

# III Bootcamp Full Stack Cibersecurity

# Módulo 5 - Blue Team



## Caso práctico

**Marcos Alonso González**  
[alonsogonzalezmarcos@gmail.com](mailto:alonsogonzalezmarcos@gmail.com)  
<https://github.com/magalorn>

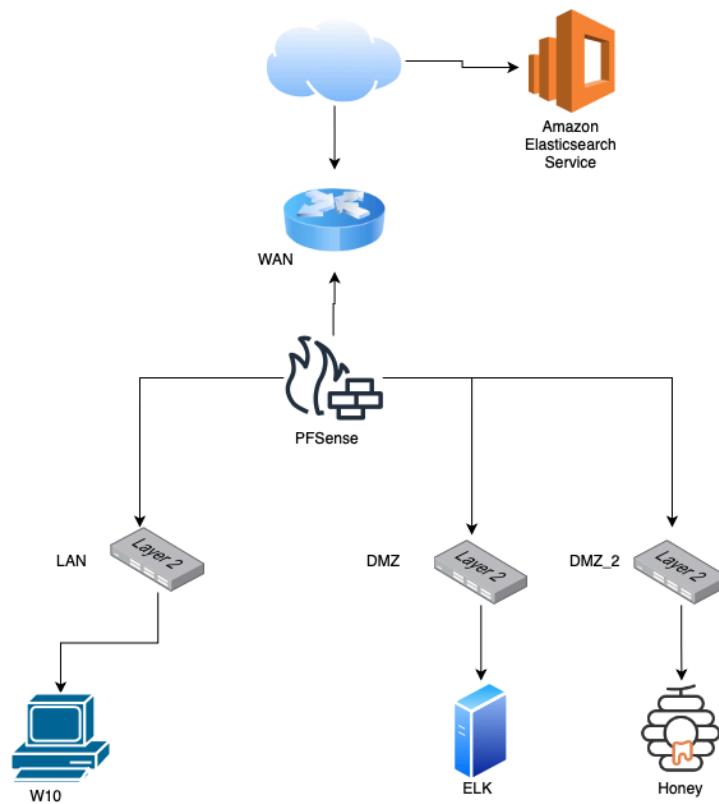
18 de marzo de 2022

# Índice

<b>Enunciado del caso práctico</b>	<b>3</b>
<b>1. Proceso de creación de la infraestructura</b>	<b>4</b>
1.1. Creación de UTM y configuración de interfaces	4
1.2. Creación de NAT	10
1.3. Creación de OpenVPN	12
1.4. Reglas DMZ	18
<b>2. Instalación de herramientas</b>	<b>20</b>
2.1. Suricata	20
2.2. Elastic Stack	22
<b>3. Conexión entre equipos de la red interna</b>	<b>23</b>
3.1. Integración de Suricata en Elastic Stack	23
3.2. Creación e integración de honeypot en Elastic Stack	28

## Enunciado del caso práctico

Queremos montar la siguiente infraestructura:



La cual debe cumplir los siguientes requisitos:

- Creación de un PfSense en bridge que conecte 3 redes, LAN, DMZ y DMZ2 estas como red interna.
- Un equipo en LAN (nombre IT) , un stack ELK en otro equipo DMZ y un honeypot en un tercer equipo que denominamos DMZ2.
- El IT debe correr Suricata y poder conectarse al ELK vía Kibana para mostrar los logs.
- El equipo DMZ2 debe alojar un honeypot sin acceso a las otras redes (solo para transmitir logs) pero si debe ser accesible desde la red WAN.
- El servidor ELK debe almacenar y mostrar los logs del honeypot.

**\*Nota:** Todos los equipos utilizados para la creación de la red interna (IT, DMZ y DMZ2) son distribuciones Kali Linux con SO Debian 64bit.

# 1. Proceso de creación de la infraestructura

## 1.1. Creación de UTM y configuración de interfaces

Utilizamos una imagen ISO de **pfSense**, una distribución personalizada de FreeBSD adaptado para su uso como firewall y enrutador.

La importamos a VMWare con la siguiente configuración:

1. Se crean 2 tarjetas o adaptadores de red más, además de la existente por defecto



2. Se conecta la primera en modo Bridge a Wi-Fi y las otras dos a las redes personalizadas creadas (vmnet2 y vmnet3)

3. Se importa la imagen de disco de pfSense y se procede a la instalación siguiendo los pasos preconfigurados que marca el instalador.
4. Una vez instalado y reiniciado, se asignan las interfaces correspondientes a las. con la opción 1) Assign interfaces
5. Se asigna la WAN a la em0, es la máquina por donde entraremos a internet
6. Se asigna la LAN a la interfaz em1, que hará de LAN o IT y será encargada del Firewalling y el modo NAT.
7. Se asigna la DMZ a la tercera red creada, la em2, como red opcional.

```

Starting package OpenVPN Client Export Utility...done.
pfSense 2.6.0-RELEASE amd64 Mon Jan 31 19:57:53 UTC 2022
Bootup complete

FreeBSD/amd64 (Myserver.marcos.local) (ttyv0)

VMware Virtual Machine - Netgate Device ID: 0c2665cc6cf1372f803b

*** Welcome to pfSense 2.6.0-RELEASE (amd64) on Myserver ***

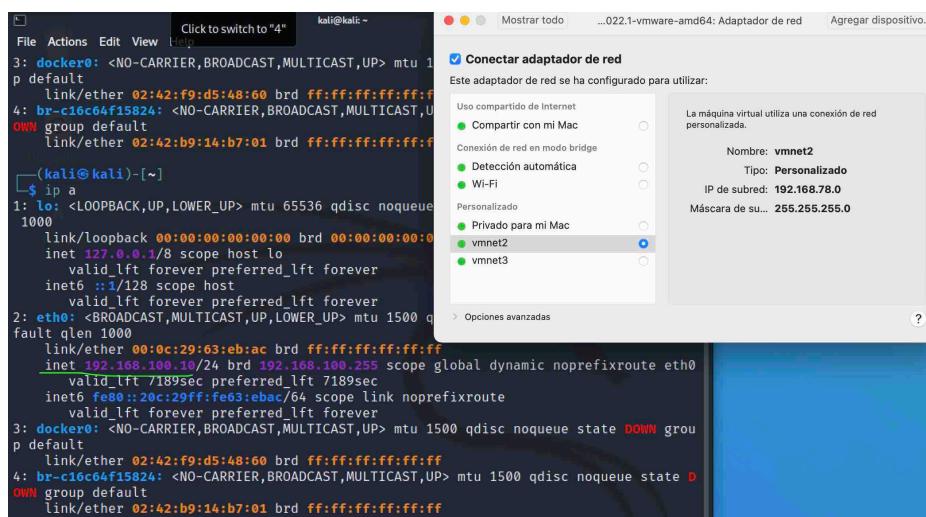
WAN (wan)      -> em0          -> v4/DHCP4: 192.168.1.67/24
IT (lan)        -> em1          -> v4: 192.168.100.1/24
DMZ (opt1)      -> em2          -> v4: 192.168.200.1/24

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: ■

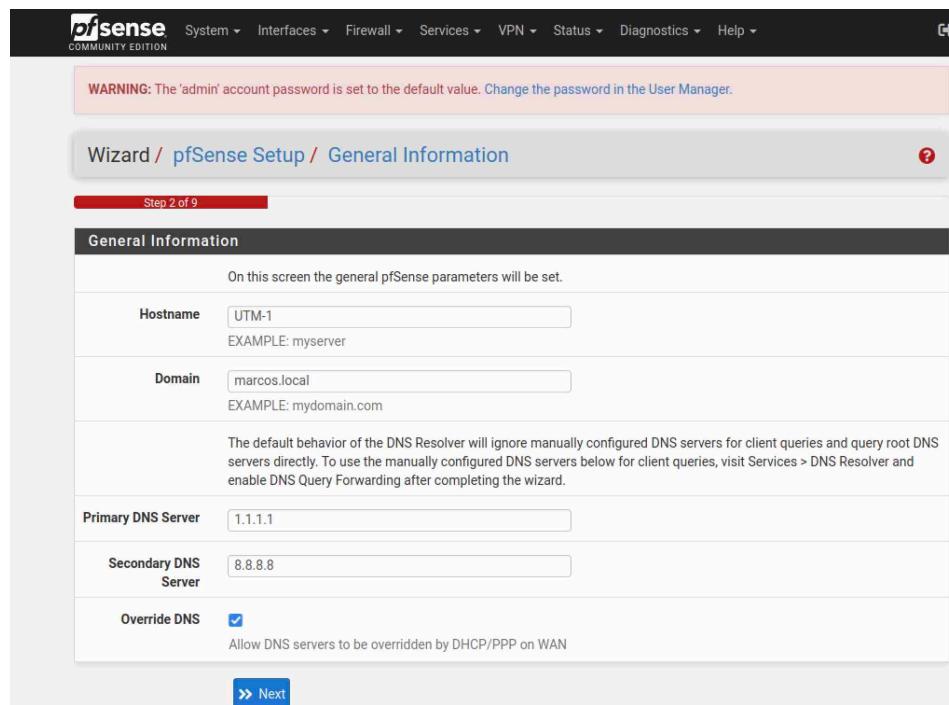
```

8. En la máquina Kali en que nos conectaremos a la interfaz de PfSense, seleccionamos el adaptador de red vmnet2, y la inet que se asigna a Kali será la 192.168.100.1

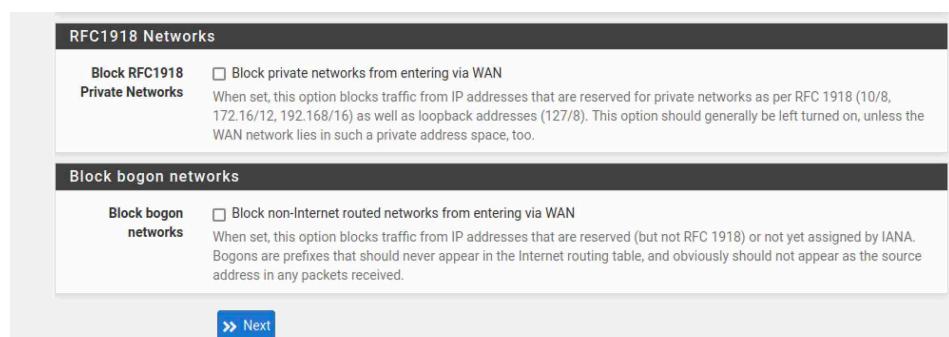


9. Vamos con el navegador a 192.168.1.1 y nos logueamos

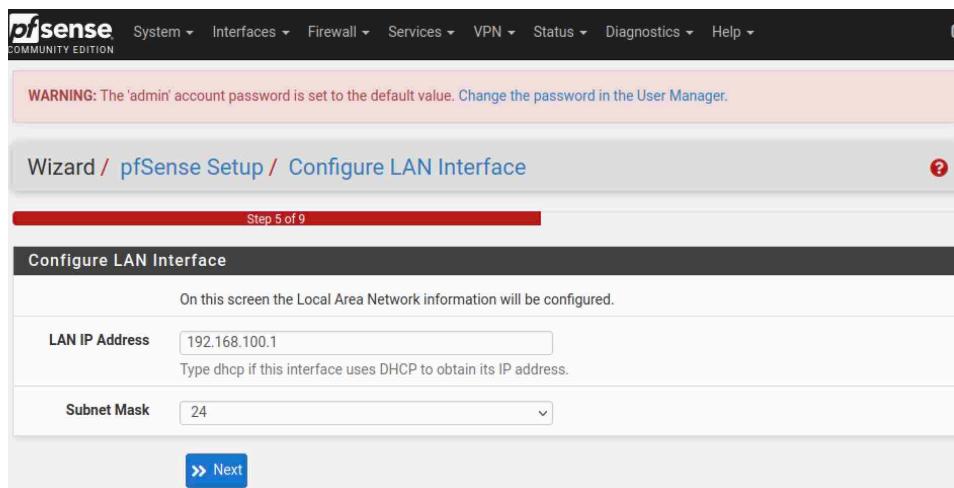
10. En la pantalla inicial de configuración se introducen estos valores para definir el hostname y el rango de dominio de nuestra red y los DNS de CloudFlare en este caso.



11. En la siguiente página se configura la interface WAN, se deja por defecto en DHCP. En esta misma página se desbloquean las redes RFC1918 y Bogon. Es una regla que declara cuales son las IPs publicas y las privadas, si esta activado no permite a las redes privadas salir afuera. Si pfSense como router o firewall está conectado directamente a internet hay que dejarlo seleccionado, como no está conectado a internet en este caso, se deja desbloqueado.



12. Seguidamente se configura la interfaz LAN, con la IP 192.168.100.1 y la subnet mask /24.

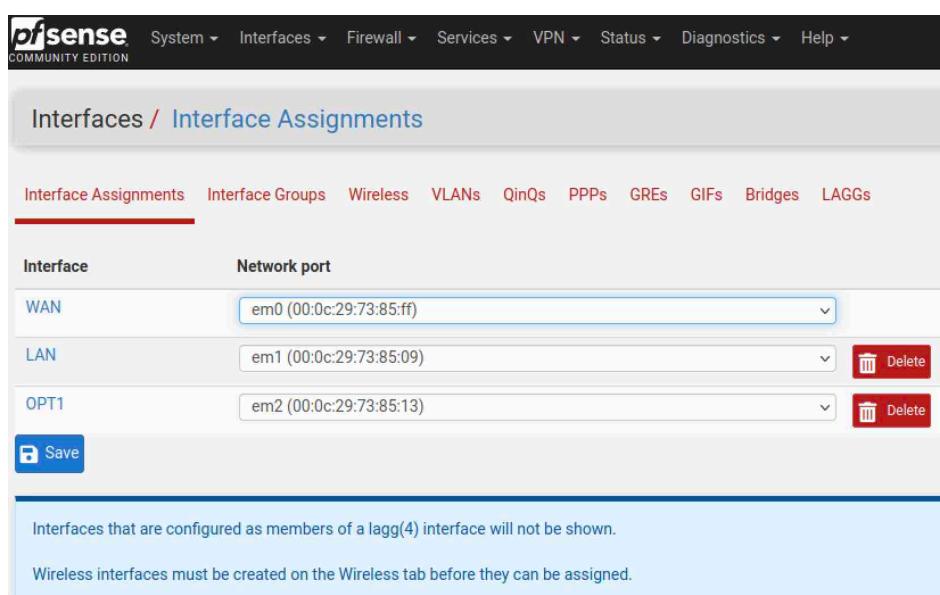


13. Finalmente se cambia la password por defecto y se recarga la herramienta.

14. Se desconecta la Kali de la red y se vuelve a conectar para que haga efecto la configuración de las interfaces.

15. Navegamos a la IP de la LAN (192.168.100.1) y llegamos al Dashboard de la herramienta.

16. Por último, vamos a configurar la interface em2. En Interfaces/Interface Assignments, se selecciona y se añade la interface em2.



17. Despues, seleccionamos la **interface LAN** (em1) y le cambiamos el nombre a **IT**.

18. Lo mismo con **OPT1** (em2), se habilita, cambiamos el nombre a **DMZ** y le asignamos una IP estatica. Despues en Services/DHCP server, se habilita el DHCP para esta **interface DMZ** y se le asigna un rango de IPs.

Seguidamente se configura el DNS servers a la misma IP que hemos puesto al propio servidor pfSense (192.168.200.1), y los DNS servers establecidos en el propio UTM porque el UTM-1 va a ser tambien un servidor DNS.

**Interfaces / OPT1 (em2)**

**General Configuration**

- Enable:**  Enable interface
- Description:** DMZ  
Enter a description (name) for the interface here.
- IPv4 Configuration Type:** Static IPv4
- IPv6 Configuration Type:** None
- MAC Address:** XX:XX:XX:XX:XX:XX  
This field can be used to modify ('spoof') the MAC address of this interface. Enter a MAC address in the following format: xx:xx:xx:xx:xx or leave blank.
- MTU:** (Input field)  
If this field is blank, the adapter's default MTU will be used. This is typically 1500 bytes but can vary in some circumstances.
- MSS:** (Input field)  
If a value is entered in this field, then MSS clamping for TCP connections to the value entered above minus 40 for IPv4 (TCP/IPv4 header size) and minus 60 for IPv6 (TCP/IPv6 header size) will be in effect.
- Speed and Duplex:** Default (no preference, typically autoselect)  
Explicitly set speed and duplex mode for this interface.  
WARNING: MUST be set to autoselect (automatically negotiate speed) unless the port this interface connects to has its speed and duplex forced.

**Static IPv4 Configuration**

- IPv4 Address:** 192.168.200.1 / 24

**Services / DHCP Server / DMZ**

**General Options**

- 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 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 below (i.e. for this interface) will get an IP address within this scope/range.
- Ignore denied clients:**  Denied clients will be ignored rather than rejected.  
This option is not compatible with failover and cannot be enabled when a Failover Peer IP address is configured.
- Ignore client identifiers:**  If a client includes a unique identifier in its DHCP request, that UID will not be recorded in its lease.  
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.
- Subnet:** 192.168.200.0
- Subnet mask:** 255.255.255.0
- Available range:** 192.168.200.1 - 192.168.200.254
- Range:** From 192.168.200.100 To 192.168.200.200

En Other options/Gateway establecemos la misma IP ya que también vamos a querer navegar por internet desde esta IP.

**Servers**

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	Leave blank to use the system default DNS servers: this interface's IP if DNS Forwarder or Resolver is enabled, otherwise the servers configured on the System / General Setup page.

**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.	
<input type="checkbox"/> 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 Options**

Gateway	192.168.200.1
The default is to use the IP on this interface of the firewall as the gateway. Specify an alternate gateway here if this is not the correct gateway for the network. Type "none" for no gateway assignment.	

19. Comprobamos en el Dashboard que aparezcan las 3 interfaces con sus IPs correspondientes.

Interfaces			
	WAN	1000baseT <full-duplex>	192.168.1.67
	IT	1000baseT <full-duplex>	192.168.100.1
	DMZ	1000baseT <full-duplex>	192.168.200.1

## 1.2. Creación de NAT

NAT permite acceder a un servicio que está dentro de una red interna, se va a crear a través de Port forwarding.

Los pasos a seguir serían:

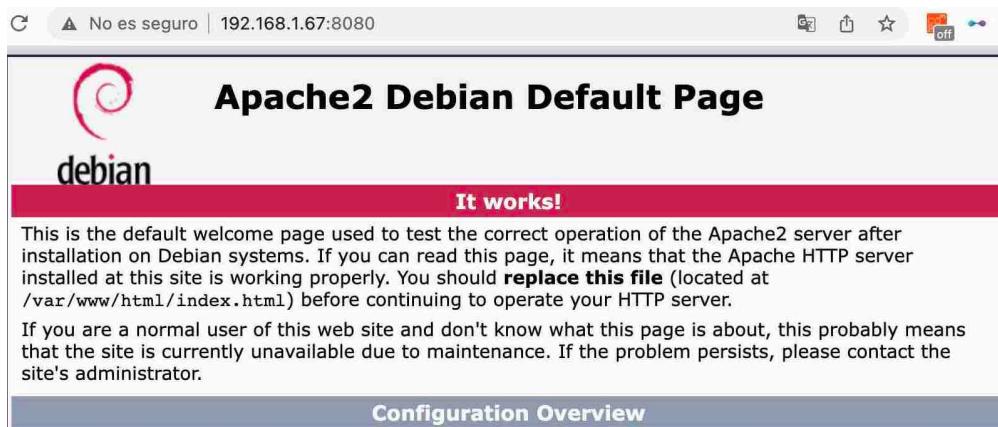
1. Levantar en nuestro Kali un servidor web con el comando

```
sudo service apache2 start
```

2. En pfSense, vamos a Firewall/NAT/Port Forward/Edit añadimos reglas, con destino WAN address puerto 8080, redirección a la IP de la Kali puerto 80. Esto significa que la IP que va a recibir la petición de nuestro localhost es la de la WAN (UTM-1) y cuando llegue esa petición, el firewall la va a reenviar al Kali (IT) por un puerto concreto, en el que está corriendo el servidor web apache2.

Actions	Description	NAT Ports	NAT IP	Dest. Ports	Dest. Address	Source Ports	Protocol	Interface	Selected
		80	192.168.100.10	8080	WAN address	*	TCP	WAN	<input checked="" type="checkbox"/>

3. Se comprueba que desde cualquier navegador de nuestro localhost tenemos acceso a través de la IP:8080 WAN al servidor apache2 que corre en la IP:80 IT.



4. En Firewall/Rules/IT se pueden comprobar todas las reglas que pfSense ha creado automáticamente a partir de la configuración realizada.

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/> 1/4.80 MiB	*	*	*	IT Address	443	*	*		Anti-Lockout Rule	
<input checked="" type="checkbox"/> 0/111 KiB	IPv4	*	IT net	*	*	*	*	none	Default allow LAN to any rule	
<input type="checkbox"/> 0/0 B	IPv6	*	IT net	*	*	*	*	none	Default allow LAN IPv6 to any rule	

## 1.3. Creación de OpenVPN

Vamos a crear una VPN para que las conexiones entrantes y salientes del WAN a la red interna y viceversa pasen por el UTM y estén controladas.

1. En pfSense, entramos en System/Package Manager/Available Packages
2. Buscamos e instalamos el paquete openvpn-client-export
3. Seguidamente vamos a System/Certificate Manager/CAs. Vamos a crear una entidad certificadora dentro de pfSense.
4. Añadimos una nueva autoridad certificadora con esta configuración

The screenshot shows the pfSense Certificate Manager interface under the 'CAs' tab. It displays a table of certificate authorities. There is one entry for 'CA\_marcos' which is self-signed. The table columns include Name, Internal, Issuer, Certificates, Distinguished Name, In Use, and Actions. The 'Actions' column for CA\_marcos contains icons for edit, delete, and other management functions. Below the table, there is a note about the certificate's validity period from March 9, 2022, to March 9, 2023.

5. Vamos a System/Certificate Manager/Certificates y añadimos un nuevo certificado con la opción Server Certificate y esta configuración

The screenshot shows the pfSense Certificate Manager interface under the 'Certificates' tab. It displays a table of certificates. There are two entries: 'webConfigurator default' (self-signed) and 'utm'. Both certificates are issued by 'CA: No' and have 'Server: Yes'. The 'Actions' column for each certificate includes icons for edit, delete, and other management functions. Below the table, there is a note about the certificate's validity period for both entries.

6. Seguidamente vamos a VPN/Open VPN/Servers para crear un servidor que va a ir a IT por la VPN. Seleccionamos Remote Access, doble factor de autenticación, por certificado y usuario y el server certificate UTM que hemos creado antes.

**General Information**

- Description:** Servidor VPN IT
- Disabled:**  Disable this server

**Mode Configuration**

- Server mode:** Remote Access ( SSL/TLS + User Auth )
- Backend for authentication:** Local Database
- Device mode:** tun - Layer 3 Tunnel Mode

**Endpoint Configuration**

- Protocol:** TCP on IPv4 only
- Interface:** WAN
- Local port:** 9194

**Cryptographic Settings**

- TLS Configuration**
  - Use a TLS Key
  - Automatically generate a TLS Key.
- Peer Certificate Authority:** CA\_marcos
- Peer Certificate Revocation list:** No Certificate Revocation Lists defined. One may be created here: System > Cert. Manager
- OCSP Check:**  Check client certificates with OCSP
- Server certificate:** utm (Server: Yes, CA: CA\_marcos)
- DH Parameter Length:** 2048 bit
- ECDH Curve:** Use Default
- Data Encryption Negotiation:**  Enable Data Encryption Negotiation

<b>Fallback Data Encryption Algorithm</b>	AES-256-CBC (256 bit key, 128 bit block)	The Fallback Data Encryption Algorithm used for data channel packets when communicating with clients that do not support data encryption algorithm negotiation (e.g. Shared Key). This algorithm is automatically included in the Data Encryption Algorithms list.
<b>Auth digest algorithm</b>	SHA256 (256-bit)	The algorithm used to authenticate data channel packets, and control channel packets if a TLS Key is present. When an AEAD Encryption Algorithm mode is used, such as AES-GCM, this digest is used for the control channel only, not the data channel. The server and all clients must have the same setting. While SHA1 is the default for OpenVPN, this algorithm is insecure.
<b>Hardware Crypto</b>	Intel RDRAND engine - RAND	
<b>Certificate Depth</b>	One (Client+Server)	When a certificate-based client logs in, do not accept certificates below this depth. Useful for denying certificates made with intermediate CAs generated from the same CA as the server.
<b>Strict User-CN Matching</b>	<input type="checkbox"/> Enforce match	When authenticating users, enforce a match between the common name of the client certificate and the username given at login.
<b>Client Certificate Key Usage Validation</b>	<input checked="" type="checkbox"/> Enforce key usage	Verify that only hosts with a client certificate can connect (EKU: "TLS Web Client Authentication").

7. Importante también es la configuración de Tunnel Settings. Si se habilita aquí la opción Redirect IPv4/IPv6 Gateway, todo el tráfico que salga de localhost va a pasar a través del UTM, y si localhost sale a internet, lo hará con la IP publica de UTM. En este caso, se deshabilita y solo se incluye la opción IPv4 Local Networks, para que solo el tráfico que se dirige de localhost a IT vaya por VPN. El resto de las opciones las dejamos con están por defecto.

Tunnel Settings		
<b>IPv4 Tunnel Network</b>	192.168.240.0/24	This is the IPv4 virtual network or network type alias with a single entry used for private communications between this server and client hosts expressed using CIDR notation (e.g. 10.0.8.0/24). The first usable address in the network will be assigned to the server virtual interface. The remaining usable addresses will be assigned to connecting clients.
<b>IPv6 Tunnel Network</b>		This is the IPv6 virtual network or network type alias with a single entry used for private communications between this server and client hosts expressed using CIDR notation (e.g. fe80::/64). The ::1 address in the network will be assigned to the server virtual interface. The remaining addresses will be assigned to connecting clients.
<b>Redirect IPv4 Gateway</b>	<input type="checkbox"/> Force all client-generated IPv4 traffic through the tunnel.	
<b>Redirect IPv6 Gateway</b>	<input type="checkbox"/> Force all client-generated IPv6 traffic through the tunnel.	
<b>IPv4 Local network(s)</b>	192.168.100.0/24	IPv4 networks that will be accessible from the remote endpoint. Expressed as a comma-separated list of one or more CIDR ranges or host/network type aliases. This may be left blank if not adding a route to the local network through this tunnel on the remote machine. This is generally set to the LAN network.
<b>IPv6 Local network(s)</b>		IPv6 networks that will be accessible from the remote endpoint. Expressed as a comma-separated list of one or more IP/PREFIX or host/network type aliases. This may be left blank if not adding a route to the local network through this tunnel on the remote machine. This is generally set to the LAN network.
<b>Concurrent connections</b>		Specify the maximum number of clients allowed to concurrently connect to this server.
<b>Allow Compression</b>	Refuse any non-stub compression (Most secure)	Allow compression to be used with this VPN instance. Compression can potentially increase throughput but may allow an attacker to extract secrets if they can control compressed plaintext traversing the VPN (e.g. HTTP). Before enabling compression, consult information about the VORACLE, CRIME, TIME, and BREACH attacks against TLS to decide if the use case for this specific VPN is vulnerable to attack.
Asymmetric compression allows an easier transition when connecting with older peers.		

8. Seguidamente vamos a crear un usuario que se pueda conectar al tunnel network. Vamos a System/User Manager/Users y añadimos nuevo usuario donde lo importante es crear un certificado de usuario con el CA creado anteriormente. Comprobamos después en System/Certificate Manager/Certificates que este certificado de usuario se ha creado correctamente.

The screenshot shows the 'User Manager' configuration page for a new user named 'marcos'. The 'Group membership' section has 'admins' selected under 'Not member of'. Below the table, there are buttons for moving items between 'Member of' and 'Not member of' lists, and a note about using the Ctrl key to select multiple items.

Effective Privileges			
Inherited from	Name	Description	Action
			<a href="#">+ Add</a>

User Certificates		
Name	CA	
marcos_vpn	CA_marcos	<a href="#">Delete</a>

9. Despu s vamos a OpenVPN/Client export. Descargamos el archivo Inline configurations/Most clients y lo llevamos a nuestro localhost.

The screenshot shows the 'OpenVPN Clients' configuration page for user 'marcos'. The 'Export' section includes a list of download options: 'Inline Configurations' (with 'Most Clients' highlighted with a green circle), 'Bundled Configurations', 'Current Windows Installers', 'Legacy Windows Installers', and 'Viscosity' options. Each option has a corresponding download button.

10. Antes de conectar nuestro localhost a la VPN, hay que crear una nueva regla en Firewall/Rules/WAN. Añadimos nueva regla, dejando todas las opciones por defecto excepto Destination y Extra options, que las configuramos así

**Destination**

**Destination**: (other) 9194 (other) 9194  
**Range**: From Custom To Custom

Specify the destination port or port range for this rule. The "To" field may be left empty if only filtering a single port.

**Extra Options**

**Log**:  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**: VPN\_IT  
 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.

**Advanced Options**: [Display Advanced](#)

11. Y otra nueva regla en Firewall/Rules/OpenVPN. En este caso, la única opción a configurar es Edit Firewall/Rule/Protocol a Any.

**Edit Firewall Rule**

**Action**: Pass  
 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.

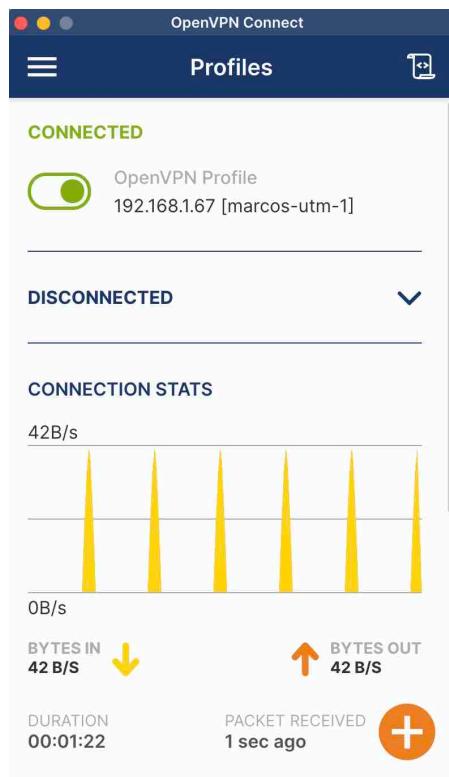
**Disabled**:  Disable this rule  
 Set this option to disable this rule without removing it from the list.

**Interface**: OpenVPN  
 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.

12. Descargamos la aplicación OpenVPN Connect en nuestro localhost y arrastramos el fichero **.ovpn** para agregar al usuario que hemos creado en pfSense. Si todo está bien configurado, al conectar la VPN a este usuario, obtenemos este resultado



13. Comprobamos después que desde el navegador de localhost podemos alcanzar el servidor apache2 activo en la IP del IT (nuestra Kali en este caso)

This is a screenshot of a web browser displaying the Apache2 Debian Default Page. The page title is "Apache2 Debian Default Page" and features the Debian logo. The main content area has a red banner with the text "It works!". Below the banner, it says: "This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should [replace this file](#) (located at `/var/www/html/index.html`) before continuing to operate your HTTP server." It also states: "If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator." A "Configuration Overview" section follows, explaining that the configuration is split into several files and points to the `README.Debian.gz` file for full documentation. It also describes the layout of the configuration files in the `/etc/apache2/` directory. At the bottom, a list provides details about the configuration files:

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the

## 1.4. Reglas DMZ

Vamos a crear reglas para que la DMZ pueda conectarse a la WAN.

1. Primero creamos un nuevo NAT en Firewall/NAT/Port Forward con la siguiente configuración. Desde la interfaz WAN con destino a la IP de WAN y tráfico HTTPS, va a reenviar el tráfico al puerto 80. La IP a la que va a redirigir es la 192.168.200.103, que es la que habíamos creado para DMZ.

The screenshot shows the 'Edit Redirect Entry' configuration page. Key settings include:

- Disabled:** Unchecked.
- No RDR (NOT):** Checked.
- Interface:** WAN.
- Address Family:** IPv4.
- Protocol:** TCP.
- Source:** Display Advanced.
- Destination:** WAN address (Type: Address/mask).
- Destination port range:** From port 8080, To port 8080.
- Redirect target IP:** Single host 192.168.100.10.

2. Después, creamos un alias en Firewall/Aliases/Ports. Un alias es un conjunto de reglas que se pueden agrupar. En este caso vamos a agrupar diferentes puertos web (80 y 443 TCP), para que se puedan agregar directamente a un grupo de equipos la misma regla.

The screenshot shows the 'Properties' section of the alias configuration. Key details include:

- Name:** webs.
- Description:** Puerto web.
- Type:** Port(s).
- Port(s):**
  - Hint:** Enter ports as desired, with a single port or port range per entry. Port ranges can be expressed by separating with a colon.
  - Port:** 80 (Entry added Thu, 10 Mar 2022 12:37:27 +0100) with a Delete button.
  - Port:** 443 (Entry added Thu, 10 Mar 2022 12:37:27 +0100) with a Delete button.

### 3. Vamos luego a Firewall/Rules/DMZ y aplicamos este alias a las rules de DMZ

**Destination**

**Destination**: any  
**Destination Port Range**: (other) From Custom To Custom  
 Specify the destination port or port range for this rule. The "To" field may be left empty if only filtering a single port.

**Extra Options**

**Log**: 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**: webs  
 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.

**Advanced Options**: [Display Advanced](#)

### 4. También creamos otra rule para el acceso de DMZ a DNS por el puerto 53 TCP/UDP

**Firewall / Rules / DMZ**

The changes have been applied successfully. The firewall rules are now reloading in the background. [Monitor](#) the filter reload progress.

Floating WAN IT **DMZ** OpenVPN

**Rules (Drag to Change Order)**

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
0 / 0 B	IPv4 TCP/UDP	*	*	*	53 (DNS)	*	none		DNS	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Save</a> <a href="#">Separator</a>
0 / 0 B	IPv4 TCP	*	*	*	webs	*	none		webs	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Save</a> <a href="#">Separator</a>

### 5. Después se asigna la interfaz DMZ a otra máquina, en este caso, a otro Kali, colocando el adaptador de red en vmnet3. Para tener conexión en esta máquina, creamos una nueva regla en Firewall/Rules/DMZ para que permita conexiones ICMP.

**Firewall / Rules / DMZ**

Floating WAN IT **DMZ** OpenVPN

**Rules (Drag to Change Order)**

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
0 / 840 B	IPv4 ICMP any	*	*	*	*	*	none			<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Save</a> <a href="#">Separator</a>
0 / 2 KIB	IPv4 TCP/UDP	*	*	*	53 (DNS)	*	none		DNS	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Save</a> <a href="#">Separator</a>
0 / 0 B	IPv4 TCP	*	*	*	webs	*	none		webs	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Save</a> <a href="#">Separator</a>

## 2. Instalación de herramientas

### 2.1. Suricata

Suricata es una herramienta IDS (Intrusion Detection System), utilizada para controlar el tráfico de red. Permite rastrear eventos de seguridad que pueden indicar ataques o posibles intrusiones en equipos de la red.

1. En primer lugar, en la máquina Kali IT con usuario root instalamos Suricata con el comando

```
apt install suricata
```

2. Levantamos el servicio. Con la flag `-c` utilizamos el archivo de configuración por defecto y con `-i` la red o IP que se quiere escuchar, en este caso seleccionamos toda la red eth0. Comprobamos con systemctl status que el servicio está activo y funciona con normalidad.

```
(kali㉿kali)-[~/var/log/suricata]
$ sudo suricata -c /etc/suricata/suricata.yaml -i eth0
[sudo] password for kali:
17/3/2022 -- 11:39:10 - <Notice> - This is Suricata version 6.0.4 RELEASE running in SYSTEM mode
17/3/2022 -- 11:39:11 - <Notice> - all 4 packet processing threads, 4 management threads initialized, engine started.
^C17/3/2022 -- 11:54:53 - <Notice> - Signal Received. Stopping engine.
17/3/2022 -- 11:54:54 - <Notice> - Stats for 'eth0': | pkts: 101, drop: 0 (0.00%), invalid checksum: 0 (0.00%)
[kali㉿kali)-[~/var/log/suricata][You are unable to load any pages, check your computer's network connection.]
$ sudo systemctl start suricata.service
[sudo] password for kali:
● suricata.service - Suricata IDS/IDP daemon
   Loaded: loaded (/lib/systemd/system/suricata.service; disabled; vendor preset: disabled)
     Active: active (running) since Thu 2022-03-17 11:55:10 EDT; 8s ago
       Docs: man:suricata(8)
              man:suricatasc(8)
              https://suricata-ids.org/docs/
      Process: 68301 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid (code=exited, status=0/SUCCESS)
 Main PID: 68304 (Suricata-Main)
    Tasks: 10 (limit: 4572)
   Memory: 57.7M
      CPU: 3.007s
     CGroup: /system.slice/suricata.service
             └─68304 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid

Mar 17 11:55:10 kali systemd[1]: Starting Suricata IDS/IDP daemon...
Mar 17 11:55:10 kali suricata[68301]: 17/3/2022 -- 11:55:10 - <Notice> - This is Suricata version 6.0.4 RELEASE running in SYSTEM mode
Mar 17 11:55:10 kali systemd[1]: Started Suricata IDS/IDP daemon.
```

3. Vamos a editar el fichero suricata.yaml que está en /etc/suricata, dejando la configuración por defecto, creamos nuevo fichero de reglas

```
default-rule-path: /etc/suricata/rules
rule-files:
  - suricata.rules
  - reglas.rules
```

4. Vamos a /etc/suricata/rules para editar el archivo reglas.rules. Creamos una nueva regla que recoja los de todas las conexiones que se realicen en la red, que provengan de cualquier sitio.

```
GNU nano 6.0
reglas.rules
$alert tcp any any → 192.168.1.104 22 (msg:"Conexion SSH detectada"; sid:100; priority:1;)
$alert tcp any any → any any (msg:"Escaneo de puertos interno"; sid:2; fragbits:1D; dsiz:0; flags:S,12; ack:0; window:1024; threshold:type both, track by_dst, count 1, seconds:1)
```

5. Puede comprobarse su funcionamiento lanzando un escaneo nmap desde la máquina Kali DMZ. Podemos ver los logs recogidos por Suricata con `tail -f fast.log` en el directorio `/var/log/suricata`

```
kali@kali: /var/docker/docker-elk × kali@kali: ~ ×
(kali㉿kali)-[~]
$ sudo nmap -sS 192.168.100.10
[sudo] password for kali:
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-18 02:37 EDT
Nmap scan report for 192.168.100.10
Host is up (0.0039s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
53/tcp    closed domain
80/tcp    closed http
443/tcp   closed https

Nmap done: 1 IP address (1 host up) scanned in 4.99 seconds
```

```
(kali㉿kali)-[~/var/log/suricata]
$ tail -f fast.log
03/13/2022-16:41:52.310796 [**] [1:1:0] Network control [**] [Classification: (null)] [Priority: 1] {TCP} 192.168.1.104:30718 → 192.168.1.47:53416
03/13/2022-16:41:52.312958 [**] [1:1:0] Network control [**] [Classification: (null)] [Priority: 1] {TCP} 192.168.1.47:53416 → 192.168.1.104:35500
03/13/2022-16:41:52.313970 [**] [1:1:0] Network control [**] [Classification: (null)] [Priority: 1] {TCP} 192.168.1.104:33500 → 192.168.1.47:53416
03/13/2022-16:42:07.245767 [**] [1:1:0] Network control [**] [Classification: (null)] [Priority: 1] {TCP} 192.168.1.104:14412 → 34.210.39.83:5316
03/13/2022-16:42:07.428153 [**] [1:1:0] Network control [**] [Classification: (null)] [Priority: 1] {TCP} 34.210.39.83:4443 → 192.168.1.104:44190
03/13/2022-16:53:12.866060 [**] [1:2:0] Escaneo de puertos interno [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.1.47:63207 → 192.168.1.104:443
03/14/2022-02:08:31.750047 [**] [1:2:0] Escaneo de puertos interno [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.200.102:54197 → 192.168.100.10:443
03/17/2022-11:41:23.691189 [**] [1:2:0] Escaneo de puertos interno [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.200.103:48304 → 192.168.100.10:443
03/17/2022-11:56:10.950258 [**] [1:2:0] Escaneo de puertos interno [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.200.103:42298 → 192.168.100.10:443
03/17/2022-14:28:37.506499 [**] [1:2:0] Escaneo de puertos interno [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.200.103:38142 → 192.168.100.10:443
03/18/2022-02:37:51.854315 [**] [1:2:0] Escaneo de puertos interno [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.200.103:50507 → 192.168.100.10:443
```

6. Dejamos activo el servicio para utilizarlo posteriormente en la ingestión y visualización de los logs en la máquina Kali DMZ con Elastic Stack.

## 2.2. Elastic Stack

Elastic Stack es una combinación de tres herramientas que sirven para recoger, procesar, almacenar, analizar y mostrar los datos de tráfico de red de un equipo o una red o sistema de equipos.

1. Vamos a realizar la instalación de Elastic Stack (ELK) con docker. En la máquina Kali en que lo vamos a instalar (DMZ) clonamos el repositorio de GitHub con el contenedor docker de ELK, situándonos previamente en una carpeta llamada docker que habremos creado previamente dentro de `/var`.

```
(kali㉿kali)-[~/var/docker]
└─$ git clone https://github.com/deviantony/docker-elk
```

2. Dentro del directorio `/docker-elk`, instalamos Docker Compose y modificamos el permiso de ejecución

```
(kali㉿kali)-[~/var/docker/docker-elk]
└─$ sudo curl -L "https://github.com/docker/compose/releases/download/1.26.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

(kali㉿kali)-[~/var/docker/docker-elk]
└─$ sudo chmod +x /usr/local/bin/docker-compose
```

3. Levantamos el servicio Docker Compose y ya tenemos ELK corriendo

```
(kali㉿kali)-[~/var/docker/docker-elk]
└─$ sudo docker-compose up
```

4. Accedemos a la interfaz web desde el navegador de esta maquina a `http://127.0.0.1:5601`. Nos logueamos con user: elastic y password: changeme.

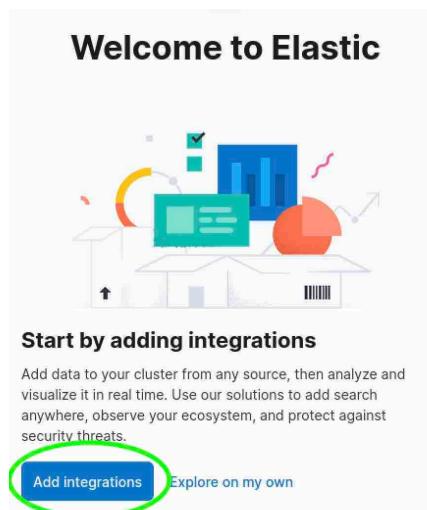
### 3. Conexión entre equipos de la red interna

En este apartado se describen los pasos seguidos para las siguientes fases del ejercicio:

- El IT debe correr Suricata y poder conectarse al ELK vía Kibana para mostrar los logs.
- El equipo DMZ2 debe alojar un honeypot sin acceso a las otras redes (solo para transmitir logs) pero si debe ser accesible desde la red WAN.
- El servidor ELK debe almacenar y mostrar los logs del honeypot.

#### 3.1. Integración de Suricata en Elastic Stack

1. En primer lugar vamos a añadir la integración con Suricata en el ELK.
2. Lo podemos agregar con un Fleet, un equipo que va a agregando servicios o de manera individual, como va a ser el caso. Seguimos los siguientes pasos:



**Suricata Events** Elastic Agent

**Suricata Integration**  
This integration is for [Suricata](#). It reads the EVE JSON output file. The EVE output writes alerts, anomalies, metadata, file info and protocol specific records as JSON.

**Compatibility**  
This module has been developed against Suricata v4.0.4, but is expected to work with other versions of Suricata.

**EVE**  
An example event for eve looks as following:

Version 1.6.0 [Add Suricata Events](#)

Screenshots 1 of 2

## Suricata Events integration added

To complete this integration, add **Elastic Agent** to your hosts to collect data and send it to Elastic Stack

[Add Elastic Agent later](#)

[Add Elastic Agent to your hosts](#)

**Suricata Events**  
Collect and parse event logs from Suricata instances with Elastic Agent.

All categories 246

Category	Count
AWS	24
Azure	22
Cloud	37
Communications	3
Config management	2
Containers	12
Custom	22

**2 Where to add this integration?**

**Create agent policy**  
Add this integration to a new set of hosts by creating a new agent policy. You can add agent in the next step.

**New agent policy name**

Collect system logs and metrics

[Advanced options](#)

[Save and continue](#)

3. Abrimos la pagina de descarga del agente en otra pestaña del navegador y descargamos la Policy (archivo .yml).

The top screenshot shows the 'Add agent' dialog in the Suricata Events interface. The 'Go to download page' button is circled in green. The bottom screenshot shows the Elastic Agent 8.1.0 download page on elastic.co. The 'LINUX 64-BIT sha' link is circled in green.

4. Nos llevamos tanto el archivo Elastic Agent 8.1.0 descargado como el archivo .yml de la Policy al directorio Downloads del equipo Kali IT

5. Descomprimimos el archivo Elastic Agent 8.1.0 con

```
(kali㉿kali)-[~]
$ cd Downloads
(kali㉿kali)-[~/Downloads]
$ tar xzvf elastic-agent-8.1.0-linux-x86_64.tar.gz
```

## 6. El archivo .yml de la Policy lo copiamos dentro del directorio del elastic-agent 8.1.0

```
(kali㉿kali)-[~/Downloads]
$ elastic-agent-8.1.0-linux-x86_64
(kali㉿kali)-[~/Downloads/elastic-agent-8.1.0-linux-x86_64] your hosts to collect data and send it to the Fleet
$ mv elastic-agent.yml elastic-agent.yml.bak
(kali㉿kali)-[~/Downloads/elastic-agent-8.1.0-linux-x86_64]
$ cd ..
(kali㉿kali)-[~/Downloads]
$ ls
elastic-agent-8.0.0-linux-x86_64  elastic-agent-8.1.0-linux-x86_64.tar.gz  elastic-agent.yml.bak
elastic-agent-8.1.0-linux-x86_64  elastic-agent.yml
Install the Elastic Agent on the hosts you wish to monitor. Do not install this agent policy on a host containing Fleet Server. You can install this agent policy on a host containing Fleet Server. You can enroll this host into Fleet by running fleetctl enroll <ip> in your agent's download page.
Linux users: We recommend the installer (Apt) over system packages.
Windows users: We recommend the Windows installer (exe) over system packages.
(kali㉿kali)-[~/Downloads]
$ cp elastic-agent.yml elastic-agent-8.1.0-linux-x86_64/elastic-agent.yml
(kali㉿kali)-[~/Downloads]
$ elastic-agent-8.1.0-linux-x86_64
(kali㉿kali)-[~/Downloads/elastic-agent-8.1.0-linux-x86_64]
$ ls
data      elastic-agent.reference.yml  elastic-agent.yml.bak  NOTICE.txt
elastic-agent  elastic-agent.yml        LICENSE.txt        README.md
Configure the agent
Copy this policy to the /etc/elasticsearch.yml on the host where the Elastic Agent is installed. Modify the username and password in the outside section of elastic-agent.yml to use your

```

## 8. Con usuario root vamos a /opt/Elastic/Agent y modificamos el archivo elastic-agent.yml para introducir la IP de la máquina Kali (DMZ) que aloja el ELK y el usuario y contraseña de inicio de sesión del Stack.

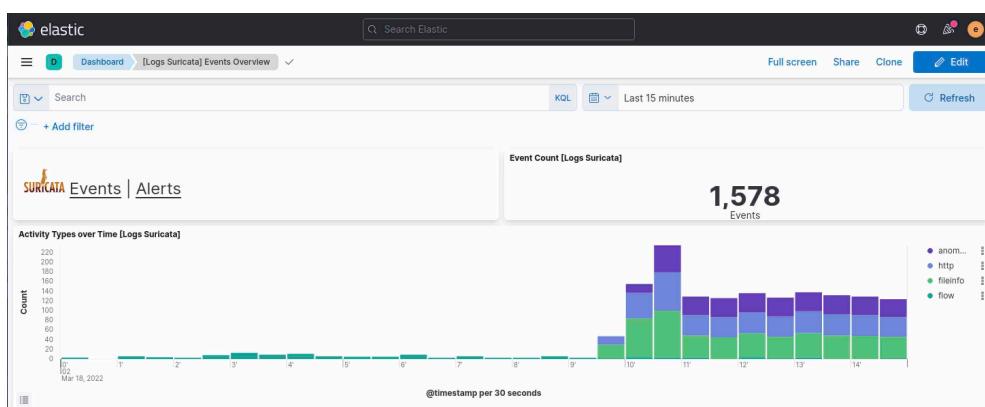
```
GNU nano 6.0                                     elastic-agent.yml
id: 73bcba40-a514-11ec-9737-d161b21155e0
revision: 2
outputs:
  default:
    type: elasticsearch
    hosts:
      - 'http://192.168.200.103:9200'
    username: 'elastic'
    password: 'changeme'
```

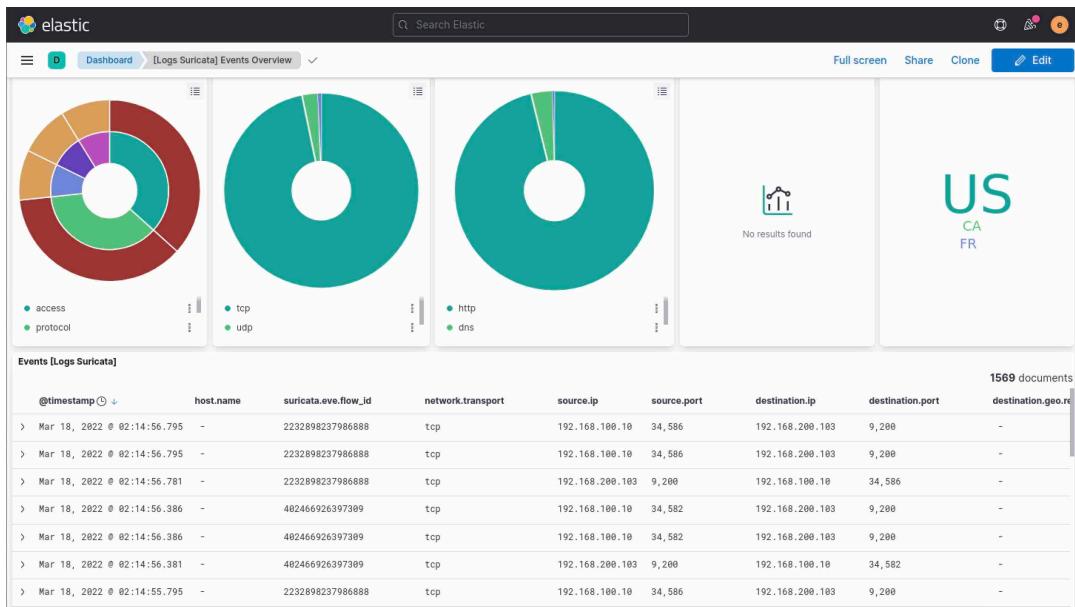
## 9. Ejecutamos elastic-agent como servicio y sin fleet (options y:n)

```
(kali㉿kali)-[~/Downloads/elastic-agent-8.1.0-linux-x86_64]
$ ls
data      elastic-agent.reference.yml  elastic-agent.yml.bak  NOTICE.txt
elastic-agent  elastic-agent.yml        LICENSE.txt        README.md
(kali㉿kali)-[~/Downloads/elastic-agent-8.1.0-linux-x86_64]
$ sudo ./elastic-agent install
[sudo] password for kali:
Configure the agent
Elastic Agent will be installed at /opt/Elastic/Agent and will run as a service. Do you want to continue?
? [Y/n]:y
Do you want to enroll this Agent into Fleet? [Y/n]:n
Elastic Agent is installed. Modify /etc/elasticsearch.yml and /etc/elastic-agent.yml in the outside section of elastic-agent.yml to use your
Elastic Agent has been successfully installed.
```

10. Ahora vamos a pfSense y creamos una regla en el firewall (Firewall/Rules/IT) para permitir las conexiones TCP desde la IP del IT (192.168.100.10) a la IP del DMZ (192.168.200.104), lo cual nos permitirá enviar los logs del Suricata instalado en el Kali IT al ELK alojado en el Kali DMZ, y permitirán la visualización de los mismos con Kibana. Abrimos todos los puertos del destino (Kali DMZ) para la comunicación entre ambas máquinas, ya que el enunciado del ejercicio no lo restringe.

11. Nos vamos a la interfaz del ELK en la Kali DMZ, cerramos las ventanas de integraciones y vamos al dashboard principal pulsando el logo de Elastic arriba a la izquierda en la interfaz. Nos movemos a Analytics/Discover desde el menú lateral izquierdo y podremos ver los eventos de tráfico registrados desde el Suricata de la máquina Kali IT





### 3.2. Creación e integración de honeypot en Elastic Stack

Un honeypot es un sistema o equipo informático que se configura como si fuese un sistema real con vulnerabilidades que pueden ser explotadas. Se encuentra aislado del resto de la red y monitorizado para poder observar y analizar cuales son los procedimientos que están siguiendo los posibles atacantes para vulnerar el sistema o equipo real.

Se han seguido estos pasos:

1. Vamos a instalar el honeypot “Cowrie” en la máquina Kali DMZ2. Primero hay que instalar el soporte para entornos virtuales de Python y otras dependencias necesarias con el comando

```
sudo apt-get install git python-virtualenv libssl-dev
libffi-dev build-essential libpython3-dev python3-minimal
authbind virtualenv
```

2. Creamos un usuario sin contraseña para cowrie con

```
sudo adduser --disabled-password cowrie
```

3. Instalamos cowrie con los siguientes comandos

```
sudo su - cowrie
```

```
git clone https://github.com/micheloosterhof/cowrie-dev.git
```

```
cd cowrie
```

4. Activamos el entorno virtual para cowrie con

The screenshot shows a terminal session on a Kali Linux system. The user runs several commands to update the package list, install Python dependencies, create a virtual environment, and activate it. The terminal output includes error messages for missing packages like 'python-virtualenv' and success messages for creating the environment and activating it.

```
(kali㉿kali)-[~/cowrie-dev]
$ sudo apt-get update
Hit:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Hit:2 http://ftp.free.fr/pub/kali kali-rolling InRelease
Reading package lists... Done

(kali㉿kali)-[~/cowrie-dev]
$ sudo apt-get install -y git python-virtualenv libssl-dev libffi-dev build-essential libpython3-dev python3-minimal authbind virtualenv
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package python-virtualenv is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'python-virtualenv' has no installation candidate

(kali㉿kali)-[~/cowrie-dev]
$ virtualenv --python=python3 cowrie-env
created virtual environment CPython3.9.10.final.0-64 in 580ms
  creator CPython3Posix(dest=/home/kali/cowrie-dev/cowrie-env, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/kali/.local/share/virtualenv)
  added seed packages: pip==22.0.2, setuptools==59.6.0, wheel==0.37.1
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

(kali㉿kali)-[~/cowrie-dev]
$ source cowrie-env/bin/activate
```

5. Seguidamente instalamos más dependencias con

```
pip install -U -r requirements.txt
```

5. Aunque Cowrie te permite simular servicio ssh y telnet, en este caso, sólo vamos a utilizar el honeypot para simular el servicio ssh, ubicándolo en el puerto 2222. Para ello cambiamos los siguientes parámetros el archivo **cowrie.cfg**, que previamente hemos copiado de **cowrie.cfg.dist** en **cowrie-dev/etc**. El resto de parámetros los dejamos como están por defecto.

```

GNU nano 5.4                                     cowrie.cfg
# DO NOT EDIT THIS FILE!
# Changes to default files will be lost on update and are difficult to
# manage and support.
#
# Please make any changes to system defaults by overriding them in
# cowrie.cfg
#
# To override a specific setting, copy the name of the stanza and
# setting to the file where you wish to override it.

# _____
# General Cowrie Options
# _____
[honeypot]

# Sensor name is used to identify this Cowrie instance. Used by the database
# logging modules such as mysql.
#
# If not specified, the logging modules will instead use the IP address of the
# server as the sensor name.
#
# (default: not specified)
sensor_name=database

# Hostname for the honeypot. Displayed by the shell prompt of the virtual
# environment
#
# (default: svr04)
hostname = database

# Directory where to save log files in.
#
# (default: log)
log_path = var/log/cowrie

```

```

# _____
# SSH Specific Options
# _____
[ssh]

# Enable SSH support
# (default: true)
enabled = true

```

```

# Endpoint to listen on for incoming SSH connections.
# See https://twistedmatrix.com/documents/current/core/howto/endpoints.html#servers
# (default: listen_endpoints = tcp:2222:interface=0.0.0.0)
# (use systemd: endpoint for systemd activation)
# listen_endpoints = systemd:domain=INET:index=0
# For both IPv4 and IPv6: listen_endpoints = tcp6:2222:interface=:\
# Listening on multiple endpoints is supported with a single space separator
# e.g listen_endpoints = "tcp:2222:interface=0.0.0.0 tcp:1022:interface=0.0.0.0" will result listening both on ports 2222 and 1022
# use authbind for port numbers under 1024
listen_endpoints = tcp:2222:interface=0.0.0.0

```

## 6. Arrancamos cowrie y comprobamos que está activo

```

[cowrie-env](kali㉿kali)-[~/cowrie-dev]
$ ./bin/cowrie start
Join the Cowrie community at: https://www.cowrie.org/slack/
Using activated Python virtual environment "/home/kali/cowrie-dev/cowrie-env"
Starting cowrie: [twistd --umask=0022 --pidfile=var/run/cowrie.pid --logger cowrie.python.logfile.logger cowrie ] ...
[cowrie-env](kali㉿kali)-[~/cowrie-dev]
$ ./bin/Cowrie status
cowrie is running (PID: 13851).
[cowrie-env](kali㉿kali)-[~/cowrie-dev]
$ 

```

A continuación, vamos a instalar en la misma máquina Kali DMZ2 la herramienta Filebeat de Elastic Stack, que permite enviar logs a ELK ya paseados, por lo que no necesita de enviarlos a Logstash y pueden visualizarse directamente con Kibana.

Los pasos a seguir son:

## 1. Instalamos desde root con los siguientes comandos

```

└# curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK

└(root㉿kali)-[~/home/kali]
└# echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elasticsearch-kali-rolling.list
deb https://artifacts.elastic.co/packages/7.x/apt stable main

└(root㉿kali)-[~/home/kali]
└# apt update
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13.7 kB]
Get:2 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [95.2 kB]
Hit:3 http://ftp.free.fr/pub/kali kali-rolling InRelease
Fetched 3,113 kB in 2s (1,594 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Reading state information... Done
1418 packages can be upgraded. Run 'apt list --upgradable' to see them.

└(root㉿kali)-[~/home/kali]
└# apt install filebeat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  filebeat
0 upgraded, 1 newly installed, 0 to remove and 1418 not upgraded.
Need to get 36.3 MB of archives.
After this operation, 144 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 filebeat amd64 7.17.1 [36.3 MB]
Fetched 36.3 MB in 1s (29.0 MB/s)
Selecting previously unselected package filebeat.
(Reading database ... 269954 files and directories currently installed.)
Preparing to unpack .../filebeat_7.17.1_amd64.deb ...
Unpacking filebeat (7.17.1) ...
Setting up filebeat (7.17.1) ...
Processing triggers for kali-menu (2021.1.4) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

```

## 2. Seguidamente configuramos el archivo filebeat.yml ubicado en /etc/filebeat

```

GNU nano 5.4                                         filebeat.yml *

# ===== Filebeat inputs =====
filebeat.inputs:
  # Each - is an input. Most options can be set at the input level, so
  # you can use different inputs for various configurations.
  # Below are the input specific configurations.

  # filestream is an input for collecting log messages from files.
  - type: log
    # Change to true to enable this input configuration.
    enabled: true

    # Paths that should be crawled and fetched. Glob based paths.
    paths:
      - /cowrie-dev/var/log/cowrie/cowrie.json
      #- c:\programdata\elasticsearch\logs\*

    # Exclude lines. A list of regular expressions to match. It drops the lines that are
    # matching any regular expression from the list.
    #exclude_lines: ['^DBG']

    # Include lines. A list of regular expressions to match. It exports the lines that are
    # matching any regular expression from the list.
    #include_lines: ['^ERR', '^WARN']

    # Exclude files. A list of regular expressions to match. Filebeat drops the files that
    # are matching any regular expression from the list. By default, no files are dropped.
    #prospector.exclude_files: ['.gz$']

    # Optional additional fields. These fields can be freely picked
    # to add additional information to the crawled log files for filtering
    #fields:
    #  level: debug
    #  review: 1

```

```
# ━━━━━━━━━━━━ Elasticsearch Output ━━━━━━━━━
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["192.168.200.103:9200"]

  # Protocol - either `http` (default) or `https`.
  #protocol: "https"

  # Authentication credentials - either API key or username/password.
  #api_key: "id:api_key"
  username: "elastic"
  password: "changeme"
```

En el apartado Filebeat inputs hemos habilitado y establecido la ruta de la cual Filebeat recogerá los logs de cowrie.

Después, en Elasticsearch Output indicamos la IP y puerto de la maquina DMZ donde tenemos alojado Elastic Stack y el usuario y password de acceso.

### 3. Arrancamos Filebeat y comprobamos que está activo

```
(root@kali)-[~/etc/filebeat]
# nano filebeat.yml

(root@kali)-[~/etc/filebeat]
# systemctl enable filebeat
Synchronizing state of filebeat.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable filebeat

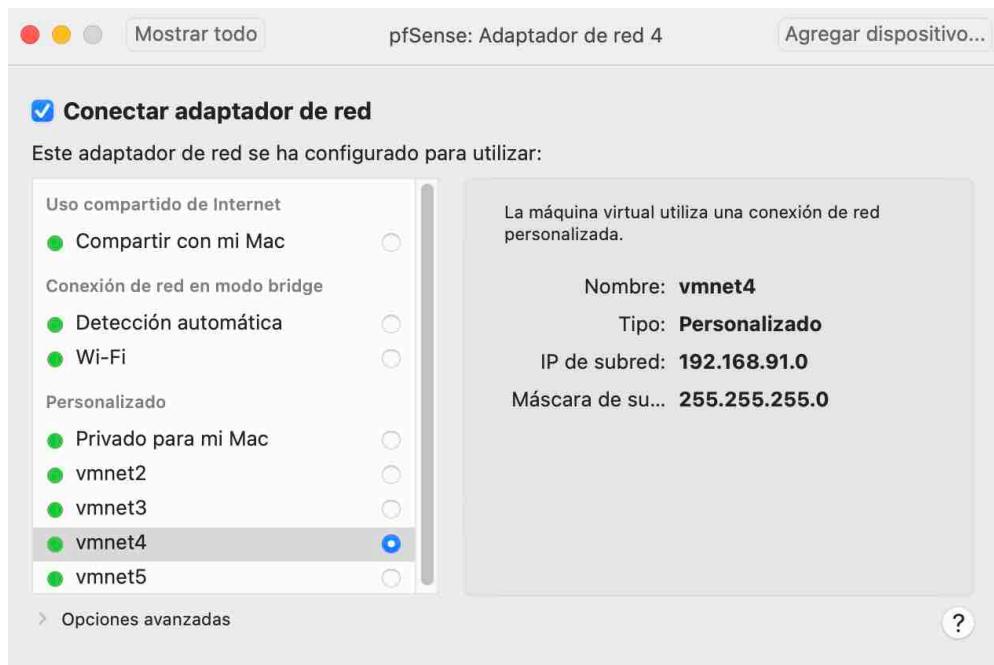
(root@kali)-[~/etc/filebeat]
# systemctl start filebeat
[root@kali ~]# systemctl status filebeat
● filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
   Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset: disabled)
     Active: active (running) since Sat 2022-03-19 12:12:30 EDT; 35min ago
       Docs: https://www.elastic.co/beats/filebeat
      Main PID: 13139 (filebeat)
         Tasks: 9 (limit: 2290)
        Memory: 58.2M
          CPU: 4.150s
        CGroup: /system.slice/filebeat.service
                └─13139 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc/filebeat/filebeat.yml --path.home /usr/share/filebeat --path.co

Mar 19 12:46:53 kali filebeat[13139]: 2022-03-19T12:46:53.901-0400      INFO      [file_watcher]      filestream/fswatch.go:137      Start >
Mar 19 12:47:03 kali filebeat[13139]: 2022-03-19T12:47:03.900-0400      INFO      [file_watcher]      filestream/fswatch.go:137      Start >
Mar 19 12:47:03 kali filebeat[13139]: 2022-03-19T12:47:03.902-0400      INFO      [monitoring]      log/log.go:184      Non-zero metrics in>
Mar 19 12:47:13 kali filebeat[13139]: 2022-03-19T12:47:13.900-0400      INFO      [file_watcher]      filestream/fswatch.go:137      Start >
Mar 19 12:47:23 kali filebeat[13139]: 2022-03-19T12:47:23.901-0400      INFO      [file_watcher]      filestream/fswatch.go:137      Start >
Mar 19 12:47:33 kali filebeat[13139]: 2022-03-19T12:47:33.901-0400      INFO      [file_watcher]      filestream/fswatch.go:137      Start >
Mar 19 12:47:39 kali filebeat[13139]: 2022-03-19T12:47:39.905-0400      INFO      [monitoring]      log/log.go:184      Non-zero metrics in>
Mar 19 12:47:39 kali filebeat[13139]: 2022-03-19T12:47:39.538-0400      ERROR     [esclientleg]      transport/logging.go:37      Error di>
Mar 19 12:47:43 kali filebeat[13139]: 2022-03-19T12:47:43.901-0400      INFO      [file_watcher]      filestream/fswatch.go:137      Start >
Mar 19 12:47:53 kali filebeat[13139]: 2022-03-19T12:47:53.901-0400      INFO      [file_watcher]      filestream/fswatch.go:137      Start >

lines: 1-21/21 (END)
```

Llegados este punto, vamos a crear una nueva red en pfSense para poder hospedar esta máquina DMZ2.

1. Creamos un nuevo adaptador de red (4) en los ajustes de la máquina virtual y le asignamos la red vmnet4



2. Comprobamos que la nueva red está creada (opt2)

```
FreeBSD/amd64 (UTM-1.marcos.local) (ttyv0)
VMware Virtual Machine - Netgate Device ID: 543ff86be1557a74d9d5
*** Welcome to pfSense 2.6.0-RELEASE (amd64) on UTM-1 ***
WAN (wan)      -> em0      -> v4/DHCP4: 192.168.1.67/24
IT (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

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: ■
```

3. Ahora, en la herramienta pfSense, vamos a asignar a la nueva red opt2 creada a la interfaz DMZ2

4. Comprobamos que la red DMZ2 está configurada correctamente yendo al dashboard principal

Interface	Status	Link Speed	IP Address
WAN	↑	1000baseT <full-duplex>	192.168.1.67
IT	↑	1000baseT <full-duplex>	192.168.100.1
DMZ	↑	1000baseT <full-duplex>	192.168.200.1
DMZ2	↑	1000baseT <full-duplex>	192.168.250.1

6. Habilitamos el DHCP server y establecemos los DNS servers que utilizaremos con DMZ2

**General Options**

- Enable**:  Enable DHCP server on DMZ2 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 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 below (i.e. for this interface) will get an IP address within this scope/range.
- Ignore denied clients**:  Denied clients will be ignored rather than rejected.  
This option is not compatible with failover and cannot be enabled when a Failover Peer IP address is configured.
- Ignore client identifiers**:  If a client includes a unique identifier in its DHCP request, that UID will not be recorded in its lease.  
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.
- Subnet**: 192.168.250.0
- Subnet mask**: 255.255.255.0
- Available range**: 192.168.250.1 - 192.168.250.254
- Range**: From 192.168.250.100 To 192.168.250.200

**Servers**

- WINS servers**: WINS Server 1
- WINS servers**: WINS Server 2
- DNS servers**: 192.168.250.1
- DNS servers**: 1.1.1.1
- DNS servers**: 8.8.8.8
- DNS servers**: DNS Server 4  
Leave blank to use the system default DNS servers: this interface's IP if DNS Forwarder or Resolver is enabled, otherwise the servers configured on the System / General Setup page.

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

**Other Options**

- Gateway**: 192.168.250.1  
The default is to use the IP on this interface of the firewall as the gateway. Specify an alternate gateway here if this is not the correct gateway for the network. Type 'none' for no gateway assignment.

5. Seguidamente vamos a crear las reglas necesarias para que DMZ2 (que aloja cowrie y filebeat) se pueda conectar a DMZ (que aloja ELK, puertos 9200 Kibana y 5601 Elasticsearch) y mostrar los logs del honeypot. Estas son las configuraciones que necesitamos:

The screenshot shows the 'Edit Firewall Rule' page. The 'Action' dropdown is set to 'Pass'. The 'Disabled' checkbox is unchecked. The 'Interface' is set to 'DMZ2'. The 'Address Family' is 'IPv4'. The 'Protocol' is 'TCP'. In the 'Source' section, the 'Source' dropdown is set to 'DMZ2 address'. In the 'Destination' section, the 'Destination' dropdown is set to 'DMZ address', and the 'Destination Port Range' is '(other) 9200 -> (other) 9200'.

This screenshot shows the same 'Edit Firewall Rule' page as the first one, but with a different destination port range. The 'Destination Port Range' is now '(other) 5601 -> (other) 5601'.

6. Despu s creamos reglas Block para que no permitan el tr fico proveniente de DMZ2 al resto de equipos de la red (IT y DMZ). Lo  nico que permitimos es el tr fico desde DMZ2 (honeypot) a DMZ (ELK) por los puertos 9200 y 5601 situando las anteriores reglas Pass por encima de las reglas Block, de esta manera:

**Firewall / Rules / DMZ2**

The changes have been applied successfully. The firewall rules are now reloading in the background.  
Monitor the filter reload progress.

Floating WAN IT DMZ DMZ2 OpenVPN

**Rules (Drag to Change Order)**

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	0 /2 KIB	IPv4 ICMP any	*	*	*	*	*	none		
<input type="checkbox"/>	0 /0 B	IPv4 TCP	DMZ2 address	*	DMZ address	9200	*	none		
<input type="checkbox"/>	0 /0 B	IPv4 TCP	DMZ2 address	*	DMZ address	5601	*	none		
<input type="checkbox"/>	0 /0 B	IPv4+6 *	*	*	DMZ2 address	*	*	none		
<input type="checkbox"/>	0 /1 KIB	IPv4+6 *	DMZ2 net	*	IT net	*	*	none		
<input type="checkbox"/>	0 /0 B	IPv4+6 *	DMZ2 net	*	DMZ net	*	*	none		
<input type="checkbox"/>	0 /0 B	IPv4 TCP	DMZ2 address	*	DMZ address	*	*	none		

**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:  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+IPv6  
Select the Internet Protocol version this rule applies to.

Protocol: Any  
Choose which IP protocol this rule should match.

**Source**

Source:  Invert match DMZ2 net Source Address /

**Destination**

Destination:  Invert match IT net Destination Address /

Comprobamos que funciona, deshabilitando primero la regla y conectando desde DMZ2 a IT por el puerto 80, logrando la conexión.

**Firewall / Rules / DMZ2**

The changes have been applied successfully. The firewall rules are now reloading in the background.  
Monitor the filter reload progress.

Floating WAN IT DMZ DMZ2 OpenVPN

**Rules (Drag to Change Order)**

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	0 /2 KIB	IPv4 ICMP any	*	*	*	*	*	none		
<input type="checkbox"/>	0 /0 B	IPv4 TCP	DMZ2 address	DMZ address	9200	*	*	none		
<input checked="" type="checkbox"/>	0 /0 B	IPv4+6 *	DMZ2 net	IT net	*	*	*	none		
<input type="checkbox"/>	0 /0 B	IPv4 TCP	DMZ2 address	DMZ address	5601	*	*	none		
<input type="checkbox"/>	0 /0 B	IPv4 TCP	DMZ2 address	DMZ address	*	*	*	none		

**Apache2 Debian Default Page**

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should [replace this file](#) (located at `/var/www/html/index.html`) before you start to operate your HTTP service.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

**Configuration Overview**

Debian's Apache2 default configuration is different from the upstream default configuration, and split several files optimized for the needs of Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation and for the documentation for the web server itself can be found by accessing the [manual](#) if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|- apache2.conf
|   '-- ports.conf
|   '-- mods-enabled
|       '-- ...
```

Habilitamos la regla Block y ya la conexión DMZ2-IT por el puerto 80 no se puede hacer

Para terminar, vamos a comprobar que Cowrie está recogiendo los logs correctamente y como se visualizan en Kibana gracias a Filebeat.

1. En Elastic Stack, vamos a Management/Index Management y comprobamos que muestra la conexión con Filebeat

## 2. Comprobamos que se muestran correctamente los logs en Kibana/Discover haciendo con un nmap desde otra maquina al puerto del honeypot

The terminal window shows the following nmap command and its output:

```
nmap -sT -p2222 192.168.250.101
Nmap done: 1 IP address (1 host up) scanned in 0.26 seconds
[+] 192.168.250.101:2222
```

The Kibana interface displays Elasticsearch logs from March 20, 2022, at 03:52:16.901 to Mar 20, 2022, at 04:07:16.901. The logs show several entries related to filebeat agents attempting to connect to the Elasticsearch host.

Sample log entries from Kibana:

- > Mar 20, 2022 0 64:07:10.629 @timestamp: Mar 20, 2022 0 64:07:10.629 agent:ephemeral\_id: f92e4549-d7cb-4093-9b6e-21d11e57b955 agent.hostname: kali agent.id: b1419043-fe85-42cc-888c-01542d75b78c agent.name: kali agent.type: filebeat agent.version: 7.17.1 container.id: log ecs.version: 1.12.0 host.architecture: x86\_64 host.containerized: false host.hostname: kali host.id: 402ee910bea34399961536e5739c0372 host.ip: 192.168.250.101, fe80::2bc:29ff:fe97:2980,
- > Mar 20, 2022 0 64:07:10.629 @timestamp: Mar 20, 2022 0 64:07:10.629 agent:ephemeral\_id: f92e4549-d7cb-4093-9b6e-21d11e57b955 agent.hostname: kali agent.id: b1419043-fe85-42cc-888c-01542d75b78c agent.name: kali agent.type: filebeat agent.version: 7.17.1 container.id: log ecs.version: 1.12.0 host.architecture: x86\_64 host.containerized: false host.hostname: kali host.id: 402ee910bea34399961536e5739c0372 host.ip: 192.168.250.101, fe80::2bc:29ff:fe97:2980,
- > Mar 20, 2022 0 64:07:10.629 @timestamp: Mar 20, 2022 0 64:07:10.629 agent:ephemeral\_id: f92e4549-d7cb-4093-9b6e-21d11e57b955 agent.hostname: kali agent.id: b1419043-fe85-42cc-888c-01542d75b78c agent.name: kali agent.type: filebeat agent.version: 7.17.1 container.id: log ecs.version: 1.12.0 host.architecture: x86\_64 host.containerized: false host.hostname: kali host.id: 402ee910bea34399961536e5739c0372 host.ip: 192.168.250.101, fe80::2bc:29ff:fe97:2980,