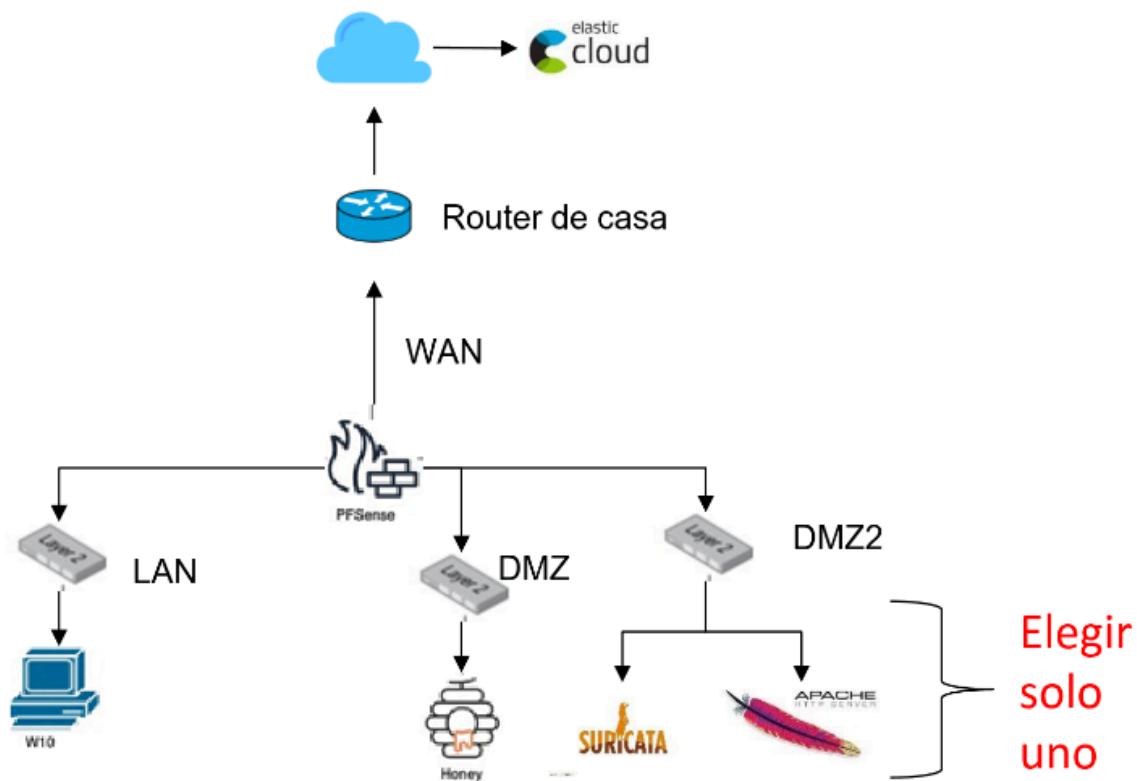


BLUE TEAM: MONTAJE DE UNA INFRAESTRUCTURA DE RED



Primero y antes de nada, descargamos el **PFSENSE** desde el siguiente enlace:

- https://drive.google.com/file/d/1c38GjIVtqJV4OotIOoKUvNbc_6REhRKC/view?usp=drive_link

Una vez descargado, **extraemos** la imagen ISO e inicializamos virtualbox, en el menú de virtualbox vamos a máquina y la damos a nueva para añadir una nueva máquina y comenzamos la instalación, en los primeros parámetros, **cargamos la imagen ISO** escogeremos **tipo BSD** y la **versión FreeBSD (64-bit)**.

En la siguiente pantalla escogeremos una **memoria base de 1250 mb** (aproximadamente) y un **procesador de 1 cpu**. Después en la siguiente pantalla dejaremos por defecto un **tamaño de disco de 16 GB** y en la última pantalla le daremos a **finalizar**.

Una vez tenemos la máquina virtual en nuestro menú de máquinas virtuales de virtualbox, haciendo clic derecho vamos a **configuración > red**, aquí nos encontraremos cuatro **adaptadores** que deberán quedar de la siguiente forma:

- Adaptador 1: Adaptador puente
- Adaptador 2: Red interna con nombre “LAN”.
- Adaptador 3: Red interna con nombre “DMZ”.
- Adaptador 4: Red interna con nombre “DMZ2”.

Después, arrancamos la máquina para que comience la **instalación de PFSENSE**:

- Primera pantalla: pulsamos aceptar.
- Segunda pantalla: pulsamos OK sin cambiar parámetros.
- Tercera pantalla: pulsamos OK sin cambiar parámetros.
- Cuarta pantalla: pulsamos SELECT sin cambiar parámetros.
- Quinta pantalla: pulsamos OK sin cambiar parámetros.
- Sexta pantalla (configuración): le damos a la barra espaciadora, debería salir un asterisco, después pulsamos OK.
- Última pantalla (destruir datos): pulsamos YES.

Después de esto empezará la instalación, cuando termine apagamos la máquina sin dejar que vuelva a la primera pantalla en negro. Volvemos a abrir VirtualBox y en la máquina de

PFSENSE hacemos clic derecho y vamos a **configuración > almacenamiento y borramos la imagen ISO**.

Si arrancamos la máquina y esperamos un minuto deberíamos ver esto:

```
blue team - pc sense [Running] - Oracle VM VirtualBox
done.
Starting CRON... done.
pfSense 2.7.2-RELEASE amd64 20231206-2010
Bootup complete

FreeBSD/amd64 (pfSense.home.arpa) (ttyv0)

VirtualBox Virtual Machine - Netgate Device ID: 8702bd502d9544309884

*** Welcome to pfSense 2.7.2-RELEASE (amd64) on pfSense ***

WAN (wan)      -> em0      -> v4/DHCP4: 192.168.1.41/24
LAN (lan)      -> em1      -> v4: 192.168.1.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults   13) Update from console
5) Reboot system               14) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option: 
```

A continuación, arrancamos una máquina Kali y en **configuración > red**, en el adaptador 1 escogeremos **red interna con nombre LAN**.

Para que nos asigne una **nueva dirección IP** desconectamos y volvemos a conectar el **adaptador de red** de la máquina Kali, en el menú inferior izquierdo, el cuarto ícono empezando por la derecha, las dos pantallas, escogeremos “connect network adapter” para desconectar y volveremos a hacerlo para conectarlo de nuevo.



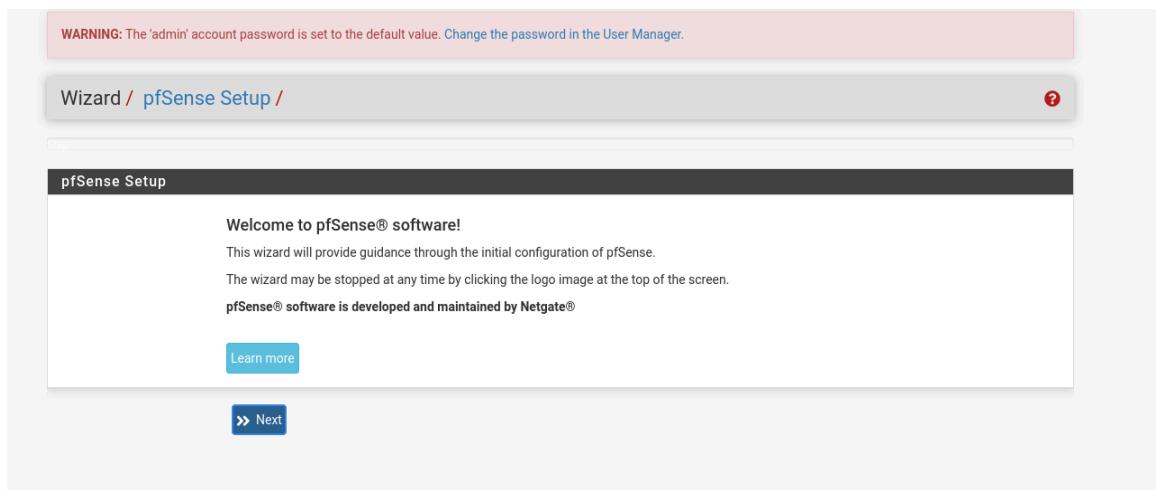
De esta manera nuestra máquina Kali estará conectada a la red LAN dentro del rango que marca nuestro PFSENSE, podemos comprobarlo ejecutando en la terminal de Kali el comando “ip a”.

Después, abrimos el navegador y ponemos la **dirección de la LAN** (192.168.1.1 en mi caso), nos saldrá una pantalla de advertencia, aquí clicamos en “**avanzado**” y hacemos clic en “**aceptar el riesgo**”.

El navegador nos llevará a una página de inicio de sesión:

- Usuario: **admin**
- Contraseña: **pfsense**

Debería salirnos una pantalla como esta:



En esta pantalla, hacemos clic en “NEXT”, en la siguiente volvemos a hacer clic en “NEXT”, y en la siguiente pondremos los siguientes parametros:

- Nombre: UTM

- Dominio: keepcoding.local
- Servidor primario DNS: 127.0.0.1
- Servidor secundario DNS: 1.1.1.1

Hacemos clic en “NEXT”, en la siguiente pantalla escogemos **zona horaria** y hacemos clic en “NEXT”, en esta pantalla debemos ir hacia al final, veremos **dos secciones** que están marcadas con un tic azul, debemos **descartarlas** quitándole el tic azul, después le damos a “NEXT”, en la siguiente pantalla ajustaremos la LAN a los parámetros de nuestra infraestructura de red, por ello pondremos en “**LAN IP ADDRESS**” **192.168.100.1**, hacemos clic en “NEXT” y en esta pantalla tendremos que cambiar nuestra contraseña, importante no olvidarla ya que si esto nos pasara tendríamos que repetir la instalación, le damos clic a “NEXT” y en la última pantalla le damos clic a “RELOAD”.

Ahora necesitamos que se nos asigne una nueva dirección IP por lo que volveremos a repetir el proceso de **conectar y desconectar el adaptador de red**.



Si hacemos “ip a” en nuestra máquina Kali podremos comprobar que se nos ha asignado correctamente esta nueva dirección IP.

```
(kali㉿kali)-[~]
$ 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 08:00:27:1f:b7:23 brd ff:ff:ff:ff:ff:ff
        inet 192.168.100.10/24 brd 192.168.100.255 scope global dynamic noprefixroute eth0
            valid_lft 7196sec preferred_lft 7196sec
        inet6 fe80::e604:bae0:d67e:2528/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 82:9a:0c:74:c7:c0 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
(kali㉿kali)-[~]
$
```

Una vez hecho esto, volvemos a entrar al PFSENSE desde el navegador, con la ip **192.168.100.1**, y ponemos el usuario (admin) con la nueva contraseña que pusimos anteriormente.

Una vez dentro en la parte superior vamos a **services > DNS Resolver**, aquí **deshabilitamos DNSSEC** y **habilitamos el DNS QUERY FORWARDING** hacemos clic en “SAVE” y después en “APPLY CHANGES”.

Después vamos a **services > DHCP Server** y cambiamos **primary address pool, DNS y Gateway** como aparecen en las siguientes imágenes y guardamos:

The screenshot shows the configuration page for the Primary Address Pool in the Kali Linux web interface. The Subnet is set to 192.168.100.0/24, and the Subnet Range is 192.168.100.1 - 192.168.100.254. The Address Pool Range is defined from 192.168.100.100 to 192.168.100.200. The WINS Servers listed are WINS Server 1 and WINS Server 2. The DNS Servers listed are 192.168.100.1, DNS Server 2, and DNS Server 3.

Primary Address Pool	
Subnet	192.168.100.0/24
Subnet Range	192.168.100.1 - 192.168.100.254
Address Pool Range	From: 192.168.100.100 To: 192.168.100.200
Additional Pools	+ Add Address Pool If additional pools of addresses are needed inside of this subnet outside the above range, they may be specified here.

Server Options	
WINS Servers	WINS Server 1 WINS Server 2
DNS Servers	192.168.100.1 DNS Server 2 DNS Server 3

Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

Subnet Range 192.168.100.1 - 192.168.100.254

Address Pool Range 192.168.100.100 To 192.168.100.200
From
The specified range for this pool must not be within the range configured on any other address pool for this interface.

Additional Pools [+ Add Address Pool](#)
If additional pools of addresses are needed inside of this subnet outside the above range, they may be specified here.

Server Options

WINS Servers WINS Server 1
WINS Server 2

DNS Servers 192.168.100.1
1.1.1.1
8.8.8.8
DNS Server 4

OMAPI

OMAPI Port OMAPI Port
Set the port that OMAPI will listen on. The default port is 7911, leave blank to disable. Only the first OMAPI configuration is used.

OMAPI Key OMAPI Key Generate New Key
Enter a key matching the selected algorithm to secure connections to the OMAPI endpoint.
Generate a new key based on the selected algorithm.

Key Algorithm HMAC-SHA256 (current bind9 default)
Set the algorithm that OMAPI key will use.

Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

Other DHCP Options

Gateway 192.168.100.1
The default is to use the IP address of this firewall interface as the gateway. Specify an alternate gateway here if this is not the correct gateway for the network. Enter "none" for no gateway assignment.

Domain Name keepcoding.local
The default is to use the domain name of this firewall as the default domain name provided by DHCP. An alternate domain name may be specified here.

Domain Search List example.com;sub.example.com
The DHCP server can optionally provide a domain search list. Use the semicolon character as separator.

Default Lease Time 7200
This is used for clients that do not ask for a specific expiration time. The default is 7200 seconds.

Maximum Lease Time 86400
This is the maximum lease time for clients that ask for a specific expiration time. The default is 86400 seconds.

Failover peer IP
Leave blank to disable. Enter the interface IP address of the other firewall (failover peer) in this subnet. Firewalls must be using CARP. Advertising skew of the CARP VIP on this interface determines whether the DHCP daemon is Primary or Secondary. Ensure the advertising skew for the VIP on one firewall is < 20 and the other is > 20.

Static ARP Enable Static ARP entries
Restricts communication with the firewall to only hosts listed in static mappings containing both IP addresses and MAC addresses. No other hosts will be able to communicate with the firewall on this interface. This behavior is enforced even when DHCP server is disabled.

Time format change Change DHCP display lease time from UTC to local time
By default DHCP leases are displayed in UTC time. By checking this box DHCP lease time will be displayed in local time and set to the time zone selected. This will be used for all DHCP interfaces lease time.

Statistics graphs Enable monitoring graphs for DHCP lease statistics
Enable this to add DHCP leases statistics to the Monitoring graphs. Disabled by default.

Ping check Disable ping check
When enabled dhcpcd sends a ping to the address being assigned, and if no response has been heard, it assigns the address. Enabled by default.

A continuación, vamos a **interfaces > assignments**, y vamos añadiendo hasta que quede como en la siguiente imagen y guardamos.

The screenshot shows the pfSense web interface under the 'Interfaces / Interface Assignments' section. It lists four interfaces: WAN, LAN, OPT1, and OPT2. Each interface is assigned to a specific network port (em0, em1, em2, em3) via a dropdown menu. Red delete buttons are present next to each assignment. A success message at the top states 'Interface has been added.' Below the table, a note says 'Interfaces that are configured as members of a lagg(4) interface will not be shown.' and 'Wireless interfaces must be created on the Wireless tab before they can be assigned.'

Ahora vamos a **interfaces > OPT1**, habilitamos la opción “Enable interface”, cambiamos la descripción a **DMZ**, ponemos la ipv4 en modo **estático**, ponemos la **nueva dirección IP** que aparece en el esquema de nuestra infraestructura (**192.168.200.1**) y cambiamos la **máscara de red** a “**/24**”, guardamos y aplicamos cambios.

The screenshot shows the pfSense configuration interface for interface OPT1. The 'General Configuration' section includes fields for 'Enable' (checked), 'Description' (set to 'DMZ'), 'IPv4 Configuration Type' (set to 'Static IPv4'), and 'MAC Address' (set to 'XX:XX:XX:XX:XX:XX'). Other fields include 'MTU', 'MSS', and 'Speed and Duplex'. The 'Static IPv4 Configuration' section shows the 'IPv4 Address' as '192.168.200.1' and the 'Subnet Mask' as '/24'. The 'IPv4 Upstream gateway' field is set to 'None'.

Si vamos a nuestro PFSENSE veremos que nos aparecen tres redes:

```
blue team - pc sense [Running] - Oracle VM VirtualBox  
pfSense 2.7.2-RELEASE amd64 20231206-2018  
Bootup complete  
FreeBSD/amd64 (UTM.keepcoding.local) (ttyv0)  
VirtualBox Virtual Machine - Netgate Device ID: 8702bd502d9544309884  
*** Welcome to pfSense 2.7.2-RELEASE (amd64) on UTM ***  
  
WAN (wan)      -> em0      -> v4/DHCP4: 192.168.1.39/24  
LAN (lan)      -> em1      -> v4: 192.168.100.1/24  
DMZ (opt1)     -> em2      -> v4: 192.168.200.1/24  
OPT2 (opt2)    -> em3      ->  
  
8) Logout (SSH only)          9) pfTop  
1) Assign Interfaces          10) Filter Logs  
2) Set interface(s) IP address 11) Restart webConfigurator  
3) Reset webConfigurator password 12) PHP shell + pfSense tools  
4) Reset to factory defaults   13) Update from console  
5) Reboot system               14) Enable Secure Shell (sshd)  
6) Halt system                 15) Restore recent configuration  
7) Ping host                   16) Restart PHP-FPM  
8) Shell  
  
Enter an option: |
```

Ahora vamos a **interfaces > OPT2** y hacemos la misma operación, pero en la descripción ponemos **DMZ2** y la **dirección IP** será “**192.168.250.1**”. Si volvemos al PFSENSE ya nos aparecerán las 4 redes con su respectiva IP y máscara de red:

```
blue team - pc sense [Running] - Oracle VM VirtualBox  
VirtualBox Virtual Machine - Netgate Device ID: 8702bd502d9544309884  
*** Welcome to pfSense 2.7.2-RELEASE (amd64) on UTM ***  
  
WAN (wan)      -> em0      -> v4/DHCP4: 192.168.1.39/24  
LAN (lan)      -> em1      -> v4: 192.168.100.1/24  
DMZ (opt1)     -> em2      -> v4: 192.168.200.1/24  
DMZ2 (opt2)    -> em3      -> v4: 192.168.250.1/24  
  
8) Logout (SSH only)          9) pfTop  
1) Assign Interfaces          10) Filter Logs  
2) Set interface(s) IP address 11) Restart webConfigurator  
3) Reset webConfigurator password 12) PHP shell + pfSense tools  
4) Reset to factory defaults   13) Update from console  
5) Reboot system               14) Enable Secure Shell (sshd)  
6) Halt system                 15) Restore recent configuration  
7) Ping host                   16) Restart PHP-FPM  
8) Shell  
  
Enter an option:  
Message from syslogd@UTM at Jan 22 20:29:52 ...  
php-fpm[398]: /index.php: Successful login for user 'admin' from: 192.168.200.10  
0 (Local Database)
```

A continuación, debemos ir a **Services > DHCP Server > DMZ**, habilitar la opción “**Enable DHCP server on DMZ interface**”, definimos el **rango** 192.168.200.100 - 192.168.200.150, y ponemos los **DNS servers** con dirección IP (192.168.200.1), cloudflare (1.1.1.1) y google (8.8.8.8).

General DHCP Options

- DHCP Backend:** ISC DHCP
- Enable:** Enable DHCP server on DMZ interface
- BOOTP:** Ignore BOOTP queries
- Deny Unknown Clients:** Allow all clients

When set to Allow all clients, any DHCP client will get an IP address within this scope/range on this interface. If set to Allow known clients from any interface, any DHCP client with a MAC address listed in a static mapping on **any** scope(s)/interface(s) will get an IP address. If set to Allow known clients from **only this interface**, only MAC addresses listed in static mappings on this interface will get an IP address within this scope/range.
- Ignore Denied Clients:** Ignore denied clients rather than reject

This option is not compatible with failover and cannot be enabled when a Failover Peer IP address is configured.
- Ignore Client Identifiers:** Do not record a unique identifier (UID) in client lease data if present in the client DHCP request

This option may be useful when a client can dual boot using different client identifiers but the same hardware (MAC) address. Note that the resulting server behavior violates the official DHCP specification.

Primary Address Pool

- Subnet:** 192.168.200.0/24
- Subnet Range:** 192.168.200.1 - 192.168.200.254
- Address Pool Range:** From 192.168.200.100 To 192.168.200.150

The specified range for this pool must not be within the range configured on any other address pool for this interface.
- Additional Pools:** [+ Add Address Pool](#)

If additional pools of addresses are needed inside of this subnet outside the above range, they may be specified here.

Server Options

- WINS Servers:** WINS Server 1, WINS Server 2
- DNS Servers:** 192.168.200.1, 1.1.1.1, 8.8.8.8
- DNS Server 4:** (empty)

Additional Pools

+ Add Address Pool

If additional pools of addresses are needed inside of this subnet outside of the above range, they may be specified here.

Server Options

- WINS Servers:** WINS Server 1, WINS Server 2
- DNS Servers:** 192.168.200.1, 1.1.1.1, 8.8.8.8
- DNS Server 4:** (empty)

OMAPI

- OMAPI Port:** OMAPI Port

Set the port that OMAPI will listen on. The default port is 7911, leave blank to disable. Only the first OMAPI configuration is used.
- OMAPI Key:** OMAPI Key

Enter a key matching the selected algorithm to secure connections to the OMAPI endpoint.

Generate New Key

Generate a new key based on the selected algorithm.
- Key Algorithm:** HMAC-SHA256 (current bind9 default)

Set the algorithm that OMAPI key will use.

Other DHCP Options

- Gateway:** 192.168.200.1

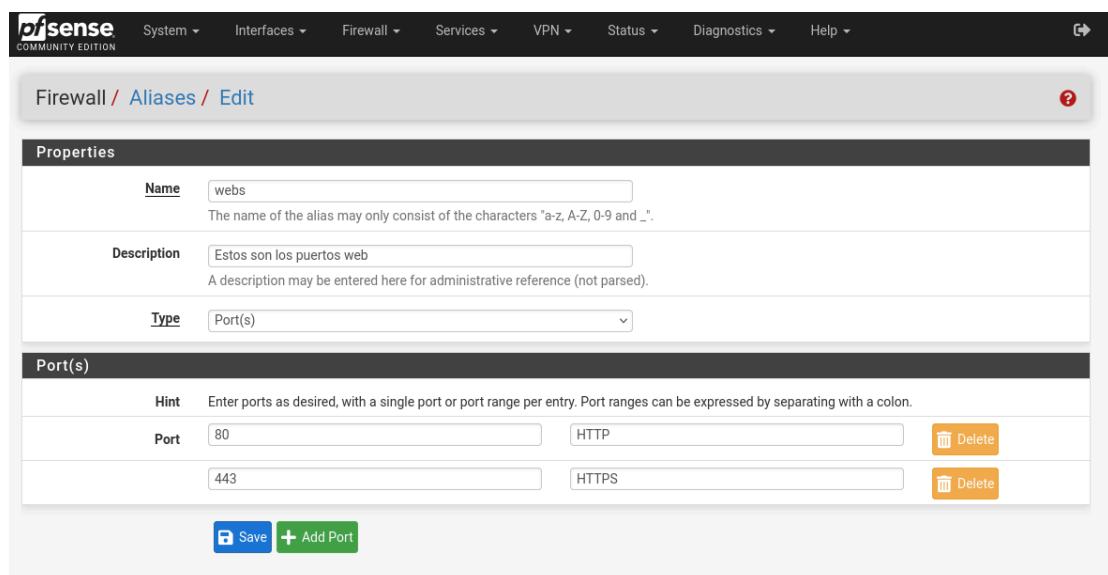
The default is to use the IP address of this firewall interface as the gateway. Specify an alternate gateway here if this is not the correct gateway for the network. Enter "none" for no gateway assignment.

En **Services > DHCP Server > DMZ2** hacemos lo mismo pero con el **rango** 192.168.250.100 - 192.168.250.150.

Si nos conectamos a las diferentes redes (machine > settings > network), por ejemplo la DMZ, y en la terminal ejecutamos “ip a”, veremos que tenemos la IP tal y como la definimos para el DMZ server y si abrimos el navegador veremos que no podemos acceder a páginas web ni tampoco al PFSENSE a través de la IP. Esto es debido a que PFSENSE por defecto cuando creamos distintas interfaces las reglas del firewall bloquean absolutamente todo el tráfico, excepto en este caso en la red LAN para poder conectarse a sí mismo al PFSENSE y hacer las configuraciones oportunas.

A continuación, vamos a crear las reglas del firewall para las redes DMZ y DMZ2 para que deje de ser tan restrictivo y nos deje pasar tráfico web ya que para estas dos redes ahora mismo no tenemos reglas que permitan que pase el tráfico lo que hace que el firewall rechace todo.

Accedemos a **Firewall > Aliases > Ports**, aquí vamos a permitir tráfico web y tráfico DNS, es decir, poder navegar, para ello usaremos los puertos 80 (HTTP) y 443 (HTTPS). Seleccionamos “ADD” y lo dejamos como en la siguiente imagen:



Ahora configuraremos las reglas del Firewall, para ello vamos a **Firewall > Rules > DMZ**, aquí le daremos al primer icono que pone “Add” para añadir la primera regla con protocolo TCP y dejaremos todo como en la siguiente imagen y guardamos.

The screenshot shows the 'Rules' configuration screen for the 'DMZ' zone. A new rule is being added for the 'TCP' protocol. The 'Source' section is set to 'Any' and 'Destination' is set to 'webs' (port 80). In the 'Extra Options' section, there is a checkbox for 'Log' which is unchecked. A description 'Regla trafico web' is entered. The 'Save' button at the bottom is highlighted.

Después añadimos otra regla, la del DNS, que es con **UDP**, dejándolo de la siguiente forma:

The screenshot shows the 'Rules' configuration screen for the 'DMZ' zone. A new rule is being added for the 'UDP' protocol. The 'Source' section is set to 'Any'. In the 'Destination' section, 'Protocol' is set to 'DNS (53)'. In the 'Extra Options' section, there is a checkbox for 'Log' which is unchecked. A description 'Regla DNS' is entered. The 'Save' button at the bottom is highlighted.

Una vez hecho esto accedemos a **Firewall > Rules > DMZ** y hacemos exactamente lo mismo añadiendo estas dos reglas anteriores. Y después deberíamos tener las reglas del Firewall de DMZ y DMZ2 de la siguiente manera:

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0/0 B	IPv4 UDP	*	*	*	53 (DNS)	*	none		Regla DNS	
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	*	*	*	webs	*	none		Regla tráfico web	

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0/0 B	IPv4 UDP	*	*	*	53 (DNS)	*	none		Regla DNS	
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	*	*	*	webs	*	none		Regla tráfico web	

La diferencia entre los protocolos TCP y UDP es que el TCP cuando envía algo espera a que le digan que ha llegado, si no lo seguirá mandando, como por ejemplo cuando quieras entrar en el correo electrónico mientras que el UDP no realiza esta espera, se usa por ejemplo para retransmisiones en directo ya que solo queremos que nos vaya mandando información de manera rápida.

Ahora si intentamos acceder a una página web veremos que ya es posible pero si en nuestra terminal hacemos “ping” a esa misma página (por ejemplo marca.com), veremos que no obtenemos respuesta esto es debido a que nos falta una regla, el http y https van por TCP, el DNS va por UDP y el ping va por ICMP, por lo cual debemos crear una nueva regla, iremos a **Firewall > Rules > DMZ** y añadiremos una regla con protocolo ICMP con los siguientes parámetros, una vez añadida haremos lo mismo para la **red DMZ2**:

whereas with block the packet is dropped silently. In either case, the original packet is discarded.

Disabled	<input type="checkbox"/> Disable this rule Set this option to disable this rule without removing it from the list.
Interface	DMZ Choose the interface from which packets must come to match this rule.
Address Family	IPv4 Select the Internet Protocol version this rule applies to.
Protocol	ICMP Choose which IP protocol this rule should match.
ICMP Subtypes	any Alternate Host Datagram conversion error Echo reply For ICMP rules on IPv4, one or more of these ICMP subtypes may be specified.
Source	
Source	<input type="checkbox"/> Invert match Any Source Address /
Destination	
Destination	<input type="checkbox"/> Invert match Any Destination Address /
Extra Options	
Log	<input type="checkbox"/> Log packets that are handled by this rule Hint: the firewall has limited local log space. Don't turn on logging for everything. If doing a lot of logging, consider using a remote syslog server (see the Status: System Logs: Settings page).
Description	Trafico ICMP A description may be entered here for administrative reference. A maximum of 52 characters will be used in the ruleset and displayed in the firewall log.

De esta manera las reglas quedarían así en este momento:

Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0/13 KIB any	IPv4 ICMP	*	*	*	*	*	none		Trafico ICMP	
<input type="checkbox"/>	✓ 0/438 KiB	IPv4 UDP	*	*	*	53 (DNS)	*	none		Regla DNS	
<input type="checkbox"/>	✓ 13/86.32 MiB	IPv4 TCP	*	*	*	webs	*	none		Regla trafico web	

Add Add Delete Toggle Copy Save Separator

Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0/0 B any	IPv4 ICMP	*	*	*	*	*	none		Trafico ICMP	
<input type="checkbox"/>	✓ 0/0 B	IPv4 UDP	*	*	*	53 (DNS)	*	none		Regla DNS	
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	*	*	*	webs	*	none		Regla trafico web	

Add Add Delete Toggle Copy Save Separator

Si hacemos ping veremos que ahora funciona:

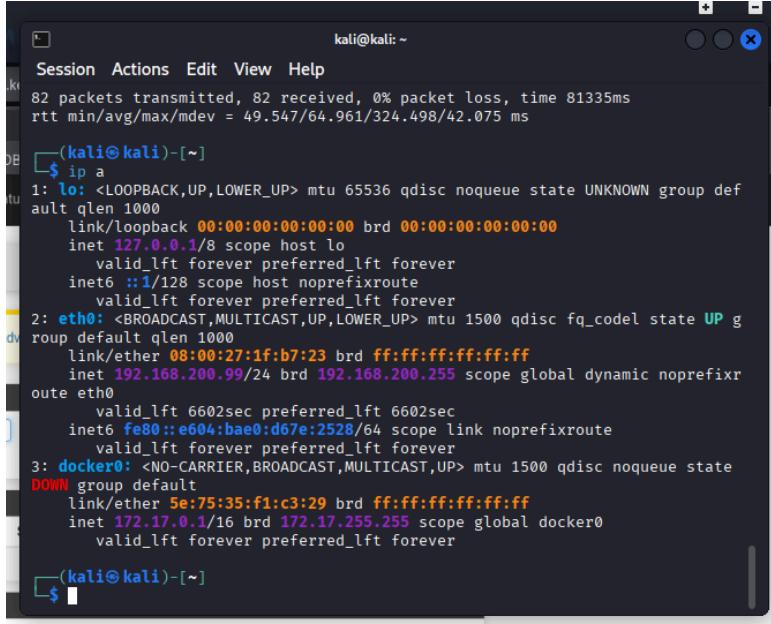
```
(kali㉿kali)-[~]
$ ping marca.com
PING marca.com (34.147.120.111) 56(84) bytes of data.
64 bytes from 111.120.147.34.bc.googleusercontent.com (34.147.120.111): icmp_seq=1 ttl=101 time=53.0 ms
64 bytes from 111.120.147.34.bc.googleusercontent.com (34.147.120.111): icmp_seq=2 ttl=101 time=69.0 ms
64 bytes from 111.120.147.34.bc.googleusercontent.com (34.147.120.111): icmp_seq=3 ttl=101 time=54.6 ms
64 bytes from 111.120.147.34.bc.googleusercontent.com (34.147.120.111): icmp_seq=4 ttl=101 time=60.8 ms
64 bytes from 111.120.147.34.bc.googleusercontent.com (34.147.120.111): icmp_seq=5 ttl=101 time=60.3 ms
64 bytes from 111.120.147.34.bc.googleusercontent.com (34.147.120.111): icmp_seq=6 ttl=101 time=56.5 ms
```

A continuación levantaremos un servidor **apache** para conectarnos a él, con una **dirección ip fija**, para eso la máquina que conectamos a la red LAN tendrá una IP fija. Para ello debemos ir a **Status > DHCP Leases** y en el apartado de Leases, vamos a actions y le damos al primer recuadro con un + (**add static mapping**), y aquí lo configuramos de la siguiente manera:

Static DHCP Mapping on DMZ	
DHCP Backend	ISC DHCP
MAC Address	08:00:27:1f:b7:23
	<input type="button" value="Copy My MAC"/>
MAC address of the client to match (6 hex octets separated by colons).	
Client Identifier	
An optional identifier to match based on the value sent by the client (RFC 2132).	
IP Address	192.168.200.99
IPv4 address to assign this client.	
Address must be outside of any defined pools. If no IPv4 address is given, one will be dynamically allocated from a pool. The same IP address may be assigned to multiple mappings.	
ARP Table Static Entry	<input checked="" type="checkbox"/> Create an ARP Table Static Entry for this MAC & IP Address pair.
Hostname	kali
Name of the client host without the domain part.	
Description	IP estatica
A description for administrative reference (not parsed).	
Server Options	
WINS Servers	WINS Server 1
	WINS Server 2
DNS Servers	192.168.200.1

Nosotros anteriormente le asignamos una IP dinámica a la red DMZ entre los rangos 192.168.200.100 - 192.168.200.150, por lo cual si ahora queremos asignar una IP fija no podemos hacerlo dentro de este rango ya que creariamos un conflicto, por lo cual escogemos una IP fuera de este rango que en este caso es 192.168.200.99, y gracias a la dirección MAC de nuestra Kali (MAC Address) lo que hará el servidor es que cuando nos conectemos con nuestra máquina nos asigne esa IP fija, y si se conecta otro dispositivo le

asignará una IP dentro del rango dinámico. Podemos comprobar que nos da esta IP fija haciendo “ip a” en la terminal:



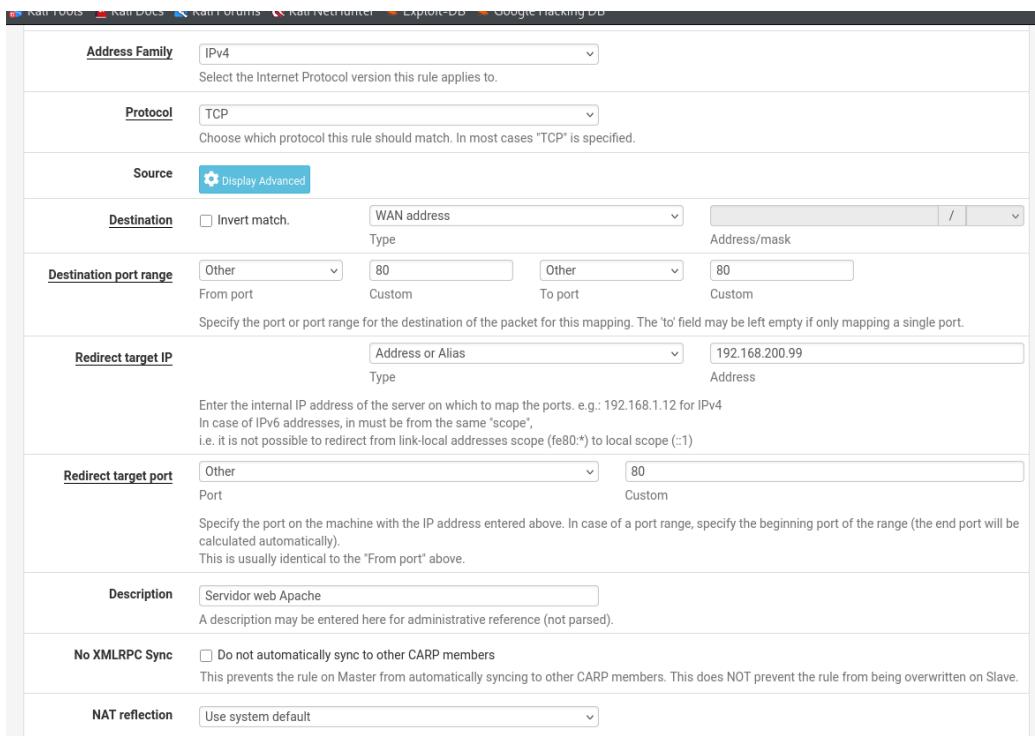
A screenshot of a terminal window titled "kali@kali: ~". The window shows the output of the command "ip a". The output lists three network interfaces: "lo" (loopback), "eth0" (ethernet), and "docker0". The "lo" interface has an IP of 127.0.0.1. The "eth0" interface has an IP of 192.168.200.99. The "docker0" interface has an IP of 172.17.0.1. The terminal prompt is "\$".

```
kali@kali: ~
82 packets transmitted, 82 received, 0% packet loss, time 81335ms
rtt min/avg/max/mdev = 49.547/64.961/324.498/42.075 ms

(kali㉿kali)-[~]
$ 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 08:00:27:1f:b7:23 brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.99/24 brd 192.168.200.255 scope global dynamic noprefixroute eth0
        valid_lft 6602sec preferred_lft 6602sec
        inet6 fe80::e604:bae0:d67e:2528/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 5e:75:35:f1:c3:29 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

(kali㉿kali)-[~]
$
```

Ahora vamos a **Firewall > Nat > Port Forward**, añadimos y ponemos los siguientes parámetros:



A screenshot of the Kali Linux Firewall configuration interface. The "Port Forward" tab is selected. A new rule is being added with the following parameters:

- Address Family:** IPv4
- Protocol:** TCP
- Source:** WAN address
- Destination:** WAN address (From port: 80, To port: 80)
- Redirect target IP:** 192.168.200.99
- Redirect target port:** 80
- Description:** Servidor web Apache
- No XMLRPC Sync:** Unchecked
- NAT reflection:** Use system default

Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions
WAN	TCP	*	*	WAN address	80 (HTTP)	192.168.200.99	80 (HTTP)	Servidor web Apache	

Legend:

Con esto lo que le estamos haciendo es que todo lo que entre a la red WAN desde el puerto 80 nos lo mande a la red DMZ.

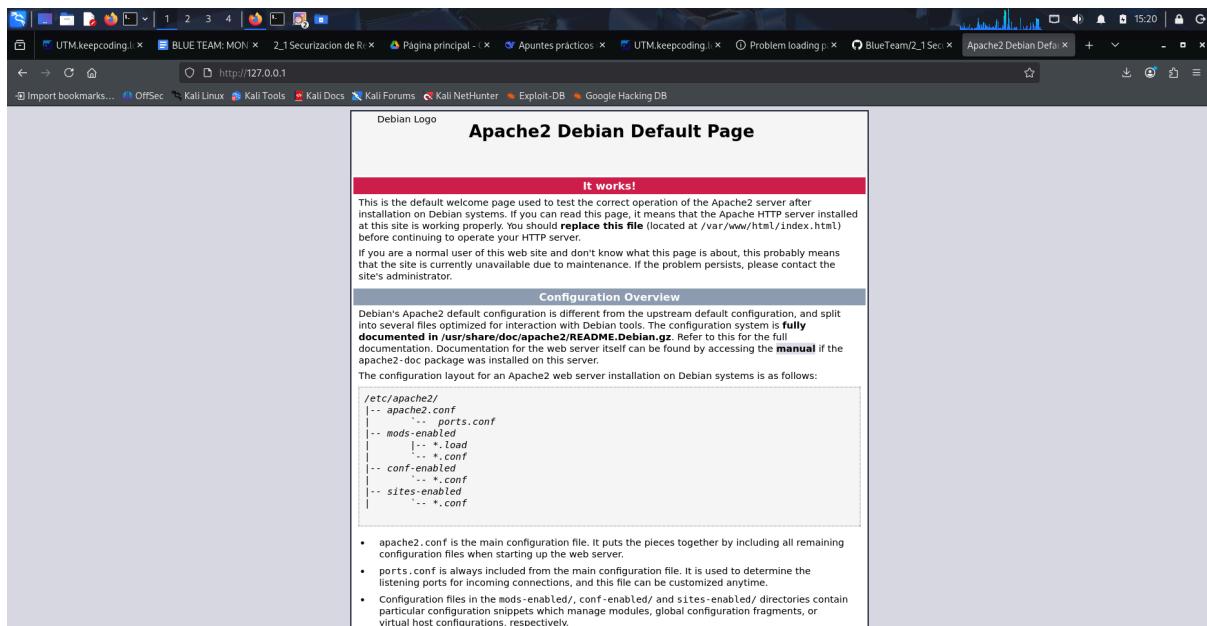
Ahora, nos vamos a la terminal y arrancamos un servidor apache con el comando (nos pedirá la contraseña de la kali):

- service apache2 start

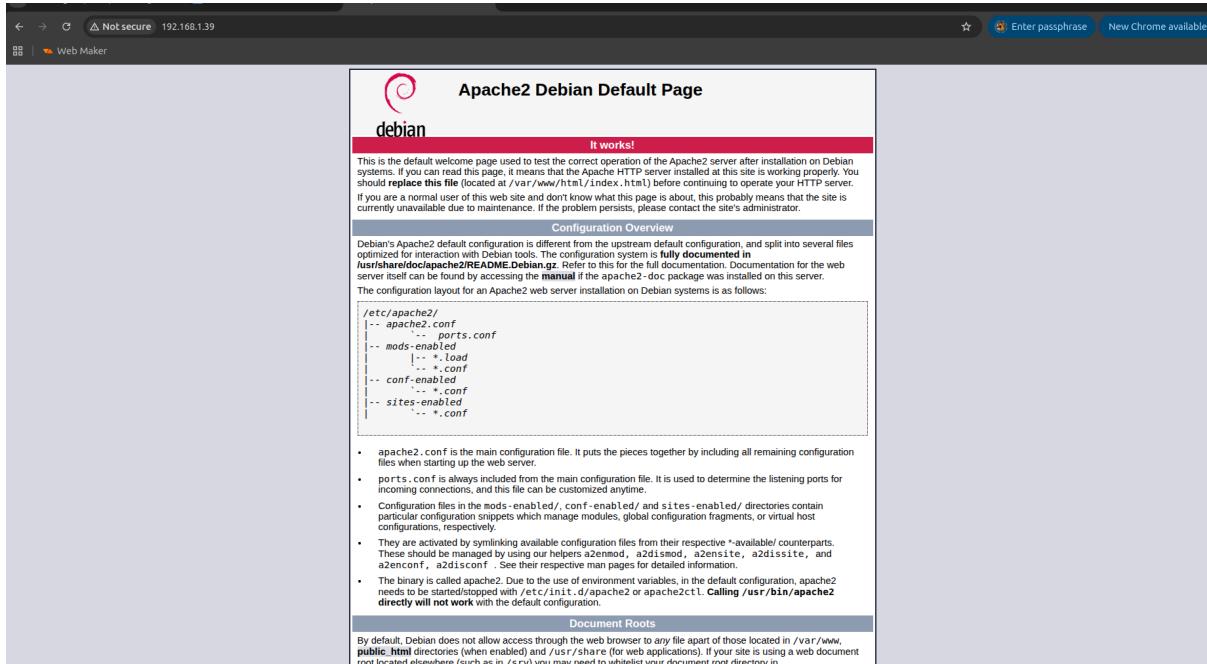
Comprobamos que ha funcionado con el comando:

- service apache2 status

Si vamos a la IP del localhost (127.0.0.1) veremos la default page de apache2:



Si vamos al navegador de la máquina física y ponemos la **IP de nuestra WAN** deberíamos ver la página, y si usamos otro dispositivo y ponemos esta IP deberíamos ver esta misma pagina tambien:



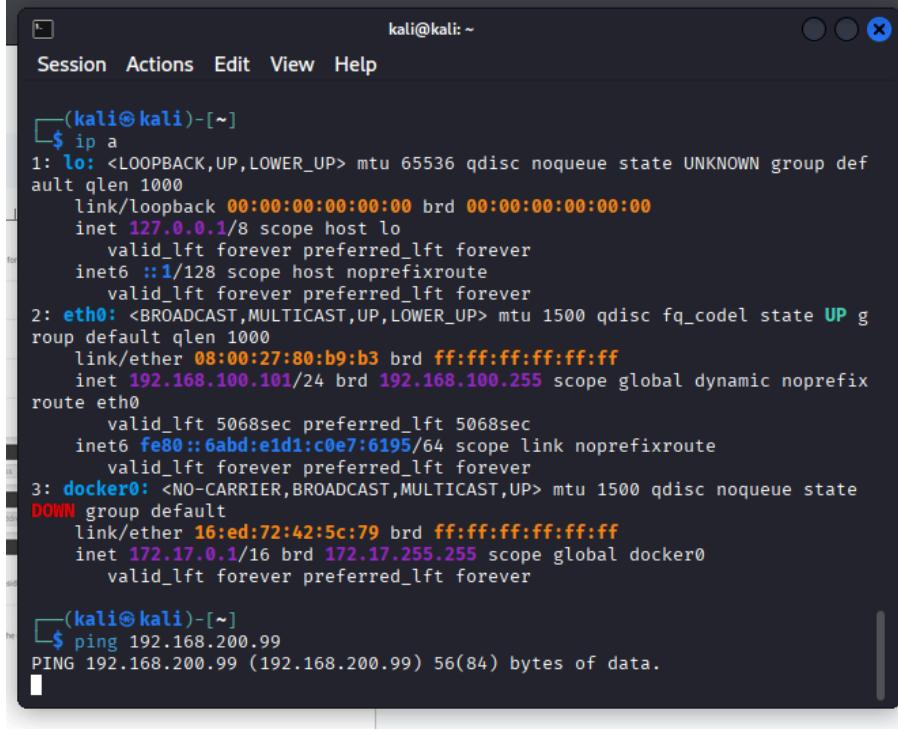
Bien, ahora debemos configurar las redes de tal manera que desde la red DMZ no se tenga acceso a la LAN ni a la red DMZ2 y viceversa, para lo cual debemos añadir reglas de firewall. Lo haremos de la siguiente manera:

- Añadimos una regla para bloquear el tráfico desde la LAN a la DMZ en Firewall > Rules > LAN.** Para ello, la acción debe ser “block”, el protocolo debe ser ICMP o Any en este caso el origen (source) son las subnets de la NAT es decir las IP que están dentro de la LAN (192.168.100.1), y el destino son las DMZ subnets.

The screenshot shows the "Edit Firewall Rule" interface. The rule is set to "Block". It has the following parameters:

- Action:** Block
- Disabled:** Disable this rule
- Interface:** LAN
- Address Family:** IPv4
- Protocol:** Any
- Source:** Source: LAN subnets
- Destination:** Destination: DMZ subnets
- Extra Options:**
 - Log:** Log packets that are handled by this rule
 - Description:** Bloqueo DMZ

Si hacemos ping desde una máquina conectada a la red LAN a la dirección IP de la red DMZ (192.168.200.99) veremos que no llega o que no obtenemos respuesta:



The screenshot shows a terminal window titled "kali@kali: ~". The window contains the following text:

```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    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
    link/ether 08:00:27:80:b9:b3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.101/24 brd 192.168.100.255 scope global dynamic noprefixroute
        route eth0
            valid_lft 5068sec preferred_lft 5068sec
        inet6 fe80::6abd:e1d1:c0e7:6195/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 16:ed:72:42:5c:79 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

(kali㉿kali)-[~]
$ ping 192.168.200.99
PING 192.168.200.99 (192.168.200.99) 56(84) bytes of data.
```

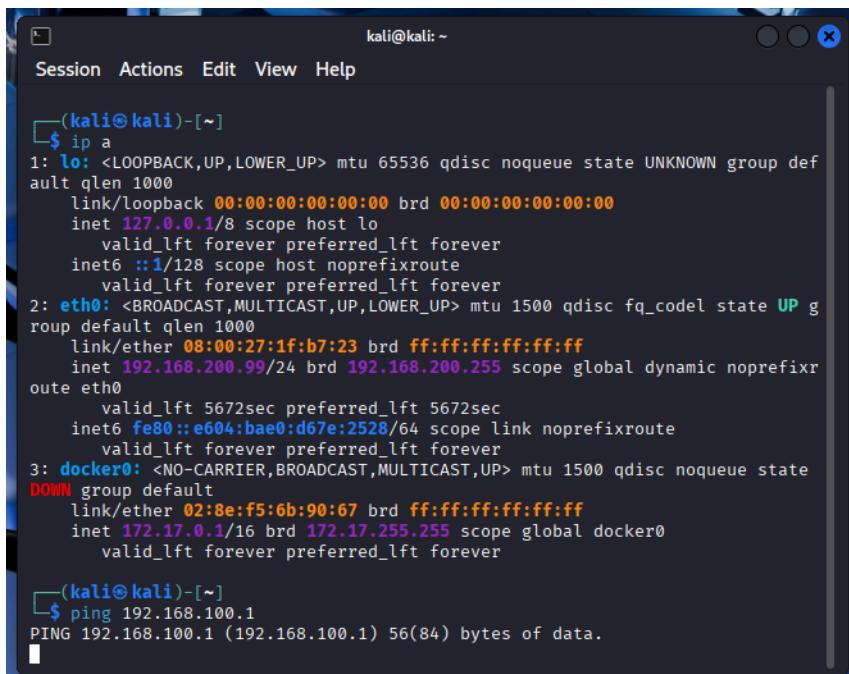
En mi caso para obtener otra máquina he clonado la Kali que ya tenía instalada.

2. **Añadimos una regla para bloquear el tráfico desde la DMZ a la LAN en Firewall > Rules > DMZ.** Para ello haremos lo mismo que antes, pero el origen será DMZ Subnets y el destino LAN Subnets.

Edit Firewall Rule

Action	Block	Choose what to do with packets that match the criteria specified below. Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.
Disabled	<input type="checkbox"/> Disable this rule	Set this option to disable this rule without removing it from the list.
Interface	DMZ	Choose the interface from which packets must come to match this rule.
Address Family	IPv4	Select the Internet Protocol version this rule applies to.
Protocol	Any	Choose which IP protocol this rule should match.
Source		
Source	<input type="checkbox"/> Invert match	DMZ subnets
Destination		
Destination	<input type="checkbox"/> Invert match	LAN subnets
Extra Options		
Log	<input type="checkbox"/>	Log packets that are handled by this rule Hint: the firewall has limited local log space. Don't turn on logging for everything. If doing a lot of logging, consider using a remote syslog server (see the Status: System Logs: Settings page).
Description	Bloqueo LAN	
A description may be entered here for administrative reference. A maximum of 52 characters will be used in the ruleset and displayed in the firewall log.		

Repetimos el proceso de hacer ping pero esta vez desde la máquina DMZ, haremos ping a la IP de la LAN (192.168.100.1) y veremos que no obtenemos respuesta.



```
(kali㉿kali)-[~]
$ 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 08:00:27:1f:b7:23 brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.99/24 brd 192.168.200.255 scope global dynamic noprefixroute
        valid_lft 5672sec preferred_lft 5672sec
        inet6 fe80::e604:bae0:d67e:2528/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:8e:f5:6b:90:67 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

(kali㉿kali)-[~]
$ ping 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
```

3. Añadimos una regla para bloquear el tráfico desde la DMZ2 a la DMZ para ello iremos a **Firewall > Rules > DMZ2**. Ponemos los mismos parámetros que antes, con origen DMZ2 Subnets y destino DMZ Subnets.

Firewall / Rules / Edit

Edit Firewall Rule

Action	Block
Choose what to do with packets that match the criteria specified below. Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.	
Disabled	<input type="checkbox"/> Disable this rule Set this option to disable this rule without removing it from the list.
Interface	DMZ2
Choose the interface from which packets must come to match this rule.	
Address Family	IPv4
Select the Internet Protocol version this rule applies to.	
Protocol	Any
Choose which IP protocol this rule should match.	
Source	
Source	<input type="checkbox"/> Invert match DMZ2 subnets
Source Address /	
Destination	
Destination	<input type="checkbox"/> Invert match DMZ subnets
Destination Address /	
Extra Options	
Log	<input type="checkbox"/> Log packets that are handled by this rule Hint: the firewall has limited local log space. Don't turn on logging for everything. If doing a lot of logging, consider using a remote syslog server (see

Ahora, la máquina Kali que teníamos conectada a la red LAN, debemos conectarla a la red DMZ2, hacemos ip a en la terminal para comprobar que estamos conectados a la IP de DMZ2 y despues hacemos ping a la IP de DMZ (192.168.200.99), veremos que no obtenemos respuesta.

```
kali㉿kali: ~
Session Actions Edit View Help
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 08:00:27:80:b9:b3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.250.101/24 brd 192.168.250.255 scope global dynamic noprefixroute
        route eth0
            valid_lft 7199sec preferred_lft 7199sec
            inet6 fe80::6abd:e1d1:c0e7:6195/64 scope link noprefixroute
                valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 16:ed:72:42:5c:79 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
└─(kali㉿kali)-[~]
    $ ping 192.168.200.99
PING 192.168.200.99 (192.168.200.99) 56(84) bytes of data.
^C
    192.168.200.99 ping statistics --
    303 packets transmitted, 0 received, 100% packet loss, time 309230ms

└─(kali㉿kali)-[~]
    $
```

4. Añadimos una regla para bloquear el tráfico desde la DMZ a la DMZ2, nos iremos a **Firewall > Rules > DMZ**. Como anteriormente, mismos parámetros con origen DMZ Subnets y destino DMZ2 Subnets.

The screenshot shows the 'Edit Firewall Rule' page in the pfSense web interface. The top navigation bar includes links for System, Interfaces, Firewall, Services, VPN, Status, Diagnostics, and Help. The main title is 'Firewall / Rules / Edit'. The rule being edited has the following parameters:

- Action:** Block
- Disabled:** Disable this rule
- Interface:** DMZ
- Address Family:** IPv4
- Protocol:** Any
- Source:** Source: DMZ subnets, Destination Address: /
- Destination:** Destination: DMZ2 subnets, Destination Address: /
- Extra Options:** Log: Log packets that are handled by this rule

Ahora si vamos a la terminal de la Kali que tenemos conectada a la red DMZ y hacemos ping a la IP de la red DMZ2 (192.168.250.1) veremos que no obtenemos respuesta.

```

kali㉿kali: ~
Session Actions Edit View Help

└──(kali㉿kali)-[~]
$ 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 08:00:27:1f:b7:23 brd ff:ff:ff:ff:ff:ff
        inet 192.168.200.99/24 brd 192.168.200.255 scope global dynamic noprefixroute
            valid_lft 4090sec preferred_lft 4090sec
            inet6 fe80::e604:bae0:d67e:2528/64 scope link noprefixroute
                valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:8e:f5:6b:90:67 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

└──(kali㉿kali)-[~]
$ ping 192.168.250.1
PING 192.168.250.1 (192.168.250.1) 56(84) bytes of data.
^C
--- 192.168.250.1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3071ms

└──(kali㉿kali)-[~]
$ 

```

Actualizamos como deberían estar las **reglas del Firewall** en WAN, LAN, DMZ y DMZ2:

Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	0/0 B	IPv4 TCP	*	*	192.168.200.99	80 (HTTP)	*	none		NAT Servidor web Apache	

Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	0/476 KiB	*	*	*	LAN Address	443 80	*	*		Anti-Lockout Rule	
<input type="checkbox"/>	0/7 KiB	IPv4 *	LAN subnets	*	DMZ subnets	*	*	none		Bloqueo DMZ	
<input checked="" type="checkbox"/>	0/35.27 MiB	IPv4 *	LAN subnets	*	*	*	*	none		Default allow LAN to any rule	
<input checked="" type="checkbox"/>	0/0 B	IPv6 *	LAN subnets	*	*	*	*	none		Default allow LAN IPv6 to any rule	

Rules (Drag to Change Order)											Actions
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✗ 0/756 B	IPv4 *	DMZ subnets	*	DMZ2 subnets	*	*	*	none	Bloqueo LAN	
<input type="checkbox"/>	✗ 0/106 KiB	IPv4 *	DMZ subnets	*	LAN subnets	*	*	*	none	Trafico ICMP	
<input type="checkbox"/>	✓ 0/3 KiB	IPv4 ICMP any	*	*	*	*	*	*	none	Regla DNS	
<input type="checkbox"/>	✓ 0/62 KiB	IPv4 UDP	*	*	*	53 (DNS)	*	none	Regla trafico web		
<input type="checkbox"/>	✓ 13/25.68 MiB	IPv4 TCP	*	*	*	webs	*	none	Regla trafico web		

Rules (Drag to Change Order)											Actions
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✗ 0/25 KiB	IPv4 *	DMZ2 subnets	*	DMZ subnets	*	*	*	none	Trafico ICMP	
<input type="checkbox"/>	✓ 0/0 B	IPv4 ICMP any	*	*	*	*	*	*	none	Regla DNS	
<input type="checkbox"/>	✓ 1/21 KiB	IPv4 UDP	*	*	*	53 (DNS)	*	none	Regla trafico web		
<input type="checkbox"/>	✓ 8/6.93 MiB	IPv4 TCP	*	*	*	webs	*	none	Regla trafico web		

A continuación, vamos a **Interfaces > OPT1** y cambiamos los siguientes parámetros:

Interfaces / DMZ (em2)

General Configuration

Enable	<input checked="" type="checkbox"/> Enable interface
Description	DMZ
IPv4 Configuration Type	Static IPv4
IPv6 Configuration Type	None
MAC Address	XXXXXXXXXXXXXX
MTU	1500
MSS	1460
Speed and Duplex	Default (no preference, typically autoselect)

Static IPv4 Configuration

IPv4 Address	192.168.200.1	/ 24
IPv4 Upstream gateway	None	+ Add a new gateway

If this interface is an Internet connection, select an existing Gateway from the list or add a new one using the "Add" button.
On local area network interfaces the upstream gateway should be "none".
Selection an upstream gateway causes the firewall to treat this interface as a WAN link interface.

Vamos a instalar ahora en la máquina Kali conectada a la DMZ el honeypot, para ello abrimos la terminal y lo primero que debemos hacer es **instalar docker** con los comandos:

- **sudo apt update**
- **sudo apt install -y docker.io**

Ahora ejecutaremos el **honeypot de cowrie** que simulará un servicio de **SSH** con el comando:

- **sudo docker run -p 222:2222 cowrie/cowrie**

Debemos configurar una regla en **Firewall > NAT > Port Forward**, que nos permitirá conectarnos desde ssh, puerto 222, al honeypot, básicamente con esta regla, al conectarte a la ip de la WAN, nos redirige al honeypot:

Edit Redirect Entry

Disabled	<input type="checkbox"/> Disable this rule
No RDR (NOT) <input type="checkbox"/> Disable redirection for traffic matching this rule This option is rarely needed. Don't use this without thorough knowledge of the implications.	
Interface	WAN
Choose which interface this rule applies to. In most cases "WAN" is specified.	
Address Family	IPv4
Select the Internet Protocol version this rule applies to.	
Protocol	TCP
Choose which protocol this rule should match. In most cases "TCP" is specified.	
Source	Display Advanced
Destination	<input type="checkbox"/> Invert match. <input type="checkbox"/> WAN address <input type="text"/> / <input type="button"/>
Type Address/mask	
Destination port range	From port <input type="text"/> To port <input type="text"/> Other Custom Other Custom
Specify the port or port range for the destination of the packet for this mapping. The 'to' field may be left empty if only mapping a single port.	
Redirect target IP	<input type="text"/> Address or Alias <input type="text"/> 192.168.200.99 Type Address
Enter the internal IP address of the server on which to map the ports. e.g.: 192.168.1.12 for IPv4 In case of IPv6 addresses, it must be from the same "scope", i.e. it is not possible to redirect from link-local addresses scope (fe80::*) to local scope (::1)	
Redirect target port	<input type="text"/> Port <input type="text"/> Custom Other Custom
Specify the port on the machine with the IP address entered above. In case of a port range, specify the beginning port of the range (the end port will be calculated automatically)	

Specify the port or port range for the destination of the packet for this mapping. The 'to' field may be left empty if only mapping a single port.

Redirect target IP	<input type="text" value="Address or Alias"/> Address or Alias	<input type="text" value="192.168.200.99"/> Type
Enter the internal IP address of the server on which to map the ports. e.g.: 192.168.1.12 for IPv4 In case of IPv6 addresses, it must be from the same "scope", i.e. it is not possible to redirect from link-local addresses scope (fe80::) to local scope (::)		
Redirect target port	<input type="text" value="Other"/> Port	<input type="text" value="222"/> Custom
Specify the port on the machine with the IP address entered above. In case of a port range, specify the beginning port of the range (the end port will be calculated automatically). This is usually identical to the "From port" above.		
Description	<input type="text" value="Regla Honeypot ssh 222"/>	
A description may be entered here for administrative reference (not parsed).		
No XMLRPC Sync	<input type="checkbox"/> Do not automatically sync to other CARP members	
This prevents the rule on Master from automatically syncing to other CARP members. This does NOT prevent the rule from being overwritten on Slave.		
NAT reflection	<input type="text" value="Use system default"/>	
Filter rule association	<input type="text" value="Rule NAT Regla Honeypot ssh 222"/>	
View the filter rule		
Rule Information		
Created	1/28/26 21:48:00 by admin@192.168.200.99 (Local Database)	
Updated	1/28/26 21:48:00 by admin@192.168.200.99 (Local Database)	
<input type="button" value="Save"/>		

Así nos quedaría finalmente:

The screenshot shows the pfSense interface under the Firewall / NAT / Port Forward tab. There are two entries in the 'Rules' table:

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

At the bottom, there are buttons for Add, Delete, Toggle, Save, and Separator.

Ahora debemos configurar una regla en **Firewall > Rules > WAN**, que salvo sorpresa, al hacer la regla del port forwarding anteriormente ya estaría creada y quedaria así:

The screenshot shows the pfSense interface under the Firewall / Rules / WAN tab. The 'WAN' tab is selected. There are two entries in the 'Rules' table:

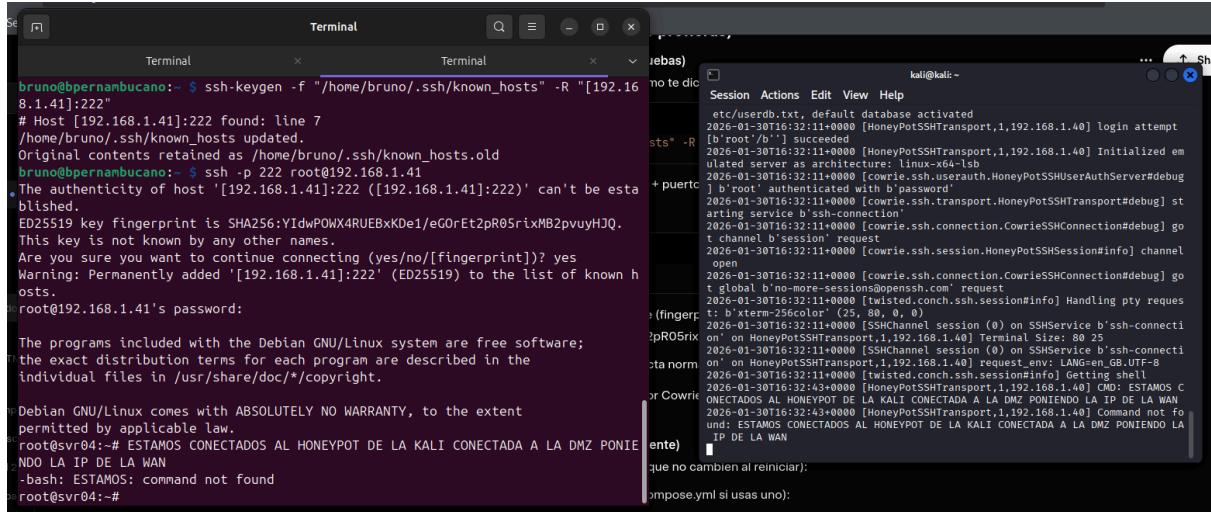
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	0/0 B	IPv4 TCP	*	*	192.168.200.99	80 (HTTP)	*	none		NAT Servidor web Apache	
<input checked="" type="checkbox"/>	0/16 KIB	IPv4 TCP	*	*	192.168.200.99	222	*	none		NAT Regla Honeypot ssh 222	

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

Para comprobar que funciona, iremos a la **terminal de nuestra máquina física**, y a través de este comando nos conectaremos mediante ssh:

- **ssh -p 222 root@(dirección WAN)**

Si ejecutamos distintos comandos en la máquina física podremos ver los **logs** en la máquina virtual.



The screenshot shows two terminal windows. The left window is titled 'Terminal' and shows the command: `ssh-keygen -f "/home/bruno/.ssh/known_hosts" -R "[192.168.1.41]:222"`. It outputs: '# Host [192.168.1.41]:222 found: line 7 /home/bruno/.ssh/known_hosts updated. Original contents retained as /home/bruno/.ssh/known_hosts.old'. Then it shows: `ssh -p 222 root@192.168.1.41`. The right window is titled 'kali@kali: ~' and shows log entries from a HoneyPot SSH server. One entry is: '2026-01-30T16:32:11+0000 [HoneyPotSSHTransport,1,192.168.1.40] login attempt [b'root'@b'] succeeded'. Another entry is: '2026-01-30T16:32:11+0000 [HoneyPotSSHTransport,1,192.168.1.40] Initialized emulated server as architecture: linux-x64-lsb'. The logs continue with various connection attempts and session details.

A continuación, nos cambiaremos a otra máquina Kali, conectada a la red **DMZ2** para instalar **suricata**, para ello ejecutaremos los comandos:

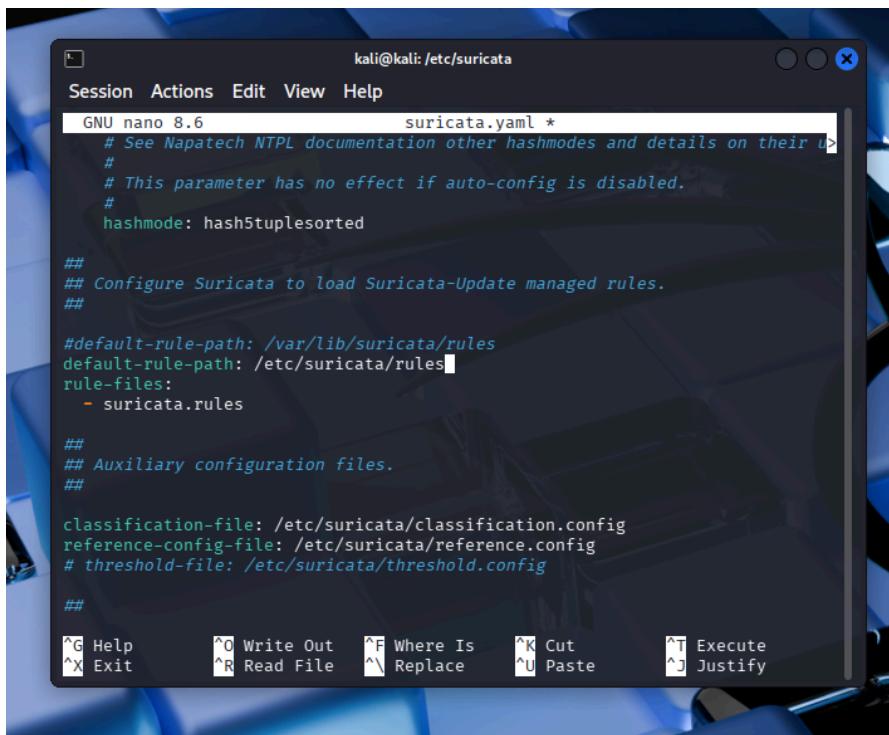
- **sudo apt update**
- **sudo apt install suricata**
- **sudo suricata -c /etc/suricata/suricata.yaml -i eth0**

Matamos el anterior comando y hacemos un "cd /etc/suricata" y aquí entramos a la carpeta rules (cd rules) y creamos un archivo que se llame **suricata.rules** (sudo touch suricata.rules).

Después lo abrimos para editarlo (sudo nano suricata.rules) y escribimos la siguiente línea:

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

Ahora retrocedemos a la carpeta de suricata (cd ..) y modificaremos el archivo **suricata.yaml** para que quede de la siguiente manera, comentando el default rule path que viene por defecto y poniendo el nuestro con la regla que acabamos de crear:



```

kali@kali: /etc/suricata
Session Actions Edit View Help
GNU nano 8.6          suricata.yaml *
# See Napatech NTPL documentation other hashmodes and details on their usage.
#
# This parameter has no effect if auto-config is disabled.
#
hashmode: hash5tuplesorted

## Configure Suricata to load Suricata-Update managed rules.
##
## default-rule-path: /var/lib/suricata/rules
default-rule-path: /etc/suricata/rules
rule-files:
- suricata.rules

## Auxiliary configuration files.
##

classification-file: /etc/suricata/classification.config
reference-config-file: /etc/suricata/reference.config
# threshold-file: /etc/suricata/threshold.config

##
^G Help      ^O Write Out    ^F Where Is    ^K Cut        ^T Execute
^X Exit      ^R Read File   ^M Replace    ^U Paste      ^J Justify

```

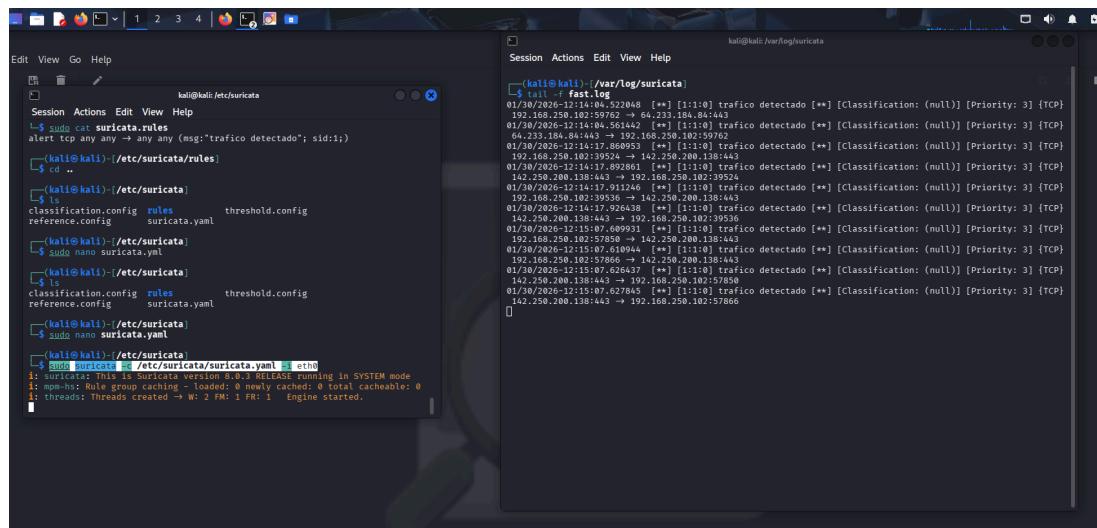
Ahora ejecutamos otra vez este comando en la terminal:

- **sudo suricata -c /etc/suricata/suricata.yaml -i eth0**

Y en otra terminal ejecutamos este otro comando (estando en la ruta var/log/suricata):

- **tail -f fast.log**

Podremos ver los **logs** que estamos generando en la terminal, en los cuales aparecerá entre otras cosas la frase “tráfico detectado”.



```

kali@kali: /var/log/suricata
Session Actions Edit View Help
Edit View Go Help
kali@kali: /etc/suricata
$ tail -f fast.log
01/30/2026-12:14:04.522046 [**] [1:1:0] tráfico detectado [**] [Classification: (null)] [Priority: 3] [TCP]
192.168.250.192:59762 → 192.168.250.64:23 [1:1:0] tráfico detectado [**] [Classification: (null)] [Priority: 3] [TCP]
01/30/2026-12:14:04.544451 [**] [1:1:0] tráfico detectado [**] [Classification: (null)] [Priority: 3] [TCP]
64.233.184.84:445 → 192.168.250.102:59762
01/30/2026-12:14:17.869953 [**] [1:1:0] tráfico detectado [**] [Classification: (null)] [Priority: 3] [TCP]
192.168.250.192:39574 → 192.168.250.102:445
01/30/2026-12:14:17.924086 [**] [1:1:0] tráfico detectado [**] [Classification: (null)] [Priority: 3] [TCP]
142.250.200.138:443 → 192.168.250.102:9524
01/30/2026-12:14:17.911249 [**] [1:1:0] tráfico detectado [**] [Classification: (null)] [Priority: 3] [TCP]
192.168.250.192:59762 → 192.168.250.102:443
01/30/2026-12:14:17.924431 [**] [1:1:0] tráfico detectado [**] [Classification: (null)] [Priority: 3] [TCP]
142.250.200.138:443 → 192.168.250.102:9536
01/30/2026-12:15:07.609931 [**] [1:1:0] tráfico detectado [**] [Classification: (null)] [Priority: 3] [TCP]
192.168.250.192:59762 → 192.168.250.102:443
01/30/2026-12:15:07.610944 [**] [1:1:0] tráfico detectado [**] [Classification: (null)] [Priority: 3] [TCP]
192.168.250.192:57866 → 142.250.200.138:443
01/30/2026-12:15:07.626437 [**] [1:1:0] tráfico detectado [**] [Classification: (null)] [Priority: 3] [TCP]
142.250.200.138:443 → 192.168.250.102:57866
01/30/2026-12:15:07.627845 [**] [1:1:0] tráfico detectado [**] [Classification: (null)] [Priority: 3] [TCP]
142.250.200.138:443 → 192.168.250.102:57866

```

Después, en la ruta etc/suricata, modificamos el archivo **classification.config**, y añadimos la siguiente línea al final:

- config classification: file-download,Descarga de archivo detectado,2

```

classification.config
classification: not-suspicious,Not Suspicious Traffic,3
classification: unknown,Unknown Traffic,3
classification: bad-unknown,Potentially Bad Unknown Traffic,2
classification: successful-recon,Successful Reconnaissance,2
classification: successful-recon-limited,Information Leak,2
classification: successful-recon-large-scale,Large Scale Information Leak,2
classification: attempted-dos,Attempted Denial of Service,2
classification: attempted-user,Attempted User Privilege Gain,1
classification: unsuccessful-user,Unsuccessful User Privileged Gain,1
classification: successful-user,Successful User Privileged Gain,1
classification: attempted-admin,Attempted Administrator Privileged Gain,1
classification: successful-admin,Successful Administrator Privileged Gain,1

e NEW CLASSIFICATIONS
config classification: rpc-portman-decode,Decode an RPC Query,2
config classification: shellcode-detect,Executable code was detected,1
config classification: string-detect,A suspicious string was detected,3
config classification: suspicious-login,An attempted login using a suspicious username was detected,2
config classification: system-call-detect,A system call was detected,2
config classification: tcp-connection,A TCP connection was detected,4
config classification: unusual-client,An unusual client was detected,1
config classification: unusual-client-port-connection,A client was using an unusual port,2
config classification: network-scan,Detection of a Network Scan,3
config classification: denial-of-service,Detection of a Denial of Service Attack,2
config classification: protocol-command-decode,Protocol Command Detection or event,2
config classification: protocol-command-decode,Generic Protocol Command Decode,3
config classification: web-application-activity,access to a potentially vulnerable web application,2
config classification: web-application-attack,Web Application Attack,1
config classification: external-ip-check,External IP Address Detected,2
config classification: external-ip-change,External IP Address Has Changed,1
config classification: pup-activity,Possibly Unwanted Program Detected,2
config classification: credential-theft,Successful Credential Theft Detected,1
config classification: social-engineering,Possible Social Engineering Attempted,2
config classification: crypto-mining,Crypto Currency Mining Activity Detected,2
config classification: command-and-control,Malware Command and Control Activity Detected,1

e Update
config classification: targeted-activity,Targeted Malicious Activity Was Detected,1
config classification: exploit-kit,Exploit Kit Activity Detected,1
config classification: external-ip-check,Device Retrieving External IP Address Detected,2
config classification: external-ip-change,External IP Address Has Changed,1
config classification: pup-activity,Possibly Unwanted Program Detected,2
config classification: credential-theft,Successful Credential Theft Detected,1
config classification: social-engineering,Possible Social Engineering Attempted,2
config classification: crypto-mining,Crypto Currency Mining Activity Detected,2
config classification: command-and-control,Malware Command and Control Activity Detected,1

e Classification profile
config classification: file-download,Descarga de archivo detectado,2

```

Después modificamos en la misma ruta el archivo **suricata.yaml** para que quede así:

```

# eve-log. If write-fieinfo is set to yes, then each file will have
# one more associated .json files that consist of the fileinfo
# record. A fileinfo file will be written for each occurrence of the
# file seen using a filename suffix to ensure uniqueness.
#
# To prune the filestore directory see the "suricatactl filestore
# prune" command which can delete files over a certain age.
- file-store:
    version: 2
    enabled: yes
    dir: /etc/suricata/files
    write-fieinfo: yes

    # Set the directory for the filestore. Relative pathnames
    # are contained within the "default-log-dir".
    #dir: filestore

```

Ahora si descargamos un archivo pdf, en una página con **http** (no vale https),veremos que en los logs nos sale “archivo pdf detectado”.

Session Actions Edit View Help

```

142.250.200.138:443 → 192.168.250.102:57850
01/30/026-12:15:07.609931 [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP}
142.250.200.138:443 → 192.168.250.102:57866
01/30/026-12:15:07.610155 [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP}
34.107.243.93:443 → 192.168.250.102:57823
01/30/026-12:18:15.795756 [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP}
192.168.250.102:57479 → 34.107.243.93:443
01/30/026-12:18:15.795756 [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP}
192.168.250.102:37100 → 151.101.133.91:443
01/30/026-12:18:15.884743 [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP}
34.168.144.191:443 → 192.168.250.102:47548
```

```

(Kali㉿kali)-[~] cd /var/log/suricata

```

01/30/026-12:15:07.609931 [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP}
192.168.250.102:57850 → 142.250.200.138:443
01/30/026-12:15:07.610044 [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP}
192.168.250.102:57866 → 142.250.200.138:443
01/30/026-12:15:07.610155 [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP}
142.250.200.138:443 → 192.168.250.102:57856
01/30/026-12:15:07.627845 [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP}
142.250.200.138:443 → 192.168.250.102:57856
01/30/026-12:18:15.795756 [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP}
34.107.243.93:443 → 192.168.250.102:57826
01/30/026-12:18:15.795756 [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP}
192.168.250.102:37100 → 151.101.133.91:443
01/30/026-12:18:15.884743 [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP}
34.168.144.191:443 → 192.168.250.102:47548
```

```

File: suricata.log

1 of 31

http://informatica.uv.es/iiguia/IST/Tema2.pdf

OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

Tema 2: Protocolo HTTP.

1. Introducción.

2. Mensajes HTTP.

- Partes del mensaje.
- Primer linea del mensaje.
- Cabeceras del mensaje.
- Cuerpo del mensaje.

3. Elementos Avanzados.

- Cookies.
- Manejo de sesiones.
- Autenticación y Autorización del cliente.
- Seguridad.
- Conexiones persistentes.
- Caché.

IST - 2006

HTTP

1

A continuación, necesitamos crear una cuenta en **Elastic**, por lo que iremos a la página de elastic en nuestro navegador de la maquina fisica (<https://www.elastic.co/es>) y accedemos a “hacer una prueba gratis”, usamos una cuenta de correo y completamos el registro. Cuando lleguemos a la parte que nos dice “Which use can you looking to try first?” escogemos la que dice “Elastic for Security” > SIEM and Security Analytics > No > Elastic Cloud Serverless > Launch.

Ahora, en el menú de la izquierda, vamos a **More > Assets > Policies**, aquí se configuran las políticas para crear los logs en elastic. Empezaremos creando la primera en “**Create agent policy**” > **Name: Politica Linux Suricata > Create agent policy**.

Si entramos a esta política, veremos que están puestos los logs del sistema por defecto, pero la idea es que se recojan los logs de suricata por lo que le daremos a la opción “**Add integration**” > **Integration: Suricata > Add integration**.

Bien, en este punto tenemos definidas las reglas que van a seguir todas las máquinas que estén definidas dentro del grupo de política linux suricata, ahora lo que falta es meter la máquina, para lo cual, le daremos a la opción “**Add agent**”, nos aparecerá más abajo un cuadro de texto con un comando, en este cuadro veremos en la parte superior izquierda un menú desplegable, escogemos **linux 86_64**, copiamos el comando y lo pegamos en la máquina kali donde tenemos suricata y comenzamos una **instalación**, si volvemos a elastic y esperamos un poco veremos que se ponen en verde las opciones de “Agent enrolled confirmed” y “Incoming data confirmed”.

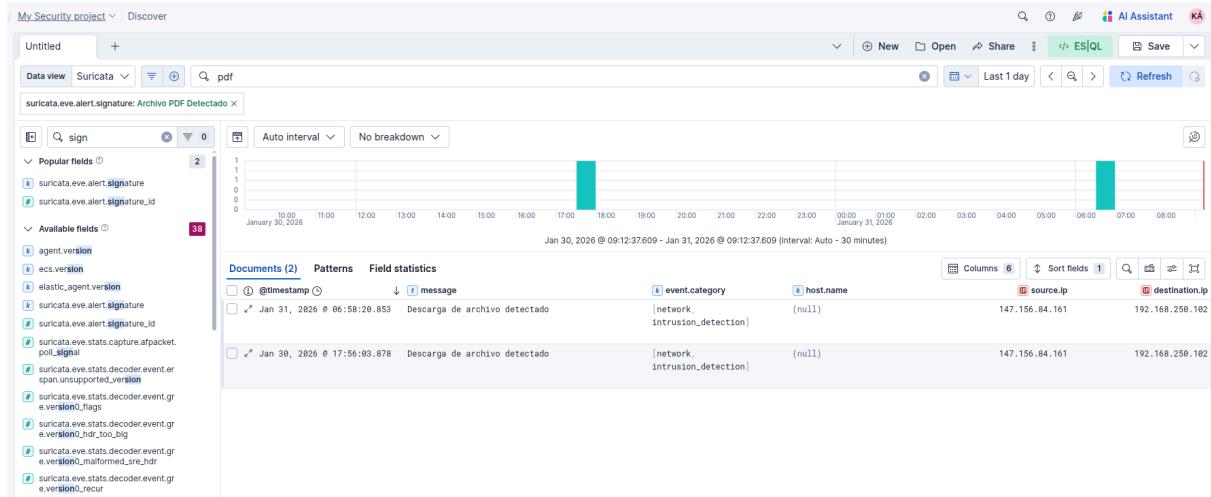
Integration policy	Integration	Namespace	Output	Actions
suricata-1	Suricata v2.27.0	default	Default output	
system-1	System v2.12.0	default	Default output	

Ahora en el menú de la izquierda vamos a “Discover”, en la parte superior izquierda en “data view”, abrimos el menú desplegable y seleccionamos “create data view”:

- Name: Suricata

- Index pattern: *suricata*

Ahora seleccionando este filtro, podemos poner arriba también filtro pdf, o seleccionar en el menú de la izquierda sign y coger suricata.eve.alert.signature y dentro de este añadir el que pone “pdf detectado”. Importante mirar



Si nos fijamos donde pone “[source.ip](#)” es la ip desde donde descargamos el archivo pdf y [destination.ip](#) es nuestra ip de la máquina virtual kali, podemos comprobarlo haciendo “ip a” en la máquina kali.

```
kali@kali: ~/elastic-agent-9.2.4-linux-x86_64
Session Actions Edit View Help
{"log.level": "info", "@timestamp": "2026-01-31T03:40:33.759-0500", "log.origin": {"function": "github.com/elastic/elastic-agent/internal/pkg/agent/cmd.(*enroll Cmd).Execute", "file.name": "cmd/enroll_cmd.go", "file.line": 207}, "message": "Successfully triggered restart on running Elastic Agent.", "ecs.version": "1.6.0"}}
Successfully enrolled the Elastic Agent.
[== ] Done [19s]
Elastic Agent has been successfully installed.

└─(kali㉿kali)-[~/elastic-agent-9.2.4-linux-x86_64]
└─$ 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 08:00:27:63:b0:05 brd ff:ff:ff:ff:ff:ff
    inet 192.168.250.102/24 brd 192.168.250.255 scope global dynamic noprefixroute
        valid_lft 4973sec preferred_lft 4973sec
    inet6 fe80::48a0:b951:3cb0:ea30/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

└─(kali㉿kali)-[~/elastic-agent-9.2.4-linux-x86_64]
```

También podemos ver el archivo json del log, en él es importante fijarse en que nos aparece nuestra ip de la máquina kali, la ruta que marcamos e incluso la regla que creamos. (**VER ANEXO 1, JSON LOG SURICATA**).

Ahora volvemos a **More > Assets > Policies**, añadimos una política con nombre Windows y otra con nombre Honeypot. Debería quedarnos así:

The screenshot shows the Fleet interface for managing Elastic Agents. The top navigation bar includes links for 'Agents', 'Agent policies' (which is the active tab), 'Enrollment tokens', 'Uninstall tokens', 'Data streams', and 'Settings'. A search bar at the top right allows filtering data using KQL syntax. Below the navigation is a message: 'We've added new privileges that let you define more granularly who can view or edit Fleet agents, policies, and settings. Learn more.' The main section is titled 'Fleet' and contains a table of agent policies. The columns are 'Name', 'Last updated on', 'Unprivileged / Privileged', 'Integrations', and 'Actions'. Three entries are listed:

Name	Last updated on	Unprivileged / Privileged	Integrations	Actions
Honeypot rev. 4	Jan 28, 2026	0 / 2 (2)	2	⋮
Windows rev. 2	Jan 27, 2026	0 / 0 (0)	2	⋮
Política Linux Suricata rev. 2	Jan 27, 2026	0 / 1 (1)	2	⋮

At the bottom, there are buttons for 'Rows per page' (set to 20) and navigation arrows (< 1 >).

Dentro de la de Windows, añadimos una integración, la integración que debemos escoger es windows (a secas), y la añadimos.

The screenshot shows the 'Agent policies' interface for the 'Windows' policy. The top navigation bar includes links for 'View all agent policies', 'Windows' (the active tab), 'Integrations' (which is the active sub-tab), and 'Settings'. A search bar at the top right allows filtering data using KQL syntax. Below the navigation is a table of integrations. The columns are 'Integration policy', 'Integration', 'Namespace', 'Output', and 'Actions'. Two entries are listed:

Integration policy	Integration	Namespace	Output	Actions
system-2	System v2.12.0	default ⓘ	Default output ⓘ	⋮
windows-1	Windows v3.4.0	default ⓘ	Default output ⓘ	⋮

Ahora añadimos un agente, copiamos el comando del cuadro de texto seleccionando windowsx86_64 y lo instalamos usando powershell en la máquina windows (arrancar powershell como administrador).

Una vez hecho esto, nos vamos a elastic, creamos un data view, *windows* y deberíamos ver los logs del sistema windows, en el archivo json deberíamos encontrar nuestra IP de la máquina windows, la cual podemos comprobar haciendo “ipconfig” en la terminal windows.

(VER ANEXO 2: JSON LOG WINDOWS)

```

{
  "windows": [
    {
      "host.ip": [
        "fe80::4a8b:33c7:775a:21fc",
        "192.168.100.102"
      ],
      "agent.type": [
        "metricbeat"
      ],
      "event.module": [
        "windows"
      ],
      "agent.name.text": [
        "Windows10"
      ],
      "host.os.version": [
        "10.0"
      ],
      "host.os.kernel": [
        "10.0.19041.5247 (WinBuild.160101.0800)"
      ],
      "windows.perfmon.metrics.working_set": [
        5578752
      ],
      "host.os.name": [
        "Windows 10 Home"
      ],
      "agent.name": [
        "Windows10"
      ],
      "elastic_agent.snapshot": [
        ...
      ]
    }
  ]
}

```

```

C:\Users\sergio>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet:

Sufijo DNS específico para la conexión . . . : keepcoding.local
Vínculo: dirección IPv6 local . . . : fe80::4a8b:33c7:775a:21fc%5
Dirección IPv4 . . . . . : 192.168.100.102
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . : fe80::a00:27ff:fe6c:39aa%5
192.168.100.1

C:\Users\sergio>

```

Ahora vamos a la política del **honeypot**, añadimos integración y escogemos como integración **Custom Logs (Filestream)**, como integration name le ponemos “Cowrie”, en

namespace ponemos “cowrie” y en custom FileStream Logs cambiamos la ruta a **/home/kali/logs-cowrie.log**

The screenshot shows the Fleet interface for managing agent policies. The top navigation bar includes a passphrase entry field and a 'Finish update' button. Below the header, the page title is 'Agent policies / Honeypot'. The main content area shows the 'Honeypot' policy details, including its revision (4), integrations (2), agents (2 agents), and last update (Jan 28, 2026). A 'Actions' dropdown menu is visible. The 'Integrations' tab is selected, showing two entries: 'cowrie' (Custom Logs (Filestream)) and 'system-3' (System). The 'cowrie' entry has a 'Namespace' dropdown set to 'cowrie'.

Creamos en el home de kali un archivo que se llame logs-cowrie.log, despues ejecutamos el comando:

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

Ahora añadimos un data view, con index *cowrie* y lo aplicamos, veremos los logs del cowrie. Igual que antes debemos comprobar, ruta, ip... (**VER ANEXO 3: JSON LOG HONEYBOT**).

The screenshot shows the Elasticsearch Data View interface for the 'cowrie' index. The left sidebar lists 'Documents (148)', 'Patterns', and 'Field statistics'. The main table displays 148 documents, each with a timestamp and log file path. The first few rows show entries from 'Jan 31, 2026 @ 19:45:24.275' with paths like '/home/kali/logs-cowrie.log'. The right side of the table shows the raw log data, which includes IP addresses and port numbers. The bottom of the interface shows pagination controls and a 'Rows per page: 100' dropdown.

ANEXOS:

ANEXO 1: JSON LOG SURICATA.

```
{  
  "_index": ".ds-logs-suricata.eve-default-2026.01.27-000001",  
  "_id": "AZwTNsRYTMj50tE-V1pq",  
  "_version": 1,  
  "_ignored": [  
    "suricata.eve.direction",  
    "suricata.eve.files",  
    "suricata.eve.flow.dest_ip",  
    "suricata.eve.flow.dest_port",  
    "suricata.eve.flow.src_ip",  
    "suricata.eve.flow.src_port",  
    "suricata.eve.tc_progress",  
    "suricata.eve.ts_progress"  
  ],  
  "_source": {  
    "@timestamp": "2026-01-31T06:58:20.853Z",  
    "agent": {  
      "ephemeral_id": "b3806607-e209-4f89-90be-d481574f509d",  
      "id": "aed4f834-4d52-4ad8-a47d-70977dd475a3",  
      "name": "kali",  
      "type": "filebeat",  
      "version": "9.2.4"  
    },  
    "data_stream": {  
      "dataset": "suricata.eve",  
      "namespace": "default",  
      "type": "logs"  
    },  
    "destination": {  
      "address": "192.168.250.102",  
      "bytes": 46826,  
      "domain": "informatica.uv.es",  
      "ip": "192.168.250.102",  
      "packets": 33,  
      "port": 55880  
    },  
    "ecs": {  
      "version": "8.17.0"  
    },  
    "elastic_agent": {  
      "id": "aed4f834-4d52-4ad8-a47d-70977dd475a3",  
      "snapshot": false,  
      "version": "9.2.4"  
    },  
    "event": {
```

```
"agent_id_status": "verified",
"category": [
    "network",
    "intrusion_detection"
],
"created": "2026-01-31T08:40:44.657Z",
"dataset": "suricata.eve",
"ingested": "2026-01-31T08:40:47Z",
"kind": "alert",
"severity": 2,
"start": "2026-01-31T06:58:20.730Z",
"type": [
    "allowed"
],
},
"http": {
    "request": {
        "method": "GET"
    },
    "response": {
        "body": {
            "bytes": 42904
        },
        "status_code": 200
    }
},
"input": {
    "type": "log"
},
"log": {
    "file": {
        "path": "/var/log/suricata/eve.json"
    },
    "offset": 2487152
},
"message": "Descarga de archivo detectado",
"network": {
    "bytes": 49024,
    "community_id": "1:IW5xBZQjVg34BmKhUzFJ8qvh/yk=",
    "packets": 60,
    "protocol": "http",
    "transport": "tcp"
},
"observer": {
    "hostname": "kali",
    "ip": [
        "192.168.250.102",
        "fe80::48a0:b951:3cb0:ea30"
    ],
    "mac": [
```

```
"08-00-27-63-B0-05"
],
"product": "Suricata",
"type": "ids",
"vendor": "OISF"
},
"related": {
"hosts": [
"informatica.uv.es"
],
"ip": [
"147.156.84.161",
"192.168.250.102"
]
},
"rule": {
"category": "Descarga de archivo detectado",
"id": "3",
"name": "Archivo PDF Detectado"
},
"source": {
"address": "147.156.84.161",
"as": {
"number": 766,
"organization": {
"name": "Entidad Publica Empresarial Red.es"
}
},
"bytes": 2198,
"geo": {
"city_name": "Valencia",
"continent_name": "Europe",
"country_iso_code": "ES",
"country_name": "Spain",
"location": {
"lat": 39.46759999729693,
"lon": -0.3771000634878874
},
"region_iso_code": "ES-V",
"region_name": "Valencia"
},
"ip": "147.156.84.161",
"packets": 27,
"port": 80
},
"suricata": {
eve": {
>alert": {
"category": "Descarga de archivo detectado",
"gid": 1,
```

```
        "rev": 0,
        "signature": "Archivo PDF Detectado",
        "signature_id": 3
    },
    "direction": "to_client",
    "event_type": "alert",
    "files": [
        {
            "filename": "/iigua/IST/Tema2.pdf",
            "size": 42904,
            "stored": false,
            "state": "UNKNOWN",
            "tx_id": 0,
            "gaps": false,
            "storing": true,
            "sid": [
                3
            ]
        }
    ],
    "flow": {
        "dest_ip": "147.156.84.161",
        "dest_port": 80,
        "src_ip": "192.168.250.102",
        "src_port": 55880
    },
    "flow_id": "1168614008984888",
    "http": {
        "http_content_type": "application/pdf",
        "protocol": "HTTP/1.1"
    },
    "in_iface": "eth0",
    "ip_v": 4,
    "pkt_src": "wire/pcap",
    "tc_progress": "response_body",
    "ts_progress": "request_complete",
    "tx_id": 0
},
},
"tags": [
    "forwarded",
    "suricata-eve"
],
"url": {
    "domain": "informatica.uv.es",
    "original": "/iigua/IST/Tema2.pdf",
    "path": "/iigua/IST/Tema2.pdf"
},
"user_agent": {
    "device": {
```

```
        "name": "Other"
    },
    "name": "Firefox",
    "original": "Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0",
    "os": {
        "name": "Linux"
    },
    "version": "140.0"
}
},
"fields": {
    "rule.id": [
        "3"
    ],
    "elastic_agent.version": [
        "9.2.4"
    ],
    "event.category": [
        "network",
        "intrusion_detection"
    ],
    "suricata.eve.files.tx_id": [
        0
    ],
    "suricata.eve.tx_id": [
        0
    ],
    "user_agent.original.text": [
        "Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0"
    ],
    "suricata.eve.flow.dest_ip": [
        "147.156.84.161"
    ],
    "observer.vendor": [
        "OISF"
    ],
    "source.geo.region_name": [
        "Valencia"
    ],
    "suricata.eve.alert.signature": [
        "Archivo PDF Detectado"
    ],
    "suricata.eve.http.protocol": [
        "HTTP/1.1"
    ],
    "source.ip": [
        "147.156.84.161"
    ],
    "agent.name": [
        "kali"
    ]
}
```

```
 ],
  "destination.address": [
    "192.168.250.102"
  ],
  "suricata.eve.event_type": [
    "alert"
  ],
  "network.community_id": [
    "1:IW5xBZQjVg34BmKhUzFJ8qvh/yk="
  ],
  "event.agent_id_status": [
    "verified"
  ],
  "http.response.status_code": [
    200
  ],
  "suricata.eve.flow_id": [
    "1168614008984888"
  ],
  "source.geo.city_name": [
    "Valencia"
  ],
  "user_agent.original": [
    "Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0"
  ],
  "event.severity": [
    2
  ],
  "source.packets": [
    27
  ],
  "input.type": [
    "log"
  ],
  "suricata.eve.in_iface": [
    "eth0"
  ],
  "tags": [
    "forwarded",
    "suricata-eve"
  ],
  "url.path": [
    "/iigua/IST/Tema2.pdf"
  ],
  "agent.id": [
    "aed4f834-4d52-4ad8-a47d-70977dd475a3"
  ],
  "source.port": [
    80
  ],
}
```

```
"suricata.eve.direction": [
    "to_client"
],
"destination.bytes": [
    46826
],
"event.start": [
    "2026-01-31T06:58:20.730Z"
],
"source.as.number": [
    766
],
"destination.port": [
    55880
],
"suricata.eve.files.size": [
    42904
],
"user_agent.os.name": [
    "Linux"
],
"destination.packets": [
    33
],
"suricata.eve.alert.category": [
    "Descarga de archivo detectado"
],
"agent.type": [
    "filebeat"
],
"related.ip": [
    "147.156.84.161",
    "192.168.250.102"
],
"suricata.eve.files.storing": [
    true
],
"observer.product": [
    "Suricata"
],
"elastic_agent.snapshot": [
    false
],
"suricata.eve.pkt_src": [
    "wire/pcap"
],
"suricata.eve.flow.src_port": [
    55880
],
"elastic_agent.id": [

```

```
"aed4f834-4d52-4ad8-a47d-70977dd475a3"
],
"suricata.eve.ip_v": [
  4
],
"destination.ip": [
  "192.168.250.102"
],
"observer.hostname": [
  "kali"
],
"event.ingested": [
  "2026-01-31T08:40:47.000Z"
],
"@timestamp": [
  "2026-01-31T06:58:20.853Z"
],
"data_stream.dataset": [
  "suricata.eve"
],
"log.file.path": [
  "/var/log/suricata/eve.json"
],
"url.domain": [
  "informatica.uv.es"
],
"suricata.eve.files.state": [
  "UNKNOWN"
],
"agent.ephemeral_id": [
  "b3806607-e209-4f89-90be-d481574f509d"
],
"suricata.eve.http.http_content_type": [
  "application/pdf"
],
"user_agent.device.name": [
  "Other"
],
"suricata.eve.tc_progress": [
  "response_body"
],
"suricata.eve.alert.rev": [
  0
],
"url.original.text": [
  "/iigua/IST/Tema2.pdf"
],
"http.request.method": [
  "GET"
],
```

```
"suricata.eve.flow.src_ip": [
    "192.168.250.102"
],
"observer.mac": [
    "08-00-27-63-B0-05"
],
"user_agent.version": [
    "140.0"
],
"source.geo.region_iso_code": [
    "ES-V"
],
"suricata.eve.alert.gid": [
    1
],
"event.kind": [
    "alert"
],
"rule.name": [
    "Archivo PDF Detectado"
],
"network.packets": [
    60
],
"log.offset": [
    2487152
],
"user_agent.name": [
    "Firefox"
],
"destination.domain": [
    "informatica.uv.es"
],
"data_stream.type": [
    "logs"
],
"ecs.version": [
    "8.17.0"
],
"observer.type": [
    "ids"
],
"event.created": [
    "2026-01-31T08:40:44.657Z"
],
"agent.version": [
    "9.2.4"
],
"related.hosts": [
    "informatica.uv.es"
]
```

```
 ],
  "observer.ip": [
    "192.168.250.102",
    "fe80::48a0:b951:3cb0:ea30"
  ],
  "suricata.eve.flow.dest_port": [
    80
  ],
  "source.geo.location": [
    {
      "coordinates": [
        -0.3771000634878874,
        39.46759999729693
      ],
      "type": "Point"
    }
  ],
  "source.address": [
    "147.156.84.161"
  ],
  "user_agent.os.name.text": [
    "Linux"
  ],
  "suricata.eve.alert.signature_id": [
    3
  ],
  "event.module": [
    "suricata"
  ],
  "network.protocol": [
    "http"
  ],
  "suricata.eve.files.filename": [
    "/iigua/IST/Tema2.pdf"
  ],
  "source.geo.country_iso_code": [
    "ES"
  ],
  "network.bytes": [
    49024
  ],
  "source.bytes": [
    2198
  ],
  "suricata.eve.files.sid": [
    3
  ],
  "source.as.organization.name.text": [
    "Entidad Publica Empresarial Red.es"
  ],
  "suricata.eve.alert.level": [
    3
  ],
  "suricata.eve.alert.severity": [
    3
  ],
  "suricata.eve.alert.signature_name": [
    "HTTP/2 - Server Misconfiguration"
  ],
  "suricata.eve.alert.signature_desc": [
    "The server responded with a status code of 200 OK, which is considered a misconfiguration for an SSL/TLS endpoint. This is because the server is using an untrusted certificate or is not properly configured to handle SSL/TLS connections. It is recommended to review the server's configuration and ensure that it is using a valid certificate and is properly configured to handle SSL/TLS connections." ]
}
```

```
"data_stream.namespace": [
    "default"
],
"suricata.eve.files.stored": [
    false
],
"source.as.organization.name": [
    "Entidad Publica Empresarial Red.es"
],
"source.geo.continent_name": [
    "Europe"
],
"message": [
    "Descarga de archivo detectado"
],
"http.response.body.bytes": [
    42904
],
"network.transport": [
    "tcp"
],
"url.original": [
    "/iiguia/IST/Tema2.pdf"
],
"suricata.eve.files.gaps": [
    false
],
"suricata.eve.ts_progress": [
    "request_complete"
],
"event.type": [
    "allowed"
],
"source.geo.country_name": [
    "Spain"
],
"rule.category": [
    "Descarga de archivo detectado"
],
"event.dataset": [
    "suricata.eve"
]
}
}
```

ANEXO 2: JSON LOG WINDOWS.

{

```
"_index": ".ds-metrics-windows.perfmon-default-2026.01.31-000001",
"_id": "99FQFZwB3VXsMkSyRmWd",
"_version": 1,
"_source": {
  "agent": {
    "name": "Windows10",
    "id": "62122fa5-836e-49a8-b75a-d4a5b8352b37",
    "ephemeral_id": "3367e998-67f9-46c0-bf02-9832b8cb7a66",
    "type": "metricbeat",
    "version": "9.2.4"
  },
  "@timestamp": "2026-01-31T18:28:26.947Z",
  "ecs": {
    "version": "8.0.0"
  },
  "service": {
    "type": "windows"
  },
  "data_stream": {
    "namespace": "default",
    "type": "metrics",
    "dataset": "windows.perfmon"
  },
  "elastic_agent": {
    "id": "62122fa5-836e-49a8-b75a-d4a5b8352b37",
    "version": "9.2.4",
    "snapshot": false
  },
  "host": {
    "hostname": "Windows10",
    "os": {
      "build": "19045.5247",
      "kernel": "10.0.19041.5247 (WinBuild.160101.0800)",
      "name": "Windows 10 Home",
      "family": "windows",
      "type": "windows",
      "version": "10.0",
      "platform": "windows"
    },
    "ip": [
      "fe80::4a8b:33c7:775a:21fc",
      "192.168.100.102"
    ],
    "name": "windows10",
    "id": "b03073f0-8c78-41e8-81ed-eb67037c01c4",
    "mac": [
      "08-00-27-AB-84-C2"
    ],
    "architecture": "x86_64"
  }
}
```

```
"metricset": {
    "period": 10000,
    "name": "perfmon"
},
"event": {
    "duration": 1010218400,
    "agent_id_status": "verified",
    "ingested": "2026-01-31T18:28:28Z",
    "module": "windows",
    "dataset": "windows.perfmon"
},
"windows": {
    "perfmon": {
        "instance": "csrss",
        "metrics": {
            "working_set": 5578752
        },
        "object": "Process"
    }
},
"fields": {
    "elastic_agent.version": [
        "9.2.4"
    ],
    "host.os.name.text": [
        "Windows 10 Home"
    ],
    "host.hostname": [
        "Windows10"
    ],
    "host.mac": [
        "08-00-27-AB-84-C2"
    ],
    "host.os.build": [
        "19045.5247"
    ],
    "service.type": [
        "windows"
    ],
    "host.ip": [
        "fe80::4a8b:33c7:775a:21fc",
        "192.168.100.102"
    ],
    "agent.type": [
        "metricbeat"
    ],
    "event.module": [
        "windows"
    ],
}
```

```
"agent.name.text": [
  "Windows10"
],
"host.os.version": [
  "10.0"
],
"host.os.kernel": [
  "10.0.19041.5247 (WinBuild.160101.0800)"
],
"windows.perfmon.metrics.working_set": [
  5578752
],
"host.os.name": [
  "Windows 10 Home"
],
"agent.name": [
  "Windows10"
],
"elastic_agent.snapshot": [
  false
],
"host.name": [
  "windows10"
],
"event.agent_id_status": [
  "verified"
],
"host.id": [
  "b03073f0-8c78-41e8-81ed-eb67037c01c4"
],
"metricset.name.text": [
  "perfmon"
],
"host.os.type": [
  "windows"
],
"elastic_agent.id": [
  "62122fa5-836e-49a8-b75a-d4a5b8352b37"
],
"windows.perfmon.object": [
  "Process"
],
"data_stream.namespace": [
  "default"
],
"windows.perfmon.instance": [
  "csrss"
],
"metricset.period": [
  10000
]
```

```
 ],
  "data_stream.type": [
    "metrics"
  ],
  "event.duration": [
    1010218400
  ],
  "host.architecture": [
    "x86_64"
  ],
  "metricset.name": [
    "perfmon"
  ],
  "event.ingested": [
    "2026-01-31T18:28:28.000Z"
  ],
  "@timestamp": [
    "2026-01-31T18:28:26.947Z"
  ],
  "agent.id": [
    "62122fa5-836e-49a8-b75a-d4a5b8352b37"
  ],
  "ecs.version": [
    "8.0.0"
  ],
  "host.os.platform": [
    "windows"
  ],
  "data_stream.dataset": [
    "windows.perfmon"
  ],
  "agent.ephemeral_id": [
    "3367e998-67f9-46c0-bf02-9832b8cb7a66"
  ],
  "agent.version": [
    "9.2.4"
  ],
  "host.os.family": [
    "windows"
  ],
  "event.dataset": [
    "windows.perfmon"
  ]
}
}
```

ANEXO 3: JSON LOG HONEYBOT

```
{  
  "_index": ".ds-logs-filestream.generic-cowrie-2026.01.28-000001",  
  "_id": "AZwVlsRYTMj50tHcmiPW",  
  "_version": 1,  
  "_source": {  
    "@timestamp": "2026-01-31T19:45:24.275Z",  
    "agent": {  
      "ephemeral_id": "a1fd2140-bdee-4163-850a-ea55defb649a",  
      "id": "981c7126-3b63-442e-bd41-cf2e37519202",  
      "name": "kali",  
      "type": "filebeat",  
      "version": "9.2.4"  
    },  
    "data_stream": {  
      "dataset": "filestream.generic",  
      "namespace": "cowrie",  
      "type": "logs"  
    },  
    "ecs": {  
      "version": "8.0.0"  
    },  
    "elastic_agent": {  
      "id": "981c7126-3b63-442e-bd41-cf2e37519202",  
      "snapshot": false,  
      "version": "9.2.4"  
    },  
    "event": {  
      "agent_id_status": "verified",  
      "dataset": "filestream.generic",  
      "ingested": "2026-01-31T19:45:34Z"  
    },  
    "host": {  
      "architecture": "x86_64",  
      "containerized": false,  
      "hostname": "kali",  
      "id": "686fa41363f049e9a84be345a7078e04",  
      "ip": [  
        "192.168.200.99",  
        "fe80::e604:bae0:d67e:2528",  
        "172.17.0.1",  
        "fe80::42:99ff:feb7:442",  
        "fe80::3c1b:2ff:feab:90cd"  
      ],  
      "mac": [  
        "02-42-99-B7-04-42",  
        "08-00-27-1F-B7-23",  
        "3E-1B-02-AB-90-CD"  
      ]  
    }  
  }  
}
```

```
        ],
      "name": "kali",
      "os": {
        "codename": "kali-rolling",
        "family": "debian",
        "kernel": "6.16.8+kali-amd64",
        "name": "Kali GNU/Linux",
        "platform": "kali",
        "type": "linux",
        "version": "2025.4"
      }
    },
    "input": {
      "type": "filestream"
    },
    "log": {
      "file": {
        "device_id": "2049",
        "fingerprint": "acef742d2c1ae69f4d3f0cf12cb64c9fb6b644e8ca612d21e30f8de98439ece6",
        "inode": "1839101",
        "path": "/home/kali/logs-cowrie.log"
      },
      "offset": "4000"
    },
    "message": "2026-01-31T19:45:24+0000 [HoneyPotSSHTransport,0,192.168.1.40] Command found: ls"
  },
  "fields": {
    "elastic_agent.version": [
      "9.2.4"
    ],
    "host.os.name.text": [
      "Kali GNU/Linux"
    ],
    "host.hostname": [
      "kali"
    ],
    "host.mac": [
      "02-42-99-B7-04-42",
      "08-00-27-1F-B7-23",
      "3E-1B-02-AB-90-CD"
    ],
    "host.ip": [
      "192.168.200.99",
      "fe80::e604:bae0:d67e:2528",
      "172.17.0.1",
      "fe80::42:99ff:feb7:442",
      "fe80::3c1b:2ff:feab:90cd"
    ],
    "agent.type": [

```

```
    "filebeat":  
    ],  
    "event.module": [  
        "filestream"  
    ],  
    "host.os.version": [  
        "2025.4"  
    ],  
    "host.os.kernel": [  
        "6.16.8+kali-amd64"  
    ],  
    "log.file.device_id": [  
        "2049"  
    ],  
    "host.os.name": [  
        "Kali GNU/Linux"  
    ],  
    "agent.name": [  
        "kali"  
    ],  
    "elastic_agent.snapshot": [  
        false  
    ],  
    "host.name": [  
        "kali"  
    ],  
    "event.agent_id_status": [  
        "verified"  
    ],  
    "host.id": [  
        "686fa41363f049e9a84be345a7078e04"  
    ],  
    "host.os.type": [  
        "linux"  
    ],  
    "elastic_agent.id": [  
        "981c7126-3b63-442e-bd41-cf2e37519202"  
    ],  
    "data_stream.namespace": [  
        "cowrie"  
    ],  
    "host.os.codename": [  
        "kali-rolling"  
    ],  
    "input.type": [  
        "filestream"  
    ],  
    "log.offset": [  
        "4000"  
    ],
```

```
"message": [
  "2026-01-31T19:45:24+0000 [HoneyPotSSHTTransport,0,192.168.1.40] Command found: ls "
],
"data_stream.type": [
  "logs"
],
"host.architecture": [
  "x86_64"
],
"event.ingested": [
  "2026-01-31T19:45:34.000Z"
],
"@timestamp": [
  "2026-01-31T19:45:24.275Z"
],
"agent.id": [
  "981c7126-3b63-442e-bd41-cf2e37519202"
],
"ecs.version": [
  "8.0.0"
],
"host.containerized": [
  false
],
"host.os.platform": [
  "kali"
],
"log.file.inode": [
  "1839101"
],
"data_stream.dataset": [
  "filestream.generic"
],
"log.file.path": [
  "/home/kali/logs-cowrie.log"
],
"agent.ephemeral_id": [
  "a1fd2140-bdee-4163-850a-ea55defb649a"
],
"agent.version": [
  "9.2.4"
],
"log.file.fingerprint": [
  "acef742d2c1ae69f4d3f0cf12cb64c9fb6b644e8ca612d21e30f8de98439ece6"
],
"host.os.family": [
  "debian"
],
"event.dataset": [
  "filestream.generic"
```

]
}
}