



# Blue Team Práctica

## Introducción

Organigrama de la infraestructura:

## Infraestructura

### Máquinas virtuales

Pfsense, creación e instalación

Kali Linux S.O.

### Puesta a punto

Red LAN

Red DMZ

Checkpoint 1

Acceso a la red LAN

Checkpoint 2

Suricata

ELK

Checkpoint final

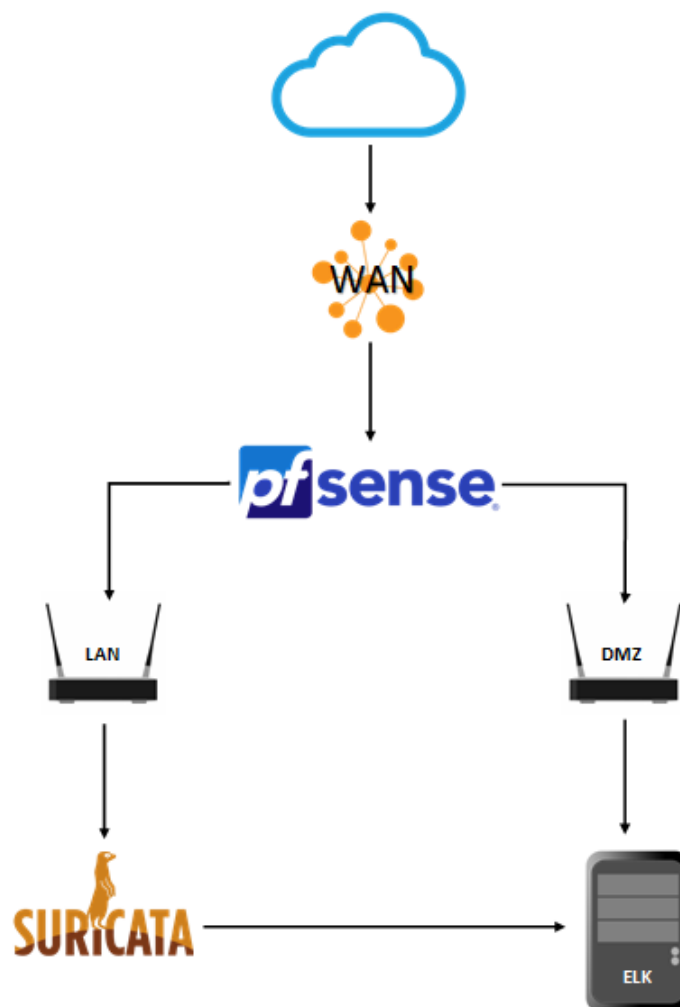
**Jose Manuel Gonzalez Gonzalez.** KeepCoding Espana.

# Introducción

Se requiere del montaje de una infraestructura “on premise”. La infraestructura debe cumplir con los siguientes requisitos:

- Contar con un **pfsense** que conecte las dos redes **DMZ** (desmilitarizada) y **LAN**.
- En la red **LAN** habrá una equipo **Kali** con ejecución de **Suricata** para reportar logs.
- En la red **DMZ** contaremos con un equipo **Kali** donde tendremos un servidor **ELK**.
- El servidor **ELK** debe almacenar y poder visualizar los logs de **SURICATA**.
- Suricata deberá de poder reportar logs al servidor **ELK**.

**Organigrama de la infraestructura:**



# Infraestructura

## Máquinas virtuales

Para la creación de la infraestructura necesitaremos descargar el siguiente software:



Crearemos la infraestructura utilizando Oracle Virtual Box.

### Download pfSense Community Edition

Product information, pfSense software announcements, and special offers. See our newsletter archive for past announcements. (view our privacy policy) Daily snapshot builds of our upcoming release are available for testing and evaluation. Join us on our forum to discuss. You can determine the files needed for your install by reading the rest of this

 <https://www.pfsense.org/download/>

### Get Kali | Kali Linux

A Kali Linux Live image on a CD/DVD/USB/PXE can allow you to have access to a full bare metal Kali install without needing to alter an already-installed operating system. This allows for quick easy access to the Kali toolset with all the advantages of a bare metal install.

 <https://www.kali.org/get-kali/>

Para las descargas de Kali, hemos seleccionado la opción de **maquinas virtuales**.

### Download - Suricata

 <https://suricata.io/download/>



Suricata lo instalaremos desde linea de comandos.

Para la descarga de ELK, lo haremos mas adelante desde el link:  
<https://github.com/deviantony/docker-elk>

## ▼ Pfsense, creación e instalación

Crearemos una máquina virtual nueva con los siguientes datos:

1. Nombre : UTM\_PFSENSE
2. Tipo: BSD
3. Version 64bits
4. Memoria 2GB
5. Crear nuevo disco virtual ahora

? X

← Crear máquina virtual

Nombre y sistema operativo

Nombre:

Carpeta de máquina:

Tipo:

Versión:

Tamaño de memoria

1024 MB

4 MB 24576 MB

Disco duro

☐ No añadir un disco duro virtual

☒ Crear un disco duro virtual ahora

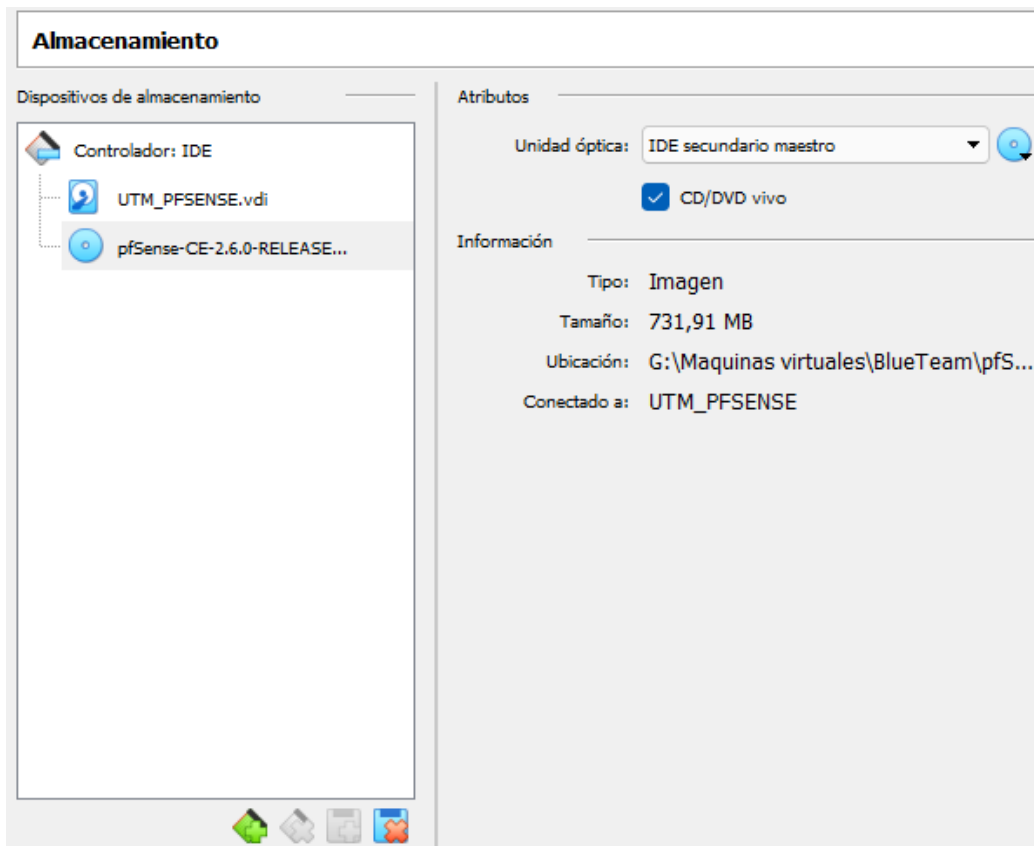
☐ Usar un archivo de disco duro virtual existente

Modo guiado Crear Cancelar

En la configuración del disco duro virtual crear con 16GB.

▶ En la puesta a punto de la máquina, tomaremos las siguientes configuraciones:

1. Entramos en Configuración.
2. En la pestaña Almacenamiento, seleccionamos el disco virtual "extraible" y seleccionamos la imagen de pfsense descargada anteriormente, y confirmamos el check-box "CD/DVD vivo".



Para la configuración de redes, en la pestaña Red, haremos las siguientes configuraciones:

#### Adaptador 1

- Habilitar adaptador de red
- Conectado a: Adaptador puente
- Nombre: Adaptador de red del host

#### Adaptador 2

- Habilitar adaptador de red
- Conectado a: Red Interna
- Nombre: LAN\*

#### Adaptador 3

- Habilitar adaptador de red
- Conectado a: Red Interna
- Nombre: DMZ\*

\*Para crear las redes, solo tenemos que pulsar encima e introducir nombre.

▶ A continuación, realizamos la instalación de **pfSense**

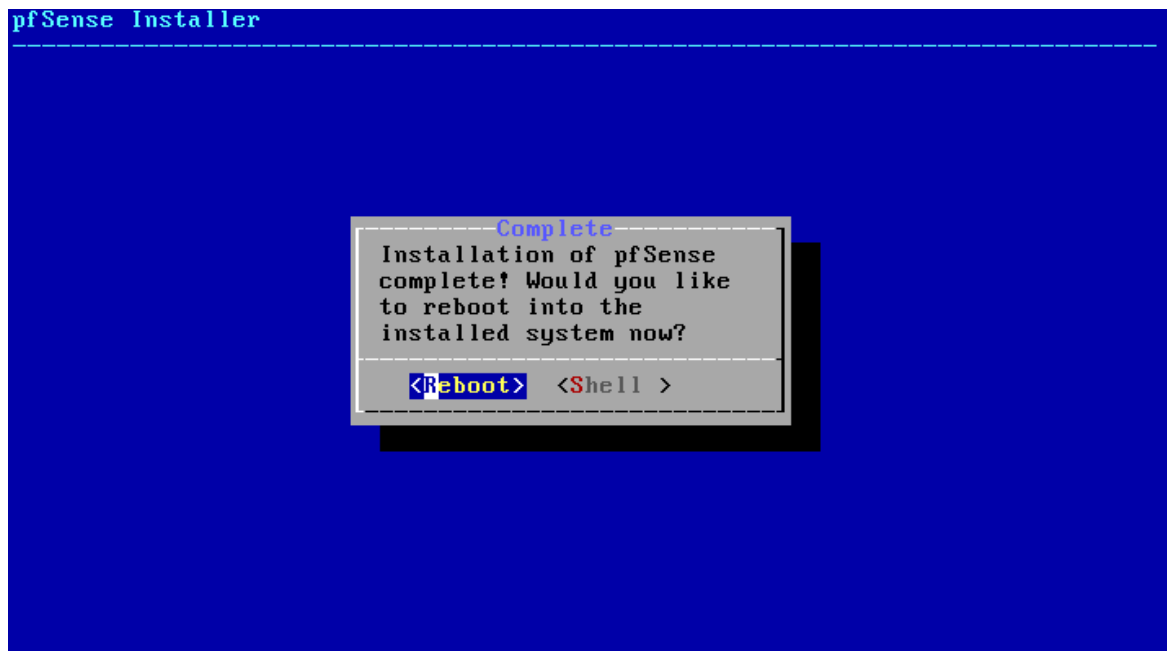
En cada uno de los pasos siguientes se debe pulsar siguiente, un paso por pantalla.

1. Iniciamos la maquina virtual
2. Aceptar terminos de uso
3. Instalar
4. Selecccion de layout del teclado.
5. El particionamiento sera por defecto, ZFS.
6. La configuracion de ZFS sera por defecto, procedemos a la instalacion.
7. De nuevo el tipo de dispositivo virtual sera stripe, sin redundancia.\*
8. El siguiente paso necesitamos pulsar espacio para seleccionar el discoduro donde instalar.

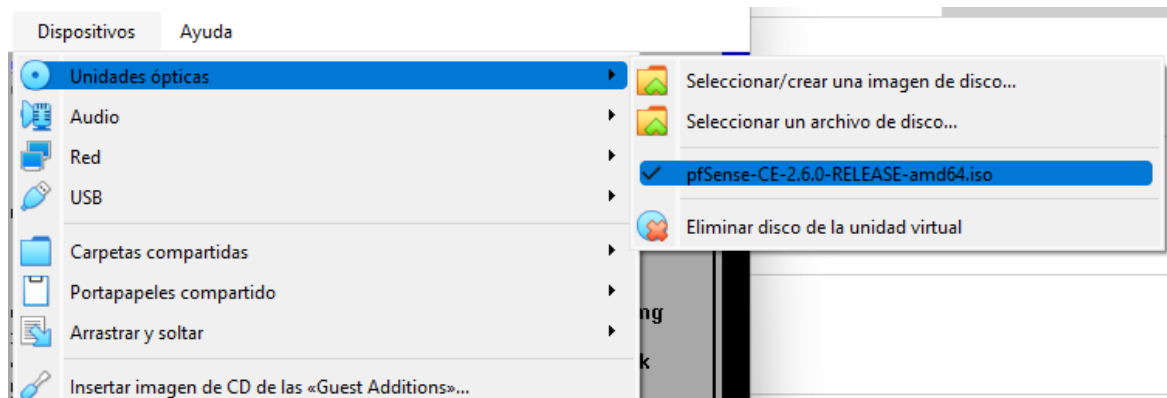
\*es interesante para la seguridad hardware.



9. A continuacion, tras pulsar OK, nos confirmara que estamos seguiros de borrar todos los da tos, pulsamos YES.
10. En este momento empieza su instalaci3n.
11. Posteriormente seleccionaremos NO, puesto que no queremos hacer ninguna modificaci3n y en la siguiente imagen seleccionaremos Reboot.



En este momento el equipo se reiniciara, y volverá a cargar para hacer la instalación. Para poder continuar sera necesario desmontar la iso de pfsense y posteriormente reiniciar la maquina virtual.



Pulsaremos sobre el archivo "pfsense-CE-2...." para desmontar la imagen.

## ▼ Kali Linux S.O.

Para alojar el software de Suricata y ELK crearemos dos maquinas iguales siguiendo los siguientes pasos:



Para en la creación de la maquinas, se indican la diferencia entre ellas.

1. Nos dirigimos a importar un servicio virtualizado en Archivo / Importar Servicio Virtualizado.
2. La fuente sera el archivo que hemos descargado anteriormente de la web de Kali, es interesante duplicarlo para crear las dos maquinas necesarias.
3. La interfaz nos mostrara informacion de la maquina virtual, para lo cual modificaremos la ubicación.

#### ← Importar servicio virtualizado

Fuente

Sistema de archivos local

actica\kali-linux-2022.1-virtualbox-amd64.ova

Configuración

Sistema virtual 1

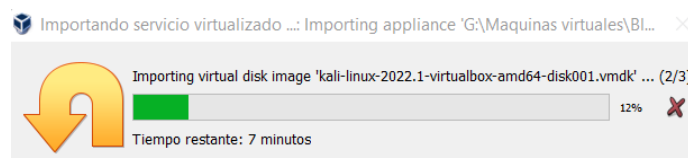
Nombre	kali-linux-2022.1-virtualbox-a...
Producto	Kali Linux
URL del producto	https://www.kali.org/
Vendedor	Offensive Security
URL del vendedor	https://www.offensive-security...
Versión	Rolling (2022.1) x64
Descripción	Kali Rolling (2022.1) x64...
Tipo de SO invitado	Debian (64-bit)
CPU	2
RAM	2048 MB
DVD	<input checked="" type="checkbox"/>
Controlador USB	<input checked="" type="checkbox"/>
Tarjeta de sonido	<input checked="" type="checkbox"/> ICH AC97
Adaptador de red	<input checked="" type="checkbox"/> Intel PRO/1000 MT Deskto...

Carpeta base de máquina: G:\Maquinas virtuales

Política de dirección MAC: Incluir solo las direcciones NAT de adaptador de red

Opciones adicionales: ☒ Importar discos como VDI

4. Pulsaremos sobre Importar, aceptaremos los términos de uso y empezara su importación.







En este punto, realizaremos de nuevo la importación de la maquina para contar con dos maquinas Kali, en mi caso, *Kali-linux-2022.1* y *Kali-Linux-2021-4*.



**Kali-Linux-2021.4**  
Apagada

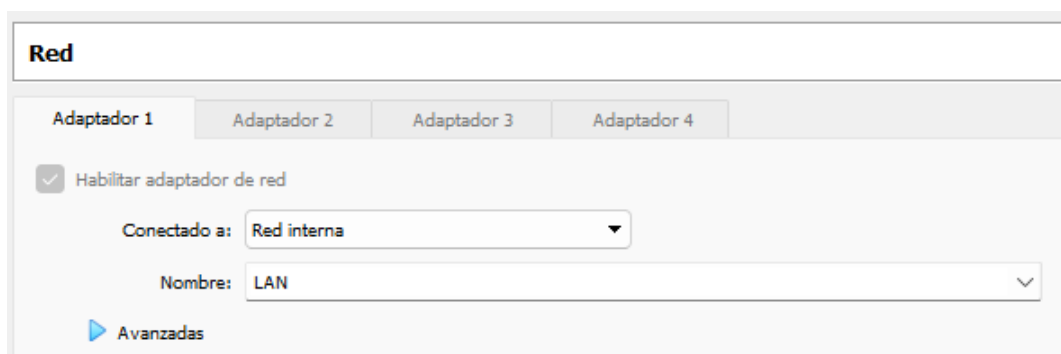


**kali-linux-2022.1-virtualbox-amd64**  
Apagada

▶ Para la puesta a punto de las dos maquinas, seguiremos los siguientes pasos:

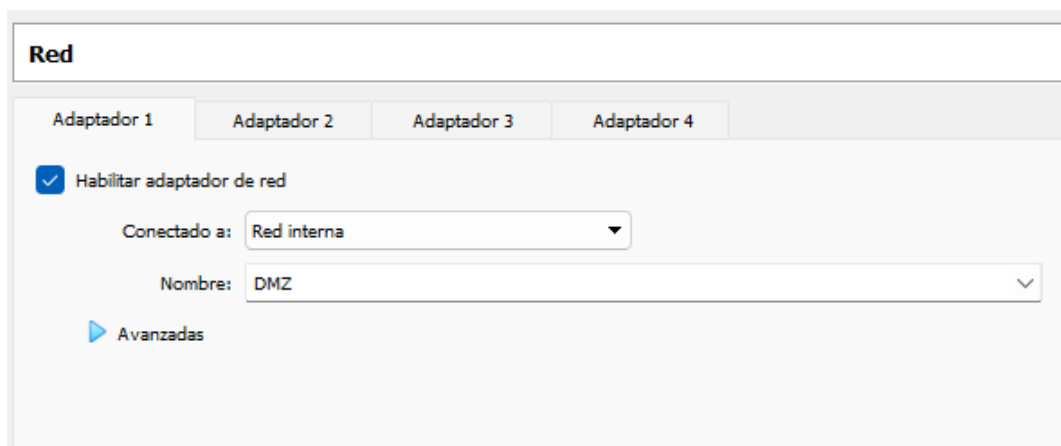
- Kali-linux-2021.4

1. Seleccionaremos la maquina y nos dirigiremos a su configuracion.
2. En el apartado Red, el primer adaptador tendra la siguiente configuracion:
  - Habilitar adaptador de red
  - Conectado a: Red Interna
  - Nombre: LAN



- Kali-linux-2022.1

1. Seleccionaremos la maquina y nos dirigiremos a su configuracion.
2. En el apartado Red, el primer adaptador tendra la siguiente configuracion:
  - Habilitar adaptador de red
  - Conectado a: Red Interna
  - Nombre: DMZ



▶ Para la primera de las maquinas, la conectada con la red LAN haremos la siguiente comprobación para conocer si el **pfSense** le ha proporcionado IP.

Para comprobarlo, lanzaremos el comando `ip a`:

```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group de
fault qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state U
P group default qlen 1000
    link/ether 08:00:27:43:73:bc brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.100/24 brd 192.168.1.255 scope global dynamic noprefixrou
te eth0
        valid_lft 7166sec preferred_lft 7166sec
    inet6 fe80::a00:27ff:fe43:73bc/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 02:42:1f:b8:7c:6b 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
```

La IP de la maquina es **192.168.111.100**

# Puesta a punto

## ▼ Red LAN

En este punto de la configuración de la infraestructura, accederemos a la maquina Kali, con IP 192.168.1.100 para configurar la red **LAN**.

1. Abrimos el navegador e introducimos la IP 192.168.1.1. El navegador nos mostrara una advertencia de seguridad que tendremos que aceptar para continuar. Para ello pulsaremos primero en el boton del numero 1 y posterior en el 2.

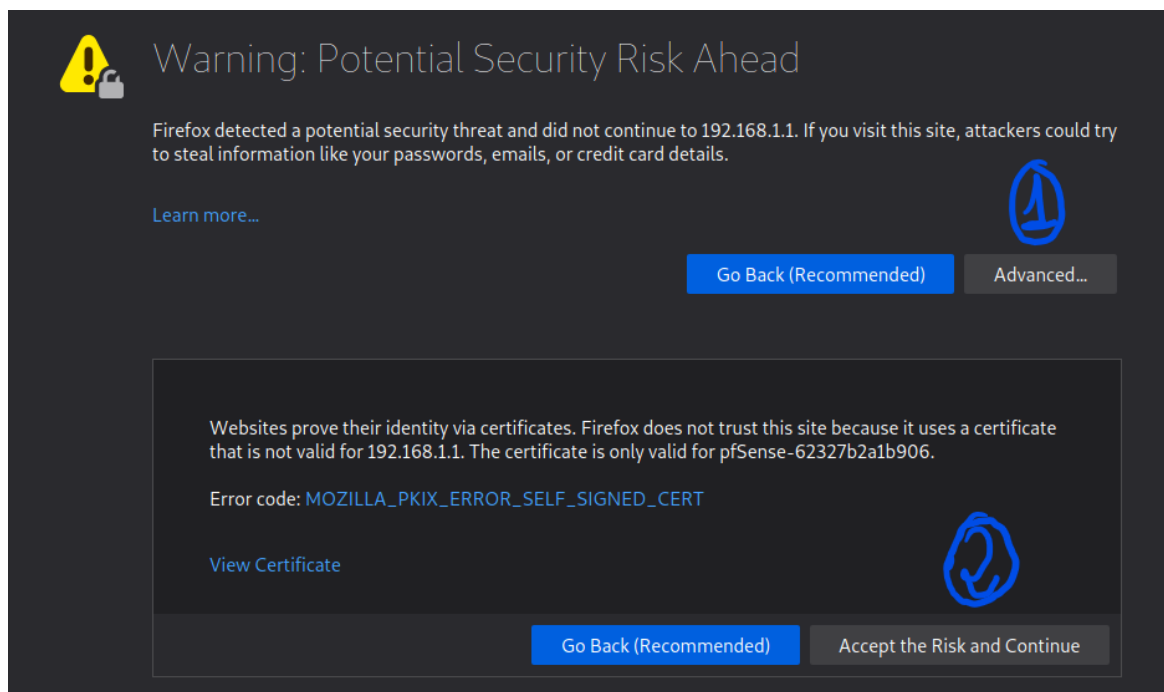


Imagen de aviso de riesgo de seguridad.

2. Para acceder a la plataforma deberemos introducir las siguientes credenciales:  
User ► admin  
Password ► pfsense

\*Una vez entrado al sistema, nos ofrecera cambiar la password.

► En la siguiente pantalla al ser la primera vez nos mostrara un asistente de configuración, donde iremos pulsando Next a medida que vamos introduciendo los

siguientes datos:

```
Hostname: UTM
Domain: practica.local
Primary DNS server: 1.1.1.1 (Cloudflare)
Secondary DNS server: 8.8.8.8 (Google)
Override DNS marcado
```

- Tips a tener en cuenta
  - Domain es el rango de dominio de nuestra red, al ser una misma red usamos la terminación “.local”.
  - Override DNS permite sobrescribir los DNS dados por nuestro proveedor IPS.

En la siguiente imagen configuraremos la zona horaria y el servidor de tiempo para sincronizar la misma, es importante para que todo funcione correctamente, sino puede ser que los sistemas se desincronicen y no funcionen correctamente.

**Time Server Information**

Please enter the time, date and time zone.

**Time server hostname**   
Enter the hostname (FQDN) of the time server.

**Timezone**

**>> Next**

▶ En la siguiente pantalla después de haber pulsado *Next* no hay que hacer nada ya que no nos afecta al ver que ya está configurado el uso como DHCP en la red WAN. Para las últimas dos configuraciones en el pie de página sobre “RFC1918 Networks” y “Block bogon networks” las desmarcaremos puesto que son útiles cuando el UTM se conecte directamente a internet, el cual no es el caso.

En la siguiente ventana, configuraremos la red LAN:

- IP: 192.168.100.1
- Mascata de subred: 24

### Configure LAN Interface

On this screen the Local Area Network information will be configured.

LAN IP Address

192.168.100.1

Type dhcp if this interface uses DHCP to obtain its IP address.

Subnet Mask

24

En el siguiente paso, configuraremos la password para la gestion del sistema de pfsense.

- abc123..

► Una vez pulsado siguiente, el configurador nos pedirá recargar el sistema, confirmamos y probaremos si la IP del equipo Kali, que pertenece a la red, ha cambiado a la configuración:

```
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast stat
e DOWN group default qlen 1000
    link/ether 08:00:27:43:73:bc brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.10/24 brd 192.168.100.255 scope global dynamic noprefi
xroute eth0
        valid_lft 7147sec preferred_lft 7147sec
    inet6 fe80::a00:27ff:fe43:73bc/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

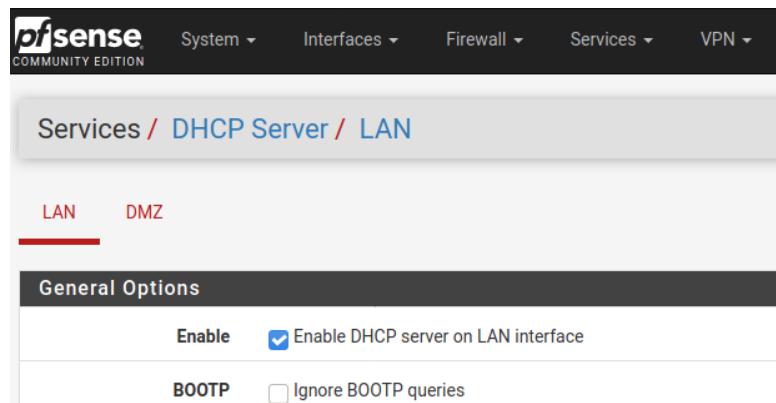
La IP ha cambiado y esta dentro de la red configurada.

► Ahora nuestro UTM habrá modificado su IP a 192.168.100.1, para acceder a la configuración del pfsense accedemos a su IP en el navegador y de nuevo aceptamos los riesgos, como hicimos anteriormente.



Para hacer el login, usaremos la nueva contraseña.

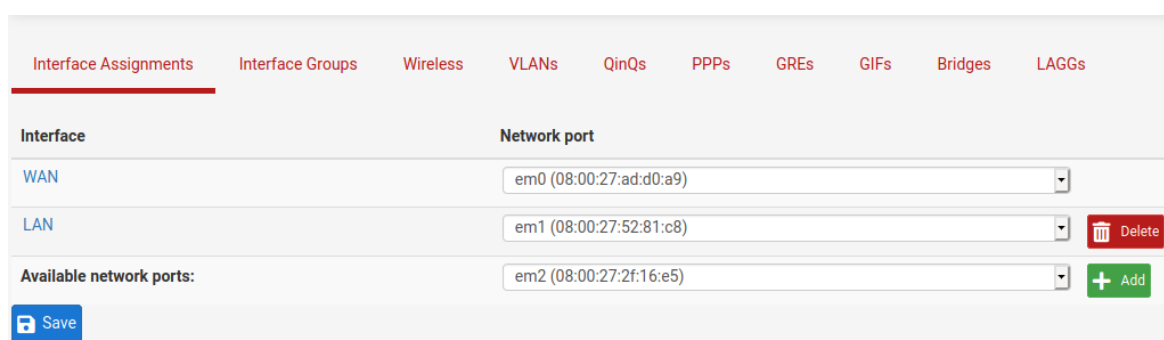
En el apartado Services / DHCP Server, red LAN podemos comprobar como la red dispone del servicio DHCP activado.



## ▼ Red DMZ

Para la configuración de la interfaz de la red DMZ, realizaremos los siguientes pasos:

Una vez en la interfaz web del pfSense y nos dirigiremos a Interfaces / Assignments y en la pantalla agregamos mediante el boton Add la interfaz sin usar.



Red disponible "em2"

Ahora pulsamos sobre la nueva interfaz creada y modificamos su Description a DMZ. También configuramos los siguientes puntos:

- ✓ Enable ► activado

- IPv4 Configuration Type ► Static IPv4
- IPv6 Configuration Type ► None, desactivado puesto que no la usamos.

**General Configuration**

Enable ☒ Enable interface

---

Description

DMZ

Enter a description (name) for the interface here.

---

IPv4 Configuration Type

Static IPv4

---

IPv6 Configuration Type

None

Para la IP fija, la fijamos como 192.168.200.1 y la mascara de red /24. La puerta de enlace no es necesaria puesto que no la usaremos.

**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".  
 Selecting an upstream gateway causes the firewall to treat this interface as a [WAN type interface](#).  
 Gateways can be managed by [clicking here](#).

Guardamos la configuracion en el botón Save y aplicamos los cambios.

A continuación para la configuración de la red **DMZ**, nos dirigimos a:

**Services ► DHCP Server ► DMZ**

En la opciones, marcaremos la siguiente configuración:

- ☒ Enable ► DHCP server
- Deny unknown clients ► All clients\*

El resto de configuraciones relacionadas con ignorar clientes carecen de relacion con el tipo de red que estamos configurando.

\*Puesto que es una red donde se permite el acceso sin restricción por su naturaleza, permitiremos todos los clientes "Allow all clients".

General Options	
Enable	<input checked="" type="checkbox"/> Enable DHCP server on DMZ interface
BOOTP	<input type="checkbox"/> Ignore BOOTP queries
Deny unknown clients	<div> <input type="text" value="Allow all clients"/> </div> <p>When set to <b>Allow all clients</b>, any DHCP client will get an IP address within this scope/range on this interface. If set to <b>Allow known clients from any interface</b>, any DHCP client with a MAC address listed on <i>any</i> scope(s)/interface(s) will get an IP address. If set to <b>Allow known clients from only this interface</b>, only MAC addresses listed below (i.e. for this interface) will get an IP address within this scope/range.</p>
Ignore denied clients	<input type="checkbox"/> 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	<input type="checkbox"/> 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	<div> <input type="text" value="192.168.200.100"/> <input type="text" value="192.168.200.200"/> </div> <div>           From           To         </div>

- Tips a tener en cuenta:
  - En la parte *Additional Pools* es interesante para crear diferentes rangos de IPs.
  - Los servidores WINS no son interesantes puesto que usaremos Kali y no Windows.

En la configuración de los servidores DNS de la red\*:

- Server1 ► 192.168.200.1\*\*
- Server2 ► 1.1.1.1 (Cloudfare)
- Server3 ► 8.8.8.8 (Google)

\*Si se dejara vacío, utilizaría los del servidor.

\*\*Introducimos el del UTM que también actúa como servidor DNS.



**Servers**

WINS servers

WINS Server 1

WINS Server 2

DNS servers

192.168.200.100

1.1.1.1

8.8.8.8

Para configurar la salida a Internet desde la red,

- Gateway/Puerta de enlace ► 192.168.200.1
- Static ARP ► No tendremos tablas estaticas ARP\* (desmarcado)

\* Las tablas estaticas ARP son registros pre-configurados sobre las MACs y las IPs de los dispositivos de la red con objetivo de dificultar un ataque MtM.

**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.

Static ARP










☐ Enable Static ARP entries

This option persists even if DHCP server is disabled. Only the machines listed below will be able to communicate with the firewall on this interface.

Para continuar, guardaremos con el botón Save.

## Checkpoint 1

▶ Hasta aquí, las interfaces creadas y configuradas son las siguientes:

Interfaces   			
 WAN		1000baseT <full-duplex>	192.168.1.242
 LAN		1000baseT <full-duplex>	192.168.100.1
 DMZ		1000baseT <full-duplex>	192.168.200.1

Evaluación de interfaces creadas,

## ▼ Acceso a la red LAN

En este punto vamos a configurar el **UTM** de forma que se pueda acceder desde Internet a un servicio en nuestra red **LAN**, en nuestra práctica se realizara desde el host. Para llevar a cabo necesitamos levantar un servidor Web como es Apache para poder comprobar que funciona.

Abrimos una Terminal, tecleamos, ejecutamos y probamos que esta funcionando:

```
(kali㉿kali)-[~]  
└─$ sudo service apache2 start
```

↓

```
(kali㉿kali)-[~]  
└─$ curl localhost
```

```
(kali㉿kali)-[~]  
└─$ service apached start  
Failed to start apached.service: Unit apached.service not found.  
  
(kali㉿kali)-[~]  
└─$ sudo service apache2 start  
[sudo] password for kali:  
  
(kali㉿kali)-[~]  
└─$ curl localhost  
  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w  
3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
  <head>  
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
    <title>Apache2 Debian Default Page: It works</title>  
    <style type="text/css" media="screen">
```

Comandos introducidos y comprobación de funcionamiento.



Levantaremos el servicio **Apache** como forma de poner a prueba nuestra red, es esencial para probar la practica con los servicios **Suricata** y **ELK**.

▶ Con el servicio funcionando abriremos la conexión desde el exterior por el pfSense mediante la creación de una regla NAT.

Para llevarlo a cabo, seguiremos los siguientes menús:

Firewall ► NAT ► Port Forward ► Boton Add para anadir una nueva regla.

Configuraremos las siguientes opciones:

- Interface ► WAN (Internet)
- Address Family ► Ipv4
- Protocol ► TCP
- Destination ► Type ► WAN address
- Destination port range ► 8080 a 8080
- Redirect target IP:
  - Type ► Single host
  - Address ► 192.168.100.10 (IP del servidor)
- Redirect target port ► Other / 80 (80 es el puerto donde se ejecuta Apache)
- Description ► Servidor Web

Guardamos y aplicamos los cambios.

\*El destino de la regla sera el servidor Apache, el equipo Kali que tiene el servicio.

- Tip a tener en cuenta:
  - **Source** ► El origen sera abierto, en caso de especificarse añadiremos seguridad.

**Edit Redirect Entry**

Disabled

☐ Disable this rule

No RDR (NOT)

☐ 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

☐ Invert match.

WAN address

Type

Address/mask

/

Destination port range

Other

From port

8080

Custom

Other

To port

8080

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

Single host

Type

192.168.100.10

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

Other

Port

80

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.

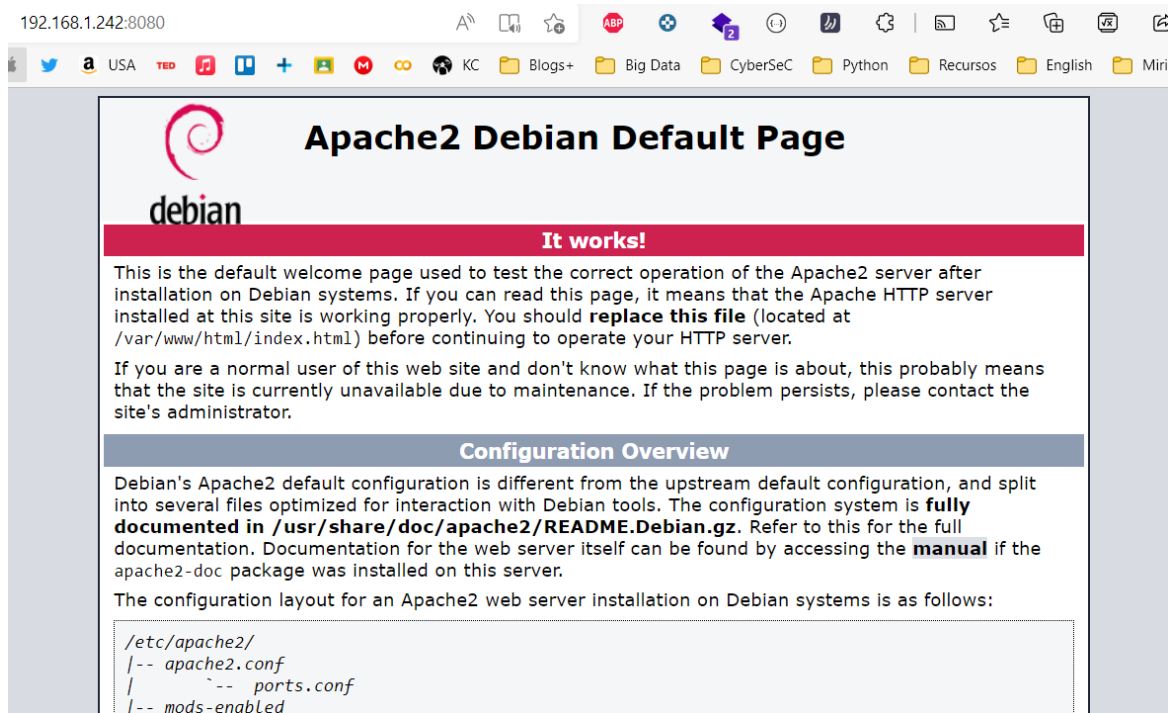
Description

Servidor Web

A description may be entered here for administrative reference (not parsed).

Configuración de acceso al servidor Apache desde internet con redirección a la IP del servidor al puerto del servidor Apache.

Para comprobar que la regla funciona, hacemos una consulta desde el navegador del host hacia la IP del **pfsense** al puerto **8080**, la cual direccionara a la IP del **Kali** servidor al puerto **80**.



Podemos acceder al servidor, la regla NAT funciona.



En nuestro caso no hemos activado la **VPN**. Suponemos que haremos la gestión siempre desde el equipo físicamente y no remotamente.

## Checkpoint 2

Hemos habilitado el acceso desde el exterior de la red a nuestro servidor, de esta forma nos aseguramos que se pueda acceder y crear logs que posteriormente desde Suricata enviaremos a ELK.

### ▼ Suricata

▶ Para la instalación de *Suricata*, en el equipo *Kali* de la red *LAN*, llevaremos a cabo lo siguiente:

En la terminal:

```
(root@kali)-[/home/kali]  
└─# sudo -su  
  
(root@kali)-[/home/kali]  
└─# apt install suricata
```

```

(kali㉿kali)-[~]
$ sudo su
[sudo] password for kali:
(kali㉿kali)-[/home/kali]
# apt install suricata
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
criu fastjar gnome-desktop3-data jarwrapper libaom0 libcbor0 libcodec2-0.9 libdap27
libdapclient6v5 libdav1d4 libepsilon1 libfluidsynth2 libgdal28
libgdk-pixbuf-xlib-2.0-0 libgdk-pixbuf2.0-0 libgeos-3.9.1 libgnome-desktop-3-19
libgupnp-1.2-0 libidn11 libintl-perl libintl-xs-perl libmodule-find-perl
libmodule-scandeps-perl libnetcdf18 libntfs-3g883 libomp-11-dev libomp5-11
libproc-processtable-perl libproj19 libsort-naturally-perl liburcu6 liburing1
libwireshark14 libwiretap11 libwsutil12 libx265-192 libxkbregistry0 libyara4
maltego needrestart python-is-python2 python3-editor python3-exif
python3-ipython-genutils python3-orjson python3-pylnk python3-stem starkiller tini
zaproxy
Use 'sudo apt autoremove' to remove them.

```

Instalación de Suricata.

► Una vez instalado Suricata, vamos a configurar su fichero de configuración, pero antes deberemos conocer que interfaz de red estamos usando.

Comprobamos cual es la interfaz en uso en el equipo para que conocer en cual de ellas trabaja ra Suricata, en nuestro caso eth0:

```

(kali㉿kali)-[/var/log/suricata]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group d
efault qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP group default qlen 1000
    link/ether 08:00:27:43:73:bc brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.10/24 brd 192.168.100.255 scope global dynamic noprefi
xroute eth0
        valid_lft 5828sec preferred_lft 5828sec
    inet6 fe80::a00:27ff:fe43:73bc/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue stat
e DOWN group default
    link/ether 02:42:07:7b:b6:5f 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

```

Verificación de interfaz de red en uso.

Accedemos a la ubicacion del fichero de configuracion

```
└─(kali@kali)-[~]  
└─$ cd /etc/suricata
```

Editamos el fichero:

```
└─(kali@kali)-[~]  
└─$ nano suricata.yaml
```

Buscamos "interface" y comprobamos que esta configurado para el uso de la interfaz que estamos usando.

```
# Linux high speed capture support  
af-packet:  
- interface: eth0  
  # Number of receive threads. "auto" uses the number of cores  
  #threads: auto  
  # Default clusterid. AF_PACKET will load balance packets based on flow.  
  cluster-id: 99  
  # Default AF_PACKET cluster type. AF_PACKET can load balance per flow or per hash.  
  # This is only supported for Linux kernel > 3.1  
  # possible value are:  
  # * cluster_flow: all packets of a given flow are sent to the same socket  
  # * cluster_cpu: all packets treated in kernel by a CPU are sent to the same sock>  
  # * cluster_qm: all packets linked by network card to a RSS queue are sent to the>  
  # socket. Requires at least Linux 3.14.  
  # * cluster_ebpf: eBPF file load balancing. See doc/userguide/capture-hardware/eb>  
  # more info.  
  # Recommended modes are cluster_flow on most boxes and cluster_cpu or cluster_qm o>  
  # with capture card using RSS (requires cpu affinity tuning and system IRQ tuning)  
  cluster-type: cluster_flow  
  # In some fragmentation cases, the hash can not be computed. If "defrag" is set
```

Comprobación de interfaz seleccionada para Suricata.

Una vez comprobado, guardamos y salimos.

▶ Ejecutamos Suricata.

```
└─(root@kali)-[/]  
└─# service suricata start
```

```

(kali@kali)~$ # service suricata start
(kali@kali)~$ # service suricata status
● suricata.service - Suricata IDS/IDP daemon
   Loaded: loaded (/lib/systemd/system/suricata.service; disabled; vendor preset: di>
   Active: active (running) since Sun 2022-03-20 13:59:42 EDT; 2h 50min ago
     Docs: man:suricata(8)
           man:suricatasc(8)
           https://suricata-ids.org/docs/
   Main PID: 8481 (Suricata-Main)
    Tasks: 8 (limit: 4616)
   Memory: 53.2M
      CPU: 44min 24.519s
   CGroup: /system.slice/suricata.service
           └─8481 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml -->

Mar 20 13:59:42 kali systemd[1]: Starting Suricata IDS/IDP daemon...
Mar 20 13:59:42 kali suricata[8480]: 20/3/2022 -- 13:59:42 - <Notice> - This is Surica>
Mar 20 13:59:42 kali systemd[1]: Started Suricata IDS/IDP daemon.
lines 1-16/16 (END)

```

## ▼ ELK

El conjunto ELK funcionara en la red **DMZ** sobre la maquina Kali en ella, en nuestro caso para usar ELK lo haremos sobre Docker.

Obtenemos la imagen de ELK.

```

(kali@kali)~$ git clone https://github.com/deviantony/docker-elk

```

```

(kali@kali)~$ git clone https://github.com/deviantony/docker-elk
Cloning into 'docker-elk' ...
remote: Enumerating objects: 2013, done.
remote: Total 2013 (delta 0), reused 0 (delta 0), pack-reused 2013
Receiving objects: 100% (2013/2013), 516.39 KiB | 2.95 MiB/s, done.
Resolving deltas: 100% (851/851), done.

```

Descarga del paquete.

Para ejecutar, desde el directorio donde hemos descargado ELK, ejecutamos lo siguiente, si pi de instalarlo docker-compose confirmamos

```

(kali@kali)~/docker-elk$ docker-compose up

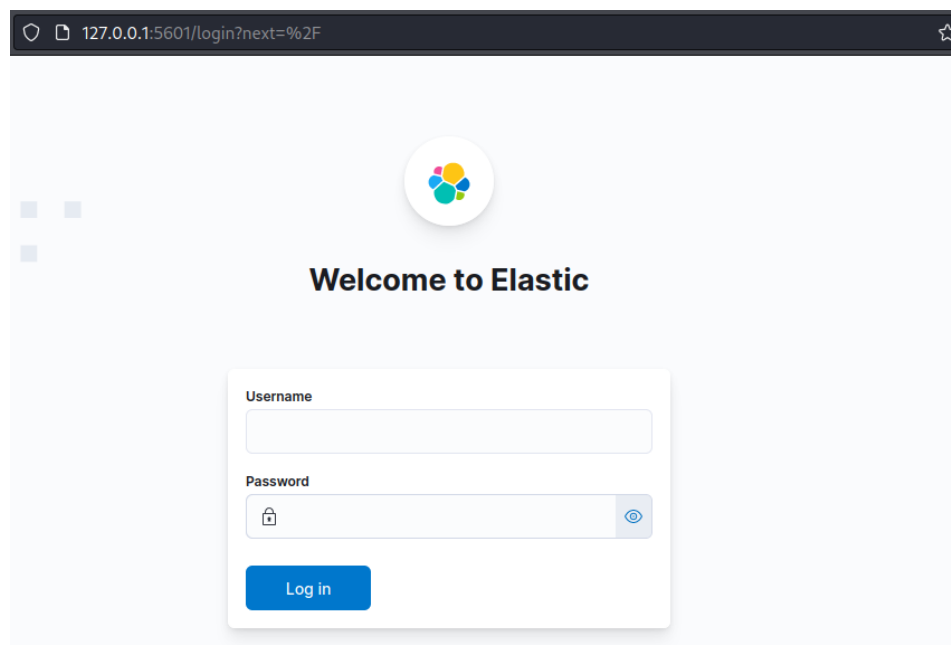
```

Una vez que se ejecuta, nos dirigimos a 127.0.0.1:5603



```
(root@kali)-[/home/kali/practica/docker-elk]
# docker-compose up
Creating network "docker-elk_elk" with driver "bridge"
Creating volume "docker-elk_setup" with default driver
Creating volume "docker-elk_elasticsearch" with default driver
Building setup
Sending build context to Docker daemon 11.78kB
Step 1/7 : ARG ELASTIC_VERSION
Step 2/7 : FROM docker.elastic.co/elasticsearch/elasticsearch:${ELASTIC_VERSION}
8.1.0: Pulling from elasticsearch/elasticsearch
4fb807caa40a: Pull complete
d544aeb95ae6: Pull complete
04cdca37d656: Pull complete
cc7a2b0f4dc3: Downloading 179.8MB/574.4MB
10afa3043f5e: Download complete
66c9eb06a8e6: Download complete
cc7a2b0f4dc3: Pull complete
10afa3043f5e: Pull complete
66c9eb06a8e6: Pull complete
ca7e1235f0fe: Pull complete
fc6892625d9a: Pull complete
9f1937e0ac61: Pull complete
9bc99e11027b: Pull complete
Digest: sha256:b58e10906c1858bfc1c113fddb64c602f8cb3e00b159fdaa940f9afc3daf80a7
Status: Downloaded newer image for docker.elastic.co/elasticsearch/elasticsearch:8.1.0
   -> b9fc6ca9dc52
Step 3/7 : USER root
   -> Running in 90e705626484
```

Ejecución del entorno ELK.



Prueba de ELK.

Credenciales  
User: elastic

Passwd: changeme

▶ A continuación necesitamos integrar ELK con Suricata.

En el menu, Managements ► Integrations buscamos Suricata Events.y agregamos la integracion c onfigurandola con los siguientes datos:

The screenshot shows the 'Configure integration' page in the Elastic Stack. The breadcrumb navigation is 'Integrations > Suricata Events > Add integration'. The page title is '1 Configure integration'. Under 'Integration settings', there is a text input for 'Integration name' with the value 'suricata-Kali1' and a text input for 'Description' with the value 'Eventos de suricata en kali'. Below these is a link for 'Advanced options'. A section titled 'Collect Suricata eve logs (input: logfile)' is expanded, showing a toggle for 'Suricata eve logs (log)' which is checked, with the description 'Collect Suricata eve logs using log input'. To the right is a 'Paths' section with a text input containing '/var/log/suricata/eve.json' and an 'Add row' link. At the bottom is a 'Preserve original event' section with a radio button selected for 'X' (checked), with the description 'Preserves a raw copy of the original event, added to the field event.original' and an 'Advanced options' link.

Al anadir confirmar nos pedira de agregar ahora o despues el agente, añadimos ahora. Y en la pestana seleccionamos Run standalone.

## Add agent



Add Elastic Agents to your hosts to collect data and send it to the Elastic Stack.

Enroll in Fleet [Run standalone](#)

Run an Elastic Agent standalone to configure and update the agent manually on the host where the agent is installed.

### 1 Download the Elastic Agent to your host

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 download the Elastic Agent binaries and verification signatures from Elastic's download page.

Linux users: We recommend the installer (TAR) over system packages (RPM/DEB) because it lets you upgrade your agent in Fleet.

[Go to download page](#)

Añadir agente.



Cambiamos a la máquina Kali de la red **LAN**.

Copiamos la dirección del link "Go to download page" y pinchamos para descargar Linux Agent 8.1.0 y descomprimos el archivo.



elastic [Products](#) [Customers](#) [Learn](#) [Company](#) [Pricing](#)

[Contact](#) [Login](#)

## Elastic Agent 8.1.0

[LINUX 64-BIT](#) [sha](#)

[LINUX AARCH64](#) [sha](#)

[DEB 64-BIT](#) [sha](#)

[DEB AARCH64](#) [sha](#)

[RPM 64-BIT](#) [sha](#)

[RPM AARCH64](#) [sha](#)

[WINDOWS 64-BIT](#) [sha](#)

[MAC](#) [sha](#)

Descarga de agente.



Cambiamos a la máquina Kali de la red **DMZ**.

En la máquina donde se está ejecutando Elastic, descargamos el agente en el botón "Download Policy".

2

## Configure the agent

Copy this policy to the `elastic-agent.yml` on the host where the Elastic Agent is installed. Modify `ES_USERNAME` and `ES_PASSWORD` in the `outputs` section of `elastic-agent.yml` to use your Elasticsearch credentials.



Copy to clipboard



Download Policy

```
id: b2ee9a90-a87e-11ec-a99f-b7884386a8a1
revision: 2
outputs:
  default:
    type: elasticsearch
    hosts:
```

Descarga de fichero de configuración del agente.

Ahora, teniendo en una máquina el agente descargado de la web de Elastic(LAN) y el fichero de scargado de nuestro Elastic en el equipo DMZ. Necesitamos transferir el archivo de configuración descargado en el Kali DMZ para sustituirlo por el archivo que está en la carpeta descomprimida en la máquina Kali de LAN.

► Es necesario realizarlo, podemos decir que desde la web hemos descargado todos los archivos necesarios y desde Elastic la personalización para nuestro ELK.



Es necesario personalizar el fichero `elastic-agent.yml`

Modificaremos los siguientes datos del fichero `elastic-agent.yml`

```
hosts:
  - 'http://192.168.200.101:9200' ► Direccion del equipo Kali en DMZ con ELK
username: 'elastic' ► Usuario de ELK
password: 'changeme' ► Passwd de ELK
```

```

1 id: b2ee9a90-a87e-11ec-a99f-b7884386a8a1
2 revision: 2
3 outputs:
4   default:
5     type: elasticsearch
6     hosts:
7       - 'http://192.168.200.101:9200'
8     username: 'elastic'
9     password: 'changeme'
10  output_permissions:
11    default:
12      _elastic_agent_monitoring:
13        indices:
14          - names:
15              - logs-elastic_agent.apm_server-default
16            privileges: &ref_0

```

▶ A continuación, vamos a instalar el agente de elastic para **Suricata** en la máquina **Kali** de la red **LAN**.

Para ello, accedemos a la carpeta descargada y como root:

```

—(root@kali)-[/home/kali/Downloads/elastic-agent-8.1.0-linux-x86_64]
└─# ./elastic-agent install

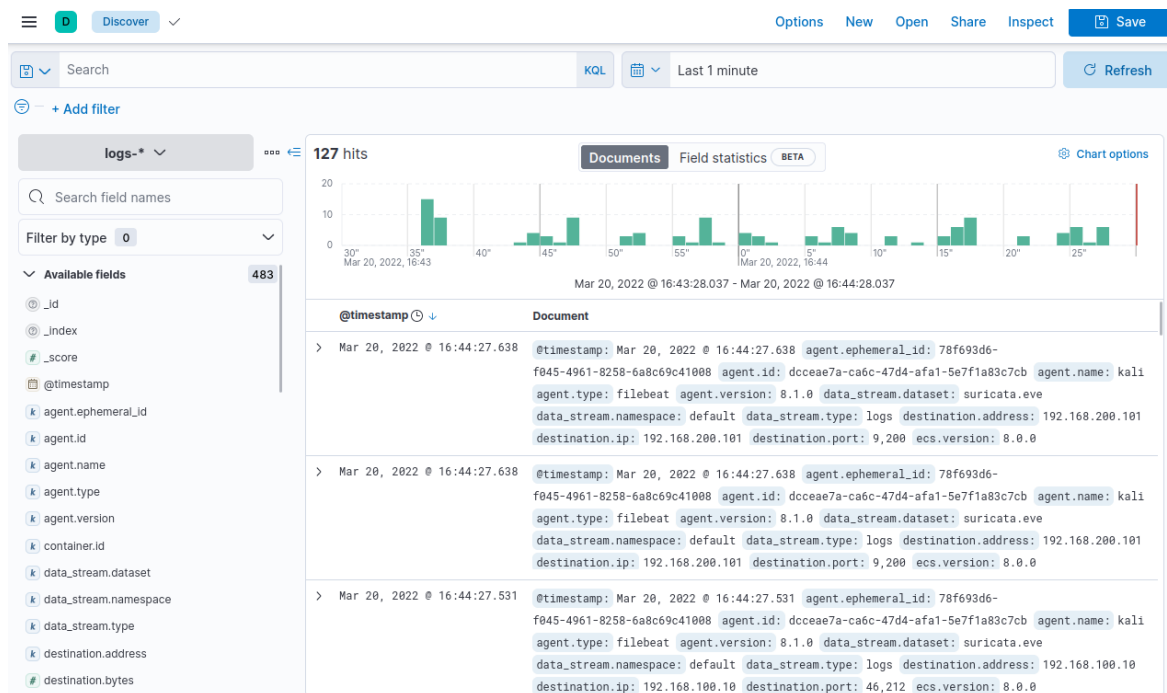
```

```

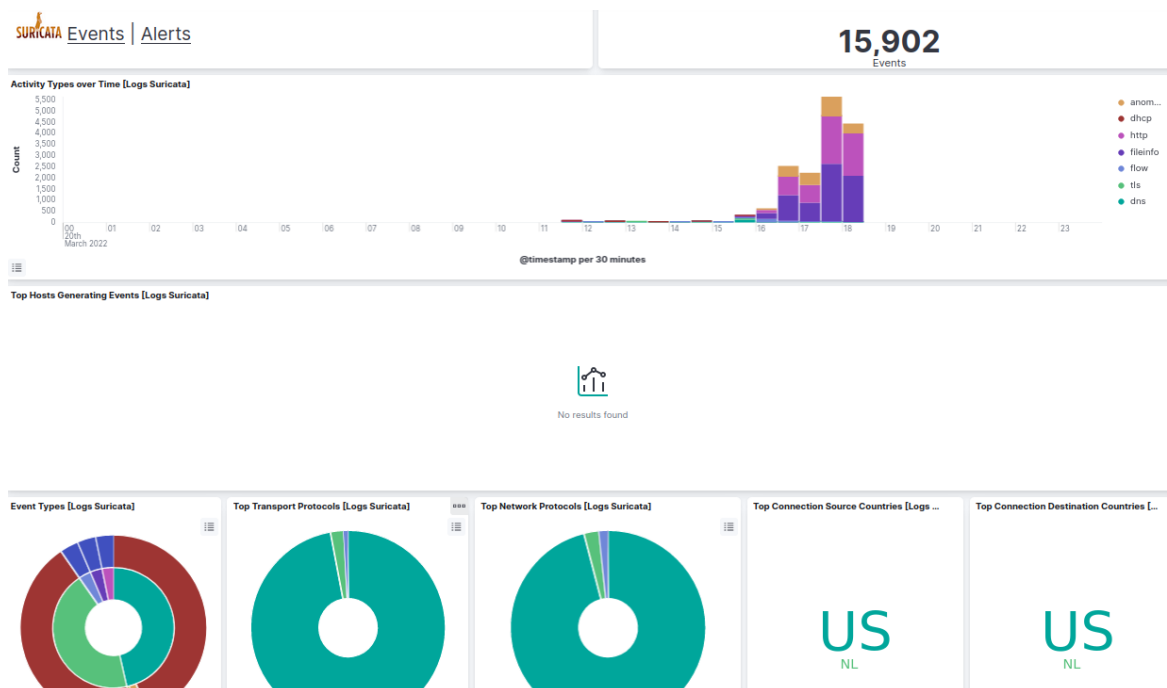
(kali@kali)-[~/Downloads/elastic-agent-8.1.0-linux-x86_64]
└─$ sudo su
[sudo] password for kali:
(kali@kali)-[~/Downloads/elastic-agent-8.1.0-linux-x86_64]
└─# ./elastic-agent install
Elastic Agent will be installed at /opt/Elastic/Agent and will run as a service. Do you
want to continue? [Y/n]:y
Do you want to enroll this Agent into Fleet? [Y/n]:n
Elastic Agent has been successfully installed.

```

▶ Ahora en el *Discover* de Elastic podemos observar los logs enviados por Suricata.



Registro de logs de Suricata en ELK.



Dashboard de logs de Suricata en ELK.

▶ Como observamos en los gráficos superiores, ELK esta recibiendo logs por parte de Suricata a través del pfsense.

## Checkpoint final

He podido enviar los logs de la máquina Kali de la red LAN donde esta Suricata instalado a la máquina Kali de la red DMZ pasando por el pfsense. Quedaría por realizar la parte opcional de añadir un honeypot en una tercera máquina en una tercera red y configurar la salida a internet y el honeypot como tal, ademas del ELK.