:

  Modo Rápido (Fast Mode), envía:

  - Hora y fecha de la alarma
  - La descripción del mensaje de alerta
  - Dirección IP origen y destino
  - Número de puerto origen y destino

  Modo Completo (Full Mode), envía:

  - Hora y fecha de la alarma
  - La descripción del mensaje de alerta
  - Cabecera completa del paquete (banderas, protocolos, direcciones IP, ...)

  Unix socket Mode, envía:

  - Alertas hacia otro sistema Unix por medio de sockets.

  Ningún modo de Alerta (No Alert Mode)

  - No genera alerta alguna, sólo guarda la información en una bitácora.

  SysLog

  - Envía la alerta correspondiente a Syslog

  .

  SNMP, envía:

  - Mensajes vía SNMP para comunicarse dentro de un ambiente centralizado.

Windows, envía:

- Alertas hacia el Sistema Operativo windows a través de ventanas emergentes (Windows Messenger Service), que emplean el cliente smbclient para su comunicación.

3.13.2 Estructura de las Reglas

Una regla esta formada por dos partes: la cabecera y las opciones de búsqueda.

Cabecera

La cabecera de una regla son las acciones que ésta realizar, al momento de coincidir con las partes internas de un paquete de red que está siendo analizado.

Estructura de la cabecera

La cabecera está conformada por: acciones, protocolos, dirección IP origen, puerto origen, dirección IP destino, puerto destino y la dirección de la operación.

Las acciones pueden ser del tipo:

- *Alert*
  Envía una alerta cuando el contenido de la regla coincide con la cabecera o payload de un paquete

- *log*
  Realiza un registro cuando el contenido de la regla coincide con la cabecera o payload de un paquete

- *pass*
  Si el contenido de la regla coincide con la cabecera o payload de un paquete, el paquete es ignorado

- *activate*
  Genera una alerta y activa una segunda regla, conocida como “regla dinámica”

- *dynamic*
  Es una regla que se mantiene en espera a ser activada por otra regla.

- *Drop*
  Si el contenido de la regla coincide con la cabecera o payload de un paquete, el paquete es descartado.

- *Reject*
  El paquete coincidente con la regla es rechazado a través de IPTables, si es un

paquete de TCP recibe un RST y si es un paquete ICMP recibe el mensaje correspondiente. Todos los paquetes rechazados son registrados.

- *Sdrop*
  Realiza la misma función que Reject, su diferencia radica en que éste no registra los paquetes que se rechazan.

Los protocolos que reconoce, son: IP, ICMP, TCP y UDP.

- La dirección IP origen, puede ser:
  Acepta cualquier dirección del tipo Ipv4.

- El puerto origen, puede ser:
  Acepta cualquier número de puerto de comunicación.

- La dirección IP destino
  Acepta cualquier dirección del tipo Ipv4.

- Puerto Destino
  Acepta cualquier número de puerto de comunicación.

- Dirección de la operación:

  - →
  - ←
  - < >

Opciones de búsqueda

Las opciones son los criterios de búsqueda que se establecen para la coincidencia de un patrón o anomalía, dentro de las cabeceras y el contenido (payload) de un paquete de red. De acuerdo al tipo de opción que se emplee será la precisión de búsqueda que se desee obtener en el uso de la regla.

Algunas de ellas son:

- *msg*
  Envía un mensaje de notificación, cuando la regla coincide con el paquete que se está analizando.

- *Content*
  Revisa el contenido de un paquete (payload), de acuerdo a los siguientes criterios:

- *offset*
  Busca coincidencias a partir de un determinado desplazamiento sobre el área de datos.

- *depth*
  Realiza un análisis en profundidad dentro del paquete.

- *content-list*
  Compara un archivo de texto que contiene las cadenas a comparar con el interior de un paquete.

- *dsize*
  Revisa el tamaño de los datos.

- *Flags*
  Revisa las banderas de : FIN, SYN, RST, PSH, ACK, URG, BIT Reservado 1, BIT Reservado 2.

- *fragbits*
  Revisa los cambios sobre los bits : Bit Reservado, Bit DF (Paquete no fragmentado), Bit MF (Paquete fragmentado).

- *icmp_id*
  Identifica algún aspecto en particular del protocolo ICMP.

- *icmp_seq*
  Identifica el número de secuencia de un paquete con el protocolo ICMP.

- *itype*
  Identifica que tipo de respuesta tiene un paquete de ICMP.

- *ip_proto*
  Identifica un protocolo por su número o su nombre.

- *priority*
  Establece la prioridad que se le asignará a una regla, siendo 1 el valor más alto. Esta opción permite otorgar niveles de importancia a las diferentes alertas que se generan cuando las reglas coinciden con un paquete.

- *ttl*
  Detecta el TTL (Time to Live) de un paquete.

- *tos*
  Detecta un tipo de servicio específico (Type of Service)

A continuación se muestran algunos ejemplos de las reglas que aplica Snort para detectar intrusos:

Ejemplo 1
Regla en la que se indica la búsqueda de un ataque de DoS como Trinoo o alguna variación de éste.

```
# alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"BAD-TRAFFIC data in TCP SYN packet";flow:stateless;dsize:>6;flags:S,12; reference:url,www.cert.org/incident_notes/IN-99-07.html; classtype:misc-activity; sid:526; rev:11;)
```

Ejemplo 2
Regla que detecta una vulnerabilidad conocida del sistema operativo Windows referente a sus servicios de impresión, en la que se emplea un paquete malformado para explotar dicha vulnerabilidad y detener dichos servicios.

```
# alert tcp $EXTERNAL_NET !721:731 -> $HOME_NET 515 (msg:"DOS WIN32 TCP print service denial of service attempt"; flow:to_server,established; dsize:>600; reference:bugtraq,1082; reference:cve,2000-0232; reference:url,www.microsoft.com/technet/security/bulletin/MS00-021.mspx; classtype:attempted-dos; sid:3442; rev:3;)
```

Capítulo 4
Propuesta

Resumen

Los sistemas de detección de intrusos (IDS) basados en red son vulnerables a la información cifrada y a los ataques que son desconocidos por su motor de análisis. Ante esta situación, se presenta un modelo que permita el aislamiento de un ordenador (host) que se encuentre comprometido y éste pueda a su vez, comprometer la seguridad de una red por medio del control e instalación de códigos maliciosos. Este aislamiento es realizado mediante la herramienta Fira que recibe las instrucciones provenientes de la central de infromación (módulo de alerta) a través de una lista que contiene las direcciones IP de los hosts detectados como comprometidos.

Objetivos

- Presentar un modelo de un detector de intrusos basado en sistema experto. que permita emplear un sistema de aislamiento hacia un host que ha sido comprometido por un atacante, quién evadió los mecanismos de seguridad en un sistema.
- Mostrar los módulos y componentes que constituyen el modelo propuesto.
- Exponer el alcance y las debilidades del modelo propuesto.

4.1 Introducción al capítulo

Como se mencionó en el Capítulo 2 la seguridad absoluta es difícil de obtener, por lo que es necesario complementar los mecanismos de seguridad que protegen a un sistema; debido a que éstos pueden ser evadidos por un atacante que posea la habilidad para hallar vulnerabilidades en los mecanismos que protegen a un sistema y/o los componentes que la integren (ordenadores, dispositivos de comunicación, etc.). Los sistemas de detección de intrusos tipo red revisan los paquetes que han sido filtrados por el cortafuegos, en busca de un indicio de intrusión o código malicioso que desee acceder hacia un sistema. Sin embargo, este tipo de IDS no puede revisar el contenido de paquetes cifrados o aquellos que no sean reconocidos por su motor de análisis (por ejemplo, ataques de Día 0), pasando éstos a ser identificados como confiables, cuando éstos pueden tener intenciones de intrusión oculta [54]. Esto representa una gran vulnerabilidad hacia los equipos de cómputo (hosts) que integran una red, los cuales pueden ser controlados por un atacante y con ello, comprometer completamente la seguridad de una red.

Cuando un IDS es evadido por un atacante, los hosts que integran la red interna son comprometidos, puesto que es difícil que éstos cuenten con una herramienta complementaria a la que ofrece un IDS de tipo red, es decir, un IDS de tipo host. Esto se debe, a que para implementar esta última opción en cada host se requiere de un estudio previo para evaluar la degradación que sufrirá en su rendimiento y en los servicios que brindan hacia otros hosts que integren la red interna, así como el costo en la infraestructura y recursos humanos que de ello derive.

4.2 Propuesta

La evidencia de que un intruso ha evadido a un cortafuegos y a un IDS, es manifestada por una reacción que se produzca en el sistema. La reacción a la que se refiere es la que denota un comportamiento no habitual o instalación de códigos maliciosos en un sistema. Esto puede ser observado por ejemplo, en la degradación del rendimiento del ordenador, saturación de procesos, modificación de información, entre otros.

4.2.1 Modelo

El modelo propuesto no se enfoca sobre las posibles mejoras hacia un modelo previamente propuesto por alguna universidad o sector comercial. Su objetivo es dar una posible respuesta activa a los paquetes malformados que no son detectados por un IDS por encontrarse cifrados o desconocidos. Se ofrece controlar y aislar situaciones que comprometan a un host antes de que éstas se propaguen, y puedan provocar un daño irreversible a la red a la que pertenecen. Su diseño le permite ser flexible para la evaluación de diferentes parámetros en un host, los cuales pueden ser previamente establecidos a criterio del propietario del sistema para ser comparados con el motor de análisis, y posteriormente emitir el sistema de alerta correspondiente. Asimismo, esta propuesta puede ser considerada como un complemento de Snort, puesto que puede

emplear sus bitácoras para detectar un ataque proveniente de una red externa hacia un host específico de la intranet, que contenga información sensible.

La operatibilidad de su diseño se ve reflejada cuando la central de información manda la señal de alerta hacia el administrador del sistema y a los hosts que todavía no son comprometidos en la red. Esta propuesta se basa en detectar a un host que ha sido comprometido por un atacante, identificarlo, etiquetarlo y en apariencia permitir que se comunique con los dispositivos que integran la red. Sin embargo, la comunicación no es establecida con ninguno de los hosts, hasta que se envíe la indicación por parte de la central hacia los hosts que todavía no se encuentran comprometidos. Ésto es realizado por medio del descarte de todos los paquetes que provengan del host comprometido a través de la herramienta propuesta denominada Fira (se detallará más adelante en este capítulo). Lo que representa el aislamiento del equipo en la red, al mismo tiempo que el sistema de alerta del modelo propuesto efectúa la notificación de la intrusión, que pueda servir como retroalimentación hacia el motor de análisis de un IDS y/o las acciones pertinentes a efectuar por parte del administrador del sistema.

4.2.1.1 Descripción general de la propuesta

El modelo propuesto tiene como objetivo el detectar un intruso que evadió los mecanismos de seguridad de una red y que llega a un host que no cuenta con un mecanismo de seguridad adicional para su protección, presentando con ello, un riesgo inherente en la red. Su diseño contradice la lógica para la comunicación entre los hosts que integran una red, puesto que se propone trabajar en forma contraria, es decir, cerrar toda comunicación con los hosts que integran la red, hasta que se le indique con qué hosts puede establecer una comunicación y con cuales no. Para cumplir el objetivo mencionado, se propone un pequeño filtro de direcciones IP (de nombre Fira) que permita aislar la comunicación entre los hosts, el cual es regulado por los criterios de detección que resulten de la inferencia sobre los parámetros previamente determinados por el propietario del sistema.

Durante el análisis de las técnicas mostradas en el estado del arte, se observó que adicionalmente a las mejoras que se le pueden proporcionar al motor de análisis de un IDS para aumentar su detección y disminuir las métricas de falsos positivos y verdaderos negativos, se requiere tomar acciones oportunas cuando el motor de análisis es evadido por un atacante, cuando éste crea un paquete cifrado o que es desconocido su patrón por el motor de análisis. Las acciones a las que se hace referencia se enfocan hacia la comunicación entre los sistemas a proteger, ya sea de manera centralizada o distribuida, así como la identificación de ciertos parámetros que indiquen la presencia de un intruso por medio de la reacción o cambios entre ellos.

El modelo está inspirado a criterio personal en el análisis de la reunión de las mejores cualidades que emplean diferentes técnicas para la detección de intrusos, como son: sistema inmune, IPS, escenarios, honeypots, sistemas expertos y la funcionalidad de los cortafuegos y las NACs (Siglas en inglés de redes de control de acceso).

Se consideraron estas técnicas a diferencia de otras que se encuentran citadas en el estado del arte, por la razón expresada de tomar acciones que inhiban la propagación en la red de un equipo comprometido por un atacante y las cualidades que enriquecen su diseño.

El enfoque que se realiza sobre la técnica del sistema inmune como lo plantea Forrest [96] se enfoca a la identificación de lo que es propio y no dentro de un sistema. Sin embargo, el enfoque que se le da a esta propuesta no es la identificación como tal, sino el modelo de reacción-acción que realiza el sistema inmune de los vertebrados, en el que se basan las teorías de Forrest. Donde se indica la forma en la que llega y es identificado un antígeno al cuerpo de un vertebrado, así como la forma en que el antígeno es atacado por un anticuerpo, el cual es producido por una célula denominada B y una célula T auxiliar, y el control que es proveído sobre los anticuerpos por medio de una célula llamada T. Asimismo, el constante monitoreo que realiza el sistema inmune en busca de antígenos.

La técnica empleada por los IPS permite tomar acciones ante un intento de intrusión proveniente de una extranet por medio del descarte de paquetes, bloqueos de puertos, entre otros [103], partiendo del análisis de su operación es posible diseñar un control de la comunicación entre los hosts que conforman una intranet a través de un cortafuegos individual. La técnica de escenarios permite representar diferentes tipos de ataques que pueden presentarse en un sistema sean conocidos o no, así como la metodología que puede emplear un intruso para vulnerar un sistema. Una de las técnicas más importantes que inspiraron el modelo propuesto, es la que utilizan los honeypots en el arte del engaño, puesto que plantea como mantener a un atacante observado sin que éste se percate de ello.

Como se mencionó en el capítulo 1, los mecanismos de defensa se encuentran en continua evolución y una solución propuesta que conjunta varios mecanismos de defensa es conocida como NAC (por sus siglas en inglés, Network Access Control) [128]. Las NACs permiten realizar el control de acceso a la red por medio del cumplimiento de las políticas establecidas por los propietarios de los sistemas a proteger, si éstas no son cumplidas es posible bloquear a los hosts hasta que satisfagan las políticas correspondientes. Mediante las NACs también es posible, la revisión sobre parámetros que puedan comprometer la seguridad de la red, tales como actualización de parches de seguridad, actualización de antivirus, entre otros. En lo referente a los sistemas expertos, se extrae la aplicación de las reglas como resultado de una inferencia sobre los hechos presentados como intentos de intrusión, para generar una conclusión que determine si se trata o no de una intrusión.

Su diseño es basado en la conjunción de las técnicas anteriormente descritas, lo que le permite desarrollar un modelo de detección de intrusos basado en sistema experto, que aplique sus reglas para controlar la comunicación entre los hosts que integran una red.

4.2.1.2 Infraestructura del modelo

Su topología es de tipo estrella debido a la ventaja que se presenta cuando un host (ordenador) se encuentra fuera de servicio (en este caso comprometido), sin que éste afecte la operación de la red.

En su arquitectura se establece el protocolo de comunicación entre los hosts y la recolección de los parámetros a analizar, un sistema de análisis basado en reglas que evalúa los parámetros previamente pactados para la detección de un intruso y un sistema de alerta, que es el encargado de regular el funcionamiento de Fira, para el esquema de aislamiento de un host comprometido.

El modelo propuesto opera bajo dos plataformas de sistemas operativos, por parte de los hosts se considera el software comercial de Windows, debido a que en la actualidad es el software más empleado por su interfaz amigable hacia los usuarios de las diferentes redes mundiales, y en la parte del análisis de intrusión, se encuentra el sistema operativo Linux, que es una plataforma de código abierto este último fue seleccionado por la flexibilidad que representa para los investigadores en el conocimiento profundo de los sistemas de detección de intrusos, en este caso particular Snort.

4.2.1.3 Estructura del modelo

El modelo está integrado por los siguientes módulos (Fig. 4.1):

- Módulo de requerimiento

  Está constituido por los hosts de una intranet, el agente de recolección de parámetros y una central de información para analizar los parámetros enviados por parte de los hosts. La central después de recibir los parámetros los envía al módulo de análisis.

  Los hosts tendrán implementada la herramienta denominada Fira, quién se encargará de comunicarse exclusivamente con el servidor, recolectar los parámetros solicitados, enviarlos y esperar la respuesta de la central de información.

- Módulo de verificación

  Este módulo tiene la consigna de buscar una anomalía en los parámetros recidos de los hosts. Emplea un algoritmo de detección para validar los parámetros recibidos y asignar diferentes códigos que indiquen el estado del host que se está evaluando, los cuales sirven como base para el módulo de alerta y aplicar el proceso de aislamiento que correspondan según el caso. Esto es si existe una diferencia con lo que se establezca como normal se marcará como intrusión. El análisis de la detección será expuesto en otra sección del presente capítulo.

- Módulo de alerta

  Este módulo está constituido por dos partes: la notificación de los resultados hallados durante la detección y el sistema de aislamiento que controla la herramienta Fira. Se encarga de generar una lista de las direccciones IP de los hosts catalogados como comprometidos y se los hace llegar a los hosts que fueron identificados durante el periodo de análisis como no comprometidos para que apliquen las políticas pertinentes para la comunicación entre los hosts de una red.

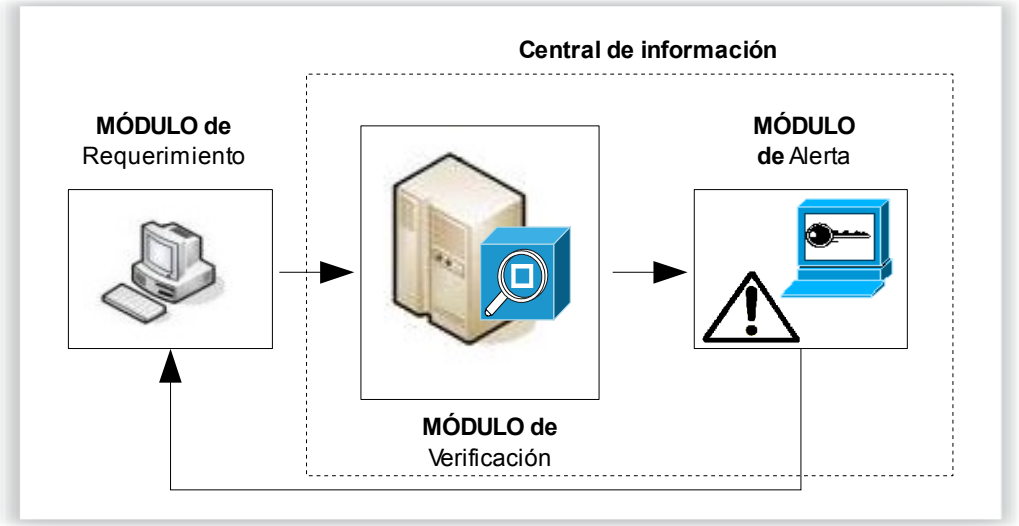


Fig. 4.1: Módulos del modelo propuesto

4.2.1.4 Arquitectura del modelo

A continuación se analiza en profundidad la funcionalidad de los módulos que integran el modelo propuesto:

4.2.1.4.1 Módulo de requerimiento

Está integrado por la herramienta denominada Fira, la cual recolecta la información de los parámetros residentes en los hosts y espera la notificación de comunicación entre los hosts que integran la red.

Protocolo de comunicación entre los hosts y la central

1. Se define que la comunicación entre los hosts y la central de información es bidireccional.

2. La comunicación comienza en el host.

3. El host establece una conexión hacia el servidor, a través de socket y espera respuesta.

4. La central de información Acepta la conexión del host, al terminar la transmisión de información proveniente del host, espera la conexión de otro host. Envía la respuesta proporcionada por el sistema de alerta hacia los hosts.

5. El host recibe la respuesta o solicitud de la central de información, según sea indicado por el sistema de alerta, y se mantiene en escucha para futuras indicaciones de la central de información.

6. Si el host no recibe respuesta de la central de información, mantiene sus mismas reglas de aislamiento hasta que éstas sean actualizadas (modificadas).

Fira

Fira es un filtro de direcciones IP diseñado para operar sobre la plataforma Windows. La elección de esta plataforma se debe a que Windows es el sistema operativo más popular que es empleado por los usuarios de diferentes redes mundiales . Para su implementación se requirió la creación de un driver de kernel que es programdo con el Windows Server DDK de Microsoft, el cual permite ocupar la extensión de un driver denominado Filter-Hook Driver, para la creación de filtros basados en direcciones IP. Debido a los cambios de versiones de sistema operativo por parte de Microsoft, esta herramienta desarrollada sólo puede operar en versiones de windows 2000 y XP, puesto que las versiones anteriores no cuentan con esta funcionalidad, y en el caso de Windows Vista, fue modificada en su totalidad esta parte [127]. Sin embargo, es posible adaptar este concepto mediante la documentación que ofrece Microsoft sobre este último sistema operativo.

En esta primera fase de investigación la herramienta Fira es implantada en cada host de forma manual, dejando para un trabajo futuro su implantación automática. Una vez que es puesta la herramienta en ejecución, ésta opera en modo restrictivo, el cual permite establecer el canal de comunicación entre la central y el host. Posteriormente, se recolectan los parámetros previamente indicados por el propietario del sistema para ser enviados y espera recibir las indicaciones de comunicación por parte de la central. Si la central no responde, el host quedará incomunicado al igual que los otros hosts hasta que se le de indique lo contrario. Esto puede verse como un ataque del tipo DoS hacia la red misma, al momento en que los hosts quedan incomunicados por no recibir indicaciones de la central de información. Sin embargo, al presentarse esta situación sería una muestra evidente, que indicaría la presencia de una red totalmente comprometida por un atacante o que se está violando el servicio de seguridad de la disponibilidad por un ataque físico realizado desde la propia intranet.

Previendo la situación anterior, el modelo fue diseñado para que en un trabajo futuro sea posible realizar un esquema de redundancia entre varias centrales de información

dentro de una intranet. El bosquejo propuesto, sería :

Opción A (Fig. 4.2)

Se propone emplear una central por cada subred, en donde cada central sea monitoreada por un control de centrales, quién revisará que no exista incosistencia en los datos contenidos en la base de conocimientos, modificación de archivos, instalación de malware, entre otros. Si alguna de las centrales se detecta que ha sido comprometida o está en mantenimiento preventivo, puede ser bloqueada la conexión y direccionar los hosts hacia la central más cercana. El análisis y control se dejan para un trabajo futuro, cabe mencionar que con esta opción se trata de mostrar que el mismo concepto de aislar un host comprometido, puede ser llevado hacia toda una subred.

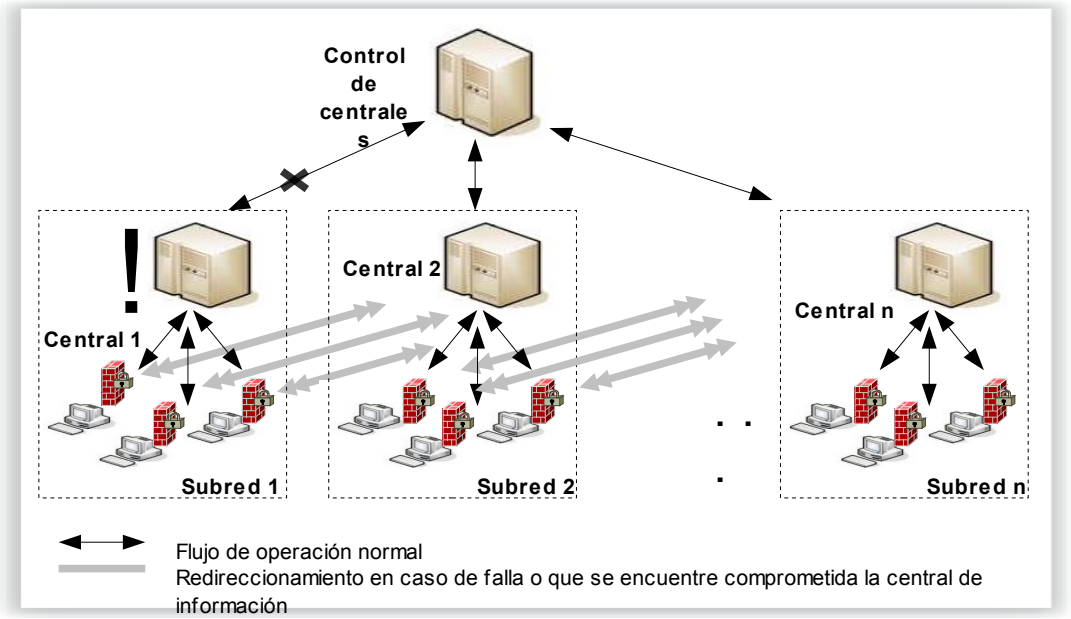


Fig. 4.2: Primera opción de redundancia

Opción B (Fig. 4.3)

A diferencia de la opción anterior ésta propone contar en la misma subred con centrales redundantes sin tener una unidad de control que valide a ambas, es decir, una central a otra se revisará su integridad. En caso de encontrar que una central ha sido comprometida se le aislará como si fuera un host más en la subred.

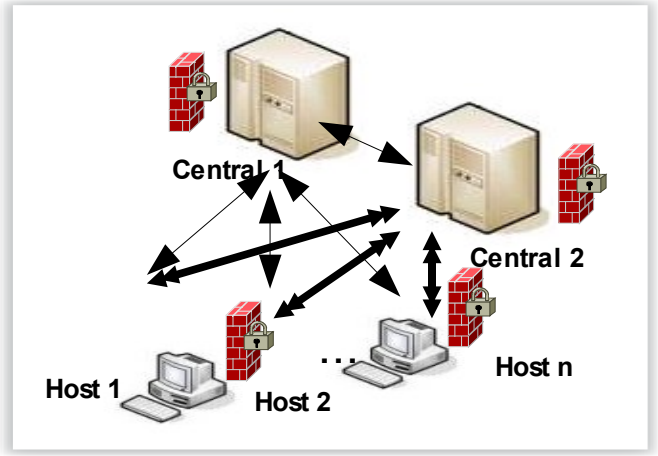


Fig. 4.3: Segunda opción de redundancia

Fracción del código de Fira

La primera actividad que debe realizar fira es habilitar el filtro de direcciones IP, a través de la inicialización del driver de kernel, denominado IpFltDrv.sys (Fig. 4.4)

```
m_ipFltDrv.WriteIo(RESTRICTIVE_MODE, NULL, 0) == DRV_SUCCESS
```

Fig. 4.4: Activación a modo restrictivo para comunicarse con la central.

Inicialización del driver de kernel para inicializar a Fira (Fig. 4.5) :

```
int main(int argc, char *argv[])
{
TDriver m_filterDriver;
TDriver m_ipFltDrv;

m_filterDriver.LoadDriver("IpFilterDriver", "System32\\Drivers\\IpFltDrv.sys", NULL, TRUE);
m_filterDriver.SetRemovable(FALSE);
m_ipFltDrv.LoadDriver("DrvFltIp", NULL, NULL, TRUE);

if(m_ipFltDrv.WriteIo(START_IP_HOOK, NULL, 0) != DRV_ERROR_IO)  {
```

Fig. 4.5: Inicialización del driver de kernel

Lista que es recibida (Fig. 4.6) por parte de la central con las direcciones IP de los hosts comprometidos:

```
char *bannedList[] = {"     "};
```

Fig. 4.6: Indicación de las direcciones IP a restringir.

El código de restricción (Fig. 4.7) para la comunicación entre los hosts en una red es bidireccional (paquetes entrantes, paquetes salientes), lo que implica que cada dirección IP será restringida en ambas direcciones por Fira:

```
IPFilter ip;

ip.sourcePort = 0; // htons(atoi((LPCTSTR)m_ssport));
ip.protocol = 0; // setproto;
ip.drop = TRUE;

for (i=0; i < (numlista*2); i++)
  {

// Filtrado para paquetes de red entrantes de acuerdo a bannedList[i]

j=numlista+1;

      for( int i=0; i<(numlista*2); i++ ) {
          if( i<numlista )  {
              ip.destinationIp = 0; // inet_addr((LPCTSTR)m_sdadd);
              ip.destinationMask = 0; // inet_addr("255.255.255.255");
              ip.sourceIp = inet_addr(bannedList[i]); // inet_addr((LPCTSTR)m_ssadd);
              ip.sourceMask = 0xFFFFFFFF; // inet_addr("255.255.255.255")
          } else {
              ip.destinationIp = inet_addr(bannedList[i-j]); // inet_addr((LPCTSTR)m_sdadd);
              ip.destinationMask = 0xFFFFFFFF; // inet_addr("255.255.255.255");
              ip.sourceIp = 0; // inet_addr((LPCTSTR)m_ssadd);
              ip.sourceMask = 0; // inet_addr("255.255.255.255")
   }

          if( m_ipFltDrv.WriteIo(ADD_FILTER, &ip, sizeof(ip)) == DRV_SUCCESS ) {
                printf("Cargada la regla con exito\n");
```

Fig. 4.7: Bloqueo de las direcciones IP indicadas por las reglas de la central.

4.2.1.4.2 Módulo de verificación

En este módulo se encuentran las reglas que se aplicarán para la detección de un intruso en la red. Su detección puede estar basada en el método de anomalías, en el método de abusos o ambos, esto dependerá de la técnica que se desee emplear para la búsqueda de un parámetro en particular. En el caso del presente documento se utiliza la metodología de anomalías, en el que se indica que cualquier parámetro que difiera de los criterios establecidos se considerará una intrusión. Esta metodología fue seleccionada de forma indistinta a la metodología de abusos, puesto que lo que se pretende mostrar es la flexibilidad que ofrece el modelo propuesto para integrar cualquiera de las metodologías en su motor de análisis, sin que ésto cuestione cuál es o no la mejor técnica.

El motor de análisis se implementa en el sistema operativo Linux, esto se hace con el objetivo de emplear un software de código libre como Snort en su ambiente nativo, permitiendo con esto la comprensión de los módulos que integran a un IDS de tipo red y con ello, analizar los puntos que pueden complementar y mejorar en sus características a este tipo de IDS. Otra razón, fue dejar a consideración de un trabajo futuro la interconexión automatizada con las bitácoras y preprocesadores de Snort a través de la central de información.

Parámetros y reglas del modelo propuesto

Los parámetros a considerar dentro del modelo, son detallados en las Tablas 4.1 y 4.2 .

En la primera, se muestran los parámetros que pueden ser usados para detectar un intruso en una intranet, sin que éstos limiten el crecimiento de la evaluación de más parámetros, y en la segunda tabla, se presentan los parámetros a ser buscados en las bitácoras de Snort, siguiendo la misma idea de no restringir su uso a sólo esos parámetros.

Sin Snort

Tabla 4.1: Parámetros que pueden ser evaluados en un sistema

Parámetros	Valores
DirIP de host	{ alterada, sin alterar }
DirMAC de host	{ alterada, sin alterar }
Servicios del hosts	{ sin adicionar, adicionar}
. . .	. . .
Parámetro n	{valor V, Valor F }

Se seleccionaron estos tres parámetros a evaluar, debido a que éstos representan algunos de los ataques de mayor peligrosidad que puede sufrir una red:

- DirIP
  Identifica la dirección desde la que pueden provenir ataques desde el exterior o interior de la red, de manera interna puede mostrar escalamiento de privilegios, un ataque de tipo spoofing hacia los servicios que brinda la red, entre otros.

- DirMAC
  Este parámetro puede parecer no muy significativo, sin embargo no es así. Un atacante puede suplantar la dirección MAC de un host, con la finalidad de realizar un ataque de tipo spoofing en cualquiera de sus modalidades, ya sea sobre los servidores o equipo de comunicación que integren la red interna.

- Servicios
  Si un atacante instala un software adicional al host; en la mayoría de los casos puede ser visualizado en los procesos activos. Sin embargo, ésto no es una regla general, puesto que si se tratase de buscar un tipo de malware es necesario listar los procesos ocultos que están activos, pero que no se visualizan en primera instancia.

Con Snort

Tabla 4.2: Parámetros que pueden ser seleccionados en Snort.

Parámetros	**Valores**
DirIP servidor	{ coincide, no coincide}
Ataque específico a buscar hacia un servidor	{ hallado, no hallado }
. . .	. . .
Parámetro n	{valor V, Valor F }

Reglas

Las reglas que se aplicarán dependerán directamente de la ubicación del sensor, es decir, de dónde provienen los datos que serán analizados por el modelo propuesto: host y/o Snort.

La estructura de las reglas es la siguiente:

Parámetro1 a evaluar = par_eval1
Parámetro2 a evaluar = par_eval2
Condición verdadera = TRUE
Condición falsa = FALSE

- Inferencia por Modus Ponens

  SI par_eval1 = TRUE && SI par_eval2 = TRUE y se sabe que par_eval1=TRUE
  ENTONCES par_eval1 & par_eval2 son TRUE

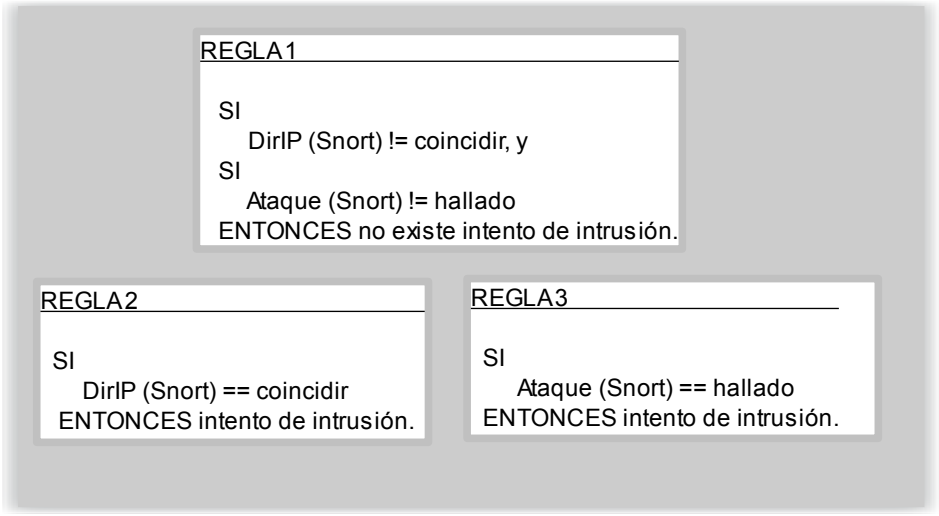
- Inferencia por Modus Tollens

  Si se sabe que par_eval1= TRUE && SI par_eval2 = FALSE
  ENTONCES par_eval1 = FALSE

Reglas con Snort (Modalidad extranet)

Para el caso de la extranet empleando las bitácoras de Snort, TRUE es el valor que le daremos a un parámetro que cumpla con los valores previamente definidos para detectar una anomalía en el sistema, lo que lleva a la conclusión de que verdaderamente existe una intrusión, y FALSE indicará que uno de los parámetros está fuera de ser considerado como anómalo, por lo que se infiere de que no se trata de una intrusión (Tabla 4.3).

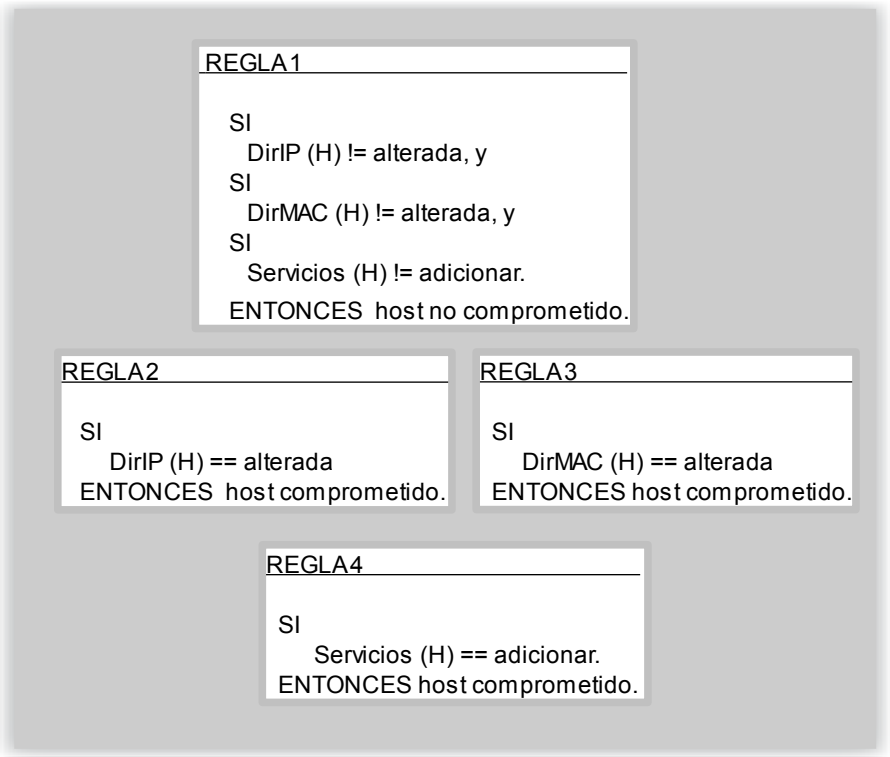
Tabla 4.3: Reglas del modelo para la extranet.



Reglas sin Snort (Modalidad intranet)

En el caso de las reglas para la intranet, TRUE es el valor que le daremos a un parámetro que cumpla con los valores previamente definidos como normales, lo que lleva a la conclusión de que verdaderamente no existe una intrusión, y FALSE indicará que uno de los parámetros está fuera de los parámetros anormales, por lo que se infiere que se trata de una intrusión (Tabla 4.4).

Tabla 4.4: Reglas del modelo para la intranet.



Algoritmo de detección

El modelo propuesto es el diseño de un complemento a los sistemas de detección de intrusos de red, para dar una respuesta activa y proactiva ante un atacante interno y/o externo. Su algoritmo de detección se basa en la recolección de parámetros previamente establecidos para la búsqueda de un patrón o comportamiento anómalo, que al ser analizados mediante la aplicación de las reglas es posible inferir si existe o no una intrusión en el sistema de red.

El modelo presentado es catalogado como un modelo de causa-reacción (Fig. 4.1), el cual actúa con base en el análisis de la información recibida (causa) y presenta una

respuesta a dicha detección (reacción). Debido a las características de su diseño, éste puede puede o no interactuar con el IDS de red Snort para detectar la presencia de un intruso desde la extranet, empleado para ello, las bitácoras propias de Snort considerándose lo registrado como un intento de intrusión. La otra opción es sin utilizar Snort, este modo operativo permite la posibilidad de buscar un intruso en cada host a través de la recolección de ciertos parámetros que puedan indicar su presencia en la intranet. También pueden ser empleados en conjunto para ampliar el dominio de protección de una red.

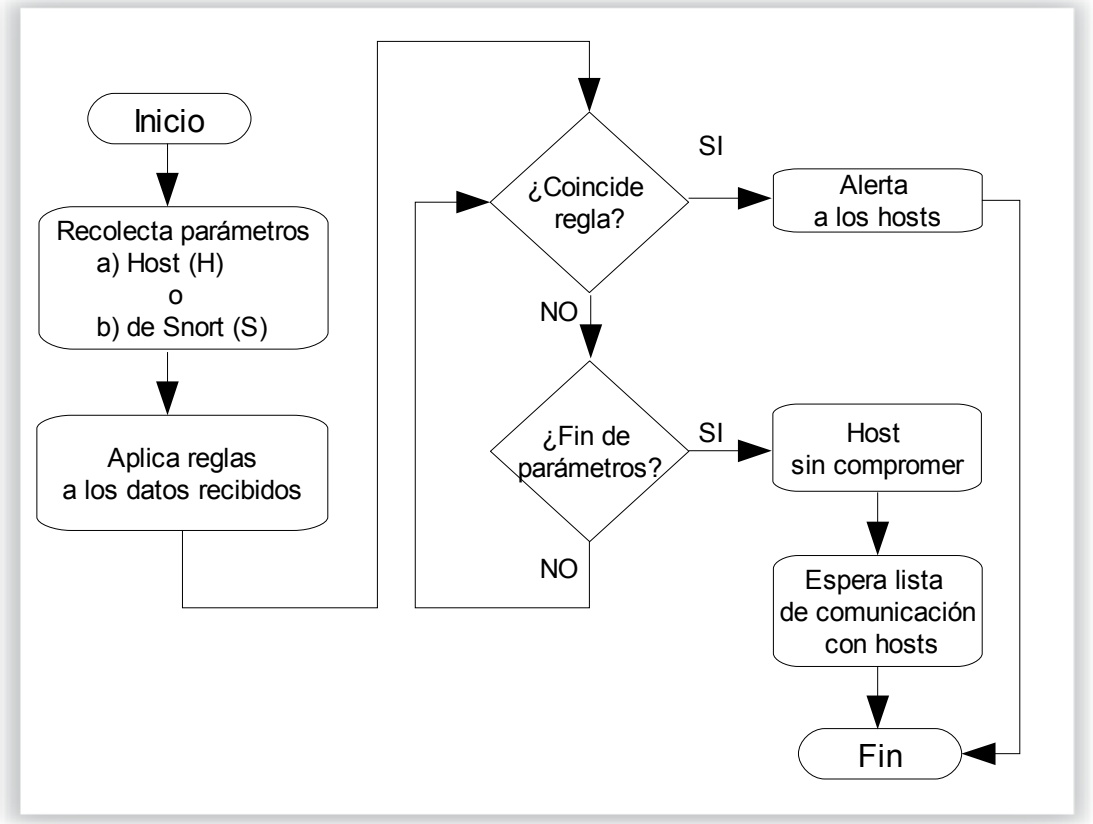


Fig. 4.8: Representación a bloques del algoritmo de detección propuesto.

A continuación se exponen las características de detección para cada caso, en los que se observará que la funcionalidad es la misma, lo único en lo que se difiere es el lugar de dónde se tomarán los datos para ser evaluados.

- Con Snort (modalidad externa)

  Supongamos el siguiente escenario: *Un atacante externo desea reconocer o acceder a un equipo de la intranet que contiene información sensible, por ejemplo, un servidor de aplicaciones, nómina, correo, etc. o que desea explotar*

una vulnerabilidad a través de exploits para vulnerar dicho servidor.

Por medio de Snort es posible detectar la dirección IP origen del posible atacante y la dirección IP destino a la que se dirige el ataque, si la IP destino coincide con la dirección IP de un servidor que contenga información sensible, es posible por medio del modelo propuesto detectar y asilar la intrusión de la siguiente forma:

Snort registra el análisis de los paquetes de red que coinciden con los patrones de sus reglas de configuración predeterminadas en diferentes tablas (),
si se parte de esta información, es posible con el modelo propuesto acceder a los registros de Snort (módulo de requerimiento) y buscar una información particular.

En este caso se propone buscar las direcciones IP destino que coincidan con las direcciones IP de un servidor o varios servidores, si existe coincidencia, entonces se obtendrá de los registros de Snort la dirección IP origen, de donde proviene el intento de intrusión. Esta misma metodología de detección es aplicada para buscar un ataque específico que vaya dirigido hacia un servidor.

Una vez que se presenta la coincidencia y se identifica la dirección IP origen del atacante, el módulo de alerta enviará a los hosts de la intranet y a los servidores en cuestión la lista de la(s) dirección(es) IP del atacante para denegar su acceso a la red. Cabe aclarar, que ésto es sólo un ejemplo de lo que se puede lograr al manipular las bitácoras de snort y no limita la búsqueda de diversos ataques a la vez, ni el control sobre ataques hacia los puertos de un servidor.

Actualmente el modelo no está implementado para interactuar de forma automática con las bitácoras de Snort, se deja como un trabajo futuro debido a que se sugiere que se evalúen las mejores técnicas de explotación de sus bitácoras, para una rápida obtención de los datos y detectar la intrusión lo más pronto posible.

- Sin Snort (modalidad interna)

En el párrafo anterior, se comentó como Snort registra en sus bitácoras lo que detecta mediante la aplicación de sus reglas como un ataque. Sin embargo, ¿qué sucede con los paquetes que son cifrados o que evaden la seguridad de un IDS, por ejemplo un ataque del tipo Día 0 y éste llega a un host de la red interna? ¿Qué sucede si un usuario de un host instala software ilícito o modifica la configuración del host, convirtiéndose esta situación en un ataque interno? El modelo propuesto emplea a Fira como una herramienta de reacción al momento de detectar una intrusión, a través de la evaluación de los parámetros que se le requieran y sean analizados para determinar la presencia de un intruso o no en la intranet.

Los parámetros que pueden ser considerados y adicionados al modelo propuesto para tratar las necesidades particulares de la seguridad sobre un sistema, son enunciados sin que esto signifique una restricción sobre la búsqueda de uno o varios parámetros en particular; como ejemplos podemos considerar: el uso de procesador, consumo de memoria, modificación de archivos, firma específica de un rootkit, verificación de la actualización del antivirus, entre otros más.

Con estos escenarios se expone la forma en la que funciona el algoritmo de detección, el cual se encarga de recolectar los datos a analizar, ya sea desde el agente de recolección de información de los hosts como parte de la herramienta Fira o bien, de los registros de Snort, los cuales son evaluados en busca de la coincidencia de anomalías establecidas en las reglas del presente modelo, estableciendo un detector de intrusos basado en sistema experto (Fig. 4.9) . El sistema experto está conformado:

- Reglas de snort.
- Reglas de parámetros.
- Base de conocimientos de Snort mediante la técnica de reconocimiento de patrones.
- Base de conocimientos de la central empleando la metodología de anomalías.
- Aplicación de las reglas como respuesta a una intrusión.

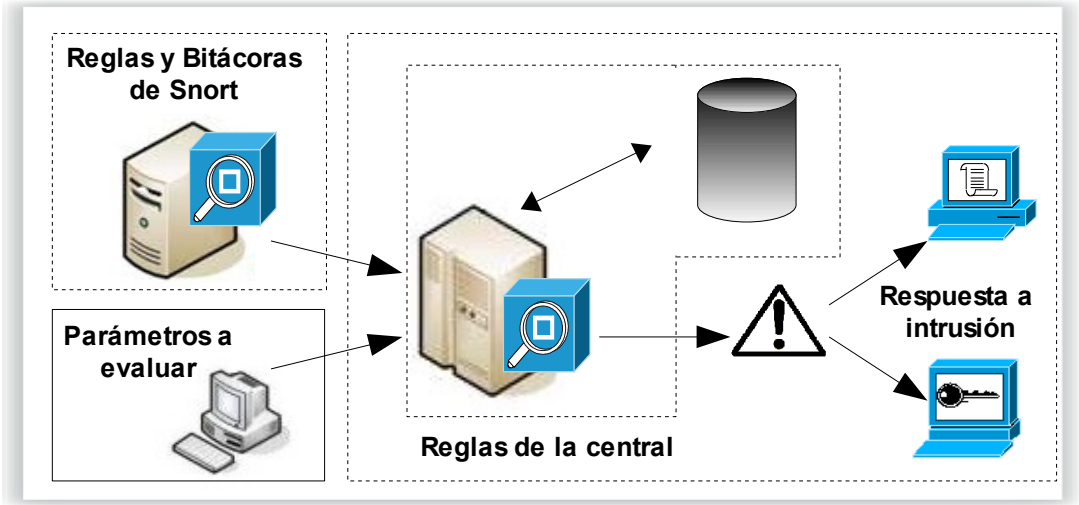


Fig. 4.9: Modelo propuesto de un IDS basado en sistema experto

Base de datos

La base de datos que es empleada por la central está constituida por dos partes: parámetros y estados.

Parámetros
Contienen la información que se desea sensar, por ejemplo: un proceso de sistema.

Estados
Especifica el estado que se le asigna a un host después de su detección, los estados que puede tener un host son:

- *007 (Comprometido)*
  Host con comportamiento anómalo y/o con instalación de código malicioso

- *111 (Libre)*
  Host que pasó la verificación de los parámetros previamente pactados.

- *222 (Fuera de Servicio)*
  Estado que se le asigna a un host que no contesta el requerimiento de la central, debido a: mantenimiento del host, daños físicos, host sin suministro de energía eléctrica, usuario del host no disponible (vacaciones o incapacidad).

- *333 (Alerta)*
  Host en estado de alerta. Es un estado adicional que se le da a un host para indicar que se encuentra fuera de servicio o comprometido.

- *000 (Valor inicial)*
  Host que inicialmente integra la red

- *777 (Nuevo)*
  Host que se anexa por primera vez a una red.

Flexibilidad de los parámetros

El modelo propuesto permite anexar diferentes parámetros que se deseen sensar en un host o en las bitácoras de Snort, como pueden ser: el consumo de memoria, uso del microprocesador, ataques de bufferoverflow, escaneo de puertos, paquetes mal formados, etc. Cada uno de éstos parámetros puede emplear cualquiera de las técnicas de detección mencionadas en el estado de arte, y cuando su motor de análisis concluya la presencia de un intruso o intento de intrusión, es posible interactuar con Fira para

aislar el host que se ha detectado como comprometido mediante el control de la comunicación entre los equipos que conforman la red.

En el presente trabajo de tesis sólo se tomaron 3 parámetros como representativos sin ser excluyentes de considerar otros que evidencien por medio de una reacción la existencia de un intruso. Fueron empleados por considerarlos como los parámetros más representativos de algunos de los ataques potencialmente más peligrosos, como son: el tipo Looping, el DoS y el Spoofing en sus diferentes versiones.

4.2.1.4.3 Módulo de alerta

El módulo de verificación por medio de su base de datos asigna los estados correspondientes, que son resultantes del proceso de análisis realizado a la información que es enviada por un host. Estos indicadores permiten efectuar las alertas correspondientes a el propietario del sistema y realizar el proceso del sistema de aislamiento hacia un host que ha sido comprometido por un atacante.

El módulo de alerta es la parte del modelo propuesto en el que reside el sistema de aislamiento. Su arquitectura está constituida por Fira en el lado de los hosts y el módulo de alerta en la central de información (Fig. 4.10).

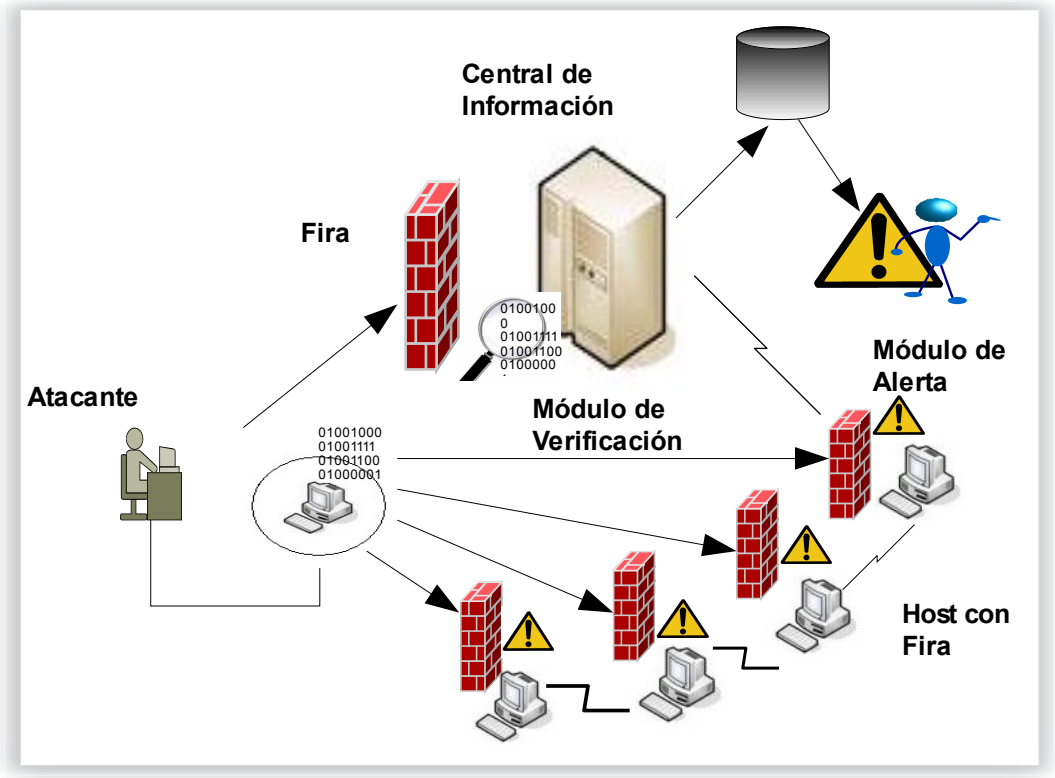


Fig. 4.10: Implantación del modelo propuesto

Después de que la central analiza la información y asigna los códigos correspondientes, el proceso de aislamiento se producirá bajo las siguientes condiciones:

- Si se detecta una diferencia entre la información registrada y la recibida, la central de información etiquetará en su base de datos a el host comprometido, colocando en el campo EDOASIG(detección) el código 007 y en EDOALER (alerta) el 333, lo que en consecuencia generará la señal de activación del aislamiento del equipo.

- Si no se detecta una diferencia entre la información registrada y la recibida, la central de información coloca en el campo de EDOASIG(Detección) el código 111, en señal que ha sido cotejada correctamente.

- Si es un nuevo host valida el código de autorización (777) que es anexado por el administrador del sistema y realiza su registro. En caso de no existir código de autorización lo etiqueta con el código 007 en el campo de Detección y 333 en el de alerta

- Si la central detecta que un host no ha enviado su información, éste envía una solicitud hacia el mismo. En caso de no tener respuesta alguna, lo etiquetará con el código 222 en el campo Detección y 333 en el campo de alerta.

El módulo de alerta revisa lo que fue asignado durante el análisis en el campo de detección, y realizará diferentes acciones acorde con los códigos asignados por el motor de análisis(Fig. 4.11).

Si el código asignado es:

- *007*
  El sistema de alerta prepara 2 tipos de archivos. Uno contendrá las direcciones IP a restringir por los hosts que todavía no se encuentran comprometidos y el otro, es un archivo cuyo contenido no presenta restricción alguna hacia los hosts que integran la red. Este último es devuelto al host comprometido, como estrategia de engaño mientras se dispara una alerta hacia el administrador de la red para que efectúe las acciones pertinentes al caso.

- *222*
  El sistema de alerta anexa al archivo que contiene las direcciones IP a restringir, la dirección IP del host con código 222, para que sean consideradas como el código 007 por los otros host, es decir, exceptuando el envío del archivo de respuesta hacia el host.

Nota: En el caso de que los hosts que se van anexando en el transcurso de un día a la red y se encuentran con código 222, éstos son nuevamente evaluados por la central de información. De acuerdo a los resultados del sistema de alerta, será cambiado su estado (111 o 007) y se notificará a los hosts restantes para que apliquen los cambios pertinentes en Fira.

- *111*
  El sistema envía un archivo con las direcciones IP correspondientes a las identificadas en el campo de detección con código 007, hacia el host solicitante.

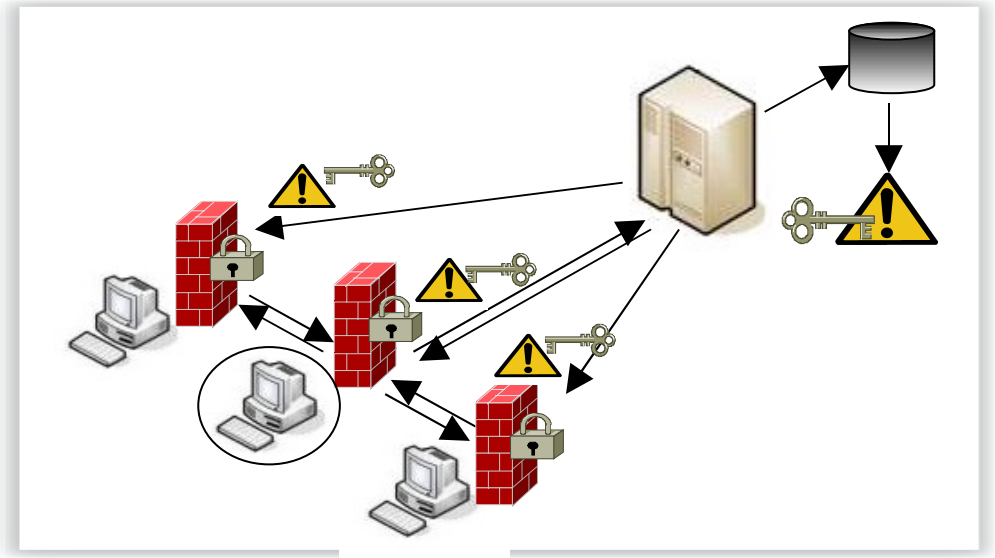


Fig. 4.11: Acción de Fira cuando recibe indicaciones de la central

4.3 Resumen de la operatividad del modelo

- El host bloquea la comunicación con los otros hosts excepto con la central por medio de fira.
- El host recolecta y envía la información previamente pactada para la evaluación.
- El host espera respuesta de central de información.
- La central evalúa la información que es enviada por un host. Si la información difiere, se genera el sistema de alerta y asignación de códigos correspondientes. (Ver estructura de la base de datos). En caso contrario, permite la comunicación regulada por la herramienta fira.
- Devuelve instrucciones al host.

Capítulo 5
Pruebas y resultados

Resumen
En este capítulo se describen las pruebas realizadas sobre el sistema propuesto.

Objetivos

- Probar la herramienta denominada como Fira para implementar el sistema de detección de ataques potenciales y el aislamiento de un host que ha sido comprometido e irrumpe la seguridad de la red interna.

- Analizar y mostrar los resultados obtenidos por el módulo de verificación y por el módulo de alerta en su interacción con la herramienta propuesta denominada Fira.

5.1 Ambiente de pruebas

El ambiente de pruebas fue implementado en la siguiente forma: Hosts y central de información.

Hosts

- En los hosts se instaló de manera manual la herramienta denominada Fira (cuyo funcionamiento se explicó en el capítulo 4).
- Fira es puesta en modo restrictivo, esto es, que no existe comunicación con otro host excepto con la central de información.
- Las pruebas en hosts se realizaron bajo la plataforma de los sistemas operativos Windows XP SP2 y Windows XP UE. La razón de emplear esta plataforma se debe a su enorme popularidad en las redes mundiales por la facilidad de uso brindada hacia los usuarios.

Central de Información

- El sistema operativo que se seleccionó para la implementación de la propuesta es Linux Slackware 10, instalando sobre dicha plataforma el software de código libre Snort Version 2.8.1 y la central de información con sus tres módulos: requerimiento, verificación y alerta, desarrollados en lenguaje C.
- La central de información contiene los parámetros previamente pactados por el propietario del sistema, que serán determinados para cada host que integre la red o subred correspondiente.
- Para probar la seguridad se utiliza el escáner de vulnerabilidades de nombre Nessus en su versión 3.0.6. Nessus es una herramienta que permite evaluar los huecos de seguridad (vulnerabilidades) que tiene una red o un host específico, esta herramienta opera en dos modos: el primero realiza la revisión de vulnerabilidades ya conocidas y de ataques que considera son de uso común por un atacante. El segundo, es más ofensivo en sus ataques buscando la Denegación de servicio del sistema que se está evaluando.
- Snort es utilizado para la detección y clasificación de los ataques que son realizados por Nessus, esto permite visualizar los ataques que son o no detectados por un IDS tipo red por medio de su motor de análisis.
- Se incluye como herramienta gráfica para la lectura de las bitácoras el software de BASE versión 1.3.9, BASE (Basic Analysis and Security Engine) es una interfaz gráfica de código libre, que es empleada para mostrar los datos de las bitácoras de snort vía web. Este software sustituye a la versión anterior denominada ACID (Analysis Console for Intrusion Databases).

5.2 Configuración de Snort

A continuación se muestra la configuración empleada para la presente propuesta, en la que se muestran las reglas y preprocesadores a emplear, en apariencia no sería relevante mostrar esta configuración, sin embargo, no es así. Esto se debe a que dependiendo del conocimiento adquirido por el administrador del sistema a proteger los parámetros pueden variar, así como el uso de las reglas (community rules y VRT rules), lo importante en este punto es hacer notar que la granularidad de su detección en primera instancia dependerá de su configuración y los conocimientos del administrador.

Inicialización de Snort

root@v:/usr/local/snort/bin#/usr/local/snort/bin/snort -c /usr/local/snort/etc/snort.config

```
  ,,_     -*> Snort! <*-
 o"  )~   Version 2.8.1 (Build 28)
  ""      By Martin Roesch & The Snort Team:
          http://www.snort.org/team.html
          (C) Copyright 1998-2008 Sourcefire Inc., et al.
          Using PCRE version: 7.0 18-Dec-2006

Running in IDS mode

       --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file /usr/local/snort/etc/snort.conf
PortVar 'HTTP_PORTS' defined :  [ 80 ]
PortVar 'SHELLCODE_PORTS' defined :  [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined :  [ 1521 ]
Frag3 global config:
    Max frags: 65536
    Fragment memory cap: 4194304 bytes
Frag3 engine config:
    Target-based policy: FIRST
    Fragment timeout: 60 seconds
    Fragment min_ttl:   1
    Fragment ttl_limit (not used): 5
    Fragment Problems: 1
Stream5 global config:
    Track TCP sessions: ACTIVE
    Max TCP sessions: 8192
    Memcap (for reassembly packet storage): 8388608
    Track UDP sessions: INACTIVE
    Track ICMP sessions: INACTIVE
Stream5 TCP Policy config:
    Reassembly Policy: FIRST
```

```
    Timeout: 30 seconds
    Min ttl:  1
    Options:
       Static Flushpoint Sizes: YES
    Reassembly Ports:
      21 client (Footprint)
      23 client (Footprint)
      25 client (Footprint)
      42 client (Footprint)
      53 client (Footprint)
      80 client (Footprint)
      110 client (Footprint)
      111 client (Footprint)
      135 client (Footprint)
      136 client (Footprint)
      137 client (Footprint)
      139 client (Footprint)
      143 client (Footprint)
      445 client (Footprint)
      513 client (Footprint)
      514 client (Footprint)
      1433 client (Footprint)
      1521 client (Footprint)
      2401 client (Footprint)
      3306 client (Footprint)
HttpInspect Config:
    GLOBAL CONFIG
      Max Pipeline Requests:    0
      Inspection Type:          STATELESS
      Detect Proxy Usage:       NO
      IIS Unicode Map Filename: /usr/local/snort/etc/unicode.map
      IIS Unicode Map Codepage: 1252
    DEFAULT SERVER CONFIG:
      Server profile: All
      Ports: 80 8080 8180
      Flow Depth: 300
      Max Chunk Length: 500000
      Max Header Field Length: 0
      Inspect Pipeline Requests: YES
      URI Discovery Strict Mode: NO
      Allow Proxy Usage: NO
      Disable Alerting: NO
      Oversize Dir Length: 500
      Only inspect URI: NO
      Ascii: YES alert: NO
      Double Decoding: YES alert: YES
      %U Encoding: YES alert: YES
      Bare Byte: YES alert: YES
      Base36: OFF
      UTF 8: OFF
      IIS Unicode: YES alert: YES
```

```
      Multiple Slash: YES alert: NO
      IIS Backslash: YES alert: NO
      Directory Traversal: YES alert: NO
      Web Root Traversal: YES alert: YES
      Apache WhiteSpace: YES alert: NO
      IIS Delimiter: YES alert: NO
      IIS Unicode Map: GLOBAL IIS UNICODE MAP CONFIG
      Non-RFC Compliant Characters: NONE
      Whitespace Characters: 0x09 0x0b 0x0c 0x0d
rpc_decode arguments:
    Ports to decode RPC on: 111 32771
    alert_fragments: INACTIVE
    alert_large_fragments: ACTIVE
    alert_incomplete: ACTIVE
    alert_multiple_requests: ACTIVE
Portscan Detection Config:
    Detect Protocols:  TCP UDP ICMP IP
    Detect Scan Type:  portscan portsweep decoy_portscan distributed_portscan
    Sensitivity Level: Low
    Memcap (in bytes): 10000000
    Number of Nodes:   36900

Tagged Packet Limit: 256
Loading dynamic engine /usr/local/snort/lib/snort_dynamicengine/libsf_engine.so... done
Loading all dynamic preprocessor libs from /usr/local/snort/lib/snort_dynamicpreprocessor/...
Loading dynamic preprocessor library
/usr/local/snort/lib/snort_dynamicpreprocessor//libsf_ftptelnet_preproc.so... done
  Loading dynamic preprocessor library
/usr/local/snort/lib/snort_dynamicpreprocessor//libsf_dcerpc_preproc.so... done
  Loading dynamic preprocessor library
/usr/local/snort/lib/snort_dynamicpreprocessor//libsf_dns_preproc.so... done
  Loading dynamic preprocessor library
/usr/local/snort/lib/snort_dynamicpreprocessor//libsf_ssl_preproc.so... done
  Loading dynamic preprocessor library
/usr/local/snort/lib/snort_dynamicpreprocessor//lib_sfdynamic_preprocessor_example.so... done
  Loading dynamic preprocessor library
/usr/local/snort/lib/snort_dynamicpreprocessor//libsf_smtp_preproc.so... done
  Loading dynamic preprocessor library
/usr/local/snort/lib/snort_dynamicpreprocessor//libsf_ssh_preproc.so... done
  Finished Loading all dynamic preprocessor libs from
/usr/local/snort/lib/snort_dynamicpreprocessor/
FTPTelnet Config:
    GLOBAL CONFIG
      Inspection Type: stateful
      Check for Encrypted Traffic: YES alert: YES
      Continue to check encrypted data: NO
    TELNET CONFIG:
      Ports: 23
      Are You There Threshold: 200
      Normalize: YES
      Detect Anomalies: NO
```

```
    FTP CONFIG:
      FTP Server: default
        Ports: 21
        Check for Telnet Cmds: YES alert: YES
        Identify open data channels: YES
      FTP Client: default
        Check for Bounce Attacks: YES alert: YES
        Check for Telnet Cmds: YES alert: YES
        Max Response Length: 256

SMTP Config:
    Ports: 25 587 691
    Inspection Type: Stateful
    Normalize: EXPN RCPT VRFY
    Ignore Data: No
    Ignore TLS Data: No
    Ignore SMTP Alerts: No
    Max Command Line Length: Unlimited
    Max Specific Command Line Length:
      ETRN:500 EXPN:255 HELO:500 HELP:500 MAIL:260
      RCPT:300 VRFY:255
    Max Header Line Length: Unlimited
    Max Response Line Length: Unlimited
    X-Link2State Alert: Yes
    Drop on X-Link2State Alert: No
    Alert on commands: None

DCE/RPC Decoder config:
    Autodetect ports ENABLED
    SMB fragmentation ENABLED
    DCE/RPC fragmentation ENABLED
    Max Frag Size: 3000 bytes
    Memcap: 100000 KB
    Alert if memcap exceeded DISABLED

DNS config:
    DNS Client rdata txt Overflow Alert: ACTIVE
    Obsolete DNS RR Types Alert: INACTIVE
    Experimental DNS RR Types Alert: INACTIVE
    Ports: 53
SSLPP config:
    Encrypted packets: not inspected
    Ports:
      443     465     563     636     989
      992     993     994     995

++++++++++++++++++++++++++++++++++++++++++++++++++
Initializing rule chains...
```

Las reglas empleadas para las pruebas son las del estándar “Community Rules”, cabe recordar que son reglas de descarga gratuita, realizadas por aportaciones de los

usuarios de Snort (comunidad).

Tipos de ataques que reconocerá snort:

- misc-attack
- misc-activity
- attempted-recon
- network-scan
- web-application-activity
- bad-unknown
- web-application-attack
- attempted-user
- policy-violation
- attempted-admin

Los ataques realizados a través de Nessus para vulnerar el sistema, se listan a continuación:

- AIX Local Security Checks
- Backdoors
- CentOS Local Security Checks
- CGI abuses
- CGI abuses : XSS
- CISCO
- Databases
- Debian Local Security Checks
- Default Unix Accounts
- Denial of Service
- DNS
- Fedora Local Security Checks
- Finger abuses
- Firewalls
- FreeBSD Local Security Checks
- FTP
- Gain a shell remotely
- Gain root remotely
- General
- Gentoo Local Security Checks
- HP-UX Local Security Checks
- MacOS X Local Security Checks
- Mandriva Local Security Checks
- Misc.

- Netware
- NIS
- Peer-To-Peer File Sharing
- Policy Compliance
- Port scanners
- Red Hat Local Security Checks
- Remote file access
- RPC
- SCADA
- Service detection
- Settings
- Slackware Local Security Checks
- SMTP problems
- SNMP
- Solaris Local Security Checks
- SuSE Local Security Checks
- Ubuntu Local Security Checks
- Useless services
- Web Servers
- Windows
- Windows : Microsoft Bulletins
- Windows : User management

5.3 Objetivos de las pruebas

- Mostrar la vulnerabilidad que los IDS tipo red presentan ante paquetes cifrados o desconocidos, que pueden ser utilizados por un atacante para comprometer a un host y poner en riesgo la seguridad de la red a la que este host pertenece.

- Presentar la funcionalidad del módulo de alerta que forma parte del modelo propuesto, para aislar a un host que ha sido comprometido por un atacante que ha evadido la seguridad del IDS, ya sea desde la parte de la extranet o la intranet.

- Exponer las debilidades y fortalezas del modelo propuesto.

5.4 Resultados de las pruebas

Las pruebas son realizadas en un ambiente de producción a diferencia de un ambiente controlado, esto se debe a que se desea mostrar la funcionalidad del modelo propuesto para una intranet y una extranet, con apoyo en primera instancia de las reglas de Snort

y las contenidas en la central de información.

En la configuración por omisión de Snort se indica que los paquetes cifrados que provienen del protocolo SSLPP no son analizados. Esto se debe a que el protocolo SSL es empleado para realizar comunicaciones seguras en una red, mediante la encriptación de los datos que se intercambiarán entre los extremos de la red (emisor-receptor), proporcionando con ello los servicios de seguridad de autenticación y confidencialidad de la información (Tabla 5.1).

Tabla 5.1: Paquetes no detectados por Snort

```
SSLPP config:
    Encrypted packets: not inspected
   Ports:
     443     465     563     636     989
      992     993     994     995
```

Este hecho es de suma relevancia para la hipótesis que sustenta el desarrollo de la presente tesis con referencia a la detección de un intruso que evade el sistema de detección de intrusos basado en red; puesto que el atacante puede emplear un ataque de tipo hombre en medio para robar la sesión y hacerse pasar por alguno de los dos extremos por medio de la técnica de spoofing.

A continuación se muestra la detección de los ataques de Nessus hacia un host específico de la red sensados por Snort.

```
Packet Wire Totals:
   Received:          3828
   Analyzed:          3827 (99.974%)
    Dropped:             0 (0.000%)
Outstanding:             1 (0.026%)
```

En la Fig. 5.1 y en la Fig. 5.2 se muestra un ataque de reconocimiento de puertos que es realizado desde una dirección IP externa 201.162.136.10 y un cliente que ha sido comprometido para efectuar reconocimiento de puertos y direcciones IP, cuya dirección IP es la 148.204.20.135 y la máquina víctima es la 148.204.20.33. Con base en la configuración de Snort considerada para este proyecto, Snort detecta la acción pero no puede aislar al host que está siendo utilizado como zombie por el atacante.

< Signature >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
[url] [cve] [icat] [cve] [icat] [bugtraq] [local] [snort] MISC UPnP malformed advertisement	148.204.45.66:1024	239.255.255.250:1900	UDP
[url] [cve] [icat] [cve] [icat] [bugtraq] [local] [snort] MISC UPnP malformed advertisement	148.204.45.66:1024	239.255.255.250:1900	UDP
[url] [cve] [icat] [cve] [icat] [bugtraq] [local] [snort] MISC UPnP malformed advertisement	148.204.45.66:1024	239.255.255.250:1900	UDP
[arachNIDS] [local] [snort] ICMP PING CyberKit 2.2 Windows	201.62.136.110	148.204.20.49	ICMP
[arachNIDS] [local] [snort] ICMP L3retriever Ping	148.204.20.135	148.204.20.33	ICMP
[url] [cve] [icat] [cve] [icat] [bugtraq] [local] [snort] MISC UPnP malformed advertisement	148.204.45.66:1024	239.255.255.250:1900	UDP
[local] [snort] SCAN UPnP service discover attempt	148.204.45.144:1028	239.255.255.250:1900	UDP
[local] [snort] SCAN UPnP service discover attempt	148.204.45.144:1028	239.255.255.250:1900	UDP
[local] [snort] SCAN UPnP service discover attempt	148.204.45.144:1028	239.255.255.250:1900	UDP
[url] [cve] [icat] [cve] [icat] [bugtraq] [local] [snort] MISC UPnP malformed advertisement	148.204.45.66:1024	239.255.255.250:1900	UDP
[bugtraq] [local] [snort] WEB-CGI awstats access	148.204.45.143:60964	148.204.45.101:80	TCP
[bugtraq] [local] [snort] WEB-CGI awstats access	148.204.45.143:60964	148.204.45.101:80	TCP
[bugtraq] [local] [snort] WEB-CGI awstats access	148.204.45.143:60964	148.204.45.101:80	TCP
[bugtraq] [local] [snort] WEB-CGI awstats access	148.204.45.143:60964	148.204.45.101:80	TCP
[bugtraq] [local] [snort] WEB-CGI awstats access	148.204.45.143:60964	148.204.45.101:80	TCP
[bugtraq] [local] [snort] WEB-CGI awstats access	148.204.45.143:60964	148.204.45.101:80	TCP
[nessus] [local] [snort] WEB-FRONTPAGE /_vti_bin/ access	148.204.45.143:60976	148.204.45.101:80	TCP
[local] [snort] WEB-CGI /cgi-bin/ access	148.204.45.143:60976	148.204.45.101:80	TCP
[local] [snort] ATTACK-RESPONSES 403 Forbidden	148.204.45.101:80	148.204.45.143:60976	TCP

Fig. 5.1: Ataques de escaneo de puertos detectados por Snort

ICMP PING CyberKit 2.2 Windows	2008-04-10 00:19:45	201.62.136.110	148.204.20.49	ICMP
ICMP L3retriever Ping	2008-04-10 00:24:29	148.204.20.135	148.204.20.33	ICMP

Fig. 5.2: Direcciones IP de los host comprometidos

Aplicación de las reglas del modelo propuesto

Table 5.1: Reglas del modelo propuesto

Host	IP	Código Asignado	Tipo de Detección	Valor inicial en la red	Alerta	Tag
Host 1	10.0.0.2	111	111	000	NULL	Host libre de intrusión.
Host 2	10.0.0.3	111	111	000	NULL	Host libre de intrusión.
Host 3	10.0.0.4	007	222	000	333	Host fuera de Servicio
Host 4	10.0.0.5	111	111	000	NULL	Host libre de intrusión.
Host 5	10.0.0.6	111	111	000	NULL	Host libre de intrusión.
Host 6	10.0.0.15	007	333	333	333	Host anexado a la red ilegalmente
Host 7	10.0.0.8	111	111	000	NULL	Host libre de intrusión.
Host 8	10.0.0.9	007	7	000	333	Host con parámetros modificados
Host 9	10.0.0.10	111	111	000	NULL	Host libre de intrusión.
Host 10	10.0.0.11	111	111	000	NULL	Host libre de intrusión.

Caso 1

Host fuera de Servicio

El módulo de verificación solicita que se reporte un host específico a la central de información, si el host no responde al requerimiento se identifica al host con el código 222 (Host fuera de servicio). El sistema de aislamiento del módulo de alerta, genera un archivo en base a los hosts identificados con el código 007.

Posteriormente, envía por medio de una lista a los hosts etiquetados como: *Host 2, Host 4, Host 5, Host 6, Host 7, Host 9 y Host 10*; las direcciones IP que corresponden a los hosts comprometidos, en este caso: *Host 3, Host 6 y Host 8.* Se anexan a dicha lista las direcciones IP restantes del rango de red asignado a la misma. Para el ambiente de pruebas se propuso un rango de direcciones IP para 62 hosts, por lo que el conjunto de direcciones IP a anexar será de la 10.0.0.12 a la 10.0.0.62.

El host identificado como *Host 3 n*o recibe ninguna dirección IP a restringir, en apariencia puede comunicarse con cualquier hosts que forme parte de la red

Caso 2

Host anexado a la red ilegalmente 

El módulo de verificación detecta la presencia de un host no autorizado por el administrador de una red. El sistema de aislamiento del módulo de alerta genera un archivo en base a los hosts identificados con el código 007. Posteriormente, envía por medio de una lista a los hosts etiquetados como: *Host 2, Host 4, Host 5, Host 6, Host 7, Host 9 y Host 10*; las direcciones IP que corresponden a los hosts comprometidos, en este caso: *Host 3, Host 6 y Host 8.* Se anexa a dicha lista las direcciones IP restantes del rango de red asignado a la misma, para el ambiente de pruebas se propuso un rango de direcciones IP para 62 hosts, por lo que conjunto de direcciones IP a anexar será de la 10.0.12 a la 10.0.0.62.

El host identificado como *Host 6 n*o recibe ninguna dirección IP a restringir, en apariencia puede comunicarse con cualquier hosts que forme parte de la red

Caso 3

Host con parámetros modificados 

El módulo de detección identifica que un parámetro ha sido modificado y no existe una autorización previa por parte del administrador del sistema. Este parámetro por lo general, es modificado por un intruso para realizar un ataque de suplantación ante los dispositivos y hosts que integran una red.

El sistema de aislamiento del módulo de alerta, genera un archivo en base a los hosts identificados con el código 007. Posteriormente, envía por medio de una lista a los hosts etiquetados como: *Host 2, Host 4, Host 5, Host 6, Host 7, Host 9 y Host 10*; las direcciones IP que corresponden a los hosts comprometidos, en este caso: *Host 3, Host 6 y Host 8.* Se anexa a dicha lista las direcciones IP restantes del rango de red asignado a la misma, para el ambiente de pruebas se propuso un rango de direcciones IP para 62 hosts, por lo que el conjunto de direcciones IP a anexar será de la 10.0.12 a la 10.0.0.62.

El host identificado como *Host 8 n*o recibe ninguna dirección IP a restringir, en apariencia puede comunicarse con cualquier hosts que forme parte de la red

Caso 4

Host libre de intrusión 

El módulo de verificación recibe los parámetros enviados por el host y busca la correspondencia con los almacenados en su base de datos. Si el host es libre de intrusión, el proceso del sistema de aislamiento del módulo de alerta, genera un archivo en base a los hosts identificados con el código 007. Posteriormente, envía por medio de una lista a *Host 2, Host 4, Host 5, Host 6, Host 7, Host 9 y Host 10* (hosts libres de intrusión); las direcciones IP que corresponden a los hosts comprometidos, en este caso: *Host 3, Host 6*

y Host 8. Se anexa a dicha lista las direcciones IP restantes del rango de red asignado a la misma, para el ambiente de pruebas se propuso un rango de direcciones IP para 62 hosts, por lo que conjunto de direcciones IP a anexar será de la 10.0.12 a la 10.0.0.62.

5.5 Comportamiento de la herramienta Fira

Fira como se comentó en el capítulo 3 puede o no interacturar como un complemento a un IDS de tipo Red. En nuestro caso particular, al ser empleada como un complemento de Snort, ésta proporciona un propuesta de solución integral para las técnicas de evasión que pueden ser efectuadas por un atacante, las cuales pueden basarse en las vulnerabilidades que presentan los IDS de red (paquetes cifrados y desconocimiento de nuevos ataques), puesto que es empleada como un segundo filtro que puede explotar las bitácoras de Snort y bloquear un ataque específico hacia un host sensible de la intranet, aunado a la detección de anomalías en un host perteneciente a la red interna que ponga en riesgo la seguridad de la red entera, el cual será realizado por el módulo de verificación. Ahora bien, si no existiese en una red un IDS como primer mecanismo de defensa, Fira puede detectar anomalías en un host basándose en los parámetros que desee el administrador del sistema hallar o que le indiquen la presencia de un intruso mediante el esquema de causa-reacción y efectuar el bloqueo de comunicación entre hosts. La funcionalidad y comportamiento en ambos casos (como complemento de snort o de forma individual) es la misma, sólo qué cuando se emplea con un IDS de tipo Red la detección de intrusos se presenta una protección más integral al momento de conjuntar el análisis de la red externa con la red interna.

Fira en modo restrictivo

En primera instancia Fira se activa en modo restrictivo, de tal forma que el host establece únicamente la comunicación con la central de información y envía la información recolectada por un agente de sistema (Dirección IP y Dirección MAC).

Aplicación de reglas

Fira espera respuesta por parte del módulo de alerta para aplicar las restricciones, una vez que ésta recibe la lista de las direcciones IP que fueron detectadas como comprometidas (10.0.0.4, 10.0.0.15 y 10.0.0.9). Procede a la aplicación del sistema de aislamiento, en este caso son 3 direcciones IP, es importante destacar que el número de reglas a aplicar será basado en el número de direcciones IP a bloquear, por ejemplo, si fuese una sola dirección IP serían 2 reglas, si fuesen 7 equipos se aplicarían 14, y así, según corresponda. En la Fig. 5.3 se pueden ver las 6 reglas ejecutadas para aislar a dichos equipos, cabe recordar que son 6 debido a que la comunicación entre hosts es bidireccional, por lo que se bloquean los paquetes entrantes y salientes hacia esas direcciones.

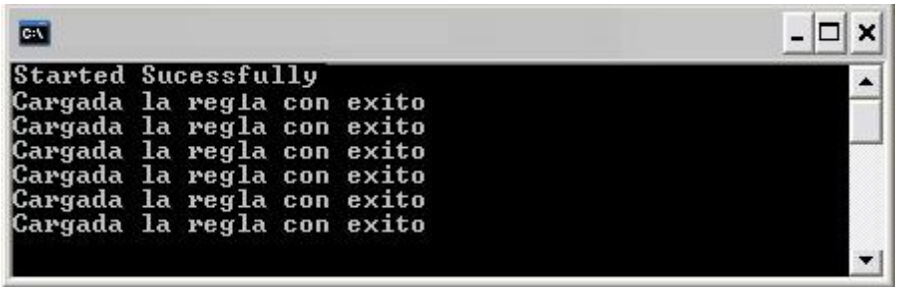


Fig. 5.3: Aislamiento indicado por la central

5.6 Métricas del modelo propuesto

Es necesario realizar más pruebas para conocer el número de falsos positivos y negativos que detecta por sí solo Snort, es decir, a diferencia de [109] y de [97] en los que se proponen adicionar un preprocesador para aumentar la eficiencia en su motor de análisis y así detectar mayores técnicas de intrusión en una red. El presente modelo es propuesto sobre un ambiente de producción en el que se confía en la inferencia producida a través de las reglas de Snort y en las reglas producidas por los parámetros que desee medir el propietario de un sistema.

En esta fase del proyecto, sólo es posible medir los falsos positivos y falsos negativos con base a las reglas de producción otorgadas por el propietario del sistema, y quedaría como un trabajo futuro demostrar la eficiencia del motor de análisis de Snort en un ambiente de producción, mediante técnicas de hacking ético (hackers de sombrero blanco). En dicha fase el sistema presenta un alto índice de falsos positivos al momento de evaluar la segunda regla de la central de información, debido a que sus criterios se basan en que si una sola regla no cumple lo establecido como un comportamiento anormal es identificada como intrusión. Esto es visible para los casos 1 y 2 (anteriormente descritos) puesto que la primera regla a evaluar es:

Si la DirIP !=coincidir se infiere que en esa primera parte todavía no hay intrusión

Sin embargo, en el caso 2 se modificó el parámetro de la dirección IP por error de asignación de direcciones, ya que el host marcaba duplicidad de dirección IP con otro host, y como no se tenía registrado en la base de conocimientos como una dirección IP conocida en la red, fue inferida como intrusión.

Esto nos lleva a decir que el proyecto operativamente restringirá lo que él detecte como anormal, pero requiere aumentar los parámetros a evaluar en el host que le hagan inferir que no se trata de una intrusión.

Los parámetros de evaluación seleccionados fueron intencionalmente propuestos para mostrar la funcionalidad del modelo, y que es necesario ir aumentando los parámetros a evaluar para considerar el grado de granularidad deseado por el administrador del

sistema.

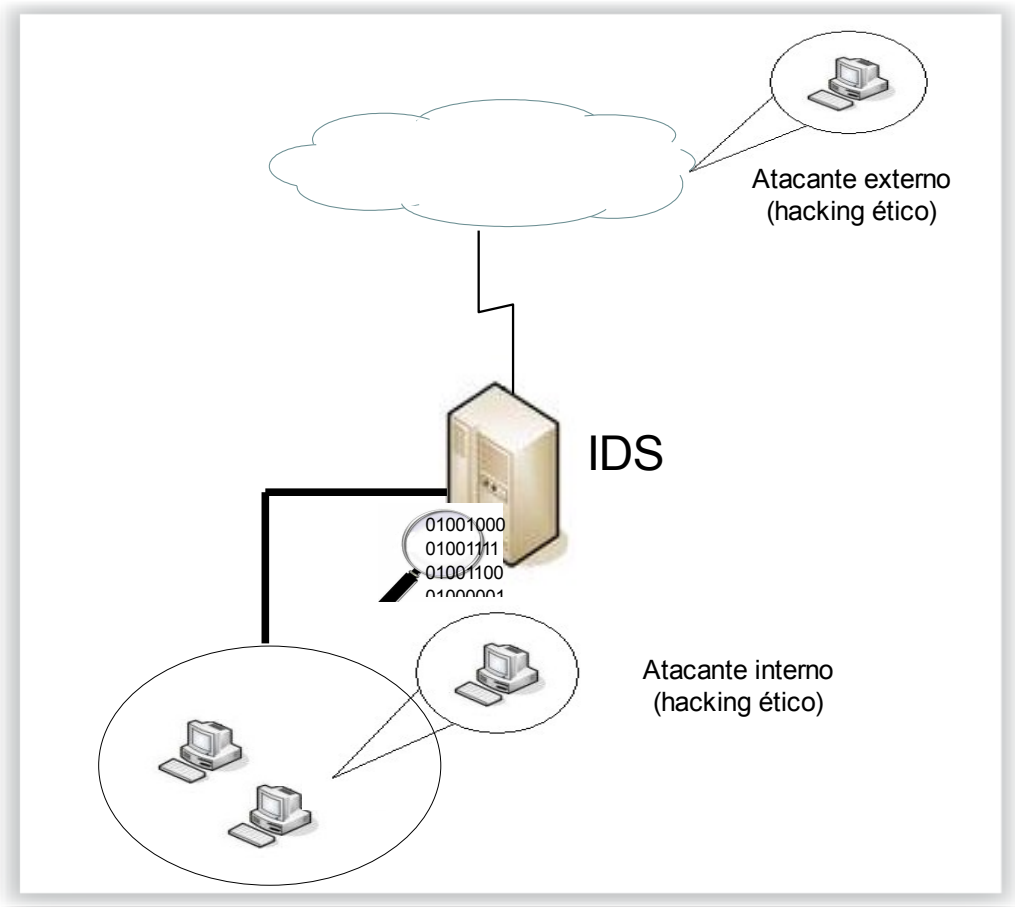


Fig. 5.4: Pruebas sugeridas para la segunda fase

Capítulo 6
Conclusiones

Resumen

En este capítulo se describen las conclusiones del análisis de las pruebas realizadas, las cuales se efectuaron conforme a los objetivos y alcances planteados durante el desarrollo del presente trabajo de tesis.

Objetivos

- Presentar en base a los objetivos y alcances planteados, las conclusiones del presente trabajo de tesis.
- Proponer futuros trabajos como seguimiento, complemento y mejora al modelo propuesto en el presente trabajo.

6.1 Conclusiones

En el presente trabajo de tesis se desarrolló un sistema de tipo red para la detección de intrusos, teniendo como marco de referencias el concepto de un sistema experto; es decir, un sistema basado en reglas lógicas definidas de acuerdo al expertise en el combate a acciones de protección, contra intentos de intrusiones malintencionadas hacia los hosts que conforman una red.

El modelo desarrollado contempla la detección del ataque con el consecuente aislamiento de los sistemas que han sido comprometidos por el atacante que evadió otros mecanismos de seguridad.

El sistema de detección de intrusos emplea la configuración de las reglas lógicas de Snort como elemento inicial, considerando que pese a la protección que brinda este recurso todavía puede ser evadido por un atacante por medio de la explotación de las debilidades del IDS tipo red; al no tener la capacidad de leer paquetes cifrados o que son desconocidos por su motor de análisis.

Para remediar esta situación se ha probado la adición de un segundo filtro de detección basado en reglas, denominado Fira, con el objetivo de operar mediante un mecanismo de causa-reacción para proteger a un host comprometido por un atacante, permitiendo con ello complementar adecuadamente la acción de detección de intrusos en la red. Cabe señalar que el diseño puede trabajar de manera conjunta con Snort para detectar intrusiones en la extranet y la intranet, así como de forma independiente para sólo revisar la intranet.

Durante las pruebas realizadas sobre el modelo y condiciones propuestas, se observó que es posible mediante las reglas de Snort reconocer intentos de intrusión hacia un hosts específico, por ejemplo, un servidor web, y mediante Fira se aplica la decisión de no permitir la comunicación con la dirección IP del atacante. El modelo propuesto puede crecer en su grado de granularidad, mediante la adición de más reglas tanto en Snort como en la parte de la central, las cuales son sujetas a consideración del propietario del sistema bajo el concepto de grado de granularidad, y sus métricas de detección dependerán estrechamente de las reglas previamente definidas por la experiencia. Asimismo, que Fira puede ser empleado o no como un complemento hacia los IDS de tipo Red, en este caso con Snort obteniendo con ello una protección más integral hacia una red interna, al momento de conjuntar el análisis de la red externa con la red interna.

6.2 Trabajos Futuros

A continuación se describen los posibles trabajos futuros que puede efectuarse, los cuales no son excluyentes a la creatividad e interés que se presente por parte del lector, sobre el trabajo propuesto:
Las propuestas son:

- Desarrollar un protocolo de comunicación entre los componentes del presente

modelo (diferente al propuesto), que involucre la protección de la integridad y la confidencialidad del intercambio de información, para que no pueda ser capturada por un intruso y debilitar el sistema de aislamiento propuesto.

- Desarrollar un sistema de redundancia y de balanceo de carga para redes distribuidas, bajo alguno de los siguientes tópicos:

  - Administración del modelo propuesto.
  - Control y administración de las bitácoras, así como las alertas en tiempo real.
  - Redundancia de las bases de datos que integran el motor de análisis de un sistema de detección.

- Sistemas espejo para la protección de los resultados obtenidos durante una detección.

- Realizar una retroalimentación hacia la base de datos durante el periodo del sistema de aislamiento propuesto, en base a las desviaciones presentadas durante la detección de una intrusión.

- Desarrollar un sistema de tolerancia a fallos para la comunicación de los ordenadores de una red con una central de información que no pertenece a su subred, debido a que su central de información directa ha sido víctima de un atacante, comprometiendo todo el segmento de la subred.

- Evaluar diversos mecanismos de cifrado que puedan aplicar para proteger los puntos señalados en el protocolo de comunicación propuesto, permitiendo el intercambio de información entre los componentes que integran el modelo propuesto.

- Demostrar la eficiencia del motor de análisis de Snort en un ambiente de producción.

- El host aplica las instrucciones en la herramienta fira.
- La central revisa que todos lo hosts registrados en su base de datos se hayan reportado. En caso de los faltantes, hace el requerimiento de envío de información:

    - Si los host faltantes no responden, los etiqueta con los códigos 222 y 007 respectivamente.

    - Si el host se reporta, evalúa la información recibida bajo los criterios del punto 4 y notifica el resultado hacia los hosts que integran la red.

El modelo ofrece:

- Flexibilidad en la configuración de sus parámetros de detección.
- Aislamiento del equipo comprometido durante el proceso de autoaprendizaje del motor de análisis de un IDS.
- Detección y prevención oportuna de la propagación de ataques de los hosts comprometidos en una red.

Referencias Bibiográficas

Artículos

[1] Anil Somayaji, Steven Hofmeyr & Stephanie Forrest. **Principles of a Computer Immune System.** *New Security Paradigms Workshop*, *pp. 75-82, ACM 1998.*

[2] John McHugh. **Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory.** *ACM Transactions on Information and System Security, Vol. 3, No. 4, November 2000, pp. 262–294.*

[3] Wenke Lee & Wei Fan. **Minning System Audit Data: Opportunities and Challenges.** *, SIGMOD Record, Vol. 30, No. 4, December 2001.*

[4] Salvatore J. Stolfo, Wenke Lee, Philip K., Chan, Wei Fan & Eleazar Eskin. **Data Mining-based Intrusion Detectors: An Overview of the Columbia IDS Project.** *SIGMOD Record, Vol. 30, No. 4, December 2001.*

[5] Daniel Barbard, Julia Couto, Sushil Jajodia, & Ningning Wu. **ADAM: A Testbed for Exploring the Use of Data Mining in Intrusion Detection.** *SIGMOD Record, Vol. 30, No. 4, December 2001.*

[6] Peng Ning, Sushil Jajodia & Xiaoyang Sean Wang. **Abstraction-Based Intrusion Detection In Distributed Environments.** *ACM Transactions on Information and System Security, Vol. 4, No. 4, November 2001, pp. 407–452.*

[7] Wenke Lee & Salvatore J. Stolfo. **A Framework for Constructing Features and Models for Intrusion Detection Systems.** *ACM Transactions on Information and System Security, Vol. 3, No. 4, November 2000, pp. 227–261*

[8] Yao-Tsung Lin, Shian-Shyong Tseng & Shun-Chieh Lin. **An Intrusion Detection Model Based Upon Intrusion Detection Markup Language (IDML).** *Journal Of Information Science and Engineering 17, pp. 899-919 (2001).*

[9] Carol Taylor & Jim Alves-Foss **An Empirical Analysis of NATE - Network Analysis of Anomalous Traffic Events.** *Proceedings of the 2002 workshop on New security paradigms NSPW '02, pp. 18 – 26. ISBN:1-58113-598-X.*

[10] Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang & S. Zhou. **Specification based Anomaly Detection: A New Approach for Detecting Network Intrusions.** *CCS'02, November 18–22, 2002. Washington, DC, USA ACM Proceedings of the 9th ACM conference on Computer and communications security ISBN:1-58113-612-9.*

[11] Klaus Julisch & Marc Dacier. **Mining Intrusion Detection Alarms for Actionable Knowledge.** *SIGKDD '02 Edmonton, Alberta, Canada. Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining pp. 366 – 375. ISBN:1-58113-567-X.*

[12] Peter Mell, Vincent Hu, Richard Lippmann, Josh Haines & Marc Zissman. **An Overview of Issues in Testing Intrusion Detection Systems.** *Defense Advanced Research Projects Agency under Air Contract F19628-00-C0002. (2002). Technical_ Report NIST IR 7007, National Institute of Standard and*

Technology.

[13] David Wagner & Paolo Soto. **Mimicry Attacks on HostBased Intrusion Detection Systems.** *CCS'02, November 18–22, 2002 / 2002 ACM 1581136129/02/0011.*

[14] Martin Botha, Rossouw Von Solms, Kent Perry, Edwin Loubser & George Yamoyany. **The Utilization of Artificial Intelligence in a Hybrid Intrusion Detection System.** *SAICSIT 2002, pp. 149 – 155.*

[15] Suresh N. Chari & Pau-Chen Cheng. **BlueBoX: A Policy-Driven, Host-Based Intrusion Detection System.** *ACM Transactions on Information and System Security, Vol. 6, No. 2, May 2003, pp. 173–200.*

[16] Klaus Julisch. **Clustering Intrusion Detection Alarms to Support Root Cause Analysis.** *ACM Transactions on Information and System Security, Vol. 6, No. 4, November 2003, pp. 443–471.*

[17] Robin Sommer & Vern Paxson. **Enhancing Byte-Level Network Intrusion Detection Signatures with Context.** *CCS'03, October 27–31, 2003, Washington, DC / 2003 ACM 1-58113-738-9/03/0010.*

[18] Daniel O. Hill. **Adding Intelligence and Action to Intrusion Detection.** *GSEC Practical Assignment. SANS Institute 2003, As part of the Information Security Reading Room.*

[19] Giovanni Vigna, Fredrik Valeur & Richard A. Kemmerer **Designing and Implementing a Family of Intrusion Detection Systems.** *ESEC/FSE'03, September 1–5, 2003, Helsinki, Finland. pp.* 88 – 97. 2003 *ACM 1-58113-743-5/03/0009.*

[20] M. Otey, S. Parthasarathy, A. Ghoting, G. Li, S. Narravula & D. Panda. **Towards NIC based Intrusion Detection.** *2003 ACM 1581137370/03/0008*.

[21] Giovanni Vigna, William Robertson & David Balzarotti. **Testing Network based Intrusion Detection Signatures Using Mutant Exploits.** *CCS'04, October 25–29, 2003, Washington, DC, USA. 2004 ACM 1581139616/04/0010.*

[22] Rong-Tai ,Chih-Hao Chen And Chia-Nan Kao & Nen-Fu Huang. **A Fast String-Matching Algorithm for Network Processor-Based Intrusion Detection System.** *ACM Transactions on Embedded Computing Systems, Vol. 3, No. 3, August 2004, pp. 614–633.*

[23] Shi Zhicai, Ji Zhenzhou & Hu Mingzeng. **A Novel Distributed Intrusion Detection Model Based on Mobile Agent.** *InfoSecu04, November 14-16, 2004, Pudong, Shanghai, China. 2004 ACM ISBN: 1-58113-955-1.*

[24] Holger Dreger, Vern Paxson, Robin Sommer & Anja Feldmann. **Operational Experiences with High-Volume Network Intrusion Detection.** *CCS'04, October 25-29, 2004, Washington, DC, USA. 2004 ACM 1-58113-961-6/04/0010.*

[25] Stefano Zanero & Sergio M. Savaresi. **Unsupervised learning techniques for an intrusion detection system.** *SAC'04 March 1417 2004, Nicosia, Cyprus. Copyright 2004 ACM 1581138121/03/04.*

[26] F Valeur, G Vigna, C Kruegel & R A. Kemmerer. **A comprehensive approach to intrusion detection alert correlation.** *IEEE Transactions On Dependable And*

Secure Computing, VOL. 1, NO. 3 pp. 146-169., July-September 2004.

[27] De-gang Yang,Yong-hong Chen & Chun-yan Hu. **A framework of cooperating Intrusion Detection based on Clustering analysis and expert system.** *InfoSecu04, November 14-16, 2004, Pudong, Shanghai, China. 2004 ACM.*

[28] M. M. Pillai, J. H.P. Eloff And H. S. Venter. **An Approach to Implement a Network Intrusion Detection System using Genetic Algorithms.** *Proceedings of SAICSIT 2004, pp. 221 – 228.*

[29] Anitha Nalluri & Dulal C. Kar. **A Web-Based System For Intrusion Detection.** Consortium for Computing Sciences in Colleges *CCSC: South Central Conference Volume 20, Issue 4 April 2005), pp. 274 – 281.*

[30] Ashlesha Joshi, Samuel T. King, George W. Dunlap, & Peter M. Chen. **Detecting Past and Present Intrusions through Vulnerability-Specific Predicates.** *SOSP'05, October 23–26, 2005, Brighton, United Kingdom. 2005 ACM 1-59593-079-5/05/0010.*

[31] Carlos Alfonso Pérez Rivera, Jaime Andres Britto Montoya & Gustavo Adolfo Isaza Echeverri. **Aplicación De Redes Neuronales Para La Detección De intrusos En Redes Y Sistemas De Información.** *Scientia et Technica Año XI, No 27, Abril 2005. UTP. ISSN 0122-1701.*

[32] Darren Mutz, Christopher Kruegel, William Robertson Giovanni Vigna & Richard A. Kemmerer. **Reverse Engineering of Network Signatures**. *AUSCERT 2005.*

[33] Stig Andersson, Andrew Clark & George Mohay. **Detecting Network-based Obfuscated Code Injection Attacks Using Sandboxing.** *AusCERT2005: Refereed R&D Stream.*

[34] Noria Foukia. **IDReAM: Intrusion Detection and Response executed with Agent Mobility Architecture and Implementation.** *AAMAS'05, July 2529, 2005, Utrecht, Netherlands. 2005 ACM 1595930949/05/0007*

[35] Richard A. Wasniowski. **Multi-Sensor Agent-Based Intrusion Detection System.** *Information Security Curriculum Development (InfoSecCD) Conference '05, September 23-24, 2005, Kennesaw, GA, USA. 2005 ACM 1-59593-261-5/05/0009.*

[36] Ill-Young Weon, Doo Heon Song & Chang-Hoon Lee. **Effective Intrusion Detection Model through the Combination of a Signature-based Intrusion Detection System and a Machine Learning-based Intrusion Detection System.** *Journal Of Information Science And Engineering 22, pp. 1447-1464 (2006).*

[37] Ashish Gehani, Surendar Chandra & Gershon Kedem. **Augmenting Storage with an Intrusion Response Primitive to Ensure the Security of Critical Data.** *ASIACCS'06, March 21-24, 2006, Taipei, Taiwán. 2006 ACM 1-59593-272-0/06/0003.*

[38] Moad Alhamaty , Ali Yazdian & Fathi Al-qadasi. **Intrusion Detection System Based On The Integrity of TCP Packet.** *Transactions On Engineering, Computing And Technology V11 February 2006 ISSN 1305-5313.*

[39] Prahlad Fogla Monirul Sharif Roberto Perdisci Oleg Kolesnikov & Wenke Lee. **Polymorphic Blending Attacks.** *15th USENIX Security Symposium Abstract Pp.*

241–256 of the Proceedings.

[40] Zonghua Zhang. **Adaptive Observation-Centric Anomaly-Based Intrusion Detection: Modeling, Analysis and Evaluation.** *GRP Report, Mar., 2006. Ph. D Thesis Japan Advanced Institute of Science and Technology.*

[41] Lin Tan, Brett Brotherton & Timothy Sherwood. **Bit-Split String-Matching Engines for Intrusion Detection and Prevention.** *ACM Transactions on Architecture and Code Optimization, Vol. 3, No. 1, March 2006, pp. 3–34.*

[42] Guofei Gu, Prahlad Fogla, David Dagon, Wenke Lee & Boris Skoric. **Measuring Intrusion Detection Capability: An Information-Theoretic Approach.** *ASIACCS '06, March 21-24, 2006, Taipei, Taiwán. 2006 ACM 1-59593-272-0/06/0003.*

[43] Yian Huang, & Wenke Lee. **A Cooperative Intrusion Detection System for Ad Hoc Networks.** *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks 2003 pp. 135 – 147 ISBN:1-58113-783-4*

[44] Geetha Ramachandran & Delbert Hart. **A P2P Intrusion Detection System based on Mobile Agents.** *ACME '04, April 2-3, 2004, Huntsville, Alabame, USA. 2004 ACM 1-58113-870-9/04/04.*

[45] Greg Vert Deborah A. Frincke Jesse C. McConnell. **A Visual Mathematical Model for Intrusion Detection.** *Proceedings of the 21st National Information Systems Security Conference, Crystal City, Arlington, VA, USA, October 5-8 1998, pp. 329-337.*

[46] Eleazar Eskin, Matthew Miller, Zhi-Da Zhong, George Yi, Wei-Ang Lee & Salvatore Stolfo. **Adaptive Model Generation for Intrusion Detection Systems**. *Proceedings of the ACMCCS Workshop on Intrusion Detection and Prevention, Athens, Greece, 2000.*

[47] Darren Mutz, Giovanni Vigna & Richard Kemmerer. **An Experience Developing an IDS Stimulator for the Black-Box Testing of Network Intrusion Detection Systems.** *Computer Security Applications Conference, 2003. Proceedings. 19th Annual Volume , Issue , 8-12 Dec. 2003 pp. 374 – 383.*

[48] Ashish Garg, Shambhu Upadhyaya & Kevin Kwiat. **Attack Simulation Management for Measuring, Detection Model Effectiveness**. *Management Workshop (SKM 2006), Polytechnic University, Brooklyn, NY, September 28-29, 2006.*

[49] Christopher Kruegel, Darren Mutz, William Robertson & Fredrik Valeur. **Bayesian Event Classification for Intrusion Detection**. *In 19th Annual Computer Security Applications Conference, Las Vegas, Nevada, December 08 - 12 2003.*

[50] Samuel Gorton & Terrence G. Champion. **Combining Evasion Techniques to Avoid Network Intrusion Detection Systems.** *A, Skaion Corporation. Skaion Research.*

[51] Peng Ning, Yun Cui & Douglas S. Reeves, **Constructing Attack Scenarios through Correlation of Intrusion Alerts.** *CCS'02, November 1822, 2002, Washington, DC, USA. 2002 ACM 1581136129/02/0011.*

[52] Jonathon T. Giffin, David Dagon, Somesh Jha, Wenke Lee & Barton P. Miller. **Environment-Sensitive Intrusion Detection.** *In Proceedings of the International*

Symposium on Recent Advances in Intrusion Detection (RAID), September 2005.

[53] Dipankar Dasgupta **Immunity-Based Intrusion Detection System: A General Framework.** *Proc. 22nd National Information Systems Security Conf. (NISSC), 1999*.

[54] Tarek Abbes, Adel Bouhoula, Michaël Rusinowitch. **On the Fly Pattern Matching For Intrusion Detection with Snort.** *Annales de Telecommunications, 2004. Vol. 59, Number: 9-10, pp. 941-967*

[55] David Wagner & Drew Dean. **Intrusion Detection via Static Análisis**. IEEE. Security and Privacy, 2001. S&P 2001. *Proceedings. 2001 IEEE Symposium on Volume , Issue , 2001 pp. 156 – 168.*

[56] Xiao-Bai Li. **A scalable decision tree system and its application in pattern recognition and intrusion detection.** *Elsevier Decision Support Systems 41 (2005) 112–130.*

[57] M. Saniee Abadeh, J. Habibi & C. Lucas. **Intrusion detection using a fuzzy genetics-based learning algorithm.** *Elsevier Journal of Network and Computer Applications 30 (2007) pp. 414–428.*

[58] Shun-Chieh Lin & Shian-Shyong Tseng. **Constructing detection knowledge for DDoS intrusion tolerance**. *Elsevier Expert Systems with Applications 27 (2004) pp. 379–390*

[59] Christos Douligeris & Aikaterini Mitrokotsa. **DDoS attacks and defense mechanisms: classification and state-of-the-art.** *Elsevier Computer Networks 44 (2004) pp. 643–666.*

[60] Ozgur Depren, Murat Topallar, Emin Anarim & M. Kemal Ciliz. **An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks.** *Elsevier Expert Systems with Applications 29 (2005) 713–722.*

[61] Dit-Yan Yeung & Yuxin Ding. **Host-based intrusion detection using dynamic and static behavioral models.** *Elsevier Pattern Recognition 36 (2003) pp. 229 – 243.*

[62] Tansel Özyer, Reda Alhajj & Ken Barker. **Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening.** *Journal of Network and Computer Applications 30 (2007) pp. 99–113.*

[63] Theuns Verwoerd & Ray Hunt. **Intrusion detection techniques and approaches**. *Elsevier Computer Communications 25 (2002) pp. 1356-1365.*

[64] Srinivas Mukkamala, Andrew H. Sung & Ajith Abraham. **Intrusion detection using an ensemble of intelligent paradigms.** *Elsevier Journal of Network and Computer Applications 28 (2005) pp. 167–182.*

[65] Wayne A. Cansen. **Intrusion detection with mobile agents.** *Elsevier Computer Communications 25 (2002) pp. 1392-1401.*

[66] W.M.P. van der Aalst & A.K.A. de Medeiros. **Process Mining and Security: Detecting Anomalous Process Executions and Checking Process Conformance.** *Elsevier Electronic Notes in Theoretical Computer Science 121*

(2005) pp. 3–21.

[67] Gordon Thomas Rohrmair & Gavin Lowe. **Using data-independence in the analysis of intrusion detection systems.** *Elsevier Theoretical Computer Science 340 (2005) pp. 82 – 101.*

[68] Ajith Abraham, Ravi Jain, Johnson Thomas & Sang Yong Han, **D-SCIDS: Distributed soft computing intrusion detection system.** *Journal of Network and Computer Applications 30 (2007) pages 81–98.*

[69] Tao Peng, Christopher Leckie, Kotagiri Ramamohanarao. **Information sharing for distributed intrusion detection systems.** J*ournal of Network and Computer Applications, Volume 30, Issue 3, August 2007, pp. 877-899.*

[70] Wu Yanga, Bin-Xing Fanga, Bo Liub & Hong-Li Zhang. **Intrusion detection system for high-speed network.** *Elsevier Computer Communications 27 (2004) pp. 1288–1294.*

[71] Chunlin Zhang, Ju Jiang & Mohamed Kamel. **Intrusion detection using hierarchical neural networks.** *Elsevier Pattern Recognition Letters 26 (2005) pp. 779–791.*

[72] Sandhya Peddabachigari, Ajith Abraham, Crina Grosan & Johnson Thomas. **Modeling intrusion detection system using hybrid intelligent systems.** *Elsevier Journal of Network and Computer Applications 30 (2007) 114–132.*

[73] Daejoon Joo, Taeho Hong & Ingoo Han. **The neural network models for IDS based on the asymmetric costs of false negative errors and false positive errors.** *Expert Systems with Applications 25 (2003) pp. 69–75.*

[74] B.A. Fessi, M. Hamdi, S. Benabdallah & N. Boudriga. **A decisional framework system for computer network intrusion detection.** *Elsevier European Journal of Operational Research 177 (2007) pp. 1824–1838.*

[75] Guisong Liu, Zhang Yi & Shangming Yang. **A hierarchical intrusion detection model based on the PCA neural networks.** *Neurocomputing, Volume 70, Issues 7-9, March 2007, pp. 1561-1568.*

[76] Zhuowei Li & Amitabha Das. **Analyzing and evaluating dynamics in stide performance for intrusion detection.** *Elsevier Knowledge-Based Systems 19 (2006) pp. 576–591.*

[77] Wei-Tsung Su, Ko-Ming Chang, Yau-Hwang Kuo. **eHIP: An energy-efficient hybrid intrusion prohibition system for cluster-based wireless sensor networks.** *Elsevier Computer Networks 51 (2007) pp. 1151–1168.*

[78] Yuehui Chen, Ajith Abraham & Bo Yang. **Feature selection and classification using flexible neural tree.** *Neurocomputing 70 (2006) pp. 305–313.*

[79] Chi-Ho Tsang, Yi Hong, Sam Kwong & HanliWang. **Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. Pattern Recognition Volume 40, Issue 9***, September 2007, pp. 2373-2391*

[80] Sanjay Rawat, Arun K. Pujari, & V. P. Gulati. **On the Use of Singular Value Decomposition for a Fast Intrusion Detection System.** *Electronic Notes in Theoretical Computer Science Vol 142 3 January 2006, pp. 215-228*

Proceedings of the First International Workshop on Views on Designing Complex Architectures (VODCA 2004)

[81] Georgios Portokalidis & Herbert Bos. **SweetBait: Zero-hour worm detection and containment using low- and high- interaction honeypots.** *Computer Networks 51 (2007) pp. 1256–1274.*

[82] Morton Swimmer. **Using the danger model of immune systems for distributed defense in modern data networks.** *Computer Networks*, *Volume 51, Issue 5*, *11 April 2007*, *pp. 1315-1333*

[83] Guisong Liu, ZhangYi & ShangmingYang. **A hierarchical intrusion detection model based on the PCA neural networks.** *Neurocomputing Volumen 70, Issues 7-9, March 2007, pp. 1561-1568 Advances in Computational Intelligence and Learning - 14th European Symposium on Artificial Neural Networks 2006, 14th European Symposium on Artificial Neural Networks 2006*

[84] Yun Wang, Inyoung Kim, Gaston Mbateng & Shih-Yieh Ho. **A latent class modeling approach to detect network intrusion.** *Elsevier Computer Communications 30 (2006) pp. 93–100*

[85] Lih-Chyau Wuu, Chi-Hsiang Hung & Sout-Fong Chen. **Building Intrusion Pattern Miner for Snort Network Intrusion Detection System.** *Journal of Systems and Software Volume 80, Issue 10. October 2007, pp. 1699-1715 Methodology of Security Engineering for Industrial Security Management Systems*

[86] Gabriel Maciá-Fernández, Jesús E. Díaz-Verdejo & Pedro García-Teodoro. **Evaluation of a low-rate DoS attack against iterative servers.** *Elsevier Computer Networks 51 (2007) pp. 1013–1030.*

[87] Deborah Frincke, Andreas Wespi & Diego Zamboni. **From intrusion detection to self-protection.** *Computer Networks, Volume 51, Issue 5, 11 April 2007, pp. 1233-1238.*

[88] Shuyuan Jin, Daniel SoYeung & Xizhao Wang. **Network intrusion detection in covariance feature space.** *Pattern Recognition Volume 40, Issue 8, August 2007, pp. 2185-2197 Part Special Issue on Visual Information Processing*

[89] Christopher J. Martinez, Wei-Ming Lin & Parimal Patel. **Optimal XOR hashing for non-uniformly distributed address lookup in computer networks. J***ournal of Network and Computer Applications Volume 30, Issue 4, November 2007, pp. 1397-1427 Special issue on Information technology*

[90] Lingyu Wang, Anyi Liu & Sushil Jajodia. **Using attack graphs for correlating, hypothesizing, and predicting intrusion alerts.** *Elsevier Computer Communications 29 (2006) pp. 2917–2933.*

[91] B.A. Fessi, M. Hamdi, S. Benabdallah & N. Boudriga**. A decisional framework system for computer network intrusion detection.** *Elsevier European Journal of Operational Research 177 (2007) pp. 1824–1838.*

[92] Paul D. Williams & Eugene H. Spafford. **CuPIDS: An exploration of highly focused, co-processor-based information system protection.** *Elsevier Computer Networks 51 (2007) pp. 1284–1298.*

[93] Angel Grediaga, Francisco Ibarra, Bernardo Ledesma & Francisco Brotons. **Utilización de redes neuronales para la detección de intrusos.** Primer Congreso Iberoamericano de Seguridad. CIBSI '02 Morelia, México.

[94] Pedro Pinacho D. & Ricardo Contreras A. **Una Propuesta de Sistema para Tratamiento de intrusos Inspirado en la Biología.** *Primer Congreso Iberoamericano de Seguridad Informática. Morelia, Mexico, Febrero 2002.*

[95] Dorothy Denning **An Intrusion detection model** IEEE 1986.

[96] Steven A. Hofmeyr & S. Forrest. **Architecture for an Artificial Immune System.** *Department of Computer Science, UNM, Albuquerque. Año 1999.*

[97] H.AbdelallahElhadj, H.M. Khelalfa & H.M. Kortebi **An Experimental Sniffer Detector: SnifferWall.** *SECI02 Septiembre 2002.*

[98] Hervé Debar, Marc Dacier & Andreas Wespi. **Towards a Taxonomy of Intrusion Detection Systems.** *Computer Networks: The International Journal of Computer and Telecommunications Networking Volume 31, Issue 9 (April 1999) Pages: 805 - 822 Year of Publication: 1999 ISSN:1389-1286 . Elsevier North-Holland, Inc*.

[99] **Intrusion Detection Systems Group Test** Group Test (Edition 2) *NSS Group Report*

[100] Brian Laing **Intrusion Detection Systems, How to Guide Implementing a Network Based Intrusion Detection System.** *Internet Security Systems 2000. http://www.snort.org/docs/iss-placement.pdf*

[101] Walter Baluja García, Marlene González García **Empleo de la tecnología IDS en la seguridad de redes.** *VI Seminario Iberoamericano de Seguridad en Tecnologías de Información y Comunicaciones. Cuba 2002. http://espejos.unesco.org.uy/simplac2002/vir.html*

[102] Alejandro Gramajo. **Introducción a conceptos de IDS y técnicas avanzadas con Snort.** *Jornadas de Software libre 2005. http://www.baicom.com/eventos/010905.pdf*

[103] R. Stiennon, M. Easley. **Intrusion Prevention Will Replace Intrusion Detection,** *Gartner Research. Agosto 2002.*

[104] Klaus-Peter Kossakowki, Julia Allen, Christopher Alberts, Cory Cohen, Gary Ford, Barbara Fraser, Eric Hayes, John Kochmar, Suresh Konda, William Wilson **Responding to Intrusions**. *February 1999. Security Improvement Module CMU/ SEI-SIM-006. Copyright 1999 by Carnegie Mellon University.*

[105] Stefan Axelsson **Intrusion Detection Systems: A Survey and Taxonomy**. *Department of Computer Engineering Chalmers University of Techonology Göteborg, Sweden. 14 Marzo 2000.*

[106] Diego González Gómez **sistemas de detección de Intrusiones Versión 1.01** *Última Revisión: Julio 2003. http://www.dgonzalez.net/pub/ids/html/*

[107] Andrés Felipe Arboleda, Charles Edward Bedón **Snort Diagrams for developers.** *Universidad del Cauca, Colombia. Abril 14 2005 Versión 0.2 alpha.*

[108] Urko Zurutuza Ortega. **Estado del Arte sistemas de detección de intrusos.** *Escuela Politécnica Superior de Mondragon Unibertsitatea, Octubre 2004.*

[109] Andrés Felipe Arboleda Torres, Charles Edgard Bedón Cortázar. **sistema de detección de intrusos Utilizando Inteligencia Artificial Anteproyecto de Trabajo de Grado.** *Universidad del Cauca Faculta de Ingeniería Electrónica y Telecomunicaciones Departamento de Sistemas. Noviembre 2004*

[110] Dominique Alessandri. **Tesis: Attack-Class-Based Análisis of Intrusion Detection Systems.** *Mayo 2004. University of Newcastle.*

[111] Emilio José Mira Alfaro **Tesis: Implantación de un sistema de detección de intrusos en la Universidad de Valencia.** *Ingeniería Informática, Universidad de Valencia.*

[112] Luis Miguel Díaz Vizcaíno Tesis Estudio Tecnológico. **sistemas de detección de intrusos**. *Universidad Carlos III de Madrid, Departamento de Ingeniería Telemática*

[113] Manual de snort http://www.snort.org/docs/

[114] Mike Fisk & George Varghese. **Fast Content-Based Packet Handling for Intrusion Detection**. *Computing, Communications and Networking Division. Los Alamos National Laboratory / Department of Computer Science and Engineering, University of California San Diego.*

[115] Siles Peláez Raul. **Análisis de seguridad de la familia de protocolos TCP/IP y sus servicios asociados**. *Editorial O'Really 1a. Edición Junio 2002*

Ligas

[116] Herramientas de Seguridad

http://www.securityfocus.com/infocus/1423
Herramientas de Seguridad.

http://www.linuxdata.com.ar/index.php?idmanual=75seguridad.htm&manuale=1
75 Herramientas de seguridad más usadas.

http://sectools.org/
100 *Herramientas de seguridad.*

http://www.seguridaddigital.info/index.php?option=com_content&task=view&id=64&Itemid=26
Herramientas de Detección.

www.psicofxp.com/forums/it-pro-servidores.243/125809-que-herramientas-usan-diariamente.html
Lista de herramientas de seguridad.

http://www.scs.carleton.ca/~dlwhyte/securitylinks.htm#ids
Herramientas para análisis de vulnerabilidades, sniffers, IDS, …

http://www.ausejo.net/seguridad/osstmm.htm
Herramientas de análisis de vulnerabilidades.

www.nessus.org
Herramienta de análisis de vulnerabilidades.

http://www.wikilearning.com/introduccion-wkccp-4913-1.htm
Tipos de escaneo con nmap.

[117] Sniffers

http://ezinearticles.com/?Detecting-Network-Sniffers&id=648410
Conceptos.

www.vilecha.com/Autodidactas/sniffers.html
Sniffers comerciales.

http://www.govannom.org/modules.php?name=News&file=article&sid=475
Switches y sniffers.

http://www.revistasic.com/revista42/agorarevista_42.htm
Técnicas de detección.

[118] Conceptos Generales

http://www.infosyssec.net/infosyssec/security/intdet1.htm
Conceptos generales sobre seguridad.

http://www.criptored.upm.es/paginas/docencia.htm
Documentos diversos sobre seguridad informática, criptografía, Tesis IDS,

http://www.securityfocus.com/infocus/1600
Define que es una anomalía y una intrusión.

http://personales.ya.com/casanchi/mat/difusa01.htm
Lógica difusa

http://ditec.um.es/laso/docs/tut-tcpip/
Lógica difusa

http://es.wikipedia.org/wiki/Miner%C3%ADa_de_datos
Minería de datos.

http://www.segu-info.com.ar/proteccion/vulnerar.htm
Rootkit.

http://www.criptos.com/article.php3?id_article=4?lang=es
Proxies

http://www.noticias3d.com/articulo.asp?idarticulo=174&pag=2
Hub, Switch.

http://www.govannom.org/modules.php?name=News&file=article&sid=475
Switches y sniffers.

http://www.securityfocus.com/infocus/1594
Implementación de Taps

http://webs.ono.com/alfonn/windump.htm
Tutorial windump y tcpdump

http://es.kioskea.net/contents/protect/firewall.php3
Concepto de Firewall

http://es.kioskea.net/contents/protect/dmz-cloisonnement.php3
Concepto de DMZ

http://www.iec.csic.es/criptonomicon/seguridad/servicio.html
servicio de seguridad

http://www.ranum.com/security/computer_security/editorials/deepinspect/
Inspección en seguridad, firewall

[119] Técnicas de Ataque

http://itpromexico.com.mx/paginas/articulos/arpspoof.htm
ARP Spoofing.

http://www.devjoker.com/contenidos/Articulos/45/Seguridad-en-Internet--SQL-Injections.aspx
SQL Injections.

[120] IDS

http://en.allexperts.com/e/i/in/intrusion-detection_system.htm
Conceptos sobre IDS

http://www.acm.org/crossroads/xrds2-4/intrus.html
Conceptos sobre IDS

http://ingenieria.ucaldas.edu.co/acad/tiki-index.php?page=intrusos
Conceptos sobre IDS

http://david.f.v.free.fr/ponencias/deteccion_de_intrusos/node4.html
Conceptos sobre IDS

http://www.linuxfocus.org/English/July2003/article294.shtml
Conceptos generales sobre IDS y HoneyPots

http://www.l0t3k.org/security/docs/ids/
Conceptos sobre IDS

http://www.stsc.hill.af.mil/crosstalk/2001/01/mchugh.html
Descripción de los IDS.

http://www.securityfocus.com/infocus/1514

Evolución de los IDS.

http://www.monografias.com/trabajos11/intru/intru.shtml
sistemas de detección de intrusos.

http://mural.uv.es/emial/informatica/html/IDS.html
sistemas de detección de intrusos.

http://www.securityfocus.com/infocus/1663
Análisis sobre detección por medio de firmas contra Análisis por protocolo.

http://www.securityfocus.com:80/infocus/1524
IDS basados en la detección por firmas.

http://www.prelude-ids.org/spip.php?article66
IDS Prelude.

http://www.securityfocus.com/infocus/1564
Importancia de Políticas de seguridad para los IDS.

http://www.securityfocus.com/infocus/1623
Puntos de evaluación para los IDS.

http://www.securityfocus.com/infocus/1754
Puntos de consideración para la implementación de un IDS.

[121] SNORT

http://www.sun.com/bigadmin/jsp/descFile.jsp?url=descAll/analyzing_snort_dat
Tutorial de BASE.

http://sourceforge.net/project/showfiles.php?group_id=103348
Obtención del software BASE.

http://media.blackhat.com/presentations/bh-usa-01/MartyRoesch/bh-usa-01-Marty-Roesch.ppt.
Algoritmo de búsqueda empleado por Snort.

http://en.wikipedia.org/wiki/Boyer%E2%80%93Moore%E2%80%93Horspool_algorithm
Boyer Moore Horspool.

http://www.glenmccl.com/bmh.htm
Boyer-Moore-Horspool.

http://www.codeproject.com/cs/algorithms/ahocorasick.asp
Aho-Corasick.

http://www.snort.org/docs/
Manual de snort

[122] Sistema Inmune

http://www.arrakis.es/~lluengo/inmunologia.html
Definición del Sistema inmunológico.

http://www.drscope.com/privados/pac/generales/inmunopatologia/celular.htm
Conceptos del Sistema Inmunológico.

http://cs.unm.edu/~immsec/html-imm/introduction.html
Conceptos sobre el Sistema Inmunológico Stephanie Forrest.

http://quest.nasa.gov/projects/flies/immune_S.html
Sistema Inmunológico.

http://www.niaid.nih.gov/final/immun/immun.htm
Sistema Inmunológico.

http://www.cancer.gov/espanol/cancer/entendiendo/sistema-inmunologico
Sistema Inmunológico.

[123] Evasión

http://www.securityfocus.com/infocus/1852
Técnicas de Evasión.

http://www.securityfocus.com/infocus/1577
Técnicas de evasion.

http://insecure.org/nmap/man/es/man-bypass-firewalls-ids.html
Evasión de cortafuegos-IDS.

http://www.penguin-soft.com/penguin/man/8/fragrouter.html
Fragrouter.

http://www.sahw.com/wp/archivos/2006/04/02/esteganografia-para-evitar-los-sistemas-de-deteccion-de-intrusos/
Esteganografía para evitar los sistemas de detección de intrusos.

http://digiassn.blogspot.com/2006/03/securityc-demonstration-of.html
Esteganografía para evitar los IDS.

http://www.imperva.com/application_defense_center/white_papers/sql_injection_signatures_evasion.html
Evasión por medio de SQL Injection.

http://www.securityfocus.com/infocus/1232
Evasión con Unicode (unicódigo).

http://loquefaltaba.com/documentacion/forense/anti.html
Técnicas Anti-Forenses.

[124] Hacking

http://www.segu-info.com.ar/amenazashumanas/definicionhacker.htm
Definición de Hacker.

http://en.wikipedia.org/wiki/Hacker_(computer_security)
Definición de hacker.

http://pilaryazmin.spaces.live.com/Blog/cns!C25854C9AB46B7C4!1970.entry
Definición de Hacker.

http://www.paralax.com.mx/antivirus/ar03-hackersyvirus.html
Definición de Hacker.

http://www.monografias.com/trabajos32/phreakers-hackers-tecnologia-matriz-cultural/phreakers-hackers-tecnologia-matriz-cultural.shtml
Hackers conceptos.

http://www.foromsn.com/index.php?Ver=Mensaje&Id=68460&VerEtiqueta=59
Clasificación de los hackers.

http://members.fortunecity.es/rebelcell/newbie.html
Clasificación de hackers.

http://informaticaiuris.tripod.com/id9.html
Clasificación de hackers: El Mundo Underground.

http://www.foromsn.com/index.php?Ver=Mensaje&Id=68460&VerEtiqueta=59
Clasificación de hackers.

http://platea.pntic.mec.es/~jdelucas/hacker.htm
Clasificación de hackers y técnicas de hacking.

http://www.seguridaddigital.info/index.php?option=com_content&task=view&id=30&Itemid=26
Clasificación de hackers

http://www.telepolis.com/cgi-bin/web/DISTRITODOCVIEW?url=/1578/doc/hacking/Personas.htm
Clasificación de Hackers

http://www.segu-info.com.ar/amenazashumanas/eticahackers.htm
La ética del Hacker.

http://www.linuxfocus.org/Castellano/March2003/article282.shtml
Tipos de ataques.

http://foro.el-hacker.com/index.php/topic,24573.0.html
Tipos de Ataque.

http://www.microsoft.com/spain/technet/security/midsizebusiness/topics/complianceandpolicies/socialengineeringthreats.mspx
Concepto de ingeniería social.

http://pacodebian.iespana.es/seguridad.html
Técnicas de hacking.

http://foro.dragonjar.us/index.php/topic,15536.0.html
Técnicas de Escaneo.

http://hackers.webcindario.com/manuales/hackingnt.php
Manuales Hacking.

http://www.hack-box.info/libro/index.html
Libro del underground.

http://www.somoslibres.org/modules.php?name=News&file=article&sid=307
Vulnerabilidades de Windows y Linux según el SANS.

[125] Protocolos

Http://html.rincondelvago.com/protocolos-de-comuniacion_1.html
Protocolos de comunicación.

[126] Propuestas de modelos de IDS

http://gost.isi.edu/cidf/
Common Intrusion Detection Framework

http://www.ietf.org/html.charters/OLD/idwg-charter.html
Intrusion Detection Exchange Format

[127] Microsoft

http://msdn.microsoft.com/en-us/library/aa504969.aspx
IPFilter

[128] NACS

http://www.itsecurity.com/features/essential-guide-nac-062308/
NACS

Gartner ID Number: G00166224, Magic Quadrant for Network Access Control, *March 2009.*

Glosario

Alerta	Notificación de la ocurrencia del valor de un umbral en igual o mayor tamaño.
Amenaza	Es la probabilidad sobre la ocurrencia de comprometer un sistema, la cual se mide de acuerdo a el factor de riesgo.
Análisis	Examinar con detenimiento evidencias o características, para buscar un determinado patrón que induzca al reconocimiento o descubrimiento de un algo.
Anti-Adware	Software para eliminar ventanas emergentes de tipo publicitario, de servicios o juegos.
Anti-Spyware	Software creado para la eliminación de programas espías que generalmente realizan la captura de contraseñas.
Appliance	Son sistemas de hardware que tienen implementado un sistema operativo en forma mínima, el cual está integrado por diversos módulos de seguridad perimetral como: cortafuegos, proxy, antivirus, anti spam, sistemas de detección de intrusos, control de navegación de páginas web. Es decir, en un sólo dispositivo se pueden tener instalados diferentes aplicaciones de seguridad.
Ataque	Es la acción que se efectúa hacia el objetivo deseado, una vez que se ha recaudado toda la información, como: horarios,

	contraseñas procedimientos, técnicas de distracción y evasión de seguridad,así como las herramientas de penetración a emplear.
Auditoría	Revisión interna o externa en busca de vulnerabilidades o intrusiones, que se realiza de manera minuciosa de un sistema.
Autenticación	Es un mecanismo de seguridad otorgado para verificar la identificación del sistema o persona que accesa a un sistema.
Bitácora	Registro en tiempo real de los sucesos de un sistema, indicando fecha, hora, evento, usuario, etc.
Conexión	Canal de comunicación que se establece entre dos dispositivos.
Confidencialidad	Propiedad o atributo que adquiere la información localizada en el interior de un documento o en un sistema, de forma física o electrónica, la cual sólo puede ser acceada por su(s) autor(es) o personas autorizadas por el propietario.
Contraataque	Estrategia ofensiva contra un ataque.
Cortafuegos (Firewall)	Implementado en hardware o software. Se emplea para filtrar la información que viene desde el exterior hacia el interior y viceversa. De ahí su función de proteger y sólo aceptar el tráfico bueno hacia la intranet. Ese control es ejercido a través de reglas que indican las acciones a seguir.
Detección	Buscar indicios de una característica y corroborarlos.
Disponibilidad	Es el garantizar el ofrecimiento de servicios o recursos de un sistema.
Dispositivo	Aparato creado con un cierto fin. Ejemplo: switch.
DMZ	Zona Desmilitarizada. Configuración en la que se colocan los servidores después de un cortafuegos para filtrar el tráfico entre dos redes: Red Externa y Red Interna.
Escenario	Conjunto de variables que sirven para describir y ejemplificar de manera clara las causas que originan un problema; buscando con ello inducir a una posible solución o investigación del tema.

Evasión	Es la técnica que se emplea para no ser detectado durante una intrusión, puede efectuarse a través de la fragmentación de paquetes, uso de unicode, etc.
Exploit	Es explotar y/o aprovechar una vulnerabilidad o error en la programación de un sistema, con fines de intrusión por medio de la creación de malware. Su significado: Exploit (Explótalo) contracción de Explot it.
Firma	Se le llama firma a la característica que autentica de manera única su origen o comportamiento.
Hackear (Hacking)	Acción que lleva a cabo un intruso al irrumpir en un sistema de cómputo.
Herramienta de Seguridad	Instrumento de software o hardware que permite realizar tareas de protección y revisión de vulnerabilidades en un host o en una red.
Host	Es una computador, también conocido como equipo(forma corta, de equipo de cómputo) que está conectado en una red.
IDS	sistema de detección de intrusos que es generalmente indentificado por sus siglas en inglés como IDS. Es un mecanismo de seguridad empleado para detectar la presencia de un intruso en un sistema de red o de host.
Intrusión	Se refiere a la indicación que se ejecuta en el momento que alguien o algo se encuentra dentro de un sistema.
intruso	Puede ser una persona de forma física o en vía remota, que accesa de manera ilícita a los sistemas. Puede ser interno o externo, cuyo objetivo es irrumpir en nuestra privacidad con motivos de aprendizaje, desafío y/o venganza. Un ejemplo de esto, puede ser un programa que no figura en la lista de programas conocidos y autorizados por las políticas de una empresa.
IPS	Es un sistema de detección de intrusos al que se le adicionó la funcionalidad de un cortafuegos para filtrar paquetes y tomar acciones de filtrado sobre los paquetes de red que sean identificados como peligrosos.

Honeypot	También conocido como tarro de miel, es un sistema de detección de intrusos que se utiliza como señuelo para conocer las tácticas y técnicas de un atacante que vulneró un sistema.
Malware (Software Malicioso)	Programa que se crea con fines maliciosos: Denegación de Servicios, bloqueo de procesos, modificación de archivos, control remoto, infiltración para perpetrar ataques a otros sitios.
Modelo	Representación gráfica de un concepto o sistema, destacando sus características y debilidades.
Modo Promiscuo	Es la deshabilitación de un filtro interno de la tarjeta de red, que le permite escuchar el tráfico de la red, aunque este no vaya dirigido exáctamente hacia esta tarjeta.
No Repudio	Es obtener al responsable de una acción determinada, sin que este pueda negar el acto. Trátese de un proceso, programa o persona.
Política de Seguridad	Directiva que se asume después de evaluar el riesgo sobre un sistema.
Protocolo	Normas que se establece para efectuar una conexión e intercambio de información entre dos dispositivos iguales o de distinta estructura.
Proxy	Es una configuración que se efectúa dentro de una red, para controlar la salida a internet de los usuarios de una intranet. Se le conoce como intermediario, puesto que él se encarga de recibir las peticiones y llevarlas a cabo con una sóla salida.
Proxy Inverso	Es la configuración inversa, aquí el Proxy recibe las peticiones de Internet y las canaliza hacia la intranet. Es decir, aquí el Proxy da el frente de protección de los servidores y luego canaliza las peticiones hacia los mismos.
Puerto	Número asociado a un punto de acceso para enviar o recibir datos dentro de una conexión.
Red Externa	Red que se encuentra fuera de la red interna (internet).

Red Interna	Es el conjunto de máquinas y dispositivos que pertenecen al interior de una organización (intranet).
Riesgo	Valor que se le asigna a una aplicación o equipo de cómputo de acuerdo a su nivel de vulnerabilidad y al daño que este puede sufrir ante un ataque.
Rootkit	Programa que se instala de forma oculta en el interior de un sistema para trabajar en forma pasiva(captura de información, intercepción de procesos,...) o activa (ejecución local, escalar privilegios (root o administrador), control remoto,.trampolín para atacar otras redes,...), el cual es un poco difícil de detectar, debido a la encriptación de sus rutinas.
Seguridad	Área de Informática, que estudia y pone en práctica diversas técnicas de detección y defenza; con la finalidad de proteger la información sensible de los intrusos.
Servidor	Es una computadora con gran capacidad en memoria y almacenamiento, que comparte recursos y/o pone a disponibilidad servicios hacia otros equipos o hosts.
Spyware	Software que se instala para realizar captura silenciosa de información sensible (contraseñas) y esta sea enviada posteriormente hacia el atacante.
TCP/IP	Es una suite de protocolos, que se ubica dentro de las capas del modelo OSI o en su propio modelo (Modelo de TCP/IP). Donde cada uno, tiene una función específica de acuerdo a la capa en la que se encuentra.
Tráfico de Red	Es el número de datos o paquetes que fluyen simultáneamente a través de la red, el cual es medidoen bits/seg.
Underground	Comunidad secreta de los hackers, donde son clasificados de acuerdo a sus diferentes niveles de aprendizaje y motivaciones.
Virus	Software malicioso, creado con el objetivo de propagarse y reproducirse a si mismo, buscando obtener control del equipo y/ o modificar la información contenida en un sistema. Puede ser residente en memoria, ser polimórfico (técnica para ocultarse), presentar mutaciones (variaciones del mismo virus), accionarse en forma manual o remota.

Vulnerabilidad	Son las debilidades que se pueden encontrar en un sistema para que este se vea comprometido

B

Implementación de un IDS

La implementación en una red es simple pero no trivial, depende de la carga de la red misma(volumen de información). En algunos casos pensando en redes muy pequeñas, es posible emplear hubs; sin embargo, estos en la mayoría de las redes han sido reemplazados por switches.

B.1 Hub

Los Hubs por su diseño permiten el dominio de broadcast, es decir todos los equipos escuchan los paquetes pero sólo el indicado es quien lo recibe. Esto permite que un IDS al adicionarse a este dispositivo, capture la información que fluye a través de la red de manera transparente y sin necesidad de configurar algo extra en el dispositivo (hub). El IDS se observa como un equipo más en la red, el cual escucha, captura, analiza y genera las alertas correspondientes (Fig. B.1).

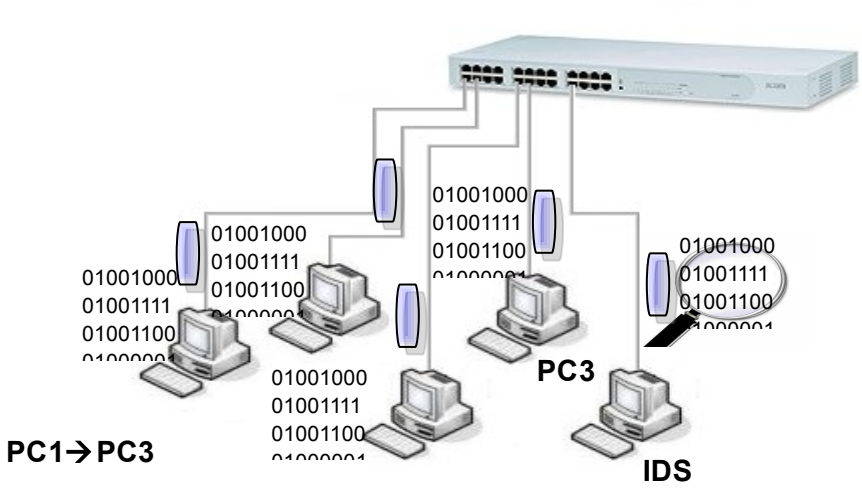


Fig. B.1: Configuración con HUB

Existen redes que emplean switches para el control de tráfico, debido su característica que tienen de aislar el dominio de broadcast. Es decir, a diferencia del hub el tráfico es

dirigido sólo al equipo correspondiente, sin que los demás equipos escuchen. Sin embargo, en este tipo de red los IDS se ven afectados para la recolección de la información que circula por esta, ya que estos requieren escuchar todo el tráfico que circula dentro de la red para realizar la detección de una posible intrusión y en el switch al no existir el dominio de broadcast, se requiere realizar algunas configuraciones extras para poder subsanar este inconveniente (Fig. B.2).

Las soluciones propuestas hasta ahora, para su implementación son:

- Configuración de un puerto SPAN (Switch Port Analyser) o Port Mirror (nombre que se le da de acuerdo a la marca del switch)
- Utilización de TAP (Test Access Point)

B.2 SPAN Port

Se emplea en casos en donde se requiera monitorear por corto tiempo una red, esto se debe, a que el puerto SPAN (Sensor Port Analyser) está configurado para escuchar el tráfico de la red que pasa a través del switch y del router. Sin embargo existen limitaciones en cuanto al equipo, tales como la pérdida de paquetes y el rendimiento del mismo.

Al existir un sólo puerto SPAN por switch, sólo se monitorea parte de la red. Lo cual no es suficiente para la detección de intrusos, por lo que, es necesario colocar un switch conocido como Switch Top Layer, para realizar el balanceo de carga entre switches y no degradar el tráfico de la red, ni el servicio del switch.

Ventajas

- Su implementación no requiere de conocimiento profundo.
- Se pueden interconectar con otros dispositivos de red para enviar comandos de finalización de sesiones y reconfiguraciones si fuese necesario.

Desventajas

- Si es una red intregrada por varios switches se requiere un Swich Top Layer (switch de capa 7), para el balanceo de carga de la red.
- Es susceptible a ataques de DoS, puesto que este tipo de configuración tiene una dirección fija que lo hace vulnerable.
- La captura de paquetes masiva, degrada el rendimiento del equipo, debido a esto varios paquetes son descartados.

- Defasamiento en la visualización de paquetes producido por retardos, así como modificación en la estructura del paquete.
- No permite observar todo el tráfico de la red, si algún paquete viene con errores lo descarta.
- No puede visualizar errores en capa 1 y capa 2.
- Requiere configurar un puerto por cada VLAN o puerto a monitorear.
- Desafortunadamente, sólo puede ver la información en un sentido. Lo que presenta una gran desventaja para el análisis global del tráfico de red.

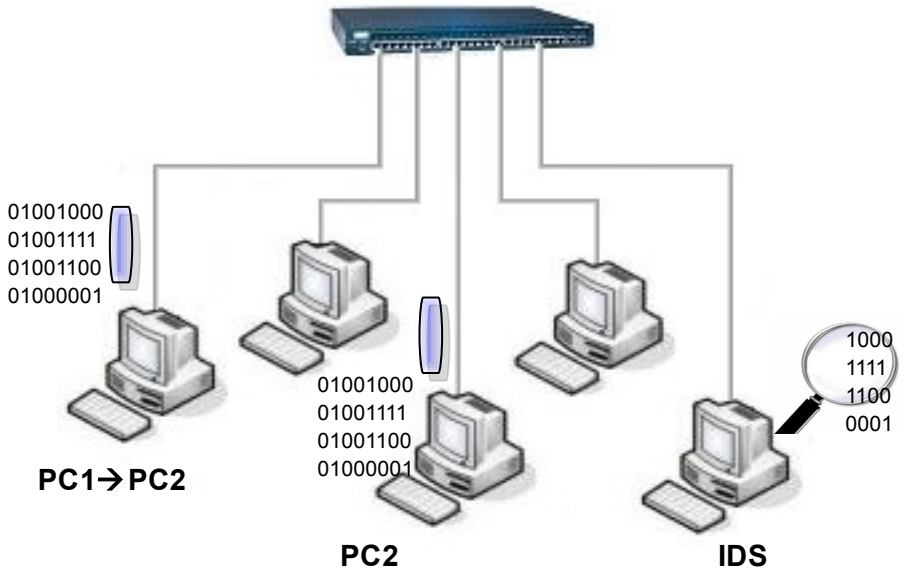


Fig. B.2: Configuración usando SPAN Port

B.3 TAP

Es un dispositivo pasivo, es decir, que no altera el tráfico de la red. Esta es una gran ventaja, en el caso de falta de suministro de energía eléctrica, el flujo del tráfico de red no se ve interrumpido en su servicio.

La finalidad de los TAPs es evitar que existan pérdidas de paquetes por la degradación de rendimiento que se presenta en los switches aparte de mezclar el tráfico de entrada y salida (cuestión que no sucede en el SPAN Port, que sólo captura el tráfico entrante) [100].

Se conecta entre el switch y el router, permitiendo la captura de la información entre ambos dispositivos, junto con una derivación en la que se coloca el IDS.

Los TAPs pueden ser de dos tipos:

- Cable (fabricados para conexión directa entre la red y el IDS).
- Dispositivos de entrada ethernet (interconexión switch-router-IDS).

Ventajas

- Es casi imperceptible en la red.
- Es un dispositivo pasivo, sólo escucha y no modifica el flujo del tráfico de red.
- No requiere de una dirección fija, lo que le permite no ser susceptible a ataques de DoS.
- No degrada el rendimiento del switch, del enrutador, ni entre ellos mismos.
- Tiene mayor capacidad de captura de información, evitando así la pérdida de paquetes.

Desventajas

- Requieren de un mayor conocimiento tecnológico para su implementación.
- Es relativamente cara la solución, puesto que se tiene que comprar un equipo adicional al switch.

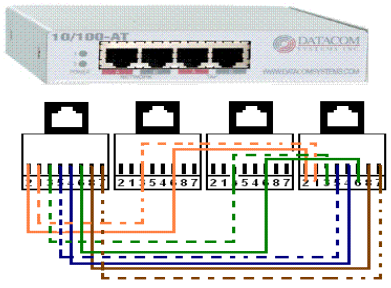
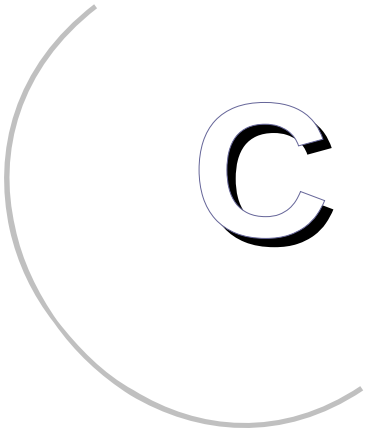


Fig. B.3: Configuración de un TAP



Clasificación del underground

- *Wannabe (Want to be)*
  Es una persona que realmente desea convertirse en hacker , por lo que empezará aprendiendo y comprendiendo el área a la que desea adentrarse para conseguir su objetivo.

- *Newbie*
  Es el novato que comienza el camino de convertirse en un hacker, quién demuestra su interés a los hackers sobre algún tema específico a través de preguntas coherentes y profundas sobre códigos propios que muestran el conocimiento que se han ido adquiriendo y el potencial que tiene para ser un hacker.

- *Hacker*
  Es el nivel que adquiere una persona sobre un cierto ámbito que le permite denominarse como experto en esa materia. El hacker adquiere la madurez necesaria en sus conocimientos para no dañar a un sistema, sino para mejorarlo.

- *Copyhacker*
  Es una persona que tiene conocimientos tecnológicos sobre un tema, pero evita esfuerzos y utiliza las herramientas que creó un hacker para su beneficio personal.

- *Cracker*
  Es el antagonista de los hackers. Se pueden dedicar a dos actividades: la primera es romper los candados que limitan el uso de software comercial, y la segunda, a irrumpir abruptamente en los sistemas, sin tener escrúpulo alguna. Según los hackers son personas que no han madurado en este último aspecto, por lo que, cualquier hazaña la realizan por venganza o

beneficio personal; sin comprender que ser un hacker no es destruir ni dañar un sistema, sino aprender de él para mejorarlo.

- *Coders*
  Son personas con amplios conocimientos de programación, que se dedica a la creación de códigos maliciosos.

- *Carders*
  Realizan fraudes financieros con tarjetas electrónicas, se dedican a clonar o a interceptar transacciones financieras realizadas con tarjetas electrónicas a través de la web.

- *Phreaker*
  Emplea sus conocimientos en telefonía, principalmente celular. Obtiene servicios telefónicos gratuitos, como pueden ser: transferencias de saldos, mensajes vía SMS, tarjetas de prepago, entre otros.

- *Blogkers*
  Son personas que modifican las páginas web de un sitio de internet.

- *Wizard*
  Se especializa generalmente en un determinado dispositivo, conoce a profundidad su diseño, las partes que lo constituyen y su operación. Es reconocido como una eminencia sobre un dispositivo en particular.

- *Gurú*
  Es el máximo nivel de conocimiento que puede ser adquirido en esta jerarquía. No sólo es un experto más, sino conoce a profundidad varios temas casi como si él fuese el creador de los mismos.

- *Nerd*
  Persona que estudia sobre diversos temas desarrollando conocimiento sobre éstos, pero no es un especialista en ellos.

- *Geek*
  Es una persona que se hace experto sólo por hobbie en un ámbito particular.

- *Lamer*
  Desea convertirse en un hacker, pero no tiene los conocimientos sobre un tema específico, ni el interés por desarrollarlo. Algunas veces se excusa en decir que está incapacitado para comprender términos que le ayuden a adquirir el conocimiento.

- *Scriptkiddie*
  Persona que cree ser un hacker por el hecho de utilizar herramientas creadas por los hackers, que dependiendo de la metología descrita por la

herramienta puede irrumpir en un sistema. No posee conocimientos sobre técnicas de intrusión, ni protocolos, ni sistemas operativos y/o lenguajes de programación. Son considerados por los hackers como personas arrogantes, que no saben lo que hacen, ni lo que dicen sobre un tema en particular. Estas personas pueden ser sorprendidos y capturados en cualquier momento, por no tener los conocimientos para no delatarse durante su irrupción a un sistema.

- *Pirata Informático*

  Es una personas que adquiere ilícitamente los recursos que tienen otros, a través del uso de herramientas creadas para ese fin. Los recursos a los que éste puede tener acceso son: música, videos, software de diversos tipos, libros electrónicos (ebooks), etc...

- *Bucanero*

  Es una persona que no tiene interés por aprender, sino por obtener un beneficio financiero. Este tipo de persona adquiere las claves de activación para el software de tipo comercial o versiones de software previamente activadas para su uso ilimitado por los crackers, así como las herramientas que desarrollan para vulnerar sistemas, trafica con ellas y obtiene una remuneración económica por ellos.

- *Programador vudú*

  Es una persona que utiliza códigos que fueron desarrollados por terceros, sin comprender su funcionamiento, es decir, no se esfuerza en profundizar como funciona la técnica expuesta u optimizarla. Sólo lo aplica en su beneficio.