

2º ASIR

## Proyecto fin de grado

Suricata: un IDS/IPS gratuito para proteger tu red corporativa. Implantación de un sistema de detección de intrusiones de ataques basando en Suricata.

---



IES Medina Azahara

Yelyzaveta Kramarenko Volodymyrivna

fecha de entrega (19/06/2024)

---

---

# Índice

1. Contexto y justificación.....	4
1.1 Descripción del problema proyecto.....	4
1.2 Objetos potenciales de influencia maliciosa.....	4
1.3 Objetivos del proyecto.....	5
1.4 Especificación de requisitos.....	7
1.5 Alternativas y propuestas de solución.....	8
2. Planificación del proyecto.....	10
2.1 Temporalización.....	10
3. Presupuesto.....	11
3.1 Presupuesto del Proyecto Suricata.....	11
4. Memoria técnica.....	12
4.1 Sistema de detección de intrusos.....	12
4.2 Estructura y arquitectura del sistema general de detección de intrusos.....	13
4.3 Categorización de los sistemas de detección de ataques.....	15
4.4 Propósito de HIDS y NIDS.....	17
4.5 Comparación de NIDS y HIDS.....	18
4.6 Sistema de detección de intrusos perfecto.....	19
4.7 Comparación de firewall e IDS.....	20
4.8 Comparación IDS e IPS.....	22
5. Sistema de detección de ataques - Suricata.....	24
5.1 Características del sistema de detección de ataques Suricatas.....	24
6. Creación de la máquina virtual Ubuntu 22.04.....	26
6.1 Requisitos del sistema.....	26
6.2 Instalacion de Ubuntu.....	30
7. Instalación de Suricata.....	35
7.1 Instalación de SSH.....	35
7.2 Instalación de Suricata y los paquetes necesarios.....	37

---

7.3 Configuración de Suricata.....	41
8. Seguimiento y control.....	48
9. Fuentes de documentación.....	51
10. Conclusion .....	52

---

# 1. Contexto y justificación

## 1.1 Descripción del problema proyecto

En el pasado, los cortafuegos y el software antivirus habituales eran suficientes para proteger las redes locales, pero ya no son lo suficientemente eficaces contra los ataques de ciberdelincuentes modernos y el malware recientemente popular. El firewall bien conocido a todos lo que hace es solamente analiza los encabezados de los paquetes, permitiéndolos o bloqueándolos de acuerdo con un conjunto formal de reglas.

No sabe nada sobre el contenido de los paquetes y tampoco puede reconocer acciones aparentemente legítimas de los atacantes. Los programas antivirus no siempre detectan malware, por lo que los administradores se enfrentan a la tarea de monitorear actividades inusuales y poner en cuarentena rápidamente los servidores infectados.

## 1.2 Objetos potenciales de influencia maliciosa

Los factores que determinar la planificación y la implementación de cualquier ciberataque normalmente son estos:

- motivos personales (venganza, envidia, etc.);
- motivos financieros;
- competencia desleal en determinados nichos de negocio;
- motivos políticos y religiosos.

Así, dependiendo del motivo que motiva el ataque, el ciber-atacante puede tomar las siguientes acciones (Fig.: 4.1.1):

1. Robo o copia de cierta información.
2. Obtener el control de los recursos de información de la red objetivo.
3. Obtener acceso a las cuentas financieras de una persona física o jurídica.

- 
4. Creación de un bot, es decir, un conjunto de nodos infectados. Esto se puede utilizar para realizar ataques posteriores de diferentes tipos y direcciones
  5. Deshabilitar el sistema de información de la infraestructura de red atacada o sus componentes individuales (subsistemas, estaciones de trabajo, servidores, etc.).
  6. Eliminar o modificar información de la base de datos en el sistema de destino.

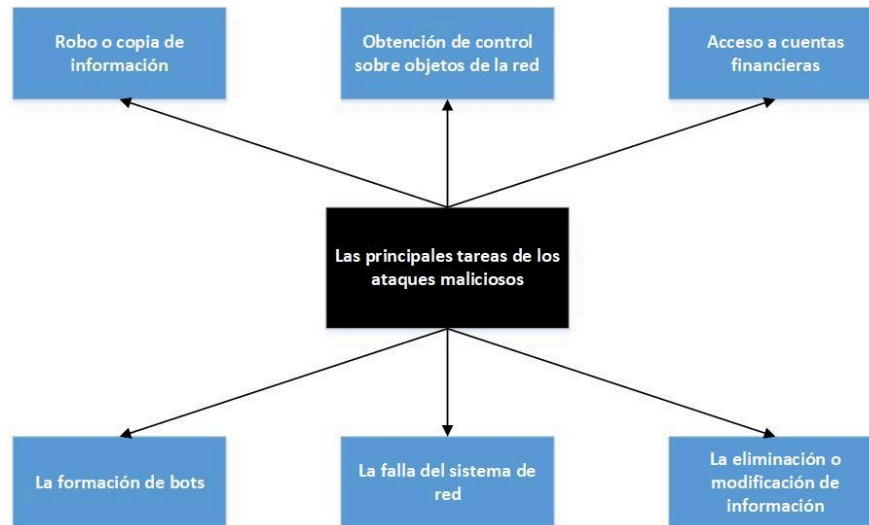


Figura 4.1.1 - Las principales acciones de los ciber-atacantes

### 1.3 Objetivos del proyecto

El propósito de este proyecto consiste en implementar y optimizar el sistema de vigilancia y detección de intrusiones (IDS/IPS) utilizando Suricata, seleccionando esta herramienta en lugares de alta demanda.

Contraste entre Snort y Suricata:

Suricata está concebida con el propósito de aprovechar las arquitecturas de hardware modernas y brinda habilidades de multiprocesamiento nativas, lo cual posibilita el uso eficiente de múltiples núcleos de CPU. Esto la hace particularmente apropiada para situaciones de alta demanda, en las que se requiere el procesamiento de grandes volúmenes de tráfico de red sin pérdida de capacidad de rendimiento. Asimismo, aunque Snort ha sido una marca

---

comercial durante muchos años, su capacidad de procesamiento puede verse reducida en comparación con Suricata, especialmente en sistemas con múltiples núcleos de CPU, debido a que su arquitectura no está optimizada de forma nativa para el procesamiento paralelo.

Asimismo, Suricata dispone de herramientas integradas de inspección profunda de paquetes (DPI), análisis de protocolos y asistencia para scripts Lua con el propósito de optimizar la detección de amenazas. Estas características avanzadas brindan una detección más precisa y una respuesta más rápida ante incidentes de seguridad. Dado que Snort se caracteriza por ser robusto en la detección basada en firmas, carece de algunas de las funciones avanzadas de DPI y la capacidad de personalizar que Suricata brinda sin la necesidad de extensiones adicionales.

Suricata se ajusta a las normas de Snort, lo cual posibilita una transición rápida para los equipos que ya utilizan Snort, aprovechando la amplia base de datos de firmas existentes. Asimismo, Suricata cuenta con una comunidad activa que brinda asistencia a actualizaciones y mejoras constante. Dado que Snort también dispone de una comunidad activa y una amplia base de firmas, la falta de algunas de las capacidades modernas sin integraciones adicionales puede obstaculizar su capacidad en entornos con requisitos complejos de seguridad.

La decisión de Suricata para este proyecto se fundamenta en su capacidad para gestionar volúmenes de tráfico con un rendimiento superior, sus habilidades avanzadas de inspección de paquetes y detección de amenazas, y su conformidad con las normas de Snort, lo que posibilita la adopción sin malgastar las inversiones previas en definiciones de amenazas. La implementación de Suricata no solo incrementará la capacidad de detectar y responder ante incidentes de seguridad, sino que también garantizará que el sistema de vigilancia esté debidamente adaptado para escalas futuras y desafíos en la infraestructura de red.

## **1.4 Especificación de requisitos**

### **Requisitos de Suricata**

- 
- **Monitorización en Tiempo Real:** El sistema debe ser capaz de monitorizar y analizar el tráfico de red en tiempo real para detectar amenazas y actividades sospechosas.
  - **Alertas de Seguridad:** El sistema debe generar alertas automáticas cuando se detecten amenazas y permitir la configuración de notificaciones vía correo electrónico o SMS.
  - **Registro de Eventos:** El sistema debe registrar detalladamente todos los eventos de seguridad detectados, incluyendo detalles como la fuente, destino, tipo de amenaza y hora de ocurrencia.
  - **Análisis de Protocolos:** El sistema debe ser capaz de analizar múltiples protocolos de red y detectar anomalías en ellos.
  - **Actualización de Reglas:** El sistema debe permitir la actualización y personalización de las reglas de detección para adaptarse a nuevas amenazas.

### Requisitos de Usabilidad

- **Manual de Usuario:** El sistema debe contar con un manual de usuario detallado que explique su instalación, configuración y operación.
- **Interfaz Gráfica:** El sistema debe disponer de una interfaz gráfica de usuario (GUI) intuitiva para facilitar la configuración y monitoreo.
- **Reportes Personalizables:** El sistema debe permitir la generación de reportes personalizables sobre las actividades de seguridad detectadas.

### Requisitos de Hardware

- **Compatibilidad con Equipos Existentes:** El sistema debe ser compatible con los routers Cisco actualmente instalados en el datacenter.
- **Requisitos Mínimos de Hardware:** El sistema no debe requerir más de 8GB de memoria RAM para su funcionamiento óptimo.
- **Ejecutabilidad en Máquina Virtual:** El sistema debe poder ser ejecutado en una máquina virtual dentro del servidor Proxmox del centro.

### Requisitos Legales

- 
- **Cumplimiento de Normativas:** El sistema debe cumplir con la normativa de protección de datos de carácter personal, asegurando la privacidad y seguridad de la información tratada.
  - **Regulaciones Locales:** El sistema debe cumplir con todas las regulaciones locales e internacionales aplicables al manejo y procesamiento de datos como por ejemplo, cumplir con el Esquema Nacional de Seguridad (ENS).

## Requisitos de Escalabilidad

- **Escalabilidad del Sistema:** El sistema debe ser escalable para manejar el crecimiento esperado de la infraestructura de red, permitiendo la adición de más sensores y nodos de monitoreo sin pérdida de rendimiento.
- **Capacidad de Integración:** El sistema debe ser capaz de integrarse con otras herramientas y plataformas de seguridad ya existentes en la infraestructura de la organización.

## 1.5 Alternativas y propuestas de solución

Snort y Suricata son dos sistemas de detección de intrusiones (IDS) y sistemas de prevención de intrusiones (IPS) populares. Ambos programas se utilizan para detectar y responder a amenazas en el tráfico de la red, pero hay varias razones por las que Suricata puede considerarse superior en algunos aspectos a Snort.

### 1. Soporte para subprocesos múltiples

**Suricata:** Una de las fortalezas clave de Suricata es su soporte para subprocesos múltiples. Suricata puede utilizar todos los núcleos de procesador disponibles para analizar el tráfico de la red, lo que le permite procesar grandes cantidades de datos de manera mucho más rápida y eficiente. Esto es especialmente útil para redes grandes con mucho tráfico.

**Snort:** Snort, por el contrario, no admite subprocesos múltiples. Esto significa que procesa el tráfico en un solo núcleo de CPU, lo que puede convertirse en un cuello de botella en redes de alta carga.



---

## 2. Soporte de varios protocolos de red.

**Suricata:** Suricata cuenta con soporte avanzado para protocolos de red. Es capaz de analizar protocolos en varias capas del modelo de red OSI, incluidos HTTP, TLS, FTP, SMTP, SMB y otros. Esto permite a Suricata comprender mejor el tráfico y detectar amenazas más complejas.

**Snort:** Snort también admite análisis multiprotocolo, pero sus capacidades pueden ser más limitadas en comparación con Suricata.

## 3. Integración con otras herramientas y soporte JSON

**Suricata:** Suricata proporciona soporte integrado para generar resultados en formato JSON, lo que facilita la integración con sistemas modernos de análisis de registros y SIEM (sistemas de gestión de eventos e información de seguridad). Esto permite una fácil integración de Suricata con herramientas como Elasticsearch, Logstash y Kibana (ELK Stack).

**Snort:** Snort puede integrarse con otras herramientas a través de personalizaciones y módulos adicionales, pero su integración no es tan simple y flexible como la de Suricata.

## 4. Comunidad y apoyo

**Suricata:** El proyecto Suricata cuenta con el apoyo y desarrollo activo de la comunidad Open Information Security Foundation (OISF). Esto garantiza que constantemente se agreguen nuevas características y mejoras, y que cualquier problema o vulnerabilidad se solucione rápidamente.

**Snort:** Snort también cuenta con una comunidad grande y activa, pero se desarrolla bajo el paraguas de Cisco, lo que puede significar algunas limitaciones en el desarrollo o implementación de nuevas características que pueden ser una prioridad para la empresa.

Suricata es un sistema IDS/IPS más avanzado y potente en comparación con Snort debido a su soporte para subprocesos múltiples, soporte de protocolo de red avanzado, fácil integración con otras herramientas a través de JSON y soporte activo de la comunidad OISF. Si bien Snort también es una herramienta poderosa con una gran comunidad de usuarios, Suricata puede brindar un mejor rendimiento y flexibilidad para las necesidades actuales de seguridad de la red.

---

## 2. Planificación del proyecto

### 2.1 Temporalización

Semana	Fecha	Tareas
<u>Semana 1</u> (3 de mayo - 10 de mayo)	3 de mayo	Preparación inicial, definición de objetivos del proyecto
	6 de mayo - 7 de mayo	Análisis de requerimientos, creación de una tarea técnica
	8 de mayo - 9 de mayo	Planificación del proyecto, creación de un cronograma de trabajo
	9 de mayo - 10 de mayo	Búsqueda de información y planificación de formato estructurado
<u>Semana 2</u> (13 de mayo - 17 de mayo)	13 de mayo - 17 de mayo	Búsqueda de información y diseño de figuras necesarias para el proyecto
<u>Semana 3</u> (20 de mayo - 24 de mayo)	20 de mayo - 21 de mayo	Descarga e instalación de Virtualbox
	22 de mayo - 23 de mayo	Instalación de máquina virtual Ubuntu para Suricata
	24 de mayo	Pruebas (pruebas unitarias, pruebas de integración), corrección de errores
<u>Semana 4</u> (27 de mayo - 31 de mayo)	27 de mayo - 29 de mayo	Instalación y configuración de Suricata
	29 de mayo - 31 de mayo	Ataque artificial a Suricata

<u>Semana 5</u> (3 de junio - 7 de junio)	3 de junio - 7 de junio	Finalización de las pruebas, corrección de errores residuales
<u>Semana 6</u> (10 de junio)	10 de junio	Preparación de presentación de proyecto

### Fechas importantes

Revisión de progreso semanal: todos los domingos para resumir la semana

Revisión de arquitectura y distribuciones: 19 de mayo.

Revisión final de todo el proyecto: 11 de junio.

## 3. Presupuesto

### 3.1 Presupuesto del Proyecto Suricata

<b>Horas de Trabajo</b>	
Periodo: 3 de mayo al 10 de junio (27 días laborales)	
Horas por semana:	10 horas
Total de horas	27 días * 2 horas/día = 54 horas
Tarifa por hora:	15 €
Costo total de horas de trabajo:	54 horas * 15€/hora = 810€
<b>Hardware</b>	
<b>Máquina virtual de Amazon (AWS)</b>	
Costo mensual:	30€
Costo por 1.5 meses:	30€ * 1.5 = 45€
<b>Portátil</b>	
Costo estimado:	800€
<b>WiFi (módem y plan de datos)</b>	

Costo mensual del plan:	40€
Costo por 1.5 meses:	40€ * 1.5 = 60€
<b>Licencias de software</b>	
Software de desarrollo (IDE, herramientas de desarrollo, etc.):	50€
<b>Tiempo de Configuración e Ingeniería</b> Configuración inicial de la máquina virtual y entorno de desarrollo.	
Horas estimados:	20 horas
Costo:	20 horas * 20€/hora = 400€
<b>Costo Total</b>	
Total: <b>2165€</b>	

## 4. Memoria técnica

### 4.1 Sistema de detección de intrusos

Un sistema de detección de ataques es una aplicación o dispositivo que monitorea el tráfico de red entrante y saliente, analiza continuamente la actividad en busca de cambios en los patrones y alerta a un administrador cuando se detecta un comportamiento inusual. Luego, el administrador revisa las alarmas y toma medidas para eliminar la amenaza.

Por ejemplo, un IDS puede inspeccionar los datos transportados por el tráfico de la red para ver si contienen malware conocido u otro contenido malicioso. Si detecta este tipo de amenaza envía una alerta al equipo de seguridad para que puedan investigarla y solucionarla. Una vez alertado, el equipo debe actuar rápidamente para evitar que un ataque se apodere del sistema.

Para garantizar que el IDS no ralentice el rendimiento de la red, estas soluciones suelen utilizar un analizador de puerto conmutado (SPAN) o un puerto de acceso de prueba (TAP) para analizar una copia del tráfico de datos integrado. Sin embargo, no bloquean las amenazas una vez que ingresan a la red, como lo

---

hacen los sistemas de prevención de intrusiones.

Independientemente de si hemos configurado un dispositivo físico o una aplicación IDS, el sistema puede:

1. Reconocer patrones de ataque en paquetes de red.
2. Realice un seguimiento del comportamiento del usuario.
3. Determinar la actividad de tráfico anormal.
4. Asegúrese de que la actividad del usuario y del sistema no entre en conflicto con las políticas de seguridad.

La información de un sistema de detección de intrusos también puede ayudar al equipo de seguridad a:

1. Verifique la red en busca de vulnerabilidades y configuraciones incorrectas.
2. Evaluar la integridad de sistemas y archivos críticos.
3. Crear medios más eficaces de control y respuesta a las incidencias.
4. Analizar la cantidad y tipos de ciberamenazas que atacan la red.

## 4.2 Estructura y arquitectura del sistema general de detección de intrusos.

Un sistema de detección de intrusos consta de cuatro partes principales. En la Fig. 5.2.1 se muestran como el repositorio de incidentes, analizadores, unidad de respuesta y sensores.

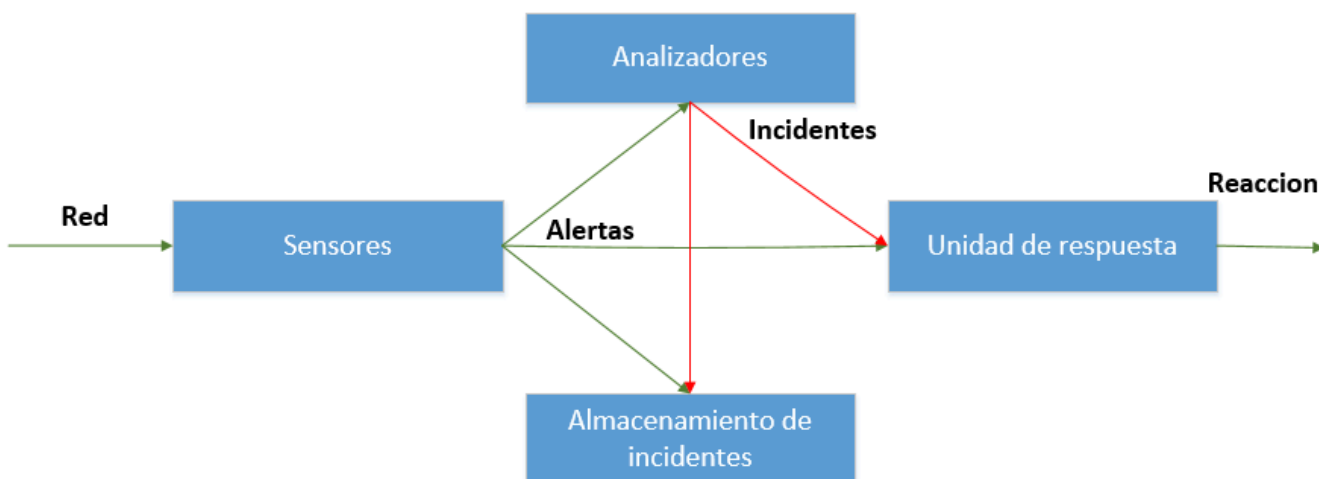


Figura 5.2.1 - La estructura del sistema de detección de intrusos

- 
- **Sensores:** determinar y enviar datos al sistema.
  - **Analizadores o Sistema Central de Monitoreo:** procesa y analiza datos enviados desde sensores.
  - **Componentes de base de datos y almacenamiento:** realice análisis de tendencias y almacene la dirección IP y la información del atacante.
  - **Campo de respuesta:** ingresa información de los componentes enumerados anteriormente y forma una respuesta determinada.

Los sistemas de detección de intrusiones siempre tienen su elemento principal: un sensor (sistema de análisis), que se encarga de detectar intrusiones. Los sensores reciben datos sin procesar de dos fuentes principales de información:

- propia base de conocimientos de IDS;
- registro del sistema;
- pistas de auditoría.

El registro del sistema puede incluir, por ejemplo, configuración del sistema de archivos, autorizaciones de usuario, etc. Esta información constituye la base para el posterior proceso de toma de decisiones. El sensor está integrado con el componente responsable de la recopilación de datos: el generador de eventos. El método de recolección está determinado.

La política del generador de eventos, que define el modo de filtrar la información sobre el evento.

Un generador de eventos (sistema operativo, red, aplicación) genera un conjunto de eventos coherente con las políticas que puede ser un registro (o auditoría) de eventos del sistema o paquetes de red.

El papel del sensor, como podemos ver en la fig. 5.2.2 es filtrar la información y descartar cualquier dato inapropiado obtenido del conjunto de eventos asociados al sistema protegido, detectando así actividades sospechosas. Para hacer esto, el analizador utiliza la base de datos de políticas de descubrimiento.

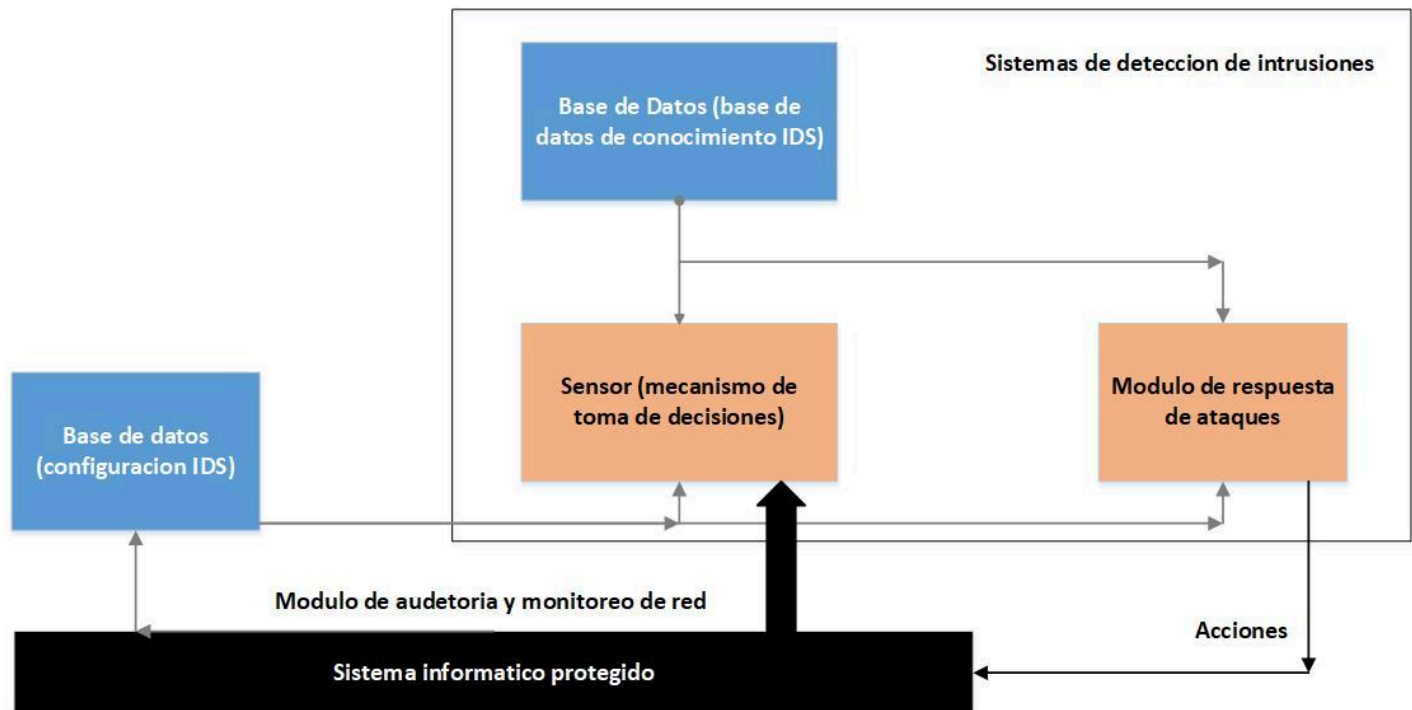
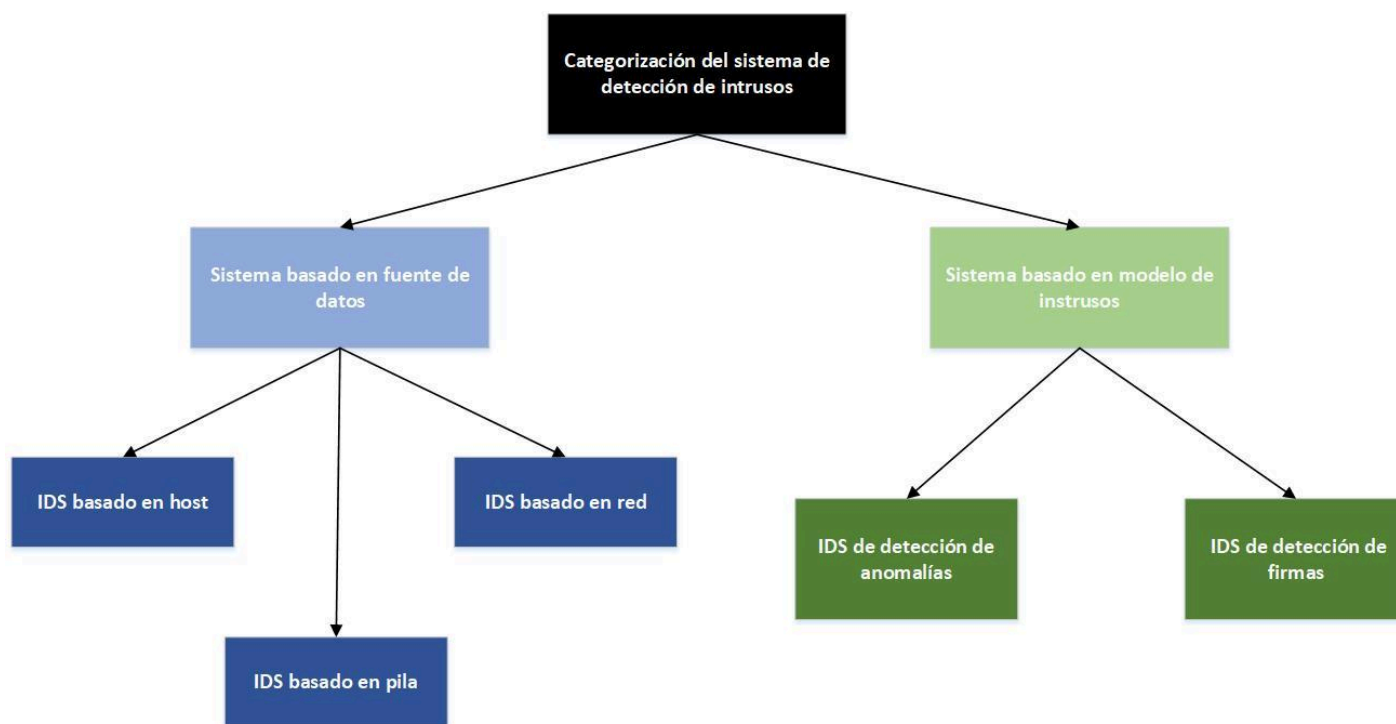


Figura 5.2.2 - Arquitectura del sistema de detección de ataques

Además, la base de datos contiene parámetros de configuración de IDS, incluidos modos de comunicación con el módulo de respuesta. El sensor también tiene su propia base de datos que contiene el historial dinámico del complejo potencial.

### 4.3 Categorización de los sistemas de detección de ataques

Dependiendo de la ubicación, el IDS funciona de manera diferente y se puede dividir en IDS de red (**NIDS**), IDS basado en host (**HIDS**) e IDS basado en pila (**SIDS**).



### 5.3.1 - Sistema de clasificación de ataques

**SIDS** - es una tecnología más nueva que funciona integrándose estrechamente con la pila TCP/IP, lo que permite observar los paquetes a medida que ascienden por las capas OSI. Al observar un paquete de esta manera, el IDS saca el paquete de la pila antes de que el sistema operativo o la aplicación tenga la oportunidad de procesar los paquetes.

**NIDS** escanea grandes cantidades de actividad de red a nivel de enrutador y señala transmisiones sospechosas como suplantación de IP, ataques DoS, envenenamiento de caché ARP y corrupción de nombres DNS.

**HIDS** monitorea múltiples archivos de registro en el host (kernel, sistema, red, firewall) para detectar uso indebido o intrusión. Además, HIDS garantiza la integridad de los datos críticos en el host al verificar las sumas de verificación de los archivos (md5 o sha1). Si las sumas de verificación no coinciden, HIDS notifica al administrador.



## 4.4 Propósito de HIDS y NIDS

Tanto **HIDS** como **NIDS** capturan el tráfico de red y comparan la información recopilada con patrones predefinidos para detectar ataques y vulnerabilidades (Figura 5.4.1).

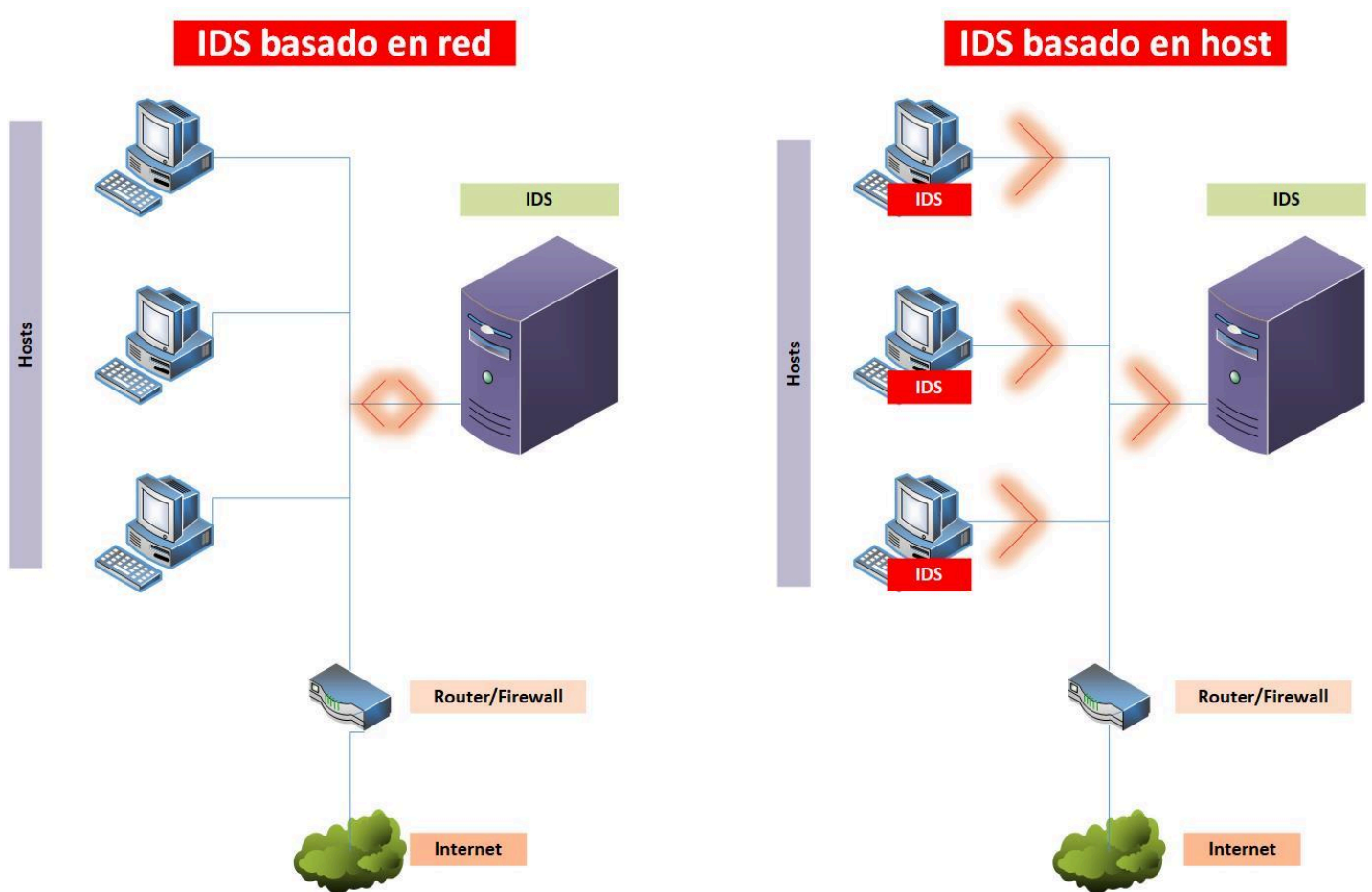


Figura 5.4.1 – Cómo funcionan NIDS y HIDS

**NIDS** (Sistema de detección de intrusiones basado en red) es un sistema de detección de intrusiones basado en red diseñado para detectar y monitorear actividades no deseadas o maliciosas en el tráfico de la red.

La función principal de NIDS es monitorear la red en busca de paquetes de datos anómalos o potencialmente maliciosos que puedan indicar ataques o intrusiones en la red.

---

**HIDS** (Sistema de detección de intrusiones basado en host) es un sistema de detección de intrusiones basado en host diseñado para detectar y detectar actividades no deseadas o maliciosas en una sola computadora o servidor.

Las funciones principales de HIDS incluyen analizar registros del sistema, monitorear la actividad del sistema de archivos, detectar cambios en la configuración del sistema y analizar el tráfico de red que pasa a través de un host en particular.

## 4.5 Comparación de NIDS y HIDS

Al comparar estos dos sistemas, he resaltado sus principales diferencias y aquí puedes ver sus ventajas y desventajas. A menudo se utilizan en combinación para proporcionar una protección completa del sistema y de la red.

### Ubicación:

**NIDS:** Ubicado en conmutadores de red, enrutadores o firewalls para monitorear el tráfico que pasa a través de la red.

**HIDS:** Ubicado en un servidor o servidor independiente para monitorear y proteger ese dispositivo específico.

### Objeto de análisis:

**NIDS:** analiza el tráfico de red que pasa a través de los puntos finales de la red para detectar actividades inusuales o sospechosas.

**HIDS:** analiza la actividad en un servidor específico, incluidos registros del sistema, archivos, registro y otros componentes del sistema.

### Detección de amenazas:

**NIDS:** Detecta amenazas de intrusión en la red como intrusiones, ataques DoS, virus, etc.

**HIDS:** Detecta amenazas originadas en el propio servidor, como intentos de intrusión, modificaciones de archivos, procesos sospechosos y más.

---

### **Detalles del análisis:**

**NIDS:** analiza el tráfico a nivel de paquetes de red para detectar patrones inusuales y actividades sospechosas.

**HIDS:** analiza la actividad a nivel de host para detectar amenazas más específicas, como ataques al sistema de archivos, intentos de infiltrarse en los procesos del sistema y más.

### **Escalada:**

**NIDS:** normalmente se escala para monitorizar toda la red, lo que puede ser difícil de administrar con mucho tráfico.

**HIDS:** se puede ampliar para monitorear hosts o servidores individuales, lo que permite un monitoreo y análisis de actividad más precisos.

### **Precisión de detección:**

**NIDS:** puede ser menos preciso debido al alto tráfico de red y la posible pérdida de información confidencial debido al cifrado.

**HIDS:** Suele ser más preciso porque analiza la actividad en el propio servidor, donde pueden estar disponibles datos más detallados.

## **4.6 Sistema de detección de intrusos perfecto**

Un sistema de detección de intrusos ideal debería resolver los siguientes problemas, independientemente del mecanismo en el que se base:

1. Debe ser muy difícil de engañar.
  2. Los costes del sistema deben mantenerse al mínimo. Un sistema que ralentiza tu ordenador no sirve de nada.
  3. El sistema debe observar desviaciones del comportamiento normal.
  4. El sistema debe funcionar continuamente sin supervisión humana.
- Debe ser lo suficientemente robusto como para poder operar en segundo plano del sistema monitoreado.

---

5.Revisar los cambios en el rendimiento del sistema cuando se agregan nuevos programas. Los perfiles del sistema cambiarán con el tiempo.

6. No tiene por qué ser una caja negra. Esto significa que el funcionamiento interno del sistema debe comprobarse desde el exterior.

7. Debe oponerse a las actividades subversivas. El sistema debe monitorearse a sí mismo para asegurarse de que no esté comprometido.

8. Cada sistema tiene sus propios patrones de uso y los mecanismos de seguridad deben adaptarse fácilmente a estos patrones.

## 4.7 Comparación de firewall e IDS.

Primero, se define qué son Firewall e IDS:

Un **firewall** es un sistema de seguridad de red que monitorea y regula el tráfico de la red de acuerdo con reglas específicas establecidas por el administrador de la red. El objetivo principal de un firewall es bloquear o permitir el tráfico de red según reglas predefinidas. Puede filtrar el tráfico por dirección IP, puerto, protocolo y otros parámetros. Los firewalls se pueden implementar como software en enrutadores, servidores o computadoras, o como dispositivos de red independientes.

Un **IDS** es un sistema de seguridad de red que se especializa en detectar actividades de red anómalas o sospechosas. El objetivo principal de IDS es monitorear el tráfico de la red e identificar amenazas potenciales como intrusiones, hackeos, virus y otros ataques. IDS puede detectar anomalías analizando cambios en el tráfico de la red, el comportamiento del usuario y encontrar firmas de amenazas o ataques conocidos. Entonces, aunque ambas herramientas se utilizan para proteger la red, tienen funciones y propósitos

diferentes. Un firewall filtra el tráfico según reglas de acceso, mientras que un IDS detecta actividad sospechosa que el firewall puede rechazar. Ambas herramientas se utilizan a menudo en combinación para lograr la máxima protección de la red.

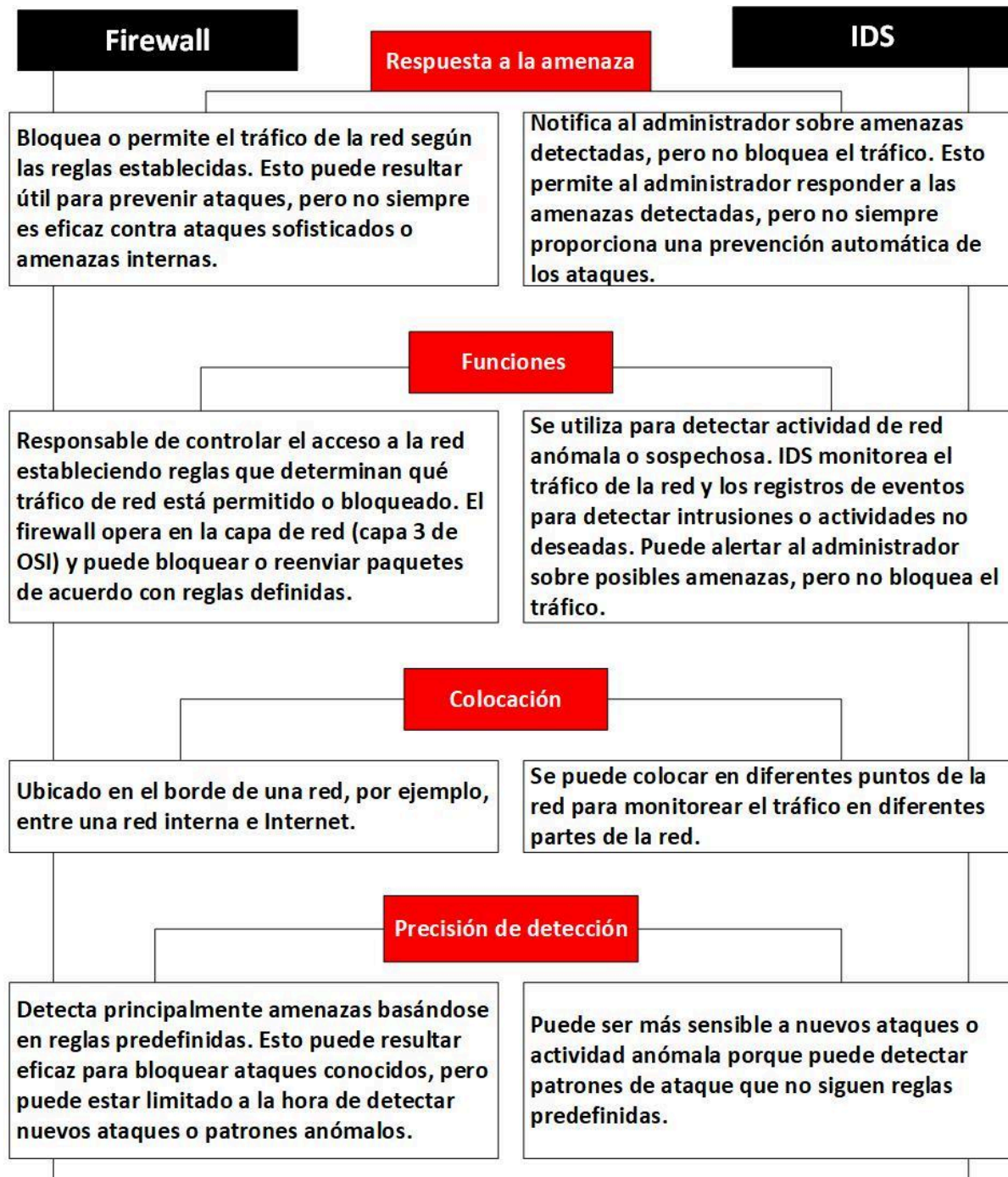




Figura 5.7.1

## 4.8 Comparación IDS e IPS

IPS es una herramienta de seguridad de red (puede ser hardware o software) que monitorea continuamente la red para detectar actividad maliciosa y toma medidas preventivas, incluidos informes, bloqueo o eliminación si ocurre.

IPS normalmente registra información relacionada con eventos observados, notifica a los administradores de seguridad sobre eventos observados importantes y genera informes.

Muchos IPS también pueden responder a una amenaza detectada para contenerla con éxito.

El sistema utiliza una variedad de métodos de respuesta que incluyen bloquear el ataque IPS, cambiar el entorno de seguridad o cambiar el contenido del ataque.

IPS se considera un complemento de los sistemas de detección de intrusos (IDS) porque tanto IPS como IDS monitorean el tráfico de la red y la actividad del sistema para detectar actividad maliciosa.

La principal diferencia entre ellos es que IDS es un sistema de seguimiento mientras que IPS es un sistema de gestión.

IDS no modifica los paquetes de red de ninguna manera, mientras que IPS impide la entrega de paquetes en función de su contenido, al igual que un firewall bloquea el tráfico en función de su dirección IP.

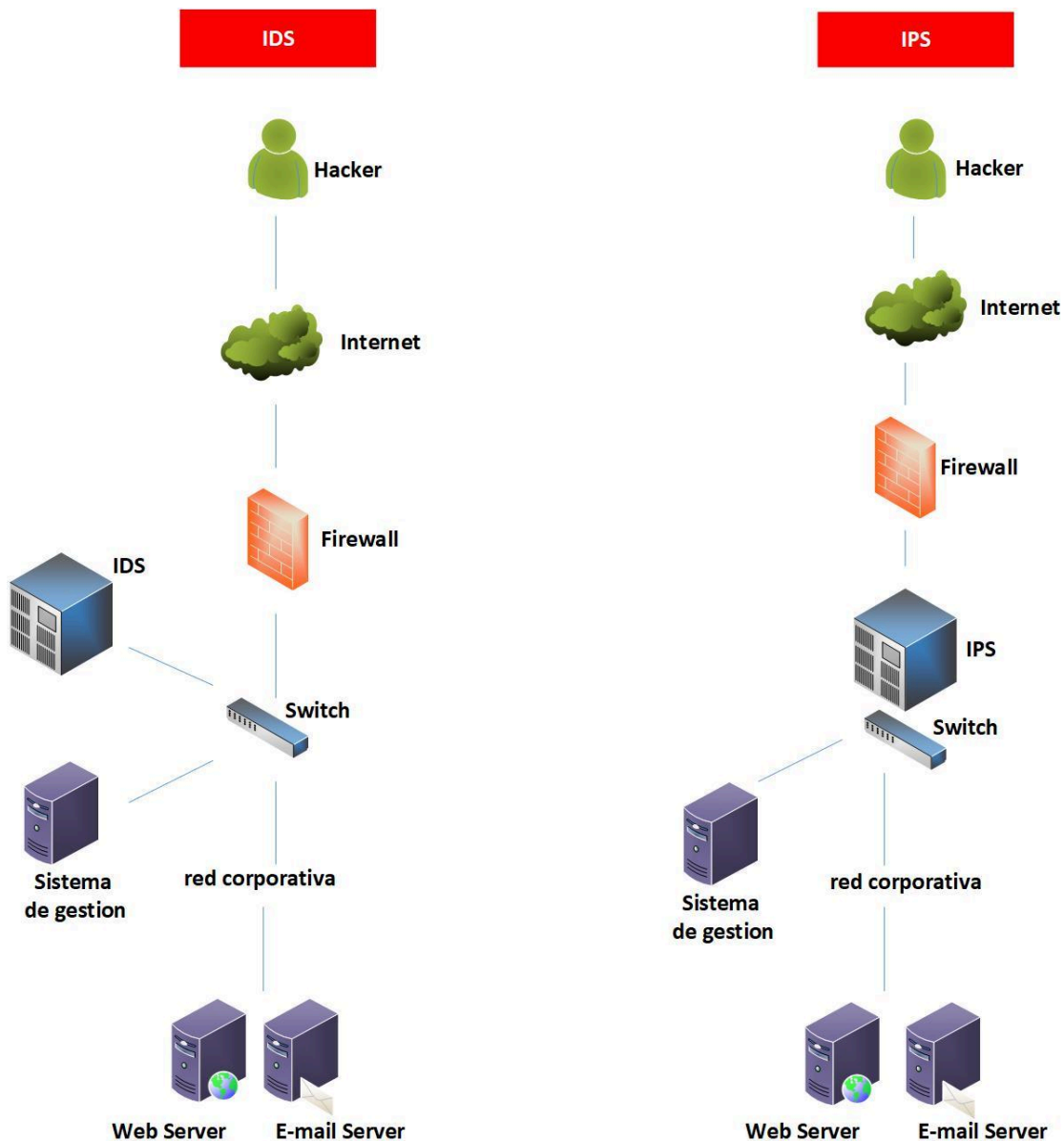


Figura 5.8.1

El IDS debe colocarse después del firewall, mientras que el IPS debe colocarse después del dispositivo firewall en las redes.



---

Las herramientas IDS están diseñadas para detectar actividad maliciosa, registrarla y enviar alertas. Lo que no hace es no sabe evitar el ataque. Los avisos que emiten siempre requieren de la intervención humana o de un sistema de seguridad adicional.

IPS responde basándose en criterios de tipo de ataque predefinidos bloqueando el tráfico y eliminando procesos maliciosos. Las herramientas IPS generan más falsos positivos porque tienen menos poder de detección que el IDS.

## **5. Sistema de detección de ataques - Suricata**

### **5.1 Características del sistema de detección de ataques Suricatas**

Suricata es una herramienta productiva y multitarea diseñada que protege redes en tiempo real, así como para recopilar y almacenar información sobre cualquier señal entrante. El trabajo del detector de ataques se basa en el análisis de firmas y heurísticas, y su conveniencia se debe a la disponibilidad de acceso abierto a código fuente. Este enfoque le permite configurar los parámetros del sistema para resolver tareas individuales.



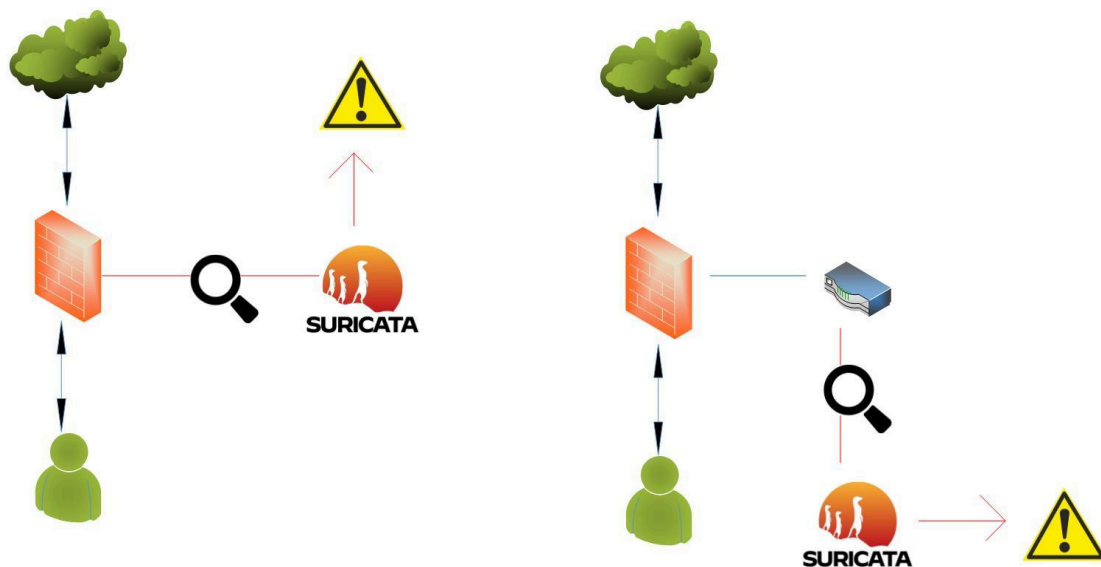


Figura 6.1.1

Suricata puede operar en modo de monitoreo de seguridad de la red y también puede configurarse como un sistema de prevención de intrusiones (IPS) o un sistema de detección de intrusiones (IDS), como se muestra en la Figura 6.1.1. El proyecto Suricata es un proyecto de código abierto y se diferencia de alternativas como Snort, Zeek o Segan en que admite subprocessos múltiples, HTTP/TLS y otras funciones útiles.

Antes de implementar Suricata, debe configurar las siguientes variables: qué interfaces de red usar, qué rangos de direcciones IP se identificarán como redes internas y qué rangos de direcciones IP se identificarán como redes externas. Home\_Net: Esta variable se utiliza para indicar la red doméstica que necesita ser protegida.

Outside\_Net se utiliza para identificar la red exterior.

Interfaz del paquete Af: la variable de interfaz en el paquete af se usa para especificar la interfaz de red que Suricata debe usar para el monitoreo.

---

El sistema está diseñado inicialmente para subprocesos múltiples, mientras que Snort es un producto de un solo subproceso. Debido a su larga historia y código heredado, no utiliza de manera óptima plataformas de hardware multiprocesador/multinúcleo, mientras que Suricata puede manejar tráfico de hasta 10 Gbps en computadoras normales de uso general. Se puede hablar durante mucho tiempo sobre las similitudes y diferencias entre los dos sistemas, pero aunque el motor Suricata funciona más rápido, para canales no demasiado anchos esto no es de fundamental importancia.

## 6. Creación de la máquina virtual Ubuntu 22.04

### 6.1 Requisitos del sistema

ISO de Ubuntu lo podemos descargar [aquí](#). El link es para instalación de Ubuntu 22.04 pero también podremos encontrar diferentes versiones. Actualmente, a la fecha de realización de este post, lo recomendable será instalar la versión 22.04 LTS. Se descargará un archivo ISO de 4.7GB el cual posteriormente uso en la instalación.


	<a href="#">ubuntu-22.04.4-desktop-amd64.iso</a>	2024-02-20 19:39	4.7G	Desktop image for 64-bit PC (AMD64) computers (standard download)
---	--	------------------	------	---

Figura 7.1.1

---

Entrando al Virtual Box, comenzó a crear una nueva máquina virtual Ubuntu 22.04 con todas las sutilezas que te indicaré a continuación.

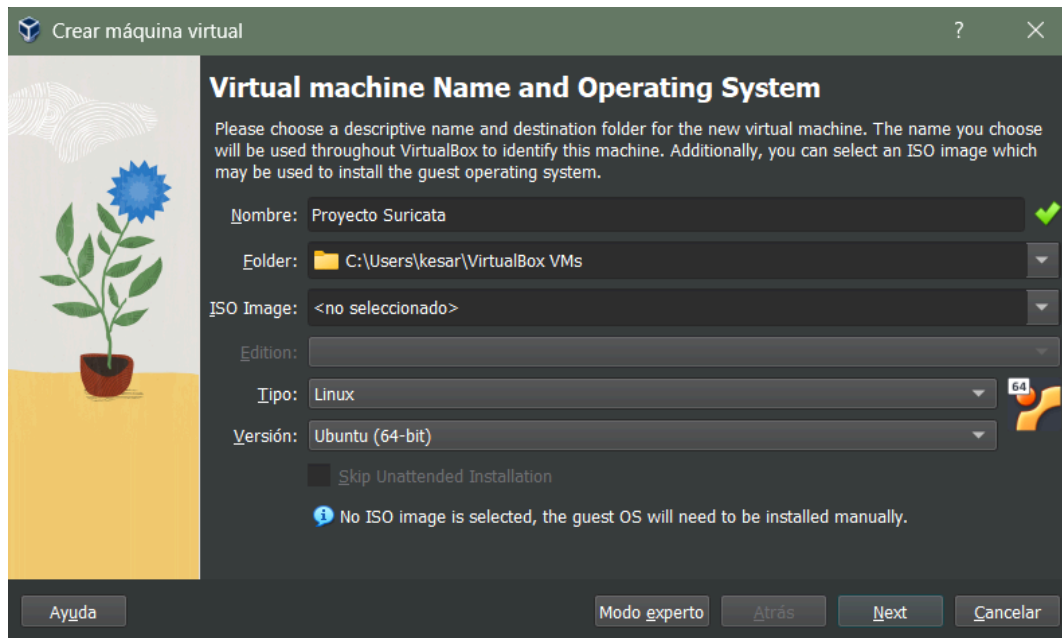


Figura 7.1.2

Eligiendo la cantidad de memoria que necesitamos, pongo 4 GB de RAM y 1 procesador CPU para un mejor rendimiento productividad, esto será suficiente para completar un proyecto de este tamaño

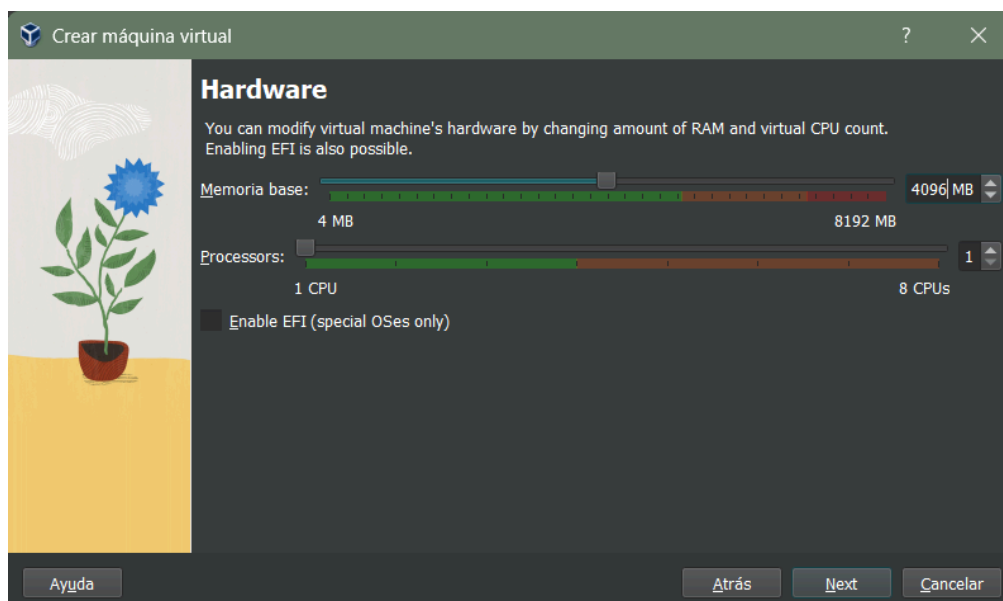


Figura 7.1.3

Siguiente paso es que necesito especificar el tamaño del disco.

Ubuntu recomienda al menos 25 GB, pero esto dependerá de cómo planeas usarlo y de los archivos que planeas almacenar en tu máquina virtual.

En nuestro caso, elegimos 50 GB:

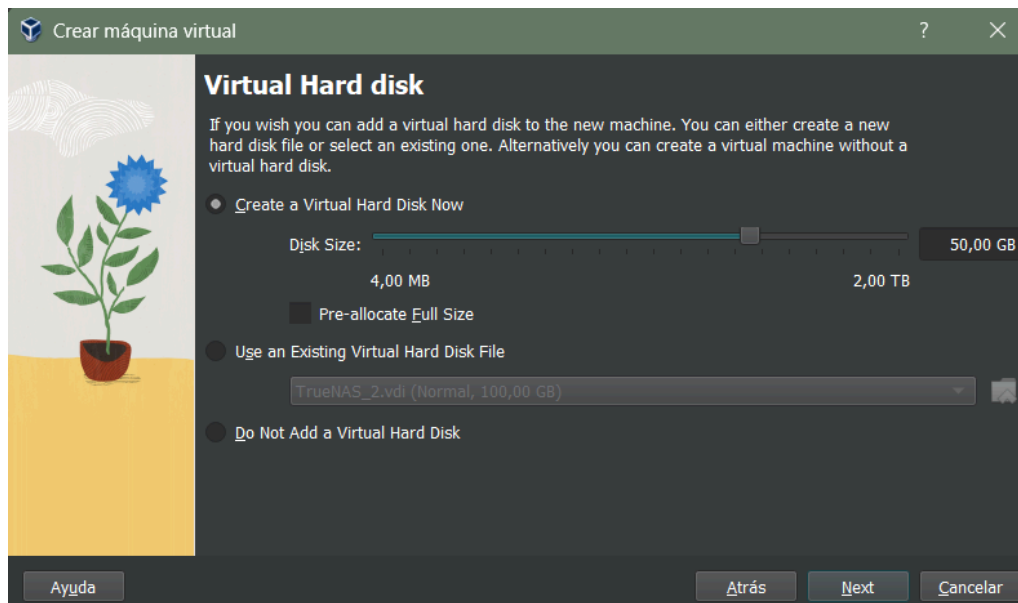


Figura 7.1.4

Esta es la información que tenemos sobre nuestra máquina virtual.

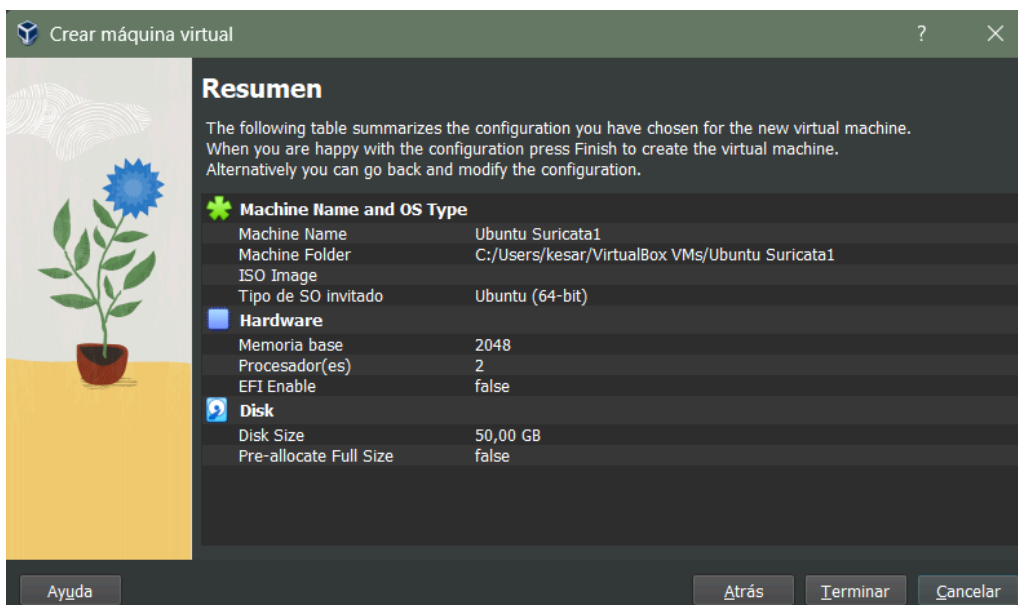


Figura 7.1.5

En este punto ya hemos creado nuestra máquina virtual con los parámetros elegidos en los pasos anteriores. El resto de elementos se configurarán durante la instalación del sistema operativo. Este proceso se puede iniciar a través del menú “Configuración”.

Aparecerá el siguiente menú. Hago clic en "Almacenamiento". Una vez aquí, hago clic en el espacio en blanco, justo debajo de "Controlador: IDE".

Esta sección aparecerá a la derecha.

Aquí haremos clic en el disco azul de la derecha y haremos clic en “Elegir archivos en el disco”.

Encontraremos el archivo de Ubuntu que descargamos antes y haremos clic en “Aceptar”.

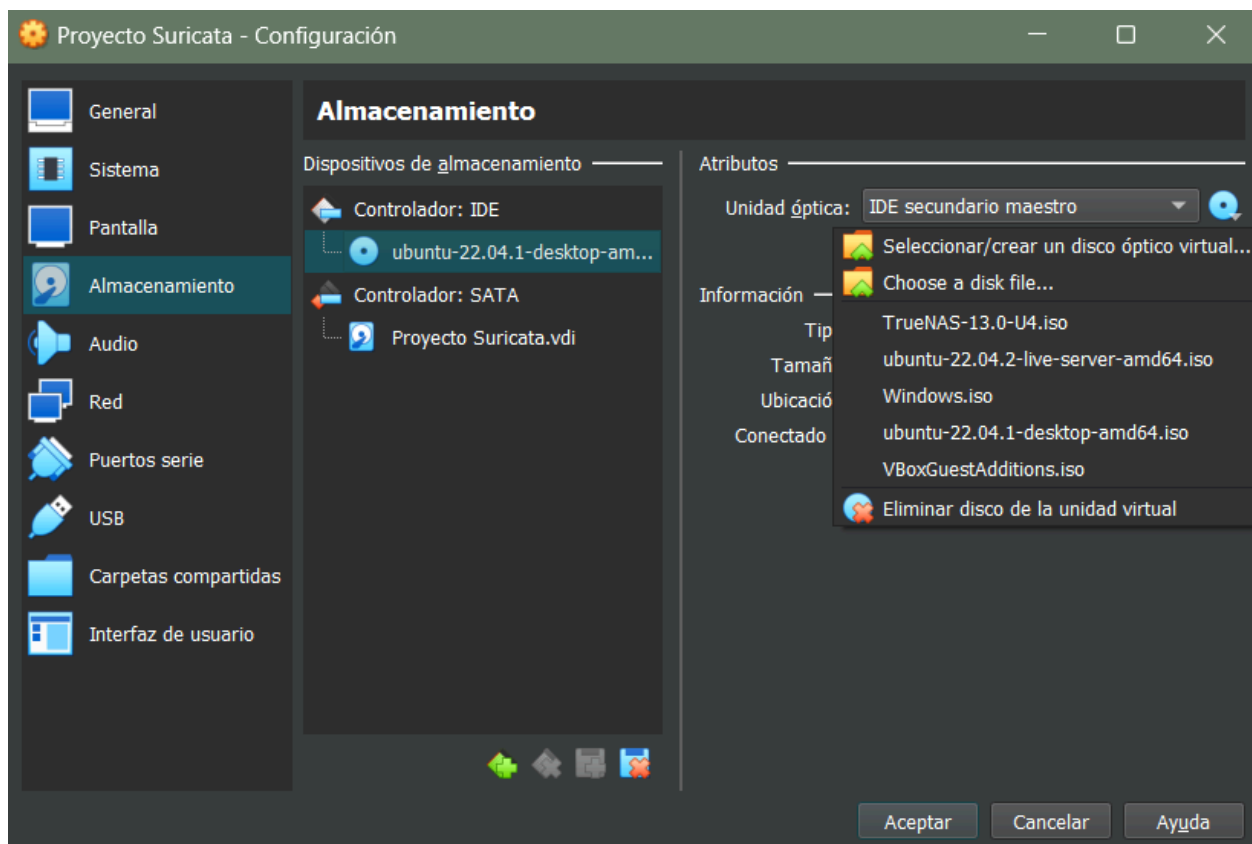


Figura 7.1.6

En este punto ya he creado mi máquina virtual de Ubuntu 22.04 con los parámetros elegidos en los pasos anteriores. En la pantalla de inicio podemos ver cómo son todas las funciones que seleccionamos al crear el coche, aunque aún quedan algunas cosas por configurar. El resto de elementos se configurarán durante la instalación del sistema operativo.

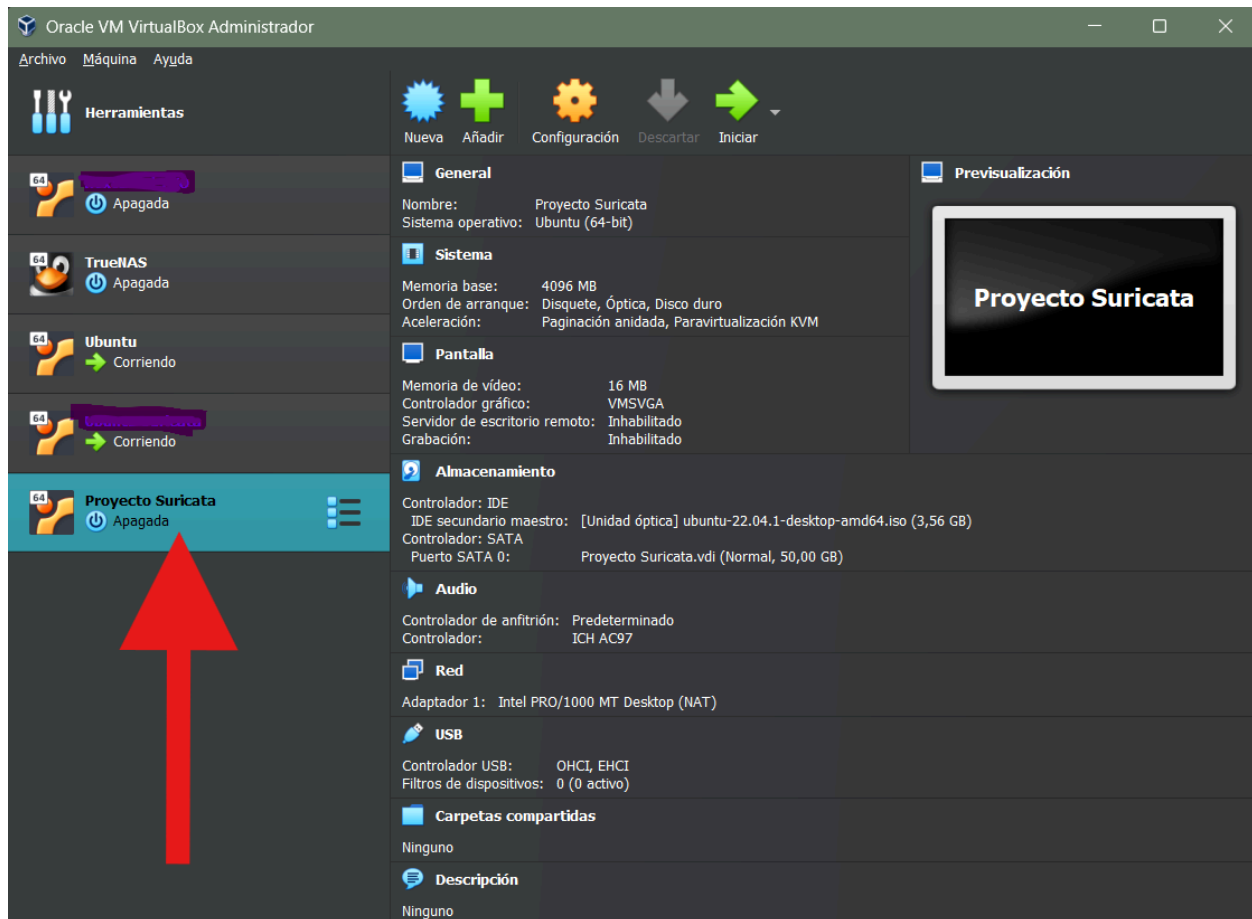


Figura 7.1.7

## 6.2 Instalacion de Ubuntu

Podría probar Ubuntu, no se eliminará ningún archivo del disco duro. Pero no es mi caso así que seleccione el idioma del instalador y seleccione "Instalar Ubuntu" en el siguiente paso.



Figura 7.2.1

Ahora elegí entre instalación normal e instalación mínima.

Mejor hacer instalación normal ya que ofrece más herramientas y paquetes de software.

Además, seleccione la opción de descarga para descargar las actualizaciones para obtener la última versión posible.

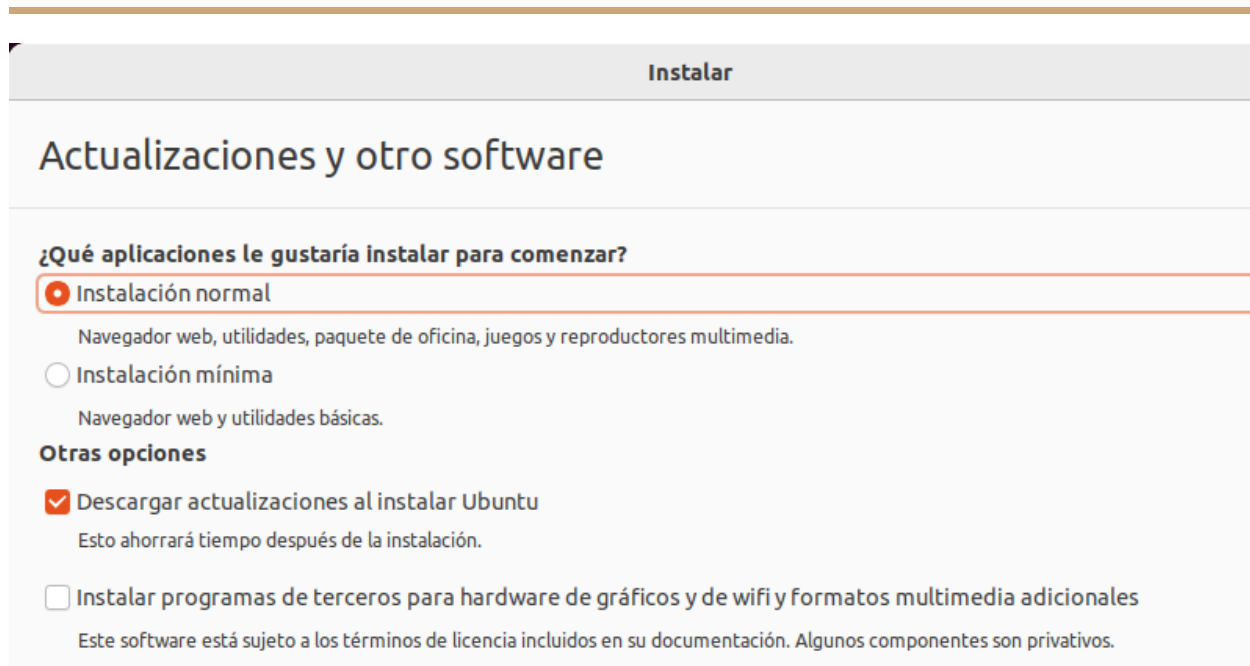


Figura 7.2.2

Para el tipo de instalación, elige “Borrar disco e instalar Ubuntu”.

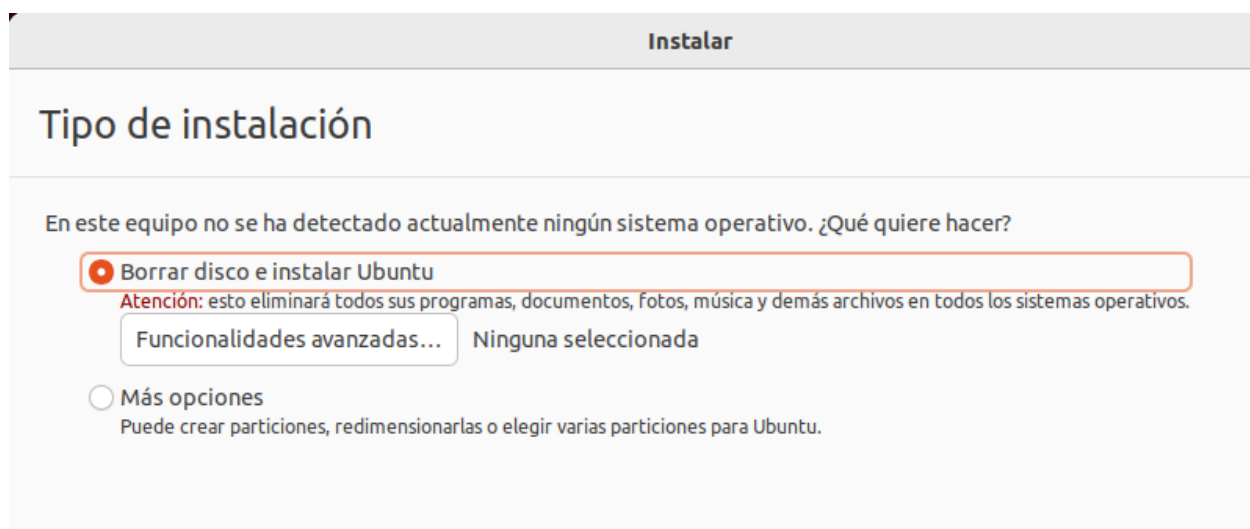


Figura 7.2.3



Haz clic en Continuar. cuando se te pregunte por la partición.

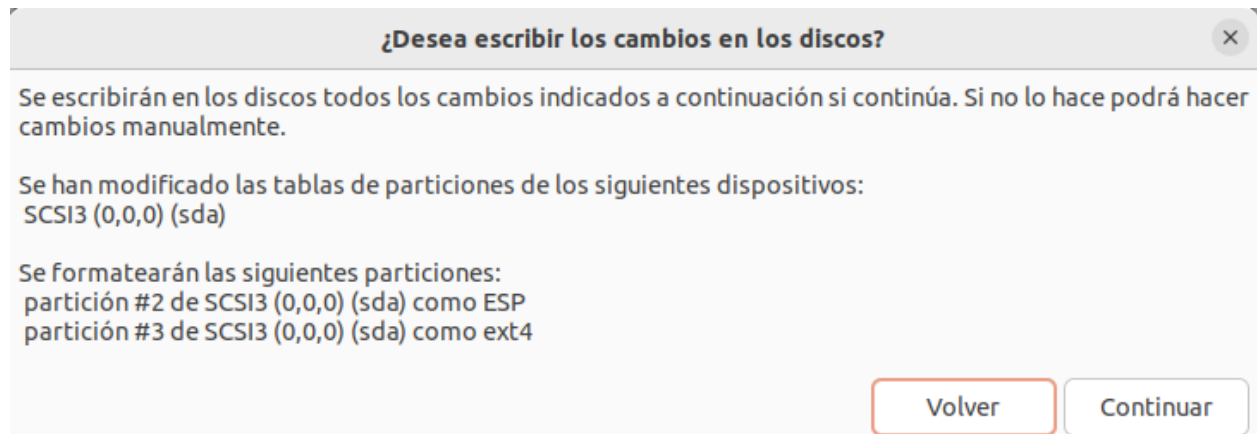


Figura 7.2.4

Y por último paso, rellena el nombre, el nombre de usuario y la contraseña.

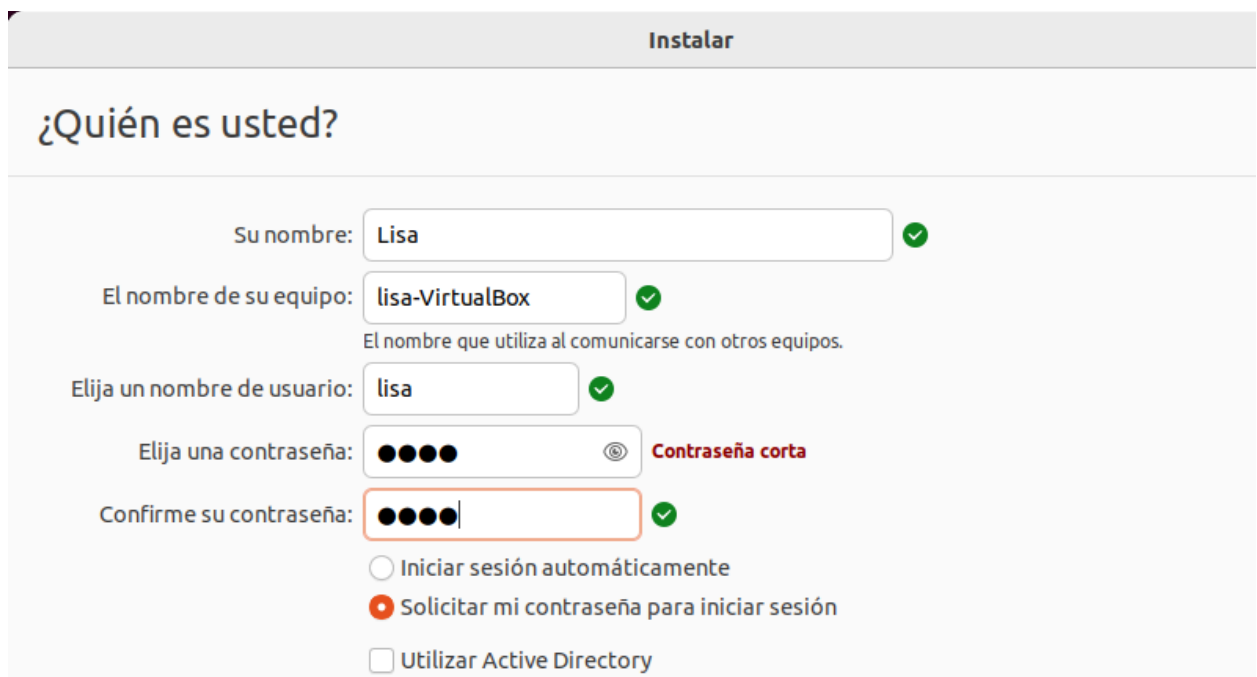


Figura 7.2.5

Después de hacer clic en continuar, se iniciará la instalación de Ubuntu.  
Esperanza tarda más o menos unos 20 minutos.

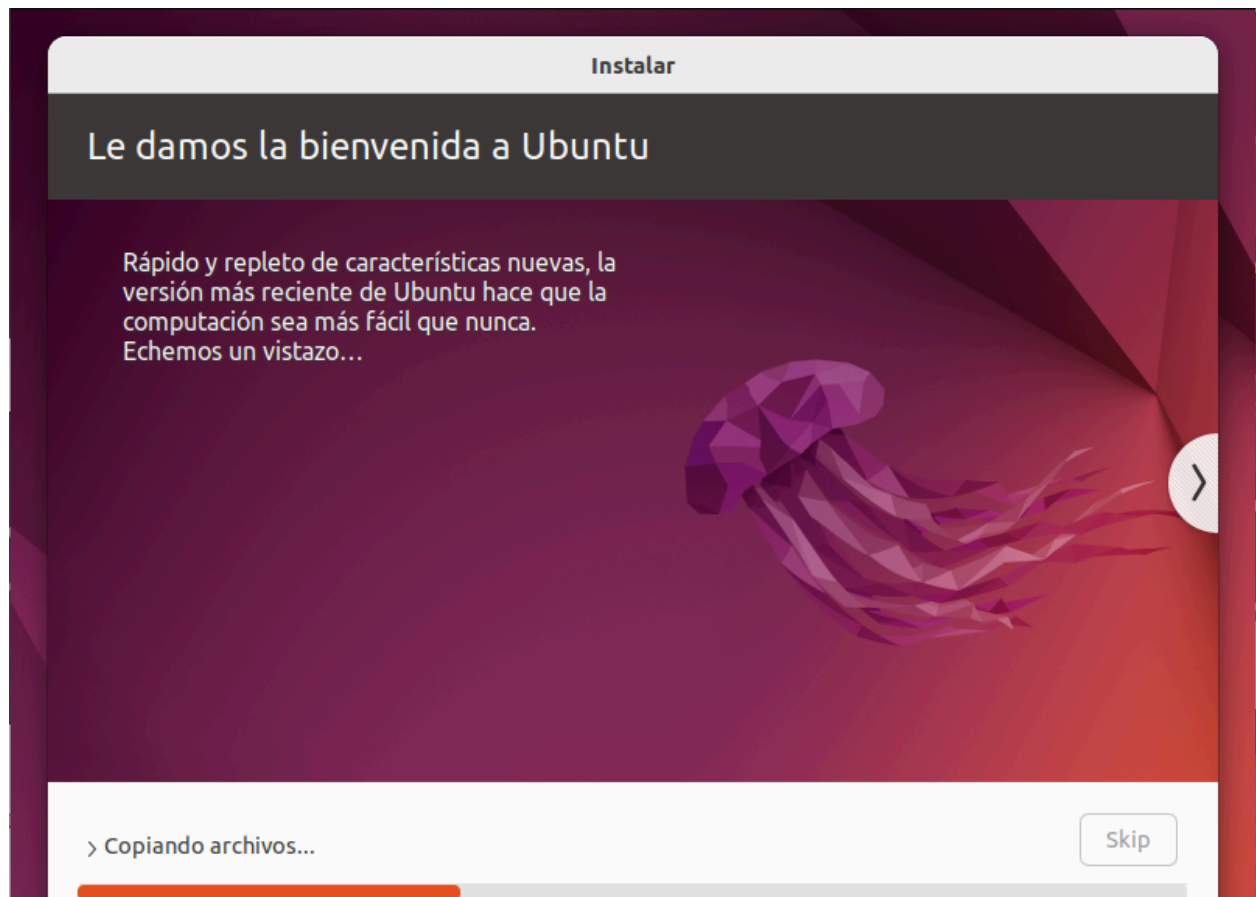


Figura 7.2.6

Pulse a “Reiniciar ahora” ordenador una vez terminada la instalación. Una vez que el ordenador arranque, deberías ver la pantalla de bienvenida por defecto

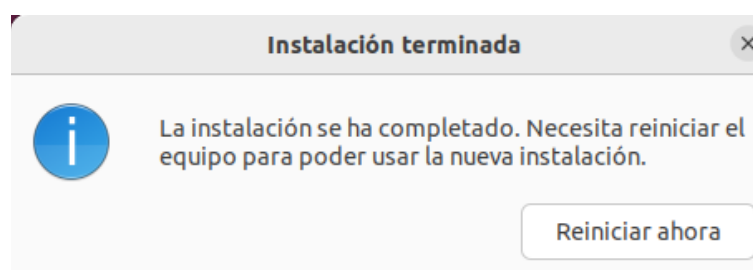


Figura 7.2.7

---

## 7. Instalación de Suricata

### 7.1 Instalación de SSH

La opción más segura es conectarse al servidor a través de SSH como usuario estándar y luego usar la utilidad sudo para escalar privilegios. Siguiendo los pasos para instalar SSH:

```
$ sudo apt update
```

```
$ sudo apt install openssh-server
```

Una vez finalizada la instalación de los paquetes, ejecute el siguiente comando para comprobar el estado del servicio SSH:

```
$ sudo systemctl status ssh
```

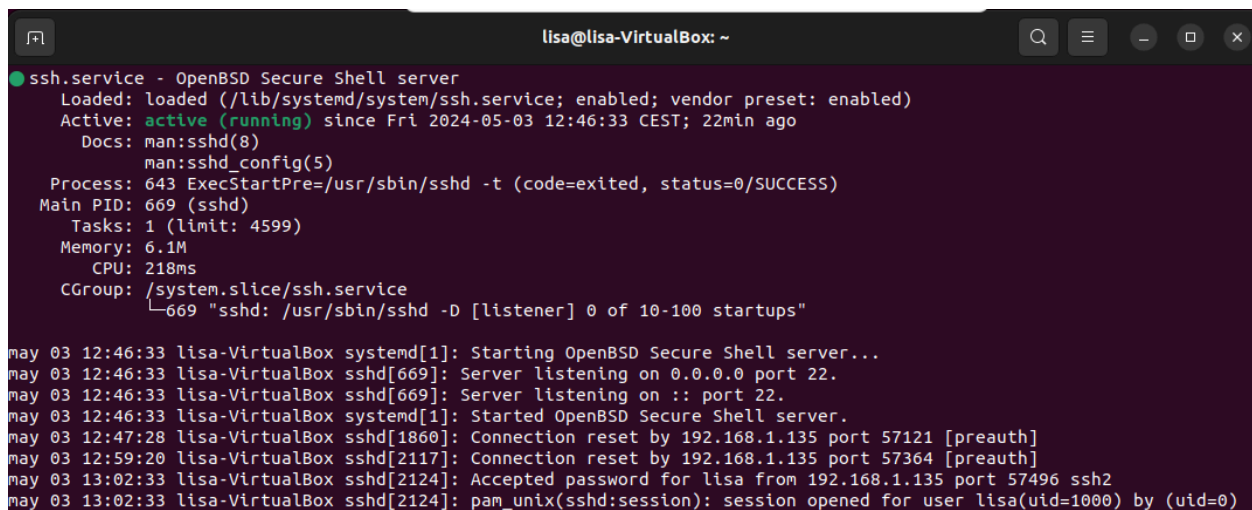
A terminal window titled 'lisa@lisa-VirtualBox: ~' displays the output of the command 'sudo systemctl status ssh'. The output shows that the 'ssh.service' is an 'OpenBSD Secure Shell server' which is 'loaded' and 'enabled'. It is currently 'active (running)' since 'Fri 2024-05-03 12:46:33 CEST; 22min ago'. The process is '643 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)'. The main PID is '669 (sshd)'. The terminal also shows a series of log messages from 'systemd' and 'sshd' indicating the server's startup, listening on port 22, and handling connection resets and a successful login for the user 'lisa'.

Figura 8.1.1

---

Pero ocurrió un error en el camino de conectarse al SSH,

```
PS C:\Users\kesar> ssh lisa@192.168.1.140
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!    @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ED25519 key sent by the remote host is
SHA256:g2gyfzz9VTLPtTjjdBf4ZG/aTmGA4VyCY4RcZ6CH3AFE.
Please contact your system administrator.
Add correct host key in C:\\Users\\kesar\\.ssh\\known_hosts to get rid of this message.
Offending ECDSA key in C:\\Users\\kesar\\.ssh\\known_hosts:42
Host key for 192.168.1.140 has changed and you have requested strict checking.
Host key verification failed.
```

Figura 8.1.2

---

Buscando varias soluciones se encuentre una que ha resuelto el error de conexión, tiene que poner este comando:

```
ssh-keygen -R <IP SERVIDOR AL QUE SE PRETENDE CONECTAR>
```

En mi caso es:

```
ssh-keygen -R 192.168.1.140
```

```
PS C:\Users\kesar> ssh-keygen -R 192.168.1.140
# Host 192.168.1.140 found: line 40
# Host 192.168.1.140 found: line 41
# Host 192.168.1.140 found: line 42
```

Figura 8.1.3

---

Teniendo SSH bien conectado, ahora tengo todo lo que necesito para empezar a instalar Suricata y los paquetes necesarios.

```
PS C:\Users\kesar> ssh lisa@192.168.1.140
lisa@192.168.1.140's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-28-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

El mantenimiento de seguridad expandido para Applications está desactivado
Se pueden aplicar 0 actualizaciones de forma inmediata.

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Last login: Fri May  3 13:02:33 2024 from 192.168.1.135
lisa@lisa-VirtualBox:~$ |
```

Figura 8.1.4

## 7.2 Instalación de Suricata y los paquetes necesarios

Según la asociación "[digitales.es](https://digitales.es)", el volumen de tráfico de la red aumenta aproximadamente un 23% cada año. Esto conduce a un aumento de la carga sobre el equipo y, en particular, aumenta los requisitos de rendimiento del IDS/IPS. Puede comprar hardware especializado costoso, pero existe una opción más económica: implementar uno de los sistemas de código abierto. Muchos administradores novatos piensan que instalar y configurar un IPS gratuito es bastante complicado. En el caso de Suricata, esto no es del todo cierto: puedes instalarlo y comenzar a repeler ataques estándar con un conjunto de reglas gratuitas en unos minutos.

Suricata se instala en una máquina virtual de Ubuntu 22.04 LTS. Todos los comandos deben ejecutarse como superusuario (root).

Primero necesitamos instalar los paquetes que necesitamos:

---

```
sudo apt -y install libpcrc3 libpcrc3-dev build-essential autoconf automake libtool  
libpcap-dev libnet1-dev libyaml-0-2 libyaml-dev zlib1g zlib1g-dev libmagic-dev  
libcap-ng-dev libjansson-dev pkg-config libnetfilter-queue-dev geoip-bin  
geoip-database geoipupdate apt-transport-https
```

```
lisa@lisa-VirtualBox:~$ sudo apt -y install libpcrc3 libpcrc3-dev build-essential autoconf automake libtool libpcap-dev  
libnet1-dev libyaml-0-2 libyaml-dev zlib1g zlib1g-dev libmagic-dev libcap-ng-dev libjansson-dev pkg-config libnetfilter-  
queue-dev geoip-bin geoip-database geoipupdate apt-transport-https  
[sudo] contraseña para lisa:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
libyaml-0-2 ya está en su versión más reciente (0.2.2-1build2).  
fijado libyaml-0-2 como instalado manualmente.  
libpcrc3 ya está en su versión más reciente (2:8.39-13ubuntu0.22.04.1).  
fijado libpcrc3 como instalado manualmente.  
zlib1g ya está en su versión más reciente (1:1.2.11.dfsg-2ubuntu9.2).  
fijado zlib1g como instalado manualmente.  
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.  
  libflashrom1 libftdi1-2 liblvm13  
Utilice «sudo apt autoremove» para eliminarlos.  
Se instalarán los siguientes paquetes adicionales:
```

Figura 8.2.1

---

Conectando un repositorio externo:

```
sudo add-apt-repository ppa:oisf/suricata-stable
```

```
lisa@lisa-VirtualBox:~$ sudo add-apt-repository ppa:oisf/suricata-stable  
[sudo] contraseña para lisa:  
Lo siento, pruebe otra vez.  
[sudo] contraseña para lisa:  
Repositorio: «deb https://ppa.launchpadcontent.net/oisf/suricata-stable/ubuntu/ jammy main»  
Descripción:  
Suricata IDS/IPS/NSM stable packages  
https://suricata.io/  
https://oisf.net/  
  
Suricata IDS/IPS/NSM - Suricata is a high performance Intrusion Detection and Prevention System and  
Network Security Monitoring engine.  
  
Open Source and owned by a community run non-profit foundation, the Open Information Security Found  
ation (OISF). Suricata is developed by the OISF, its supporting vendors and the community.
```

Figura 8.2.2

---

```
sudo apt-get update
```

```
lisa@lisa-VirtualBox:~$ sudo apt-get update
Obj:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Obj:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Obj:3 https://ppa.launchpadcontent.net/oisf/suricata-stable/ubuntu jammy InRelease
Obj:4 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:5 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Leyendo lista de paquetes... Hecho
lisa@lisa-VirtualBox:~$ |
```

Figura 8.2.3

---

Instale la última versión estable de Suricata:

```
sudo apt-get install suricata
```

```
lisa@lisa-VirtualBox:~$ sudo apt-get install suricata
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
 libflashrom1 libftdi1-2 libllvm13
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
 libevent-core-2.1-7 libevent-pthreads-2.1-7 libhiredis0.14 libhttp2
 libhyperscan5 liblua5.1-2 liblua5.1-common liblzma-dev
Paquetes sugeridos:
 liblzma-doc
Se instalarán los siguientes paquetes NUEVOS:
 libevent-core-2.1-7 libevent-pthreads-2.1-7 libhiredis0.14 libhttp2
 libhyperscan5 liblua5.1-2 liblua5.1-common liblzma-dev suricata
0 actualizados, 9 nuevos se instalarán, 0 para eliminar y 4 no actualizados.
Se necesita descargar 6.291 kB de archivos.
Se utilizarán 28,6 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

Figura 8.2.4

---

Después de instalar Suricata, hay que comprobar qué versión de Suricata se está ejecutando y con qué parámetros, así como el host del servicio.

```
sudo suricata --build -info
```

```
lisa@lisa-VirtualBox:~$ sudo suricata --build -info
This is Suricata version 7.0.5 RELEASE
```

Figura 8.2.5

---

Ahora que el paquete está instalado, habilite `suricata.service` para que se ejecute cuando se reinicie su sistema. Utilice este comando para habilitarlo:

```
sudo systemctl enable suricata.service
```

```
lisa@lisa-VirtualBox:~$ sudo systemctl enable suricata.service
[sudo] contraseña para lisa:
suricata.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable suricata
lisa@lisa-VirtualBox:~$
```

Figura 8.2.6

---

Con este comando podemos comprobar si Suricata está instalado bien y está activo ahora mismo:

```
sudo systemctl status suricata
```

```
lisa@lisa-VirtualBox:~$ sudo systemctl status suricata
● suricata.service - LSB: Next Generation IDS/IPS
   Loaded: loaded (/etc/init.d/suricata; generated)
   Active: active (exited) since Fri 2024-05-03 13:54:57 CEST; 1h 37min ago
     Docs: man:systemd-sysv-generator(8)
    CPU: 136ms

may 03 13:54:57 lisa-VirtualBox systemd[1]: Starting LSB: Next Generation IDS/IPS...
may 03 13:54:57 lisa-VirtualBox suricata[4256]: Starting suricata in IDS (af-packet)
may 03 13:54:57 lisa-VirtualBox systemd[1]: Started LSB: Next Generation IDS/IPS.
lines 1-9/9 (END)
lisa@lisa-VirtualBox:~$
```

Figura 8.2.6

---

No lo necesito, si el programa estuvo funcionando durante una hora de ajuste, entonces lo escribo con el comando:

```
sudo systemctl stop suricata
```

```
lisa@lisa-VirtualBox:~$ sudo systemctl stop suricata
[sudo] contraseña para lisa:
lisa@lisa-VirtualBox:~$
```

Figura 8.2.7



---

Mirando el contenido del directorio /etc/suricata/, puede ver el archivo configuración – suricata.yaml

`ls -al /etc/suricata/`

```
lisa@lisa-VirtualBox:~$ ls -al /etc/suricata/
total 112
drwxr-xr-x  2 root root  4096 may  3 13:54 .
drwxr-xr-x 131 root root 12288 may  3 13:54 ..
-rw-r--r--  1 root root  3327 abr 23 07:35 classification.config
-rw-r--r--  1 root root  1375 abr 23 07:35 reference.config
-rw-r--r--  1 root root 85224 abr 23 15:05 suricata.yaml
-rw-r--r--  1 root root  1643 abr 23 07:35 threshold.config
lisa@lisa-VirtualBox:~$
```

Figura 8.2.8

---

## 7.3 Configuración de Suricata

Definir la interfaz y la dirección IP en la que Suricata debe verificar la red paquetes:

`ifconfig`

```
lisa@lisa-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.1.137  netmask 255.255.255.0  broadcast 192.168.1.255
    inet6 fe80::76ff:1a7a:f3d3:7214  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:21:71:f1  txqueuelen 1000  (Ethernet)
    RX packets 86  bytes 16790 (16.7 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 174  bytes 23265 (23.2 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Bucle local)
    RX packets 137  bytes 11747 (11.7 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 137  bytes 11747 (11.7 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Figura 8.3.1

---

Podemos ver que mi interfaz es enp0s3 y la subred es 192.168.1.137

ip a s

```
lisa@lisa-VirtualBox:~$ ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:21:71:f1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.137/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
        valid_lft 86183sec preferred_lft 86183sec
    inet6 fe80::76ff:1a7a:f3d3:7214/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
lisa@lisa-VirtualBox:~$
```

Figura 8.3.2

---

Para realizar cambios en el archivo /etc/suricata/suricata.yaml, utilice el editor nano y los derechos de superusuario

sudo nano /etc/suricata/suricata.yaml

Y se cambian siguientes líneas:

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

Figura 8.3.3

```
# Linux high speed capture support
af-packet:
  - interface: enp0s3|
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
    cluster-id: 99
```

Figura 8.3.4

---

```
# Cross platform libpcap capture support
pcap:
- interface: enp0s3|
  # On Linux, pcap will try to use mmap'ed capture and will use "buffer-size"
  # as total memory used by the ring. So set this to something bigger
```

Figura 8.3.5

False -> True

```
# enable/disable the community id feature.
community-id: true|
# Seed value for the ID output. Valid values are 0-65535.
community-id-seed: 0
```

Figura 8.3.6

---

Para determinar el nombre del dispositivo para la interfaz de red predeterminada, puede usar el este comando:

```
ip -p -j route show default
```

```
lisa@lisa-VirtualBox:~$ ip -p -j route show default
[ {
  "dst": "default",
  "gateway": "192.168.1.1",
  "dev": "enp0s3",
  "protocol": "dhcp",
  "metric": 100,
  "flags": [ ]
} ]
lisa@lisa-VirtualBox:~$ |
```

Figura 8.3.7

---

Se debe ejecutar Suricata una vez para que los cambios surjan efecto.

```
sudo suricata-update
```

---

```
lisa@lisa-VirtualBox:~$ sudo suricata-update
[sudo] contraseña para lisa:
6/5/2024 -- 12:02:18 - <Info> -- Using data-directory /var/lib/suricata.
6/5/2024 -- 12:02:18 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
6/5/2024 -- 12:02:18 - <Info> -- Using /usr/share/suricata/rules for Suricata provided rules.
6/5/2024 -- 12:02:18 - <Info> -- Found Suricata version 7.0.5 at /usr/bin/suricata.
6/5/2024 -- 12:02:18 - <Info> -- Loading /etc/suricata/suricata.yaml
6/5/2024 -- 12:02:18 - <Info> -- Disabling rules for protocol pgsql
6/5/2024 -- 12:02:18 - <Info> -- Disabling rules for protocol modbus
6/5/2024 -- 12:02:18 - <Info> -- Disabling rules for protocol dnp3
```

Figura 8.3.8

```
6/5/2024 -- 12:02:27 - <Info> -- Testing with suricata -T.
6/5/2024 -- 12:02:56 - <Info> -- Done.
lisa@lisa-VirtualBox:~$ |
```

Figura 8.3.9

---

Cuando se complete la actualización, el programa ejecutará una prueba de configuración de Suricata para asegurarme de que no haya ningún problema con las opciones y la sintaxis que estoy usando.

`sudo ls -al /var/lib/suricata/rules`

```
lisa@lisa-VirtualBox:~$ sudo ls -al /var/lib/suricata/rules/
total 27896
drwxr-x--- 2 root root    4096 may  6 12:02 .
drwxr-xr-x 4 root root    4096 may  6 12:02 ..
-rw-r--r-- 1 root root   3228 may  6 12:02 classification.config
-rw-r--r-- 1 root root 28552065 may  6 12:02 suricata.rules
```

Figura 8.3.10

---

Suricata ofrece la posibilidad de especificar fuentes personalizadas, por lo que de forma predeterminada me permite elegir de qué fuentes me gustaría obtener reglas.

`sudo suricata-update list-sources`

```

lisa@lisa-VirtualBox:~$ sudo suricata-update list-sources
20/5/2024 -- 10:50:25 - <Info> -- Using data-directory /var/lib/suricata.
20/5/2024 -- 10:50:25 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
20/5/2024 -- 10:50:25 - <Info> -- Using /usr/share/suricata/rules for Suricata provided rules.
20/5/2024 -- 10:50:25 - <Info> -- Found Suricata version 7.0.5 at /usr/bin/suricata.
Name: et/open
Vendor: Proofpoint
Summary: Emerging Threats Open Ruleset
License: MIT
Name: et/pro

```

Figura 8.3.11

Para agregar o habilitar malsiro/win-malware.

`sudo suricata-update enable-source malsilo/win-malware`

```

lisa@lisa-VirtualBox:~$ sudo suricata-update enable-source malsilo/win-malware
20/5/2024 -- 10:56:25 - <Info> -- Using data-directory /var/lib/suricata.
20/5/2024 -- 10:56:25 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
20/5/2024 -- 10:56:25 - <Info> -- Using /usr/share/suricata/rules for Suricata provided rules.
20/5/2024 -- 10:56:25 - <Info> -- Found Suricata version 7.0.5 at /usr/bin/suricata.
20/5/2024 -- 10:56:25 - <Info> -- Creating directory /var/lib/suricata/update/sources
20/5/2024 -- 10:56:25 - <Info> -- Enabling default source et/open
20/5/2024 -- 10:56:25 - <Info> -- Source malsilo/win-malware enabled
lisa@lisa-VirtualBox:~$

```

Figura 8.3.12

`sudo suricata-update`

```

lisa@lisa-VirtualBox:~$ sudo suricata-update
20/5/2024 -- 11:01:40 - <Info> -- Using data-directory /var/lib/suricata.
20/5/2024 -- 11:01:40 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml

```

Figura 8.3.13

Quiero especificar mis propias reglas y el archivo de configuración real, pero primero necesito ejecutar Suricata.

`sudo systemctl start suricata.service`

`sudo systemctl status suricata.service`

```

lisa@lisa-VirtualBox:~$ sudo systemctl status suricata.service
● suricata.service - LSB: Next Generation IDS/IPS
   Loaded: loaded (/etc/init.d/suricata; generated)
   Active: active (running) since Mon 2024-05-06 13:21:21 CEST; 23s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 4487 ExecStart=/etc/init.d/suricata start (code=exited, status=0/SUCCESS)
    Tasks: 1 (limit: 4599)
   Memory: 338.9M
      CPU: 23.766s
    CGroup: /system.slice/suricata.service
            └─4496 /usr/bin/suricata -c /etc/suricata/suricata.yaml --pidfile /var/run/suricata.pid --af-packet -D -vvv

may 06 13:21:21 lisa-VirtualBox systemd[1]: Starting LSB: Next Generation IDS/IPS...
may 06 13:21:21 lisa-VirtualBox suricata[4487]: Likely stale PID 872 with /var/run/suricata.pid exists, but process is not running!
may 06 13:21:21 lisa-VirtualBox suricata[4487]: Removing stale PID file /var/run/suricata.pid
may 06 13:21:21 lisa-VirtualBox suricata[4487]: Starting suricata in IDS (af-packet) mode... done.
may 06 13:21:21 lisa-VirtualBox systemd[1]: Started LSB: Next Generation IDS/IPS.
lisa@lisa-VirtualBox:~$

```

Figura 8.3.12

Esta configuración utiliza las últimas configuraciones recomendadas para el modo de operación IDS para configuraciones básicas.

Lo siguiente que cambio es el camino de las reglas. Adjunto un archivo de reglas personalizadas.

```
rule-files:
- suricata.rules
- /etc/suricata/rules/local.rules
```

Un estado activo significa que Suricata en realidad está monitoreando el tráfico de la red y registrando todos los registros en un directorio.

`ls -al /var/log/suricata`

```
lisa@lisa-VirtualBox:~$ ls -al /var/log/suricata/
total 988
drwxr-xr-x  5 root root    4096 may  3 13:54 .
drwxrwxr-x 14 root syslog  4096 may  6 10:45 ..
drwxr-xr-x  2 root root    4096 abr 23 15:05 certs
drwxr-xr-x  2 root root    4096 abr 23 15:05 core
-rw-r--r--  1 root root  657623 may  6 13:27 eve.json
-rw-r--r--  1 root root  21448 may  6 13:26 fast.log
drwxr-xr-x  2 root root    4096 abr 23 15:05 files
-rw-r--r--  1 root root 225985 may  6 13:27 stats.log
-rw-r--r--  1 root root  58803 may  6 13:21 suricata.log
-rw-r--r--  1 root root   1234 may  6 13:21 suricata-start.log
lisa@lisa-VirtualBox:~$ |
```

Figura 8.3.13

Para ejecutar una prueba rápida, se utiliza una de las reglas de Suricata que se incluyen en el archivo de reglas predeterminado.

`curl http://testmynids.org/uid/index.html`

```
lisa@lisa-VirtualBox:~$ curl http://testmynids.org/uid/index.html
uid=0(root) gid=0(root) groups=0(root)
lisa@lisa-VirtualBox:~$ |
```

Figura 8.3.14

Pero puede ser antes puede salir error como este, por eso hay que hacer una instalación de curl

```
lisa@lisa-VirtualBox:~$ curl http://testmynids.org/uid/index.html
No se ha encontrado la orden «curl», pero se puede instalar con:
sudo snap install curl # version 8.1.2, or
sudo apt install curl # version 7.81.0-1ubuntu1.14
Consulte «snap info curl» para ver más versiones.
```

Figura 8.3.15

Se instala con este comando

```
sudo apt install curl
```

```
lisa@lisa-VirtualBox:~$ sudo apt install curl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  libflashrom1 libftdi1-2 libllvm13
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes NUEVOS:
  curl
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 4 no actualizados.
Se necesita descargar 194 kB de archivos.
Se utilizarán 454 kB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7.81.0-1ubuntu1.16 [194 kB]
Descargados 194 kB en 0s (521 kB/s)
Seleccionando el paquete curl previamente no seleccionado.
(Leyendo la base de datos ... 213246 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../curl_7.81.0-1ubuntu1.16_amd64.deb ...
Desempaquetando curl (7.81.0-1ubuntu1.16) ...
Configurando curl (7.81.0-1ubuntu1.16) ...
Procesando disparadores para man-db (2.10.2-1) ...
lisa@lisa-VirtualBox:~$
```

Figura 8.3.16

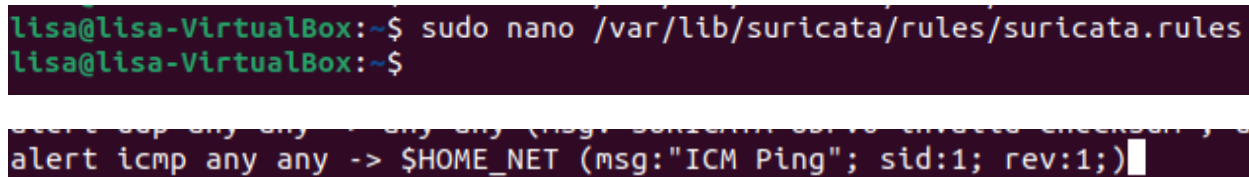
```
sudo cat /var/log/suricata/fast.log
```

```
05/20/2024-11:08:26.521982   [**] [1:2013028:7] ET POLICY curl User-Agent Outbound  
nd [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.16  
8.1.145:35358 -> 18.154.22.58:80  
05/20/2024-11:08:26.542072   [**] [1:2100498:7] GPL ATTACK_RESPONSE id check returned root [*lllislislllislislislisalisllisa@lisa-VirtualBox::~$  
lisa@lisa-VirtualBox::~$ |
```

---

Figura 8.3.17

Suricata se ha estado ejecutando en segundo plano y podemos ver dónde está el registro real, de dónde viene y hacia dónde se dirige.



The image shows a terminal window with a dark background. The prompt is 'lisa@lisa-VirtualBox:~\$'. The user enters 'sudo nano /var/lib/suricata/rules/suricata.rules'. The prompt changes to 'lisa@lisa-VirtualBox:~\$'. Below this, a line of Suricata rule syntax is visible: 'alert icmp any any -> \$HOME\_NET (msg:"ICM Ping"; sid:1; rev:1);'.

```
lisa@lisa-VirtualBox:~$ sudo nano /var/lib/suricata/rules/suricata.rules
lisa@lisa-VirtualBox:~$
alert icmp any any -> $HOME_NET (msg:"ICM Ping"; sid:1; rev:1);
```

## 8. Seguimiento y control

### Planificación de las Pruebas

**Fecha de Ejecución:** 05 de Junio de 2024

**Objetivo de las Pruebas:** Verificar que Suricata es capaz de detectar ataques de red, específicamente un escaneo de red utilizando NMAP, y generar los correspondientes reportes de seguridad. Se evaluará la capacidad del sistema para cumplir con los requisitos funcionales establecidos.

### Requisitos Evaluados:

1. Monitorización en tiempo real de amenazas.
2. Generación de alertas automáticas.
3. Registro detallado de eventos de seguridad.
4. Análisis de protocolos de red.
5. Capacidad de actualizar y personalizar reglas de detección.

### Descripción de las Pruebas

#### Prueba 1: Detección de un Ataque NMAP



- 
- **Descripción:** Realizar un escaneo de red utilizando la herramienta NMAP para simular un ataque. Suricata debe detectar este escaneo y generar una alerta.
  - **Herramientas Utilizadas:** NMAP para generar el ataque y Suricata para detectar el ataque.
  - **Método:** Se ejecutará el comando `nmap -sS <IP-Objetivo>` desde una máquina atacante dentro de la red. En este caso otra máquina de Ubuntu de un solo uso, en la misma red de VirtualBox que la máquina principal.

### **Pasos para la Ejecución:**

#### **1. Configuración del Entorno:**

- Instalar y configurar Suricata en una máquina virtual con Virtualbox.
- Se ha configurado otra maquina Ubuntu en VirtualBox en la misma red virtual que Suricata.

#### **2. Ejecución del Ataque:**

- Desde la máquina atacante, ejecutar el comando `nmap -sS <IP-Objetivo>` para iniciar un escaneo SYN en la red.

#### **3. Monitoreo y Detección:**

- Observar en tiempo real el log de Suricata para verificar la detección del escaneo de red.

#### **4. Generación de Reporte:**

- Recopilar los logs generados por Suricata y crear un reporte detallado de la actividad detectada.

### **Resultados de las Pruebas**

#### **Prueba 1: Detección de un Ataque NMAP**

- **Ejecución del Ataque:**

- 
- El comando `nmap -sS 192.168.100.3` se ejecutó exitosamente desde la máquina atacante a las 12:08 del 5 de Junio de 2024.
  - **Detección en Tiempo Real:**
    - A las 12:08, Suricata detectó el escaneo SYN y generó una alerta en tiempo real, la cual fue visible en el log de Suricata.
  - **Detalle del Evento:**
    - **Alerta Generada:** Escaneo de red SYN detectado.
    - **Fuente del Ataque:** IP de la máquina atacante.
    - **Destino del Ataque:** IP-Objetivo.
    - **Hora de Detección:** 12:08.
    - **Descripción:** Suricata identificó múltiples intentos de conexión SYN no establecidos, indicando un escaneo de puertos.
  - **Generación de Reporte:**
    - Se generó un reporte detallado del evento, incluyendo la fuente, destino, tipo de ataque, y hora de detección. El reporte se almacenó en el sistema de gestión de logs de Suricata.

## Verificación del Cumplimiento de Objetivos

### Objetivos del Proyecto Evaluados:

1. **Monitorización en Tiempo Real:** Confirmado. Suricata monitorizó y detectó el ataque en tiempo real.
2. **Registro Detallado de Eventos:** Confirmado. Los eventos fueron registrados con detalles precisos sobre la naturaleza del ataque.
3. **Análisis de Protocolos de Red:** Confirmado. Suricata analizó el tráfico SYN del protocolo TCP.

- 
4. **Actualización y Personalización de Reglas:** Confirmado. Las reglas de detección fueron adecuadamente configuradas y pudieron ser actualizadas.

Las alertas fueron generadas en tiempo real, los eventos fueron registrados de manera detallada y se verificó la capacidad del sistema para analizar protocolos y actualizar sus reglas de detección. Los resultados confirman la eficacia de Suricata para la monitorización y seguridad de la red.

## 9. FUENTES DE DOCUMENTACIÓN

1. GeeksforGeeks. (2022, 20 julio). *Approaches to Intrusion Detection and Prevention*. GeeksforGeeks.  
<https://www.geeksforgeeks.org/approaches-to-intrusion-detection-and-prevention/?ref=lbp>
2. Wallen, D. (2020, 3 marzo). *Intrusion Detection Systems: A Deep Dive Into NIDS & HIDS - Security Boulevard*. Security Boulevard.  
<https://securityboulevard.com/2020/03/intrusion-detection-systems-a-deep-dive-into-nids-hids>
3. William, J., & William, J. (2020, 28 octubre). *NIDs vs HIDs: Purpose, Core Functions & Benefits*. Temok Hosting Blog.  
<https://www.temok.com/blog/nids-vs-hids>.
4. *Compare Firewall and Intrusion Detection System (IDS)*. (s. f.).  
<https://www.ques10.com/p/13428/compare-firewall-and-intrusion-detection-system-id>

- 
5. Leblond, E. (s. f.). *Suricata: The First 12 Years of Innovation*.  
<https://www.stamus-networks.com/blog/suricata-the-first-12-years-of-innovation>
  6. Rapid. (2017, 8 diciembre). *How to Install Suricata NIDS on Ubuntu Linux*.  
Rapid7.  
<https://www.rapid7.com/blog/post/2017/02/14/how-to-install-suricata-nids-on-ubuntu-linux>
  7. Camisso, J. (2021, 22 octubre). *How To Install Suricata on Ubuntu 20.04*.  
DigitalOcean.  
<https://www.digitalocean.com/community/tutorials/how-to-install-suricata-on-ubuntu-20-04>
  8. *Suricata User Guide — Suricata 6.0.0 documentation*. (s. f.).  
<https://suricata.readthedocs.io/en/suricata-6.0.0>

## 10. CONCLUSIONES

Actualmente, las redes empresariales manejan cada vez más tráfico y muchas de ellas suelen transmitir 10GB por segundo en la red troncal.

La naturaleza multiproceso de Suricata permite a los usuarios escalar horizontalmente en un solo dispositivo agregando subprocesos de procesamiento de paquetes según sea necesario.

Suricata es una gran herramienta para tener en su arsenal de detección de intrusiones. Los datos obtenidos de Suricata pueden ayudar a crear un desglose geográfico del tráfico que entra y sale de la red.

Este trabajo examinó e instaló el sistema de detección de ataques de Suricata. Después de instalar Suricata, se editó la configuración predeterminada para agregar un ID de hilo comunitario para usar con otras herramientas de

---

seguridad, se habilitó la recarga de reglas en tiempo real y se cargó el conjunto de reglas inicial.

Luego de verificar la configuración de Suricata, se inició el proceso y se generó tráfico HTTP de prueba, lo que confirmó que Suricata puede detectar tráfico sospechoso.