



**UNIVERSIDAD
NACIONAL DE LOJA**



Facultad de Energía, las Industrias y los Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

“CONFIGURACIÓN DE UNA HERRAMIENTA OPEN SOURCE PARA LA DETECCIÓN DE INTRUSOS EN REDES WIFI CASO DE ESTUDIO: SURICATA IDS.”

*“Trabajo de titulación previo
a la obtención del título de
Ingeniera en Sistemas”*

Autora:

Jimena Gabriela Gutiérrez Jiménez.

Asesor académico:

Ing. Mario Enrique Cueva Hurtado, Mg. Sc

LOJA-ECUADOR

2019

CERTIFICACIÓN DEL DIRECTOR

Ing. Mario Enrique Cueva Hurtado, Mg. Sc.

DIRECTOR DE LA CARRERA DE INGENIERÍA EN SISTEMAS DE LA FELIRNNR-UNL

CERTIFICA:

Que la egresada **JIMENA GABRIELA GUTIÉRREZ JIMÉNEZ** autora del presente trabajo de titulación, cuyo tema versa **sobre “CONFIGURACIÓN DE UNA HERRAMIENTA OPEN SOURCE PARA LA DETECCIÓN DE INTRUSOS EN REDES WIFI CASO DE ESTUDIO: SURICATA IDS”** Ha sido dirigido, orientado y discutido bajo mi asesoramiento y reúne a satisfacción los requisitos exigidos en una investigación de este nivel por lo cual autorizo su presentación y sustentación.

Loja, 8 de agosto 2019



Ing. Mario Enrique Cueva Hurtado, Mg. Sc

DIRECTOR DE TRABAJO DE TITULACIÓN

AUTORÍA

Yo **Jimena Gabriela Gutiérrez Jiménez** declaro ser la autora del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido de esta.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional Biblioteca virtual.

Firma: _____



CI: 110473210-0

Fecha: 8 de agosto de 2019

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DE LA AUTORA, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL, Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO

Yo, **JIMENA GABRIELA GUTIERREZ JIMENEZ**, declaro ser la autora de la tesis titulada: **“CONFIGURACIÓN DE UNA HERRAMIENTA OPEN SOURCE PARA LA DETECCIÓN DE INTRUSOS EN REDES WIFI CASO DE ESTUDIO: SURICATA IDS”**, como requisito para optar el grado de: **INGENIERO EN SISTEMAS**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, publique al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los 13 días del mes de noviembre del dos mil diecinueve.

Firma:



Autor: Jimena Gabriela Gutiérrez Jiménez

Cédula: 1105159022

Dirección: Loja (Cdla. 8 de diciembre Juan María Riofrio y José María Riofrio)

Correo electrónico: jggutierrezj@gmail.com

Celular: 0979086121

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Mario Enrique Cueva Hurtado, Mg. Sc

Tribunal de Grado: Ing. Ángel Freddy Ganazhapa Mg. Sc

Ing. María del Cisne Ruilova Mg. Sc

Ing. Cristian Ramiro Narváez Guillen Mg. Sc

AGRADECIMIENTO

Agradezco a mi director de tesis, quien estuvo guiándome académicamente con su experiencia y profesionalismo, gracias a sus consejos y correcciones hoy puedo culminar este trabajo a los docentes de la carrera por sus conocimientos que me motivaron a desarrollarme como persona y profesional en la Universidad Nacional de Loja.

La Autora.

DEDICATORIA

A mi madre Blanca gracias por la paciencia, y a todos aquellos que me apoyaron moral y económicamente.

La Autora.

INDICE DE CONTENIDOS

ÍNDICE GENERAL

CERTIFICACIÓN DEL DIRECTOR	II
AUTORÍA	III
CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DE LA AUTORA, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL, Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO	IV
AGRADECIMIENTO	V
DEDICATORIA	VI
TABLA DE CONTENIDOS	VII
1. TÍTULO	1
“CONFIGURACIÓN DE UNA HERRAMIENTA OPEN SOURCE PARA LA DETECCIÓN DE INTRUSOS EN REDES WIFI CASO DE ESTUDIO: SURICATA IDS.”	1
2. RESUMEN	2
SUMMARY	3
3. INTRODUCCION	4
4. REVISION DE LITERATURA	6
4.1 Estado actual de herramientas open source para la detección de intrusos.	6
4.1.1 Sistema de detección de Intrusos (IDS)	7
4.1.2 Sistema de prevención de intrusos	11
4.1.3 Sistemas IDS open source	13
4.1.4 Ubicación de un IDS para su implementación.	17
4.2 Generalidades sobre las redes de área local inalámbricas	18
4.2.1 Aplicaciones de las WLAN	19
4.2.2 Arquitectura de las redes 802.11	20
4.2.3 Sistemas de detección de intrusiones en redes Wifi	21
5. MATERIALES Y METODOS	28

6. RESULTADOS	29
Fase 1: Analizar el estado actual de herramientas open source para la detección de intrusos.	29
Comparativa de IDS	29
Análisis del cuadro comparativo	30
Ventajas y desventajas de los sistemas de detección de intrusos	31
Fase 2: Diseñar un esquema de red doméstica y entorno virtual de equipos.	34
Diseño del entorno virtual de equipos.	37
Fase 3: Configuración y evaluación la herramienta Suricata	38
Establecer las reglas necesarias para configuración sistema IDS	38
Selección de los tipos de los ataques más comunes en redes wifi para poner a prueba la herramienta IDS.	47
Evaluación de la herramienta.	48
7. DISCUSIÓN	67
8. CONCLUSIONES	70
9. RECOMENDACIONES	71
10. BIBLIOGRAFIA	72
11. ANEXOS	74

ÍNDICE DE FIGURAS

Figura 1. Esquema de un IDS [7].....	7
Figura 2. Esquema general de un IPS	12
Figura 3. Red Ad Hoc.	20
Figura 4. Red Tipo Infraestructura	21
Figura 5. Comparación trimestral de números de ataques DDos	27
Figura 6. Esquema lógico del escenario.	37
Figura 7. Configuración de variables.....	38
Figura 8. Topología Direccionamiento de red	38
Figura 9. Reglas en el archivo de configuración.....	39
Figura 10. Activación reglas Suricata-update	40
Figura 11. Reglas habilitadas.....	41
Figura 12 Listado de reglas habilitadas.....	42
Figura 13. IDS Ejecutándose	43
Figura 14. Tipos de salidas del IDS	44
Figura 15. Información archivo fast.log	45
Figura 16. Información archivo http.log.	45
Figura 17. Información archivo stats.log.....	46
Figura 18. Información archivo eve.json.....	46
Figura 19. IDS puesta en marcha una CPU	49
Figura 20. IDS puesta en marcha dos CPU	49
Figura 21. Porcentaje de consumo de CPU y memoria.....	49
Figura 22. Exploración de red con Nmap	52
Figura 23. Alerta generada por exploración con Nmap	53
Figura 24. Ejecución del IDS.....	54
Figura 25. Consumo de recursos durante el ataque	54
Figura 26. Esquema ataque DoS	55
Figura 27. Inicio de Metasploit	55
Figura 28. Ataque DoS	56
Figura 29. Puesta en marcha de Suricata	57
Figura 30. Registro de alertas del ataque DoS.....	58
Figura 31. Consumo de recursos.....	59
Figura 32. Ataque DOS con Slowloris	60
Figura 33. Alerta de ataque DOS	61
Figura 34. Ataque MITM.	61
Figura 35. Vista principal Ettercap	62
Figura 36. Selección de Interfaz	62
Figura 37. Selección de escaneos de host.....	63
Figura 38. Lista de host en la red.....	63
Figura 39. Ataque MITM por envenenamiento arp	64
Figura 40. Ejecución de ataque	64
Figura 41. Conexiones en el host infectado	65
Figura 42. Captura de información ataque	65
Figura 43. Registro del ataque en archivo eve.log	66
Figura 44. Consumo de recursos en ataque MITM	66

ÍNDICE DE TABLAS

TABLA I LUGAR PARA IMPLEMENTAR UN IDS.....	17
TABLA II COMPARATIVA IDS.....	29
TABLA III. ELEMENTOS NECESARIOS PARA SERVIDORES [25].....	34
TABLA IV. ELEMENTOS NECESARIOS EN ESTACIONES CLIENTE [25].....	35
TABLA V. ELEMENTOS NECESARIOS PARA MONTAR EL ESCENARIO	36
TABLA VI. COMPONENTES DEL ESQUEMA DE RED	37
TABLA VII. ATAQUES COMUNES DETECTADOS POR UN IDS.	47
TABLA VIII. SIMULACIONES DE ATAQUES	50

1. TÍTULO

“CONFIGURACIÓN DE UNA HERRAMIENTA OPEN SOURCE PARA LA DETECCIÓN DE INTRUSOS EN REDES WIFI CASO DE ESTUDIO: SURICATA IDS.”

2. RESUMEN

Los sistemas IDS surgen a medida que avanza la tecnología en telecomunicaciones al igual que los ataques a las distintas redes, dado que la información se ha convertido en un activo sustancial en organizaciones y empresas, cualquiera que se encuentre conectado a la red de internet está expuesto a que sus datos sean recolectados con fines maliciosos si no se cuenta con un sistema o herramienta de protección adecuado, generalmente el resguardo de información se limita al uso de antivirus y firewall si bien estos ayudan, no siempre podrán defender ya que algunos ataques y virus están fuera de su alcance, y es ahí en donde seleccionar, implementar y configurar un IDS para que contribuya a la protección de información se torna necesario para quienes desean agregar un plus a su red.

El presente trabajo de titulación está orientado a la implementación y configuración de una herramienta open source para la detección de intrusos en una red Wi-Fi, se realizó una Revisión Sistemática de Literatura haciendo uso de la metodología de Bárbara A. Kitchenham, obteniendo datos relevantes como ventajas, desventajas, características y ataques comunes detectados por un IDS, con esta información se obtuvo el estado actual de los Sistemas de detección de intrusos, concluyendo que los ataques más comunes en una red WIFI son: exploración de red, denegación de servicio (DoS) y hombre en el medio (MiTM).

Se elaboró un entorno virtual de pruebas en el cual se estableció el diseño de la red (Wlan), hardware y software donde se implementó y configuró la herramienta, haciendo uso del monitoreo por detección de firmas y activando el uso de una defensa adicional como es Suricata-update, una opción recientemente integrada con el fin de optimizar el proceso de descarga y actualización de las reglas con las que trabajará el IDS.

Se sometió al motor IDS a tres ataques haciendo uso de herramientas de sistema operativo kali linux (Nmap, Metasploit y Ettercap), comprobando así el funcionamiento del sistema de detección de intrusos a través de la identificación de las alertas emitidas por cada amenaza en la red.

SUMMARY

IDS systems surge as telecommunication technology advances as do attacks on different networks, given that information has become a substantial asset in organizations and businesses, anyone connected to the Internet network is exposed to having their data collected for malicious purposes without an adequate protection system or tool, generally the information safeguard is limited to the use of antivirus and firewall although these help, they will not always be defending since some attacks and virus are out of their reach, and it is there where to select, implement and configure an IDS to contribute to the protection of information becomes necessary for those who want to add a plus to their network.

The present titulation work is oriented to the implementation and configuration of a open source tool for the detection of intruders in a Wi-Fi network. A Literature Systematic Review was carried out using the methodology of Barbara A. Kitchenham, getting relevant data such as advantages, disadvantages, characteristics and common attacks detected by an IDS, concluding that the most common attacks in a WIFI network are: network scan, denial of service (DoS) and man in the middle (MiTM).

A virtual testing environment was elaborated in which the design of the network (Wlan), hardware and software was established, where the tool was implemented and configured, making use of monitoring by signature detection and activating the use of an additional defense such as Suricata-update, a recently integrated option in order to optimize the process of downloading and updating the rules with which the IDS worked.

The IDS engine was subjected to three attacks using kali linux operating system tools (Nmap, Metasploit and Ettercap), thus checking the functioning of the intrusion detection system by identifying the alerts issued by each threat in the network.

3.INTRODUCCION

La seguridad en las redes de datos tanto internas como externas es una de las principales preocupaciones de las organizaciones, empresas, e instituciones, que están interesadas en salvaguardar sus datos de intrusos, con el constante avance de la tecnología los ataques a las redes se han vuelto más sofisticados, lo cual amerita nuevas técnicas de prevención y resguardo. El usuario no solo debe limitarse a protegerse con un antivirus o un firewall, también necesita tener una herramienta que detecte los posibles ataques a su red, y que la respuesta a estos ataques sea en tiempo real.

Los Sistemas de Prevención y Detección de Intrusos (IPS/IDS) contribuyen a la seguridad en las redes de datos, previniendo vulnerabilidades que en algunos casos los firewalls simplemente no pueden responder de forma inmediata [1]. La mayoría de estos sistemas son pagados y costosos, pero también existen herramientas de código abierto las cuales se adaptan perfectamente las necesidades de cualquier organización, es por ello por lo que en el presente documento se da a conocer de manera general algunas de estas herramientas su funcionamiento, así como la comparativa y el análisis de este, sus ventajas y desventajas.

Actualmente el IDS Suricata es una herramienta que cuenta con una amplia gama de funciones tanto para detección como prevención de intrusos, además su comunidad es una de las más activas en cuanto a renovación de la herramienta, cabe destacar que para aprovechar el máximo potencial de esta herramienta es necesario una correcta guía de instalación, así como su configuración, ubicación y creación de reglas, dependiendo de la información que se desee proteger.

El objetivo del presente trabajo de titulación es la configuración de una herramienta open source para la detección de intrusos en redes wifi caso de estudio: Suricata IDS.; para lo cual se emplea un entorno virtual. Este objetivo general se descompone en los siguientes objetivos específicos:

- Estado actual de herramientas open source para la detección de intrusos.
- Diseñar un esquema de red doméstico y entorno virtual de equipos.
- Configurar y evaluar la herramienta Suricata.

El presente trabajo de titulación está conformado por secciones y anexos que permiten la comprensión del trabajo realizado.

En la sección de *Revisión de Literatura*, se especifica la información teórica relacionada al proyecto como: Estado actual de los sistemas de detección de intrusos, esta se realizó utilizando el método de revisión sistemática de literatura.

En la sección *Materiales y Métodos* se detallan la metodología y fases aplicadas a lo largo del proyecto.

La sección *Resultados* se hace referencia a las fases seguidas para lograr el cumplimiento de los objetivos planteados.

En la sección *Discusión*, se valora los resultados obtenidos según los objetivos. Finalmente, se redacta las Conclusiones obtenidas al finalizar el presente trabajo y las Recomendaciones generadas a favor de un buen manejo de un IDS en una red Wifi.

La *Bibliografía* es el apoyo de la investigación del trabajo realizado y por último los *Anexos* que sirven de información adicional para la justificación y sustento del proyecto realizado.

4. REVISION DE LITERATURA

4.1 Estado actual de herramientas open source para la detección de intrusos.

La primera persona capaz de documentar la necesidad de un mecanismo que autoriza la revisión de los eventos de seguridad fue James P. Anderson publicó un trabajo titulado "Computer Security Threat Monitoring and Surveillance" donde se establecen las bases de la detección de intrusos en sistemas de computadores, principalmente mediante la consulta de archivos de registros de sucesos [2]. Anderson propuso un sistema de clasificación que diferenciaba entre ataques internos y ataques externos, basado en si los usuarios tenían permiso de acceso o no al computador [3]. Estos eran los principales objetivos de los mecanismos de auditoría de seguridad:

- Debían proporcionar suficiente información para que los encargados de seguridad localicen el problema, pero no para efectuar un ataque [4].
- Debía ser capaz de obtener datos de distintos recursos del sistema [4].
- Para evitar ataques internos, debía detectar usos indebidos o fuera de lo normal por parte de los usuarios [4].
- El diseño del mecanismo de auditoría debía ser capaz de obtener la estrategia usada por el atacante para entrar en las cuentas [4].

Este sistema sirvió para dar solución al problema que representaba el hecho de que intrusos se apoderarán de cuentas legítimas de usuarios, lo que dio lugar a una gran parte de la investigación sobre IDS a lo largo de los años ochenta y noventa. A partir de aquí empezaron a surgir los primeros IDS físicos alcanzando su auge en 1995 con la crisis del Firewall, naciendo así IDS como "Computer Watch", "ISOA"

Por otra parte, Dorothy E. Denning, asistido por Peter Neuman, publicaron un modelo de IDS en 1986 denominado IDES (Intrusion Detection Expert System) basado en reglas, que sirvió de base para muchos sistemas el día de hoy. IDES tiene un doble enfoque con una regla basada en sistema experto para detectar los tipos de intrusiones, más una anomalía estadística de detección de los componentes sobre la base de perfiles de los usuarios [4].

En 1991, investigadores de la Universidad de California crearon un prototipo de Sistema Distribuido de Detección de Intrusos (DIDS), que fue también un sistema

experto. NADIR, también en 1991, fue un prototipo IDS desarrollado en el Laboratorio Nacional de Los Álamos de la Red Integrada de Computación (RIC), y fue fuertemente influenciado por la labor de Denning y Lunt. NADIR utilizó una estadística basada en detector de anomalía y un sistema experto. A partir de este momento, se han ido proponiendo y creando nuevos sistemas de detección de intrusos [3].

En la actualidad, los IDS han ido evolucionando notablemente, poseen mejores características que en sus inicios no consideraron y se han enfocado en la protección en sistemas distribuidos, es decir sistemas que están alejados o separados del sistema a analizar, con el fin de que no sean detectados y anulados por los intrusos.

4.1.1 Sistema de detección de Intrusos (IDS)

Un sistema de detección de intrusos es un componente más dentro del modelo de seguridad de una organización. Consiste en detectar actividades inapropiadas, incorrectas o anómalas desde el exterior- interior de un sistema informático [5]. De esta forma, un IDS ha de ser capaz de distinguir entre un acceso “normal” al sistema, que puede surgir de la puesta en marcha de servicios ofertados, y un intento de vulnerar de algún modo dichos servicios [2]. Por lo tanto, un IDS deberá ser capaz de, al menos, generar alertas en todas aquellas situaciones que puedan ser consideradas como eventos de intrusión.[6].

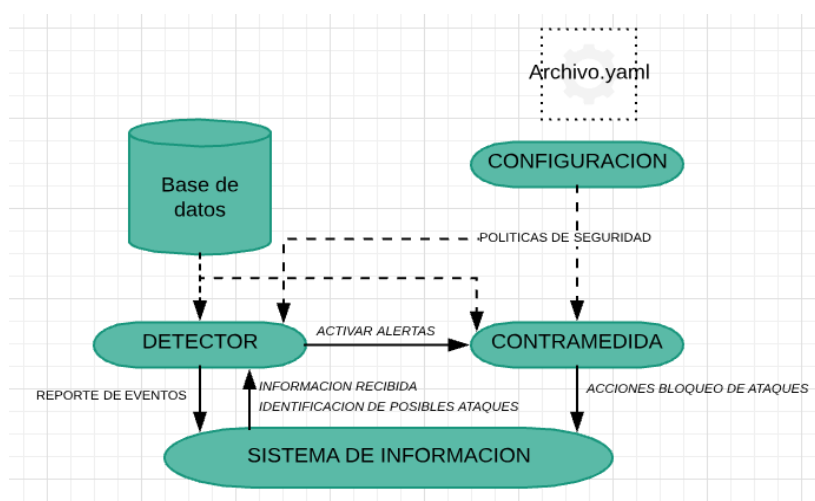


Figura 1. Esquema de un IDS [7]

Un IDS está estructurado por:

- Una base de datos con la información de los ataques percibidos por el IDS.

- El archivo de configuración debidamente establecido con sus reglas y protocolos a seguir [7].
- Sus funciones de contramedida se activarán en caso de estar en modo inline.
- El detector en sí, que genera las alertas en los archivos log [7]
- Su sistema de información con el cual interactúa para su funcionamiento y provee de los datos [7].

El funcionamiento de estas herramientas de detección de intrusos se basa en el análisis pormenorizado del tráfico de red, el cual al entrar al analizador se compara con firmas de ataques ya conocidos, o comportamientos sospechosos como, por ejemplo: escaneo de puertos, paquetes malformados o anómalos, incumplimiento de políticas de seguridad, contenido de los paquetes y su comportamiento [8].

NOTA: No hace falta disponer de un cortafuego para la utilización de un IDS. Se han comentado las alternativas anteriores ya que el uso de los IDS es más habitual en las organizaciones cuya información es más valiosa para los intrusos, y el uso de un firewall se hace indispensable para evitar las conexiones no autorizadas a un sistema interno desde el exterior [8].

4.1.1.1 Clasificación de los IDS

Existen diferentes tipos de sistemas de detección de intrusos, clasificados según su situación física, forma en la que detectan las intrusiones y según su reacción al detectar un posible ataque

Dependiendo de la fuente de información analizada:

- *Sistemas de detección de intrusos de red (NIDS):* Escanea los paquetes de red al nivel del enrutador o host, audita la información de los paquetes y registra cualquier paquete sospechoso en un archivo de registros especial con información extendida, un ejemplo de estos IDS de red sería: BRO ids, SNORT, Suricata, Security Onion [2].

Basándose en estos paquetes sospechosos, un IDS basado en la red puede escanear su propia base de datos de firmas de ataques a la red y asignarles un nivel de severidad para cada paquete. Si los niveles de severidad son lo suficientemente altos, se enviará un correo electrónico o un mensaje de advertencia a los miembros del equipo de seguridad para que ellos puedan investigar la naturaleza de la anomalía [9]. Los IDS que son capaces de escanear grandes volúmenes de actividad en la red y

etiquetar transmisiones dudosas, dentro de la industria de seguridad debido a la inseguridad inherente de los protocolos TCP/IP, desarrollar herramientas para la defensa de la información se ha tornado imperativo entre estos están los escáneres, husmeadores y otras herramientas de auditoría y detección para así prevenir violaciones de seguridad por actividades maliciosas en la red, tales como [10]:

- Engaño de direcciones IP (IP Spoofing)
- Ataques de rechazo de servicio (dos)
- Envenenamiento de caché arp
- Corrupción de nombres DNS.
- Ataques de hombre en el medio [9].
- *Sistema de detección de intrusiones en el host o máquina:* Analiza diferentes áreas para determinar el uso incorrecto (actividades maliciosas o abusivas dentro de la red) o alguna intrusión (violaciones desde afuera).

Los IDS basados en host de Linux y Unix hacen uso extensivo de syslog y de su habilidad para separar los eventos registrados por severidad, por ejemplo, mensajes menores de impresión versus advertencias importantes del kernel [11]. Filtran los registros (lo cual, en el caso de algunas redes y registros de eventos del kernel pueden ser bastante detallados), los analizan, vuelven a etiquetar los mensajes anómalos con su propia clasificación de severidad y los reúne en su propio registro para que sean analizados por el administrador [11]. Se encargan de verificar la integridad de los datos de archivos y ejecutables importantes mediante el uso de una base de datos de archivos confidenciales (y cualquier archivo añadido por el administrador) y crea una suma de verificación de cada archivo con una utilidad de resumen de archivos de mensajes tal como md5sum (algoritmo de 128-bit) o sha1sum (algoritmo de 160-bit). El IDS basado en host luego almacena las sumas en un archivo de texto plano y periódicamente compara las sumas de verificación contra los valores en el archivo de texto [12].

- ***Dependiendo del tipo de análisis ejecutado:***

Conocimiento: Un IDS basado en conocimiento hace referencia a una base de datos de perfiles de vulnerabilidades de sistemas ya conocidos para identificar intentos de intrusión activos. En este caso, es de suma importancia que la estructura tenga una política de actualización continua de la base de datos o firmas, las firmas juegan un papel muy importante en los IDS, las más utilizadas son: Amenazas emergentes,

Amenazas emergentes Pro y VRT de Source Fire, ya que estas podrán garantizar la continuidad de la seguridad del entorno a proteger, teniendo en cuenta que lo que no se conoce, no será protegido [13].

Comportamiento: El IDS basado en comportamiento, por otro lado, analiza el comportamiento del tráfico siguiendo una línea de base o estándar de actividad normal del sistema para identificar intentos de intrusión. En el caso de que haya desviaciones de este patrón o líneas de base, se pueden tomar algunas acciones, ya sea bloqueando ese tráfico temporalmente, alarmas para núcleos de operación de red (NOC/SNOC), permitiendo que esa anomalía pueda ser mejor investigada, liberada o permanentemente bloqueada [13].

- ***Dependiendo del tipo de respuesta activada***

Activo: Se define un IDS como activo desde el momento en que se define para bloquear automáticamente ataques o actividades sospechosas que sean de su conocimiento, sin necesidad de la intervención del administrador. Aunque potencialmente es un modelo extremadamente interesante, es importante un ajuste de parámetros adecuado a los ambientes protegidos para minimizar falsos positivos, bloqueando conexiones legítimas y causando trastornos para las empresas [13].

Pasivo: Estos IDS realizan la función de notificar a la autoridad competente o al administrador de la red que hubo un intento de ataque mediante el sistema que sea: alerta, log, etc. Pero no actúan sobre el ataque o el atacante. Estos IDS solamente se dedican a procesar la información en busca de intrusos, una vez que se encuentra con un intruso o un intento de intrusión el IDS emite una alerta y deja que el operador o administrador de la red tome una decisión para realizar una acción en consecuencia a la intrusión, este sistema carece de las unidades de respuesta [14].

4.1.1.2 Tipos de intrusiones

Una actividad intrusiva resulta del agregado de otras actividades individuales que por sí solas no constituyen un comportamiento intrusivo de ningún tipo. Así las intrusiones pueden clasificarse en:

- Intrusivas, pero no anómalas: denominados Falsos Negativos (el sistema erróneamente indica ausencia de intrusión). En este caso la actividad es intrusiva pero como no es anómala no es detectada. No son deseables, porque dan una falsa sensación de seguridad del sistema [15].

- No intrusivas pero anómalas: denominados Falsos Positivos (el sistema erróneamente indica la existencia de intrusión). En este caso la actividad es no intrusiva, pero como es anómala el sistema "decide" que es intrusiva. Deben intentar minimizarse, ya que en caso contrario se ignorarán los avisos del sistema, incluso cuando sean acertados [15].
- No intrusiva ni anómala: son Negativos Verdaderos, la actividad es no intrusiva y se indica como tal [15].
- Intrusiva y anómala: se denominan Positivos Verdaderos, la actividad es intrusiva y es detectada. Los detectores de intrusiones anómalas requieren mucho gasto computacional, ya que se siguen normalmente varias métricas para determinar cuánto se aleja el usuario de lo que se considera comportamiento normal [15].

4.1.2 Sistema de prevención de intrusos

Cabe indicar que cuando un IDS tiene capacidad de modificar la configuración de elementos de red, se conoce como Sistema de Prevención de Intrusos (IPS). También se pueden clasificar según su funcionamiento, ya sea por detección de patrones conocidos o por detección de anomalías, para la configuración de estos últimos, se necesitan conocer los perfiles de los usuarios legítimos, esta característica hace que en la mayoría de los IDS disponibles sea de detección de patrones, actualmente aplicaciones IDS son complementadas con técnicas de defensa basados en IPS [16].

Los IPS fueron inventados de forma independiente por Jed Haile y Vern Paxson para resolver ambigüedades en el monitoreo pasivo de redes de computadoras, al situar sistemas de detecciones en la vía del tráfico.

Los IPS presentan una mejora importante, al tomar decisiones de control de acceso basados en los contenidos del tráfico, en lugar de direcciones IP o puertos. Tiempo después, algunos IPS fueron comercializados por la empresa One Secure, la cual fue finalmente adquirida por netscreen Technologies, que a su vez fue adquirida por Juniper Networks en 2004[3].

Dado que los IPS fueron extensiones literales de los sistemas IDS, en algunos sistemas trabajan conjuntamente [1].

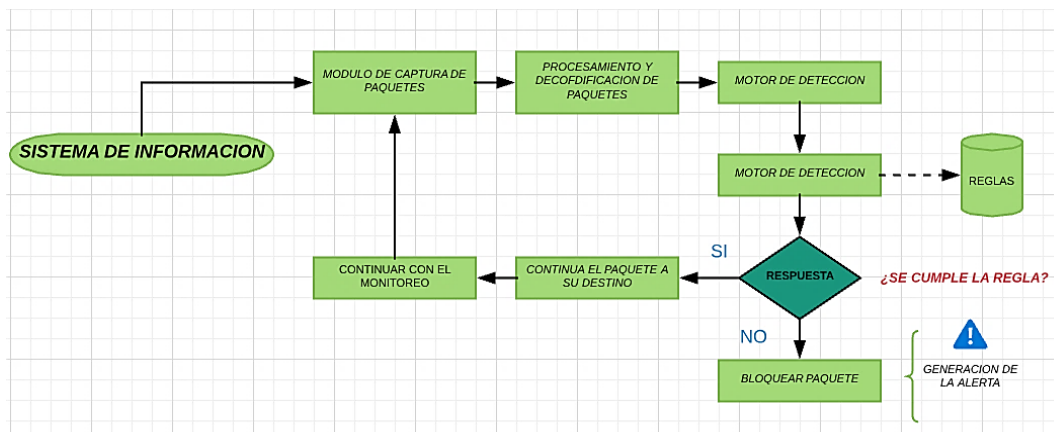


Figura 2. Esquema general de un IPS

“Un Sistema de Prevención de Intrusos IPS (Intrusion Prevention System) es un dispositivo que ejerce el control de acceso en una red informática para proteger a los sistemas computacionales de ataques y abusos. Los IPS no solo detectan, sino que son capaces de poder detener la falla o la vulnerabilidad.” [3] en la figura 2 se explica su funcionamiento general.

4.1.2.1 Clasificación

Los sistemas de prevención de intrusiones se pueden clasificar en cuatro tipos diferentes:

- Prevention Intrusion Network (PIN): Basado en la red detectan tráfico sospechoso, mediante el análisis de la actividad de protocolo [17].
- Wireless Intrusion Prevention System (WIPS): Sistemas de prevención de intrusiones inalámbricas, detectan tráfico malicioso mediante el análisis de protocolos de redes inalámbricas [17].
- Network Behavior Analysis (NBA): Análisis de comportamiento de la red, examina el tráfico para identificar las amenazas que generan flujos de tráfico inusuales, como escaneo de red, denegación de servicios, ciertas formas de malware y violaciones de política [17].
- Host-based Intrusion Prevention System (HIPS): Los sistemas de prevención de intrusos basados en host controlan y detectan actividades sospechosas como acceso no autorizado y cambios en la configuración en el equipo que monitorean [17].

4.1.2.2 Métodos de Detección.

- Detección basada en firmas: Este método de detección utiliza reglas basadas en patrones de ataque que se comparan con el paquete capturado y si cumple con los parámetros de inmediato es bloqueado [17].
- Anomalía basada en estadísticas de detección: Mediante este método los IPS crean una línea base que representa la actividad normal de los usuarios, generalmente cuando existe un ataque el sistema detecta un desvío del comportamiento normal de la red y genera la acción [17].
- De estado de detección de análisis de protocolo: Este método identifica anomalías en protocolo comparando los encabezados de los paquetes y tomando la acción si no cumple con los parámetros que lo identifican [17].

4.1.2.3 Características de los IPS

Los IDS que han tenido más acogida debido a su eficiencia y constante actualización están Snort, Suricata, debido a que cumplen con ciertas características que benefician a aquellas empresas u organizaciones que desean implantarse, estas características son:

- Capacidad de reacción automática ante incidentes.
- Aplicación de nuevos filtros conforme detecta ataques en progreso.
- Disminución de falsas alarmas de ataques a la red
- Bloqueo automático frente a ataques efectuados en tiempo real Protección de sistemas no parchados
- Optimización en el rendimiento del tráfico de la red [7]

4.1.3 Sistemas IDS open source

Una plataforma Open Source de código abierto significa que es de libre acceso, con lo que el usuario es autónomo para manipular ese software, por lo tanto, una vez obtenido puede ser: usado, estudiado, cambiado y redistribuido libremente.

4.1.3.1 SNORT

Snort un sniffer de paquetes y un sistema de detección de intrusos basado en red o NIDS. Implementa un motor de detección de ataques el cual permite registrar, alertar y responder ante cualquier anomalía previamente definida. Estos registros quedan almacenados en formato binario el cual se puede convertir a formato PCAP 2 u otros formatos más legibles.

Además, también se puede guardar en bases de datos como es mysql. Así mismo, existen herramientas complementarias a Snort que hacen que este IDS sea un sistema muy completo y fácil de administrar. Como, por ejemplo, herramientas que almacenan las alertas detectadas por Snort en una base de datos (Barnyard2) y otras que recogen de esta base de datos las alertas y las muestran en una interfaz gráfica de fácil manejo.[8]

A su vez, también existen otros programas complementarios para mostrar informes en tiempo real (ACID) o para convertir a Snort, además de IDS, en IPS. Snort implementa un lenguaje de creación de reglas flexibles, potentes y sencillas, pudiendo generar todas las alertas que se requiera.[13]

Un usuario puede formar una regla y compartirla a través de Internet para que todos los demás usuarios de Snort se puedan beneficiar de esta firma. Existen grandes comunidades las cuales nos ofrecen gran cantidad de reglas de ataques y conexiones sospechosas que podemos incluir en nuestro IDS de manera gratuita, aunque también hay conjuntos de reglas de pago.[8]

Este sistema de compartición de conocimiento contra ataques hace que Snort sea un sistema para detectar cualquier tipo de ataque. Snort, es un software gratuito, bajo licencia GPL y puede ser instalado tanto en sistemas operativos Windows como en sistemas UNIX/Linux [8].

Además, es un sistema probado y fiable y que cuenta con un gran soporte y actualizaciones conforme se van descubriendo nuevas vulnerabilidades a través de los distintos boletines de seguridad.

Elementos de Snort

Antes de iniciar la instalación y configuración de Snort es importante conocer los elementos que lo componen.

Los elementos que componen el esquema básico de su arquitectura son:

- Módulo de captura del tráfico. Es el encargado de capturar todos los paquetes de la red utilizando la librería libpcap [18].
- Decodificador. Se encarga de formar las estructuras de datos con los paquetes capturados e identificar los protocolos de enlace, de red, etc [18].
- Preprocesadores. Permiten extender las funcionalidades preparando los datos para la detección. Existen diferentes tipos de preprocesadores dependiendo del

tráfico que queremos analizar (por ejemplo, existen los preprocesadores http, telnet) [18].

- Motor de Detección. Analiza los paquetes en base a las reglas definidas para detectar los ataques [18].
- Archivo de Reglas. Definen el conjunto de reglas que regirán el análisis de los paquetes detectados [18].
- Plugins de detección. Partes del software que son compilados con Snort y se usan para modificar el motor de detección [18].
- 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.[8]

4.1.3.2 SURICATA

Suricata es una herramienta escalable. Este monitor de seguridad hace uso de las funciones multi-hilo de manera que solo con ejecutarse en una instancia el monitor balanceará su carga entre todos los procesadores disponibles, evitando incluso alguno de ellos si así de ser necesario. Gracias a ello, esta herramienta es capaz de procesar un ancho de banda de hasta 10 gigabits por segundo sin que ello repercuta sobre el rendimiento [6]. Esta herramienta también es capaz de identificar los principales protocolos de red, siendo capaz de controlar en todo momento todo el tráfico que se genera en el sistema y controlando posibles amenazas de malware.

Elementos de Suricata

- Multi-threading: esta característica permite ejecutar varios procesos subprocesos de manera simultánea, de esta manera definir una arquitectura multinúcleo y administrar cada núcleo del procesador para que se encargue de uno o más hilos [19].
- Estadísticas de rendimiento: módulo que permite llevar un conteo de variada información y presentarlos como estadísticas al administrador [19].
- Detección automática de protocolos: facilita la implementación de reglas utilizando palabras claves de los protocolos como FTP, HTTP, TLS, SMB [19].
- Módulo Log HTTP: lleva un registro de las peticiones http y las almacena en un formato log apache.

- Altamente escalable: esto permite que el Hardware de productos básicos puede alcanzar hasta 10 gigabit de velocidad de tráfico real, sin sacrificar la cobertura del conjunto de reglas.
- Identificación del archivo, sumas de verificación MD5 y Archivo de Extracción: Suricata puede identificar miles de tipos de archivos, mientras cruza la red. No solo puede identificarla, pero debe decidir si quiere ir más lejos, se puede etiquetar para la extracción y el archivo será escrito en el disco con un archivo de metadatos que describen la captura y el flujo de la situación.

4.1.3.3 BRO IDS

Bro, que fue renombrado Zeek a finales de 2018 y a veces se conoce como Bro-IDS o ahora Zeek-IDS, Es un sistema de detección de intrusiones para UNIX/Linux Open Source, algo distinto a Snort y Suricata, en cierto modo, Bro es tanto un IDS basado en anomalías como en firmas. El tráfico capturado generará una serie de eventos [6].

Por ejemplo, un evento podría ser un inicio de sesión de usuario a un FTP, conexión a servicio web o casi cualquier cosa. Es un Intérprete de Políticas Script. Con su propio lenguaje de administración (Bro-Script) ofrece posibilidades muy interesantes, Bro permite descargar ficheros encontrados en nuestro entorno, remitirlos para un análisis de malware, notificar si se encuentra con algún problema y después introducir en la lista negra la fuente del mismo, incluso permite gestionar el apagado del equipo remoto del usuario [6].

La arquitectura de Bro IDS, se basa en dos componentes:

- Motor de eventos (event engine): Reduce el flujo de paquetes, organizándose para ser llevados a un nivel superior.
- Intérprete de Scripts (policy script interpreter): Este motor de políticas tiene su propio lenguaje (Bro-Script) y puede realizar algunas tareas muy potentes y versátiles, cuando se detecta una actividad determinada.[6]

Características

Gran capacidad de análisis a nivel de protocolo y las políticas especializadas y configurables y la capacidad de ser una herramienta de análisis forense.

- Es de elevado rendimiento y capacidad para gestionar grandes volúmenes de tráfico.

- Sus alertas pueden ser configuradas para generar eventos de log, alertas en tiempo real y hasta ejecución de comandos de sistema.
- Las políticas además de generar logs por actividad sospechosa, puede generar logs de actividad normal dependiente de las configuraciones dadas.
- A través del lenguaje de scripts de políticas, se pueden crear políticas específicas para un entorno de red o una actividad concreta de acuerdo con las necesidades tecnológicas de una institución.
- Se puede ejecutar Bro para detección en tiempo real usando una determinada interfaz de red o leyendo un fichero pcap.
- No hay una interfaz gráfica de usuario nativa, pero existen herramientas de código abierto de terceros disponibles para un front-end web para consultar y analizar alertas procedentes de Bro-IDS, por ejemplo, la pila ELK.

4.1.4 Ubicación de un IDS para su implementación.

Los IDS pueden ser implementados en diferentes partes de la red, por lo general, se colocan en zonas de fuera de esta, con la finalidad de analizar y estudiar los posibles ataques, así como también se colocan zonas internas para analizar las solicitudes que hayan pasado a través la seguridad externa de la red.

TABLA I LUGAR PARA IMPLEMENTAR UN IDS.

Puntos	Ventajas	Desventajas
Delante del contrafuego externo	Monitorizar el número y el tipo de ataques dirigidos contra la infraestructura de la organización Detectar ataques cuyo objetivo es los cortafuegos principales.	No permite detectar ataques que utilicen en sus comunicaciones algún método para ocultar información, como algoritmos de encriptación o estenografía. En caso de gran cantidad de tráfico de red, puede causar saturación descartando parte de información. El NIDS puede convertirse en blanco fácil si algún atacante logra identificarlo.
Detrás del Cortafuegos externo	Se monitorizan intrusiones que logran atravesar el firewall principal. Detectar ataques a servidores que ofrecen servicios públicos. En caso de no detectar ataques con éxito, puede reconocer algunas consecuencias de estos, como intentos de conexiones salientes, realizadas desde servidores comprometidos.	No permite identificar ataques que utilicen métodos de encriptación de información. Normalmente en este segmento de red el NIDS no puede analizar todo el tráfico, descartando datos. La seguridad del NIDS mejora con la inclusión de los cortafuegos que lo separa de la red del exterior, sin embargo, esto no excluye tomar medidas adicionales para evitar que pueda ser comprometido por atacantes.
Redes principales	Detectar ataques producidos desde dentro de la propia red, como los realizados por personal interno.	No permite identificar ataques que utilicen métodos de encriptación de información. Las características de estos dispositivos podrían impedir la monitorización de los miembros de la red.

		Vulnerabilidad de los sistemas frente ataques internos de la red.
Subredes de valor crítico	Detectar ataques realizados contra elementos críticos de la red. Dedicar especial atención a los recursos más valiosos de la infraestructura.	No permite identificar ataques que utilicen métodos de encriptación de información. No están estratégicamente bien situados ante ataques de origen interno.
Máquinas	Evitar ataques que utilicen métodos de encriptación de información.	Reducción del rendimiento de la máquina que monitoriza. Que la máquina anfitriona sea Comprometida puede traer consecuencias respecto que el detector pierda eficacia, controlado por el atacante, obtener información de la infraestructura, enviar falsas alarmas, etc.

Fuente: José Eduardo Arteaga Pucha [6]

4.2 Generalidades sobre las redes de área local inalámbricas

Las redes inalámbricas han ganado mucha popularidad en los últimos tiempos, esta popularidad ha crecido hasta tal punto en que las podemos encontrar en casi cualquier ámbito de nuestra vida cotidiana, teléfonos inalámbricos, ordenadores y teléfonos móviles son algunos de los ejemplos más evidentes. La implementación más popular de red inalámbrica para entornos de redes de área local es el estándar IEEE 802.11 también popularmente conocidas como redes Wi-Fi [16].

Una red de área local inalámbrica puede definirse como a una red de alcance local que tiene como medio de transmisión el aire. Por red de área local entendemos una red que cubre un entorno geográfico limitado, con una velocidad de transferencia de datos relativamente alta (mayor o igual a 1 Mbps tal y como especifica el IEEE), con baja tasa de errores y administrada de forma privada. Por red inalámbrica entendemos una red que utiliza ondas electromagnéticas como medio de transmisión de la información que viaja a través del canal inalámbrico enlazando los diferentes equipos o terminales móviles asociados a la red. Estos enlaces se implementan básicamente a través de tecnologías de microondas y de infrarrojos.[20]

En las redes tradicionales cableadas esta información viaja a través de cables coaxiales, pares trenzados o fibra óptica. Una red de área local inalámbrica, también llamada Wireless LAN (WLAN), es un sistema flexible de comunicaciones que puede implementarse como una extensión o directamente como una alternativa a una red cableada. Este tipo de redes utiliza tecnología de radiofrecuencia minimizando así la necesidad de conexiones cableadas. Este hecho proporciona al usuario una gran movilidad sin perder conectividad.[20]

El atractivo fundamental de este tipo de redes es la facilidad de instalación y el ahorro que supone la supresión del medio de transmisión cableado. Aun así, debido a que sus prestaciones son menores en lo referente a la velocidad de transmisión que se sitúa entre los 2 y los 10 Mbps frente a los 10 y hasta los 100 Mbps ofrecidos por una red convencional, las redes inalámbricas son la alternativa ideal para hacer llegar una red tradicional a lugares donde el cableado no lo permite, y en general las WLAN se utilizarán como un complemento de las redes fijas.[20]

4.2.1 Aplicaciones de las WLAN

Las aplicaciones más típicas de las redes de área local que podemos encontrar actualmente son las siguientes:

- Implementación de redes de área local en edificios históricos, de difícil acceso y en general en entornos donde la solución cableada es inviable.
- Posibilidad de reconfiguración de topología de la red sin añadir costes adicionales.
- Esta solución es muy típica en entornos cambiantes que necesitan una estructura de red flexible que se adapte a estos cambios.
- Redes locales para situaciones de emergencia o congestión de la red cableada.

Estas redes permiten el acceso a la información mientras el usuario se encuentra en movimiento, habitualmente esta solución es requerida en hospitales, fábricas, almacenes.

Generación de grupos de trabajo eventuales y reuniones ad-hoc. En estos casos no valdría la pena instalar una red cableada. Con la solución inalámbrica es viable implementar una red de área local, aunque sea para un plazo corto de tiempo.

En ambientes industriales con severas condiciones ambientales este tipo de redes sirve para interconectar diferentes dispositivos y máquinas.

Interconexión de redes de área local que se encuentran en lugares físicos distintos. Por ejemplo, se puede utilizar una red de área local inalámbrica para interconectar dos o más redes de área local cableadas situadas en dos edificios distintos.[20]

El grado de complejidad de una red de área local inalámbrica es variable, dependiendo de las necesidades a cubrir y en función de los requerimientos del sistema que queramos implementar podemos utilizar diversas configuraciones de red.

La configuración más básica es la llamada de igual a igual o ad-hoc y consiste en una red de dos terminales móviles equipados con la correspondiente tarjeta adaptadora para comunicaciones inalámbricas. En la figura 3 mostramos un ejemplo. Para que la comunicación entre estas dos estaciones sea posible hace falta que se vean mutuamente de manera directa, es decir, que cada una de ellas esté en el rango de cobertura radioeléctrica de la otra. Las redes de tipo ad-hoc son muy sencillas de implementar y no requieren ningún tipo de gestión administrativa.[20]

4.2.2 Arquitectura de las redes 802.11

El modelo desarrollado por el grupo de trabajo del IEEE 802.11 se basa en sistemas divididos en células y permiten dos tipos de arquitectura:

Ad-hoc: La estructura de una red ad hoc puede ser tan simple como que dos computadoras compartan información de ida y vuelta, pero la complejidad aumenta si las computadoras necesitan tener acceso a Internet. Con esta configuración, una computadora actúa como servidor, que comparte el acceso a Internet con los equipos que están conectados a esta de forma inalámbrica. La computadora que actúa como servidor normalmente se conectará directamente a un cable o módem DSL vía Ethernet.[16]

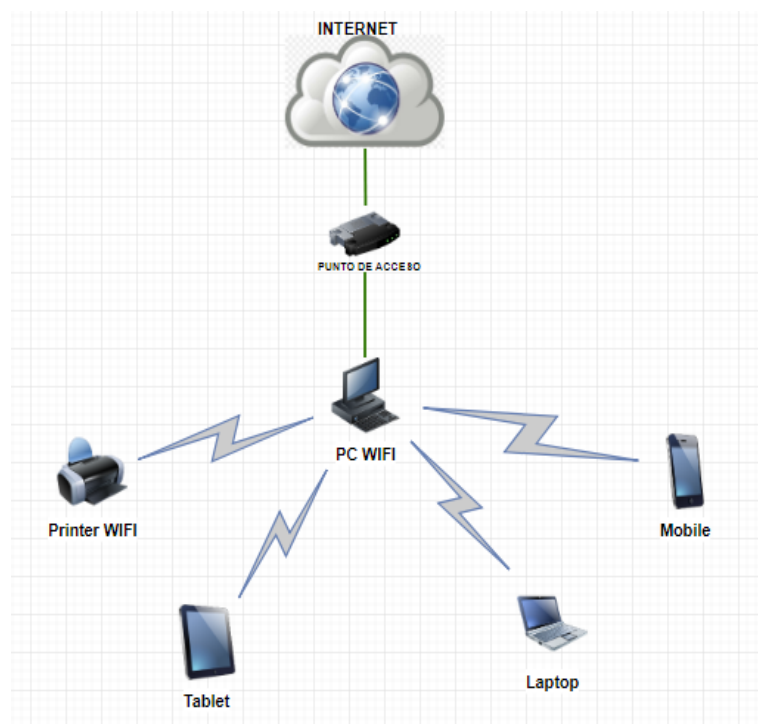


Figura 3. Red Ad Hoc.

Infraestructura: todos los dispositivos realizan la comunicación inalámbrica a través de un punto de acceso o AP.

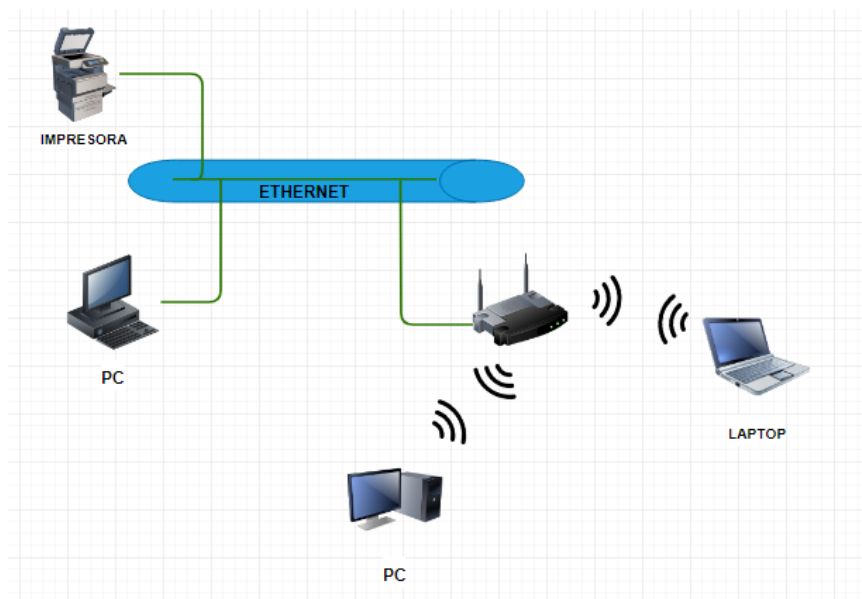


Figura 4. Red Tipo Infraestructura

4.2.3 Sistemas de detección de intrusiones en redes Wifi

Un sistema inalámbrico de detección de intrusiones (WIDS) está basado en un conjunto de sensores y un núcleo que recibe toda la información proporcionada en todas las áreas de cobertura de los sensores inalámbricos.

Pueden ser:

- Centralizado: Basado en la combinación de sensores individuales los cuales recopilan y remiten todos los datos 802.11 a un analizador central, donde los datos son almacenados y procesados [16].
 - Ventajas: Permite una fácil administración de protección a áreas grandes de redes 802.11. Expansiones a la red afectan solamente a él analizador. Permite una gran visión de lo que ocurre en todas las partes de las redes 802.11.
 - Desventajas: Si el analizador falla, los sensores se vuelven inútiles y toda la red queda sin la protección [16].
- Distribuido: Suele incluir uno o más dispositivos que se encargan tanto de la recolección y procesamiento de la información de los IDS.
 - Ventajas. No hay un solo punto de fallo.
 - Desventajas. El costo de sensores con alta capacidad de procesamiento puede llegar a ser exagerado cuando muchos sensores son requeridos.

La administración de múltiples sensores de procesamiento de información puede ser más difícil que la de un modelo centralizado.

Expansiones en la red provocará una reprogramación en todos los sensores [16].

4.2.3.1 Técnicas en la detección de intrusiones en redes wifi

Entre ellas tenemos por uso indebido, por firmas, por análisis este se basa en patrones de comportamiento y anomalías:

- **Detección de uso indebido**

Consiste en la detección de uso indebido (patrones) para reconocer ataques previamente conocidos. La mayoría de los IDS disponibles en el mercado son de este tipo, donde algunos de los más populares son SNORT y BRO IDS. Sin embargo, la detección de uso indebido puede acarrear varios problemas como la incapacidad de detectar los ataques nuevos y sus variantes [16]. La detección de usos indebidos se puede implementar de las siguientes formas:

- **Firmas Simples**

La detección de firmas compara los eventos que ocurren, con las cadenas o firmas almacenadas en una base de datos de escenarios de ataque en busca de coincidencias. Su principal inconveniente es la necesidad de desarrollar e incorporar a la base de datos una firma nueva para cada nuevo tipo de ataque o vulnerabilidad descubierta, en el caso de SNORT y SURICATA al hacer uso de OINKMATER se compensa dicho inconveniente debido a que esta aplicación ayuda a la actualización de las firmas [16].

- **Análisis de Transición de Estados**

Se crean a partir de la construcción de una máquina de estados finitos. Los escenarios de ataques se representan como una secuencia de transiciones que caracterizan la evolución del estado de seguridad de un sistema. Cuando el autómata alcanza un estado considerado como una intrusión, se lanza la alarma [16].

Algunas ventajas son:

- Las transiciones ofrecen una forma de identificar una serie de patrones que conforman un ataque.

- El diagrama de estados define la forma más sencilla posible de definir un ataque. Así, el motor de análisis puede utilizar variantes de este para identificar ataques similares.
- El sistema puede detectar ataques coordinados y lentos.
- Sin embargo, presentan algunas desventajas:
- El lenguaje utilizado para describir los ataques es demasiado limitado, y en ocasiones puede resultar insuficiente para recrear ataques más complejos.
- El análisis de algunos estados puede requerir más datos del objetivo, por parte del motor. Esto reduce el rendimiento del sistema.

- Detección de Anomalías

Los IDS basados en anomalías, por otro lado, detectan desviaciones en el comportamiento esperado o normal de los sistemas y las redes, las cuales pudieran constituir intentos de ataques. Por tal motivo, los IDS basados en el descubrimiento de anomalías son potencialmente capaces de detectar los ataques existentes y los nuevos, sin la necesidad de ser pre configurados o actualizados de ninguna manera [16].

Los eventos de interés para los IDS basados en anomalías pueden estar definidos de dos maneras:

- Modelos Estadísticos: Los modelos estadísticos hacen uso de variables o características para estimar el comportamiento de la red, pero necesitan de un periodo de entrenamiento para determinar cuál es el comportamiento esperado o normal de la red.
- Modelos basados en la especificación: Este método se basa en describir el comportamiento normal y expresarlo a manera de especificaciones. Las desviaciones a estas especificaciones son tratadas como un evento anormal, pudiéndose tratar de una intrusión. Una especificación puede estar basada en la transición de estados que puede ocurrir durante el comportamiento normal y/o por una expresión específica basada en políticas de seguridad previamente declaradas [16].

Detectar intrusiones en redes wifi dada su complejidad se convierte en un reto muy grande ya sea por los puntos de acceso o como estas anomalías en la red se comportan de manera no determinada, dependiendo el tráfico que envíe o reciba cada

nodo, así como la interferencia en el medio debido al número de nodos conectados o a fallas en la transmisión [16].

El IDS en una red wifi debe poder detectar eventos que se desvíen de su comportamiento habitual y determinar si ese comportamiento inusual se debe a una posible intrusión o a la interferencia en el medio de comunicación. Adicionalmente, debe detectar ataques en los protocolos usados en la red inalámbrica, por lo cual necesita concentrarse en protocolos de la capa física y de enlace de datos para detectar potenciales ataques [16].

4.2.3.2 Tipos de ataques y amenazas en redes inalámbricas

Se puede definir como amenaza a todo elemento o acción capaz de atentar contra la seguridad de la información. La presencia de una amenaza es una advertencia de que puede ser inminente el daño a algún activo de la información, o bien es un indicador de que el daño se está produciendo o ya se ha producido.

La identificación de amenazas requiere conocer los tipos de ataques, el tipo de acceso, la forma operacional y los objetivos del atacante. Las consecuencias de los ataques se podrían clasificar en:

- Data Corruption: La información que no contenía defectos pasa a tenerlos.
- Denial of Service (dos): Servicios que deberían estar disponibles no lo están.
- Leakage: Los datos llegan a destinos a los que no deberían llegar [21].

Un "ataque" consiste en aprovechar una vulnerabilidad de un sistema informático, con propósitos que son ignorados por el operador del sistema y que, por lo general, causan algún daño. Entre los Tipos de ataque más comunes están:

Ingeniería Social: Es la manipulación de las personas para convencerlas de que ejecuten acciones o actos que normalmente no realizan para que revele todo lo necesario para superar las barreras de seguridad. Si el atacante tiene la experiencia suficiente, puede engañar fácilmente a un usuario en beneficio propio [13].

- Ataques de Monitorización: Este tipo de ataque se realiza para observar a la víctima y su sistema, con el objetivo de obtener información, establecer sus vulnerabilidades y posibles formas de acceso futuro [13].
- Decoy: Son programas diseñados con la misma interfaz que otro original. En ellos se imita la solicitud de un para ingresar el usuario y contraseña luego el programa guardará esta información y dará paso a las actividades normales del

sistema. La información recopilada será utilizada por el atacante para futuras “visitas” [22].

- Scanning (Búsqueda): El escaneo, como método de descubrir canales de comunicación susceptibles de ser explotados, lleva en uso mucho tiempo. La idea es recorrer (escanear) tantos puertos de escucha como sea posible, y guardar información de aquellos que sean receptivos o de utilidad para cada necesidad en particular [22].
- Sniffing: Obtener la información sin modificarla. Aquí, además de interceptar el tráfico de red, el atacante ingresa a los documentos, mensajes de correo electrónico y otra información guardada, realizando en la mayoría de los casos una descarga de esa información, para luego hacer un análisis exhaustivo de la misma.
- Utilización de Backdoors: Las puertas traseras son trozos de código en un programa que permiten a quien las conoce saltarse los métodos usuales de autenticación para realizar ciertas tareas. Habitualmente son insertados por los programadores del sistema para agilizar la tarea de probar código durante la fase de desarrollo.

4.2.3.3 Descripción de algunos ataques procedentes de internet

Esta sección quiere demostrar lo fácil que resulta entender las estrategias básicas asociadas a ataques de los piratas informáticos conocidos como “hackers”.

Los administradores de la red deben reconocer los principios asociados a los ataques procedentes de Internet a fin de tomar una decisión apropiada respecto a los equipos de seguridad con los que cuentan, así como los riesgos que con éstos se pretende atenuar. Debido a que es imposible tratar íntegramente todas las amenazas procedentes de Internet, nos centraremos en los ataques más representativos.

Los ataques dos han evolucionado a través del tiempo, cada uno de ellos se generan a partir de la explotación de vulnerabilidades que se puedan encontrar en una red o sistema. Los ataques modernos llegan incluso a ser una combinación de varios tipos de ataques, volviéndose más fuertes a la hora de actuar [23].

Entre los tipos de ataques dos están:

- Ddos: este tipo de ataque es una variante de los ataques dos. En la actualidad resulta más rentable comercialmente producirlos debido a su efectividad y

capacidad de denegación de Servicios Distribuida. La cual implica cientos o miles de estaciones que sincronizan una herramienta dos y lanzan el ataque coordinando un destino en una hora específica. Las estaciones utilizadas para realizar el ataque pueden ser voluntarias o infectadas sin darse cuenta.[23]

- Volumétricos: Este tipo de ataque buscar consumir y saturar el ancho de banda de la red, generando exceso de tráfico, lo cual produce una congestión en la red. Puede ser dirigido a una la red (LAN) o una red WAN.[23]
- Ataque dos por consumo de recursos: el atacante intentar consumir los recursos asignados a un servidor, estos pueden ser: procesamiento, ancho de banda, memoria, almacenamiento. Se lo realiza a través de una generación excesiva de peticiones, las cuales intentar responder el servidor, llegando a tal punto que se sature.[23]
- Ddos Botnets: El ataque consiste en varios equipos que han sido infectados con un programa maligno el cual generar gran cantidad de peticiones hacia un objetivo en común en un momento determinado, sin que los propietarios de los hosts infectados se den cuenta. A los hosts infectados también se los conoce como zombis.[23]

Ataques a nivel de infraestructura

Esta clasificación abarca todos aquellos ataques que han enfocado su diseño basándose en las vulnerabilidades de los protocolos pertenecientes a las capas de red y transporte del modelo OSI (capa 3 y 4 respectivamente). Por lo general estos son los protocolos que con mayor frecuencia se han utilizado para realizar ataques dos, debido a su baja complejidad de construcción y ejecución.[23]

Ataques a capa de Aplicación

Aquí se podrán encontrar aquellos ataques que han basado su diseño identificando y aprovechando los puntos débiles de la capa de aplicación, con la finalidad de producir una degradación o interrupción del servicio, Dentro de esta clasificación el protocolo mayormente explotado es HTTP, sin embargo, protocolos como NTP, SMTP o DNS son también ampliamente utilizados.[23].

En el último año en el cuarto trimestre del 2018 Kaspersky emitió un informe sobre las actividades ddos que bajo su observación notaron que disminuyeron en un 13% en comparación con el año anterior. La disminución de la cantidad de ataques se observó en todos los trimestres, excepto el tercero, en el que hubo muchos más

ataques que en 2017, debido a que septiembre de 2018 fue un mes de actividad anormal. Los indicadores tuvieron su punto más bajo en el cuarto trimestre, cuando la cantidad de ataques fue del 70% en comparación con 2017.

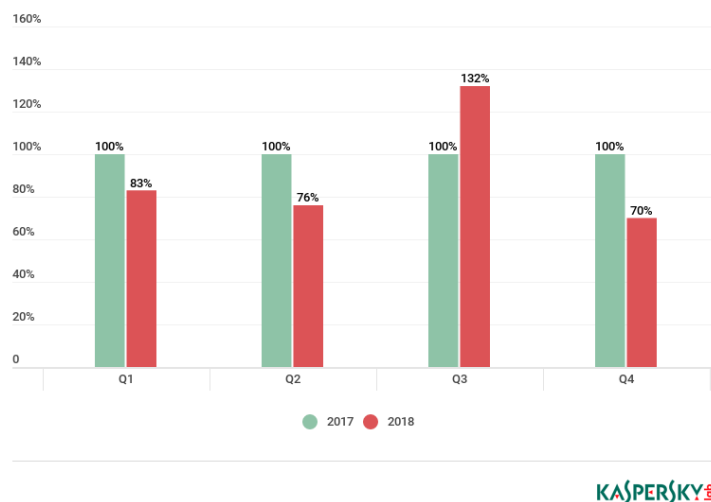


Figura 5. Comparación trimestral de números de ataques DDos

Fuente: Kaspersky Labs[24]

En el último año (julio 2018- julio 2019) las tecnologías de Kaspersky bloquearon 45 intentos de infección en América Latina cada segundo. Eso significa que **cada segundo, un usuario en América Latina sufre un ataque.**

Entre los países más atacados a nivel global, solamente aparecen dos de América Latina: Brasil (puesto 7) y México (puesto 11).

Las principales amenazas a las que se enfrentan los usuarios en América Latina son las infecciones que ocurren con la piratería en Windows de 64 bits y el adware que invade la privacidad del usuario con anuncios invasivos durante la navegación. [24]

5. MATERIALES Y METODOS

En esta sección, se da a conocer los métodos, técnicas y metodologías necesarias para la realización de este trabajo, porque a través de ellas, se recolectó información relevante para realizar un proceso investigativo y de esta forma cumplir con éxito los objetivos planteados al comienzo de esta investigación.

Métodos

- Método deductivo: permitió la comprensión de conceptos en el ámbito de la seguridad en una red, y el tipo de ataques a enfocarse en lo relacionado a la detección de intrusos en una red.
- Método experimental: con este método se pone a prueba el sistema de detección de intrusos, utilizando un entorno virtual en el cual se lo someterá a ataques y se estará registrando los datos obtenidos por el IDS.
- Método de revisión sistemática de Barbara Kitchenham: este método consiste la formulación de una cadena de búsqueda la cual permite indagar en los diferentes repositorios y bases científicas con el fin de recopilar, efectuar la selección de información relevante respecto al tema de investigación sobre sistemas de detección de intrusos open source.

Técnicas

- Revisión Bibliográfica: Mediante esta técnica sirvió para el sustento teórico y desarrollo del proyecto, apoyándose en: consultas de libros, artículos científicos, páginas web confiables entre otros.
- Tutorías: Por medio de las tutorías se pudo corregir errores o solucionar inconvenientes que aparezcan en el avance del proyecto, esta técnica se basa en la colaboración por parte del docente tutor.

Metodología de desarrollo

Para la configuración de la herramienta IDS se llevó como metodología las fases establecidas en el alcance del proyecto: en la primera fase se realiza un análisis de la situación actual de los sistemas de detección de intrusos, en la segunda fase se creó un entorno virtual para la implementación de la herramienta y finalmente en la tercera fase se realizó la configuración y la puesta en marcha del IDS en un entorno virtual, en el que fue sometido a pruebas, bajo ataques específicos.

6.RESULTADOS

En este apartado se establece los resultados de cada fase del proyecto de titulación de acuerdo con los objetivos planteados.

En la fase 1 se realiza un análisis el estado actual de herramientas open source para la detección de intrusos, considerando sus características realizando una comparativa entre soluciones IDS más conocidas, en la fase 2 se realiza el diseño un esquema de red y entorno virtual para la implementación de la herramienta IDS Suricata, tomando en cuenta tanto el hardware como software para su creación y finalmente en la fase 3 se realiza la evaluación de la herramienta.

Fase 1: Analizar el estado actual de herramientas open source para la detección de intrusos.

En esta fase se realizó una revisión sistemática de literatura con el fin de obtener información sobre el estado actual de los sistemas de detección de intrusos, se realizó una comparativa y análisis del sistema Suricata frente a otras dos soluciones en base a sus características, así como establecer las ventajas y desventajas de los IDS.

Comparativa de IDS

Los sistemas de detección de intrusos deben cumplir ciertos requisitos para que su trabajo sea confiable y puedan desarrollar su labor de manera efectiva. Cualquier sistema de detección de intrusos indistintamente del mecanismo en que esté basado deberá contar las siguientes características mostradas en la Tabla II.

TABLA II COMPARATIVA IDS

CARACTERÍSTICAS	SNORT	SURICATA	BRO
Capacidad para detectar un amplio espectro de ataques	SI	SI	SI
Eficiencia en la detección sin afectar su eficacia	SI	SI	SI
Facilidad de gestión.	SI	SI	NO
Escalabilidad	SI	SI	SI
Dinamismo	SI	SI	NO
Plataforma soportada	Win, Macos, Unix	Win, Macos, Unix	Unix Like system, Mac OS
Licencia	GNU GPL	GNU GPL	BSD

	V2	V2	
Característica de IPS	SI	SI	NO
PGP signed	SI	SI	NO
DNP3	SI	SI	SI
Soporte a redes de alta velocidad	Medio	Alto	Alto
GUÍA de configuración	Yes	Yes	No
Offline Analysis	Sí, para varios archivos	Sí, para un solo archivo	Sí, para un solo archivo
Threads	Single Thread	Multithreaded	Single Thread
IPV6	SI	SI	SI
Instalación y el Despliegue	Fácil	Fácil	Conocimientos amplios en comandos para Unix

Análisis del cuadro comparativo

Tomando en cuenta las características de cada herramienta se concluye:

La capacidad de detección se define como el abanico de ataques que puede ser analizado por el detector es decir depende en gran medida del número de características de entrada que sean tomadas para la clasificación ya que, a mayor cantidad de éstas, es mayor el espectro de ataques que puede ser analizado en este caso Snort y Suricata cumplen con esta característica debido a que su despliegue se lo realiza en redes amplias.

En cuanto a eficiencia las tres herramientas no presentan problemas siempre y cuando su configuración sea correcta sin ambigüedades al momento de la creación de sus reglas y características con las que trabaja dado que, si el número de características de entrada es muy grande, los clasificadores consumirán mayor cantidad de tiempo y recursos. Por tanto, para lograr mayor eficiencia de detección, la cantidad de variables de entrada debe ser minimizada y priorizada.

La facilidad con que un sistema se administra es importante debido a que la fase de aprendizaje no tomará más del tiempo necesario, en este caso tanto Snort como Suricata son más fáciles de aprender a administrar, lo que es contrario al IDS Bro que para este se necesita de amplios conocimientos en cuanto al uso de comandos.

Snort a pesar de poseer características que lo hacen un motor de prevención y detección de intrusos tiende a producir un mayor número de falsos positivos, lo cual, en un entorno extenso de análisis de tráfico, disminuye el rendimiento de este.

En lo que respecta a la gestión Bro es un poco más complicado dado que este no cuenta con GUI, a diferencia de Snort y Suricata que a través de herramientas complementarias se pueden visualizar resultados, tráfico, lo que no es posible en Bro.

Con la información obtenida, se ha logrado evidenciar las características de cada una de las herramientas lo que permitió, concluir que Suricata en si cumple con las particularidades adecuadas para ser implementado en una red de datos pequeña o de gran tamaño dado que es una herramienta escalable, IDS hace uso de las funciones multi-hilo de manera que solo con ejecutarse en una instancia el sistema balanceará su carga entre todos los procesadores disponibles, evitando incluso alguno de ellos si así lo especificamos en su configuración, esta herramienta es capaz de procesar un ancho de banda de hasta 10 gigabits por segundo sin que ello repercuta sobre el rendimiento de la red.

Suricata también controla los archivos que viajan por la red, siendo capaz de identificar un gran número de formatos diferentes, así como realizar comprobaciones MD5 para comprobar que no ha sido modificado y también es capaz de extraer temporalmente ciertos archivos para identificar posible malware escondido.

Ventajas y desventajas de los sistemas de detección de intrusos

De acuerdo con la información obtenida al realizar una revisión sistemática de literatura (ver Anexo 1) se estableció algunas ventajas y desventajas de los sistemas de detección de intrusos, teniendo en cuenta que las desventajas en la mayoría de los casos surgen debido a que no se tiene en claro lo que protegerá, falta de conocimiento, falta de recursos tanto de hardware como software.

Ventajas de los sistemas de detección de intrusos

Las ventajas de un IDS se presentan de acuerdo con el tipo de red en el que se desplegará, sea está en una red Wifi o cableada en esta última es más seguro su despliegue.

- Las soluciones de código abierto son más económicas, pero la configuración y la administración son más complicadas si se despliegan en entornos amplios, aunque se puede resolver ubicándolos en puntos estratégicos.

- Son un mecanismo pasivo de detención ya que su tarea fundamental es alertar.
- Existen IDS que soportan múltiples plataformas como Linux, Mac OS, Solaris, AIX y Windows con emulador POSIX.
- Para los sistemas basados en Linux, el IDS Samhain se integra con la auditoría del Kernel para encontrar información sobre archivos modificados como usuario, fecha y hora, esta operación no es posible en otros sistemas operativos.
- La existencia de Framework de aplicaciones web de como IRONBEE para firewalls proporciona un alto grado de flexibilidad para construir el firewall según las necesidades de una organización.
- Al instalar un IDS en un sistema inalámbrico como 802.16 proporciona una oportunidad al administrador de descubrir y controlar el tráfico de anomalías, ataques y amenazas.
- Un IDS es útil para supervisar y controlar a los intrusos, el sistema IDS puede ser integrado con Honeypots para juntar la información sobre intrusiones y rastrearlos respectivamente.
- Hay IDS que son adecuados para funcionar en entornos de alta velocidad y capaz para capturar datos de redes Gbps.
- IDS basados en DES (Eventos discretos del sistema) han demostrado ser un mecanismo eficaz para detectar ataques a la red sin necesidad de modificar el protocolo, cifrar o instalar hardware propietario.
- Los IDS basados en firmas generan baja tasa de falsos positivos.
- Los IDS basados en anomalías (se utiliza para detectar ataques desconocidos) se pueden utilizar para adquirir la información de la firma utilizada por los IDS mal escritos y no es necesario escribir reglas.
- Los IDS se utilizan para diferentes campos y aplicaciones no solo en la computadora o redes, pero para poder ser utilizados con éxito deben ser exactos en el sentido que no pueden dar información errónea.
- La principal ventaja de los IDS basados en anomalías es su capacidad para detectar ataques desconocidos que no tienen firmas existentes.
- Un IDS en una red 802.11 centralizada permite una fácil administración de protección a áreas grandes, así como una gran visión de lo que ocurre en todas las partes de la red.
- Algunos IDS se pueden adaptar para el soporte de protocolos SCADA.

- Los IDS se consideran como una de las principales herramientas disponibles en para ser implementados en entornos críticos.

Desventajas de los sistemas de detección de intrusos

Por lo general las desventajas surgen por:

- No existe un parche para la mayoría de los bugs de seguridad, se requiere de un tiempo de espera para su solución por parte de las comunidades que lo desarrollan.
- Se producen falsas alarmas, si las alertas y las reglas no están bien configuradas.
- No es sustituto para un Firewall, una auditoría de seguridad.
- Alta tasa de falsas alarmas dado que no es posible cubrir todo el ámbito del comportamiento de un sistema de información durante la fase de aprendizaje.
- El sistema puede sufrir ataques durante la fase de aprendizaje, con lo que el perfil de comportamiento contendrá un comportamiento intrusivo el cual no será considerado anómalo.
- Los falsos positivos podrían representar la auto negación de servicio o un problema general con la red, pues de manera automática, aplicaría las reglas en su firewall interno para poder mitigar “el ataque”.
- Los IDS que constan de una sola instancia para procesar información tiene dificultades al procesar el tráfico en redes más rápidas.
- La configuración predeterminada de algunas soluciones no está diseñada para redes grandes.
- No todos los IDS son compatible con la función IPS.
- Los sistemas IDS son lo suficientemente inteligentes como para detectar malware o actividades sospechosas mediante la supervisión de todo el sistema, por lo tanto, no producen alarmas o denuncias contra tales actividades.
- Los ataques que ocurren en redes LAN inalámbrica son más difíciles de detectar que sus contrapartes cableadas debido a la movilidad, las obstrucciones causadas por obstáculos en el camino, el medio ruidoso, la cobertura limitada del AP y las limitaciones de la potencia de procesamiento de los nodos inalámbricos.
- El consumo de CPU y memoria de un IDS depende en gran medida del procedimiento de este.

- Al hacer uso de un IDS con un Honeypot sus probabilidades de cubrir todos los ataques son aceptables en redes LAN.
- Los IDS basados en firmas solo pueden detectar los ataques que previamente están almacenados en la base de datos, no pueden detectar los ataques online.
- Los IDS basados en anomalías generan alta tasa de falsas alarmas, definir el conjunto de reglas es difícil, no precisar la naturaleza del ataque tiene una baja tasa de detección.
- Para que los IDS ayuden a evitar posibles ataques en la red es necesario mantener actualizado el Software del equipo.
- La principal desventaja de los IDS híbridos es la sobrecarga computacional del uso de la coincidencia de firmas y la detección de anomalías para analizar las conexiones de red entrantes.
- Si el analizador del IDS falla, sus sensores se vuelven inútiles y toda la red queda sin la protección.
- En entornos distribuidos la administración de múltiples sensores de procesamiento de información puede ser más difícil que la de un modelo centralizado.

Fase 2: Diseñar un esquema de red doméstica y entorno virtual de equipos.

En esta fase el esquema de red a utilizar será una Wlan doméstica para la cual se realiza la elección del software y hardware en el que se desplegará la herramienta

Para la implementación de un IDS, en un entorno real sea en servidores o en una sola estación el equipo debe contar con los siguientes aspectos:

En servidores:

TABLA III. ELEMENTOS NECESARIOS PARA SERVIDORES [25]

Dato	Valor mínimo	Valor recomendado
Memoria RAM	Intel Q6600 2.4 ghz, 4GB RAM, 8 GB	La memoria RAM debe ser con una capacidad mayor al ancho de banda de la red o porción de red a proteger.
Tamaño Almacenamiento	50 Gb	Dependerá de los tipos de formatos de salidas que se requiera, por ejemplo, si activamos como salidas unified2, pcap, o log, el sistema volcará las alertas en el directorio propuesto y esto consumirá la capacidad de almacenamiento del sistema dedicado a IDPS. La capacidad requerida mínima es de 1 Tb.

Estaciones cliente:

TABLA IV. ELEMENTOS NECESARIOS EN ESTACIONES CLIENTE [25]

Dato	Valor mínimo	Valor recomendado
Procesador	Intel i3 2.13 ghz	Varios núcleos
Memoria RAM	4 GB	8 GB
Tamaño Almacenamiento	20 GB	50 GB

Para la conectividad

Se recomienda también que se tengan las tarjetas NIC con capacidad de acuerdo con el tráfico que existe en la red normalmente incluido el tiempo donde más actividad existe dentro de la red. Con esto las tarjetas NIC pueden operar acorde al tráfico de red a la cual el sensor está sometido [19].

Para crear el laboratorio virtual, se requiere un equipo con las siguientes características:

- Equipo: Toshiba
- Procesador: cuatro núcleos procesador AMD.
- RAM: 8 GB
- Almacenamiento 1TB
- Tipo de sistema: Windows 10, de 64 bits
- Tarjeta de red: Wireless-N 2230

Al configurar un entorno virtual formado por varias máquinas en una red WLAN inalámbrica se utilizará una aplicación de virtualización así, se tendrá un entorno controlado para la implementación y puesta en marcha del IDS.

Existen varias plataformas de virtualización muy populares, como por ejemplo VMware esxi, Microsoft Hyper-V o Citrix xenserver. Todas estas tienen alguna versión gratuita, pero en general son de pago, al elaborar el escenario se ha escogido una herramienta Open Source de las más populares como es Virtualbox.

Virtualbox: es una de las muchas soluciones de virtualización existentes y actualmente es propiedad de Oracle. Existe una versión con licencia GPL denominada

virtualbox OSE (Open Source Edition), que se puede utilizar libremente. Esta herramienta permite a su vez crear máquinas virtuales de varios sistemas operativos como GNU/Linux, Mac OS X, OS/2Warp, Microsoft Windows, y Solaris/Open Solaris, tanto en arquitecturas de 32 como de 64 bits.

En el escenario se utilizó los siguientes elementos ver en la Tabla V.

TABLA V. ELEMENTOS NECESARIOS PARA MONTAR EL ESCENARIO

Descripción	Red	Características	S.O.	Herramientas
Máquina Anfitrión	Red Interna Y Externa	Memoria RAM 16 GB 8 procesadores Disco Duro de 1 TB	Windows 10	Virtual Box Servidor HTTP Xampp Mysql
Máquina IDS SURICATA	Red interna	Memoria RAM 4 GB 1 procesadores Disco Duro de 60 GB	Debian 9	Suricata
Máquina Atacante Kali Linux	Red interna	Memoria RAM 4 GB 1 procesador Disco Duro de 30 GB	Kali Linux	Metasploit Ettercap Nmap
Router	WLAN	Velocidad inalámbrica (300 Mbps)		Punto de acceso a la red

Utilizando el software de Oracle VM VirtualBox en su versión 6.0.0 se crearon 3 máquinas virtuales con los recursos necesarios para su funcionamiento, el sistema operativo en el cual se aloja el IDS es Debian 9 sobre el cual se procedió con la instalación y configuración del IDS de estudio, así como las herramientas y librerías necesarias para el registro de ataques generados a la red.

Definir el esquema de red

El objetivo del entorno de pruebas es ejecutar ataques informáticos hacia las máquinas que se encuentran en la Wlan con el fin de generar alarmas con los ataques dirigidos contra las máquinas dentro de la red.

En el presente trabajo el esquema utiliza los siguientes componentes.

TABLA VI. COMPONENTES DEL ESQUEMA DE RED

TIPO	RED	IP	SISTEMA OPERATIVO
SALIDA INTERNET	WAN	172.16.XX.X	N/A
IDS SURICATA	LAN	172.16.XX.X	Debian 9
Kali Linux	LAN	172.16.XX.X	Atacante
Móvil atacante	Wifi	172.16.XX.X	Atacante

Diseño del entorno virtual de equipos.

El esquema lógico cuenta con varios elementos de red que representan: la red interna (LAN), y red externa (WAN). Cabe indicar que dentro de cada segmento están instalados las máquinas virtuales con los respectivos sistemas operativos, el IDS SURICATA y los clientes para simular los ataques internos.

La máquina virtual está configurada para proveer de internet en modo puente, esta configuración permite conectar forma que todas las MV que estén en esta red sean visibles al IDS.

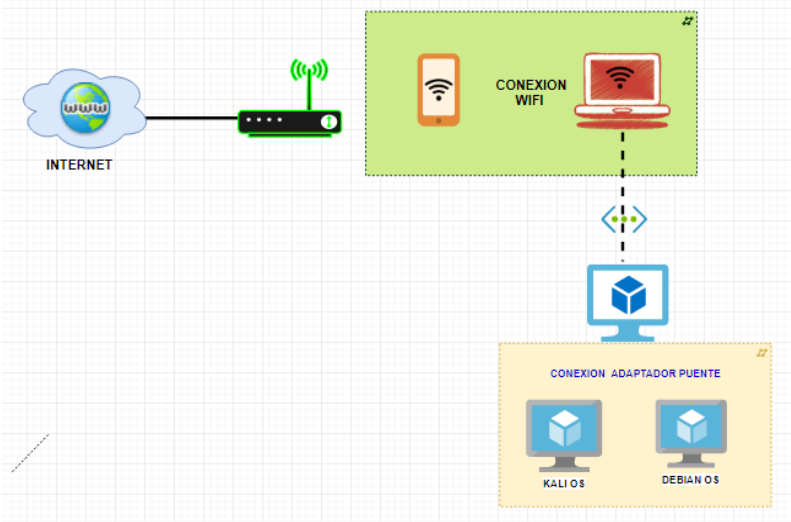


Figura 6. Esquema lógico del escenario.

El objetivo del escenario es recolectar información sobre eventos sospechosos en la red y generar las alertas correspondientes por el IDS Suricata.

Fase 3: Configuración y evaluación la herramienta Suricata.

Establecer las reglas necesarias para configuración sistema IDS.

Una vez instalado el sistema IDS Suricata (ver instalación Anexo 2) se ingresa en el terminal como root para poder acceder al archivo de configuración.

Paso 1: Para la configuración del archivo `suricata.yaml` se ingresa al directorio `/etc/suricata/` y se abre con un editor en este caso `gedit` `suricata.yaml` ver figura 8. Aquí se configura las direcciones IP con las que trabajará el motor de detección.

Bajo la sección "`vars`", se encuentran las variables más importantes utilizadas por Suricata "`HOME_NET`" esta variable debe apuntar a la red local para ser inspeccionado por Suricata "`!$ HOME_NET`" (asignado a `EXTERNAL_NET`) se refiere a cualquier otra red. Ver figura 7 y 8.

```
vars:
# more specific is better for alert accuracy and performance
address-groups:
HOME_NET: "[172.16.12.0/24,172.16.12.1/24,172.16.12.8/32,172.16.12.15/32]"
#HOME_NET: "[192.168.0.0/16]"
#HOME_NET: "[10.0.0.0/8]"
#HOME_NET: "[172.16.0.0/12]"
#HOME_NET: "any"

EXTERNAL_NET: "!$HOME_NET"
#EXTERNAL_NET: "any"
```

Figura 7. Configuración de variables

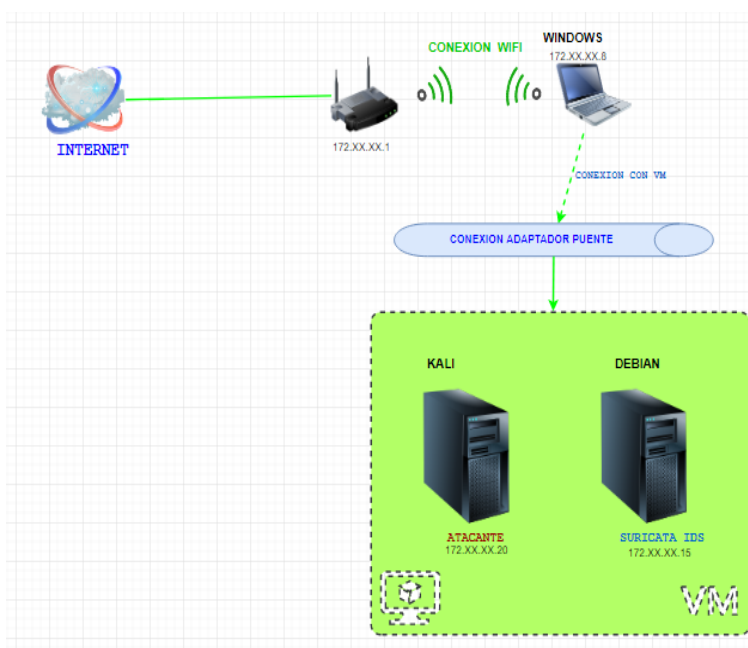


Figura 8. Topología Direccionamiento de red

Paso 2 Establecer qué reglas o firmas estarán activas dentro del archivo de configuración.

Las firmas juegan un papel muy importante en Suricata ver Figura 8, su funcionamiento necesita de estas, actualmente para las versiones 4.0 en adelante cuenta con su propia función de actualización `suricata-update`.

```
default-rule-path: /var/lib/suricata/rules
rule-files:
- suricata.rules

##
## Advanced rule file configuration.
##
## If this section is completely commented out then your configuration
## is setup for suricata-update as it was most likely bundled and
## installed with Suricata.
##

#default-rule-path: /var/lib/suricata/rules

#rule-files:
# - botcc.rules
# - botcc.portgrouped.rules
# - ciarmy.rules
# - compromised.rules
# - drop.rules
# - dshield.rules
## - emerging-activex.rules
# - emerging-attack_response.rules
# - emerging-chat.rules
# - emerging-current_events.rules
```

Figura 9. Reglas en el archivo de configuración

Un ejemplo de regla en Suricata:

```
Drop tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"ET TROJAN Likely Bot Nick in IRC
(USA +..)"; flow:established,to_server; flowbits:isset,is_proto_irc; content:"NICK ";
pcre:"/NICK .*USA.*[0-9]{3,}/i"; reference:url,doc.emergingthreats.net/2008124;
classtype:trojan-activity; sid:2008124; rev:2;)
```

En este ejemplo, se puede observar las partes de una regla como son: la acción (rojo), el encabezado (verde) y las opciones (azul).

- **Cabecera:** (cabecera de regla) Tipo de regla (acción), protocolo, dirección-origen, puerto origen, dirección-destino y puerto destino.
- **Opciones:** (opciones de regla) cuenta con más de 50 opciones disponibles entre ellas: mensaje a mostrar (`msg`), contenido (`content:`), `sid`: Id de regla (`sid:`),
- **Número de revisión** (`rev:`), entre otras. Ejemplo de regla: `Alert tcp any -> $HOME_NET 80 (msg: "Coincidencia GET"; content:"47 45 54"; sid: 10001; rev:1; clasification: icmp_event;)`

Suricata para su modo IDPS permite las siguientes acciones:

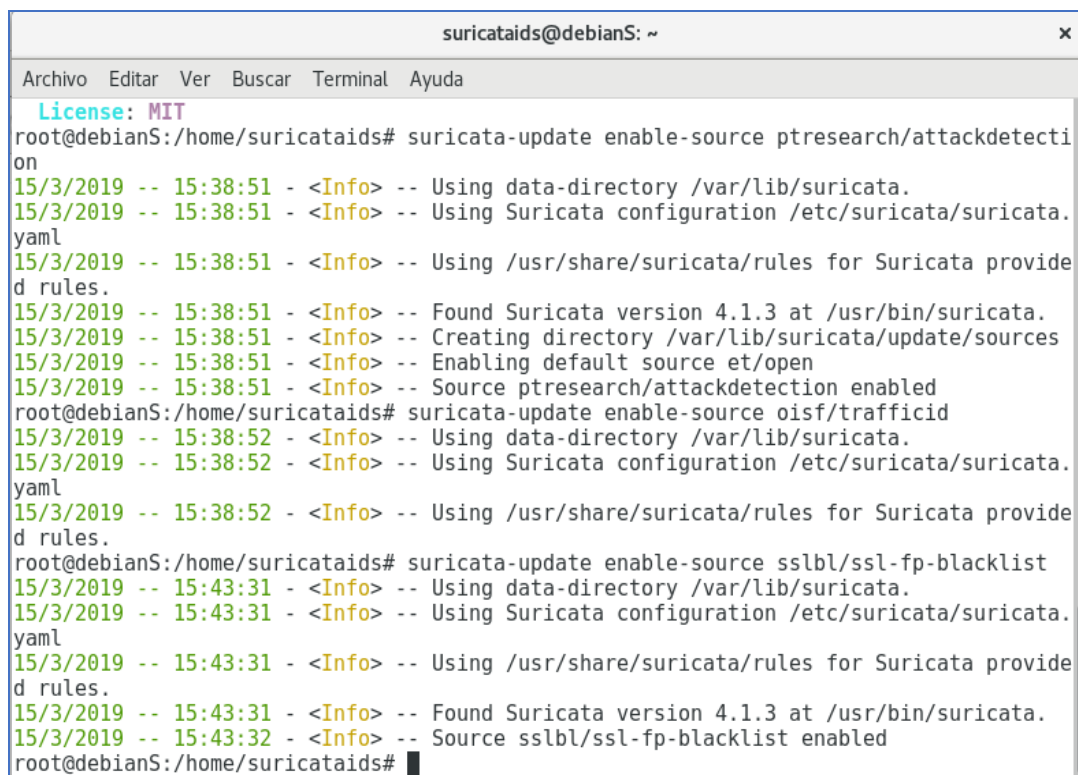
- **Ass:** Ignora el paquete
- **Reject:** bloquea el paquete, almacena el registro y además envía una respuesta de reseteo para las comunicaciones TCP o puerto rechazado para las comunicaciones UDP.
- **Drop:** bloquea de paquete y lo registra.
- **Alert:** Almacena registro de los paquetes.

Activación de nuevas reglas a través de Suricata-update

Para la activación de las reglas se ingresa en consola:

```
Suricata-update enable-source ptresearch/attackdetection
suricata-update enable-source oisf/trafficid
suricata-update enable-source sslbl/ssl-fp-blacklist
```

Dando como resultado la descarga e instalación de las reglas (ver figura 10).



```
suricataids@debianS: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
License: MIT
root@debianS:/home/suricataids# suricata-update enable-source ptresearch/attackdetection
15/3/2019 -- 15:38:51 - <Info> -- Using data-directory /var/lib/suricata.
15/3/2019 -- 15:38:51 - <Info> -- Using Suricata configuration /etc/suricata/suricata.
yaml
15/3/2019 -- 15:38:51 - <Info> -- Using /usr/share/suricata/rules for Suricata provided rules.
15/3/2019 -- 15:38:51 - <Info> -- Found Suricata version 4.1.3 at /usr/bin/suricata.
15/3/2019 -- 15:38:51 - <Info> -- Creating directory /var/lib/suricata/update/sources
15/3/2019 -- 15:38:51 - <Info> -- Enabling default source et/open
15/3/2019 -- 15:38:51 - <Info> -- Source ptresearch/attackdetection enabled
root@debianS:/home/suricataids# suricata-update enable-source oisf/trafficid
15/3/2019 -- 15:38:52 - <Info> -- Using data-directory /var/lib/suricata.
15/3/2019 -- 15:38:52 - <Info> -- Using Suricata configuration /etc/suricata/suricata.
yaml
15/3/2019 -- 15:38:52 - <Info> -- Using /usr/share/suricata/rules for Suricata provided rules.
root@debianS:/home/suricataids# suricata-update enable-source sslbl/ssl-fp-blacklist
15/3/2019 -- 15:43:31 - <Info> -- Using data-directory /var/lib/suricata.
15/3/2019 -- 15:43:31 - <Info> -- Using Suricata configuration /etc/suricata/suricata.
yaml
15/3/2019 -- 15:43:31 - <Info> -- Using /usr/share/suricata/rules for Suricata provided rules.
15/3/2019 -- 15:43:31 - <Info> -- Found Suricata version 4.1.3 at /usr/bin/suricata.
15/3/2019 -- 15:43:32 - <Info> -- Source sslbl/ssl-fp-blacklist enabled
root@debianS:/home/suricataids#
```

Figura 10. Activación reglas Suricata-update

Colocamos en consola `suricata-update` con esto se habilitan las reglas ver figura 11.

```

suricataids@debian$ -
Archivo Editar Ver Buscar Terminal Ayuda
root@debian$:/home/suricataids# suricata-update
15/3/2019 -- 15:45:22 - <Info> -- Using data-directory /var/lib/suricata.
15/3/2019 -- 15:45:22 - <Info> -- Using Suricata configuration /etc/suricata/suricata.
yaml
15/3/2019 -- 15:45:22 - <Info> -- Using /usr/share/suricata/rules for Suricata provide
1 rules.
15/3/2019 -- 15:45:22 - <Info> -- Found Suricata version 4.1.3 at /usr/bin/suricata.
15/3/2019 -- 15:45:22 - <Info> -- Loading /etc/suricata/suricata.yaml
15/3/2019 -- 15:45:23 - <Info> -- Disabling rules with proto modbus
15/3/2019 -- 15:45:23 - <Info> -- Disabling rules with proto enip
15/3/2019 -- 15:45:23 - <Info> -- Disabling rules with proto dnp3
15/3/2019 -- 15:45:23 - <Info> -- Fetching https://raw.githubusercontent.com/jasonish/
suricata-trafficid/master/rules/traffic-id.rules.
100% - 9855/9855
15/3/2019 -- 15:45:23 - <Info> -- Done.
15/3/2019 -- 15:45:24 - <Info> -- Checking https://rules.emergingthreats.net/open/suri
cata-4.1.3/emerging.rules.tar.gz.md5.
15/3/2019 -- 15:45:26 - <Info> -- Fetching https://rules.emergingthreats.net/open/suri
cata-4.1.3/emerging.rules.tar.gz.
100% - 2334876/2334876
15/3/2019 -- 15:45:29 - <Info> -- Done.
15/3/2019 -- 15:45:29 - <Info> -- Fetching https://sslbl.abuse.ch/blacklist/sslblackli
st.rules.
100% - 25556/25556
15/3/2019 -- 15:45:31 - <Info> -- Done.
15/3/2019 -- 15:45:31 - <Info> -- Loading distribution rule file /usr/share/suricata/r
ules/app-layer-events.rules
15/3/2019 -- 15:45:31 - <Info> -- Loading distribution rule file /usr/share/suricata/r
ules/decoder-events.rules
15/3/2019 -- 15:45:31 - <Info> -- Loading distribution rule file /usr/share/suricata/r
ules/dnp3-events.rules
15/3/2019 -- 15:45:31 - <Info> -- Loading distribution rule file /usr/share/suricata/r
ules/dns-events.rules
15/3/2019 -- 15:45:31 - <Info> -- Loading distribution rule file /usr/share/suricata/r
ules/files.rules
15/3/2019 -- 15:45:31 - <Info> -- Loading distribution rule file /usr/share/suricata/r
ules/http-events.rules
15/3/2019 -- 15:45:31 - <Info> -- Loading distribution rule file /usr/share/suricata/r
ules/ipsec-events.rules
15/3/2019 -- 15:45:31 - <Info> -- Loading distribution rule file /usr/share/suricata/r
ules/kerberos-events.rules
15/3/2019 -- 15:45:31 - <Info> -- Loading distribution rule file /usr/share/suricata/r
ules/modbus-events.rules
15/3/2019 -- 15:45:32 - <Info> -- Loading distribution rule file /usr/share/suricata/r
ules/nfs-events.rules
15/3/2019 -- 15:45:32 - <Info> -- Loading distribution rule file /usr/share/suricata/r
ules/ntp-events.rules
15/3/2019 -- 15:45:32 - <Info> -- Loading distribution rule file /usr/share/suricata/r
ules/smb-events.rules
15/3/2019 -- 15:45:32 - <Info> -- Loading distribution rule file /usr/share/suricata/r
ules/smtp-events.rules
15/3/2019 -- 15:45:32 - <Info> -- Loading distribution rule file /usr/share/suricata/r
ules/stream-events.rules
15/3/2019 -- 15:45:32 - <Info> -- Loading distribution rule file /usr/share/suricata/r
ules/tls-events.rules
15/3/2019 -- 15:45:51 - <Info> -- Loaded 30270 rules.
15/3/2019 -- 15:45:56 - <Info> -- Disabled 14 rules.
15/3/2019 -- 15:45:56 - <Info> -- Enabled 0 rules.
15/3/2019 -- 15:45:56 - <Info> -- Modified 0 rules.
15/3/2019 -- 15:45:56 - <Info> -- Dropped 0 rules.
15/3/2019 -- 15:45:59 - <Info> -- Enabled 196 rules for flowbit dependencies.
15/3/2019 -- 15:45:59 - <Info> -- Backing up current rules.
15/3/2019 -- 15:46:24 - <Info> -- Writing rules to /var/lib/suricata/rules/suricata.ru
les: total: 30270; enabled: 22854; added: 3065; removed 2; modified: 1213
15/3/2019 -- 15:46:29 - <Info> -- Testing with suricata -T.
15/3/2019 -- 15:47:11 - <Info> -- Done.
root@debian$:/home/suricataids#

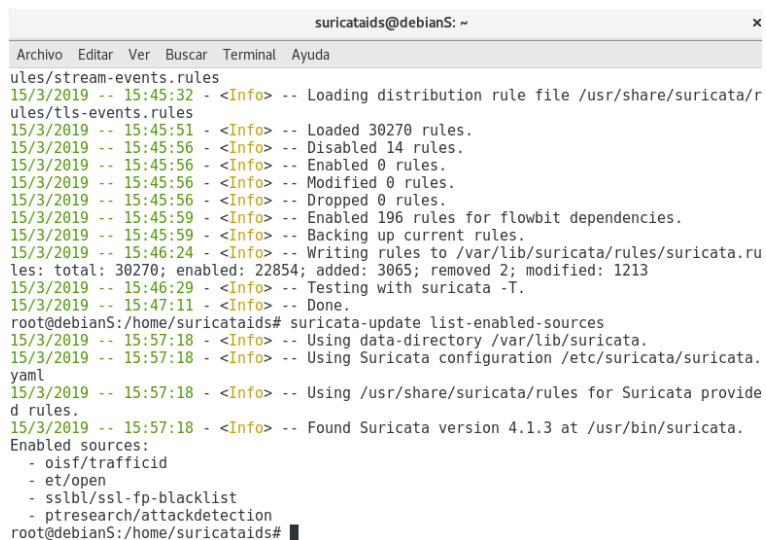
```

Figura 11. Reglas habilitadas

Para ver qué fuentes están habilitadas colocamos

```
Suricata-update list-enabled-sources
```

Nos muestra en la figura 12 el listado de las reglas que están a disposición del sistema IDS:



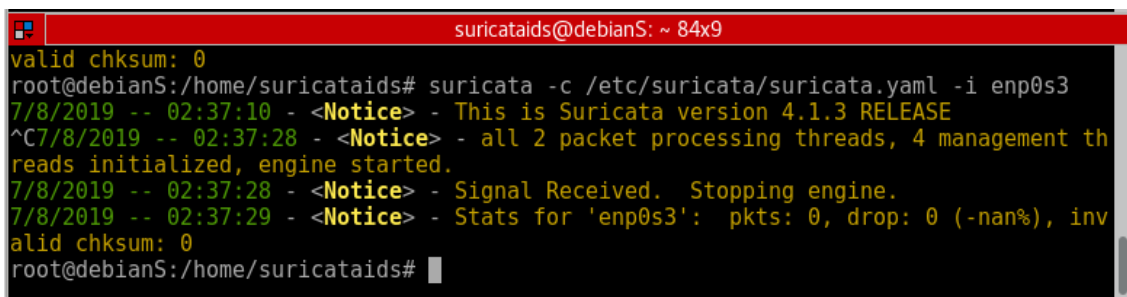
```
suricataids@debianS: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
ules/stream-events.rules
15/3/2019 -- 15:45:32 - <Info> -- Loading distribution rule file /usr/share/suricata/r
ules/tls-events.rules
15/3/2019 -- 15:45:51 - <Info> -- Loaded 30270 rules.
15/3/2019 -- 15:45:56 - <Info> -- Disabled 14 rules.
15/3/2019 -- 15:45:56 - <Info> -- Enabled 0 rules.
15/3/2019 -- 15:45:56 - <Info> -- Modified 0 rules.
15/3/2019 -- 15:45:56 - <Info> -- Dropped 0 rules.
15/3/2019 -- 15:45:59 - <Info> -- Enabled 196 rules for flowbit dependencies.
15/3/2019 -- 15:45:59 - <Info> -- Backing up current rules.
15/3/2019 -- 15:46:24 - <Info> -- Writing rules to /var/lib/suricata/rules/suricata.ru
les: total: 30270; enabled: 22854; added: 3065; removed 2; modified: 1213
15/3/2019 -- 15:46:29 - <Info> -- Testing with suricata -T.
15/3/2019 -- 15:47:11 - <Info> -- Done.
root@debianS:/home/suricataids# suricata-update list-enabled-sources
15/3/2019 -- 15:57:18 - <Info> -- Using data-directory /var/lib/suricata.
15/3/2019 -- 15:57:18 - <Info> -- Using Suricata configuration /etc/suricata/suricata.
yaml
15/3/2019 -- 15:57:18 - <Info> -- Using /usr/share/suricata/rules for Suricata provide
d rules.
15/3/2019 -- 15:57:18 - <Info> -- Found Suricata version 4.1.3 at /usr/bin/suricata.
Enabled sources:
- oisf/trafficid
- et/open
- sslbl/ssl-fp-blacklist
- ptresearch/attackdetection
root@debianS:/home/suricataids#
```

Figura 12 Listado de reglas habilitadas

Para poner en marcha Suricata coloca en consola:

```
Suricata -c /etc/suricata/suricata.yaml -I enp0s3
```

Tendrá que dar como resultado ver figura 13, si no ha mostrado algún error el IDS corre sin ningún problema

A terminal window with a red title bar containing the text 'suricataids@debianS: ~ 84x9'. The terminal output shows the execution of Suricata. It starts with a valid checksum of 0. The user runs 'suricata -c /etc/suricata/suricata.yaml -i enp0s3'. The logs show the version (4.1.3 RELEASE), initialization of packet processing threads, and the engine starting. A signal is received to stop the engine, and the final stats for 'enp0s3' are shown: 0 packets, 0 drops, and 0 invalid checksums. The terminal ends with the prompt 'root@debianS:/home/suricataids#'.

```
valid chksum: 0
root@debianS:/home/suricataids# suricata -c /etc/suricata/suricata.yaml -i enp0s3
7/8/2019 -- 02:37:10 - <Notice> - This is Suricata version 4.1.3 RELEASE
^C7/8/2019 -- 02:37:28 - <Notice> - all 2 packet processing threads, 4 management th
reads initialized, engine started.
7/8/2019 -- 02:37:28 - <Notice> - Signal Received. Stopping engine.
7/8/2019 -- 02:37:29 - <Notice> - Stats for 'enp0s3': pkts: 0, drop: 0 (-nan%), inv
alid chksum: 0
root@debianS:/home/suricataids#
```

Figura 13. IDS Ejecutándose

Paso 3 Selección del tipo de salidas LOG que Suricata genera, bastará con habilitar los diferentes archivos log con los que se desea poner en marcha el sistema en el archivo de configuración ver figura 14.

```

## Step 2: select outputs to enable
##

# The default logging directory. Any log or output file will be
# placed here if its not specified with a full path name. This can be
# overridden with the -l command line parameter.
default-log-dir: /var/log/suricata/

# global stats configuration
stats:
  enabled: yes
  # The interval field (in seconds) controls at what interval
  # the loggers are invoked.
  interval: 8
  # Add decode events as stats.
  #decoder-events: true
  # Decoder event prefix in stats. Has been 'decoder' before, but that leads
  # to missing events in the eve.stats records. See issue #2225.
  decoder-events-prefix: "decoder.event"
  # Add stream events as stats.
  #stream-events: false

# Configure the type of alert (and other) logging you would like.
outputs:
  # a line based alerts log similar to Snort's fast.log
  - fast:
      enabled: yes
      filename: fast.log
      append: yes
      #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'

  # Extensible Event Format (nicknamed EVE) event log in JSON format
  - eve-log:
      enabled: yes
      filetype: regular #regular|syslog|unix_dgram|unix_stream|redis
      filename: eve.json
      #prefix: "@cee: " # prefix to prepend to each log entry
      # the following are valid when type: syslog above
      #identity: "suricata"
      #facility: local5
      #level: Info ## possible levels: Emergency, Alert, Critical,
      # Error, Warning, Notice, Info, Debug

      #redis:
      # server: 127.0.0.1
      # port: 6379
      # async: true ## if redis replies are read asynchronously
      # mode: list ## possible values: list|push (default), rpush, channel|publish
      # ## lpush and rpush are using a Redis list. "list" is an alias for

  # a line based log of HTTP requests (no alerts)
  - http-log:
      enabled: yes
      filename: http.log
      append: yes
      #extended: yes # enable this for extended logging information
      #custom: yes # enabled the custom logging format (defined by customformat)
      #customformat: "%D-%H:%M:%S}t.%z %X-Forwarded-For}i %H %m %h %u %s %B %a:%p ->
      %A:%P"
      #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'

  # a line based log of TLS handshake parameters (no alerts)
  - tls-log:
      enabled: no # Log TLS connections.
      filename: tls.log # File to store TLS logs.
      append: yes
      #extended: yes # Log extended information like fingerprint
      #custom: yes # enabled the custom logging format (defined by customformat)
      #customformat: "%D-%H:%M:%S}t.%z %a:%p -> %A:%P %v %n %d %D"
      #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'
      # output TLS transaction where the session is resumed using a
      # session id
      #session-resumption: no

  # output module to store certificates chain to disk
  - tls-store:
      enabled: no

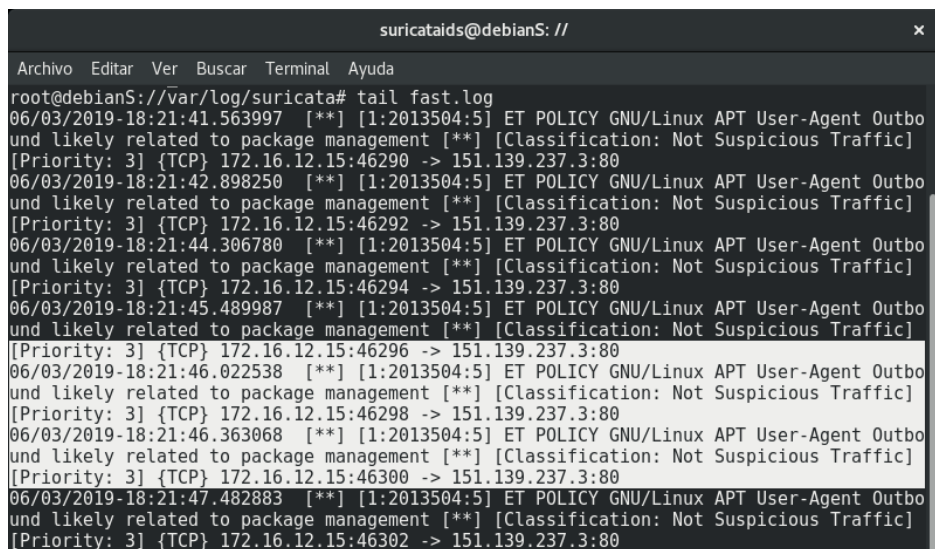
```

Figura 14. Tipos de salidas del IDS

A continuación, se presentan algunos de los archivos con los que se pone en marcha al IDS.

Registro de alertas basadas en línea (fast.log)

Este registro contiene alertas que consisten en una sola línea. Ejemplo de la aparición de una sola línea fast.log-file

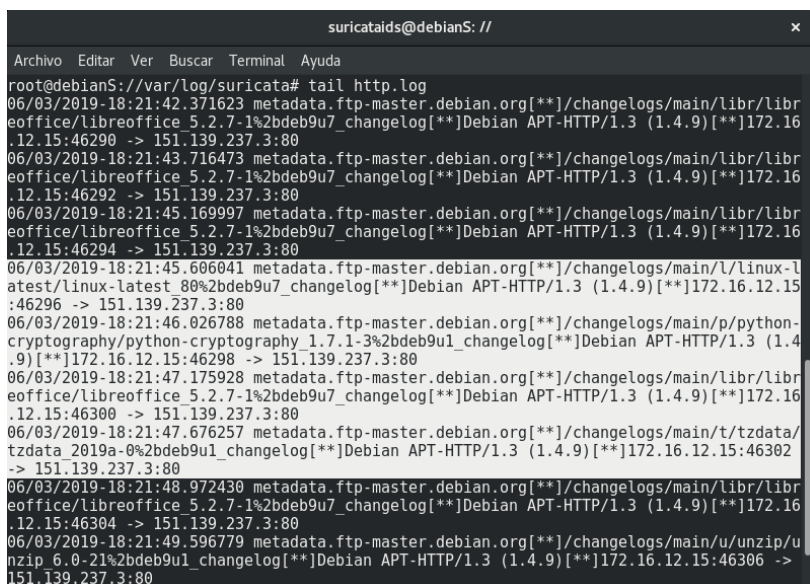


```
suricataids@debianS: //
Archivo Editar Ver Buscar Terminal Ayuda
root@debianS: //var/log/suricata# tail fast.log
06/03/2019-18:21:41.563997 [**] [1:2013504:5] ET POLICY GNU/Linux APT User-Agent Outbo
und likely related to package management [**] [Classification: Not Suspicious Traffic]
[Priority: 3] {TCP} 172.16.12.15:46290 -> 151.139.237.3:80
06/03/2019-18:21:42.898250 [**] [1:2013504:5] ET POLICY GNU/Linux APT User-Agent Outbo
und likely related to package management [**] [Classification: Not Suspicious Traffic]
[Priority: 3] {TCP} 172.16.12.15:46292 -> 151.139.237.3:80
06/03/2019-18:21:44.306780 [**] [1:2013504:5] ET POLICY GNU/Linux APT User-Agent Outbo
und likely related to package management [**] [Classification: Not Suspicious Traffic]
[Priority: 3] {TCP} 172.16.12.15:46294 -> 151.139.237.3:80
06/03/2019-18:21:45.489987 [**] [1:2013504:5] ET POLICY GNU/Linux APT User-Agent Outbo
und likely related to package management [**] [Classification: Not Suspicious Traffic]
[Priority: 3] {TCP} 172.16.12.15:46296 -> 151.139.237.3:80
06/03/2019-18:21:46.022538 [**] [1:2013504:5] ET POLICY GNU/Linux APT User-Agent Outbo
und likely related to package management [**] [Classification: Not Suspicious Traffic]
[Priority: 3] {TCP} 172.16.12.15:46298 -> 151.139.237.3:80
06/03/2019-18:21:46.363068 [**] [1:2013504:5] ET POLICY GNU/Linux APT User-Agent Outbo
und likely related to package management [**] [Classification: Not Suspicious Traffic]
[Priority: 3] {TCP} 172.16.12.15:46300 -> 151.139.237.3:80
06/03/2019-18:21:47.482883 [**] [1:2013504:5] ET POLICY GNU/Linux APT User-Agent Outbo
und likely related to package management [**] [Classification: Not Suspicious Traffic]
[Priority: 3] {TCP} 172.16.12.15:46302 -> 151.139.237.3:80
```

Figura 15. Información archivo fast.log

Registro basado en línea de solicitudes HTTP (http.log)

Este registro realiza un seguimiento de todos los eventos de tráfico HTTP. Contiene la solicitud HTTP, el nombre de host, el URI y el User-Agent. Esta información se almacenará en http.log (nombre predeterminado, en el directorio de registro de suricata).



```
suricataids@debianS: //
Archivo Editar Ver Buscar Terminal Ayuda
root@debianS: //var/log/suricata# tail http.log
06/03/2019-18:21:42.371623 metadata.ftp-master.debian.org[**]/changelogs/main/libr/libr
eoffice/libreoffice 5.2.7-1%2bdeb9u7_changelog[**]Debian APT-HTTP/1.3 (1.4.9)[**]172.16
.12.15:46290 -> 151.139.237.3:80
06/03/2019-18:21:43.716473 metadata.ftp-master.debian.org[**]/changelogs/main/libr/libr
eoffice/libreoffice 5.2.7-1%2bdeb9u7_changelog[**]Debian APT-HTTP/1.3 (1.4.9)[**]172.16
.12.15:46292 -> 151.139.237.3:80
06/03/2019-18:21:45.169997 metadata.ftp-master.debian.org[**]/changelogs/main/libr/libr
eoffice/libreoffice 5.2.7-1%2bdeb9u7_changelog[**]Debian APT-HTTP/1.3 (1.4.9)[**]172.16
.12.15:46294 -> 151.139.237.3:80
06/03/2019-18:21:45.606041 metadata.ftp-master.debian.org[**]/changelogs/main/l/linux-l
atest/linux-latest 80%2bdeb9u7_changelog[**]Debian APT-HTTP/1.3 (1.4.9)[**]172.16.12.15
:46296 -> 151.139.237.3:80
06/03/2019-18:21:46.026788 metadata.ftp-master.debian.org[**]/changelogs/main/p/python-
cryptography/python-cryptography 1.7.1-3%2bdeb9u1_changelog[**]Debian APT-HTTP/1.3 (1.4
.9)[**]172.16.12.15:46298 -> 151.139.237.3:80
06/03/2019-18:21:47.175928 metadata.ftp-master.debian.org[**]/changelogs/main/libr/libr
eoffice/libreoffice 5.2.7-1%2bdeb9u7_changelog[**]Debian APT-HTTP/1.3 (1.4.9)[**]172.16
.12.15:46300 -> 151.139.237.3:80
06/03/2019-18:21:47.676257 metadata.ftp-master.debian.org[**]/changelogs/main/t/tzdata/
tzdata 2019a-0%2bdeb9u1_changelog[**]Debian APT-HTTP/1.3 (1.4.9)[**]172.16.12.15:46302
-> 151.139.237.3:80
06/03/2019-18:21:48.972430 metadata.ftp-master.debian.org[**]/changelogs/main/libr/libr
eoffice/libreoffice 5.2.7-1%2bdeb9u7_changelog[**]Debian APT-HTTP/1.3 (1.4.9)[**]172.16
.12.15:46304 -> 151.139.237.3:80
06/03/2019-18:21:49.596779 metadata.ftp-master.debian.org[**]/changelogs/main/u/unzip/u
nzip 6.0-21%2bdeb9u1_changelog[**]Debian APT-HTTP/1.3 (1.4.9)[**]172.16.12.15:46306 ->
151.139.237.3:80
```

Figura 16. Información archivo http.log.

Stats.log

Registra estadísticas del motor, como los contadores de paquetes, los contadores de uso de memoria entre otros.

```
suricataids@debianS: //
Archivo Editar Ver Buscar Terminal Ayuda
root@debianS://var/log/suricata# tail stats.log
app_layer.flow.failed_udp | Total | 4
flow_mgr.closed_pruned | Total | 39
flow_mgr.new_pruned | Total | 15
flow_mgr.est_pruned | Total | 84
flow.spare | Total | 10000
flow_mgr.rows_checked | Total | 65536
flow_mgr.rows_skipped | Total | 65536
tcp.memuse | Total | 573440
tcp.reassembly_memuse | Total | 98304
flow.memuse | Total | 7234912
root@debianS://var/log/suricata#
```

Figura 17. Información archivo stats.log.

Formato de evento extensible (Eve.json)

Estos archivos son salidas JSON para alertas y eventos, la información que este proporciona son datos sobre la hora fecha, tipo de evento ip de origen e ip de destino, puertos y protocolos. Permite una fácil integración con herramientas de terceros como logstash en el caso de integrar a Suricata con una interfaz gráfica.

```
suricataids@debianS: //
Archivo Editar Ver Buscar Terminal Ayuda
root@debianS://var/log/suricata# tail eve.json
{"timestamp":"2019-06-03T18:35:53.016168+0200","event_type":"stats","stats":{"uptime":929,"capture":{"kernel_packets":22892,"kernel_drops":0,"errors":0},"decoder":{"pkts":22879,"bytes":20639598,"invalid":0,"ipv4":22584,"ipv6":255,"ethernet":22879,"raw":0,"null":0,"sll":0,"tcp":22568,"udp":253,"sctp":0,"icmpv4":0,"icmpv6":18,"ppp":0,"pppoe":0,"gre":0,"vlan":0,"vlan_qinq":0,"ieee8021ah":0,"teredo":0,"ipv4_in_ipv6":0,"ipv6_in_ipv6":0,"mpls":0,"avg_pkt_size":902,"max_pkt_size":1514,"erspan":0,"event":{"ipv4":{"pkt_too_small":0,"hlen_too_small":0,"iplen_smaller_than_hlen":0,"trunc_pkt":0,"opt_invalid":0,"opt_invalid_len":0,"opt_malformed":0,"opt_pad_required":0,"opt_eol_required":0,"opt_duplicate":0,"opt_unknown":0,"wrong_ip_version":0,"icmpv6":0,"frag_pkt_too_large":0,"frag_overlap":0,"frag_ignored":0},"icmpv4":{"pkt_too_small":0,"unknown_type":0,"unknown_code":0,"ipv4_trunc_pkt":0,"ipv4_unknown_ver":0},"icmpv6":{"unknown_type":0,"unknown_code":0,"pkt_too_small":0,"ipv6_unknown_version":0,"ipv6_trunc_pkt":0,"mld_message_with_invalid_hl":0,"unassigned_type":0,"experimentation_type":0},"ipv6":{"pkt_too_small":0,"trunc_pkt":0,"trunc_exthdr":0,"exthdr_dupl_fh":0,"exthdr_useless_fh":0,"exthdr_dupl_rh":0,"exthdr_dupl_hh":0,"exthdr_dupl_dh":0,"exthdr_dupl_ah":0,"exthdr_dupl_eh":0,"exthdr_invalid_optlen":0,"wrong_ip_version":0,"exthdr_ah_res_not_null":0,"hopopts_unknown_opt":0,"hopopts_only_padding":0,"dstopts_unknown_opt":0,"dstopts_only_padding":0,"rh_type_0":0,"zero_len_padn":0,"fh_non_zero_reserved_field":0,"data_after_none_header":0,"unknown_next_header":0,"icmpv4":0,"frag_pkt_too_large":0,"frag_overlap":0,"frag_ignored":0,"ipv4_in_ipv6_too_small":0,"ipv4_in_ipv6_wrong_version":0,"ipv6_in_ipv6_too_small":0,"ipv6_in_ipv6_wrong_version":0},"tcp":{"pkt_too_small":0,"hlen_too_small":0,"invalid_optlen":0,"opt_invalid_len":0,"opt_duplicate":0},"udp":{"pkt_too_small":0,"hlen_too_small":0,"hlen_invalid":0},"sll":{"pkt_too_small":0},"ethernet":{"pkt_too_small":0},"ppp":{"pkt_too_small":0},"vju_pkt_too_small":0,"ip4_pkt_too_small":0,"ip6_pkt_too_small":0,"wrong_type":0,"unsup_proto":0},"pppoe":{"pkt_too_small":0,"wrong_code":0,"malformed_tags":0},"gre":{"pkt_too_small":0,"wrong_version":0,"version0_recur":0,"version0_flags":0,"version0_hdr_too_big":0,"version0_malformed_sre_hdr":0,"version1_chksum":0,"version1_route":0,"versi
```

Figura 18. Información archivo eve.json

Selección de los tipos de los ataques más comunes en redes wifi para poner a prueba la herramienta IDS.

En esta sección se especifica en Tabla VII los ataques más comunes descubiertos por un sistema de detección de intrusos en redes cableadas e inalámbricas, los ataques se deducen gracias al análisis realizado en la revisión literaria con el nombre **“Revisión Sistemática de literatura de herramientas open source para la detección de intrusos”** correspondiente al anexo (ver Anexo 1),

TABLA VII. ATAQUES COMUNES DETECTADOS POR UN IDS.

Tipo de ataque	Modo de operación	Red en la que se ejecutó
Exploración de red	El atacante realiza un escaneo de toda la red obteniendo así información sobre puertos abiertos, las máquinas conectadas a la red e incluso información del sistema operativo.	Cableada, Wifi
Man-in-the-middle	El atacante tiene conexiones independientes con las víctimas y transmite mensajes entre ellos, haciéndoles creer que están hablando directamente entre sí a través de una conexión privada, cuando en realidad toda la conversación es controlada por el atacante.	Cableada, WIMAX
Dos Y ddos	<p>Tiene tres modos de operación</p> <p>Volumétricos por Inundación (TCP SYN, UDP, e inundaciones HTTP). Que tienen como objetivo principal, el consumo de ancho de banda, solo causan obstrucciones.</p> <p>Ataques Reflexivos (NTP, DNS, SSDP/upnp, Chargen, SNMP); Este ataque suelen generar de 2,56 a 10.5 veces más tráfico de respuesta que el propio enviado por el atacante, dependerá en buena medida del servicio atacado, dándose casos de llegar a factores de amplificación superiores a 50 en ataques multi vectores. Estos tipos de ataques son extremadamente potentes.</p> <p>Agotamiento de recursos: genera tráfico fragmentado y mal formado, bajo y solicitudes lentas, lo que hace que los objetivos se ralenticen intentando resolver las peticiones erróneas.</p>	Cableada, WIMAX, Mobile ad-hoc, Networks (manets), 802.11, SDN

Para poner a prueba la herramienta Suricata se optó por los ataques exploración de red, dos y Man-in-the-middle los cuales se lo efectúa a través de la máquina atacante Kali Linux.

Evaluación de la herramienta.

Para la evaluación del IDS se utilizó el método experimental, para esto se creó un escenario virtual sobre el cual se instaló Suricata y se realizó la implementación de los ataques más comunes a un IDS los mismos que se establecieron en la revisión sistemática de literatura. El IDS se puso en marcha haciendo uso de la técnica de detección por firmas, y se creó tres escenarios de ataques.

En la evaluación se realizó mediante los siguientes pasos: puesta en marcha del IDS, ejecución de ataques y análisis.

- **Puesta en marcha del IDS**

Al poner en marcha el IDS se considera los siguientes aspectos:

Facilidad de instalación: Suricata puede ser instalado como un NIDS o un HIDS esto dependerá del uso que se desee darle, cabe destacar que este IDS cuenta con una comunidad muy activa, así como su guía de instalación en línea la cual especifica claramente cómo ser instalado.

Coexistencia: este IDS puede operar sin causar algún problema, en el archivo `suricata.yaml` se especifica claramente que acción estará por realizar, si es necesario que este haga uso de otra aplicación ya sea para la presentación de datos o recolección de información este es compatible con algunos programas que se desempeñen en el área que Suricata lo hace.

Utilización de recursos: el IDS hace uso del número de CPU disponibles del sistema operativo en el que este se ha instalado, en la figura 19 se asigna una CPU al sistema operativo Debian, al poner en marcha Suricata este se inicia con un hilo de procesamiento de paquetes y 4 subprocesos para administración de hilos.

Como se observa en la figura 20 al asignarle 2 CPU al sistema IDS Suricata tomará este recurso disponible para su ejecución.

```

suricataids@debianS: ~
Archivo Editar Ver Buscar Terminal Ayuda
suricataids@debianS:~$ su
Contraseña:
root@debianS:/home/suricataids# suricata -c /etc/suricata/suricata.yaml -i enp0s3
3/6/2019 -- 18:20:20 - <Notice> - This is Suricata version 4.1.3 RELEASE
3/6/2019 -- 18:20:56 - <Notice> - all 1 packet processing threads, 4 management
threads initialized, engine started.

```

Figura 19. IDS puesta en marcha una CPU

```

suricataids@debianS: ~
Archivo Editar Ver Buscar Terminal Ayuda
suricataids@debianS:~$ su
Contraseña:
root@debianS:/home/suricataids# suricata -c /etc/suricata/suricata.yaml -i enp0s3
3 --simulate-ips
5/6/2019 -- 09:30:00 - <Info> - Setting IPS mode
5/6/2019 -- 09:30:00 - <Notice> - This is Suricata version 4.1.3 RELEASE
5/6/2019 -- 09:30:22 - <Notice> - all 2 packet processing threads, 4 management
threads initialized, engine started.
^C5/6/2019 -- 09:39:09 - <Notice> - Signal Received. Stopping engine.
5/6/2019 -- 09:39:10 - <Notice> - Stats for 'enp0s3': pkts: 54969, drop: 0 (0.0
0%), invalid chksum: 0
root@debianS:/home/suricataids#

```

Figura 20. IDS puesta en marcha dos CPU

Para ver el porcentaje de memoria y CPU que el IDS (ver figura 21) consume se colocó en consola el comando top que es el encargado de dar información acerca del uso de estos recursos y los procesos en ejecución en tiempo real.

```

suricataids@debianS: ~
Archivo Editar Ver Buscar Terminal Ayuda
top - 09:37:52 up 11 min, 1 user, load average: 0,11, 0,32, 0,29
Tasks: 150 total, 1 running, 149 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1,9 us, 0,3 sy, 0,0 ni, 97,5 id, 0,3 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 2052412 total, 616792 free, 995948 used, 439672 buff/cache
KiB Swap: 2095100 total, 2095100 free, 0 used. 906036 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
 1271 root        20   0   895572   380200 12000 S   2,0   18,5   0:48.99 Suricata-M+
   825 suricat+   20   0   2288724 232608  84700 S   1,3   11,3   0:22.31 gnome-shell
   733 suricat+   20   0   349908   47048  29328 S   1,0    2,3   0:02.30 Xorg
  1122 suricat+   20   0   601444   34352  25388 S   1,0    1,7   0:01.14 gnome-term+
  1553 suricat+   20   0   44904    3652   3084 R   0,3    0,2   0:00.36 top
     1 root        20   0   139008    6808   5304 S   0,0    0,3   0:02.02 systemd
     2 root        20   0         0         0       0 S   0,0    0,0   0:00.00 kthreadd
     3 root        20   0         0         0       0 S   0,0    0,0   0:00.38 ksoftirqd/0
     5 root        0 -20         0         0       0 S   0,0    0,0   0:00.00 kworker/0:0+
     6 root        20   0         0         0       0 S   0,0    0,0   0:00.07 kworker/u4+
     7 root        20   0         0         0       0 S   0,0    0,0   0:00.59 rcu_sched
     8 root        20   0         0         0       0 S   0,0    0,0   0:00.00 rcu_bh
     9 root        rt    0         0         0       0 S   0,0    0,0   0:00.00 migration/0

```

Figura 21. Porcentaje de consumo de CPU y memoria

Bases de datos: al ser Suricata un sistema basado en firmas se utilizan las reglas que viene por defecto en el IDS si no se ha configurado previamente reglas propias, para la actualización de las bases de datos de firmas se puede hacer uso de **Oinkmaster** que es una herramienta para la actualización de estas, pero a partir de la versión 4.0.1 Suricata tiene su propia herramienta (suricata-update) para la actualización de sus base de datos de firmas las cuales pueden ser descargadas y actualizadas en línea, en el caso de ser utilizado en entornos en donde la información a procesar es grande y se necesita presentarla en un formato amigable se hace uso de bases de datos como Mysql.

Ejecución ataques a la red.

Los ataques que se realizaron para poner a prueba al IDS son: Exploración de información con Nmap, denegación de servicio, y mitm.

TABLA VIII. SIMULACIONES DE ATAQUES

Ataque	Herramienta	Origen	Destino
Exploració de información	Nmap	Kali Linux	Red Local
Denegación de servicio	Metasploit	Kali Linux	Puerta de enlace
Mitm	Ettercap	Kali linux	Debian

Escenario 1 Exploración de información con Nmap (Network mapper) de Kali Linux:

Nmap permite explorar, administrar y auditar una red de ordenadores, entre sus utilidades está: encontrar hosts online, escanear puertos y servicios corriendo a través de ellos, obtener información del sistema operativo, firewalls entre otros servicios. Principalmente esta herramienta es utilizada para 3 cosas:

- Auditorías de seguridad
- Pruebas rutinarias de escaneo de redes
- Recolección de información para futuros ataques

Se realiza un escaneo de red completa sigiloso con detección de SO con el fin de constatar que las máquinas virtuales se encuentran en la misma red como si estuvieran conectadas físicamente.

Paso 1. Para el escaneo de la red entramos en consola como root y colocamos el siguiente comando `nmap -ss -O <IP/máscara>`.

Este tipo de escaneo genera los siguientes datos:

- **Tipo de dispositivo (device type):** normalmente aparecerá “general purpose” o propósito general en ambientes domésticos.
- **Sistema operativo:** intentará reconocer el sistema o kernel (en versiones Linux).
- **Distancia de red (network distance):** se lanzará además una traza de red que nos indicará cuantos saltos nos separan del dispositivo/red analizado.

Paso 2. Una vez finalizado el escaneo de Nmap como resultado se muestra, ver figura 22 las diferentes máquinas encontradas, así como información obtenida de cada host conectado a la red.

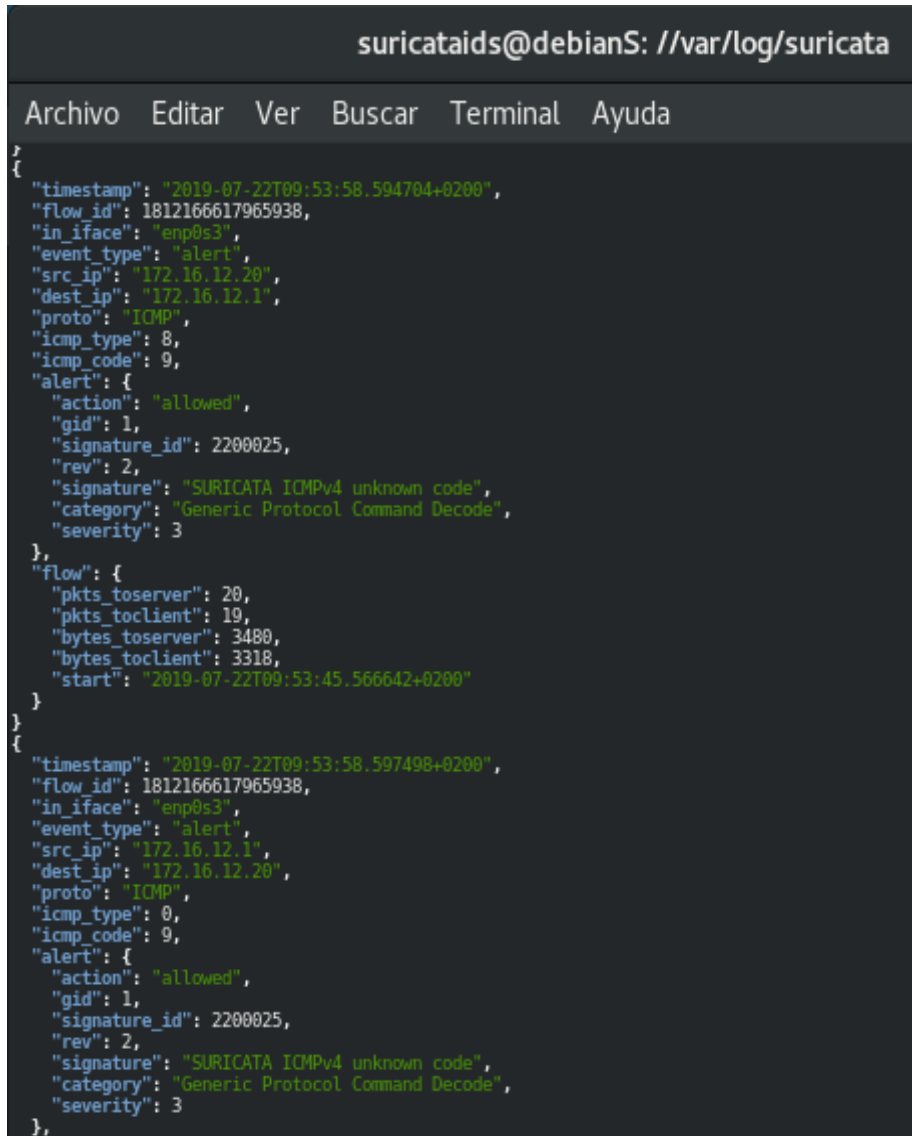
```
root@kali-linux-2017: ~  
Archivo Editor Ver Buscar Terminal Ayuda  
root@kali-linux-2017:~# nmap -sS -O 172.16.12.1/24  
  
Starting Nmap 7.60 ( https://nmap.org ) at 2019-06-03 19:46 CEST  
Nmap scan report for 172.16.12.1  
Host is up (0.0043s latency).  
Not shown: 995 closed ports  
PORT      STATE SERVICE  
53/tcp    open  domain  
80/tcp    open  http  
3333/tcp  open  dec-notes  
5555/tcp  open  freeciv  
49152/tcp open  unknown  
MAC Address: [REDACTED] (Netgear)  
No exact OS matches for host (If you know what OS is running on it, see https://  
nmap.org/submit/ ).  
TCP/IP fingerprint:  
OS:  
OS:  
OS:  
OS:  
OS:  
OS:  
OS:  
OS:  
OS:  
OS:  
  
Network Distance: 1 hop  
  
Nmap scan report for 172.16.12.3  
Host is up (0.0041s latency).  
All 1000 scanned ports on 172.16.12.3 are closed  
MAC Address: [REDACTED] (Samsung Electronics)  
Too many fingerprints match this host to give specific OS details  
Network Distance: 1 hop  
  
Nmap scan report for 172.16.12.7  
Host is up (0.0048s latency).  
Not shown: 996 closed ports  
PORT      STATE SERVICE  
554/tcp   open  rtsp  
5555/tcp  open  freeciv  
7100/tcp  open  font-service  
8000/tcp  open  http-alt  
MAC Address: [REDACTED] (Unknown)  
Device type: phone  
Running: CyanogenMod 12.X, Google Android 5.X  
OS CPE: cpe:/o:cyanogenmod:cyanogenmod:12 cpe:/o:google:android:5.0.2  
OS details: CyanogenMod 12 (Android 5.0.2)  
Network Distance: 1 hop  📶  
  
Nmap scan report for 172.16.12.8  
Host is up (0.00076s latency).  
All 1000 scanned ports on 172.16.12.8 are filtered  
MAC Address: [REDACTED] (Chicony Electronics)  
Too many fingerprints match this host to give specific OS details  
Network Distance: 1 hop  
  
Nmap scan report for 172.16.12.12  
Host is up (0.013s latency).  
All 1000 scanned ports on 172.16.12.12 are closed  
MAC Address: [REDACTED] (Huawei Technologies)  
Too many fingerprints match this host to give specific OS details  
Network Distance: 1 hop  
  
Nmap scan report for 172.16.12.20  
Host is up (0.000084s latency).  
All 1000 scanned ports on 172.16.12.20 are closed  
Too many fingerprints match this host to give specific OS details  
Network Distance: 0 hops  
  
OS detection performed. Please report any incorrect results at https://nmap.org/  
submit/ .  
Nmap done: 256 IP addresses (6 hosts up) scanned in 25367.58 seconds  
root@kali-linux-2017:~#
```

Figura 22. Exploración de red con Nmap

Resultados del IDS frente al ataque de exploración

Una vez realizada la exploración el archivo eve.json muestra la alerta de un ataque y que este se ha dado con éxito ver figura 23 en el archivo se ve el e tipo de evento, ip

del atacante, ip de la víctima en este caso el router, así como el protocolo, si la acción ha sido permitida, la firma de este ataque y la severidad el evento.



```
suricataids@debianS: //var/log/suricata
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
{
  {
    "timestamp": "2019-07-22T09:53:58.594704+0200",
    "flow_id": 1812166617965938,
    "in_iface": "enp0s3",
    "event_type": "alert",
    "src_ip": "172.16.12.20",
    "dest_ip": "172.16.12.1",
    "proto": "ICMP",
    "icmp_type": 8,
    "icmp_code": 9,
    "alert": {
      "action": "allowed",
      "gid": 1,
      "signature_id": 2200025,
      "rev": 2,
      "signature": "SURICATA ICMPv4 unknown code",
      "category": "Generic Protocol Command Decode",
      "severity": 3
    },
    "flow": {
      "pkts_toserver": 20,
      "pkts_toclient": 19,
      "bytes_toserver": 3480,
      "bytes_toclient": 3318,
      "start": "2019-07-22T09:53:45.566642+0200"
    }
  }
  {
    "timestamp": "2019-07-22T09:53:58.597498+0200",
    "flow_id": 1812166617965938,
    "in_iface": "enp0s3",
    "event_type": "alert",
    "src_ip": "172.16.12.1",
    "dest_ip": "172.16.12.20",
    "proto": "ICMP",
    "icmp_type": 0,
    "icmp_code": 9,
    "alert": {
      "action": "allowed",
      "gid": 1,
      "signature_id": 2200025,
      "rev": 2,
      "signature": "SURICATA ICMPv4 unknown code",
      "category": "Generic Protocol Command Decode",
      "severity": 3
    },
    "flow": {
      "pkts_toserver": 20,
      "pkts_toclient": 19,
      "bytes_toserver": 3480,
      "bytes_toclient": 3318,
      "start": "2019-07-22T09:53:45.566642+0200"
    }
  }
}
```

Figura 23. Alerta generada por exploración con Nmap

En la figura 24 se muestra la puesta en marcha y la finalización del IDS durante el ataque, en él se evidencia que todos los paquetes enviados a la red han cruzado incluyendo a aquellos destinados a la extracción de información de la red.

```

suricataids@debianS: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda

root@debianS:/home/suricataids# suricata -c /etc/suricata/suricata.yaml -i enp0s3 --simulate-ips
22/7/2019 -- 09:48:30 - <Info> - Setting IPS mode
22/7/2019 -- 09:48:30 - <Notice> - This is Suricata version 4.1.3 RELEASE
22/7/2019 -- 09:48:51 - <Notice> - all 2 packet processing threads, 4 management threads initialized, engine started.
^C22/7/2019 -- 10:26:57 - <Notice> - Signal Received. Stopping engine.
22/7/2019 -- 10:26:58 - <Notice> - Stats for 'enp0s3': pkts: 12877, drop: 0 (0.00%), invalid checksum: 0
root@debianS:/home/suricataids#

```

Figura 24. Ejecución del IDS

```

suricataids@debianS: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda

1  [|||||] 8.7% Tasks: 105, 322 thr; 1 running
2  [|||||] 9.9% Load average: 1.41 0.55 0.31
Mem[|||||] 1.29G/1.96G Uptime: 02:25:30
Swp[|] 56.1M/2.00G

  PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
  ---
 826 suricatai  20   0 2257M 200M 34668 S  6.6 10.0 4:30.52 /usr/bin/gnome-sh
 733 suricatai  20   0 354M 40140 13364 S  2.0  2.0 1:20.26 /usr/lib/xorg/Xor
1123 suricatai  20   0 635M 32916 20416 S  1.3  1.6 1:05.45 /usr/lib/gnome-te
3853 root       20   0 874M 367M 8072 S  2.7 18.3 0:28.49 suricata -c /etc/
3854 root       20   0 24544 3592 2964 R  4.0  0.2 0:08.83 htop
 857 suricatai -6   0 1906M 7900 4872 S  1.3  0.4 1:14.46 /usr/bin/pulseaud
 829 suricatai  20   0 2257M 200M 34668 S  0.7 10.0 0:42.01 /usr/bin/gnome-sh
 830 suricatai  20   0 2257M 200M 34668 S  0.0 10.0 0:45.14 /usr/bin/gnome-sh
 856 suricatai  9 -11 1906M 7900 4872 S  1.3  0.4 1:19.25 /usr/bin/pulseaud
3867 suricatai  20   0 12292 1312 1120 S  0.7  0.1 0:02.19 jq .
3856 root       20   0 874M 367M 8072 S  0.7 18.3 0:01.67 suricata -c /etc/
1475 suricatai  20   0 1735M 91060 39532 S  0.0  4.4 0:44.13 /usr/lib/firefox-
3858 root       20   0 874M 367M 8072 S  0.0 18.3 0:00.67 suricata -c /etc/
1384 suricatai  20   0 1858M 234M 46640 S  4.0 11.7 3:15.34 /usr/lib/firefox-
3857 root       20   0 874M 367M 8072 S  0.0 18.3 0:02.11 suricata -c /etc/
3855 root       20   0 874M 367M 8072 S  0.7 18.3 0:01.82 suricata -c /etc/
1597 suricatai  20   0 460M 5256 3772 S  0.0  0.3 0:01.59 /usr/lib/speech-d
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice F9Kill F10Quit

```

Figura 25. Consumo de recursos durante el ataque

Durante el ataque de exploración (un lapso de 10 min) el consumo de recursos por parte de IDS fue de un 2.7% CPU y 18.3% de memoria. Ver figura 25.

Escenario 2 Ataque DoS.

Para realizar el ataque de denegación de servicio se hace uso de Metasploit desde el atacante 1 y Slowloris desde el atacante 2

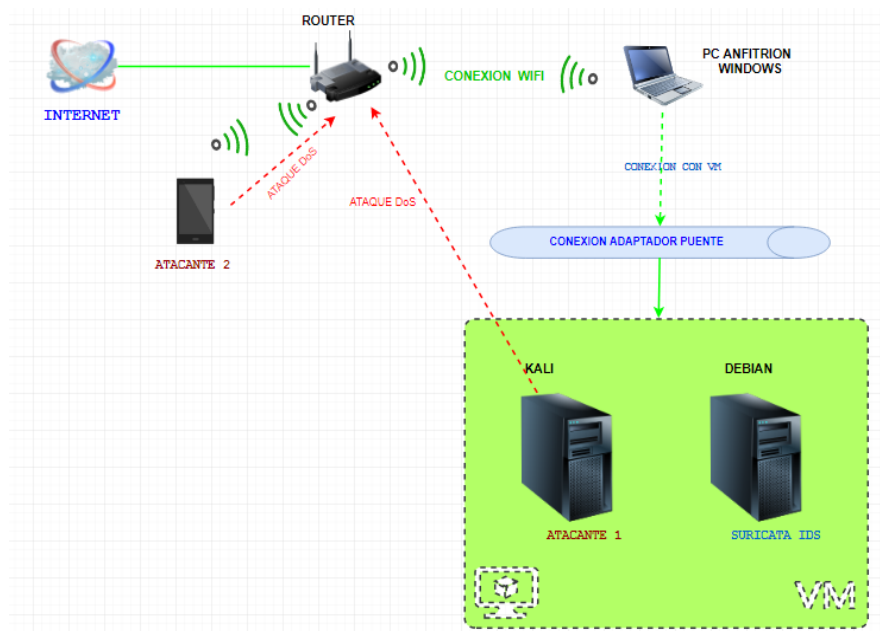


Figura 26. Esquema ataque DoS

Ataque con Metasploit: esta herramienta está diseñada para explotar las vulnerabilidades, el ataque que se realiza es de tipo SYN Flood este consiste en un envío masivo de solicitudes de conexión TCP.

- **Paso 1.** Iniciamos el programa ver Figura 27.

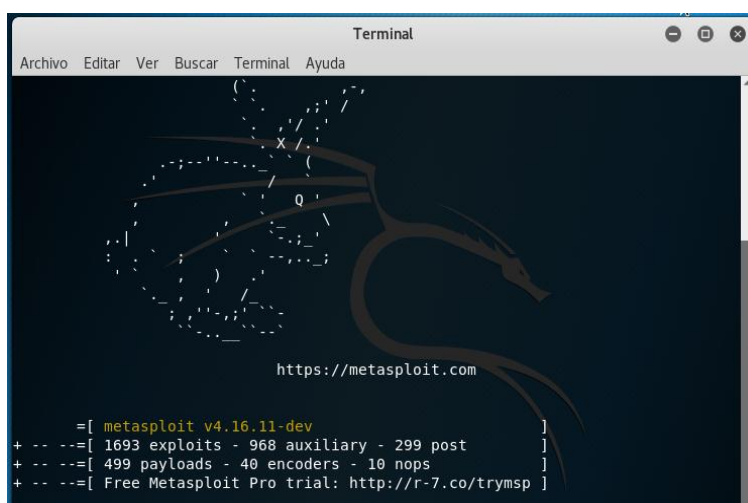


Figura 27. Inicio de Metasploit

- **Paso 2.** Se coloca en consola `use /auxiliary/dos/tcp/synflood` con esto el ataque a realizar será a la capa de transporte.
- **Paso 3.** Se establece el host o ip de la víctima, con el comando `set RHOST (ip victima)` en este caso se realizó el ataque al router WIFI ver figura 28.
- **Paso 4.** Se procede con la explotación del ataque colocando en consola `exploit`.

```

Terminal
Archivo  Editar  Ver  Buscar  Terminal  Ayuda

https://metasploit.com

=[ metasploit v4.16.11-dev ]
+ -- --=[ 1693 exploits - 968 auxiliary - 299 post ]
+ -- --=[ 499 payloads - 40 encoders - 10 nops ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use auxiliary/dos/tcp/synflood
msf auxiliary(synflood) > set RHOST 172.16.12.1
RHOST => 172.16.12.1
msf auxiliary(synflood) > exploit

[*] SYN flooding 172.16.12.1:80...

```

Figura 28. Ataque DoS

Ya estando el ataque en proceso se comprueba si este afectó al tráfico de la red.

- **Paso 5.** Finaliza el ataque colocando “`ctrl+C`”

Resultados del IDS frente al ataque dos

Suricata se puso en marcha utilizando la configuración básica en modo IPS y la configuración personalizada en modo IDS, el ataque se realizó durante 10 minutos tiempo en el cual el tráfico de la red no se vio afectado por el ataque dos, durante este lapso el IDS registro 190252 paquetes en total y los cuales 6515 resultaron paquetes caídos ver figura 28.

```
root@debianS:/home/suricataids# suricata -c /etc/suricata/suricata.yaml -i enp0s3 --simulate-ips
11/7/2019 -- 03:23:40 - <Info> - Setting IPS mode
11/7/2019 -- 03:23:40 - <Notice> - This is Suricata version 4.1.3 RELEASE
11/7/2019 -- 03:24:12 - <Notice> - all 2 packet processing threads, 4 management threads initialized, engine started.
^C11/7/2019 -- 03:33:55 - <Notice> - Signal Received. Stopping engine.
11/7/2019 -- 03:33:58 - <Notice> - Stats for 'enp0s3': pkts: 190252, drop: 6515 (3.42%), invalid chksum: 0
root@debianS:/home/suricataids#
```

Figura 29. Puesta en marcha de Suricata

En la figura 29 el ataque registrado se lo realizó en diferentes elementos de la red (rex externa, puerta de enlace, máquina anfitrión y máquina donde se ejecuta Suricata) las alertas muestran el tipo de evento, puerto (TCP y UDP), el destino y origen del ataque, su estado, el tipo de alerta o inicio y finalización, así como el tipo de alerta en este caso se ha considerado falsa debido a que el ataque no interfiere significativamente con el tráfico de la red.

suricataids@debianS: ~	suricataids@debianS: ~
<pre> Archivo Editar Ver Buscar Terminal Ayuda { "timestamp": "2019-08-06T16:57:24.102801+0200", "flow_id": 561977499126103, "event_type": "flow", "src_ip": "172.16.12.13", "src_port": 43906, "dest_ip": "172.16.12.0", "dest_port": 60747, "proto": "UDP", "app_proto": "failed", "flow": { "pkts_toserver": 1, "pkts_toclient": 0, "bytes_toserver": 60, "bytes_toclient": 0, "start": "2019-08-06T16:56:53.198999+0200", "end": "2019-08-06T16:56:53.198999+0200", "age": 0, "state": "new", "reason": "timeout", "alerted": false } } </pre> <p>Ataque a la red</p>	<pre> Archivo Editar Ver Buscar Terminal Ayuda { "timestamp": "2019-08-06T17:08:43.002942+0200", "flow_id": 1585670113749826, "event_type": "flow", "src_ip": "172.16.12.21", "src_port": 53214, "dest_ip": "172.16.12.1", "dest_port": 53, "proto": "UDP", "app_proto": "dns", "flow": { "pkts_toserver": 2, "pkts_toclient": 0, "bytes_toserver": 184, "bytes_toclient": 0, "start": "2019-08-06T17:08:12.551746+0200", "end": "2019-08-06T17:08:12.551935+0200", "age": 0, "state": "new", "reason": "timeout", "alerted": false } } </pre> <p>Ataque a la Puerta de enlace</p>
<pre> suricataids@debianS: ~ Archivo Editar Ver Buscar Terminal Ayuda } } Ataque a la maquina anfitrión { "timestamp": "2019-08-06T17:04:30.354455+0200", "flow_id": 1091804692693994, "event_type": "flow", "src_ip": "172.16.12.13", "src_port": 56264, "dest_ip": "172.16.12.8", "dest_port": 61121, "proto": "UDP", "app_proto": "failed", "flow": { "pkts_toserver": 1, "pkts_toclient": 0, "bytes_toserver": 60, "bytes_toclient": 0, "start": "2019-08-06T17:03:59.869354+0200", "end": "2019-08-06T17:03:59.869354+0200", "age": 0, "state": "new", "reason": "timeout", "alerted": false } } </pre>	<pre> suricataids@debianS: ~ 50x23 { "timestamp": "2019-08-07T05:41:39.015010+0200", "flow_id": 1288585889601321, "event_type": "flow", "src_ip": "147.13.119.146", "src_port": 7727, "dest_ip": "172.16.12.15", "dest_port": 80, "proto": "TCP", "flow": { "pkts_toserver": 1, "pkts_toclient": 1, "bytes_toserver": 60, "bytes_toclient": 54, "start": "2019-08-07T05:40:38.546601+0200", "end": "2019-08-07T05:40:38.546656+0200", "age": 0, "state": "closed", "reason": "timeout", "alerted": false } }, </pre> <p>Ataque al host que aloja al IDS</p>

Figura 30. Registro de alertas del ataque DoS

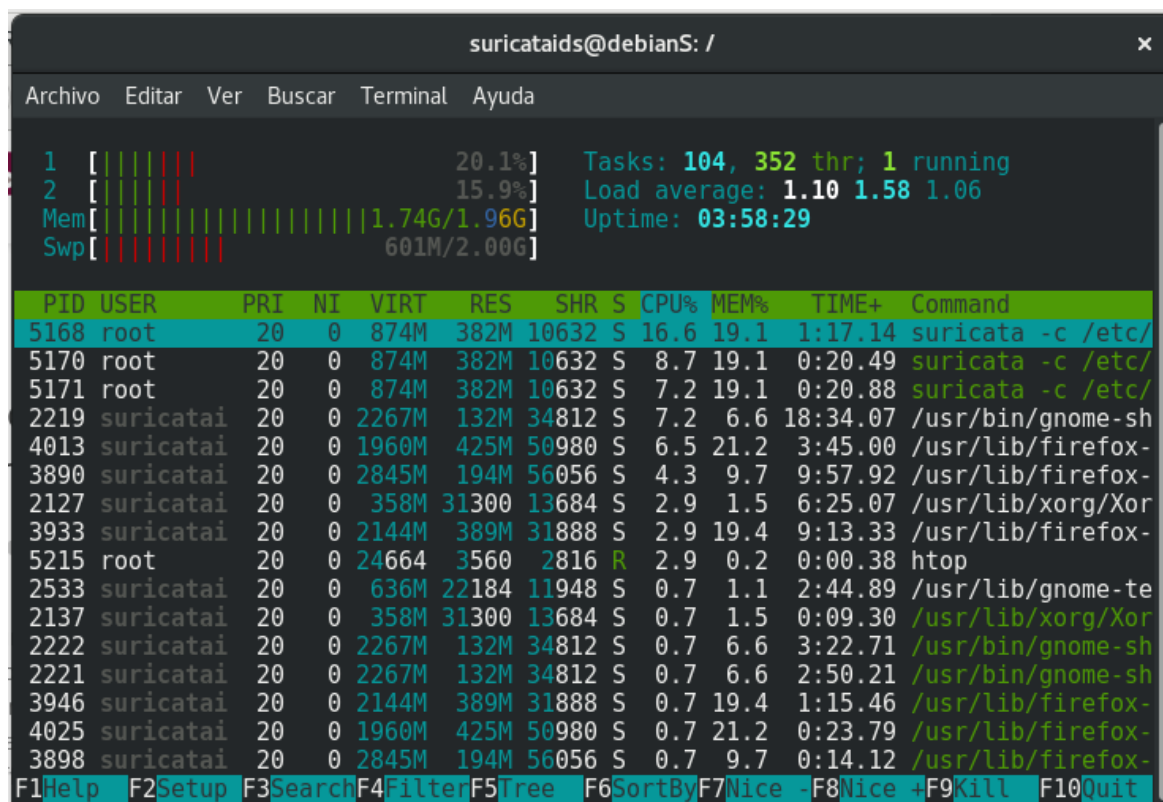


Figura 31. Consumo de recursos

Durante el ataque (un lapso de 10 min por elemento de la red atacado) el consumo de recursos por parte de IDS fue de un 16.6 % CPU y 19.3% de memoria, este aumento de uso en el CPU se evidencia gracias a que el IDS pudo parar el ataque dos. Ver figura 30.

Ataque con Slowloris: Es un script hecho en Perl implementa una potente e inteligente manera de generar una denegación de servicio sobre un servidor web Apache. Para ello, se basa en la cantidad de peticiones que es capaz de mantener un servidor web de forma concurrente.

- **Paso 1.** Ya instalada la aplicación en el dispositivo móvil ejecutamos el siguiente comando para realizar el ataque `slowloris -p 80 -s 15 172.16.12.1` como se observa en la figura 32.

```
15:49 [calendar] [terminal] [wifi] [signal] [battery]
Report issues at https://termux.com/issues

$ pip install slowloris
Collecting slowloris
  Downloading https://files.pythonhosted.org/packages/a6/37/5ae3d027727122039f52a22d278f1d73f564e03e5fdb93f10e3a2f26aa06/Slowloris-0.2.0.tar.gz
Installing collected packages: slowloris
  Running setup.py install for slowloris ... done
Successfully installed slowloris-0.2.0
$ slowloris -p 80 -s 15 172.16.12.1
[30-08-2019 15:44:55] Attacking 172.16.12.1 with 15 sockets
.
[30-08-2019 15:44:55] Creating sockets...
[30-08-2019 15:44:55] Sending keep-alive headers... Socket
count: 15
[30-08-2019 15:45:10] Sending keep-alive headers... Socket
count: 15
[30-08-2019 15:45:25] Sending keep-alive headers... Socket
count: 15
[30-08-2019 15:45:40] Sending keep-alive headers... Socket
count: 15
[30-08-2019 15:45:55] Sending keep-alive headers... Socket
count: 15
[30-08-2019 15:46:10] Sending keep-alive headers... Socket
count: 15
[30-08-2019 15:46:25] Sending keep-alive headers... Socket
count: 15
[30-08-2019 15:46:40] Sending keep-alive headers... Socket
count: 15
[30-08-2019 15:46:55] Sending keep-alive headers... Socket
count: 15
[30-08-2019 15:47:10] Sending keep-alive headers... Socket
count: 15
[30-08-2019 15:47:45] Sending keep-alive headers... Socket
count: 15
[30-08-2019 15:48:02] Sending keep-alive headers... Socket
count: 15
[30-08-2019 15:48:17] Sending keep-alive headers... Socket
count: 15
[30-08-2019 15:48:32] Sending keep-alive headers... Socket
count: 15
[30-08-2019 15:48:47] Sending keep-alive headers... Socket
count: 15
[30-08-2019 15:49:02] Sending keep-alive headers... Socket
count: 15
█

ESC  [terminal] CTRL  ALT  —  ↓  ↑
```

Figura 32. Ataque DOS con Slowloris

Este ataque se lo realizó en un lapso de 5 min enviando 100 paquetes a través de 4 hilos cada 60 segundos logrando saturar la puerta de enlace y denegando el servicio de internet. En la figura 33 se observa la alerta que esta lanza al darse el ataque, en este consta el tipo de evento la dirección ip del dispositivo que lo genera y el protocolo.

```
suricataids@debianS: //var/log/suricata
suricataids@debianS: //var/log/suricata 87x28
}
}
}
{
  "timestamp": "2019-08-30T23:32:55.007232+0200",
  "flow_id": 2152098954881894,
  "event_type": "flow",
  "src_ip": "172.16.12.3",
  "src_port": 1900,
  "dest_ip": "239.255.255.250",
  "dest_port": 1900,
  "proto": "UDP",
  "app_proto": "failed",
  "flow": {
    "pkts_to_server": 18,
    "pkts_to_client": 0,
    "bytes_to_server": 2484,
    "bytes_to_client": 0,
    "start": "2019-08-30T23:31:38.797542+0200",
    "end": "2019-08-30T23:32:24.270136+0200",
    "age": 46,
    "state": "new",
    "reason": "timeout",
    "alerted": false
  }
}
```

Figura 33. Alerta de ataque DOS

Escenario 3. Man in the Middle

Para este ataque se hace uso de la herramienta Ettercap que es una suite completa para el ataque de hombre en el medio. Cuenta con la detección de conexiones en vivo, filtrado de contenido, admite la disección activa y pasiva de muchos protocolos e incluye muchas funciones para el análisis de redes tanto cableadas como Wifi y hosts.

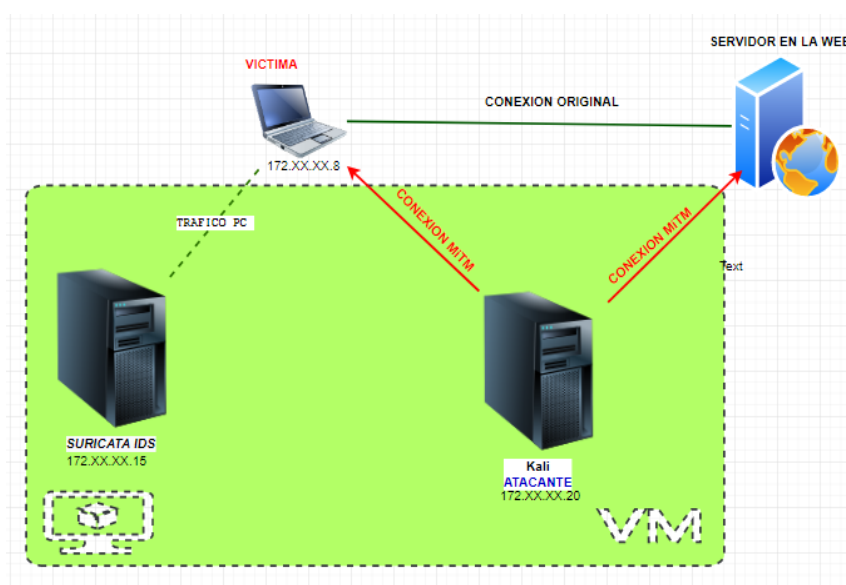


Figura 34. Ataque MITM

Paso 1. Abrir el programa Ettercap e ir a la sección Sniff ver figura 35 se selecciona la opción Unified Sniffing, se elige la interfaz a utilizar para llevar a cabo el ataque ver figura 36.



Figura 35. Vista principal Ettercap

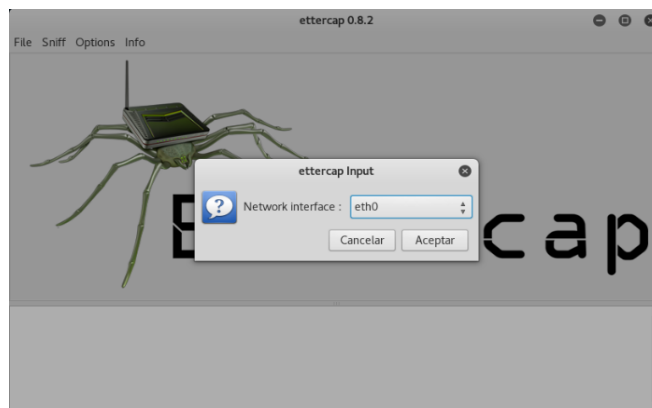


Figura 36. Selección de Interfaz

Paso 2. En la barra de menú seleccionar “Scan for Host” con esta opción Ettercap escanea los hosts para visualizar todas las máquinas conectadas a la red, una vez escaneados se elige “host list” muestra los dispositivos que se encuentran en nuestra red, ver Figura 37.

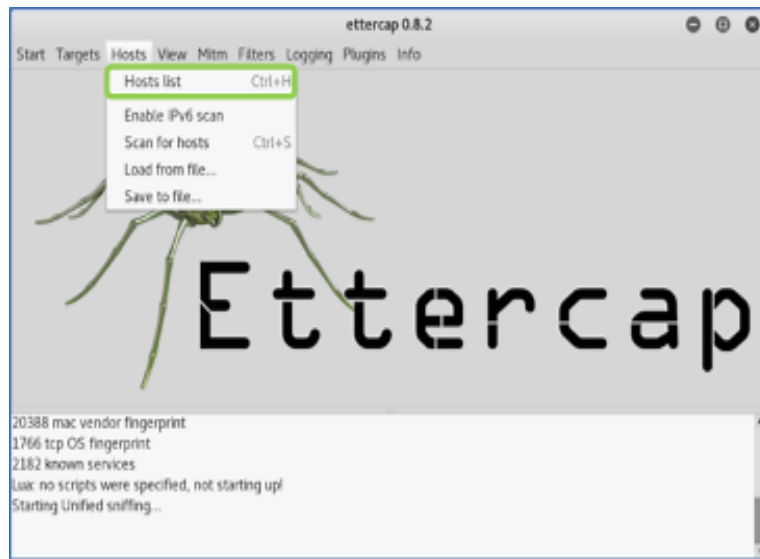


Figura 37. Selección de escaneos de host

Paso 3. Una vez identificada a la víctima se coloca su dirección IP a Target 1, junto a su puerta de enlace en Target 2 ver figura 38.

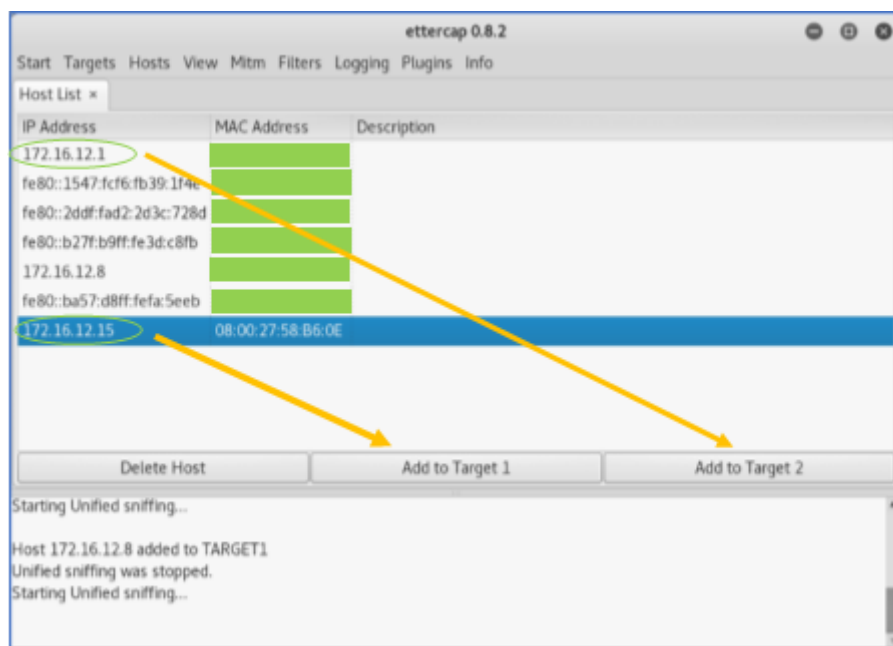


Figura 38. Lista de host en la red

Paso 4. En la sección mitm se despliega una opción “ARP poisoning” ver Figura 39, ya seleccionada pide el tipo de ataque arp a realizar, en este caso se ha escogido “Sniff remote connection” por la cual solo realiza el ataque para el tráfico de los hosts dados ver figura 40.

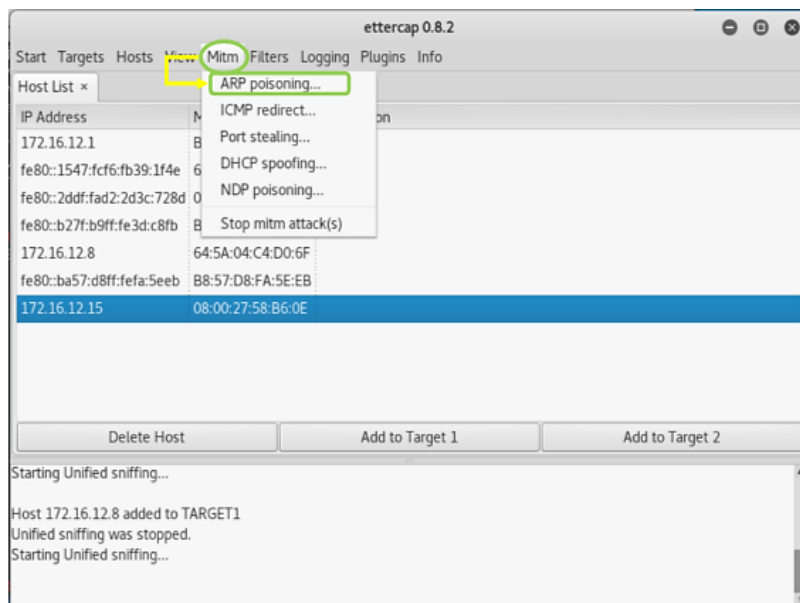


Figura 39. Ataque MITM por envenenamiento arp

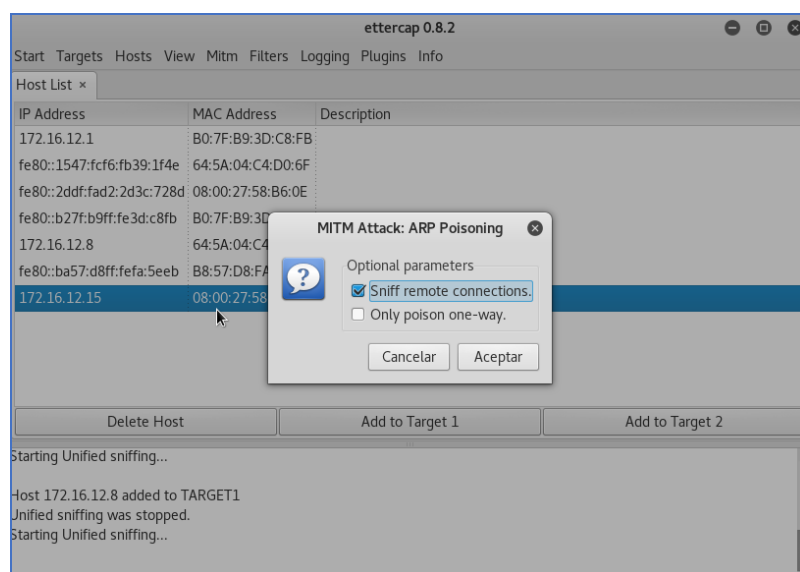


Figura 40. Ejecución de ataque

Finalmente, en la Figura 41 se observa las conexiones y los puertos por los que se da el ataque, los datos presentados comprueba que el ataque ha capturado el usuario y password al logearse ver figura 42.

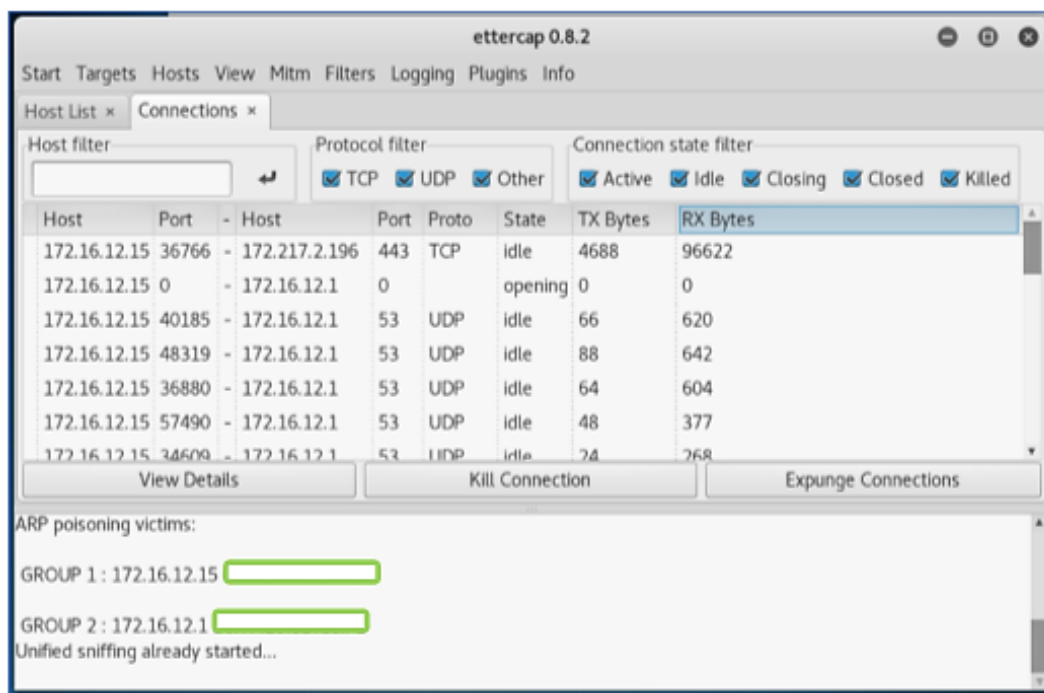


Figura 41. Conexiones en el host infectado

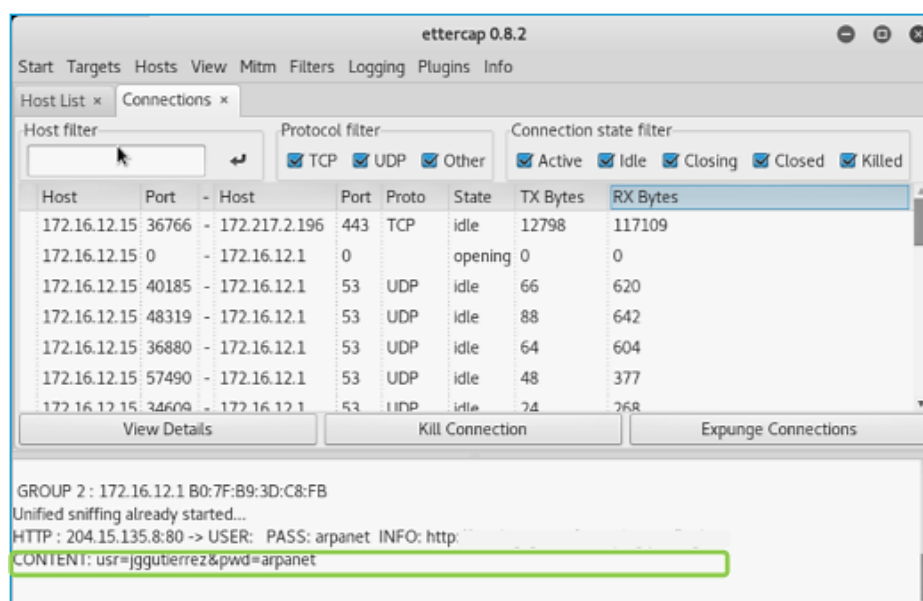


Figura 42. Captura de información ataque

Resultado del IDS frente al ataque

Como se puede observar en la figura 40 y 41 el ataque fue efectivo obteniendo las credenciales al ingresar a una página, la alerta emitida por el ids se puede observar al en el archivo `eve.json` en la ruta `cd var/log/suricata` este muestra que el ataque tiene una severidad de tipo 3, el protocolo por el cual se ejecutó y la ip desde donde se realizó el ataque, y la ip de donde se obtiene la información.

este, se obtuvo como resultado que Suricata para detección de ataques desde la red externa se deberá colocarse detrás del firewall esto se realiza con el fin de evitar que el IDS priorice falsos positivos en lugar de las alertas reales.

Una vez efectuadas las intrusiones a la red y el IDS Suricata determinó que el ataque de exploración y MiTM tuvieron un mayor grado de severidad (grado 3), indicando que las firmas con las cuales el IDS se desplegó cumple su función al alertar de manera exitosa los ataques realizados, brindar información detallada sobre estos, así como asignarles un grado de rigor al momento de presentarse en los diferentes archivos de alerta.

En cuanto al ataque DoS al ser realizado a través de Metasploit y Slowloris, el IDS en su archivo eve.json mostró las respectivas alertas sobre estos ataques, por su parte el sistema de detección al estar en modo IDS su función principal es el de detectar y alertar estas anomalías, dado que el presente trabajo se enfocó más en la configuración IDS y las reglas con las que este se puso en marcha.

7.DISCUSIÓN

El presente trabajo de titulación denominado “Configuración de una herramienta open source para la detección de intrusos en redes Wifi caso de estudio Suricata”, da como resultado final la generación de las respectivas alertas frente a cada ataque al que fue expuesto el IDS Suricata en su versión 4.1.3

El desarrollo del presente trabajo de titulación se basa en el cumplimiento de cada uno de los objetivos específicos que fueron abarcados en su totalidad tal y como se describe a continuación:

- 1. Objetivo específico 1.** Estado actual de herramientas open source para la detección de intrusos.

Este objetivo se lo realizó de la siguiente forma:

Se realizó una búsqueda bibliográfica en tesis, artículos y trabajos de titulación, acerca de los sistemas de detección de intrusos. Se realiza una comparativa entre varias soluciones con respecto al IDS Suricata, se exponen ventajas y desventajas de los sistemas de detección de intrusos, para el cumplimiento de este objetivo se realizó una revisión sistemática de literatura (ver anexo 1), con

la cual se obtuvo información relevante y pertinente respecto a los IDS, tomándose como mejor IDS SURICATA.

2. Objetivo específico 2. Diseñar un esquema de red y entorno virtual de equipos,

Para el cumplimiento de Este objetivo se lo abordó en 3 partes:

Selección del hardware para la instalación de la herramienta

Para el cumplimiento de este objetivo se realiza una búsqueda sobre la herramienta Suricata en los repositorios oficiales, así como la revisión en diferentes fuentes bibliográficas determinando requisitos mínimos, y sistema operativo más adecuado para su implementación. En la selección de la herramienta de virtualización se considera a VirtualBox es una aplicación multiplataforma, dentro de esta se instaló Kali Linux que actuó como máquina atacante y Debian 9 en donde se instaló la herramienta.

Definir el esquema de red

Esquema de red se realizó con una Wlan, se consideró este tipo de red debido a que representan una solución tecnológica de gran interés en el sector de las comunicaciones inalámbricas de banda ancha principalmente por: su movilidad, su fácil instalación y flexibilidad ya que estas no necesitan un medio físico guiado para llevar la información de un punto a otro.

En una red Wlan cabe destacar que la seguridad de esta red está comprometida si no es debidamente protegida pues un atacante con un adaptador inalámbrico podría comunicarse con un punto de acceso privado si no se disponen de las medidas de seguridad adecuadas.

Diseño del entorno virtual

Para cumplir con este aspecto se hace uso de la información y las herramientas establecidas en las partes 1 y 2

3. Objetivo específico 3. Configuración y evaluación la herramienta Suricata.

Este objetivo se abordó en 2 partes:

Establecer las reglas necesarias para configuración sistema IDS.

Una vez instalada y configurada la herramienta Suricata por defecto tiene un conjunto de reglas o firmas con las cuales puede trabajar sin ningún problema, a excepción de que estas se vuelven obsoletas con el tiempo y su actualización implica descargar y desempaquetar para proceder a su actualización o hacer uso de alguna herramienta aparte para que se encargue de ello. Dado que desde la versión 4.0 en el IDS Suricata viene integrada un complemento suricata -update que es una opción para la administración de firmas la cual se optó por hacer uso de ésta dada su facilidad de instalación y actualización.

Evaluación de la herramienta.

Durante la generación de los ataques el IDS Suricata realizó correctamente el registro de estos, en el escenario 1 escaneo con Nmap se observa en el archivo de alertas eve.json la información correspondiente al ataque como: dirección ip, tipo de ataque generado, nivel de severidad, firma con la cual coincide, así como el barrido a la red es decir los puertos por el cual el atacante obtuvo información sobre la red ver figura 22 sección resultados pág. 52.

Durante el ataque de denegación de servicio se realizó en cuatro elementos de la red como se ve en la figura 30 y 32 sección resultados pág. 55, la generación de alertas por cada uno brindó información importante respecto a cada elemento atacado dando como resultado una alerta falsa es decir que la red no se vio afectada de manera significativa.

Al ejecutar el ataque MiTM se hace uso de la herramienta Ettercap esta realizó un escaneo con el propósito de obtener los hosts conectados a la red, se selecciona a la víctima llevando a cabo el ataque, comprobando así que el IDS al registrar la alerta le asigna una prioridad más alta respecto a los otros ataques, como se observa en los archivos eve.log ver figura 43 sección resultados pág. 62.

8. CONCLUSIONES

- El análisis previo para la implementación del IDS SURICATA contribuyó a la configuración y elaboración del escenario para su implementación, así mismo se evidencio que este sistema a partir de la versión 4.0 cuenta con una herramienta para la descarga y actualización de sus reglas.
- La implementación del IDS de monitoreo Suricata ha mejorado en un 80% respecto a sus versiones anteriores dentro del rango de experimentación por lo que se concluye como exitoso la implementación del mismo.
- Luego de las pruebas realizadas en el entorno local se concluye que IDS Suricata es una herramienta eficaz al realizar el monitoreo, detección y generación de alertas frente a los ataques de exploración de red, DoS, y MiTM.
- Se consideró que el tiempo en que tardó el IDS en alertar los distintos ataques fue inmediato sobre todo en el ataque de exploración y MiTM a diferencia del ataque de denegación de servicio que alertó en cuanto esté

saturó por completo la red al aumentar el número de peticiones TCP enviadas a la puerta de enlace inhabilitando así el servicio de internet.

9. RECOMENDACIONES

- Es recomendable tener correctamente configurado las reglas internas del IDS ya que al ser desplegado en un entorno real y no contar con los requisitos necesarios este no responderá adecuadamente a lo que se desea proteger.
- Se recomienda colocar al IDS Suricata en una implementación real detrás del firewall puesto que en base a la experimentación realizada esto servirá para minimizar cualquier actividad que no sea intrusiva debido a que el firewall filtra los paquetes con posibles amenazas, dejando así al IDS con la tarea de revisar la información que el firewall ya no logra procesar y que representan una amenaza real a la red.
- Se recomienda realizar pruebas en un entorno real con la finalidad de ver otros resultados, así como su integración con herramientas que faciliten la administración de los datos que el IDS genera.

10. BIBLIOGRAFIA

- [1] C. A. Ocampo, Y. Viviana, C. Bermúdez, y G. R. Solarte Martínez, «Sistema de detección de intrusos en redes corporativas Intrusion Detection System in Corporate Networks», 2017.
- [2] R. Salazar, «Sistema de detección de intrusos mediante modelado de URI», 2015.
- [3] Emmanuel Renán Cetina González Sonia González Rosales Fernando Ruiz López, «Sistema Preventor de intrusos para la Esime Zacatenco», 2010.
- [4] J. A. M. V. Edwin Santiago Acosta Cortez, «Análisis e implementación de un DIDS para generación de firmas de comportamientos anómalos en la red del edificio matriz de la empresa eléctrica Quito», 2015.
- [5] César Alejandro Vallejo de la Torre Patricia María Marcillo Sánchez Martha Viviana Uvidia Vélez, *Sistemas de prevención de intrusos (IDS) en la gestión de la información*, vol. 53, n.º 9. 2013.
- [6] J. Eduardo, A. Pucha, E. L. Tribunal, D. E. L. Trabajo, y D. E. T. Certifica, «Evaluación de las funcionalidades de los sistemas de detección de intrusos basados en la red de plataformas open source utilizando la técnica de detección de anomalías», 2018.
- [7] L. G. M. Montalvo, «Módulo De Seguridad Informática», *Cybsec*, p. 16, 2008.

- [8] J. Polvereda, «Sistema De Monitorización Del Ids Snort», 2017.
- [9] Jayner Ahmed Herrera Quintero, «Implementación de un sistema de detección de intrusos en la red interna de la alcaldía de Montería usando software libre», 2018.
- [10] I. Red Hat, «Red Hat Enterprise Linux 4 Manual de seguridad», 2005.
- [11] C. M. F. Díaz, «Implantación de un sistema de seguridad perimetral», 2013.
- [12] G. R. Solarte Martinez, C. A. Ocampo, y Y. V. Castro Bermúdez, «Sistema de detección de intrusos en redes corporativas», *Sci. Tech.*, vol. 22, n.º 1, p. 60, 2017.
- [13] M. I. G. García, «Utilización de Sistemas de Detección de Intrusos como Elemento de Seguridad Perimetral», 2003.
- [14] M. Javier y O. D. E. Mora, «Modelo de gestión de seguridad a través del uso de buenas prácticas de ITIL y COBIT enfocados a los sistemas de detección de intrusos», 2015.
- [15] C. B. Segu.Info, «Detección de Intrusos en Tiempo Real», *Segu-Info*, 2009.
- [16] I. F. S. Luis Andrés Balseca Guzmán Ing. Carlos Romero, «Estado del arte en la detección de intrusiones en redes 802.1».
- [17] Rafael Luis Moscote Medina, «Sistema de detección y prevención de intrusos ips para la Vlan de servidores de la sociedad minera de Santander s.a.s. en Bucaramanga (Santander)», 2013.
- [18] J. L. Gomez, *Optimización de sistemas de detección de intrusos en red utilizando técnicas computacionales avanzadas*. 2009.
- [19] J. A. A. HERRERA y A. A. J. M. F. M. O. FLORES, «Adaptación del IDS/IPS Suricata para que se pueda convertir en una solución empresarial», p. 149, 2011.
- [20] F. L. Ortiz, «El estándar IEEE 802.11 Wireless LAN», p. 23.
- [21] C. F. Borghell, «Amenazas Lógicas - Tipos de Ataques», *Segur. Informática*, p. 2009, 2009.
- [22] Martha Yolanda Carranza Suica Rosley Amparo Naranjo Barragán, «“Modelo basado en las técnicas de minería de datos aplicada a la detección de ataques en las redes de datos de la facultad de informática y electrónica”», 2014.
- [23] Villarreal Roberto Fabián Cárdenas, «Propuesta de una solución de seguridad que provea una respuesta activa frente a ataques de denegación de servicio basada en la integración de herramientas open source sobre un prototipo de red jerárquica», 2016.
- [24] A. Khalimonenko, J. Strohschneider, y O. Kupreev, «DDoS attacks in Q4 2016», 2017.
- [25] B. Brumen y J. Legvart, «Performance analysis of two open source intrusion detection systems», p. 6, 2016.
- [26] B. Kitchenham, «Kitchenham_Procedures for Performing Systematic Reviews_2004.pdf».

11. ANEXOS

A continuación, se presenta un listado de los anexos correspondientes al presente trabajo de titulación.

Anexo 1. Revisión sistemática de literatura

Anexo 2. Instalación del IDS Suricata

Anexo 1. Revisión sistemática de literatura

