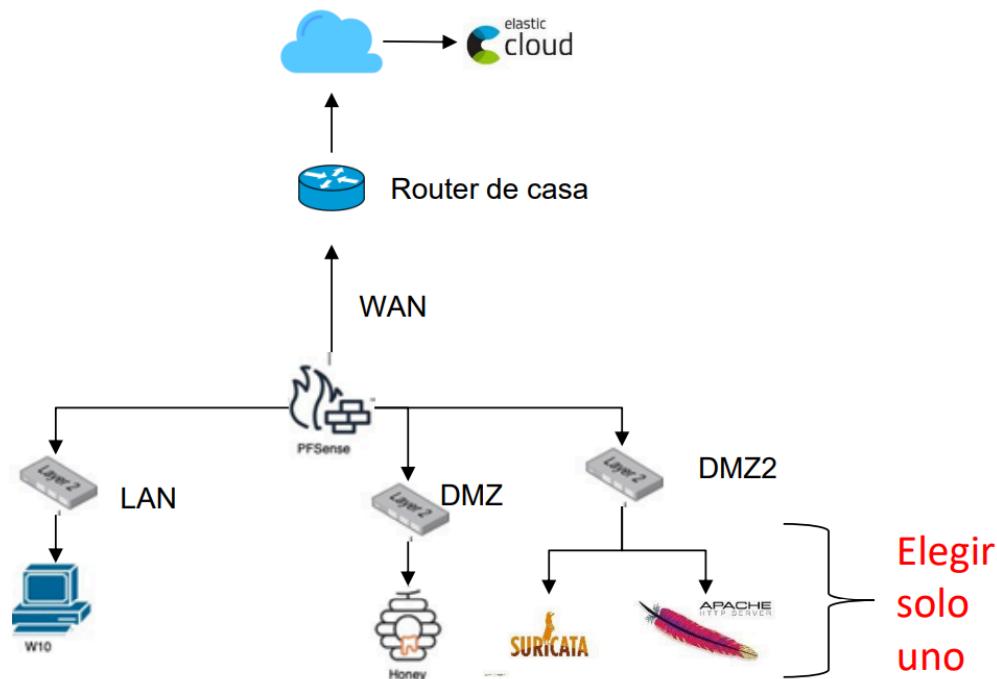


Enunciado

Queremos montar la siguiente infraestructura:



Los requisitos que debe cumplir son los siguientes:

1. Debe tener un PfSense en el que se interconecten las redes LAN, DMZ y DMZ2.
2. En la red LAN debe haber un equipo Windows que envíe logs al servidor de Elastic.
3. En la red DMZ debe haber un honeypot que envíe los logs al servidor de Elastic.
4. Este honeypot no debe tener acceso a ninguna red interna (LAN y DMZ2), ni debe poder accederse desde ellas. Sí debe ser accesible desde el exterior (red WAN/máquina host) en ambos sentidos.
5. En la red DMZ2 debe haber otra fuente diferente de logs a las dos mencionadas anteriormente. Se propone Suricata o Apache Server como posibles fuentes, pero se deja a elección del alumno.
6. El servidor de Elastic debe recibir, almacenar y poder visualizar los logs del honeypot, el W10 y la fuente elegida ubicada en la DMZ2. WAN Router de casa DMZ2 Elegir solo uno Criterios de evaluación de la memoria:

Criterios de evaluación de la memoria:

1. Debe contener evidencias y explicaciones que demuestren la correcta creación de la infraestructura de red en el PfSense.

2. Debe contener explicación y captura de las reglas de firewall elegidas para cada red (WAN, NAT, LAN, DMZ y DMZ2)
3. Debe contener evidencias de las políticas e integraciones asignadas a cada agente del SIEM (Elastic)
4. Debe contener evidencias que demuestren la correcta recepción de los logs, de todas las fuentes especificadas en el enunciado, en el SIEM (Elastic).

Adicionalmente, esos logs deben estar generados según el esquema que se especifica en el enunciado. Es decir, los logs del honeypot deben ser generados en una máquina con una IP perteneciente a la red DMZ, los de Windows con una perteneciente a la red LAN y los de Suricata/Apache a una perteneciente a la red DMZ2.

5. Los logs seleccionados deben estar debidamente explicados, es decir, debe quedar claro a qué elemento corresponde el log y que significan cada uno de los campos incluido en él.

1. Red WAN (Exterior)

La red WAN representa el exterior de la infraestructura y constituye la interfaz de conexión con el router doméstico y, por extensión, con Internet. En el contexto de esta práctica, la WAN simula una red no confiable desde la cual pueden provenir ataques externos dirigidos hacia los servicios expuestos intencionalmente.

A diferencia de las redes internas (LAN, DMZ y DMZ2), la WAN no es una red bajo control directo del alumno, sino que depende del router doméstico que asigna direccionamiento mediante DHCP. El firewall pfSense actúa como barrera entre esta red hostil y las redes internas, permitiendo únicamente el tráfico explícitamente autorizado hacia servicios específicos que deben ser accesibles desde el exterior.

1.1 Configuración de la interfaz WAN en pfSense

La interfaz WAN de pfSense está configurada en modo DHCP, lo que significa que recibe automáticamente su dirección IP del router doméstico. Esta configuración es adecuada para entornos de laboratorio donde el direccionamiento externo no está bajo control del alumno y puede variar entre sesiones.

En el momento de la documentación, la configuración de la interfaz WAN presenta los siguientes parámetros:

- **IPv4 Configuration Type:** DHCP
- **IPv4 Address:** 192.168.1.227 (asignada por el router doméstico)
- **Subnet mask:** 255.255.255.0

- **Gateway IPv4:** 192.168.1.1 (router doméstico)
- **DNS servers:** 192.168.1.1
- **Status:** up (interfaz activa y funcional)

El gateway apunta al router doméstico (192.168.1.1), que es el responsable de enrutar el tráfico hacia Internet y de proporcionar servicios de DNS para la resolución de nombres. La interfaz se encuentra operativa y ha procesado tráfico entrante y saliente, como se refleja en las estadísticas de paquetes.

WAN Interface (wan, em0)	
Status	up
DHCP	up Release WAN <input type="checkbox"/> Relinquish Lease
MAC Address	08:00:27:a4:3f:28
IPv4 Address	192.168.1.227
Subnet mask IPv4	255.255.255.0
Gateway IPv4	192.168.1.1
IPv6 Link Local	fe80::a00:27ff:fea4:3f28%em0
DNS servers	192.168.1.1
MTU	1500
Media	1000baseT <full-duplex>
In/out packets	25152/24611 (4.86 MiB/1.23 MiB)
In/out packets (pass)	25152/24611 (4.86 MiB/1.23 MiB)
In/out packets (block)	1067/0 (166 KiB/0 B)
In/out errors	609/0
Collisions	0
Interrupts	36598 (4/s)

Captura wan-1

1.2 Reglas de firewall en la interfaz WAN

La interfaz WAN cuenta con reglas de firewall diseñadas bajo el principio de "denegar por defecto, permitir explícitamente". Esto significa que, por defecto, pfSense bloquea todo el tráfico entrante desde la WAN, y únicamente se permiten conexiones hacia servicios que han sido explícitamente autorizados mediante reglas específicas.

Las reglas activas en la interfaz WAN son las siguientes:

Regla 1 - WAN → Cowrie SSH:

- Action: Pass
- Protocol: TCP
- Source: any (cualquier origen externo)
- Destination: WAN address, puerto 222
- Descripción: WAN -> Cowrie SSH

Esta regla permite que cualquier origen externo pueda conectarse al puerto 222 de la dirección WAN. Como se documentó en la sección de DMZ, existe una regla de Port Forward que redirige el tráfico entrante en este puerto hacia el honeypot Cowrie ubicado en la DMZ (192.168.200.98:222). Esta es la única entrada permitida desde la WAN hacia la infraestructura interna, y su propósito es exponer intencionalmente el honeypot para capturar intentos de intrusión y generar logs de actividad maliciosa.

Regla implícita de bloqueo por defecto:

Aunque no aparece explícitamente en la lista de reglas visibles, pfSense aplica una política de bloqueo por defecto al final de la cadena de reglas. Esto significa que cualquier tráfico que no coincida con las reglas de permiso anteriores será bloqueado automáticamente. Esta política es fundamental para la seguridad de la infraestructura, ya que impide accesos no autorizados desde redes externas.

Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	*	*	WAN address	222	*	none		WAN > Cowrie SSH	
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	*	*	192.168.200.99	80 (HTTP)	*	none		NAT Servidor web Apache	
<input type="checkbox"/>	✓ 0/15 KiB	IPv4 TCP	*	*	192.168.200.98	222	*	none		NAT	

Captura wan-2

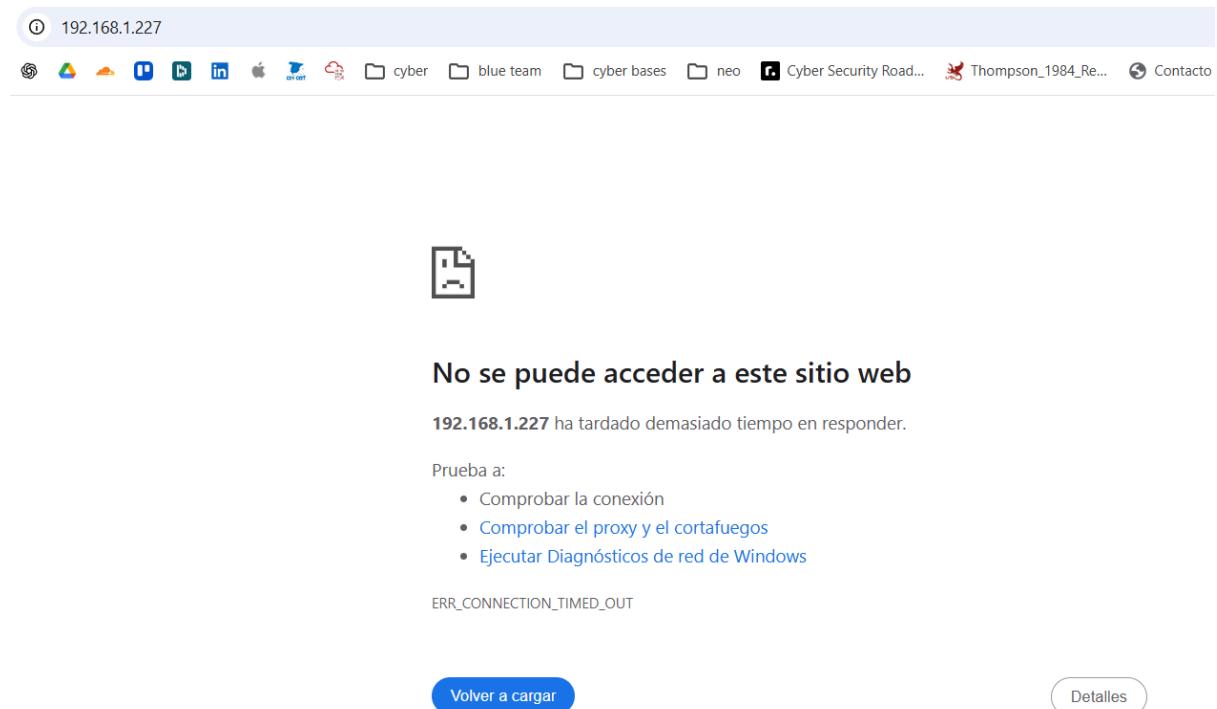
1.3 Protección del acceso administrativo - Anti-lockout

Una configuración crítica de seguridad en pfSense es el mecanismo de "anti-lockout", que garantiza que el acceso a la interfaz de administración web (WebConfigurator) esté disponible únicamente desde la red LAN, impidiendo que administradores o atacantes puedan acceder a la GUI desde la WAN.

En la configuración del sistema, la opción "Disable webConfigurator anti-lockout rule" **no está marcada**, lo que significa que el anti-lockout está **activo**. Esta protección asegura que siempre exista una regla implícita en la interfaz LAN que permite el acceso administrativo al firewall, mientras que desde la WAN este acceso permanece bloqueado.

Para verificar esta protección, se realizó un intento de acceso a la GUI de pfSense desde un equipo ubicado en la red WAN (192.168.1.249), utilizando la dirección IP de la interfaz WAN (<https://192.168.1.227>). El resultado fue un timeout de conexión (ERR_CONNECTION_TIMED_OUT), confirmando que el firewall bloquea correctamente cualquier intento de acceso administrativo desde redes externas.

Esta medida de seguridad es esencial para prevenir ataques dirigidos a la interfaz de administración del firewall, que podría ser objetivo de fuerza bruta, explotación de vulnerabilidades o intentos de acceso no autorizado.



Captura wan-3

1.4 Verificación de acceso al honeypot desde WAN

Una vez confirmado que el acceso administrativo está bloqueado desde la WAN, se verificó que el único servicio expuesto —el honeypot SSH en el puerto 222— es correctamente accesible desde el exterior. Esta verificación es fundamental para cumplir el requisito del enunciado de que el honeypot debe ser accesible desde la red WAN en ambos sentidos.

Se realizó una prueba de conectividad TCP desde un equipo Windows ubicado en la red WAN (192.168.1.249) hacia la dirección WAN de pfSense (192.168.1.227) en el puerto 222, utilizando el comando:

```
Test-NetConnection -ComputerName 192.168.1.227 -Port 222
```

El resultado de la prueba fue exitoso, con el parámetro **TcpTestSucceeded: True**, confirmando que:

1. El puerto 222 es accesible desde la WAN
2. La regla de firewall "WAN -> Cowrie SSH" está funcionando correctamente
3. El Port Forward redirige el tráfico hacia el honeypot en la DMZ
4. La infraestructura cumple el requisito de exponer únicamente el honeypot hacia el exterior

Esta prueba demuestra que la segmentación de red está correctamente implementada: desde la WAN solo es posible alcanzar el honeypot (servicio intencionalmente expuesto), mientras que el resto de la infraestructura interna (LAN, DMZ2 y la propia administración de pfSense) permanece inaccesible desde el exterior.

```
PS C:\Users\miche> Test-NetConnection -ComputerName 192.168.1.227 -Port 222

ComputerName      : 192.168.1.227
RemoteAddress     : 192.168.1.227
RemotePort        : 222
InterfaceAlias    : Wi-Fi
SourceAddress     : 192.168.1.249
TcpTestSucceeded  : True
```

Captura wan-4

1.5 Conclusión de la configuración de WAN

Con estos ajustes y verificaciones, la red WAN cumple su función en la arquitectura de seguridad de la práctica:

- La interfaz WAN está operativa y conectada al router doméstico mediante DHCP
- Las reglas de firewall implementan el principio de mínimo privilegio: todo bloqueado por defecto, solo se permite el acceso al honeypot
- El acceso administrativo a pfSense está protegido y es inaccesible desde redes externas
- El honeypot es accesible desde la WAN, cumpliendo el requisito del enunciado de exposición bidireccional
- La segmentación entre WAN y redes internas está correctamente implementada

La red WAN actúa como punto de entrada controlado para atacantes simulados, permitiendo la captura de actividad maliciosa en el honeypot sin comprometer la seguridad de las redes internas (LAN y DMZ2).

2. Red LAN (Endpoint Windows)

La red LAN corresponde a la red interna de la infraestructura y representa el entorno más protegido del escenario. Según el enunciado de la práctica, su función principal es alojar un equipo Windows 10 que actúa como endpoint interno y como fuente de generación de logs, los cuales deben ser enviados al servidor SIEM (Elastic) para su almacenamiento y análisis posterior.

La red LAN no debe ser accesible desde redes externas ni desde otras redes internas como la DMZ o la DMZ2. Su diseño responde a un modelo clásico de segmentación, en el que la LAN inicia comunicaciones hacia el exterior cuando es necesario, pero no expone servicios ni acepta conexiones entrantes desde otras zonas de la red.

2.1 Configuración de red del sistema Windows 10

Como primer ajuste dentro de la red LAN, se configuró el equipo Windows 10 con una dirección IP fija perteneciente a la red 192.168.100.0/24. Este cambio se realizó directamente en el sistema operativo, sustituyendo la asignación dinámica por DHCP.

La asignación de una IP fija responde a una buena práctica recomendada en entornos de laboratorio y evaluación, ya que permite identificar de forma inequívoca el origen de los logs generados por el sistema Windows en el SIEM. De este modo, se evita que un cambio dinámico de dirección IP pueda introducir confusión durante el análisis de eventos o comprometer la claridad de las evidencias.

La configuración de red del equipo Windows 10 quedó establecida con los siguientes parámetros:

- **Dirección IPv4:** 192.168.100.50
- **Máscara de subred:** 255.255.255.0
- **Puerta de enlace predeterminada:** 192.168.100.1 (pfSense LAN)
- **Servidor DNS:** 192.168.100.1

Tras la configuración de la IP fija, se verificó la conectividad con la puerta de enlace de la red LAN mediante ping, confirmando que el equipo mantiene comunicación correcta con el firewall (0% de pérdida de paquetes). Asimismo, se verificó la salida a Internet mediante ping a 8.8.8.8, confirmando que el direccionamiento y el enrutamiento funcionan correctamente.

Captura Ian-1

2.2 Reglas de firewall aplicadas a la red LAN

Una vez validado el direccionamiento, se revisaron las reglas de firewall asociadas a la interfaz LAN en pfSense. La configuración de firewall para LAN sigue el principio de confianza en la red interna: se permite el tráfico saliente sin restricciones, pero no se aceptan conexiones entrantes desde otras redes.

Las reglas activas en la interfaz LAN son las siguientes:

Regla 1 - Anti-Lockout Rule:

Esta regla especial garantiza el acceso administrativo al firewall pfSense desde la red LAN. Permite conexiones al puerto 443 (HTTPS) y 80 (HTTP) hacia la dirección IP de la interfaz

LAN (192.168.100.1), asegurando que el administrador siempre pueda acceder a la GUI de configuración desde la red interna, incluso si otras reglas están mal configuradas.

Regla 2 - Default allow LAN to any rule:

- Action: Pass
- Protocol: IPv4 any
- Source: LAN subnets
- Destination: any

Esta regla permite que cualquier equipo en la red LAN inicie conexiones hacia cualquier destino, ya sea Internet, otras redes internas o el propio pfSense. Esta política es coherente con el diseño de seguridad: la LAN es la red más confiable y sus equipos deben poder comunicarse libremente hacia el exterior para actualizar software, acceder a servicios externos y enviar logs al servidor Elastic.

Regla 3 - Default allow LAN IPv6 to any rule:

Regla equivalente a la anterior pero para tráfico IPv6. Esta regla está deshabilitada en la práctica actual, ya que el ejercicio se basa exclusivamente en direccionamiento IPv4. Esta decisión permite simplificar el escenario, evitar rutas de tráfico no documentadas y mantener la coherencia con el alcance definido.

El enunciado de la práctica no exige limitar la salida de tráfico desde la red LAN, sino impedir accesos entrantes hacia ella desde otras redes. La protección de la LAN se implementa mediante las reglas de bloqueo configuradas en las interfaces DMZ y DMZ2, que impiden explícitamente cualquier conexión hacia la red 192.168.100.0/24.

Rules (Drag to Change Order)										
States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓ 7/2.56 MiB	*	*	*	LAN Address	443	*	*		Anti-Lockout Rule	
✓ 0/3 Kib	IPv4	*	LAN subnets	*	*	*	*	none	Default allow LAN to any rule	
✓ 0/0 B	IPv6	*	LAN subnets	*	*	*	*	none	Default allow LAN IPv6 to any rule	

Captura Ian-2

2.3 Verificación de la segmentación de red

Para confirmar que la segmentación está correctamente implementada, se realizaron pruebas de conectividad desde las redes DMZ y DMZ2 hacia el equipo Windows 10 en la red LAN (192.168.100.50).

Prueba 1 - Bloqueo desde DMZ:

Desde la máquina KALI-HONEY ubicada en la DMZ (192.168.200.98), se ejecutó:

```
ping -c 2 192.168.100.50
```

Resultado: 100% packet loss. La DMZ no puede alcanzar la red LAN, confirmando el aislamiento entre ambas zonas.

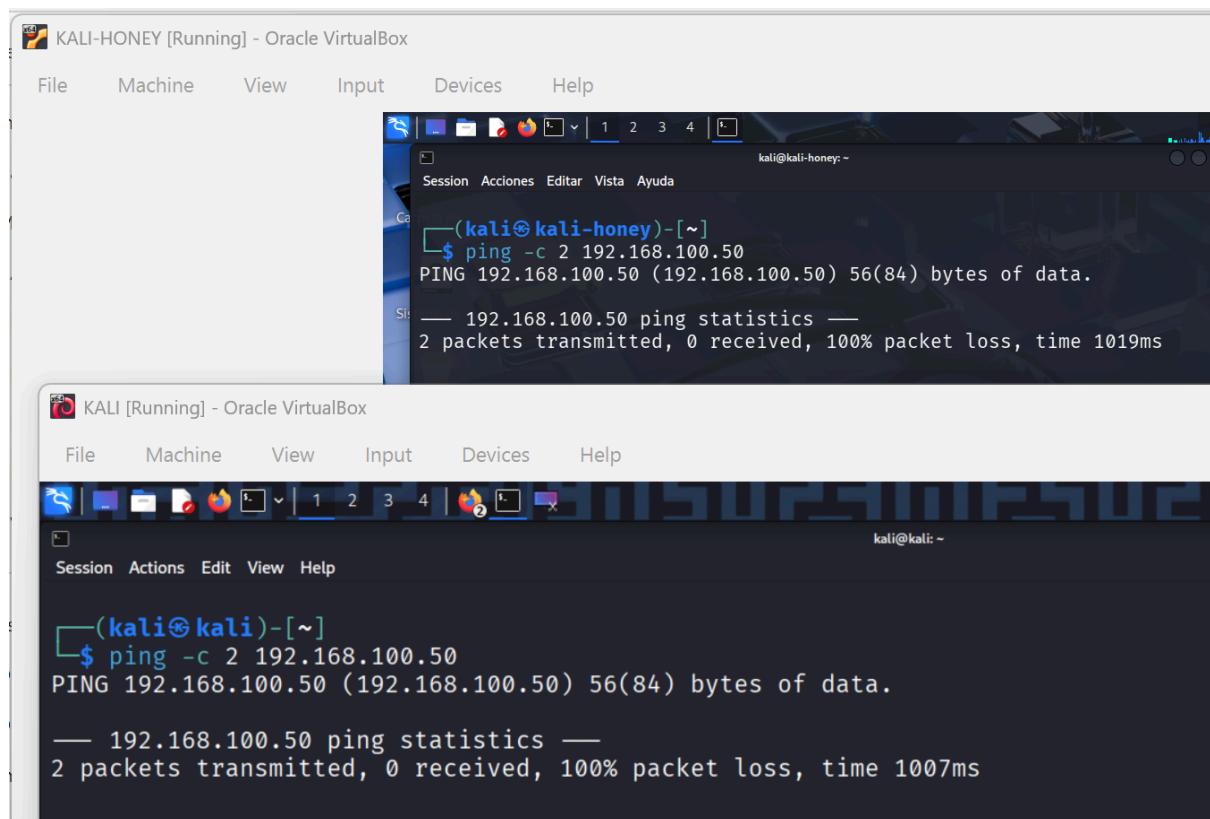
Prueba 2 - Bloqueo desde DMZ2:

Desde la máquina Kali-Suricata ubicada en la DMZ2 (192.168.250.100), se ejecutó:

```
ping -c 2 192.168.100.50
```

Resultado: 100% packet loss. La DMZ2 no puede alcanzar la red LAN, confirmando el aislamiento entre ambas zonas.

Estas pruebas demuestran que las reglas de firewall configuradas en las interfaces DMZ y DMZ2 (que bloquean explícitamente el tráfico hacia 192.168.100.0/24) están funcionando correctamente. La red LAN permanece protegida de accesos no autorizados desde las zonas desmilitarizadas, cumpliendo el requisito de segmentación del diseño de seguridad.



```
(kali㉿kali-honey)-[~]
$ ping -c 2 192.168.100.50
PING 192.168.100.50 (192.168.100.50) 56(84) bytes of data.
— 192.168.100.50 ping statistics —
2 packets transmitted, 0 received, 100% packet loss, time 1019ms

(kali㉿kali)-[~]
$ ping -c 2 192.168.100.50
PING 192.168.100.50 (192.168.100.50) 56(84) bytes of data.
— 192.168.100.50 ping statistics —
2 packets transmitted, 0 received, 100% packet loss, time 1007ms
```

Captura Ian-3

2.4 Configuración de NAT para LAN

La red LAN utiliza NAT de salida (Outbound NAT) en modo automático para traducir las direcciones privadas de la red 192.168.100.0/24 a la dirección pública de la interfaz WAN al comunicarse con Internet.

Como se documentó en la sección de configuración general de NAT, pfSense genera automáticamente las reglas necesarias para todas las redes internas definidas en el sistema. La red LAN (192.168.100.0/24) está incluida en estas reglas automáticas, realizando NAT hacia la dirección de la interfaz WAN (192.168.1.227), lo que permite la salida a Internet desde todos los equipos de la red LAN.

Esta configuración es coherente con el diseño de la práctica: la red LAN debe poder comunicarse con servicios externos (actualizaciones, DNS, y futuro envío de logs a Elastic) sin restricciones de salida.

2.5 Conclusión de la configuración de LAN

Con estos ajustes y verificaciones, la red LAN queda correctamente configurada según los requisitos de la práctica:

- El equipo Windows 10 cuenta con dirección IP fija (192.168.100.50) para identificación inequívoca en los logs
- La conectividad básica está verificada: acceso al gateway local y salida a Internet funcionan correctamente
- Las reglas de firewall permiten salida sin restricciones desde la LAN, pero no aceptan conexiones entrantes desde otras zonas
- La segmentación está correctamente implementada: DMZ y DMZ2 no pueden acceder a la red LAN
- El NAT de salida permite la comunicación con servicios externos, necesaria para el envío futuro de logs a Elastic

La red LAN está operativa y preparada para cumplir su función como fuente de logs de un endpoint Windows dentro de la arquitectura segmentada de la práctica. El siguiente paso será instalar y configurar el agente de envío de logs desde Windows hacia el servidor Elastic una vez este último esté desplegado.

3. Red DMZ (Honeypot Cowrie)

La red DMZ constituye la zona expuesta de la infraestructura y está diseñada para alojar sistemas accesibles desde el exterior de forma controlada. Según el enunciado de la práctica, en esta red debe ubicarse un honeypot que actúe como punto de interacción con la red WAN y como fuente de generación de logs para el sistema SIEM (Elastic).

La DMZ se encuentra aislada de la red LAN y de la red DMZ2, de modo que cualquier acceso o actividad maliciosa dirigida al honeypot quede contenida en esta zona y no

comprometa otras redes internas. Este enfoque permite simular escenarios reales de exposición manteniendo una segmentación clara y verificable.

3.1 Configuración de red del honeypot en la DMZ

Antes de desplegar los servicios de honeypot, se configuró el sistema destinado a esta función con una **dirección IP fija dentro de la red DMZ (192.168.200.0/24)**. El honeypot, basado en Kali Linux, se conectó a la interfaz DMZ de pfSense y se configuró con la dirección IP **192.168.200.98**, situada fuera del rango DHCP definido para esta red.

La asignación de una IP fija permite garantizar que los logs generados por el honeypot tengan un origen estable y claramente identificable en el SIEM, facilitando su análisis y la correcta interpretación de las evidencias durante la evaluación de la práctica. Tras aplicar la configuración, se verificó el estado de la interfaz de red, confirmando que el sistema utiliza exclusivamente la dirección IP fija asignada y que la comunicación se realiza a través de la puerta de enlace de la DMZ.

Con esta configuración, el sistema queda correctamente preparado para el despliegue del honeypot y para la aplicación posterior de las reglas de firewall y accesos controlados exigidos por el enunciado.

```
(kali㉿kali-honey) [~]
$ ip a
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 noprefixroute
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:28:bb:bb brd ff:ff:ff:ff:ff:ff
        inet 192.168.200.98/24 brd 192.168.200.255 scope global noprefixroute eth0
            valid_lft forever preferred_lft forever
        inet6 fe80::f2dc:120:f22d:6eef/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
```

3.2 Reglas de Firewall y NAT para habilitar salida a Internet desde la DMZ

Objetivo

Habilitar **salida a Internet desde la red DMZ** (donde está el honeypot) para permitir:

- instalación de dependencias (p. ej., Docker y paquetes del sistema)
 - envío de logs a Elastic
- Manteniendo la segmentación: **sin acceso desde/hacia LAN y DMZ2** salvo lo estrictamente definido.

Regla de Firewall en DMZ (permitir tráfico saliente)

Qué se hizo

Se creó una regla en **Firewall → Rules → DMZ** que permite el tráfico **saliente** desde la DMZ hacia cualquier destino.

Por qué es necesario

Sin una regla “Pass” en la interfaz DMZ, pfSense **bloquea por defecto** el tráfico que entra desde esa red. Esto impide incluso alcanzar el gateway externo y hace imposible cualquier actualización/installación o envío de logs.

Configuración (resumen técnico)

- Action: Pass
- Interface: DMZ
- Address Family: IPv4
- Protocol: any
- Source: DMZ subnet
- Destination: any

The screenshot shows the pfSense Firewall Rules configuration. The top navigation bar has tabs for Floating, WAN, LAN, DMZ (which is highlighted in red), and DMZ2. Below the tabs is a table titled "Rules (Drag to Change Order)". The table columns are: States, Protocol, Source, Port, Destination, Port, Gateway, Queue, Schedule, Description, and Actions. There are four rows in the table:

- Row 1: States 0/1008 B, Protocol IPv4 *, Source DMZ subnets, Port *, Destination *, Port *, Gateway *, Queue none, Schedule none, Description DMZ -> any (salida), Actions (edit, delete, copy, save).
- Row 2: States 0/0 B, Protocol IPv4 ICMP any, Source *, Destination *, Port *, Gateway *, Queue none, Schedule none, Description Regla tráfico ICMP, Actions (edit, delete, copy, save).
- Row 3: States 0/0 B, Protocol IPv4 UDP, Source *, Destination *, Port 53 (DNS), Gateway *, Queue none, Schedule none, Description Regla DNS, Actions (edit, delete, copy, save).
- Row 4: States 0/0 B, Protocol IPv4 TCP, Source *, Destination *, Port webs, Gateway *, Queue none, Schedule none, Description Regla tráfico web, Actions (edit, delete, copy, save).

At the bottom of the table are buttons for Add, Delete, Toggle, Copy, Save, and Separator.

Captura 1 (evidencia)

Pantalla de **Firewall → Rules → DMZ** mostrando la regla “Pass” creada.

Outbound NAT para DMZ (traducción de direcciones)

Qué se hizo

Se habilitó NAT de salida para la DMZ en **Firewall → NAT → Outbound**:

- Se cambió el modo a **Hybrid Outbound NAT**
- Se añadió una regla que realiza NAT del tráfico con origen DMZ hacia la **WAN address** de pfSense.

Por qué es necesario

Aunque el firewall permita el tráfico, sin Outbound NAT los paquetes salen con IP privada (DMZ) y **no pueden recibir respuesta desde Internet**. El NAT es imprescindible para que la DMZ navegue/actualice y pueda comunicarse con servicios externos.

Configuración (resumen técnico)

- Mode: Hybrid Outbound NAT
- Interface: WAN
- Source: DMZ subnet
- Translation / Address: WAN address

Interface	Source	Source Port	Destination	Destination Port	NAT Address	NAT Port	Static Port	Description	Actions
WAN	DMZ subnets	*	*	*	WAN address	*			<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Pantalla de **Firewall → NAT → Outbound** mostrando:

- el modo **Hybrid**
- la regla de NAT para **DMZ subnet → WAN address**

Verificación de conectividad (prueba desde KALI-HONEY)

Qué se hizo

Se validó la conectividad desde el host en DMZ (KALI-HONEY) con ICMP:

1. Comprobación de salida hacia el gateway WAN (router):

- `ping -c 3 192.168.1.1`

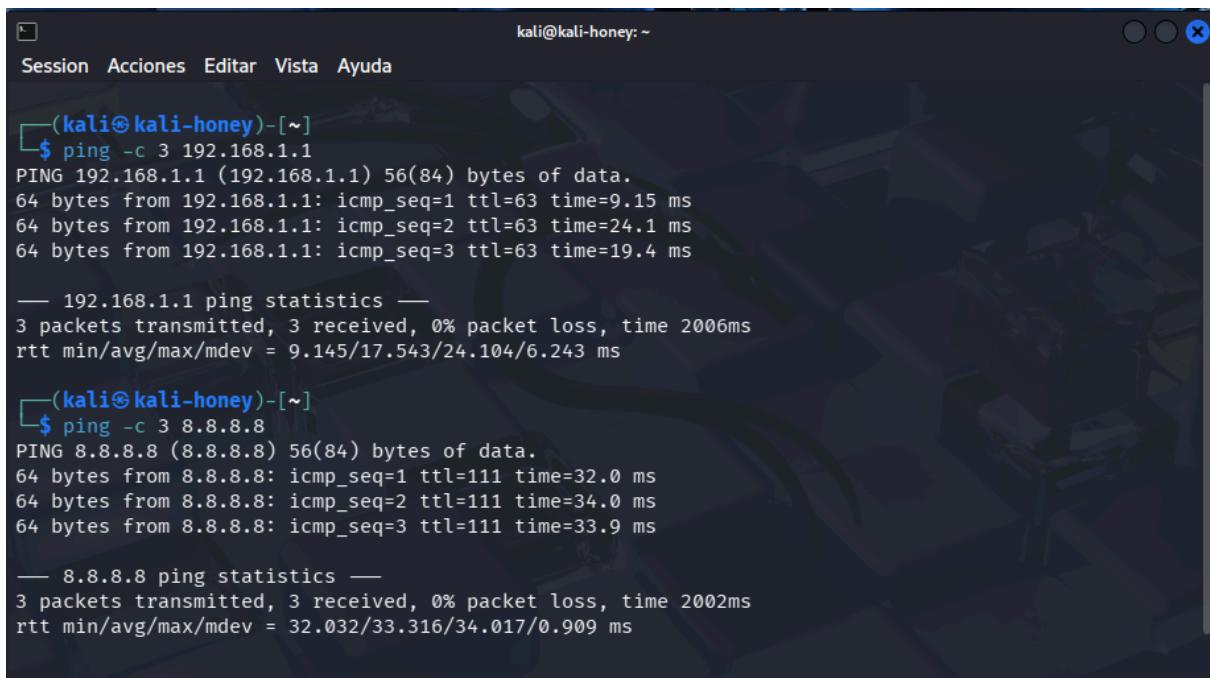
2. Comprobación de salida real a Internet:

- `ping -c 3 8.8.8.8`

Por qué esta verificación es válida

Estas pruebas demuestran que:

- la regla de **Firewall** permite el tráfico desde la DMZ
- el **NAT de salida** traduce correctamente y se reciben respuestas
- la DMZ tiene conectividad funcional para instalación y envío de logs



```
kali@kali-honey: ~
Session  Acciones  Editar  Vista  Ayuda

[(kali㉿kali-honey)-[~]
$ ping -c 3 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=63 time=9.15 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=63 time=24.1 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=63 time=19.4 ms

--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 9.145/17.543/24.104/6.243 ms

[(kali㉿kali-honey)-[~]
$ ping -c 3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=111 time=32.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=111 time=34.0 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=111 time=33.9 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 32.032/33.316/34.017/0.909 ms
```

Captura 3 (evidencia)

Terminal de KALI-HONEY mostrando el `ping -c 3 8.8.8.8` con respuesta (0% packet loss).

3.3 Configuración de resolución DNS en el host de la DMZ

Contexto

Tras habilitar la salida a Internet desde la DMZ, se detectó que el host no resolvía nombres

de dominio. Aunque la conectividad IP funcionaba (ICMP), los servicios que dependen de DNS (como `apt`) fallaban.

Causa

El archivo `/etc/resolv.conf` del host en la DMZ estaba vacío, por lo que no había servidores DNS configurados.

Acción realizada

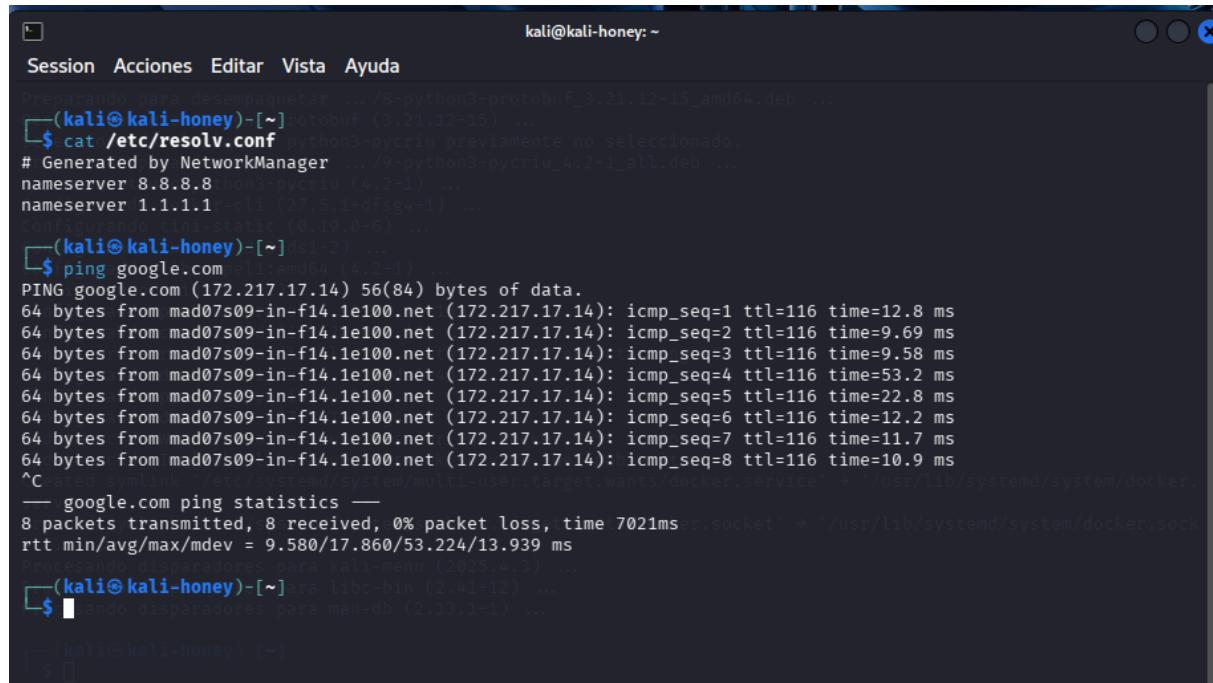
Se configuraron manualmente servidores DNS públicos en el propio host de la DMZ, sin modificar ajustes globales de pfSense:

- `8.8.8.8` (Google Public DNS)
- `1.1.1.1` (Cloudflare DNS)

Justificación

La configuración es local al host, mínima y suficiente para permitir resolución de nombres, manteniendo la segmentación de red y sin afectar a LAN ni DMZ2.

Evidencias



The screenshot shows a terminal window with the following session history:

```
kali@kali-honey: ~
Session  Acciones  Editar  Vista  Ayuda
└──(kali㉿kali-honey) [~] rotobuf (3.21.12-15) ...
$ cat /etc/resolv.conf
# Generated by NetworkManager .../9/python3-pycriu_4.2-1_all.deb ...
nameserver 8.8.8.8
nameserver 1.1.1.1
ping google.com (172.217.17.14) 56(84) bytes of data.
44 bytes from mad07s09-in-f14.1e100.net (172.217.17.14): icmp_seq=1 ttl=116 time=12.8 ms
44 bytes from mad07s09-in-f14.1e100.net (172.217.17.14): icmp_seq=2 ttl=116 time=9.69 ms
44 bytes from mad07s09-in-f14.1e100.net (172.217.17.14): icmp_seq=3 ttl=116 time=9.58 ms
44 bytes from mad07s09-in-f14.1e100.net (172.217.17.14): icmp_seq=4 ttl=116 time=53.2 ms
44 bytes from mad07s09-in-f14.1e100.net (172.217.17.14): icmp_seq=5 ttl=116 time=22.8 ms
44 bytes from mad07s09-in-f14.1e100.net (172.217.17.14): icmp_seq=6 ttl=116 time=12.2 ms
44 bytes from mad07s09-in-f14.1e100.net (172.217.17.14): icmp_seq=7 ttl=116 time=11.7 ms
44 bytes from mad07s09-in-f14.1e100.net (172.217.17.14): icmp_seq=8 ttl=116 time=10.9 ms
^C
--- google.com ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7021ms
rtt min/avg/max/mdev = 9.580/17.860/53.224/13.939 ms
Procesando disparadores para kali-menu (2025.4.3) ...
└──(kali㉿kali-honey) [~] jara libc-bin (2.41-12) ...
$ S
Siendo disparadores para man-db (2.13.1-1) ...
└──(kali㉿kali-honey) [~]
```

- Captura del contenido de `/etc/resolv.conf`.
- Captura de `ping http.kali.org` con respuesta.

3.4. NAT – Port Forward (DMZ)

En la red DMZ se ha configurado una regla de redirección de puertos (Port Forward) para exponer el honeypot hacia el exterior, cumpliendo el requisito del enunciado de que el honeypot debe ser accesible desde WAN.

Configuración de la regla – Honeypot (SSH)

- **Interfaz:** WAN
- **Protocolo:** TCP
- **Destino:** WAN address
- **Puerto destino:** 222
- **IP interna (Redirect target IP):** 192.168.200.98
- **Puerto interno (Redirect target port):** 222
- **Filter rule association:** Add associated filter rule

Esta regla redirige las conexiones SSH entrantes desde WAN (puerto 222) hacia el honeypot Cowrie en la DMZ (192.168.200.98:222), permitiendo que sea accesible desde el exterior sin comprometer las redes internas.

Resolución de incidencia técnica

Durante la configuración inicial, se detectó que el acceso desde WAN hacia el honeypot fallaba con timeout, a pesar de que la regla de Port Forward estaba creada. Tras el análisis mediante Packet Capture, se confirmó que los paquetes llegaban a pfSense pero no se procesaban correctamente.

La causa raíz fue que el parámetro "**Filter rule association**" estaba configurado como "**None**", lo que impedía la vinculación correcta entre la regla NAT y la regla de firewall en la interfaz WAN. Al cambiar este parámetro a "**Add associated filter rule**" y aplicar los cambios, el acceso desde WAN quedó funcional, verificándose mediante conexión SSH exitosa desde el host externo.

The screenshot shows the pfSense configuration interface for Port Forwarding. The top navigation bar includes links for Firewall, NAT, and Port Forward. Below the navigation, there are tabs for Port Forward, 1:1, Outbound, and NPT. The Port Forward tab is selected. The main area displays a table titled "Rules" with the following data:

	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions
<input type="checkbox"/>	WAN	TCP	*	*	WAN address	222	192.168.200.98	222		
<input type="checkbox"/>	WAN	TCP	*	*	WAN address	80 (HTTP)	192.168.200.99	80 (HTTP)	Servidor web Apache	

At the bottom of the table are several action buttons: Add, Add, Delete, Toggle, Save, and Separator.

Reglas de Firewall – Interfaz DMZ (estado actual)

En la interfaz **DMZ** se mantiene **una única regla activa**, destinada a permitir la **salida** desde la red DMZ. El resto de reglas que aparecen en la lista (ICMP, DNS y web) están **desactivadas**, por lo que **no aplican** en el filtrado actual.

Regla activa – DMZ → any (salida)

Se permite tráfico IPv4 desde **DMZ subnets** hacia **cualquier destino**. Esta regla asegura que la máquina del honeypot en DMZ pueda comunicarse hacia el exterior cuando sea necesario (por ejemplo, para conectividad básica y futuras integraciones), manteniendo el control desde pfSense.

Aislamiento DMZ

No existen reglas de permiso adicionales desde DMZ hacia **LAN** ni hacia **DMZ2**. Por tanto, cualquier tráfico no cubierto por la regla activa queda **bloqueado por la política por defecto** en pfSense, preservando el aislamiento de la DMZ frente a redes internas.

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
51/696 KiB	IPv4 *	DMZ subnets	*	*	*	*	none		DMZ-> any (salida)	
0/0 B	IPv4 ICMP any		*	*	*	*	none		Regla tráfico ICMP	
0/0 B	IPv4 UDP	*	*	*	53 (DNS)	*	none		Regla DNS	
0/0 B	IPv4 TCP	*	*	*	webs	*	none		Regla tráfico web	

Firewall – NAT Outbound

Modo de funcionamiento

El firewall pfSense se encuentra configurado en **Automatic Outbound NAT**, lo que permite que el propio sistema genere de forma automática las reglas necesarias para la traducción de direcciones de salida hacia la interfaz WAN. Este modo es el recomendado para la práctica y coincide con la configuración empleada por el profesor.

Redes incluidas en las reglas

Las reglas automáticas de Outbound NAT contemplan como origen las siguientes subredes definidas en la arquitectura del laboratorio:

- **LAN:** 192.168.100.0/24
- **DMZ:** 192.168.200.0/24
- **DMZ2:** 192.168.250.0/24

En todos los casos, la traducción se realiza hacia la **dirección de la interfaz WAN**, permitiendo la salida a Internet de las máquinas internas.

Coherencia con la práctica

Las direcciones IP incluidas en las reglas automáticas coinciden exactamente con las subredes solicitadas en el enunciado. No existen redes adicionales ni configuraciones inconsistentes, por lo que este apartado se considera correcto y alineado con la arquitectura definida.

Impacto en la seguridad

La configuración de Outbound NAT no afecta al tráfico entrante ni a las reglas de acceso desde la WAN hacia la DMZ. Su función se limita a permitir la conectividad de salida de las redes internas, manteniendo el aislamiento entre LAN, DMZ y DMZ2 según lo establecido en la práctica.

The screenshot shows the Winbox interface for Outbound NAT configuration. The 'Outbound' tab is active. In the 'Mode' section, 'Automatic outbound NAT rule generation' is selected. The 'Mappings' table shows a single entry: Interface WAN, Source DMZ subnets, Destination WAN address, and NAT Port *. Below the table are buttons for Add, Delete, Toggle, and Save. The 'Automatic Rules' section lists two rules:

Interface	Source	Source Port	Destination	Destination Port	NAT Address	NAT Port	Static Port	Description	Actions
WAN	DMZ subnets	*	*	*	WAN address	*	*		

Interface	Source	Source Port	Destination	Destination Port	NAT Address	NAT Port	Static Port	Description
WAN	127.0.0.0/8 ::1/128 192.168.100.0/24 192.168.200.0/24 192.168.250.0/24	*	*	500	WAN address	*	*	Auto created rule for ISAKMP
WAN	127.0.0.0/8 ::1/128 192.168.100.0/24 192.168.200.0/24 192.168.250.0/24	*	*	*	WAN address	*	*	Auto created rule

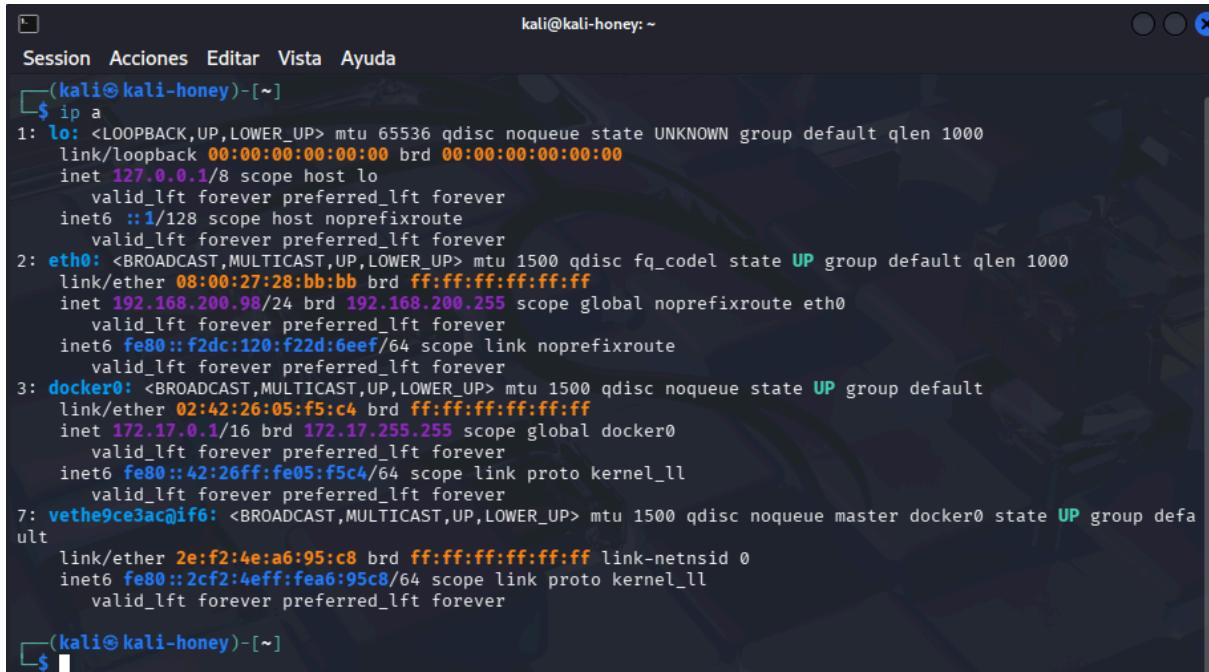
3.5 Despliegue del honeypot Cowrie en la red DMZ

En este apartado demostramos dos cosas: (1) que la máquina del honeypot está realmente en la DMZ (IP del rango DMZ) y (2) que el honeypot está operativo y generando logs. La generación de logs es la evidencia principal porque es lo que se integrará después en Elastic.

IP y rol en la arquitectura

La red DMZ se utiliza para alojar el honeypot (Cowrie) y exponerlo hacia el exterior (WAN)

sin permitir acceso hacia redes internas (LAN y DMZ2), cumpliendo el requisito de segmentación y aislamiento. El honeypot se ejecuta en la máquina **KALI-HONEY** con IP estática **192.168.200.98**, perteneciente a la subred **192.168.200.0/24**, con pfSense como puerta de enlace de la DMZ (**192.168.200.1**).



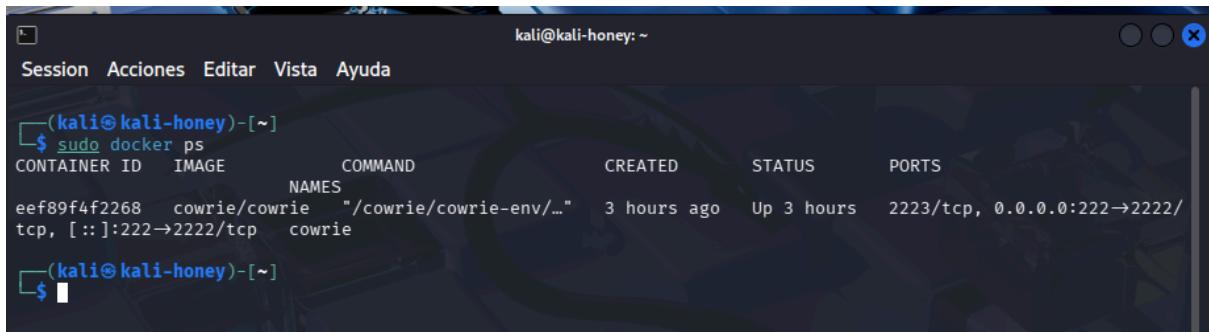
```
kali@kali-honey: ~
Session Acciones Editar Vista Ayuda
[(kali㉿kali-honey)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:28:bb:bb brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.98/24 brd 192.168.200.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::f2dc:120:f22d:6eef/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:26:05:f5:c4 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
    inet6 fe80::42:26ff:fe05:f5c4/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
7: vethe9ce3ac@if6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker0 state UP group defa
ult
    link/ether 2e:f2:4e:a6:95:c8 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::2cf2:4eff:fea6:95c8/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
[(kali㉿kali-honey)-[~]
$
```

Captura (Figura 3.3-1) — Evidencia de IP en DMZ

Inserta aquí una captura de pantalla de `ip a` en KALI-HONEY donde se vea claramente la interfaz activa y la IP **192.168.200.98/24**. Esta imagen es la evidencia de que el honeypot está ubicado en la red DMZ.

Estado del servicio (Cowrie activo)

Cowrie se despliega como contenedor Docker en KALI-HONEY. Para evidenciar que el honeypot está funcionando, se verifica que el contenedor está levantado y que el puerto de exposición está publicado en el host.



```
kali@kali-honey: ~
Session Acciones Editar Vista Ayuda
[(kali㉿kali-honey)-[~]
$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS      PORTS
NAMES
eef89f4f2268   cowrie/cowrie  "/cowrie/cowrie-env..."   3 hours ago   Up 3 hours   2223/tcp, 0.0.0.0:222->2222/tcp, [::]:222->2222/tcp
cowrie
[(kali㉿kali-honey)-[~]
$
```

Captura (Figura 3.3-2) — Contenedor activo

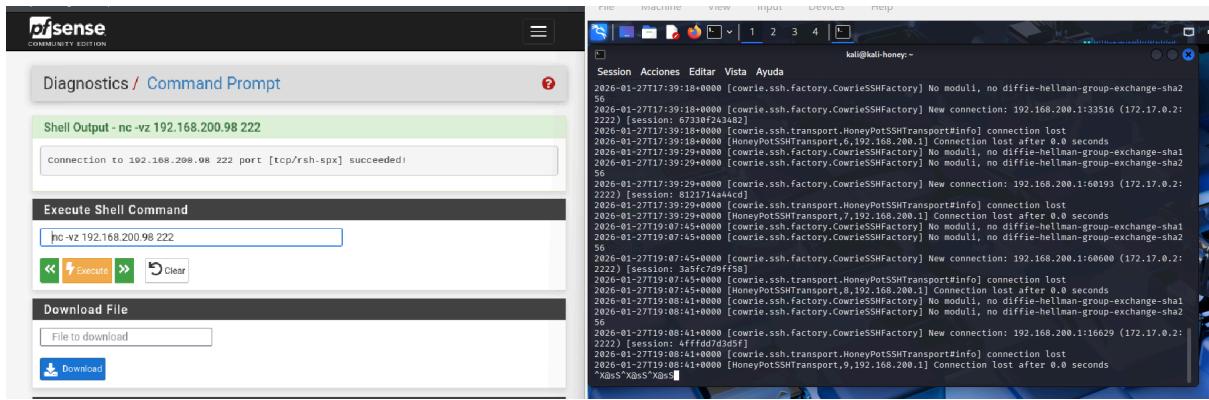
Inserta aquí la captura de `sudo docker ps` donde se vea el contenedor **cowrie** en estado **Up** y el mapeo de puertos (por ejemplo **0.0.0.0:222->2222/tcp**). Esta evidencia demuestra que el honeypot está operativo y listo para recibir conexiones y generar eventos.

Generación de logs (Evidencia operativa del honeypot)

La verificación más importante es demostrar que Cowrie genera logs al recibir intentos de conexión. Para ello se monitorizan los logs del contenedor y se provoca tráfico hacia el servicio SSH del honeypot desde una fuente dentro de la DMZ.

Prueba realizada

En KALI-HONEY se ejecuta la visualización de logs del contenedor (p. ej. `sudo docker logs cowrie --since 5m` o `sudo docker logs -f cowrie`) y, desde pfSense (IP DMZ **192.168.200.1**), se genera un intento de conexión al puerto publicado del honeypot (por ejemplo `nc -vz 192.168.200.98 222`). Esto fuerza la creación de un evento de “nueva conexión” en Cowrie. En los logs se observa el origen **192.168.200.1**, lo que confirma que el evento proviene de la red DMZ y que el honeypot registra correctamente la actividad.



The screenshot shows the pfSense Diagnostic Command Prompt interface. In the terminal window, the user has run the command `nc -vz 192.168.200.98 222`. The output indicates a successful connection to port 222. Below this, the user has entered the command `nc -vz 192.168.200.98 222` again, with the "Execute" button highlighted. At the bottom, there is a "Download File" section with a "File to download" input field and a "Download" button. The main area of the window displays a log from the Cowrie SSH Factory. The log entries show multiple connections from the IP **192.168.200.1** at various timestamps, indicating repeated attempts to connect to the honeypot's SSH port. The log entries are as follows:

```
2026-01-27T17:39:18+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256 [session: 6733bf42a82]
2026-01-27T17:39:18+0000 [cowrie.ssh.transport.HoneyPotSSHTransportInfo] New connection: 192.168.200.1:3516 (172.17.0.2:222) [session: 6733bf42a82]
2026-01-27T17:39:18+0000 [cowrie.ssh.transport.HoneyPotSSHTransportInfo] connection lost
2026-01-27T17:39:18+0000 [HoneyPotSSHTransport, 6,192.168.200.1] Connection lost after 0.0 seconds
2026-01-27T17:39:29+0000 [cowrie.ssh.Factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256 [session: 3a5fc7d9ff58]
2026-01-27T17:39:29+0000 [cowrie.ssh.Factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256 [session: 3a5fc7d9ff58]
2026-01-27T17:39:29+0000 [cowrie.ssh.transport.HoneyPotSSHTransportInfo] connection lost
2026-01-27T17:39:29+0000 [HoneyPotSSHTransport, 7,192.168.200.1] Connection lost after 0.0 seconds
2026-01-27T19:07:45+0000 [cowrie.ssh.Factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256 [session: 4fffd7d3d5f]
2026-01-27T19:07:45+0000 [cowrie.ssh.Factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256 [session: 4fffd7d3d5f]
2026-01-27T19:07:45+0000 [cowrie.ssh.transport.HoneyPotSSHTransportInfo] connection lost
2026-01-27T19:08:41+0000 [HoneyPotSSHTransport, 9,192.168.200.1] Connection lost after 0.0 seconds
2026-01-27T19:08:41+0000 [HoneyPotSSHTransport, 9,192.168.200.1] Connection lost after 0.0 seconds
*x085*x085*x085
```

Captura (Figura 3.3-3) — Logs generados en DMZ

Inserta aquí la captura de `sudo docker logs cowrie --since 5m` donde se vean entradas recientes del tipo “New connection” con IP origen **192.168.200.1** y timestamp. Esta es la evidencia de que el honeypot genera logs desde la red DMZ y queda listo para su envío posterior a Elastic.

Verificación de acceso SSH desde WAN y generación de logs

Objetivo

Demostrar que el honeypot Cowrie es accesible desde la red WAN a través del NAT Port Forward configurado y que registra correctamente la actividad de los atacantes, incluyendo intentos de autenticación y comandos ejecutados.

Prueba de conexión SSH real desde WAN

Una vez resuelto el problema de accesibilidad, se realizó una conexión SSH completa desde el host Windows (red WAN) hacia el honeypot para validar su funcionamiento operativo y la generación de logs detallados.

Desde el equipo Windows (IP 192.168.1.249), se ejecutó el siguiente comando:

```
ssh -p 222 root@192.168.1.227
```

El honeypot respondió solicitando autenticación. Tras varios intentos con credenciales comunes (simulando el comportamiento de un atacante), se logró acceso al shell falso del honeypot. Una vez dentro, se ejecutaron comandos típicos de reconocimiento que un atacante real utilizaría:

- `whoami` – verificación de usuario actual
- `ls -la` – listado de archivos y directorios
- `pwd` – directorio de trabajo actual
- `uname -a` – información del sistema operativo

Finalmente, se cerró la sesión mediante el comando `exit`.

```
PS C:\Users\miche> ssh -p 222 root@192.168.1.227
The authenticity of host '[192.168.1.227]:222 ([192.168.1.227]:222)' can't be established.
ED25519 key fingerprint is SHA256:kaNKbIYUULw9Cp39nTalow6mCJEEP3Z6QE8+oeSBpxQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.1.227]:222' (ED25519) to the list of known hosts.
root@192.168.1.227's password:
Permission denied, please try again.
root@192.168.1.227's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@svr04:~# whoami
root
root@svr04:~# ls -la
drwx----- 1 root root 4096 2013-04-05 12:25 .
drwxr-xr-x 1 root root 4096 2013-04-05 12:03 ..
drwx----- 1 root root 4096 2013-04-05 11:58 .aptitude
-rw-r--r-- 1 root root 570 2013-04-05 11:52 .bashrc
-rw-r--r-- 1 root root 140 2013-04-05 11:52 .profile
drwx----- 1 root root 4096 2013-04-05 12:05 .ssh
root@svr04:~# pwd
/root
root@svr04:~# uname -a
Linux svr04 3.2.0-4-amd64 #1 SMP Debian 3.2.68-1+deb7u1 x86_64 GNU/Linux
root@svr04:~# exit
Connection to 192.168.1.227 closed.
PS C:\Users\miche>
```

– Sesión SSH desde Windows hacia el honeypot mostrando la conexión exitosa (prompt `root@svr04:~#`), la ejecución de los comandos de reconocimiento y sus respuestas, y la desconexión limpia mediante `exit`.

Registro de la actividad en Cowrie

Inmediatamente después de la sesión SSH, se consultaron los logs del contenedor Cowrie en la máquina KALI-HONEY para verificar que todos los eventos fueron registrados correctamente.

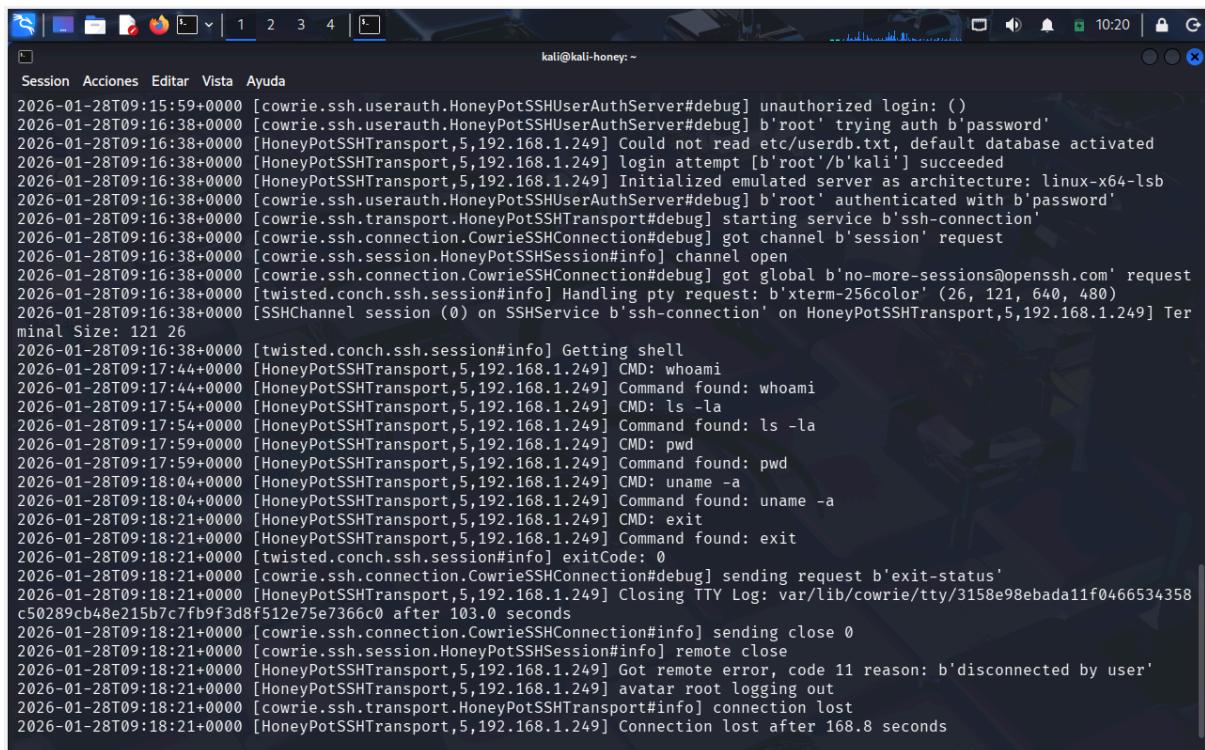
Se ejecutó el comando:

```
sudo docker logs cowrie --tail 100
```

Los logs confirmaron el registro completo de la sesión, incluyendo:

- Establecimiento de la conexión desde 192.168.1.249
- Intentos de autenticación (login attempts)
- Autenticación exitosa
- Cada comando ejecutado durante la sesión (whoami, ls -la, pwd, uname -a)
- Cierre de sesión

Esta evidencia demuestra que el honeypot no solo es accesible desde WAN, sino que cumple su función principal: registrar de forma detallada toda la actividad maliciosa o no autorizada dirigida hacia la infraestructura.



```
kali@kali-honey: ~
Session Acciones Editar Vista Ayuda
2026-01-28T09:15:59+0000 [cowrie ssh userauth HoneyPotSSHUserAuthServer#debug] unauthorized login: ()
2026-01-28T09:16:38+0000 [cowrie ssh userauth HoneyPotSSHUserAuthServer#debug] b'root' trying auth b'password'
2026-01-28T09:16:38+0000 [HoneyPotSSHTransport,5,192.168.1.249] Could not read etc/userdb.txt, default database activated
2026-01-28T09:16:38+0000 [HoneyPotSSHTransport,5,192.168.1.249] login attempt [b'root'/b'kali'] succeeded
2026-01-28T09:16:38+0000 [HoneyPotSSHTransport,5,192.168.1.249] Initialized emulated server as architecture: linux-x64-lsb
2026-01-28T09:16:38+0000 [cowrie ssh userauth HoneyPotSSHUserAuthServer#debug] b'root' authenticated with b'password'
2026-01-28T09:16:38+0000 [cowrie ssh transport HoneyPotSSHTransport#debug] starting service b'ssh-connection'
2026-01-28T09:16:38+0000 [cowrie ssh connection CowrieSSHConnection#debug] got channel b'session' request
2026-01-28T09:16:38+0000 [cowrie ssh session HoneyPotSSHSession#info] channel open
2026-01-28T09:16:38+0000 [cowrie ssh connection CowrieSSHConnection#debug] got global b'no-more-sessions@openssh.com' request
2026-01-28T09:16:38+0000 [twisted.conch.ssh.session#info] Handling pty request: b'xterm-256color' (26, 121, 640, 480)
2026-01-28T09:16:38+0000 [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotSSHTransport,5,192.168.1.249] Terminal Size: 121 26
2026-01-28T09:16:38+0000 [twisted.conch.ssh.session#info] Getting shell
2026-01-28T09:17:44+0000 [HoneyPotSSHTransport,5,192.168.1.249] CMD: whoami
2026-01-28T09:17:54+0000 [HoneyPotSSHTransport,5,192.168.1.249] Command found: whoami
2026-01-28T09:17:54+0000 [HoneyPotSSHTransport,5,192.168.1.249] CMD: ls -la
2026-01-28T09:17:59+0000 [HoneyPotSSHTransport,5,192.168.1.249] Command found: ls -la
2026-01-28T09:17:59+0000 [HoneyPotSSHTransport,5,192.168.1.249] CMD: pwd
2026-01-28T09:17:59+0000 [HoneyPotSSHTransport,5,192.168.1.249] Command found: pwd
2026-01-28T09:18:04+0000 [HoneyPotSSHTransport,5,192.168.1.249] CMD: uname -a
2026-01-28T09:18:04+0000 [HoneyPotSSHTransport,5,192.168.1.249] Command found: uname -a
2026-01-28T09:18:21+0000 [HoneyPotSSHTransport,5,192.168.1.249] CMD: exit
2026-01-28T09:18:21+0000 [HoneyPotSSHTransport,5,192.168.1.249] Command found: exit
2026-01-28T09:18:21+0000 [twisted.conch.ssh.session#info] exitCode: 0
2026-01-28T09:18:21+0000 [cowrie ssh connection CowrieSSHConnection#debug] sending request b'exit-status'
2026-01-28T09:18:21+0000 [HoneyPotSSHTransport,5,192.168.1.249] Closing TTY Log: var/lib/cowrie/tty/3158e98ebada11f0466534358c50289cb48e215b7c7fb9f3d8f512e75e7366c0 after 103.0 seconds
2026-01-28T09:18:21+0000 [cowrie ssh connection CowrieSSHConnection#info] sending close 0
2026-01-28T09:18:21+0000 [cowrie ssh session HoneyPotSSHSession#info] remote close
2026-01-28T09:18:21+0000 [HoneyPotSSHTransport,5,192.168.1.249] Got remote error, code 11 reason: b'disconnected by user'
2026-01-28T09:18:21+0000 [HoneyPotSSHTransport,5,192.168.1.249] avatar root logging out
2026-01-28T09:18:21+0000 [cowrie ssh transport HoneyPotSSHTransport#info] connection lost
2026-01-28T09:18:21+0000 [HoneyPotSSHTransport,5,192.168.1.249] Connection lost after 168.8 seconds
```

– Logs detallados de Cowrie mostrando el output de `sudo docker logs cowrie --tail 100`, donde se observa la IP origen 192.168.1.249, los intentos de login, los comandos ejecutados y el cierre de sesión.

4. Red DMZ2 (IDS Suricata)

La red DMZ2 constituye la tercera zona segmentada de la infraestructura y está diseñada para alojar un sistema de detección de intrusiones (IDS) que actúa como fuente adicional de logs para el SIEM. Según el enunciado de la práctica, en esta red debe ubicarse Suricata o

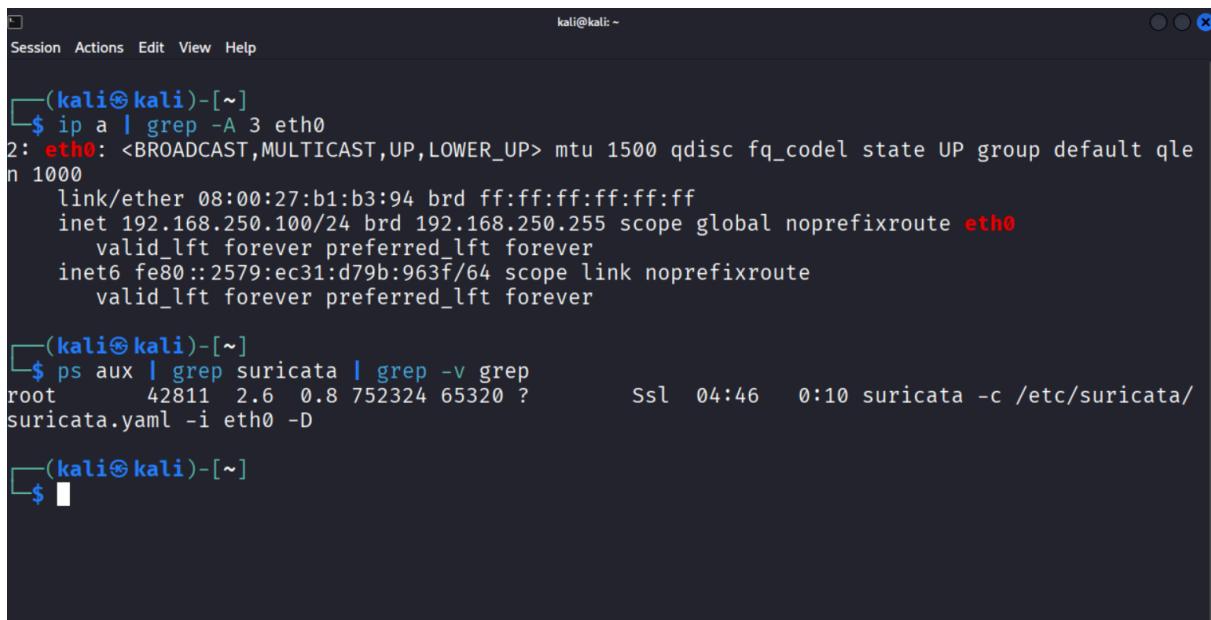
Apache Server como fuente de logs diferente a las dos mencionadas anteriormente (Windows en LAN y honeypot en DMZ).

La elección de Suricata responde a su capacidad para monitorizar tráfico de red en tiempo real, detectar amenazas mediante reglas personalizadas y generar logs estructurados en formato JSON, facilitando su posterior análisis en Elastic. La red DMZ2 se encuentra aislada de LAN y DMZ, de modo que cualquier anomalía detectada por Suricata queda contenida en esta zona sin comprometer otras redes internas.

4.1 Configuración de red del sistema en DMZ2

Antes de desplegar Suricata, se configuró el sistema Kali Linux destinado a esta función con una dirección IP fija dentro de la red DMZ2 (192.168.250.0/24). La máquina se configuró con la dirección IP 192.168.250.100, situada fuera del rango DHCP definido para esta red.

La asignación de una IP fija permite garantizar que los logs generados por Suricata tengan un origen estable y claramente identificable en el SIEM, facilitando su análisis y la correcta interpretación de las evidencias durante la evaluación de la práctica.



A terminal window titled 'kali@kali: ~' showing the following command-line session:

```
kali@kali: ~
Session Actions Edit View Help
[(kali㉿kali)-[~]
└─$ ip a | grep -A 3 eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b1:b3:94 brd ff:ff:ff:ff:ff:ff
    inet 192.168.250.100/24 brd 192.168.250.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::2579:ec31:d79b:963f/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[(kali㉿kali)-[~]
└─$ ps aux | grep suricata | grep -v grep
root      42811  2.6  0.8 752324 65320 ?        Ssl  04:46   0:10 suricata -c /etc/suricata/suricata.yaml -i eth0 -D
[(kali㉿kali)-[~]
└─$ █
```

Captura dmz2-1

4.2 Instalación y configuración de Suricata

Suricata se encontraba previamente instalado en el sistema como parte de un ejercicio realizado con el profesor. Se verificó la versión instalada (Suricata 8.0.3) y se realizó un test de configuración para confirmar que el archivo principal de configuración (/etc/suricata/suricata.yaml) era válido y no presentaba errores de sintaxis.

La configuración de Suricata incluye los siguientes elementos relevantes:

- **Interfaz de escucha:** eth0 (interfaz de red de la DMZ2)
- **Modo de captura:** af-packet (recomendado para capturas de alto rendimiento en Linux)
- **Archivo de logs:** /var/log/suricata/eve.json (formato JSON estructurado)
- **Archivo de reglas:** /etc/suricata/rules/suricata.rules (reglas personalizadas configuradas previamente)

Suricata se inició en modo daemon para que permanezca corriendo en segundo plano y capture tráfico de forma continua. Se verificó que el proceso estaba activo y escuchando correctamente en la interfaz eth0.

```
(kali㉿kali)-[~]
└─$ ps aux | grep suricata | grep -v grep
root      42811  2.2  0.8 752324 66072 ?          Ssl  04:46   1:27 suricata -c /etc/suricata/suricata.yaml -i eth0 -D

(kali㉿kali)-[~]
└─$ sudo suricata -T -c /etc/suricata/suricata.yaml
i: suricata: This is Suricata version 8.0.3 RELEASE running in SYSTEM mode
i: mpm-hs: Rule group caching - loaded: 1 newly cached: 0 total cacheable: 1
i: suricata: Configuration provided was successfully loaded. Exiting.

(kali㉿kali)-[~]
└─$ ls -lh /var/log/suricata/eve.json
-rw-r--r-- 1 root root 11M Jan 28 05:51 /var/log/suricata/eve.json

(kali㉿kali)-[~]
└─$ sudo tail -5 /var/log/suricata/eve.json | jq -c '{timestamp, event_type, src_ip}'
[{"timestamp": "2026-01-28T05:51:17.202488-0500", "event_type": "stats", "src_ip": null}, {"timestamp": "2026-01-28T05:51:25.203570-0500", "event_type": "stats", "src_ip": null}, {"timestamp": "2026-01-28T05:51:33.204614-0500", "event_type": "stats", "src_ip": null}, {"timestamp": "2026-01-28T05:51:41.208361-0500", "event_type": "stats", "src_ip": null}, {"timestamp": "2026-01-28T05:51:49.209672-0500", "event_type": "stats", "src_ip": null}]
```

Captura dmz2-7

4.3 Reglas de detección configuradas en Suricata

El archivo de reglas de Suricata contiene dos reglas activas configuradas durante el ejercicio previo con el profesor:

Regla SID 2: Detecta tráfico SSH hacia la dirección 192.168.1.138 en el puerto 22. Esta regla genera una alerta clasificada como "attempted-admin" cuando se detecta este tipo de tráfico. Aunque la IP objetivo no pertenece a la infraestructura actual de la práctica, la regla permanece activa como parte de la configuración heredada del ejercicio anterior.

Regla SID 3: Detecta descargas de archivos PDF mediante HTTP. Esta regla analiza el contenido de las respuestas HTTP en busca de la firma "%PDF-" al inicio del archivo, y cuando se detecta, genera una alerta clasificada como "file-download" y almacena una copia del archivo para análisis posterior.

Estas reglas permiten demostrar la capacidad de Suricata para generar alertas basadas en patrones de tráfico específicos, cumpliendo el requisito de la práctica de contar con una fuente de logs diferenciada en la DMZ2.

```
kali㉿kali:[~]
$ cat /etc/suricata/rules/suricata.rules
alert tcp any any → any any (msg:"trafico detectado"; sid:1;)
alert tcp any any → 192.168.1.138 22 (msg:"Trafico SSH detectado"; sid:2; classtype:attempted-admin;)
alert http any any → any any (msg:"Archivo PDF Detectado"; flow:established,to_client; file_data; content:"%PDF-"; within:5; filestore ; sid:3; classtype:file-download;)

(kali㉿kali:[~]
$ sudo grep "event_type": "alert" /var/log/suricata/eve.json | tail -3 | jq '.'
{
  "timestamp": "2026-01-26T17:02:57.976847-0500",
  "flow_id": 536353185133206,
  "in_iface": "eth0",
  "event_type": "alert",
  "src_ip": "192.168.1.249",
  "src_port": 59946,
  "dest_ip": "192.168.1.138",
  "dest_port": 22,
  "proto": "TCP",
  "ip_v": 4,
  "pkt_src": "wire/pcap",
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 2,
    "rev": 0,
    "signature": "Trafico SSH detectado",
    "category": "Attempted Administrator Privilege Gain",
    "severity": 1
  },
  "direction": "to_server",
  "flow": {
    "pkts_toserver": 1,
    "pkts_toclient": 0,
    "bytes_toserver": 138,
    "bytes_toclient": 0,
    "start": "2026-01-26T17:02:57.976847-0500",
    "src_ip": "192.168.1.249",
    "dest_ip": "192.168.1.138"
  }
}
```

Captura dmz2-8

4.4 Generación de logs en formato EVE JSON

Suricata genera logs en formato EVE (Extensible Event Format) en el archivo /var/log/suricata/eve.json. Este formato JSON estructurado facilita la integración con sistemas SIEM como Elastic, ya que cada evento se registra como un objeto JSON independiente con campos claramente definidos.

Los logs de Suricata incluyen múltiples tipos de eventos:

- **flow:** Información sobre flujos de red (conexiones TCP/UDP)
- **dns:** Consultas y respuestas DNS
- **http:** Transacciones HTTP detectadas
- **tls:** Conexiones cifradas TLS/SSL
- **alert:** Alertas generadas por las reglas de detección
- **stats:** Estadísticas periódicas de captura y procesamiento

Cada evento incluye metadatos como timestamp, dirección IP de origen (src_ip), dirección IP de destino (dest_ip), protocolo y tipo de evento. Esta información es esencial para el análisis posterior en el SIEM.

```

Session Actions Edit View Help
{"timestamp": "2026-01-28T04:56:02.076461-0500", "event_type": "dns", "src_ip": "8.8.8.8", "dest_ip": "192.168.250.100"}, {"timestamp": "2026-01-28T04:56:02.076463-0500", "event_type": "dns", "src_ip": "8.8.8.8", "dest_ip": "192.168.250.100"}, {"timestamp": "2026-01-28T04:56:02.253471-0500", "event_type": "tls", "src_ip": "192.168.250.100", "dest_ip": "34.36.137.203"}, {"timestamp": "2026-01-28T04:56:02.283206-0500", "event_type": "quic", "src_ip": "192.168.250.100", "dest_ip": "34.36.137.203"}, {"timestamp": "2026-01-28T04:56:02.283361-0500", "event_type": "quic", "src_ip": "192.168.250.100", "dest_ip": "34.36.137.203"}, {"timestamp": "2026-01-28T04:56:04.160096-0500", "event_type": "stats", "src_ip": null, "dest_ip": null}, {"timestamp": "2026-01-28T04:56:12.163642-0500", "event_type": "stats", "src_ip": null, "dest_ip": null}, {"timestamp": "2026-01-28T04:56:20.165105-0500", "event_type": "stats", "src_ip": null, "dest_ip": null}, {"timestamp": "2026-01-28T04:56:28.166798-0500", "event_type": "stats", "src_ip": null, "dest_ip": null}, {"timestamp": "2026-01-28T04:56:36.168195-0500", "event_type": "stats", "src_ip": null, "dest_ip": null}, {"timestamp": "2026-01-28T04:56:44.170636-0500", "event_type": "stats", "src_ip": null, "dest_ip": null}, {"timestamp": "2026-01-28T04:56:52.172517-0500", "event_type": "stats", "src_ip": null, "dest_ip": null}, {"timestamp": "2026-01-28T04:56:59.671645-0500", "event_type": "flow", "src_ip": "192.168.250.100", "dest_ip": "34.36.137.203"}, {"timestamp": "2026-01-28T04:57:00.175929-0500", "event_type": "stats", "src_ip": null, "dest_ip": null}, {"timestamp": "2026-01-28T04:57:08.182400-0500", "event_type": "stats", "src_ip": null, "dest_ip": null}, {"timestamp": "2026-01-28T04:57:10.649310-0500", "event_type": "flow", "src_ip": "192.168.250.100", "dest_ip": "8.8.8.8"}, {"timestamp": "2026-01-28T04:57:16.183974-0500", "event_type": "stats", "src_ip": null, "dest_ip": null}, {"timestamp": "2026-01-28T04:57:24.186702-0500", "event_type": "stats", "src_ip": null, "dest_ip": null}, {"timestamp": "2026-01-28T04:57:32.188403-0500", "event_type": "stats", "src_ip": null, "dest_ip": null}

```

Captura dmz2-2

4.5 Reglas de firewall aplicadas a la red DMZ2

Una vez validado el funcionamiento de Suricata, se revisaron las reglas de firewall asociadas a la interfaz DMZ2 en pfSense. Durante esta revisión se detectó un problema crítico de segmentación: las reglas configuradas previamente permitían que DMZ2 alcanzara tanto la red LAN como la red DMZ, violando el principio de aislamiento entre zonas.

Para corregir este problema, se reconfiguraron las reglas de firewall de DMZ2 siguiendo el principio de seguridad de "denegar por defecto, permitir explícitamente". Las reglas se diseñaron en el siguiente orden (de más específica a más general):

Regla 1 - Bloquear DMZ2 → LAN:

- Action: Block
- Protocol: any
- Source: DMZ2 subnets
- Destination: 192.168.100.0/24

Esta regla bloquea cualquier tipo de tráfico desde DMZ2 hacia la red LAN, protegiendo la zona más sensible de la infraestructura.

Regla 2 - Bloquear DMZ2 → DMZ:

- Action: Block
- Protocol: any
- Source: DMZ2 subnets
- Destination: 192.168.200.0/24

Esta regla impide que DMZ2 pueda comunicarse con la DMZ, donde se encuentra el honeypot. Ambas zonas deben permanecer aisladas entre sí.

Regla 3 - Permitir DMZ2 → Internet:

- Action: Pass
- Protocol: any
- Source: DMZ2 subnets
- Destination: any

Esta regla permite la salida de DMZ2 hacia cualquier destino. Como las dos reglas anteriores ya han bloqueado el acceso a redes internas (LAN y DMZ), esta regla solo permite el acceso hacia Internet, que es necesario para que Suricata pueda enviar logs al servidor Elastic y para mantener el sistema actualizado.

El orden de las reglas es crítico: pfSense procesa las reglas de arriba hacia abajo y aplica la primera que coincide con el tráfico. Por este motivo, las reglas de bloqueo (más específicas) deben estar antes que la regla de permiso general (más amplia).

Rules (Drag to Change Order)										
States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✗ 0/336 B	IPv4 *	DMZ2 subnets	*	192.168.100.0/24	*	*	none		Bloquear DMZ2 hacia LAN	
✗ 0/168 B	IPv4 *	DMZ2 subnets	*	192.168.200.0/24	*	*	none		Bloquear DMZ2 hacia DMZ	
✓ 1/2.25 MiB	IPv4 *	DMZ2 subnets	*	*	*	*	none		Permitir salida Internet	
✓ 0/0 B	IPv4 ICMP any		*	*	*	*	none		Regla tráfico ICMP	
✓ 0/0 B	IPv4 UDP	*	*	*	53 (DNS)	*	none		Regla DNS	
✓ 0/0 B	IPv4 TCP	*	*	*	webs	*	none		Regla tráfico web	

4.6 Configuración de NAT para DMZ2

Se verificó la configuración de NAT de salida (Outbound NAT) en pfSense para confirmar que la red DMZ2 puede traducir sus direcciones privadas a la dirección pública de la interfaz WAN al comunicarse con Internet.

El modo de NAT configurado es "Automatic Outbound NAT", lo que significa que pfSense genera automáticamente las reglas necesarias para todas las redes internas definidas en el sistema. En la sección de reglas automáticas se confirmó la presencia de las siguientes redes:

- 192.168.100.0/24 (LAN)
- 192.168.200.0/24 (DMZ)
- 192.168.250.0/24 (DMZ2)

Todas estas redes realizan NAT hacia la dirección de la interfaz WAN, permitiendo la salida a Internet desde cualquiera de ellas.

Interface	Source	Destination	NAT Address	NAT Port	Static Port	Description	Actions
WAN	DMZ subnets	*	*	*	*		

Interface	Source	Source Port	Destination	Destination Port	NAT Address	NAT Port	Static Port	Description
WAN	127.0.0.0/8 ::1/128 192.168.100.0/24 192.168.250.0/24	*	*	500	WAN address	*	*	Auto created rule for ISAKMP
WAN	127.0.0.0/8 ::1/128 192.168.100.0/24 192.168.200.0/24 192.168.250.0/24	*	*	*	WAN address	*	*	Auto created rule

Captura dmz2-4

4.7 Exposición desde WAN - Port Forward

Se verificó que la red DMZ2 no cuenta con ninguna regla de redirección de puertos (Port Forward) desde la interfaz WAN. Esto es coherente con el diseño de seguridad de la práctica: a diferencia de la DMZ, que expone el honeypot hacia el exterior mediante una regla de Port Forward en el puerto 222, la red DMZ2 no necesita ser accesible desde el exterior.

La única regla de Port Forward activa en el sistema es:

- WAN:222 → 192.168.200.98:222 (Honeypot Cowrie en DMZ)

Esta configuración garantiza que DMZ2 permanece inaccesible desde redes externas, minimizando la superficie de ataque y cumpliendo el principio de mínimo privilegio.

Firewall / NAT / Port Forward										
Port Forward 1:1 Outbound NPT										
Rules										
	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions
<input type="checkbox"/>	WAN	TCP	*	*	WAN address	222	192.168.200.98	222		
<input checked="" type="checkbox"/>	WAN	TCP	*	*	WAN address	80 (HTTP)	192.168.200.99	80 (HTTP)	Servidor web Apache	

Captura dmz2-5

4.8 Verificación de la segmentación de red

Una vez aplicadas las reglas de firewall, se realizaron pruebas de conectividad desde la máquina Kali en DMZ2 para verificar que la segmentación funciona correctamente:

Prueba 1 - Conectividad con gateway DMZ2:

```
ping -c 2 192.168.250.1
```

Resultado: Exitoso (0% packet loss). La comunicación con el gateway de la propia red DMZ2 funciona correctamente.

Prueba 2 - Bloqueo hacia LAN:

```
ping -c 2 192.168.100.1
```

Resultado: Bloqueado (100% packet loss). DMZ2 no puede alcanzar la red LAN, confirmando el aislamiento.

Prueba 3 - Bloqueo hacia DMZ:

```
ping -c 2 192.168.200.1
```

Resultado: Bloqueado (100% packet loss). DMZ2 no puede alcanzar la red DMZ, confirmando el aislamiento mutuo.

Prueba 4 - Salida a Internet:

```
ping -c 2 8.8.8.8
```

Resultado: Exitoso (0% packet loss). DMZ2 puede comunicarse con Internet correctamente, lo que permitirá el envío de logs a Elastic.

Estas pruebas confirman que la segmentación de red está correctamente implementada: DMZ2 puede salir a Internet pero permanece aislada de las redes internas (LAN y DMZ), cumpliendo los requisitos de seguridad de la práctica.

```

Session Actions Edit View Help
kali㉿kali:[~]
└─$ ping -c 2 192.168.250.1
PING 192.168.250.1 (192.168.250.1) 56(84) bytes of data.
64 bytes from 192.168.250.1: icmp_seq=1 ttl=64 time=1.69 ms
64 bytes from 192.168.250.1: icmp_seq=2 ttl=64 time=2.09 ms

--- 192.168.250.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.688/1.890/2.092/0.202 ms

└─$ ping -c 2 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.

--- 192.168.100.1 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1012ms

└─$ ping -c 2 192.168.200.1
PING 192.168.200.1 (192.168.200.1) 56(84) bytes of data.

--- 192.168.200.1 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1010ms

└─$ ping -c 2 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=112 time=32.5 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=112 time=42.5 ms

--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 32.480/37.476/42.473/4.996 ms

└─$ 

```

Captura dmz2-6

4.9 Conclusión de la configuración de DMZ2

Con estos ajustes, la red DMZ2 queda correctamente configurada como fuente de logs para el SIEM:

- Suricata está operativo y capturando tráfico en la interfaz eth0
- Los logs se generan en formato JSON estructurado en /var/log/suricata/eve.json
- Las reglas de detección están activas y generan alertas cuando se cumplen las condiciones definidas
- La segmentación de red está correctamente implementada: DMZ2 está aislada de LAN y DMZ, pero tiene salida a Internet
- El NAT de salida permite la comunicación con servicios externos como Elastic

- No existen puntos de entrada desde WAN hacia DMZ2, minimizando la superficie de ataque

La red DMZ2 está preparada para integrarse con el servidor Elastic y comenzar el envío de logs una vez este último sea instalado y configurado.

Elastic

Status	Host	Agent policy	CPU	Memory	Last active	Version	Actions
Healthy	Windows10	Windows rev. 2	7.32 %	315 MB	27 seconds ago	9.2.4	...
Healthy	kali-honey	Honeypot rev. 6	N/A	227 MB	17 seconds ago	9.2.4	...
Healthy	kali	Politica Linux Suricata rev. 2	6.27 %	257 MB	16 seconds ago	9.2.4	...
Healthy	daf1c20bd33b	Elastic Cloud agent policy rev. 4	N/A	N/A	12 seconds ago	9.2.4	...

The screenshot shows the Fleet interface with the following details:

- Title:** Fleet
- Description:** Centralized management for Elastic Agents.
- Navigation:** Agents, Agent policies (highlighted), Enrollment tokens, Uninstall tokens, Data streams, Settings.
- Search Bar:** Filter your data using KQL syntax.
- Reload Button:** Reload the page.
- Create agent policy Button:** Create a new agent policy.
- Table Headers:** Name (with sort arrow), Last updated on, Unprivileged / Privileged, Integrations, Actions.
- Data Rows:**
 - Honeypot rev. 6 (Last updated: Jan 28, 2026, 0 / 1 (1) integrations, 2 actions)
 - Windows rev. 2 (Last updated: Jan 28, 2026, 0 / 1 (1) integrations, 2 actions)
 - Politica Linux Suricata rev. 2 (Last updated: Jan 28, 2026, 0 / 1 (1) integrations, 2 actions)
 - Elastic Cloud agent policy (Last updated: Jan 28, 2026, 1 / 0 (1) integrations, 2 actions)
 - Default agent policy for agents hosted on Elastic Cloud
- Page Controls:** Rows per page: 20, Page number: 1, Total pages: 1.

5. Conexión a Elastic

5.1. Honeypot DMZ

Contexto

El honeypot Cowrie en DMZ (192.168.200.98) genera logs de intentos de intrusión SSH que deben ser enviados a Elastic Cloud para análisis centralizado.

dmz-honeypot-1: Port Forwarding en pfSense

Objetivo: Exponer el honeypot a WAN para captar intentos de ataque externos.

Configuración en Firewall → NAT → Port Forward:

- **Regla 1:** WAN:222 → 192.168.200.98:222 (Cowrie SSH honeypot)
- **Regla 2:** WAN:22 → 192.168.200.98:22 (SSH real administrativo)

Justificación técnica:

- Puerto 222: Cowrie simula SSH en puerto no estándar
- Puerto 22: Acceso administrativo real a la máquina Kali-Honey

Firewall / NAT / Port Forward											
Port Forward		1:1		Outbound		NPT					
Rules											
	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions	
<input type="checkbox"/>	✓ WAN	TCP	*	*	WAN address	222	192.168.200.98	222	Regla honeypot ssh 222		
<input type="checkbox"/>	✓ WAN	TCP	*	*	WAN address	22 (SSH)	192.168.200.98	22 (SSH)	SSH KaliHoney		
<input type="checkbox"/>	✓ WAN	TCP	*	*	WAN address	80 (HTTP)	192.168.200.99	80 (HTTP)	Servidor web Apache		

Captura: dmz-honeypot-1.png - Reglas de Port Forward activas

dmz-honeypot-2: Verificación de logs de Cowrie

Problema identificado: Al intentar conectar desde Windows físico con `ssh -p 222 root@192.168.1.227`, la conexión quedaba en timeout.

Diagnóstico:

bash

`sudo docker ps`

Resultado: Sin contenedores activos. Cowrie no estaba corriendo.

Solución aplicada:

bash

`sudo docker run -d --name cowrie -p 222:2222 cowrie/cowrie`

Verificación:

bash

`sudo docker ps`

Resultado: Contenedor activo con mapeo `0.0.0.0:222->2222/tcp ✓`

Generación de tráfico de prueba:

bash

`ssh -p 222 root@localhost`

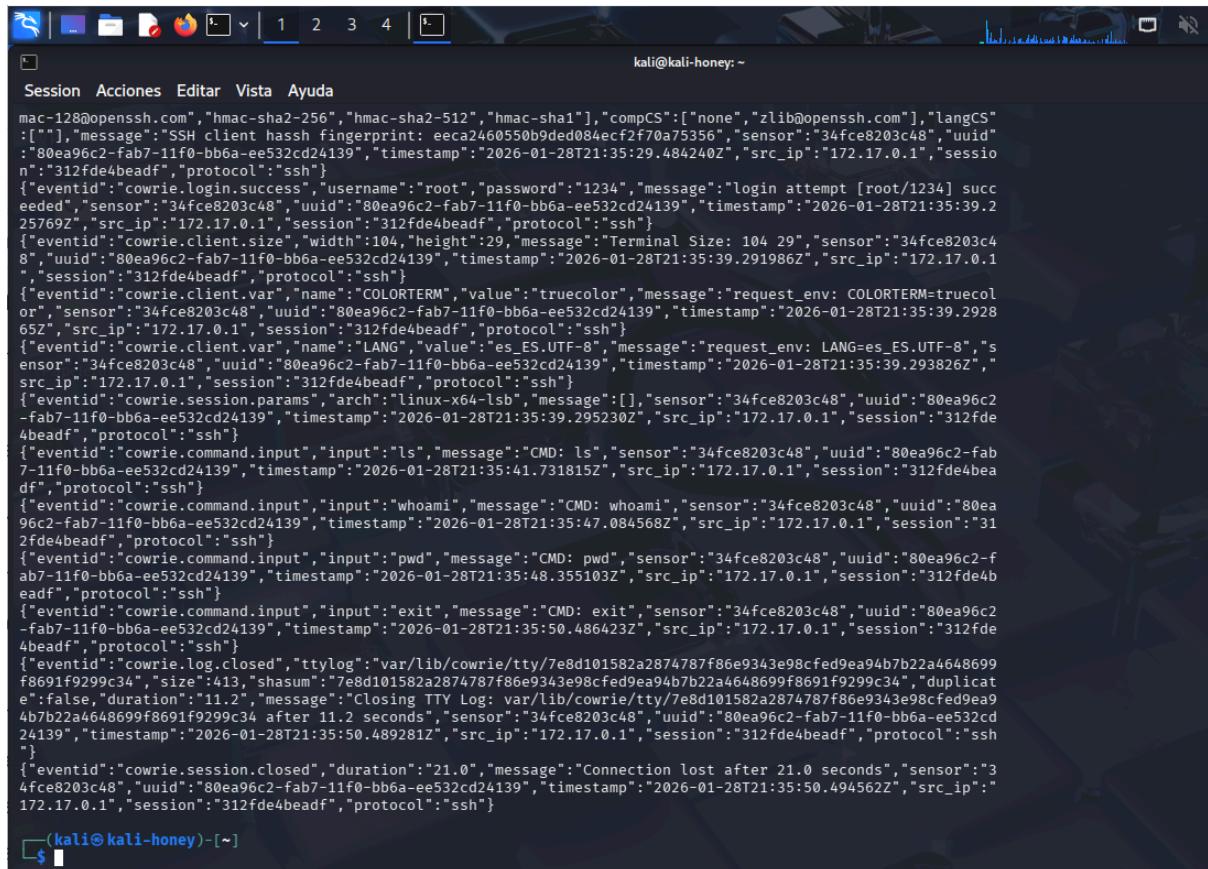
Comandos ejecutados: `ls, whoami, pwd, exit`

Extracción de logs:

bash

```
sudo docker cp cowrie:/cowrie/cowrie-git/var/log/cowrie/cowrie.json
```

```
/home/kali/cowrie-logs/cowrie.json
```



```
mac-128@openssh.com", "hmac-sha2-256", "hmac-sha2-512", "hmac-sha1"], "compCS": ["none", "zlib@openssh.com"], "langCS": "", "message": "SSH client hash fingerprint: eeca2460550b9ded084ecf2f70a75356", "sensor": "34fce8203c48", "uuid": "80ea96c2-fab7-11f0-bb6a-ee532cd24139", "timestamp": "2026-01-28T21:35:29.484240Z", "src_ip": "172.17.0.1", "session": "312fde4beadf", "protocol": "ssh"} {"eventid": "cowrie.login.success", "username": "root", "password": "1234", "message": "Login attempt [root/1234] succeeded", "sensor": "34fce8203c48", "uuid": "80ea96c2-fab7-11f0-bb6a-ee532cd24139", "timestamp": "2026-01-28T21:35:39.25769Z", "src_ip": "172.17.0.1", "session": "312fde4beadf", "protocol": "ssh"} {"eventid": "cowrie.client.size", "width": 104, "height": 29, "message": "Terminal Size: 104 29", "sensor": "34fce8203c48", "uuid": "80ea96c2-fab7-11f0-bb6a-ee532cd24139", "timestamp": "2026-01-28T21:35:39.291986Z", "src_ip": "172.17.0.1", "session": "312fde4beadf", "protocol": "ssh"} {"eventid": "cowrie.client.var", "name": "COLORTERM", "value": "truecolor", "message": "request_env: COLORTERM=truecolor", "sensor": "34fce8203c48", "uuid": "80ea96c2-fab7-11f0-bb6a-ee532cd24139", "timestamp": "2026-01-28T21:35:39.292865Z", "src_ip": "172.17.0.1", "session": "312fde4beadf", "protocol": "ssh"} {"eventid": "cowrie.client.var", "name": "LANG", "value": "es_ES.UTF-8", "message": "request_env: LANG=es_ES.UTF-8", "sensor": "34fce8203c48", "uuid": "80ea96c2-fab7-11f0-bb6a-ee532cd24139", "timestamp": "2026-01-28T21:35:39.293826Z", "src_ip": "172.17.0.1", "session": "312fde4beadf", "protocol": "ssh"} {"eventid": "cowrie.session.params", "arch": "linux-x64-lsb", "message": [], "sensor": "34fce8203c48", "uuid": "80ea96c2-fab7-11f0-bb6a-ee532cd24139", "timestamp": "2026-01-28T21:35:39.295230Z", "src_ip": "172.17.0.1", "session": "312fde4beadf", "protocol": "ssh"} {"eventid": "cowrie.command.input", "input": "ls", "message": "CMD: ls", "sensor": "34fce8203c48", "uuid": "80ea96c2-fab7-11f0-bb6a-ee532cd24139", "timestamp": "2026-01-28T21:35:41.731815Z", "src_ip": "172.17.0.1", "session": "312fde4beadf", "protocol": "ssh"} {"eventid": "cowrie.command.input", "input": "whoami", "message": "CMD: whoami", "sensor": "34fce8203c48", "uuid": "80ea96c2-fab7-11f0-bb6a-ee532cd24139", "timestamp": "2026-01-28T21:35:47.084568Z", "src_ip": "172.17.0.1", "session": "312fde4beadf", "protocol": "ssh"} {"eventid": "cowrie.command.input", "input": "pwd", "message": "CMD: pwd", "sensor": "34fce8203c48", "uuid": "80ea96c2-fab7-11f0-bb6a-ee532cd24139", "timestamp": "2026-01-28T21:35:48.355103Z", "src_ip": "172.17.0.1", "session": "312fde4beadf", "protocol": "ssh"} {"eventid": "cowrie.command.input", "input": "exit", "message": "CMD: exit", "sensor": "34fce8203c48", "uuid": "80ea96c2-fab7-11f0-bb6a-ee532cd24139", "timestamp": "2026-01-28T21:35:50.486423Z", "src_ip": "172.17.0.1", "session": "312fde4beadf", "protocol": "ssh"} {"eventid": "cowrie.log.closed", "ttylog": "var/lib/cowrie/tty/7e8d101582a2874787f86e9343e98cfed9ea94b7b22a4648699f8691f0299c34", "size": 413, "shasum": "7e8d101582a2874787f86e9343e98cfed9ea94b7b22a4648699f8691f0299c34", "duplicate": false, "duration": "11.2", "message": "Closing TTY Log: var/lib/cowrie/tty/7e8d101582a2874787f86e9343e98cfed9ea94b7b22a4648699f8691f0299c34 after 11.2 seconds", "sensor": "34fce8203c48", "uuid": "80ea96c2-fab7-11f0-bb6a-ee532cd24139", "timestamp": "2026-01-28T21:35:50.489281Z", "src_ip": "172.17.0.1", "session": "312fde4beadf", "protocol": "ssh"} {"eventid": "cowrie.session.closed", "duration": "21.0", "message": "Connection lost after 21.0 seconds", "sensor": "34fce8203c48", "uuid": "80ea96c2-fab7-11f0-bb6a-ee532cd24139", "timestamp": "2026-01-28T21:35:50.494562Z", "src_ip": "172.17.0.1", "session": "312fde4beadf", "protocol": "ssh"}
```

dmz-honeypot-3: Integración con Elastic Cloud

Configuración en Fleet:

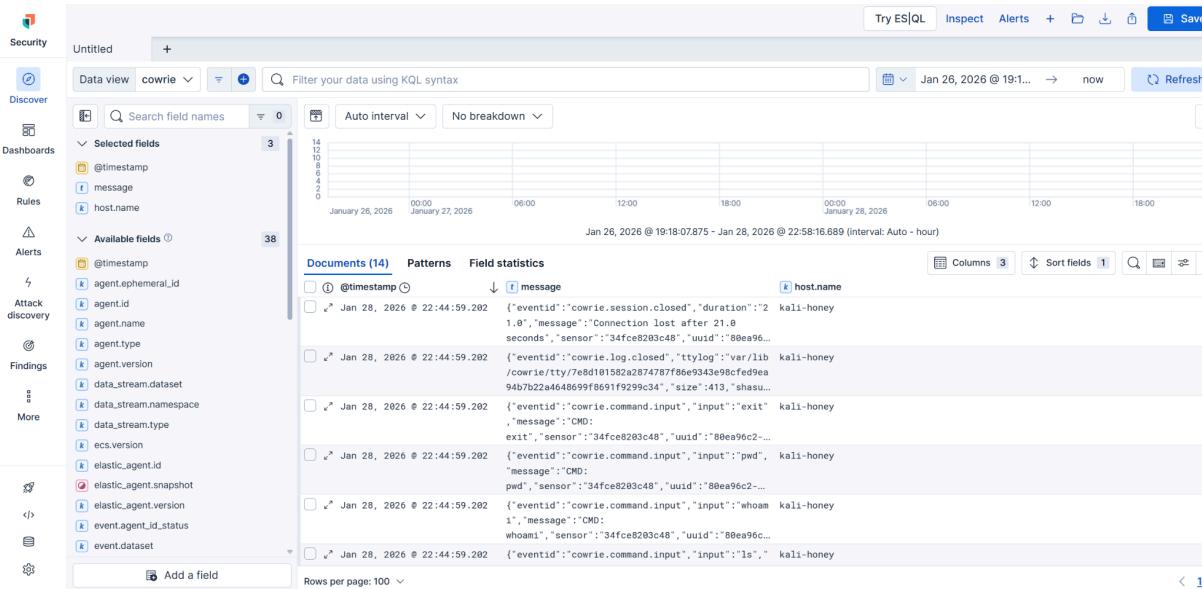
1. Política: **Honeypot** (ya existente con agente instalado en kali-honey)
2. Integración añadida: **Custom Logs**
 - **Dataset name:** cowrie
 - **Log file path:** /home/kali/cowrie-logs/cowrie.json
3. Índice generado: logs-filestream.generic-cowrie

Creación de Data View:

- **Name:** Cowrie Honeypot
- **Index pattern:** logs-filestream.generic-cowrie
- **Timestamp field:** @timestamp

Resultado: 14 eventos de Cowrie correctamente indexados en Elastic, incluyendo:

- `cowrie.session.closed` - Sesión finalizada
- `cowrie.command.input` - Comandos ejecutados (ls, whoami, pwd, exit)
- `cowrie.log.closed` - Cierre de logs TTY



Captura: `dmz-honeypot-3.png` - Discover mostrando eventos filtrados del honeypot

Archivo adjunto: `cowrie-honeypot-logs.json` - Logs completos en formato JSON para análisis

Explicación de campos de los logs de Cowrie:

Los logs de Cowrie se generan en formato JSON con los siguientes campos principales:

- **eventid:** Tipo de evento capturado (ej: `cowrie.command.input`, `cowrie.login.success`, `cowrie.session.closed`)
- **input:** Comando ejecutado por el atacante en el honeypot
- **message:** Descripción legible del evento
- **timestamp:** Fecha y hora exacta del evento en formato ISO 8601
- **src_ip:** Dirección IP de origen del atacante (172.17.0.1 indica red interna Docker)
- **session:** Identificador único de la sesión SSH
- **protocol:** Protocolo utilizado (ssh)
- **sensor:** Identificador del honeypot
- **uuid:** Identificador único del evento

Ejemplo de evento capturado:

```
json
"eventid": "cowrie.command.input"
```

```
"input": "exit"  
"src_ip": "172.17.0.1"  
"session": "312fde4beadf"
```

Este evento indica que se ejecutó el comando "exit" durante la sesión identificada con el ID especificado.

5.2. Windows 10 LAN - Conexión a Elastic

Contexto

Windows 10 en LAN (192.168.100.50) genera logs del sistema operativo que deben ser enviados a Elastic Cloud para monitorización centralizada.

lan-windows-1: Agente Elastic en Windows 10

Configuración en Fleet:

- **Host:** Windows10
- **IP:** 192.168.100.50 (LAN)
- **Política:** Windows (rev. 2)
- **Integración activa:** System v2.12.0
- **Estado:** Healthy ✓
- **Última actividad:** Hace 18 segundos

Justificación técnica: La integración "System" captura logs del sistema operativo Windows, incluyendo eventos de aplicación, sistema, seguridad y errores operativos.

agents	filters	Status	Host	Agent policy	CPU	Memory	Last activ...	Version
		Healthy	Windows10	Windows rev. 2	9.48 %	324 MB	18 seconds ago	9.2.4
		Offline	kali-honey	Honeypot rev. 6	N/A	N/A	10 minutes ago	9.2.4
		Healthy	kali	Honeypot rev. 6	5.09 %	259 MB	19 seconds ago	9.2.4
		Healthy	daf1c20bd33b	Elastic Cloud agent policy rev. 4	N/A	N/A	30 seconds ago	9.2.4

Rows per page: 20

< 1 >

Captura: [lan-windows-1.png](#) - Agente Windows10 Healthy en Fleet

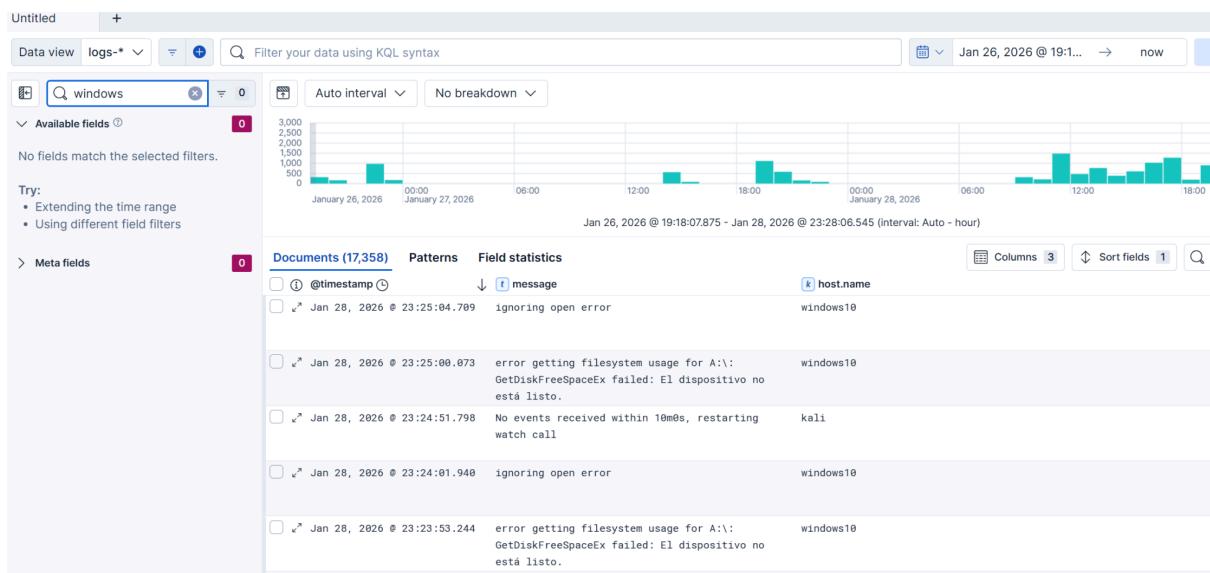
Ian-windows-2: Logs recibidos en Elastic

Visualización en Discover:

- **Data View:** logs-*
- **Filtro aplicado:** windows (busca en todos los campos)
- **Documentos recibidos:** 17,358 eventos ✓
- **Host confirmado:** windows10 ✓

Tipos de logs capturados:

- Errores de sistema (`ignoring open error`)
- Errores de filesystem (`GetDiskFreeSpaceEx failed`)
- Eventos de reinicio de servicios (`No events received within 10ms, restarting watch call`)



Captura: [lan-windows-2.png](#) - Discover mostrando logs de Windows10

Archivo adjunto: [windows10-lan-logs.json](#) - Logs completos en formato JSON para análisis

Explicación de campos de los logs de Windows:

Los logs de Windows Security se capturan en formato ECS (Elastic Common Schema) con los siguientes campos principales:

- **event.code:** Código del evento de Windows (ej: 4624 = inicio de sesión exitoso)
- **event.action:** Acción realizada (logged-in, logoff, etc.)
- **event.outcome:** Resultado del evento (success, failure)
- **user.name:** Nombre de usuario involucrado en el evento
- **user.domain:** Dominio del usuario (NT AUTHORITY, WORKGROUP)
- **host.name:** Nombre del equipo que generó el log (windows10)
- **host.ip:** Dirección IP del equipo (192.168.100.50)

- **winlog.event_id**: ID del evento de Windows
- **winlog.logon.type**: Tipo de inicio de sesión (Service=5, Interactive=2, Network=3)
- **process.executable**: Proceso que generó el evento
- **@timestamp**: Marca temporal del evento

Ejemplo de evento capturado:

json

```
"event.code": "4624"

"event.action": "logged-in"

"user.name": "SYSTEM"

"host.ip": "192.168.100.50"

"winlog.logon.type": "Service"
```

Este evento indica un inicio de sesión exitoso de tipo servicio por parte de la cuenta SYSTEM en el equipo Windows10.

5.3. Suricata DMZ2

Contexto

Suricata IDS en DMZ2 (192.168.250.100) monitoriza el tráfico de red y genera alertas de seguridad que deben ser enviadas a Elastic Cloud para análisis centralizado.

dmz2-suricata-1: Logs recibidos en Elastic

Configuración en Fleet:

- **Host**: kali
- **IP**: 192.168.250.100 (DMZ2)
- **Política**: Política Linux Suricata (rev. 2)
- **Integración activa**: Suricata v1.18.1
- **Estado**: Healthy
- **Última actividad**: Hace 19 segundos

Visualización en Discover:

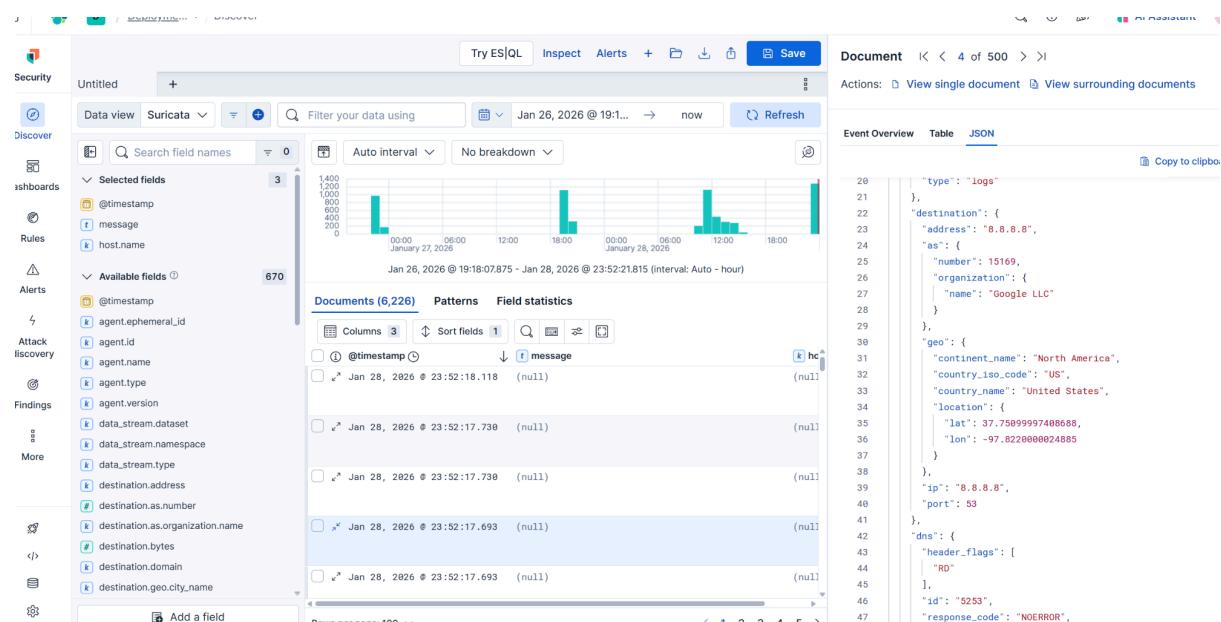
- **Data View**: Suricata
- **Documentos recibidos**: 4,943 eventos
- **Index**: logs-suricata.eve-default

- **Log file path:** /var/log/suricata/eve.json

Tipos de eventos capturados:

- **DNS queries** - Consultas DNS monitorizadas
 - **TLS connections** - Conexiones cifradas detectadas
 - **Network flows** - Flujos de red analizados
 - **Source IP**: 192.168.250.100 (Kali DMZ2)
 - **Destinations**: 8.8.8.8 (Google DNS), Elastic Cloud endpoints

Justificación técnica: Suricata actúa como IDS (Intrusion Detection System) monitorizando todo el tráfico de la interfaz eth0 en DMZ2, generando logs en formato JSON Eve que son enviados automáticamente a Elastic mediante Filebeat.



Captura: dmz2-suricata-1.png - Discover mostrando eventos de Suricata

Archivo adjunto: `suricata-dmz2-logs.json` - Logs completos en formato JSON para análisis

Explicación de campos de los logs de Suricata:

Los logs de Suricata EVE se generan en formato JSON con los siguientes campos principales:

- **suricata.eve.event_type**: Tipo de evento detectado (dns, tls, flow, alert, http)
 - **observer.type**: Tipo de observador (ids = Intrusion Detection System)
 - **observer.product**: Producto que generó el log (Suricata)
 - **observer.hostname**: Nombre del host donde corre Suricata (kali)
 - **observer.ip**: Dirección IP del sistema IDS (192.168.250.100)
 - **source.ip**: Dirección IP de origen del tráfico

- **source.port**: Puerto de origen
- **destination.ip**: Dirección IP de destino
- **destination.port**: Puerto de destino
- **network.protocol**: Protocolo de red (dns, http, tls)
- **network.transport**: Protocolo de transporte (udp, tcp)
- **dns.type**: Tipo de consulta DNS (request, response)
- **@timestamp**: Marca temporal del evento

Ejemplo de evento capturado:

```
json
"suricata.eve.event_type": "dns"
"observer.ip": "192.168.250.100"
"source.ip": "192.168.250.100"
"destination.ip": "8.8.8.8"
"network.protocol": "dns"
```

Este evento indica una consulta DNS desde el sistema Suricata en DMZ2 hacia el servidor DNS público de Google, siendo detectada y registrada por el IDS.