SISTEMAS DETECTORES DE INTRUSOS Y ANÁLISIS DE FUNCIONAMIENTO DEL PROYECTO DE CÓDIGO ABIERTO SNORT

Jairo Alberto Rojas
Ingeniero en Telecomunicaciones, estudiante de la Especialización en Teleinformática en la Universidad Distrital "Francisco José de Caldas".
Rojas7j@hotmail.com

Hector C. Manta
Ingeniero Electrónico, MSc. en Ciencias de la Información y las Comunicaciones de la Universidad Distrital "Francisco José de Caldas", CCNA, CCNP., asistente de la Decanatura de la Facultad de Ingenierías
hcmantac@udistrital.edu.co

Tipo: Artículo de reflexión

Fecha de Recepción: Marzo 31 de 2011
Fecha de Aceptación: Mayo 10 de 2011

INTRUSION DETECTION SYSTEMS (IDS) AND OPERATION ANALYSIS OF THE SNORT OPEN-CODE PROJECT

Abstract
In this article presents the analysis of SNORT operation, which is a software tool intended for Network Intrusion Detection Systems (NIDS). A detailed explanation of rule generation and some tests carried out is also presented, where packets were captured according to both an existing rule and generated rule using SNORT. The course of development to achieve these results follows a brief introduction to the known types of information attacks and intrusions as well as a description of the classification types of Intrusion Detection Systems (IDS), focusing on the network-oriented type.

Keywords: SNORT, sniffer, IDS, snort Rules, types of vulnerabilities.

Resumen
En este artículo se presenta el análisis de funcionamiento del SNORT, herramienta de software libre para la detección de intrusiones en red, así como una detallada explicación de la creación de reglas y algunas pruebas realizadas, obteniendo capturas de los paquetes de red por medio del software con una regla existente y otra creada. Lo anterior siguiendo un desarrollo en el cual se trata una breve introducción a los tipos conocidos de ataques informáticos e intrusiones, así como una descripción de la clasificación de los tipos de sistemas detectores de intrusos (IDS), centrándonos en el tipo orientado a red.

Palabras clave: SNORT, niffer, IDS, reglas de Snort, ipos de vulnerabilidades.

1. INTRODUCCIÓN

La primera red de computadores que se desarrollo tiene sus inicios en el año de 1958 cuando la agencia gubernamental ARPA de Estados Unidos se organiza en respuesta a los desafíos tecnológicos y militares de Rusia durante la guerra fría, su objetivo era crear una estructura resistente a fallos [1]. En la actualidad los sistemas informáticos

y las redes de computadores, son una realidad que gracias a la creciente popularidad del internet, ya no son tema de unos pocos y al contrario de eso es casi una obligación el manejo y la familiarización con este tipo de tecnologías, en el mundo actual.

Este documento muestra un análisis de funcionamiento del sistema detector de intrusos de código abierto, Snort, utilizado por profesionales de seguridad informática en todo el mundo. Provee un listado de reglas (que son la base de su funcionamiento) actualizadas cada cierto tiempo y revisadas por profesionales de todo el mundo, lo que garantiza la confiabilidad en la herramienta.

2. ATAQUES INFORMATICOS E INTRUSIONES

2.1 Tipos de Intrusiones

Una intrusión no necesariamente consiste en un acceso no autorizado a una maquina, sino que también puede ser una negación de servicio. Se pueden producir de varias formas [2]:

- **Externos:** Atacantes que acceden a los sistemas desde internet.
- **Internos:** Atacantes dentro de la misma red.
- **Abusos de Recursos:** Son usuarios autorizados del sistema que intentan ganar privilegios adicionales para los cuales no están autorizados, y/o usuarios que hacen mal uso de los privilegios o recursos que se le han asignado.

2.2 Tipos de Vulnerabilidades

En el año de 1972 Robert Thomas Morris desarrolla el que se considera el primer virus informático denominado Creeper, que transmitía y desplegaba el mensaje en la pantalla del computador: "Soy una enredadera, atrápame si puedes". Con el fin de poder eliminar este problema se crea un antivirus llamado Reaper que buscaba el virus y lo eliminaba.

En el año de 1974 aparece una nueva aplicación maliciosa experta en reproducirse denominado Rabbit.

En 1975 se da origen al primer troyano de la historia denominado Animal[3].

Categoría de las vulnerabilidades:

- Alta: El atacante puede obtener privilegios de administrador de forma remota.
- Media: El atacante puede ganar acceso remoto sin privilegios.
- Baja: Permite revelar algún tipo de información con el fin de desarrollar ataques de riesgo mucho mayor.
- Informativas: Revela información acerca del funcionamiento de la red.

2.2.1 Troyanos

Los troyanos son programas que se introducen en un sistema para obtener información o posibilitar acceso a través de red sin necesidad de algún tipo de permiso, estos programas se ejecutan sin presentar alguna sospecha ocultando una actividad maliciosa.

Funcionamiento:

- Infección: A diferencia de los virus, los troyanos no pueden reproducirse, la infección se realiza ejecutando el fichero.
- Asegurando posiciones: Cuando el fichero es ejecutado este se encarga de modificar el sistema para que se ejecute cada vez que se inicialice el equipo.
- Comunicación con el atacante: Si el troyano se ha ejecutado al conectarse el usuario a internet enviara un mensaje al atacante con el número de puerto que ha dejado libre para trabajar y la IP; el contenido de estos datos depende de la configuración del troyano.

2.2.2 Spoofing

En seguridad informática Spoofing se refiere al uso de técnicas de suplantación de

identidad generalmente con usos maliciosos o de investigación. Existen varios tipos [4]:

- **IP Spoofing:** Consiste en sustituir la dirección IP origen de un paquete TCP/IP por otra dirección IP a la cual se desea suplantar.
- **ARP Spoofing:** Se trata de la construcción de tramas de solicitud y respuesta ARP modificadas con el objetivo de falsear la tabla ARP de una víctima y forzarla a que envié los paquetes a un host atacante.
- **DNS Spoofing:** Se basa en la falsificación de una relación "Nombre de dominio-IP" ante una consulta de resolución de nombre, es decir, se refiere a resolver con una dirección IP falsa un cierto nombre DNS o viceversa.
- **Web Spoofing:** Enruta la conexión de una víctima a través de una página falsa hacia otras páginas web con el objetivo de obtener información de dicha victima (páginas vistas, información de formularios, contraseñas, etc.).
- **Mail Spoofing:** Esta técnica es usada con mucha frecuencia para el envió de e-mails falsos y para SPAM.

2.2.3 Ataques de denegación de servicio

Un ataque de denegación de servicio, también llamado ataque **DoS** (de las siglas en ingles Denial of Service), es una ofensiva a un sistema de computadoras o red que causa que un recurso sea inaccesible a los usuarios legítimos. Normalmente provoca la pérdida de la conectividad de la red por el consumo del ancho de banda de la red de la víctima o sobrecarga de los recursos computacionales del sistema de la víctima [5].

También existen los denominados ataques de amplificación, que son más conocidos como ataque distribuido de denegación de servicio (DDoS), están basados en la instalación de troyanos en muchas computadoras que pueden estar localizadas en diferentes puntos de todo el mundo. El invasor consigue controlar todas estas computadoras zombis a su gusto, generando mucho tráfico hacia una maquina o red objetivo, y de este modo consigue colapsar el sistema víctima.

2.3 Métodos de ataque

Un ataque DoS puede ser realizado de muchas formas, pero básicamente consisten en:

- Consumo de recursos computacionales, tales como ancho de banda, espacio de disco, o tiempo de procesador.
- Alteración de información de configuración, tales como información de rutas de encaminamiento.
- Alteración de información de estado, tales como interrupción de sesiones TCP (TCP reset).
- Interrupción de componentes físicos de red.
- Obstrucción de medios de comunicación entre usuarios de un servicio y la víctima, de manera que ya no puedan comunicarse adecuadamente.

2.3.1 Inundación SYN (SYN Flood)

La inundación SYN se efectúa cuando se envía un flujo de paquetes TCP/SYN, solicitándole al servidor establecer una conexión, el equipo cliente o atacante desecha o nunca contesta los paquetes TCP/SYN+ACK que envía como respuesta el servidor. De este modo el servidor tiene que mantener múltiples conexiones "semi-abiertas" hasta que se vaya produciendo el "timeout". Estos intentos de conexión consumen recursos en el servidor y limitan el número de conexiones que se pueden hacer, reduciendo la disponibilidad del servidor para responder peticiones legítimas de conexión [6].

2.3.2 Inundación ICMP (ICMP Flood)

Es una técnica DoS que pretende agotar el ancho de banda de la víctima. Consiste en enviar gran cantidad de paquetes ICMP del tipo Echo request (ping) de mayor tamaño, y de esta forma la víctima ha de responder

con paquetes ICMP Echo reply (ping). Sí la capacidad de procesamiento del atacante es mucho mayor, la víctima no podrá manejar el tráfico generado [7].

2.3.3 Smurf

El Smurf amplifica considerablemente los efectos de un ataque ICMP. Existen tres partes en un Smurf: el agresor, el intermediario y la victima.

En el ataque Smurf, el atacante dirige paquetes ICMP de tipo ping a una dirección IP de broadcast, usando como dirección IP origen, la dirección de la víctima, se espera que los equipos conectados a dicha red respondan a la petición, usando Echo reply, a la maquina origen (victima). Todas estas respuestas son dirigidas a la victima intentando colapsar sus recursos de red [8].

2.3.4 Inundación UDP (UDP Flood)

Básicamente este ataque consiste en generar grandes cantidades de paquetes UDP contra la víctima elegida. Debido a la naturaleza sin conexión del protocolo UDP, este tipo de ataques suele venir acompañado de IP Spoofing.

Es usual dirigir este tipo de agresiones contra maquinas que ejecutan el servicio Echo, elevando el tamaño de las solicitudes [9].

3. SISTEMAS DE DETECCION DE INTRUSOS

Detectar intrusos es un proceso que requiere del monitoreo y el análisis, de los paquetes de trafico que circulan por la red, para de esta forma brindar confiabilidad, integridad, disponibilidad y confidencialidad a un sistema y a la información soportada por este.

La mayoría de los sistemas de detección de intrusos IDS, operan bajo los mismos parámetros de detección. Capturan los paquetes que transitan por la red y realizan la comparación del tráfico capturado con unos patrones que se encuentran definidos en las reglas o firmas establecidas para la detección de anomalías.

"Existen diferentes niveles de protección dentro de una red bien administrada que sirven para dar soporte ante una intrusión:

- **Preventivos:** Firewalls y todo tipo de sistemas de filtrado (anti-spam, antivirus, contenido de sitios Web). Los sistemas de redundancia también se consideran preventivos.
- **De detección:** IDS y otros sistemas de vigilancia en tiempo real, que informaran en el momento de un incidente, pero no tratarán de evitarlo.
- **Correctivos:** Respaldos de todo tipo de controles utilizados para restaurar un sistema afectado por un incidente de seguridad" [10].

3.1 Tipos de IDS

Los sistemas de detección de intrusos se clasifican en tres grupos: Host IDS (HIDS) que funcionan en una sola maquina, Network IDS (NIDS) que funcionan en un segmento de red, y los sistemas de prevención de intrusiones (IPS) que se encargan de prevenir una intrusión, reaccionando activamente ante alguna sospecha de acceso no autorizado.

3.1.1 IDS basados en host (HIDS)

Este tipo de IDS se instalan en una sola maquina, se encargan de recolectar y validar la información contenida en el host, como logs, llamadas al sistema desde el espacio de usuario al núcleo del sistema operativo, verificación de carga de la CPU y de la memoria. Fueron los primeros IDS desarrollados e implementados y permiten ver el resultado de un intento de ataque, al igual que pueden acceder directamente y monitorizar los ficheros de datos y procesos del sistema atacado.

Ventajas:

- Los IDS basados en host, al tener la ca-

pacidad de monitorizar eventos locales a un host, pueden detectar que no pueden ser vistos por un IDS basado en red.
- Pueden operar en un entorno en el cual el tráfico de red viaja cifrado, ya que la fuente de información es analizada antes de que los datos sean cifrados en el host origen y/o después de que los datos sean descifrados en el host destino.

Desventajas:

- Son más costosos de administrar, ya que deben ser gestionados y configurados en cada host monitorizado.
- Si la estación de análisis se encuentra dentro del host monitorizado, el IDS puede ser deshabilitado si un ataque logra tener éxito sobre la maquina.
- No son adecuados para detectar ataques a toda una red, puesto que el HIDS solo ve a aquellos paquetes de red enviados a él.
- Pueden ser deshabilitados por ciertos ataques DoS [11].

3.1.2 IDS basado en red (NIDS)

La mayoría de los sistemas de detección de intrusos están basados en red. Estos IDS detectan un ataque por medio de la captura y análisis de los paquetes que viajan por un segmento de red. Un IDS basado en red puede monitorizar el tráfico generado por varios hosts de la red, para de este modo protegerlos de posibles ataques contra su integridad.

Están conformados por un tipo de sensores que normalmente se ubican en distintos lugares o puntos de la red, y su función es la de capturar todo el trafico de red que circule por dichos puntos para luego ser analizada en busca de trafico malicioso. La mayoría de estos sensores son diseñados para funcionar en modo oculto, razón por la cual es más difícil que un atacante advierta su presencia así como su localización [12].

Ventajas:

- Pueden llegar a monitorizar una red grande, desde que se cumpla con la capacidad necesaria para analizar todo el tráfico.
- Como funcionan en la red, no consumen recursos de host, y su impacto es mínimo, siendo prácticamente dispositivos pasivos e invisibles, que no interfieren en el funcionamiento habitual de la misma.

Desventajas:

- Pueden presentar inconvenientes en redes demasiado grandes o con mucho tráfico, en donde le es imposible capturar todo el tráfico de la red y puede dejar pasar ataques lanzados en esos momentos.
- No analizan información cifrada.
- Solo pueden detectar ataques cuando un ataque fue lanzado, pero no pueden saber si el ataque tuvo éxito.

3.1.3 Sistema de prevención de intrusiones (IPS)

Un IPS es un sistema de prevención y protección que ejerce control permanente sobre el tráfico en el interior de las redes tanto en el lado de usuario como del servidor; permitiendo detectar cualquier ataque aun en su fase más temprana.

Su principal característica es que tiene la habilidad de bloquear o detener las intrusiones, mientras el tráfico legítimo no sufre ninguna restricción, de esta forma la red puede seguir operando en normalidad.

4. ALTERNATIVAS DE ANALIZADORES DE VULNERABILIDADES

A continuación se relacionan cinco de los más populares analizadores de red, en la tabla 1, se muestra una comparación entre las herramientas mencionadas a continuación.

4.1 Snort

Es un sistema de código abierto de detección de intrusiones de red, que realiza un barrido de los puertos en tiempo real, Snort está disponible bajo licencia GPL, y funciona bajo plataformas Windows y UNIX/Linux. Dispone de una gran cantidad de filtros o patrones ya predefinidos, así como actualizaciones constantes ante casos de ataques, barridos o vulnerabilidades que vayan siendo detectadas a través de los distintos boletines de seguridad [13].

4.2 Wireshark

Es una versión actualizada de Ethereal. Es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones para desarrollo de software y protocolos, y como una herramienta didáctica para el análisis de tráfico. Es importante tener presente que WireShark no es un IDS (Instrusion Detection System) ya que no es capaz de generar una alerta cuando se presentan casos anómalos en la red [14].

4.3 Smartdefense

Bloquea los ataques por tipo y clase, utilizando la tecnología Stateful Inspection (revisión de estados) patentada por Check Point y una consola única centralizada que muestra información de los ataques en tiempo real, así como la detección, bloqueo, registro, auditoría y alertas de ataques. Información de intentos de acceso en tiempo real: Por medio del centro de información de accesos no autorizados en línea de Check Point, los administradores de la seguridad pueden obtener información actualizada de cada tipo de ataque [15].

4.4 Symantec network security 7100

Proporciona una solución de seguridad de red con despliegue simplificado, administración centralizada y soporte completo. Detecta agresiones sin tener conocimiento de una vulnerabilidad conocida o divulgada [16].

4.5 Sensor Cisco secure IDS 4250

El sensor Cisco Secure IDS 4250 es un "dispositivo" hardware de seguridad, que detecta la actividad no autorizada que la atraviesa, como por ejemplo ataques por parte de hackers, mediante el análisis del tráfico en tiempo real, y permite a los usuarios responder con rapidez a las amenazas de seguridad. Cuando se detecta una actividad no autorizada, el sensor puede enviar alarmas a la consola de administración con detalles de la actividad y puede controlar otros sistemas, como los routers, para terminar las sesiones no autorizadas [17].

4.6 Análisis comparativo

En resumen se puede ver que SNORT y Wireshark son multiplataforma pero como se mencionó anteriormente Wireshark no es IDS, es decir solo captura trafico pero no lo analiza, adicionalmente estas dos aplicaciones son software libre y se basan en la librería PCAP.

SmartDefense y Symantec Network Security 7100 son herramientas propietarias, lo que indica que se debe pagar una licencia para utilizarlas, y no se sabe mucho sobre el desarrollo de las mismas ya que por razones obvias de negocio no son reveladas y solo funcionan en plataformas Microsoft Windows.

El Sensor Cisco Secure IDS 4250 es uno de los mejores IDS existentes y a diferencia de las otras herramientas, es hardware que se coloca en una porción de la network, lo que lo convierte en independiente del sistema operativo que se maneje y le da mayor rapidez en la captura y análisis de flujos de información, ya que concentra todos sus recursos para este fin.

En la tabla1 se puede ver una comparativa entre las herramientas mencionadas.

Tabla 1. Tabla comparativa entre las herramientas mencionadas.

CARACTERISTICAS	SNORT	WIRESHARK	SMARTDEFENSE	SYMANTEC NETWORK SECURITY 7100	SENSOR CISCO SECURE IDS 4250
Es un IDS (*Instrusion Detection System*)	x		x	x	x
Es un Sniffer de paquetes	x	x	x	x	x
Detector de intrusos basado en red.	x		x	x	x
Funciona bajo plataformas Windows y UNIX/Linux.	x	x			x
Utilización de firmas de ataques	x		x	x	x
Utiliza libpcap o WinPcap como librería base	x	x			
Modo de operación promiscuo como en modo no promiscuo.	x	x	x	x	x(únicamente promiscuo)
Filtra los paquetes que cumplan con un criterio definido previamente	x	x	x	x	x
Captura de los protocolos más importantes	x	x	x	x	x
Información de ataques en tiempo real	x		x	x	x

5. FUNCIONAMIENTO DE IDS ORIENTADO A RED SNORT

SNORT puede funcionar como Sniffer o como Detector de Intrusos basado en Red y que está desarrollado bajo extensiones PCAP, que es la base del funcionamiento de conocidos proyectos libres como son: Wireshark (antiguo Ethereal), Tcpdump/Windump, entre otros.

En la configuración NIDS el Snort se basa en una amplia cantidad de firmas, las cuales permiten analizar el tráfico que circule por la red y que se convierten en su principal característica, ya que de ahí depende el éxito de la detección.

En la Fig. 1 se puede ver el diagrama de bloques básicos de funcionamiento del SNORT, el cual se basa en tres etapas: captura de paquetes, análisis de paquetes y registro de vulnerabilidades en la base de datos.

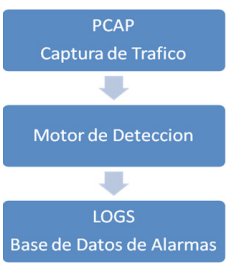

Fig. 1. Diagrama de bloques básico, funcionamiento.

5.1 Librería PCAP

PCAP es una librería open source escrita en C, que es conocida como Libpcap para su versión en Unix/Linux y como Winpcap para su versión en MS Windows, y es la encargada de capturar los paquetes en la capa de red.

Para usar la librería, es necesario descargarla e instalarla, de su respectivo sitio web: http://www.tcpdump.org para Unix/Linux y http://www.windump.org para MS Windows.

La librería PCAP es la encargada de capturar el flujo que circula y ofrece una amplia gama de opciones como son: captura de interfaces de red, configuración de tarjeta en modo normal o modo promiscuo, rearma los paquetes IP fragmentados obteniendo nuevamente el paquete original, filtrado de tráfico dependiendo el protocolo u otras opciones y funciones para el manejo de archivos (trafico offline) [18], [19].

5.2 Motor de Detección

Es la parte encargada de detectar si alguna actividad de intrusión existe en algún paquete. Para realizar este propósito el motor de detección emplea una serie de reglas, las cuales se comparan con los paquetes y en caso de tener alguna coincidencia se toma una acción determinada, que puede ser una alerta.

El comportamiento del motor de detección es una parte muy importante y afecta de manera directa el desempeño total del sistema, dependiendo de la potencia del servidor y de la cantidad de reglas. Si el tráfico de red es muy alto es posible que se dejen pasar paquetes sin analizar [20].

5.2.1 Reglas Snort

Se pueden dividir en dos secciones lógicas: cabecera y opciones, como se ve en la Fig. 2.

Fig. 2. Estructura de una regla [21].

La cabecera contiene la acción de la regla, protocolo, IPs, máscaras de red y puertos.

La sección opciones contiene los mensajes y la información necesaria para la decisión a tomar por parte de la alerta en forma de opciones.

- Cabecera

Contiene 6 campos en los que se especifican condiciones que se deben cumplir para que la regla sea efectiva. A continuación se muestra formato de la cabecera, figura 3 [22].

Fig. 3. Estructura cabecera de una regla [21].

Acción: "La acción de la regla" dice que hacer cuando un paquete coincide con los criterios incluidos. Hay cinco acciones por defecto y son:

- **Alert.** Genera una alerta mediante el método de alerta seleccionado.
- **Log.** Registra el paquete.
- **Pass.** Ignora el paquete.
- **Actívate.** Alerta y luego activa una regla dinámica.
- **Dynamic.** Permanece inactivo hasta que se activa por una regla active y entonces actúa como una regla log.

Protocolo: Se manejan cuatro protocolos: TCP, UDP, IP e ICMP.
Red_Origen: Es la dirección de red desde la cual se genero el paquete, si se define ANY, quiere decir que no importa la red origen.
Puerto_Origen: Es el puerto en el cual se genero el paquete, si se define ANY, quiere decir que puede ser cualquier puerto.
Red_Destino: Es el destino al cual llega el paquete, es decir, la red atacada. Define la dirección de red del usuario.
Puerto_Destino: Es el puerto al cual llega el paquete, si se define ANY, quiere decir que puede ser cualquier puerto.

La Fig.4 muestra un ejemplo de cabecera de una regla:

Acción	Protocolo	Red Origen	Puerto Origen	Dirección	Red Destino	Puerto Destino
alert	tcp	$EXTERNAL_NET	any	→	$HOME_NET	53

Fig. 4. Ejemplo cabecera Regla de Snort [21].

Indica que en caso de haber coincidencia con el paquete la regla generara una alerta, analizará el protocolo ICMP, en paquetes provenientes de cualquier red externa y que vayan dirigidos hacia cualquier equipo de la red.

- Opciones

Constituyen el núcleo del motor de detección de intrusos de Snort, que combina facilidad de uso con poder y flexibilidad. Todas las opciones de las reglas están separadas entre sí por un punto y coma (;). Las palabras clave se separan de sus argumentos con dos puntos (:). A continuación se muestra el formato de las opciones [23].

(flags: PA; content: "|E8C0FFFFFF|/bin"; activates: 1; msg: "IMAP buffer overflow!";)

Hay cuatro categorías principales en las opciones.

- **general:** Proveen información de la regla, pero no tienen ningún efecto en la detección. En la tabla 2 se relacionan las opciones generales.

Tabla 2. Opciones generales de las reglas de Snort.

Keyword	Descripción
msg	Es el mensaje que se mostrara al producir una alerta
reference	Permite incluir referencias de la intrusión, para saber de que tipo es
gid	Es usada para identificar que parte de Snort genera el evento en una regla particular
sid	Es un identificador único de la regla usada por Snort
rev	Es usada para saber el número de revisiones de la regla
classtype	Se utiliza para categorizar el tipo de ataque en una regla
priority	Indica el nivel de prioridad de la regla utilizado por Snort
metadata	Permite incluir información adicional sobre la regla

- **payload:** Estas opciones buscan concordancias dentro de la carga útil del paquete y pueden ser interrelacionados. En la tabla 3 se relacionan algunas opciones payload.

Tabla 3. Opciones Payload, de las reglas de Snort.

Keyword	Descripción
content	Permite establecer el contenido especifico que se busca en la carga útil de un paquete
rawbytes	Permite ver el paquete ignorando cualquier decodificación hecha por el preprocesador
depth	Especifica los primeros bytes de la carga útil en que se deben buscar coincidencias con el patrón content
offset	Especifica donde empieza la búsqueda en un patrón witlin
distance	Especifica a partir de qué punto debe empezar a buscar el patrón de coincidencia del paquete
within	Es un modificador de contenido que asegura que la mayoría de N bytes coinciden con el patrón content
unicontent	Indica una dirección URL dentro del paquete
isdataat	Especifica que la carga útil tiene datos en una ubicación especifica
pcre	Permite a las reglas tener compatibilidad con Per1
byte_test	Revisa un byte especificado por un valor
byte_jump	Permite leer una porción de datos y saltar una posición adelante
ftpbounce	Detecta ataques FTP
asnl	Decodifica una porción del paquete y busca síntomas maliciosas
cvs	Detecta cadenas de texto invalidas

- **non-payload:** Buscan que no hayan concordancias dentro de la carga útil del paquete. En la tabla 4 se relacionan la mayoría de opciones Non-payload.

Tabla 4. Opciones Non-Payload, de las reglas de Snort.

Keyword	Descripción
fragoffset	Permite comparar el campo offset del fragmento IP con un valor decimal
til	Es usado para verificar el valor TTl
tos	Permite verificar el campo IP TOS para un valor especifico
id	Permite verificar el campo IP ID para un valor especifico
ipopts	Se usa para verificar que una opción especifica esté presente en el paquete
fragbits	Se usa para verificar si los bits de fragmentación y reservado están activados en la cabecera IP
dsize	Se usa para confirmar el tamaño de la carga útil del paquete
flow	Permite aplicar la regla solo en ciertas direcciones del flujo de trafico
flowbits	Permite saber el estado a raves de la sesión del protocolo de transporte
seq	Permite especificar el número de secuencia TCP
ack	Permite especificar el número ACK TCP
windows	Permite especificar el tamaño de la ventana TCP
itype	Se usa para especificar el campo type de ICMP
icode	Se usa para especificar el campo code de ICMP
icmp_id	Se usa para especificar el campo Id de ICMP
icmp_seq	Se usa para especificar el campo Seq de ICMP
ip_proto	Se usa para verificar la cabecera del protocolo IP
sameip	Verifica si la dirección IP origen es igual a la dirección IP destino

- **post-detection:** Especifican lo que sucederá luego de que una alerta fue detectada. En la tabla 5 se relacionan la mayoría de opciones post-detection.

Tabla 5. Opciones Post-detection, de las reglas de Snort.

Keyword	Descripción
logto	Registra todos los paquetes en un archivo de registro
session	Está diseñado para extraer los datos de usuario de la sesión TCP
resp	Se usa para cerrar las sesiones cuando se activa una alerta
react	Permite la posibilidad de reaccionar ante una coincidencia de trafico con una regla, por medio del cierre de la conexión y el envió de un aviso
tag	Permite a las reglas registrar mas paquetes que simplemente el que lanza la alerta
activates	Permite especificar una regla para agregar, cuando una red especifica produce el evento
activated_by	Permite especificar dinámicamente una regla cuando una regla active es lanzada
count	Permite especificar cuantos paquetes han sido habilitados después que la regla es activada. Debe usarse con activated_by

Seguidamente a manera de ejemplo se muestra las “opciones ”de una regla:

(msg:"ICMP Echo Reply"; icode:0; itype:0; classtype:misc-activity; sid:408; rev:5;)

El código anterior corresponde a una regla para detectar un ping realizado a alguno de los equipos de la red. Genera el mensaje **"ICMP Echo Reply"**, cuando el paquete tiene **icode:0; e itype:0;, classtype:misc-activity;** corresponde a la categorización de Snort para el tipo de ataque correspondiente a un Ping, **sid:408**; se refiere al número de identificación de la regla en Snort y **rev:5**; es el numero de revisiones que ha tenido la regla (5 para este caso).

6. PRUEBAS Y RESULTADOS

Para la realización de pruebas se implementó el montaje mostrado en la Fig. 5, en donde el PC1 tiene instalado el Snort y será el encargado de analizar el tráfico que circula por la red.

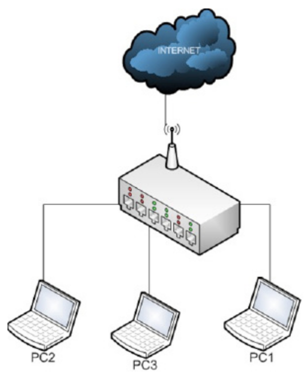

Fig. 5. Esquema de red montado para pruebas.

Las IP de los equipos fueron:
PC1: 192.168.0.99
PC2:192.168.0.100
PC3:192.168.0.101

6.1 Prueba 1

Se implementa una regla para que se genere alertas ante la aparición de tráfico tipo ping. Específicamente se realiza un ping del PC2 al PC3, los resultados arrojados son las capturas mostradas en la Fig. 6.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

06/27-07:56:40.846116 00:1D:D9:E3:EB:CD -> C4:17:FE:08:A7:E8 type:0x800 len:0x4A

192.168.0.100 -> 192.168.0.101 ICMP TTL:128 TOS:0x0 ID:1360 IpLen:20 DgmLen:60
Type:8  Code:0  ID:768   Seq:768  ECHO
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70  abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69  qrstuvwabcdefghi

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

06/27-07:56:40.904026 C4:17:FE:08:A7:E8 -> 00:1D:D9:E3:EB:CD type:0x800 len:0x4A

192.168.0.101 -> 192.168.0.100 ICMP TTL:64 TOS:0x0 ID:49514 IpLen:20 DgmLen:60
Type:0  Code:0  ID:768  Seq:768  ECHO REPLY
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70  abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69  qrstuvwabcdefghi

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

06/27-07:56:41.847068 00:1D:D9:E3:EB:CD -> C4:17:FE:08:A7:E8 type:0x800 len:0x4A

192.168.0.100 -> 192.168.0.101 ICMP TTL:128 TOS:0x0 ID:1362 IpLen:20 DgmLen:60
Type:8  Code:0  ID:768   Seq:1024  ECHO
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70  abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69  qrstuvwabcdefghi

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Fig. 6. Captura obtenida con el Snort.

Verificando en el archivo de capturas de alertas se encuentra que el sistema encuentra tres alertas por medio de 3 reglas que trae creadas para paquetes PING (Fig. 7).

```
[**] [1:768:7] ICMP PING REPLY [**]
[Classification: Misc activity] [Priority: 3]
06/27-00:56:11.003706 00:1D:D9:E3:EB:CD -> 00:22:B0:8E:10:E5 type:0x800 len:0x4A
192.168.0.100 -> 192.168.0.101 ICMP TTL:128 TOS:0x0 ID:14548 IpLen:20 DgmLen:60
Type:8  Code:0  ID:768   Seq:1280  ECHO

[**] [1:384:5] ICMP PING [**]
[Classification: Misc activity] [Priority: 3]
06/27-00:56:11.003706 00:1D:D9:E3:EB:CD -> 00:22:B0:8E:10:E5 type:0x800 len:0x4A
192.168.0.100 -> 192.168.0.101 ICMP TTL:128 TOS:0x0 ID:14548 IpLen:20 DgmLen:60
Type:8  Code:0  ID:768   Seq:1280  ECHO

[**] [1:408:5] ICMP Echo Reply [**]
[Classification: Misc activity] [Priority: 3]
06/27-00:56:11.008640 00:22:B0:8E:10:E5 -> 00:1D:D9:E3:EB:CD type:0x800 len:0x4A
192.168.0.101 -> 192.168.0.100 ICMP TTL:64 TOS:0x0 ID:29959 IpLen:20 DgmLen:60
Type:0  Code:0  ID:768  Seq:1280  ECHO REPLY
```

Fig. 7. Alertas generadas.

A continuación las reglas que generaron las alertas:

Regla1- ICMP PING REPLY
alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"ICMP PING REPLY"; itype:8; content:"abcdefghijklmnopqrstuv"; classtype:misc-activity; sid:768; rev:7;)

Regla2- ICMP PING
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING"; icode:0; itype:8; classtype:misc-activity; sid:384; rev:5;)
Regla3- ICMP ECO REPLY
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Echo Reply"; icode:0; itype:0; classtype:misc-activity; sid:408; rev:5;)

Se puede ver que es necesario que en cada criterio se cumplan todos los argumentos que se comparan con el paquete, para que se genere la alarma.

6.2 Prueba 2

Se crea una nueva regla para generar alertas cuando se ingrese al sitio web www.youtube.com.

Los paquetes que se capturaron para bus-

car algún patrón que caracterice el ingreso a este sitio web, son los mostrados en las Fig. 8 y 9.

Como se puede ver, un patrón básico en el flujo de información es www.youtube.com; por tal motivo se crea una regla básica haciendo uso del protocolo TCP o UDP. Particularmente para este análisis se crea con TCP, y en la opción content, se coloca, "www.youtube.com", de esta forma cuando el dato se compare con la regla, buscara que el protocolo sea TCP y que en el contenido del paquete este incluido el "content" mencionado.

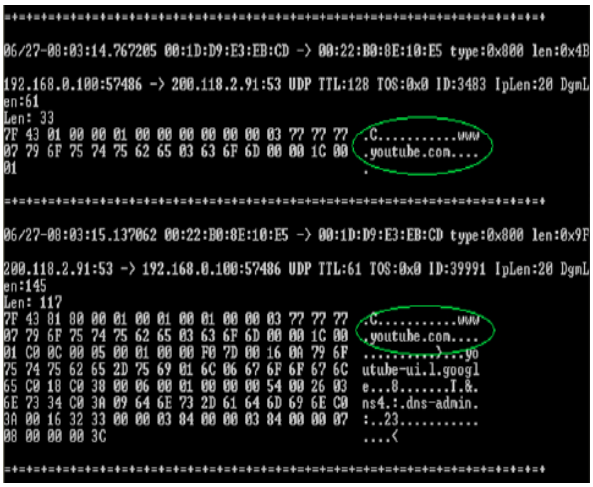

Fig. 8. Paquetes de tráfico capturados.

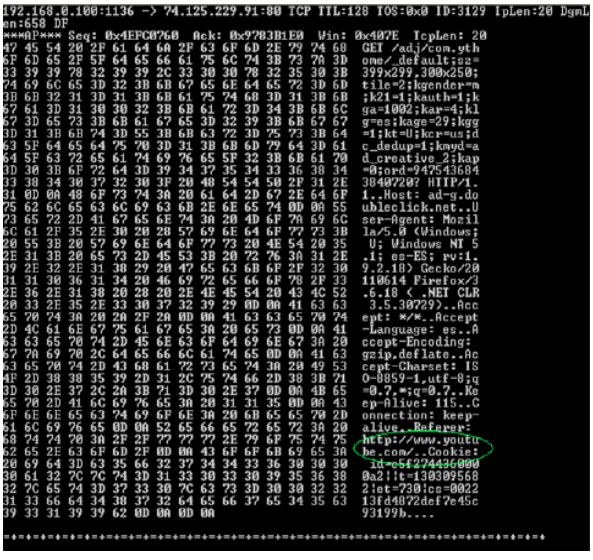

Fig. 9. Capturas obtenidas.

La regla creada es la que se muestra a continuación:

Los paquetes deben ser TCP enviados desde una red externa hacia nuestra red; en el sid, se coloca un valor grande para que no interfiera con los valores de las reglas de Snort y en rev, se coloca el número de revisiones que para este caso es cero:

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"INGRESANDO A YOUTUBE, BY JAIRO"; content:"www.youtube.com"; classtype:misc-activity; sid:10000; rev:0;)

Después de ingresar al sitio web www.youtube.com, se obtienen los siguientes mensajes de alerta visualizados en la Fig. 10. De esta manera se comprueba que la regla está siendo tomada.

```
[**] [1:10000:0] INGRESANDO A YOUTUBE, BY JAIRO [**]
[Classification: Misc activity] [Priority: 3]
06/27-08:27:10.711262 00:1D:D9:E3:EB:CD -> 00:22:B0:8E:10:E5 type:0x800 len:0x21C
192.168.0.100:1172 -> 74.125.229.44:80 TCP TTL:128 TOS:0x0 ID:5872 IpLen:20 DgmLen:526 DF
***AP*** Seq: 0x34B755EE  Ack: 0xCD82E0AD  Win: 0x4308  TcpLen: 20

[**] [1:10000:0] INGRESANDO A YOUTUBE, BY JAIRO [**]
[Classification: Misc activity] [Priority: 3]
06/27-08:27:10.711348 00:1D:D9:E3:EB:CD -> 00:22:B0:8E:10:E5 type:0x800 len:0x224
192.168.0.100:1173 -> 74.125.229.44:80 TCP TTL:128 TOS:0x0 ID:5873 IpLen:20 DgmLen:534 DF
***AP*** Seq: 0x881A0955  Ack: 0xCD360099  Win: 0x4308  TcpLen: 20
```

Fig. 10. Mensajes de atención.

7. CONCLUSIONES

Se pudo demostrar una de las principales características de Snort y es la flexibilidad que posee para la creación de reglas, ya que mediante la manipulación de unos cuantos parámetros es posible crear o personalizar las reglas.

El sistema de detección de intrusos basado en firmas nos permite tener ventajas frente a ataques conocidos, pero ofrecen capacidad casi nula de detección de nuevos ataques que no estén registrados en la base de datos.
Como se puede presumir cada vez serán más las reglas que se irán incorporando a este tipo de sistemas, lo que genera un gran

inconveniente ya que cada paquete debe ser comparado con cada requerimiento existente, lo que ocasiona que se requiera cada vez más capacidad de procesamiento de maquina en donde se ejecute el IDS, y de esta forma responder a las demandas de trafico establecidas; todo ello implica que es pertinente buscar nuevos métodos para analizar el flujo circundante en una networking.

Referencias Bibliográficas

[1] Artículo tomado como base para la introducción, [en línea]. Consultado en Agosto 5 de 2010, disponible en: http://www.psicofxp.com/forums/seguridad-informatica.47/74862-intrusion-detection-systems.html

[2] R. Carhuatocto, 1st Peruvian Workshop on IT Security, Sistemas de Detección de Intrusos. Instituto de Investigación UNI, p.p: 2. Barcelona-España 29 de Diciembre de 2003.

[3] C. Borghello, Cronología de los virus Informáticos, p.p: 7, Diciembre de 2009.

[4] The IP Smart Spoofing, Laurent Licour, Octubre de 2002, [en línea]. Consultado en Marzo de 2011, disponible en: http://www.net-security.org/dl/articles/smartspoof-en.pdf

[5] Tomado de L Campo, Diseño de sistemas distribuidos de detección de anomalías de red.. Universidad Politécnica de Madrid. Trabajo de investigación realizado por alumnos de doctorado, p.p: 11-12, Madrid-España, 2006.

[6] C. Borghello, Amenazas Lógicas - Tipos de Ataques - Denial of Service (DoS), [en línea]. Consultado en Agosto 23 de 2010, disponible en: http://www.segu-info.com.ar/ataques/ataques_dos.htm

[7] A. Escudero, Curso de Seguridad Tipos de ataques de Denegación de Servicio, p.p: 9, Septiembre de 2006.

[8] G. Verdejo, Articulo sobre Seguridad en Redes IP, [en línea]. Consultado en Septiembre 14 de 2010, disponible en: http://www.elrinconcito.com/articulos/SeguridadIP/SeguridadIP.pdf

[9] A. Escudero, Curso de Seguridad Tipos de ataques de Denegación de Servicio, p.p: 8, Septiembre de 2006.

[10] A. Arboleda, Sistema de detección de intrusos utilizando inteligencia artificial. Universidad del Cauca, Departamento de Sistemas, p.p: 25-32. Popayán, 2004.

[11] E. Mira, Implantación de un sistema de detección de intrusos en la Universidad de Valencia. Universidad de Valencia. Proyecto final de carrera, p.p: 18, 2009

[12] E. Mira, Implantación de un sistema de detección de intrusos en la Universidad de Valencia. Universidad de Valencia. Proyecto final de carrera, p.p: 19, 2009.

[13] Snort, User's Manual 2.9.0, The snort Project, [en línea]. Consultado en Octubre 26 de 2010, disponible en: http://www.snort.org/assets/166/snort_manual.pdf

[14] Análisis de la herramienta sniffer, Wireshark, Consultado en Noviembre 15 de 2010, disponible en: http://www.wireshark.org/docs/wsdg_html_chunked/.

[15] L. Campo, Diseño de sistemas distribuidos de detección de anomalías de red. Universidad Politécnica de Madrid. Trabajo de investigación realizado por alumnos de doctorado, p.p: 6, Madrid-España 2005/2006.

[16] Análisis de la herramienta Symantec: Symantec network security 7100 series, [en línea]. Consultado en Diciembre 5 de 2010, disponible en: http://www.symantec.com/region/la/product/appliance/7100/

[17] L. Campo, Diseño de sistemas distribuidos de detección de anomalías de red. Universidad Politécnica de Madrid. Trabajo de investigación realizado por alumnos de doctorado, p.p: 9, Madrid-España 2005/2006.

[18] A. López, Aprendiendo a programar con Libpcap, [en línea]. Consultado en Enero 11 de 2011, disponible en: http://www.tcpdump.org/#documentation

[19] C. Saavedra, Iniciando un proyecto GNOME con Libglade y Autotools. Enero de 2006, [en línea]. Consultado en Febrero 4 de 2011, Disponible en: http://www.gnome.org/~csaavedra/documents/usinglibglade.pdf

[20] C. PFC, Diseño y optimización de un sistema de detección de intrusos hibrido. Consultado en Febrero 16 de 2011, [en línea]. Consultado en Marzo de 2011, disponible en: http://www.adminso.es/images/8/86/Pfc_Carlos_cap5.pdf

[21] M. Giménez, Utilización de Sistemas de Detección de Intrusos como Elemento de Seguridad Perimetral, p.p: 48, Julio de 2008, [en línea]. Consultado en Marzo del 2011, disponible en: http://www.adminso.es/images/1/1d/PFC_marisa.pdf

[22] Snort: Users Manual 2.9.0, The snort project, [en línea]. Consultado en Marzo 9 de 2011. Disponible en: http://www.snort.org/assets/166/snort_manual.pdf

[23] Snort: Users Manual 2.9.0, The snort project, [en línea]. Consultado en Marzo 28 de 2011. Disponible en: http://www.snort.org/assets/166/snort_manual.pdf