PRÁCTICA BLUE TEAM

Infraestructura de Monitorización y Análisis de Logs

Sistema SIEM con pfSense, Honeypot y Suricata

pfsense COWRIE SURICATA KALI

Nicolás Navares

Bootcamp de Ciberseguridad – KeepCoding

Enero 2026

Índice de Contenidos

1. Introducción

Este proyecto documenta la implementación de una infraestructura de seguridad completa basada en un sistema **SIEM (Security Information and Event Management)** utilizando **Elastic Stack**, integrado con múltiples fuentes de logs distribuidas en diferentes segmentos de red.

El objetivo principal de esta práctica es demostrar la capacidad de diseñar, implementar y gestionar una arquitectura de red segmentada con monitorización centralizada, aplicando principios de defensa en profundidad y aislamiento de servicios.

1.1. Objetivos de la Práctica

- Diseñar e implementar una infraestructura de red segmentada con pfSense
- Configurar reglas de firewall para aislar correctamente los segmentos de red
- Desplegar un honeypot SSH en la DMZ para capturar intentos de intrusión
- Implementar Suricata IDS/IPS para detección de amenazas en red
- Centralizar y analizar logs de múltiples fuentes en Elastic SIEM
- Validar la correcta recepción y visualización de eventos de seguridad

1.2. Tecnologías Utilizadas

Componente	Descripción
pfSense	Firewall y router open source basado en FreeBSD para segmentación de red
Elastic Stack	Plataforma SIEM cloud para agregación, análisis y visualización de logs
Cowrie	Honeypot SSH/Telnet de media interacción para captura de ataques
Suricata	Sistema IDS/IPS open source con motor de detección de amenazas
Windows 10	Sistema operativo cliente con agente de monitorización
Kali Linux	Distribución Linux para despliegue de honeypot y Suricata
Elastic Agent	Agente unificado para recolección y envío de logs al SIEM

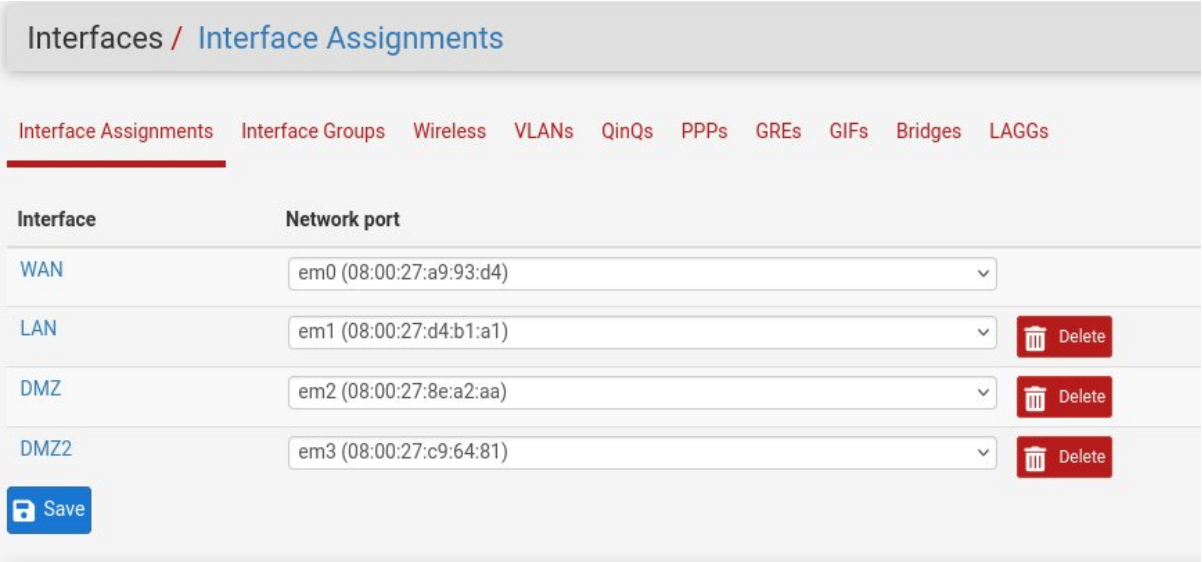
2. Arquitectura de Red

La infraestructura implementada sigue un modelo de segmentación de red con tres zonas diferenciadas y controladas por pfSense, conectadas a Elastic Cloud a través de Internet.

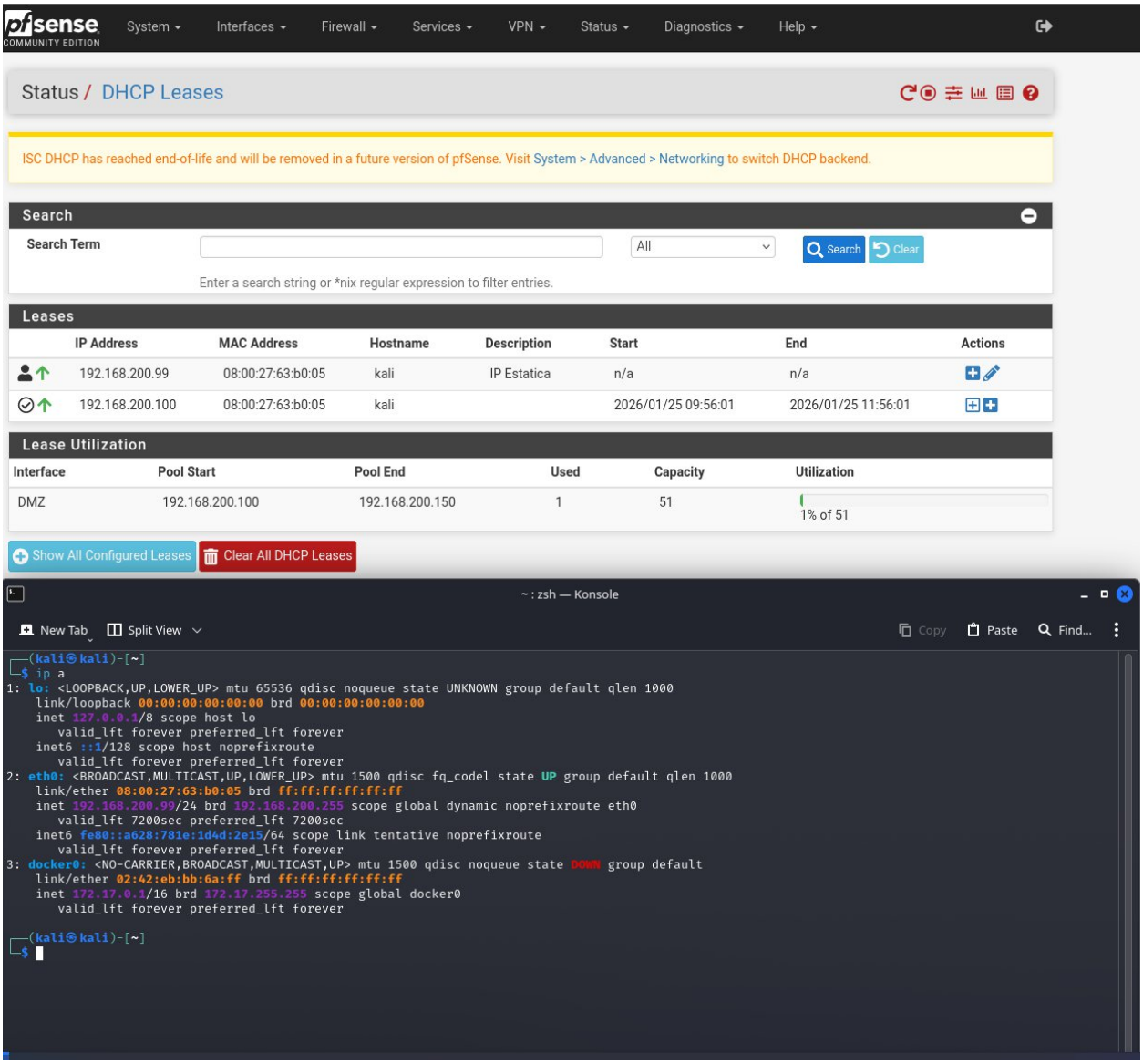
2.1. Topología de Red

Se han configurado cuatro interfaces de red en pfSense:

- **WAN** (em0): Interfaz de conexión a Internet (192.168.0.84/24)
- **LAN** (em1): Red de usuarios internos (192.168.100.1/24)
- **DMZ** (em2): Zona desmilitarizada para honeypot (192.168.200.1/24)
- **DMZ2** (em3): Segunda DMZ para servicios de monitorización (192.168.250.1/24)



Para garantizar que la conexión entrante vaya al **HONEYPOT** se le asigna una dirección IP estática a la máquina **Kali** que aloja el contenedor **Docker de Cowrie en la red DMZ:**



Asignación de IP estática al honeypot en la red **DMZ** En la captura se observa la configuración de **DHCP Leases del pfSense**, donde se ha establecido la dirección IP **192.168.200.99** como estática para el sistema que aloja el **honeypot Cowrie**. Esta asignación se basa en la dirección **MAC 08:00:27:63:b0:05** del adaptador de red de la máquina virtual Kali Linux en la interfaz DMZ.

Mantener una dirección IP consistente que permita configurar las reglas **NAT/Port Forward** de forma permanente y facilitar la correlación de **logs** en el **SIEM**, ya que la IP de origen será siempre la misma. Simplificar la administración del firewall al no depender de asignaciones dinámicas por DHCP. La interfaz **DMZ** utiliza el **pool DHCP 192.168.200.100-150**, por lo que la IP **99** queda fuera del rango dinámico, evitando conflictos de direccionamiento.

Figura 1: Configuración de interfaces de red en pfSense

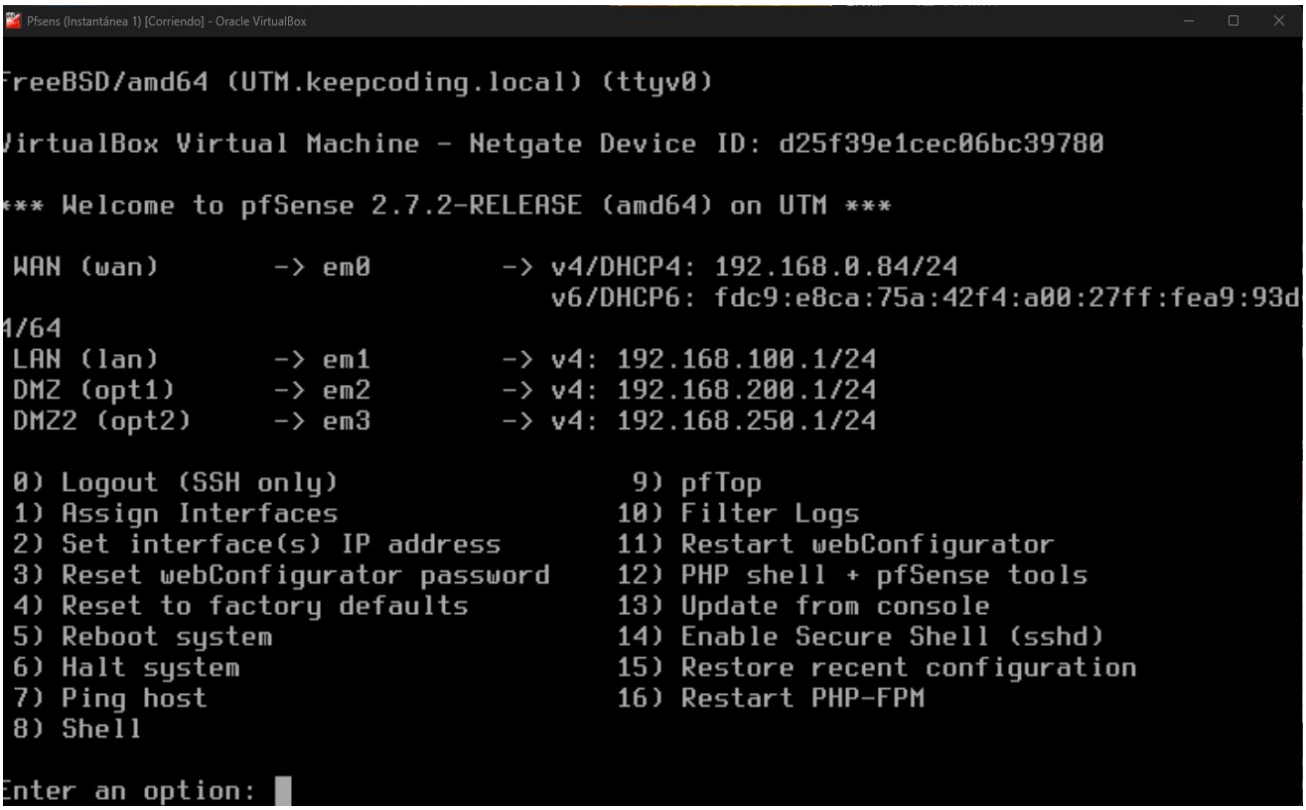


Figura 2: Resumen de configuración de red en pfSense

2.2. Distribución de Servicios

Segmento	Dirección IP	Servicios
LAN	192.168.100.103	Windows 10 + Elastic Agent
DMZ	192.168.200.99	Cowrie Honeypot (SSH) + Elastic Agent
DMZ2	192.168.250.103	Suricata IDS/IPS + Elastic Agent

2.3. Modelo de Seguridad

La arquitectura implementa los siguientes principios de seguridad:

- Segmentación de red: Separación física y lógica de servicios
- Principio de mínimo privilegio: Solo se permite el tráfico estrictamente necesario
- Defensa en profundidad: Múltiples capas de protección (firewall + IDS + honeypot)
- Aislamiento del honeypot: Sin acceso a redes internas (LAN y DMZ2)
- Monitorización centralizada: Todos los eventos se envían al SIEM cloud

3. Configuración del Firewall pfSense

Las reglas de firewall han sido diseñadas para garantizar el aislamiento entre segmentos de red, permitiendo únicamente el tráfico necesario para el correcto funcionamiento de los servicios de monitorización y acceso controlado desde el exterior.

3.1. Alias de Puertos Web

Para facilitar la gestión de reglas, se ha creado un alias de puertos web que agrupa los puertos HTTP (80) y HTTPS (443):

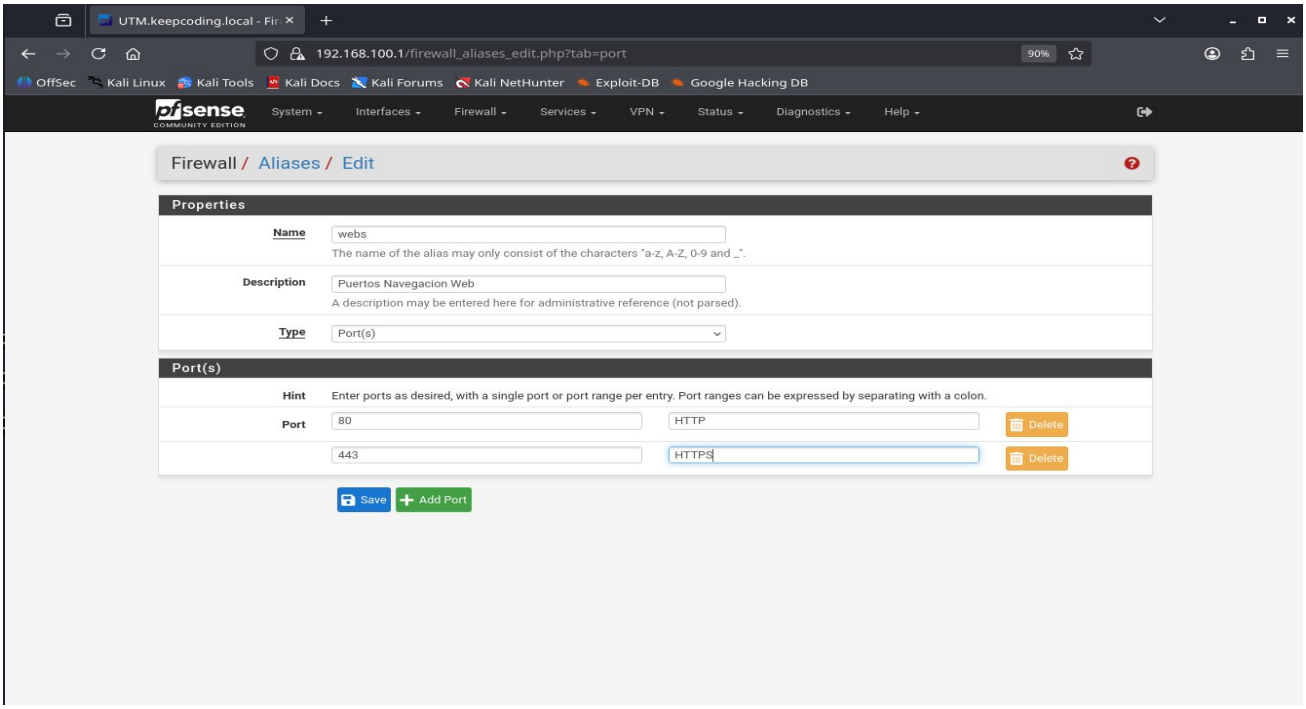


Figura 3: Configuración del alias de puertos web en pfSense

3.2. Reglas de la Red LAN

La red LAN tiene las siguientes políticas de firewall:

- **Regla Anti-Lockout:** Permite acceso administrativo al firewall desde la LAN
- **Bloqueo hacia DMZ:** Deniega todo el tráfico desde LAN hacia la red DMZ
- **Bloqueo hacia DMZ2:** Implícito - no hay regla de permiso explícita
- **Permitir salida a Internet:** Reglas por defecto permiten tráfico IPv4/IPv6 hacia cualquier destino

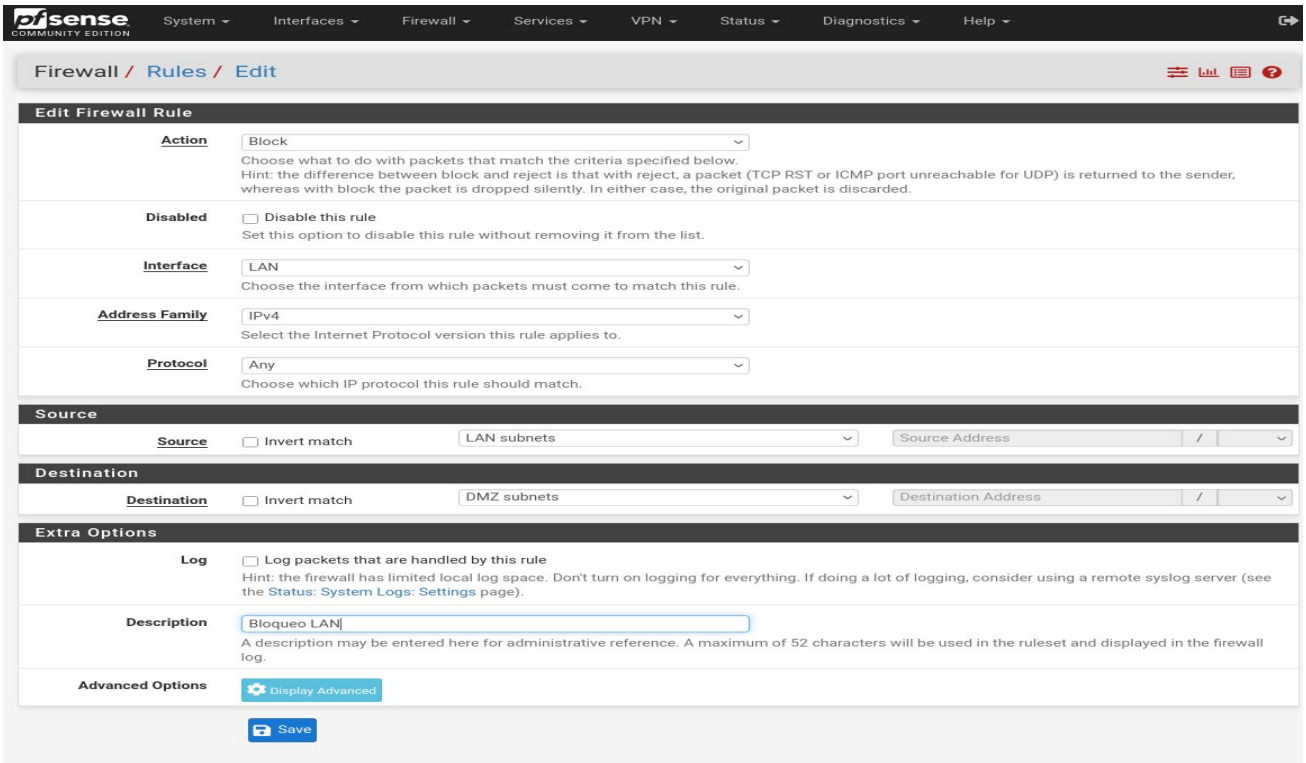


Figura 4: Regla de bloqueo de tráfico desde LAN hacia DMZ

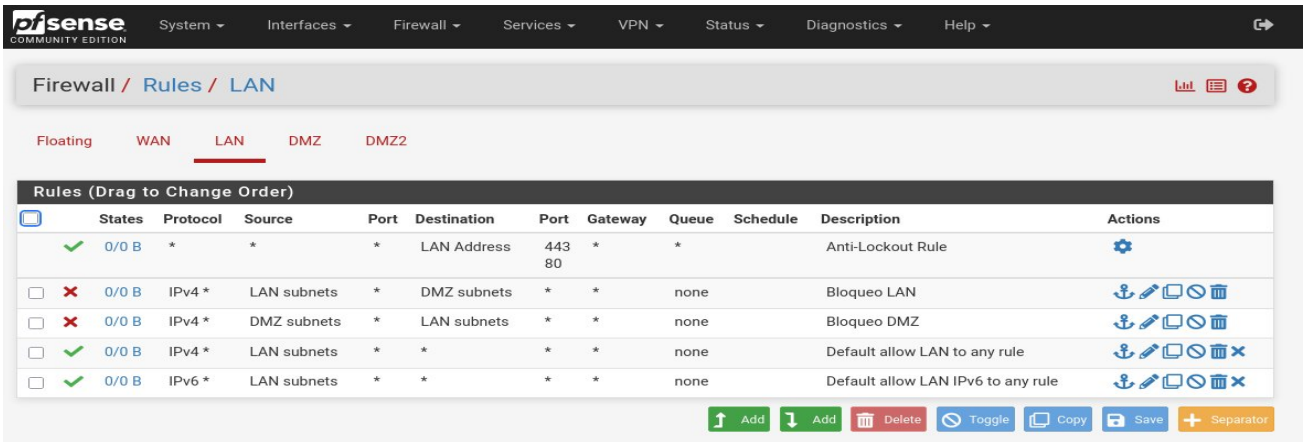


Figura 5: Resumen de reglas de firewall en la interfaz LAN

3.3. Reglas de la Red DMZ (Honeypot)

La DMZ donde reside el honeypot Cowrie tiene configuradas las siguientes reglas:

- **Bloqueo hacia DMZ2:** Se deniega explícitamente todo el tráfico desde DMZ hacia DMZ2
- **Bloqueo hacia LAN:** Se deniega todo el tráfico desde DMZ hacia la red interna LAN
- **Permitir ICMP:** Se permite tráfico ICMP (ping) para diagnóstico de red
- **Permitir DNS:** Se permite tráfico UDP al puerto 53 para resolución de nombres

- **Permitir tráfico web:** Se permite tráfico TCP hacia puertos 80/443 usando el alias **webs**

Estas reglas aseguran que el honeypot:

- No puede acceder a ninguna red interna (LAN ni DMZ2)
- Solo puede realizar consultas DNS y acceder a servicios web en Internet
- Puede enviar logs al SIEM cloud que está en Internet

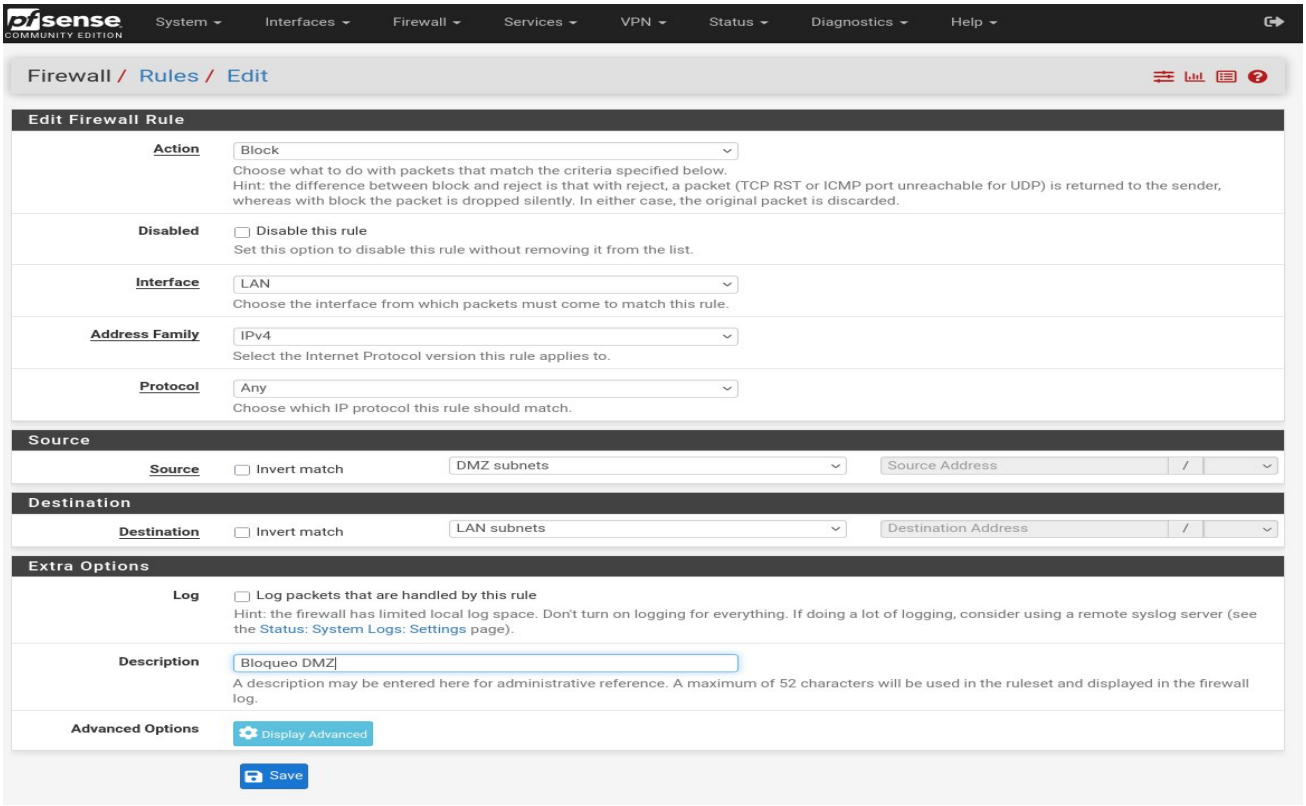


Figura 6: Regla de bloqueo desde DMZ hacia LAN

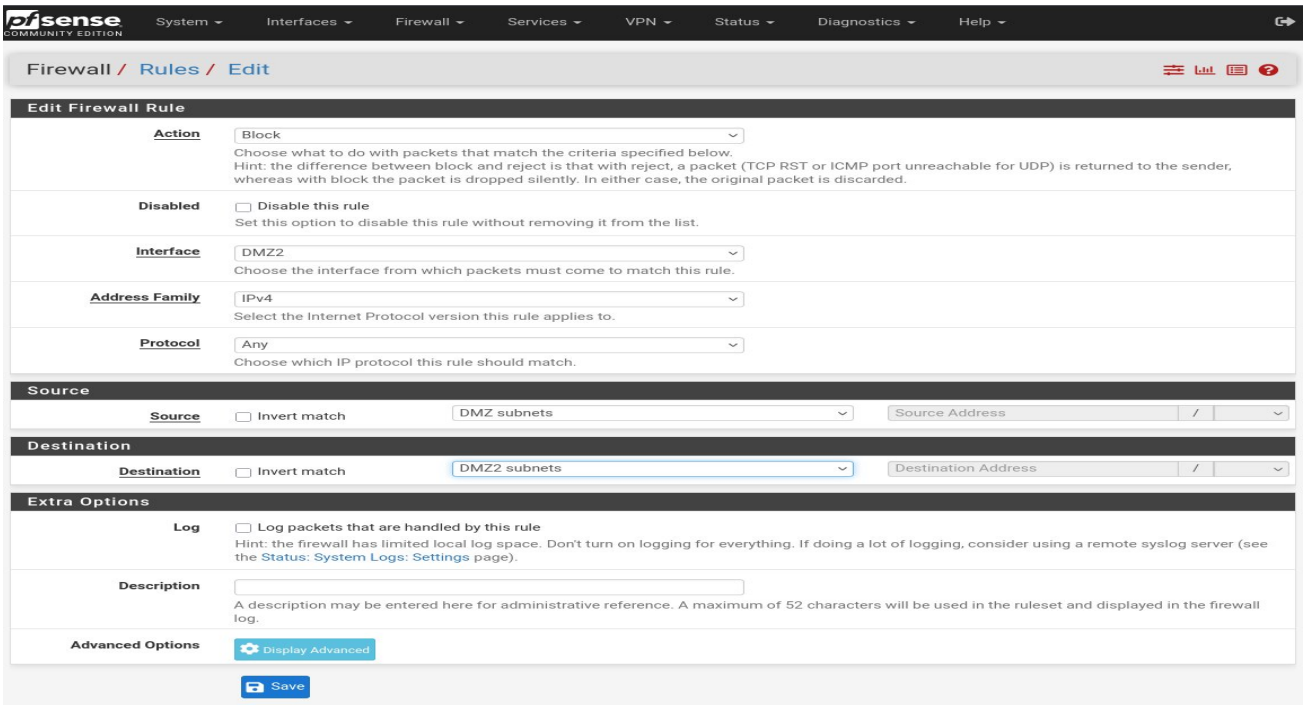


Figura 7: Regla de bloqueo desde DMZ hacia DMZ2

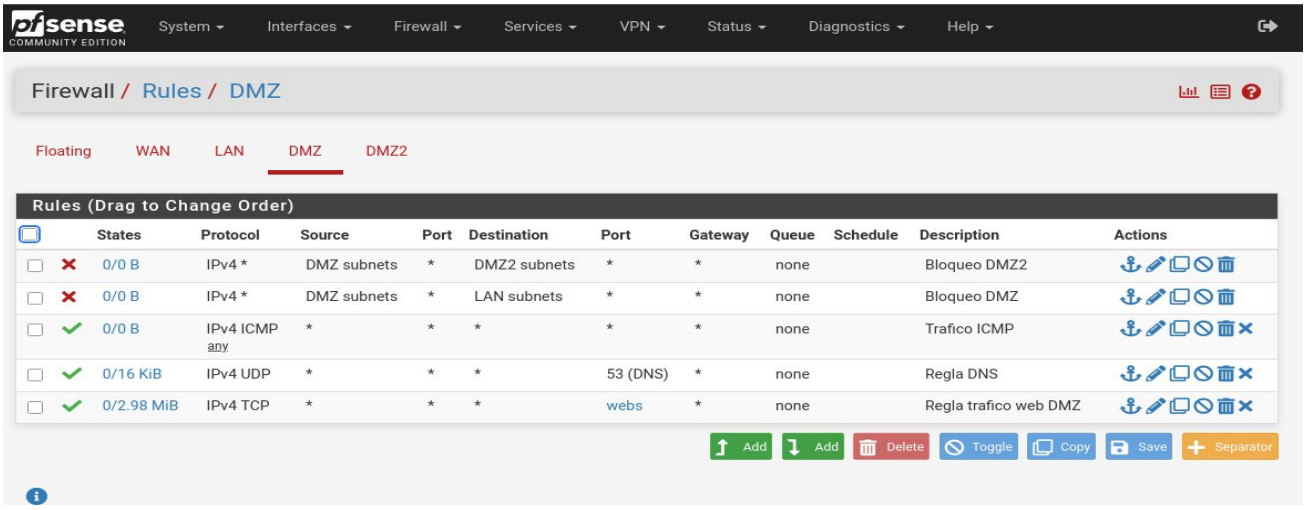


Figura 8: Resumen de reglas de firewall en la interfaz DMZ

3.4. Reglas de la Red DMZ2 (Suricata)

La **DMZ2** donde se ejecuta **Suricata** tiene las siguientes reglas configuradas:

- **Bloqueo hacia DMZ:** Se deniega todo el tráfico desde DMZ2 hacia la DMZ del honeypot
- **Bloqueo hacia LAN**: Implícito - no hay regla de permiso explícita
- **Permitir DNS:** Tráfico UDP al puerto 53 para resolución de nombres
- **Permitir tráfico web:** Tráfico TCP a puertos 80/443 usando el alias **webs**

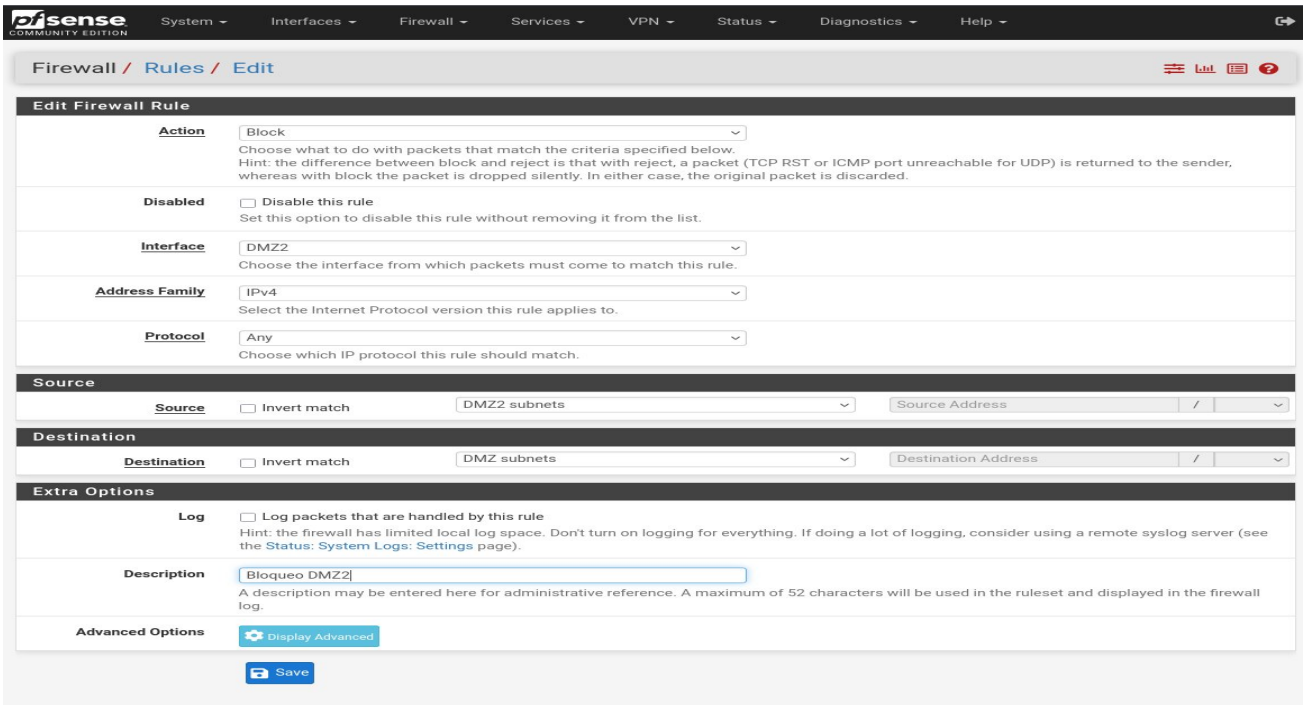


Figura 9: Regla de bloqueo desde DMZ2 hacia DMZ

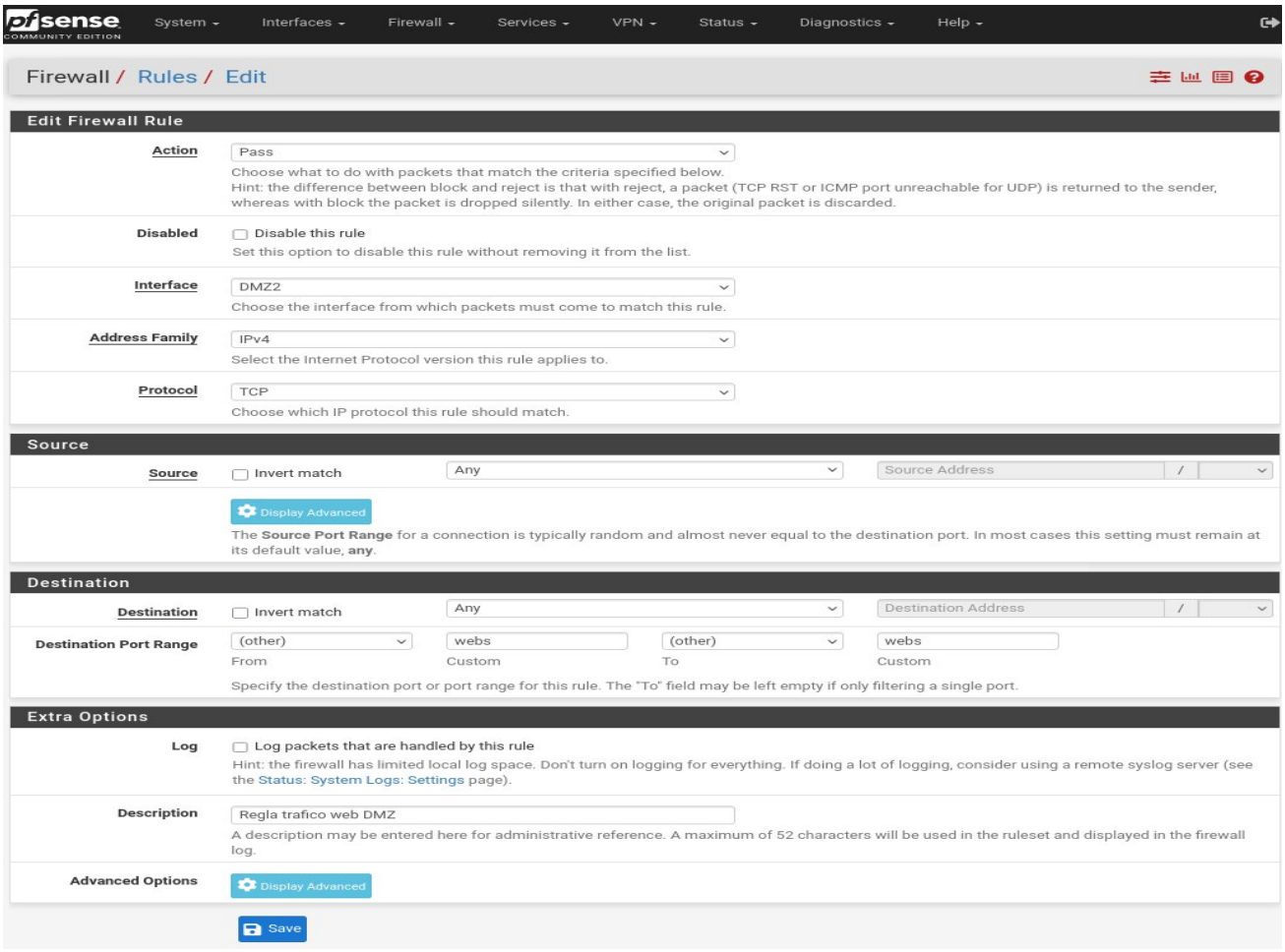


Figura 10: Configuración detallada de regla de tráfico web en DMZ2

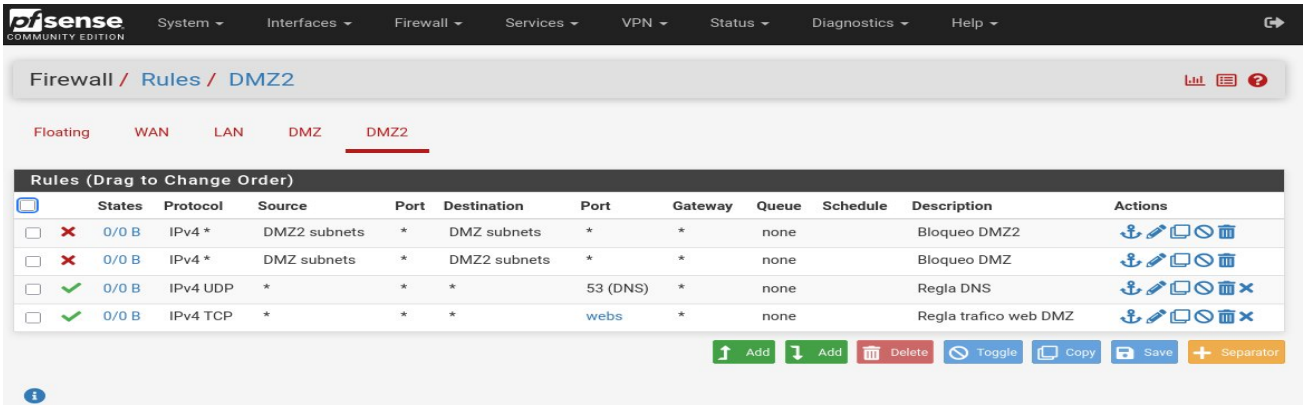
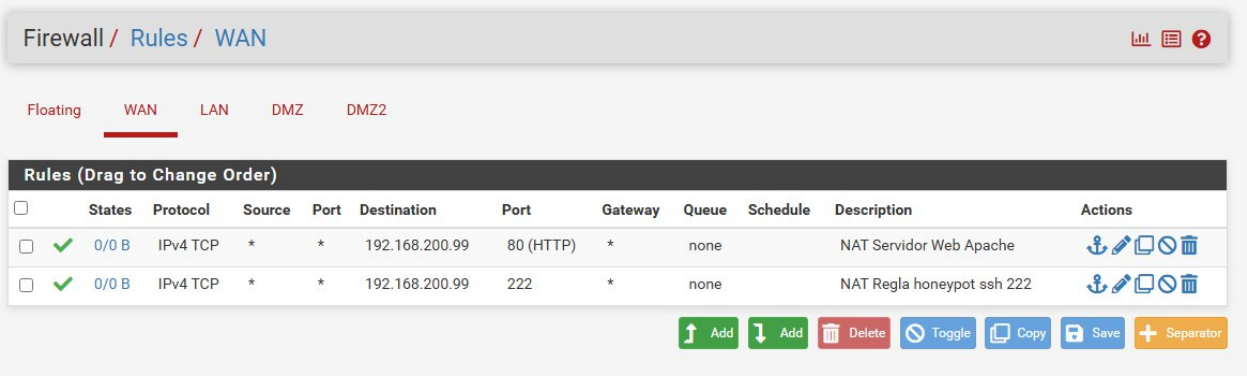


Figura 11: Resumen de reglas de firewall en la interfaz DMZ2

3.5. Reglas de la Interfaz WAN

La interfaz WAN está configurada para permitir el acceso desde el exterior (Internet) hacia el honeypot ubicado en la DMZ. Esto cumple con el requisito del enunciado de que el honeypot debe ser accesible desde la red WAN/máquina host en ambos sentidos:



Se han configurado dos reglas de entrada en la WAN:

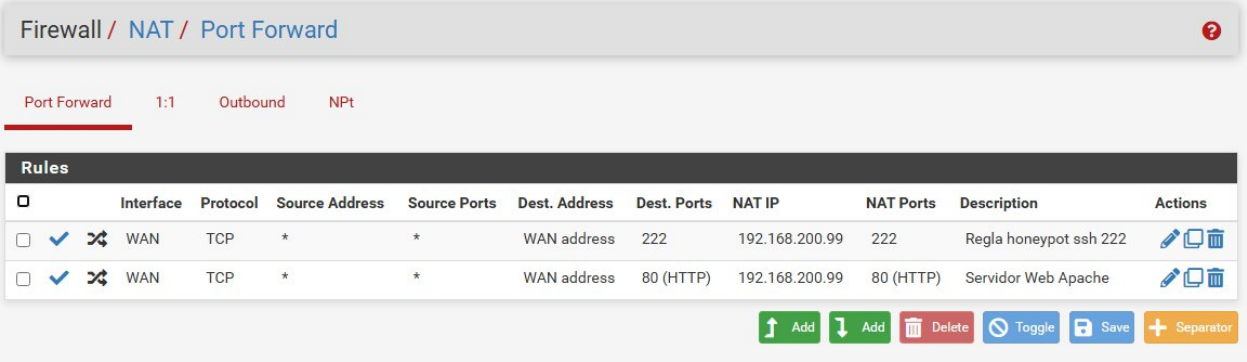
1. **NAT Servidor Web Apache**: Permite tráfico TCP desde cualquier origen hacia el puerto 80 (HTTP) de la dirección 192.168.200.99 (honeypot en DMZ)

2. **NAT Regla honeypot ssh 222**: Permite tráfico TCP desde cualquier origen hacia el puerto 222 del honeypot, que internamente está escuchando SSH en el puerto 2222 del contenedor Docker de Cowrie

Estas reglas funcionan en conjunto con las reglas NAT/Port Forward que se detallan en la siguiente sección.

3.6. Reglas NAT / Port Forward

Para facilitar el acceso externo al honeypot, se han configurado reglas de **Port Forward (NAT)** que redirigen el tráfico entrante desde la interfaz **WAN** hacia la dirección **IP del honeypot en la DMZ**:



Configuración de Port Forward en NAT Reglas configuradas: - Puerto **SSH (222)**: Redirige el tráfico TCP del puerto 222 de la WAN hacia el puerto 222 de la **IP 192.168.200.99**, donde **Cowrie** escucha las conexiones.

Puerto HTTP (80): Redirige el tráfico web (puerto 80) hacia el honeypot para capturar posibles ataques web (aunque en esta práctica el foco principal es el honeypot SSH) Esta configuración permite que atacantes externos puedan conectarse al honeypot como si fuera un servidor SSH real expuesto a Internet, mientras que las **reglas de firewall** garantizan que el **honeypot** permanece **aislado** de las redes internas **(LAN y DMZ2).**

4. Configuración del SIEM (Elastic Stack)

Se ha implementado una arquitectura de monitorización centralizada utilizando Elastic Cloud como plataforma SIEM, con agentes Elastic Agent desplegados en cada uno de los sistemas a monitorizar.

4.1. Políticas de Agente

Se han configurado tres políticas de agente diferenciadas, una para cada fuente de logs:

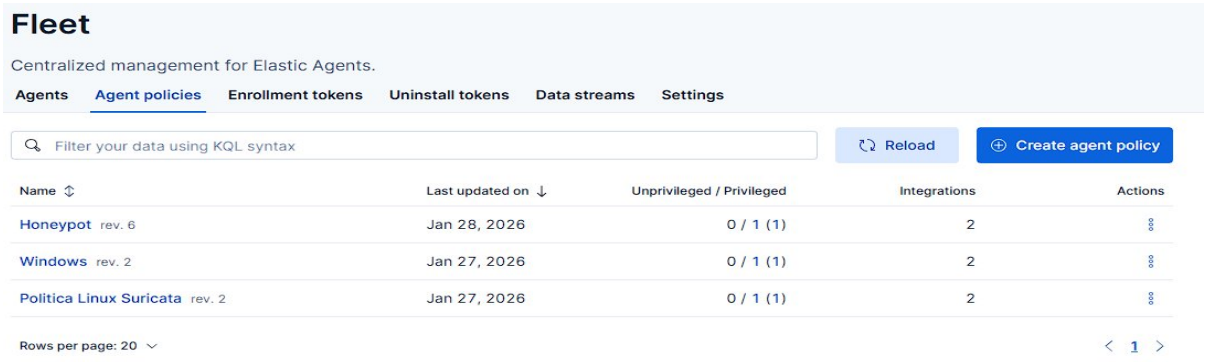


Figura 12: Resumen de políticas de agente en Elastic Fleet

4.1.1. Política Windows

Configurada para recolectar métricas y logs del sistema operativo Windows 10:

- **Integración:** System v2.12.0 (métricas de rendimiento del sistema)
- **Integración**: Windows v3.4.0 (logs de eventos de Windows)
- **Agente asignado:** 1 (Windows10)

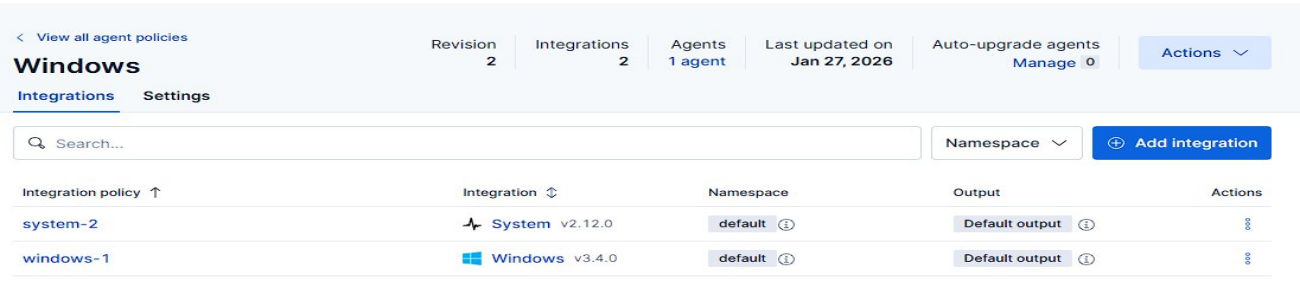


Figura 13: Configuración de la política Windows en Elastic Fleet

4.1.2. Política Honeypot

Configurada para recolectar logs del honeypot Cowrie mediante Custom Logs:

- **Integración:** System v2.12.0 (métricas del host Linux)
- **Integración:** Custom Logs (Filestream) v2.3.1 (logs de Cowrie)
- **Agente asignado:** 1 (kali - máquina con Cowrie en DMZ)
- **Namespace:** cowrie
- **Ruta del log:** /home/kali/logs-cowrie.log

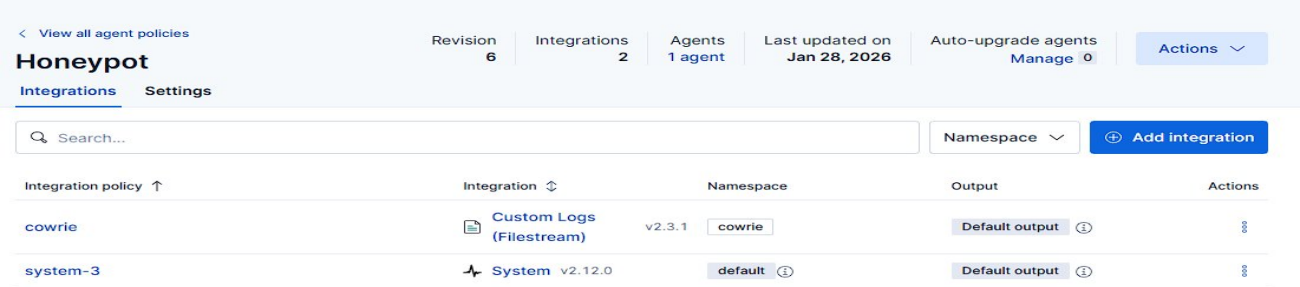


Figura 14: Configuración de la política Honeypot en Elastic Fleet

4.1.3. Política Linux Suricata

Configurada para recolectar eventos de Suricata IDS/IPS:

- **Integración:** Suricata v2.27.0 (logs EVE JSON de Suricata)
- **Integración:** System v2.12.0 (métricas del sistema)
- **Agente asignado:** 1 (kali - sistema con Suricata en DMZ2)
- **Ruta del log:** /var/log/suricata/eve.json

Figura 15: Configuración de la política Suricata en Elastic Fleet

4.2. Arquitectura de Agentes

Los agentes Elastic Agent se instalan localmente en cada sistema y se encargan de:

- Recolectar logs y métricas según la política asignada
- Procesar y enriquecer los datos antes del envío
- Establecer conexión segura con Elastic Cloud a través de Internet
- Mantener buffering local en caso de pérdida de conectividad
- Actualizar automáticamente las configuraciones desde Fleet

5. Implementación del Honeypot Cowrie

El **honeypot Cowrie** ha sido desplegado en la red DMZ (192.168.200.99) para simular un servidor **SSH vulnerable** y capturar intentos de intrusión.

5.1. Despliegue con Docker

Cowrie se ha ejecutado mediante Docker para facilitar su gestión y aislamiento:

Comando de ejecución:

sudo docker run -p 222:2222 cowrie/cowrie > logs-cowrie.log

Características del despliegue:

- **Puerto expuesto:** 222 (mapeado al puerto 2222 interno del contenedor)
- **Logs redirigidos a:** /home/kali/logs-cowrie.log
- **Modo de operación:** SSH honeypot de media interacción
- **Elastic Agent** lee el archivo de logs en tiempo real

Figura 16: Ejecución de Cowrie con Docker y generación de logs

5.2. Validación de Conectividad

Se ha verificado el correcto funcionamiento del honeypot mediante conexiones SSH de prueba desde la red WAN (máquina host):

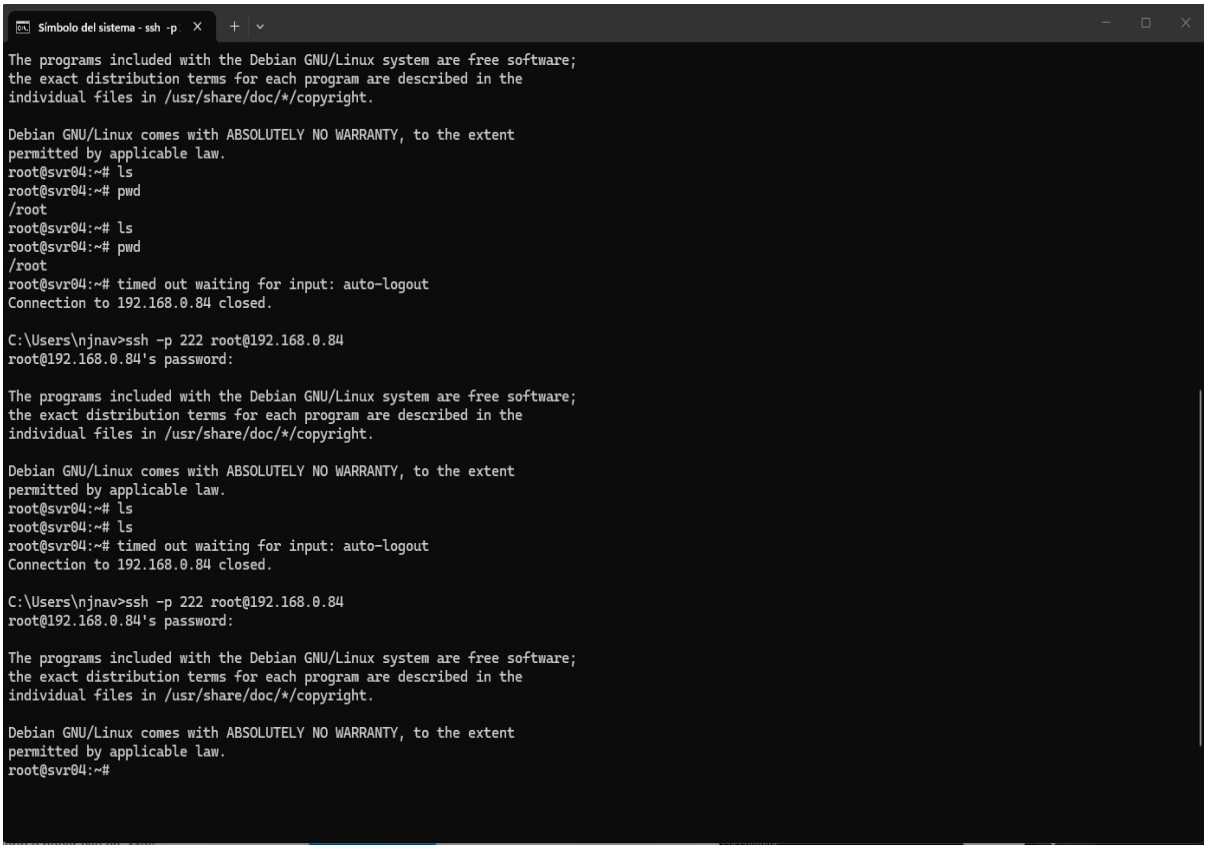


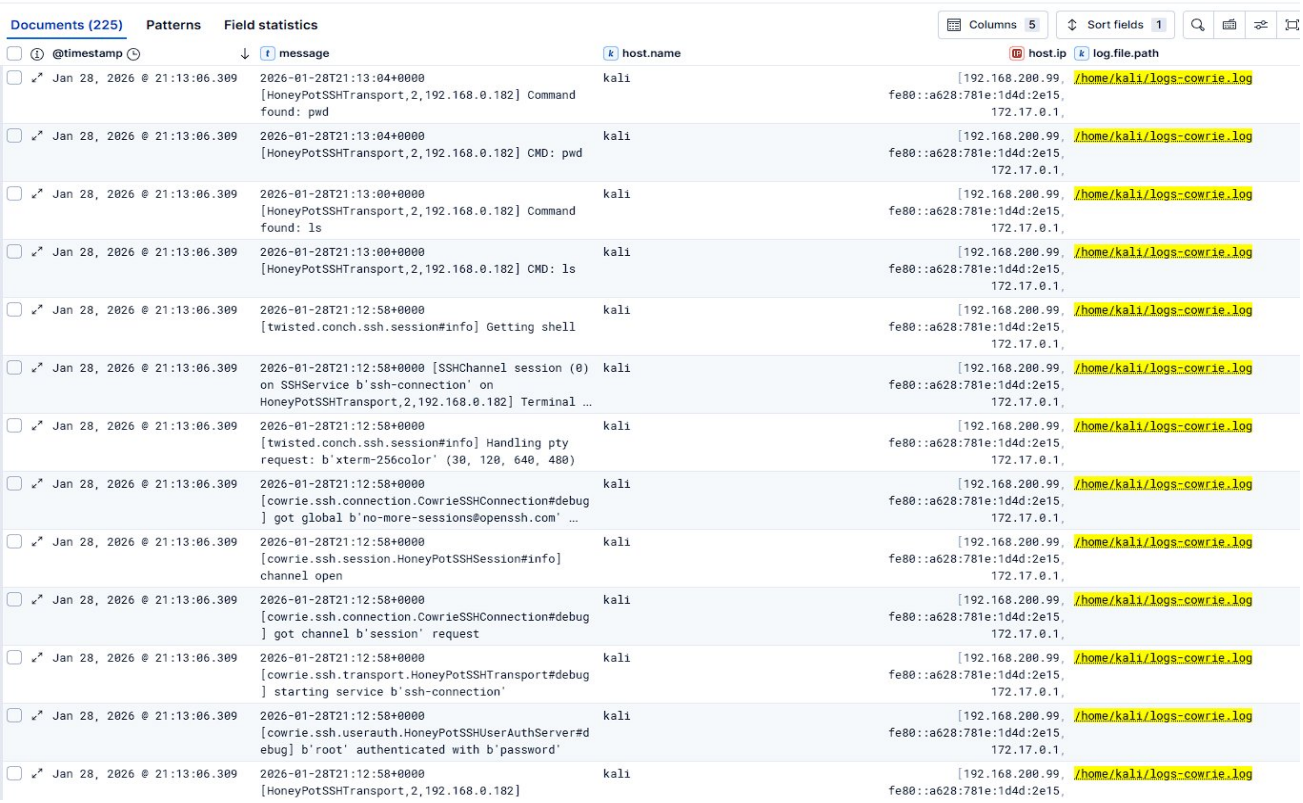
Figura 17: Conexiones SSH al honeypot desde el exterior (WAN)

En la captura se observa:

- Múltiples conexiones SSH desde 192.168.0.84 (WAN) al puerto 222
- Intentos de autenticación con credenciales (usuario: root, password: password)
- El honeypot responde y simula un sistema Debian GNU/Linux
- Los comandos ejecutados (ls, pwd) son registrados por Cowrie
- Todas las sesiones finalizan por timeout tras inactividad

5.3. Recepción de Logs en el SIEM

Los logs generados por **Cowrie** son procesados por **Elastic Agent** y enviados al **SIEM** en tiempo real:



Documents (225) | Patterns | Field statistics | Columns 5 | Sort fields 1

@timestamp	message	host.name	host.ip	log.file.path
Jan 28, 2026 @ 21:13:06.309	2026-01-28T21:13:04+0000 [HoneyPotSSHTransport,2,192.168.0.182] Command found: pwd	kali	[192.168.200.99, fe80::a628:781e:1d4d:2e15, 172.17.0.1,	/home/kali/logs-cowrie.log
Jan 28, 2026 @ 21:13:06.309	2026-01-28T21:13:04+0000 [HoneyPotSSHTransport,2,192.168.0.182] CMD: pwd	kali	[192.168.200.99, fe80::a628:781e:1d4d:2e15, 172.17.0.1,	/home/kali/logs-cowrie.log
Jan 28, 2026 @ 21:13:06.309	2026-01-28T21:13:00+0000 [HoneyPotSSHTransport,2,192.168.0.182] Command found: ls	kali	[192.168.200.99, fe80::a628:781e:1d4d:2e15, 172.17.0.1,	/home/kali/logs-cowrie.log
Jan 28, 2026 @ 21:13:06.309	2026-01-28T21:13:00+0000 [HoneyPotSSHTransport,2,192.168.0.182] CMD: ls	kali	[192.168.200.99, fe80::a628:781e:1d4d:2e15, 172.17.0.1,	/home/kali/logs-cowrie.log
Jan 28, 2026 @ 21:13:06.309	2026-01-28T21:12:58+0000 [twisted.conch.ssh.session#info] Getting shell	kali	[192.168.200.99, fe80::a628:781e:1d4d:2e15, 172.17.0.1,	/home/kali/logs-cowrie.log
Jan 28, 2026 @ 21:13:06.309	2026-01-28T21:12:58+0000 [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotSSHTransport,2,192.168.0.182] Terminal …	kali	[192.168.200.99, fe80::a628:781e:1d4d:2e15, 172.17.0.1,	/home/kali/logs-cowrie.log
Jan 28, 2026 @ 21:13:06.309	2026-01-28T21:12:58+0000 [twisted.conch.ssh.session#info] Handling pty request: b'xterm-256color' (30, 120, 640, 480)	kali	[192.168.200.99, fe80::a628:781e:1d4d:2e15, 172.17.0.1,	/home/kali/logs-cowrie.log
Jan 28, 2026 @ 21:13:06.309	2026-01-28T21:12:58+0000 [cowrie.ssh.connection.CowrieSSHConnection#debug] got global b'no-more-sessions@openssh.com' …	kali	[192.168.200.99, fe80::a628:781e:1d4d:2e15, 172.17.0.1,	/home/kali/logs-cowrie.log
Jan 28, 2026 @ 21:13:06.309	2026-01-28T21:12:58+0000 [cowrie.ssh.session.HoneyPotSSHSession#info] channel open	kali	[192.168.200.99, fe80::a628:781e:1d4d:2e15, 172.17.0.1,	/home/kali/logs-cowrie.log
Jan 28, 2026 @ 21:13:06.309	2026-01-28T21:12:58+0000 [cowrie.ssh.connection.CowrieSSHConnection#debug] got channel b'session' request	kali	[192.168.200.99, fe80::a628:781e:1d4d:2e15, 172.17.0.1,	/home/kali/logs-cowrie.log
Jan 28, 2026 @ 21:13:06.309	2026-01-28T21:12:58+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-connection'	kali	[192.168.200.99, fe80::a628:781e:1d4d:2e15, 172.17.0.1,	/home/kali/logs-cowrie.log
Jan 28, 2026 @ 21:13:06.309	2026-01-28T21:12:58+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' authenticated with b'password'	kali	[192.168.200.99, fe80::a628:781e:1d4d:2e15, 172.17.0.1,	/home/kali/logs-cowrie.log
Jan 28, 2026 @ 21:13:06.309	2026-01-28T21:12:58+0000 [HoneyPotSSHTransport,2,192.168.0.182]	kali	[192.168.200.99, fe80::a628:781e:1d4d:2e15,	/home/kali/logs-cowrie.log

Figura 18: Visualización de logs del honeypot en Elastic SIEM

Los eventos capturados incluyen:

- **Comandos ejecutados por los atacantes (pwd, ls, etc.)**
- **Información de las sesiones SSH (origen, credenciales usadas)**
- **Hostname: kali (sistema donde corre el honeypot)**
- **IP de origen: 192.168.200.99 (red DMZ)**
- **Ruta del log: /home/kali/logs-cowrie.log**
- **Timestamp preciso de cada evento**

6. Implementación de Suricata IDS/IPS

Suricata ha sido desplegado en la red DMZ2 (192.168.250.103) para realizar detección y prevención de intrusiones mediante análisis de tráfico de red.

6.1. Configuración de Suricata

Se ha configurado Suricata con los siguientes parámetros:

- **Interfaz monitoreada:** eth0 (interfaz de red de la DMZ2)
- **Formato de logs:** EVE JSON (/var/log/suricata/eve.json)
- **Motor de detección:** Suricata.
- **Reglas activas:** Detección de tráfico general y alertas personalizadas

6.2. Reglas de Detección

Se ha implementado una regla básica de detección para validar el funcionamiento:

alert tcp any any -> any any (msg:"trafico detectado"; sid:1;)

Esta regla genera alertas para todo el tráfico de red capturado, permitiendo verificar que Suricata está analizando correctamente los paquetes y generando eventos en formato EVE JSON.

6.3. Visualización de Alertas en el SIEM

Los eventos de **Suricata** son procesados y visualizados en el **SIEM** con información detallada:

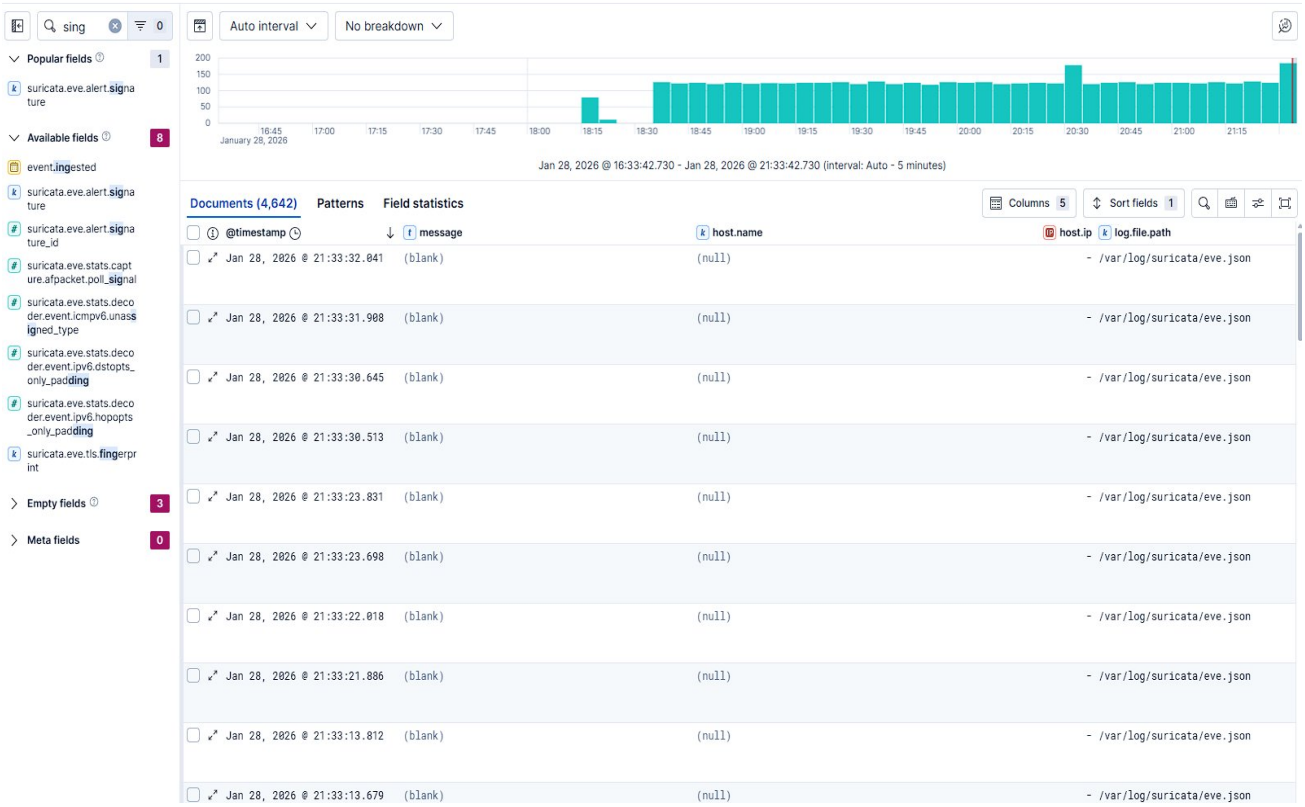


Figura 19: Visualización de alertas de Suricata en Elastic SIEM

La interfaz del SIEM muestra:

- **Gráfico temporal de alertas:** Distribución de eventos a lo largo del tiempo
- **Campos populares:** suricata.eve.alert.signature con el nombre de la regla
- **Campos disponibles:** Más de 8 campos diferentes para análisis
- **Total de documentos:** 4,642 eventos capturados
- **Período de análisis:** Desde las 16:33 hasta las 21:33 del 28 de enero de 2026
- **Fuente de logs:** /var/log/suricata/eve.json

7. Monitorización del Sistema Windows

El equipo **Windows 10** ubicado en la red **LAN (192.168.100.103)** envía métricas de rendimiento y logs del sistema al SIEM mediante Elastic Agent.

7.1. Métricas Recolectadas

El agente en Windows está configurado para recolectar:

- Métricas de rendimiento (CPU, memoria, disco, red)
- Eventos del sistema operativo
- Logs de seguridad de Windows
- Información de procesos en ejecución

7.2. Visualización en el SIEM

Los eventos del sistema Windows son correctamente recibidos y procesados por Elastic:

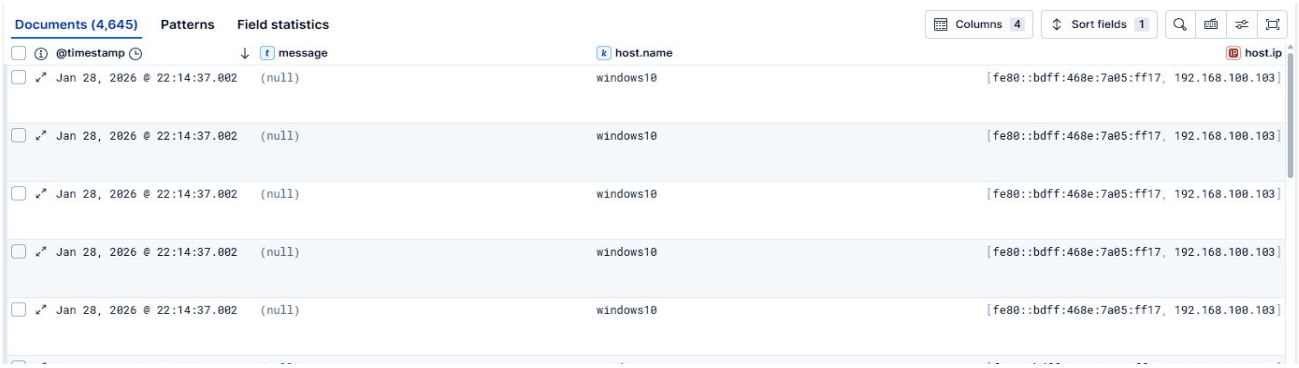


Figura 20: Logs del sistema Windows 10 en Elastic SIEM

Se puede observar:

- **Hostname:** windows10 (nombre del equipo en la LAN)
- **Host IP:** fe80::bdff:468e:7a05:ff17 (IPv6), 192.168.100.103 (IPv4)
- **Total de eventos**: 4,645 documentos recolectados
- **Timestamp:** 22:14:37.002 del 28 de enero de 2026
- **Data stream:** metrics-windows.perfmon-default

8. Validación y Pruebas Realizadas

Se han realizado múltiples pruebas para validar el correcto funcionamiento de la infraestructura implementada.

8.1. Pruebas de Aislamiento de Red

Se ha verificado que las reglas de firewall funcionan correctamente:

- El honeypot (DMZ) NO puede acceder a la LAN
- El honeypot (DMZ) NO puede acceder a la DMZ2
- La LAN NO puede acceder a la DMZ
- DMZ2 NO puede acceder a la DMZ
- El tráfico desde WAN hacia el honeypot (puerto 222) es permitido
- Todas las redes pueden alcanzar Internet y Elastic Cloud

8.2. Pruebas de Captura de Logs

Se ha confirmado la correcta recepción de logs en el SIEM:

- Logs de Cowrie: Comandos SSH y sesiones capturadas correctamente
- Logs de Suricata: Alertas de tráfico detectadas y procesadas
- Logs de Windows: Métricas de rendimiento recibidas cada 10 segundos
- Todos los logs incluyen la IP correcta del segmento de red correspondiente

8.3. Pruebas de Honeypot

Se han realizado múltiples conexiones SSH al honeypot para verificar su funcionamiento:

- Conexiones desde la red WAN (192.168.0.84) al puerto 222
- Autenticación con credenciales de prueba (root:password)
- Ejecución de comandos comunes (ls, pwd, cd)
- Todos los comandos fueron registrados en logs-cowrie.log
- Los logs fueron procesados y enviados al SIEM correctamente

8.4. Pruebas de Detección de Suricata

Se ha generado tráfico de red para validar la detección de Suricata:

- Navegación web desde la máquina en DMZ2
- Generación de más de 4,600 alertas en un período de 5 horas
- Verificación de geolocalización de IPs de origen
- Confirmación de análisis de múltiples protocolos (TCP, UDP, ICMP)

9. Análisis Detallado de Logs JSON

Como último paso del curso, se presenta el análisis detallado de la estructura JSON de los logs recibidos desde cada fuente en el SIEM, explicando el significado de los campos más relevantes. Este análisis es fundamental para comprender cómo interpretar y correlacionar eventos de seguridad.

9.1. Estructura de Logs del Honeypot (Cowrie)

A continuación se presenta el análisis detallado de la estructura JSON de un log capturado por el honeypot Cowrie desde el SIEM Elastic:

BLOQUE JSON COMPLETO - LOG HONEYPOT COWRIE

```
{
  "_index": ".ds-logs-filestream.generic-cowrie-2026.01.28-000001",
  "_id": "AZwGdAPtV5wNIakYI_MC",
  "_version": 1,
  "_source": {
    "@timestamp": "2026-01-28T21:13:06.309Z",
    "agent": {
      "ephemeral_id": "979ea20c-99a0-4073-b5d7-36dec8646159",
      "id": "7a33f160-d194-4cbb-b927-6ac5597a473d",
      "name": "kali",
      "type": "filebeat",
      "version": "9.2.4"
    },
    "data_stream": {
      "dataset": "filestream.generic",
      "namespace": "cowrie",
      "type": "logs"
    },
```

```
  "host": {
    "hostname": "kali",
    "id": "b69758c0cad3481e967dcad827001d56",
    "ip": ["192.168.200.99", "fe80::a628:781e:1d4d:2e15", "172.17.0.1"],
    "mac": ["02-42-B0-70-B4-A7", "08-00-27-63-B0-05", "82-9B-84-59-73-E2"],
    "name": "kali",
    "os": {
      "name": "Kali GNU/Linux",
      "version": "2025.4",
      "kernel": "6.18.5+kali-amd64",
      "family": "debian"
    }
  },
  "log": {
    "file": {
      "path": "/home/kali/logs-cowrie.log",
      "inode": "4198472",
      "device_id": "2049"
    },
    "offset": "11204"
  },
  "message": "2026-01-28T21:13:04+0000 [HoneyPotSSHTransport,2,192.168.0.182]
        Command found: pwd"
 }
}
```


FIN DEL BLOQUE JSON

EXPLICACIÓN DE CAMPOS PRINCIPALES:

Campo	Descripción
_index	Índice de Elasticsearch donde se almacena el documento. Incluye el namespace *cowrie* y la fecha de creación.
@timestamp	Marca temporal del evento en formato ISO 8601 UTC (2026-01-28T21:13:06.309Z)
agent.id	Identificador único del agente Elastic Agent (7a33f160-d194-4cbb-b927-6ac5597a473d)
agent.name	Nombre del host donde se ejecuta el agente (kali)
agent.type	Tipo de agente que recolecta los logs (filebeat)
data_stream.namespace	Namespace personalizado para organizar los logs (cowrie)
host.ip	Direcciones IP del sistema: 192.168.200.99 (DMZ), fe80::a628:781e:1d4d:2e15 (IPv6), 172.17.0.1 (Docker)
host.os.name	Sistema operativo del host (Kali GNU/Linux 2025.4)
log.file.path	Ruta completa del archivo de log monitorizado (/home/kali/logs-cowrie.log)
message	Contenido del log generado por Cowrie. Incluye timestamp, sesión SSH identificada [HoneyPotSSHTransport,2,192.168.0.182] y comando ejecutado por el atacante (pwd)

PUNTOS CLAVE:

- La IP **192.168.200.99** confirma que el honeypot está en la red **DMZ**
- El mensaje captura el comando **pwd** ejecutado por el atacante desde **192.168.0.182**
- El namespace *cowrie* permite filtrar fácilmente estos logs en el SIEM
- Elastic Agent v9.2.4 recolecta y envía los logs en tiempo real

9.2. Estructura de Logs de Suricata IDS/IPS

A continuación se presenta el análisis detallado de una alerta JSON generada por Suricata desde el SIEM Elastic:

BLOQUE JSON COMPLETO - ALERTA SURICATA


```
{
  "_index": ".ds-logs-suricata.eve-default-2026.01.27-000001",
  "_id": "AZwGhgPtV5wNIam4Lyce",
  "_version": 1,
  "_source": {
    "@timestamp": "2026-01-28T21:33:32.041Z",
    "agent": {
      "id": "3bc54355-ff61-4717-b744-2882c2f7d82b",
      "name": "kali",
      "type": "filebeat",
      "version": "9.2.4"
    },
    "event": {
      "kind": "alert",
      "category": ["network", "intrusion_detection"],
      "type": ["allowed"],
      "severity": 3,
      "start": "2026-01-28T21:33:31.908Z"
    },
    "source": {
      "ip": "34.68.177.88",
      "port": 443,
      "bytes": 74,
```

```
  "packets": 1,
  "geo": {
    "city_name": "Council Bluffs",
    "country_name": "United States",
    "country_iso_code": "US",
    "region_name": "Iowa",
    "location": { "lat": 41.2591, "lon": -95.8517 }
  },
  "as": {
    "number": 396982,
    "organization": { "name": "Google LLC" }
  }
},
"destination": {
  "ip": "192.168.250.103",
  "port": 40710,
  "bytes": 74,
  "packets": 1
},
"network": {
  "transport": "tcp",
  "bytes": 148,
  "packets": 2,
  "community_id": "1:G3zUgXtao3lgnKvMVEd4F5Kr1aQ="
},
"rule": {
  "id": "1",
  "name": "trafico detectado"
},
"suricata": {
```

```
    "eve": {
      "event_type": "alert",
      "flow_id": "1086289812340734",
      "in_iface": "eth0",
      "alert": {
        "signature": "trafico detectado",
        "signature_id": 1,
        "gid": 1,
        "rev": 0
      }
    }
  },
  "observer": {
    "hostname": "kali",
    "ip": ["192.168.250.103", "fe80::61a8:e17c:6790:b2fe", "172.17.0.1"],
    "product": "Suricata",
    "vendor": "OISF",
    "type": "ids"
  }
 }
}
```

FIN DEL BLOQUE JSON

EXPLICACIÓN DE CAMPOS PRINCIPALES:

Campo	Descripción
event.kind	Tipo de evento generado (alert = alerta de seguridad detectada)
event.category	Categorías del evento: network (tráfico de red) e intrusion_detection (detección de intrusión)
event.severity	Nivel de severidad de la alerta (3 = medio)
source.ip	Dirección IP de origen del tráfico (34.68.177.88 - Google LLC en Iowa, USA)
source.port	Puerto de origen (443 = HTTPS)
source.geo	Información de geolocalización: Council Bluffs, Iowa, United States con coordenadas lat/lon
source.as	Sistema autónomo: AS396982 perteneciente a Google LLC
destination.ip	IP de destino: 192.168.250.103 (máquina con Suricata en la red DMZ2)
destination.port	Puerto de destino: 40710 (puerto efímero del cliente)
network.transport	Protocolo de transporte utilizado (tcp)
network.community_id	Hash único que identifica el flujo de red para correlación
rule.name	Nombre de la regla de Suricata que generó la alerta (trafico detectado)
suricata.eve.flow_id	Identificador único del flujo en Suricata (1086289812340734)
suricata.eve.in_iface	Interfaz de red donde se capturó el tráfico (eth0)
observer.product	Producto de seguridad que generó el evento (Suricata)
observer.type	Tipo de sistema de observación (ids = Intrusion Detection System)

PUNTOS CLAVE:

- La IP de destino **192.168.250.103** confirma que **Suricata** está en la red **DMZ2**
- La geolocalización muestra tráfico desde servidores de Google en Estados Unidos
- El flujo completo (148 bytes, 2 paquetes) está siendo analizado por Suricata
- La regla ID 1 detecta y registra todo el tráfico de red para validación

9.3. Estructura de Logs de Windows 10

A continuación se presenta el análisis detallado de una métrica de rendimiento de Windows desde el SIEM Elastic:

BLOQUE JSON COMPLETO - MÉTRICA WINDOWS

```
{
  "_index": ".ds-metrics-windows.perfmon-default-2026.01.28-000001",
  "_id": "NaisBpwBHTTPT16MZ50O",
  "_version": 1,
  "_source": {
    "@timestamp": "2026-01-28T22:14:37.002Z",
    "agent": {
      "id": "032bfb48-14a8-4146-9ef3-b68cf8f81007",
      "name": "Windows10",
      "type": "metricbeat",
      "version": "9.2.4"
    },
    "service": {
      "type": "windows"
    },
    "data_stream": {
      "type": "metrics",
      "dataset": "windows.perfmon",
      "namespace": "default"
    },
    "host": {
      "hostname": "Windows10",
```

```
    "name": "windows10",
    "id": "b03073f0-8c78-41e8-81ed-eb67037c01c4",
    "ip": ["fe80::bdff:468e:7a05:ff17", "192.168.100.103"],
    "mac": ["08-00-27-09-7D-E3"],
    "architecture": "x86_64",
    "os": {
      "name": "Windows 10 Home",
      "version": "10.0",
      "build": "19045.5247",
      "kernel": "10.0.19041.5247 (WinBuild.160101.0800)",
      "family": "windows",
      "platform": "windows"
    }
  },
  "metricset": {
    "name": "perfmon",
    "period": 10000
  },
  "event": {
    "duration": 1056261200,
    "module": "windows",
    "dataset": "windows.perfmon"
  },
  "windows": {
    "perfmon": {
      "object": "Process",
      "instance": "svchost",
      "metrics": {
        "working_set": 14532608
      }
```

```
    }
   }
  }
}
```

FIN DEL BLOQUE JSON

EXPLICACIÓN DE CAMPOS PRINCIPALES:

Campo	Descripción
_index	Índice específico para métricas de Windows Performance Monitor
@timestamp	Marca temporal de cuando se capturó la métrica (2026-01-28T22:14:37.002Z)
agent.type	Tipo de agente utilizado (metricbeat - especializado en métricas de rendimiento)
data_stream.type	Tipo de flujo de datos (metrics = métricas de rendimiento, no logs)
data_stream.dataset	Dataset específico (windows.perfmon = Windows Performance Monitor)
host.name	Nombre del equipo Windows (windows10)
host.ip	Direcciones IP: fe80::bdff:468e:7a05:ff17 (IPv6) y 192.168.100.103 (IPv4 en red LAN)
host.os.name	Sistema operativo completo (Windows 10 Home)
host.os.build	Build específico de Windows (19045.5247)
metricset.name	Nombre del conjunto de métricas (perfmon)
metricset.period	Intervalo de recolección en milisegundos (10000 = cada 10 segundos)
event.duration	Duración de la recolección de la métrica en nanosegundos (1056261200 ns ≈ 1.056 segundos)

windows.perfmon.object	Objeto de rendimiento monitorizado (Process = procesos del sistema)
windows.perfmon.instance	Instancia específica del objeto (svchost = proceso de servicios de Windows)
windows.perfmon.metrics.working_set	Memoria utilizada por el proceso en bytes (14532608 bytes ≈ 13.86 MB)

PUNTOS CLAVE:

- La IP **192.168.100.103** confirma que el sistema Windows está en la red **LAN**
- Las métricas se recolectan cada 10 segundos de forma automática
- Se monitoriza el uso de memoria del proceso **svchost** (servicio crítico de Windows)
- Elastic Agent 9.2.4 con módulo metricbeat gestiona la recolección
- El build 19045.5247 indica una versión actualizada de Windows 10 Home

10. Conclusiones

La práctica ha permitido implementar con éxito una infraestructura completa de monitorización y análisis de eventos de seguridad, cumpliendo todos los objetivos planteados.

10.1. Objetivos Cumplidos

- Infraestructura de red segmentada con pfSense correctamente configurada
- Reglas de firewall implementadas según los requisitos de aislamiento
- Honeypot Cowrie operativo capturando intentos de intrusión SSH
- Suricata IDS/IPS detectando y registrando tráfico de red
- Sistema Windows monitoreado con métricas de rendimiento
- SIEM centralizado recibiendo logs de todas las fuentes
- Logs correctamente identificados por segmento de red (IPs correctas)
- Visualización y análisis de eventos en Elastic Cloud

10.2. Aprendizajes Clave

Durante la realización de esta práctica se han adquirido las siguientes competencias:

- Diseño de arquitecturas de red segmentadas con múltiples DMZs
- Configuración avanzada de pfSense (interfaces, alias, reglas de firewall)
- Implementación de principios de defensa en profundidad
- Despliegue y configuración de honeypots para inteligencia de amenazas
- Implementación de sistemas IDS/IPS con Suricata
- Integración de múltiples fuentes de logs en un SIEM centralizado
- Gestión de políticas de agentes en Elastic Fleet
- Análisis de logs en formato JSON para correlación de eventos

10.3. Mejoras Futuras

Posibles extensiones de esta práctica incluirían:

- Implementar reglas de Suricata más específicas para detectar ataques reales
- Configurar alertas automáticas en Elastic para eventos críticos
- Añadir más honeypots de diferentes tipos (web, base de datos, etc.)
- Implementar correlación avanzada de eventos entre fuentes
- Crear dashboards personalizados para análisis de seguridad
- Configurar respuestas automáticas ante detección de amenazas
- Integrar threat intelligence feeds para enriquecer los eventos

10.4. Consideraciones de Seguridad

Es importante destacar las siguientes consideraciones de seguridad:

- El honeypot está correctamente aislado de las redes internas
- Las credenciales SSH del honeypot son ficticias y no comprometen sistemas reales
- Suricata está en modo monitor (no bloquea tráfico activamente)
- Los logs contienen información sensible y deben protegerse adecuadamente
- Elastic Cloud utiliza conexiones cifradas para la transmisión de logs
- Las reglas de firewall siguen el principio de mínimo privilegio

Anexos

Anexo A: Direccionamiento IP

Red	Rango	Gateway
WAN	192.168.0.0/24	192.168.0.84 (pfSense)
LAN	192.168.100.0/24	192.168.100.1 (pfSense)
DMZ	192.168.200.0/24	192.168.200.1 (pfSense)
DMZ2	192.168.250.0/24	192.168.250.1 (pfSense)

Anexo B: Versiones de Software

Software	Versión
pfSense	2.7.2-RELEASE (amd64)
Elastic Stack	Cloud (última versión disponible enero 2026)
Elastic Agent	9.2.4
Cowrie	Docker image cowrie/cowrie:latest
Suricata	6.x (integración v2.27.0 en Elastic)
Windows 10 Home	Build 19045.5247
Kali Linux	2025.4 con kernel 6.18.5+kali-amd64

Anexo C: Referencias y Documentación

- pfSense Official Documentation: https://docs.netgate.com/pfsense/
- Elastic Security Documentation: https://www.elastic.co/guide/en/security/current/
- Cowrie Honeypot GitHub: https://github.com/cowrie/cowrie
- Suricata Documentation: https://docs.suricata.io/
- Elastic Agent Documentation: https://www.elastic.co/guide/en/fleet/current/