BLUE TEAM

DIEGO MARTÍN OLea

2023

***Práctica:***

***La cual debe cumplir los siguientes requisitos:***

***• Creación de un PFsense en bridge que conecte 3 redes, LAN, DMZ y DMZ\_2 estas como red interna.***

***• Un equipo W10 en LAN, un stack ELK en DMZ y un grupo de honeypots en DMZ\_2.***

***• Queremos transmitir los logs de los honeypots al ELK stack, pero los honeypots no deben tener***

***acceso a las otras redes(solo para transmitir logs) pero los honeypots deben ser accesible desde la***

***red WAN.***

***• El servidor ELK debe almacenar y poder visualizar los diferentes logs de los honeypots.***

***• El W10 debe poder conectarse a ELK vía Kibana.***

***ÍNDICE***

1. *Proceso de creación de la infraestructura* **3**

*1.1. Creación de UTM y configuración de interfaces* 3

*1.2.Creación de NAT* 10

1.3. *Creación de OpenVPN* 11

1.4. *Reglas DMZ* 17

1. *Instalación de herramientas* **19**

*2.1. Suricata* 19

*2.2. Elastic Stack* 20

1. *Conexión entre equipos de la red interna 20*

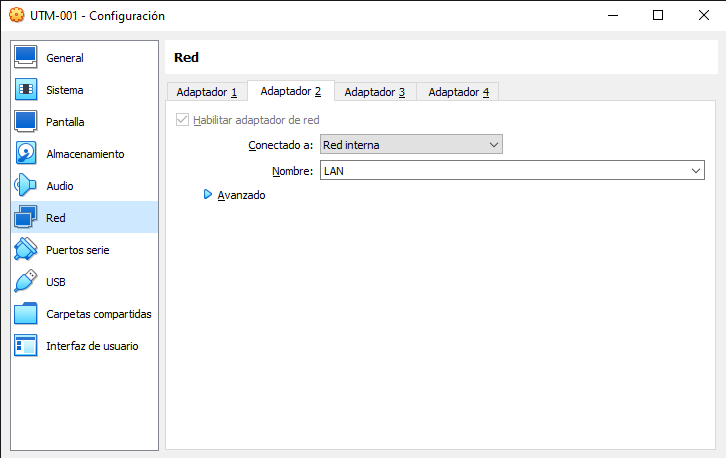
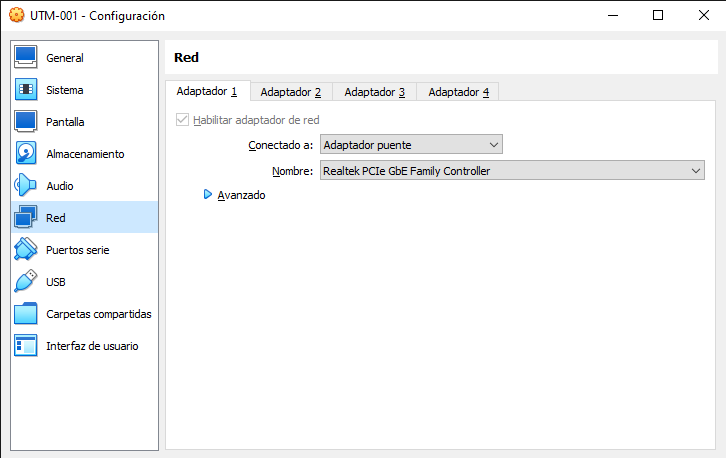
*3.1. Integración de Suricata en Elastic Stack 20*

*3.2. Creación e integración de honeypot en Elastic Stack23*

1. ***Proceso de creación de la infraestructura***

***1.1. Creación de UTM y configuración de interfaces***

En primer lugar descomprimimos el archivo para poder tener la imagen ISO del pfSense en nuestro ordenador para poder montar la máquina virtual. Posteriormente montamos la imagen ISO en VirtualBox,la llamamos UTM-001 y procedemos a la instalación de la máquina virtual. Después de finalizar la instalación configuramos las redes que vamos a usar posteriormente y las llamamos LAN,DMZ,DMZ\_2.



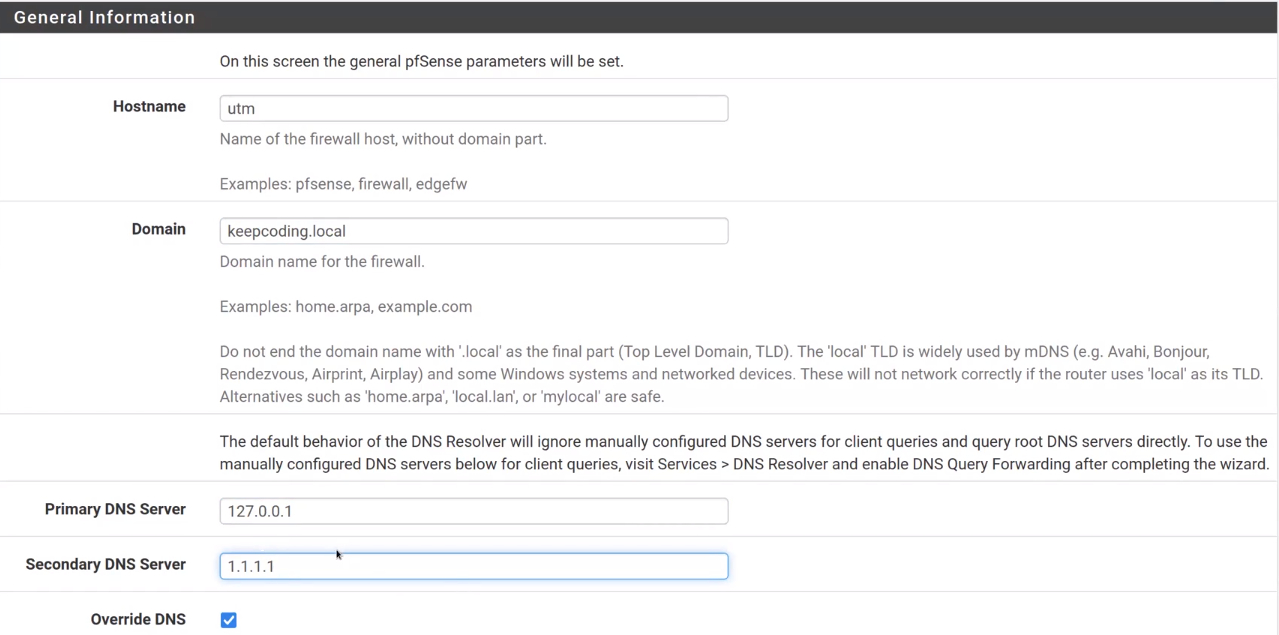
Reiniciamos la máquina y nos saldrán ya las redes configuradas en el UTM.



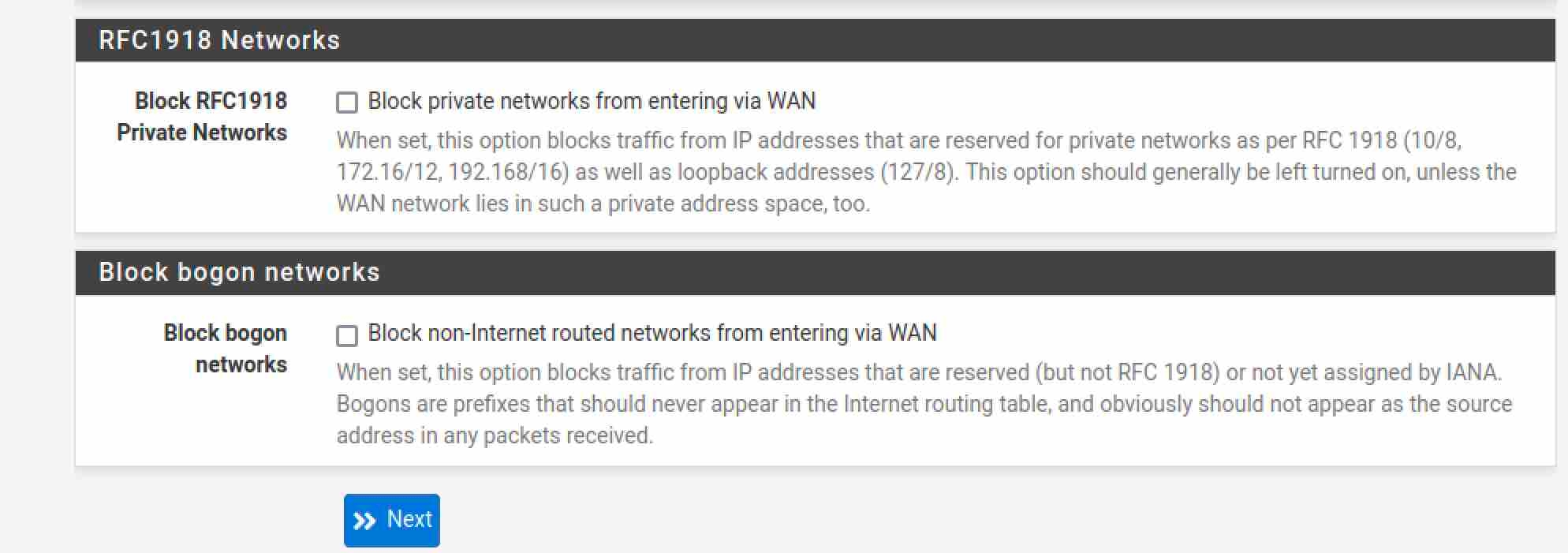
Una vez que hayamos hecho estos pasos procedemos a iniciar nuestra máquina virtual linux donde buscaremos a través de nuestra terminal mediante el comando ifconfig nuestra ip para poder acceder al pfSense. En nuestro caso será la ip 192.168.1.1. Esta ip la usaremos en nuestro navegador y podremos acceder al pfSense donde una vez que nos registremos con un usuario y contraseña procederemos a configurar el pfSense.

Vamos con el navegador a 192.168.1.1 y nos logueamos

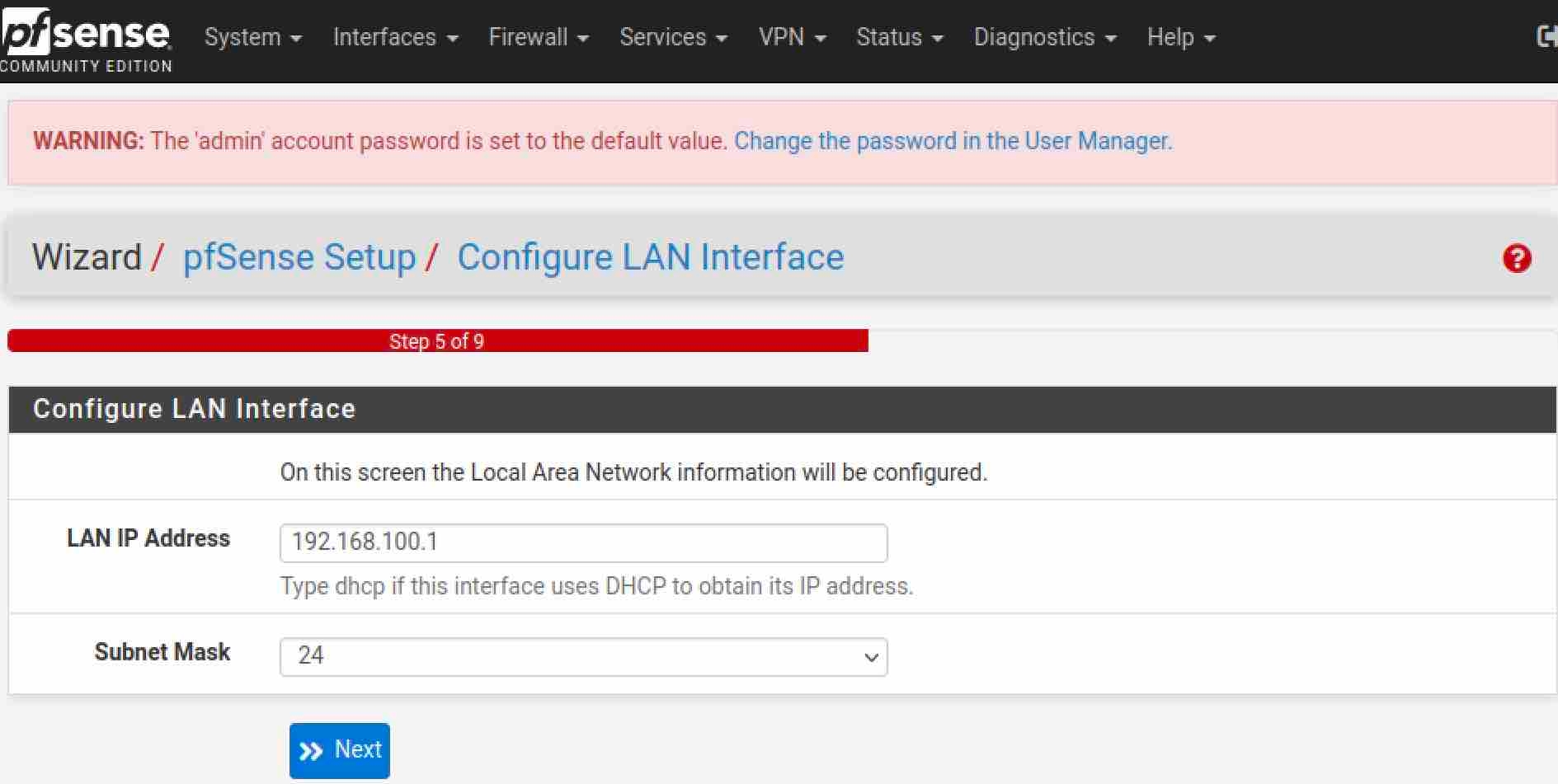
En la pantalla inicial de configuración se introducen estos valores para definir el hostname y el rango de dominio de nuestra red y los DNS de CloudFlare en este caso. En nuestro caso será UTM con el dominio keepcoding.local y añadiremos los DNS.



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 está 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.



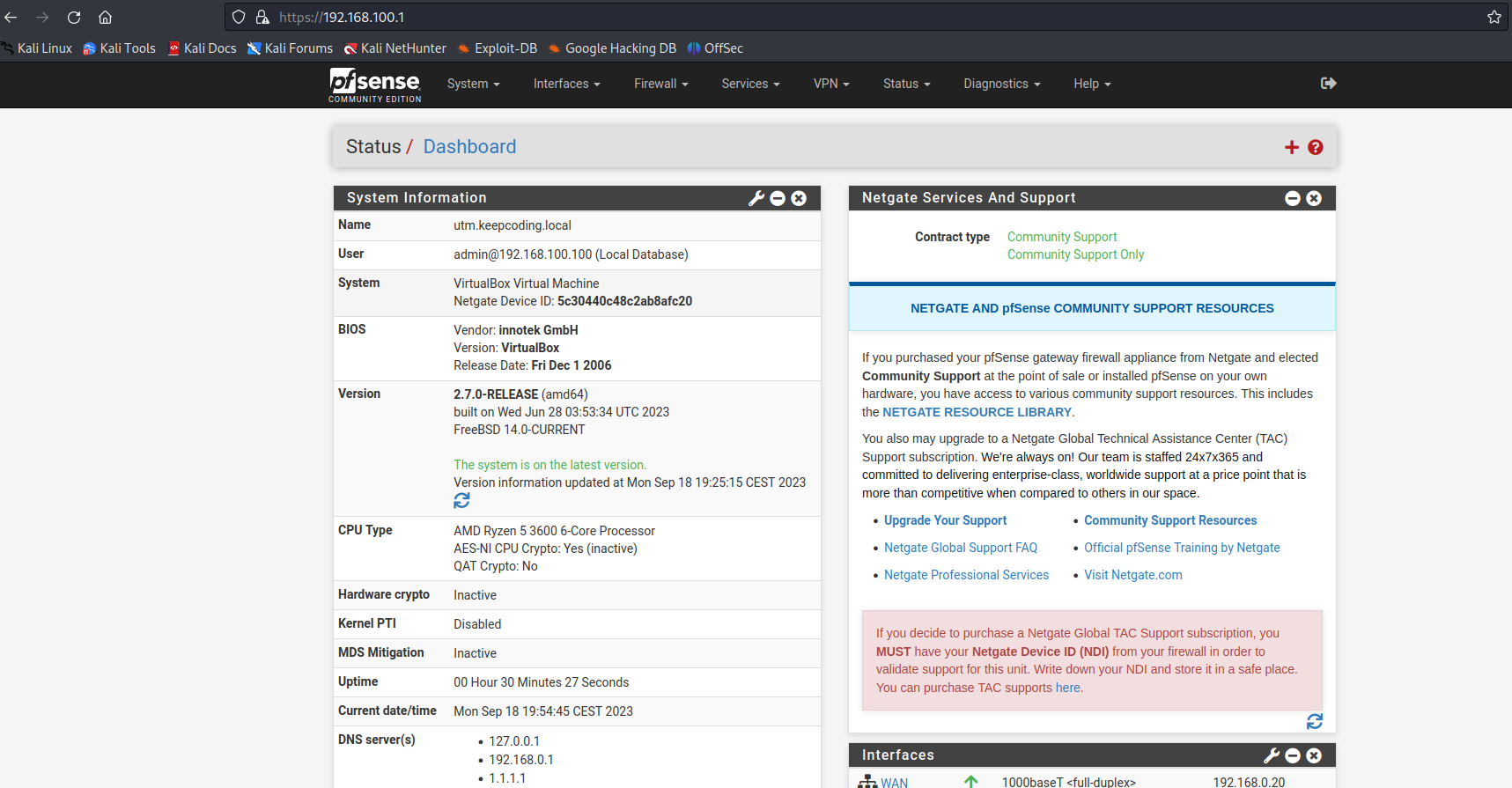
Seguidamente se configura la interfaz LAN, con la IP 192.168.100.1 y la subnet mask /24.



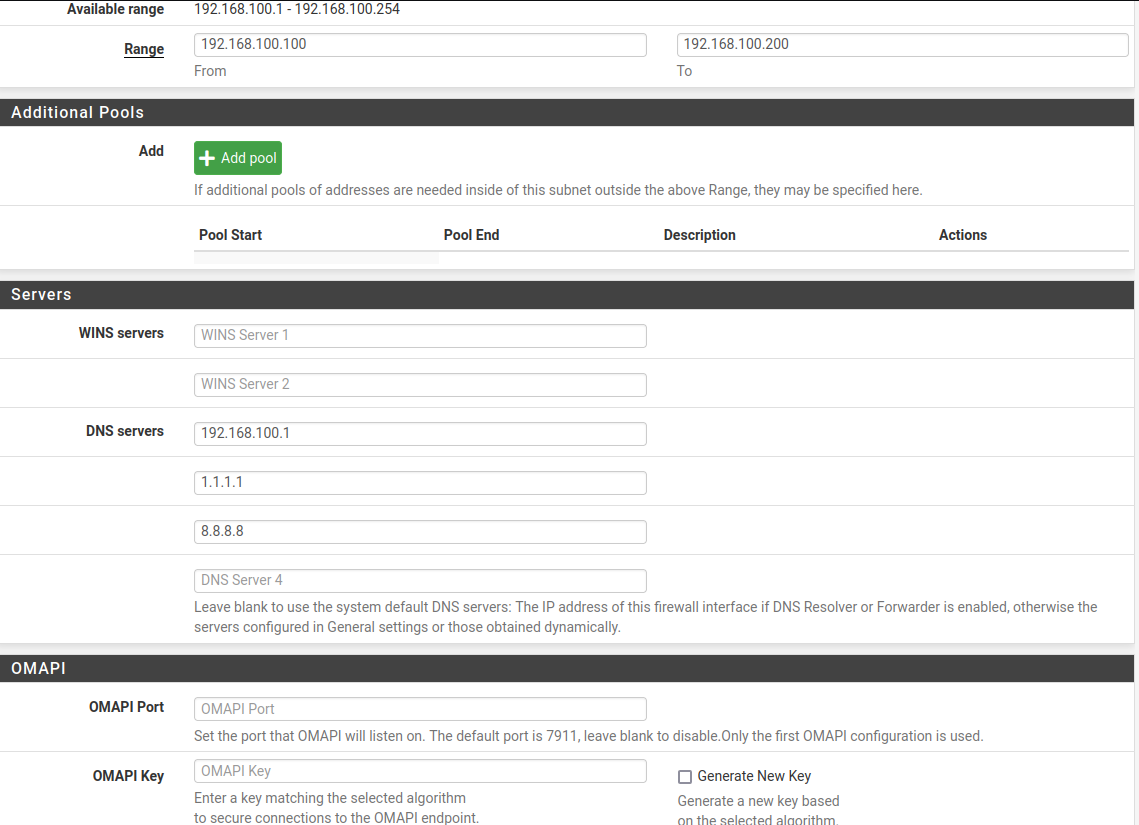
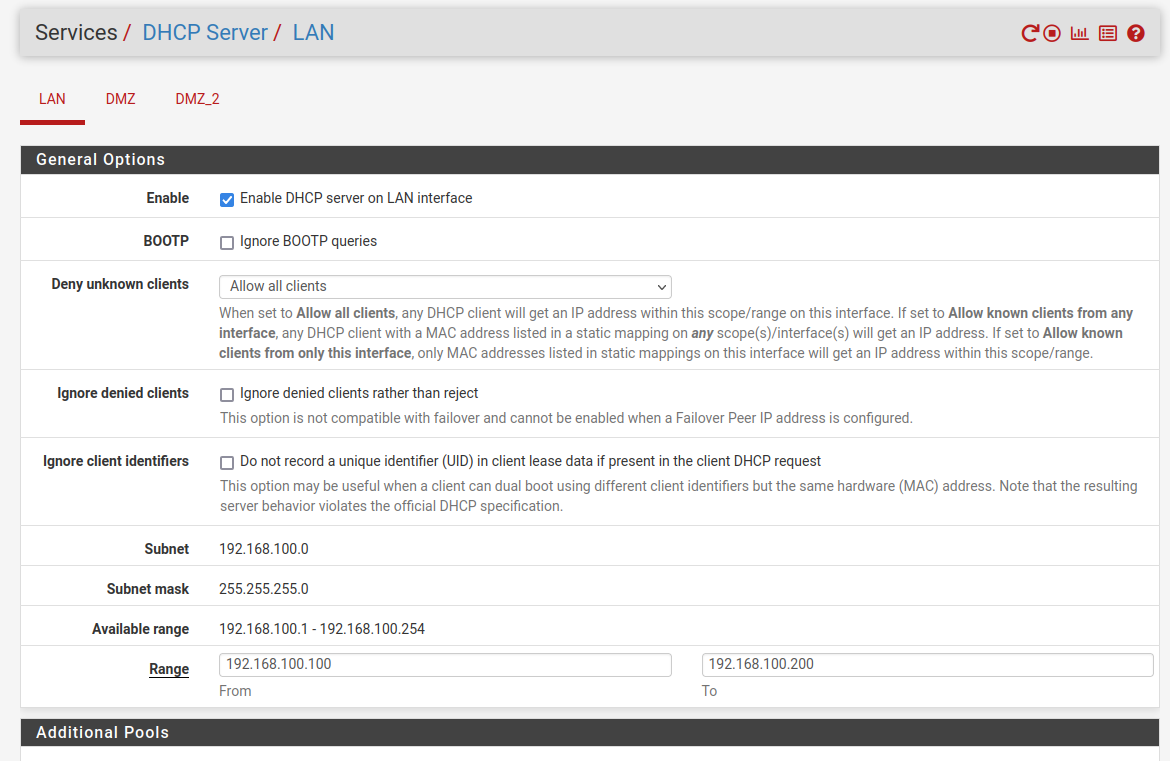
Finalmente se cambia la password por defecto y se recarga la herramienta.

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

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

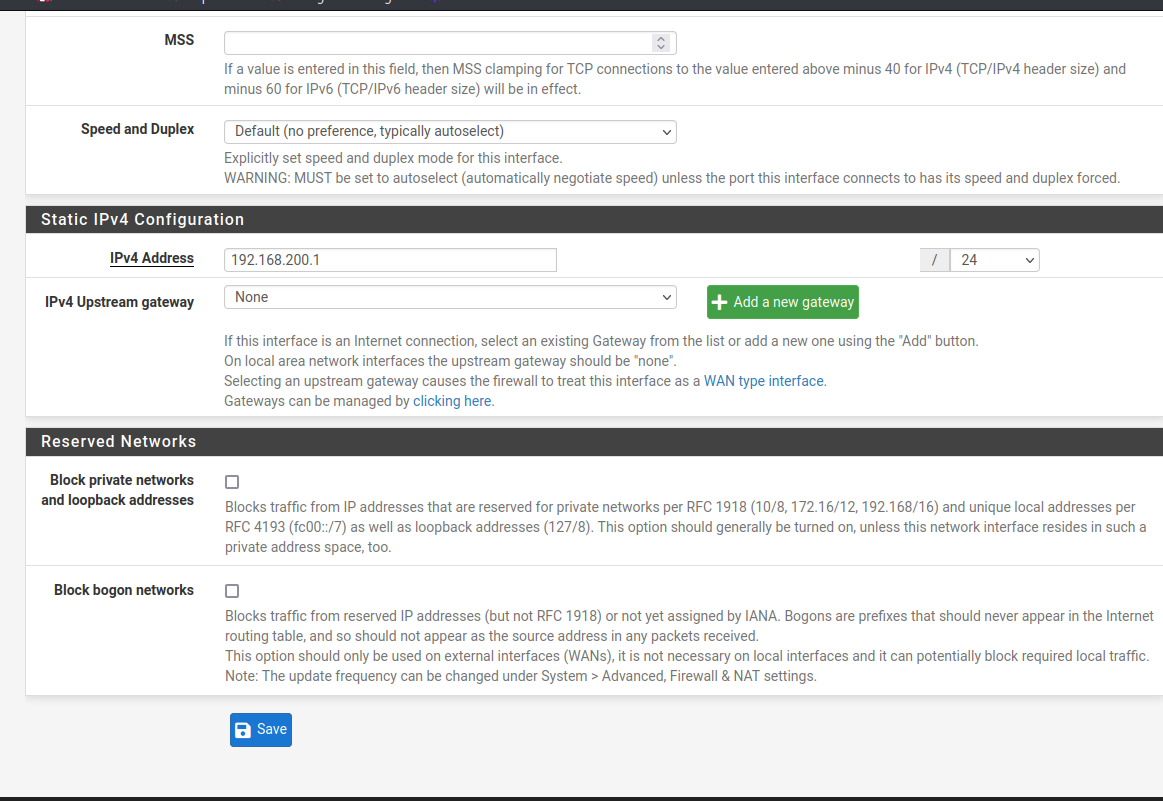
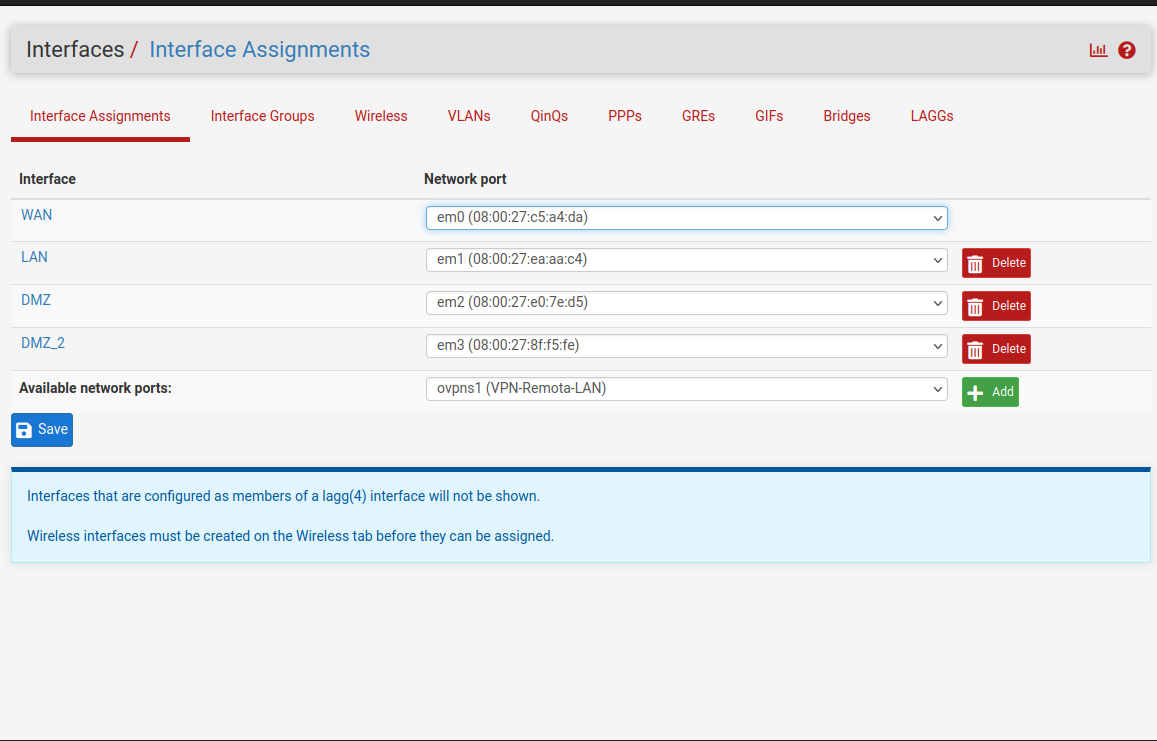


Una vez que nos encontramos en el pfSense vamos a DHCP Server y configuramos el rango de la LAN y lo ponemos de 192.168.100.100 a 192.168.100.200 y ponemos los servidores de DNS.



En gateway pondremos la ip 192.168.100.1 para que nuestro equipo se conecte a internet a través del pfSense.

Posteriormente creamos dos redes que ya configuramos previamente en la máquina virtual que son la DMZ y la DMZ\_2. Poniendo IP estática IPV4 en la configuración. En DMZ 192.168.200.1 en DMZ\_2 192.168.250.1

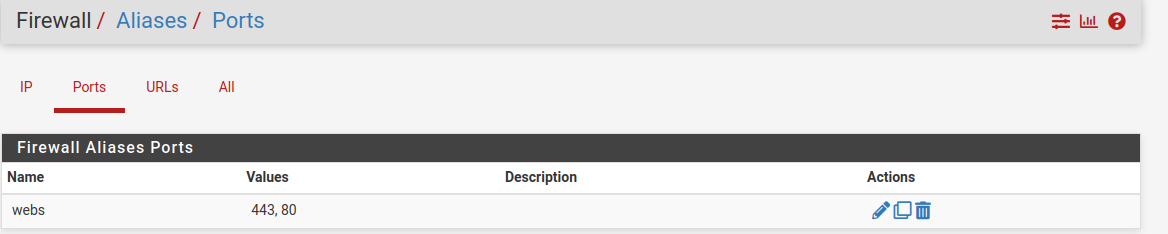


Y podemos comprobar en que ya en el UTM nos salen las 4 redes configuradas.

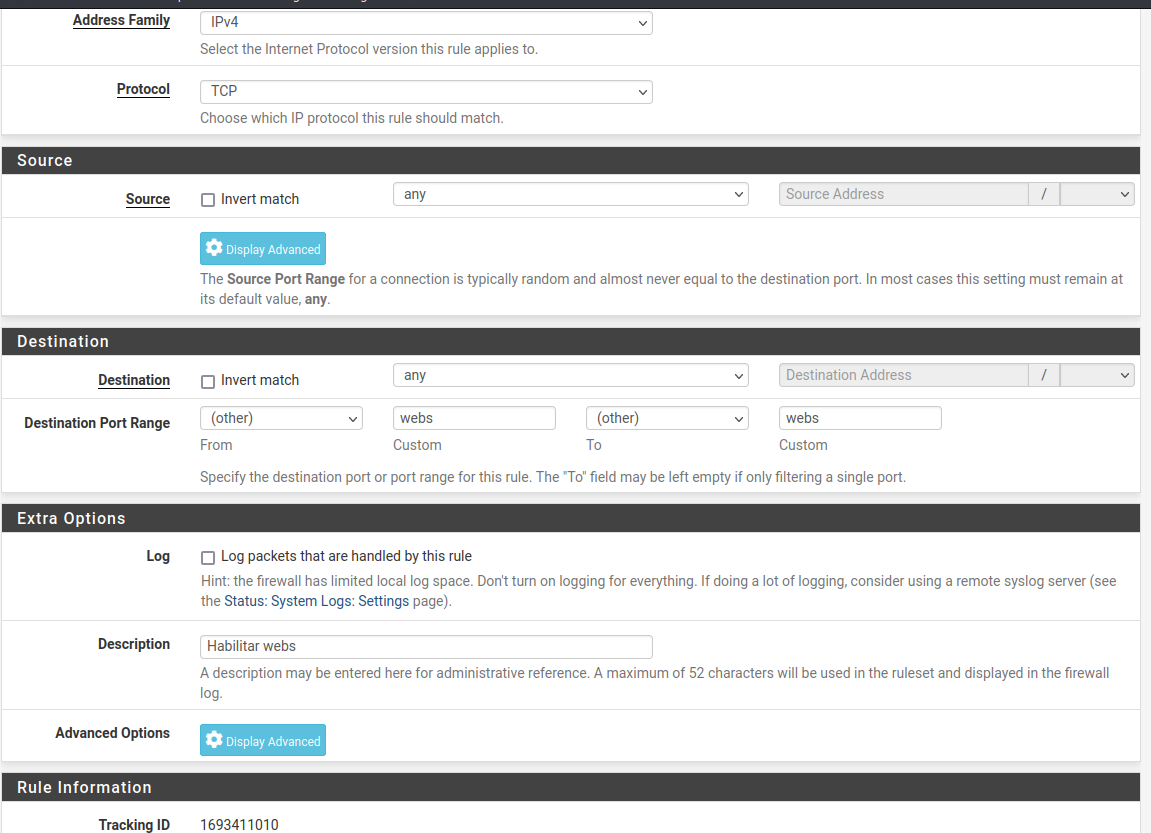


Una vez que tenemos estas 4 redes hacemos lo mismo en DHCP Sserver y configuramos como anteriormente hicimos con la LAN en este caso con ambas DMZ configurando el rango,los servidores DNS y gateway.

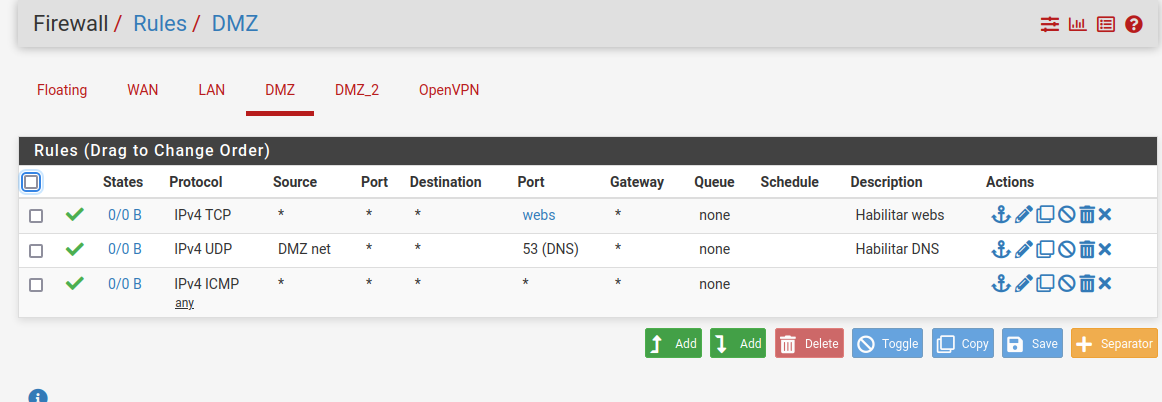
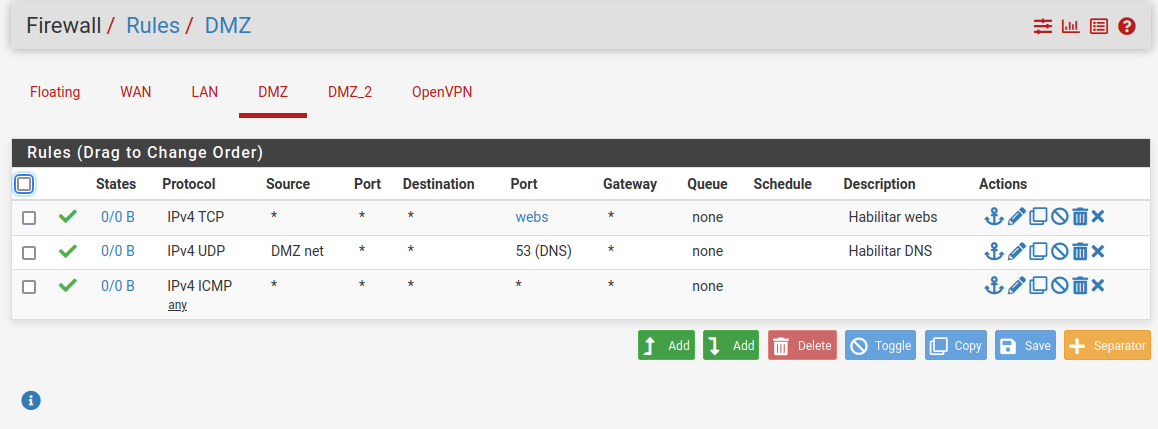
Una vez que tenemos configuradas todas las redes debemos configurar una serie de reglas para establecer la navegación a internet en estas redes. Añadimos los puertos 443 y 80 bajo una regla llamada webs.



En LAN vienen reglas predeterminadas por lo que vamos DMZ para añadir una serie de reglas a nuestra red. Habilitamos las webs previamente guardadas en la anterior regla al configurarlo así



También habilitamos DNS con otra regla y ya tendríamos las reglas de DMZ



* 1. ***Creación de NAT***

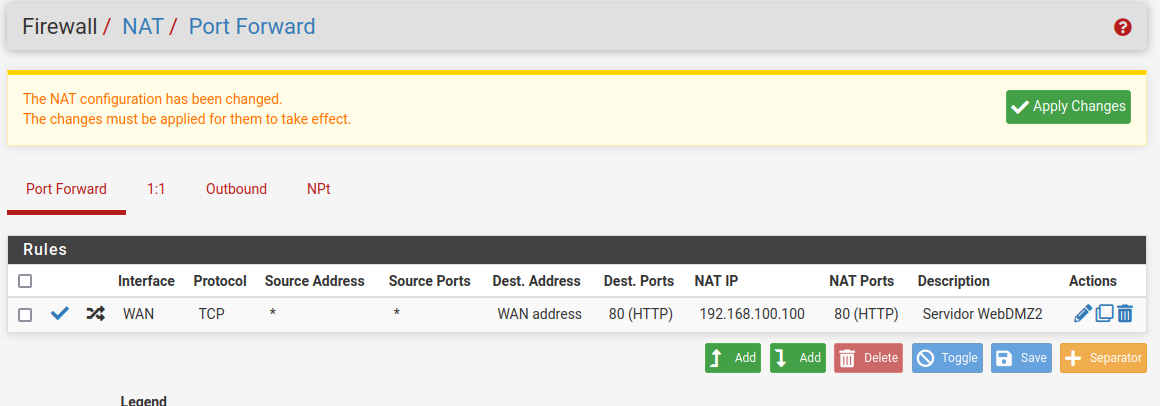
NAT permite acceder a un servicio que está dentro de una red interna, se va a

crear a través de Port forwarding.

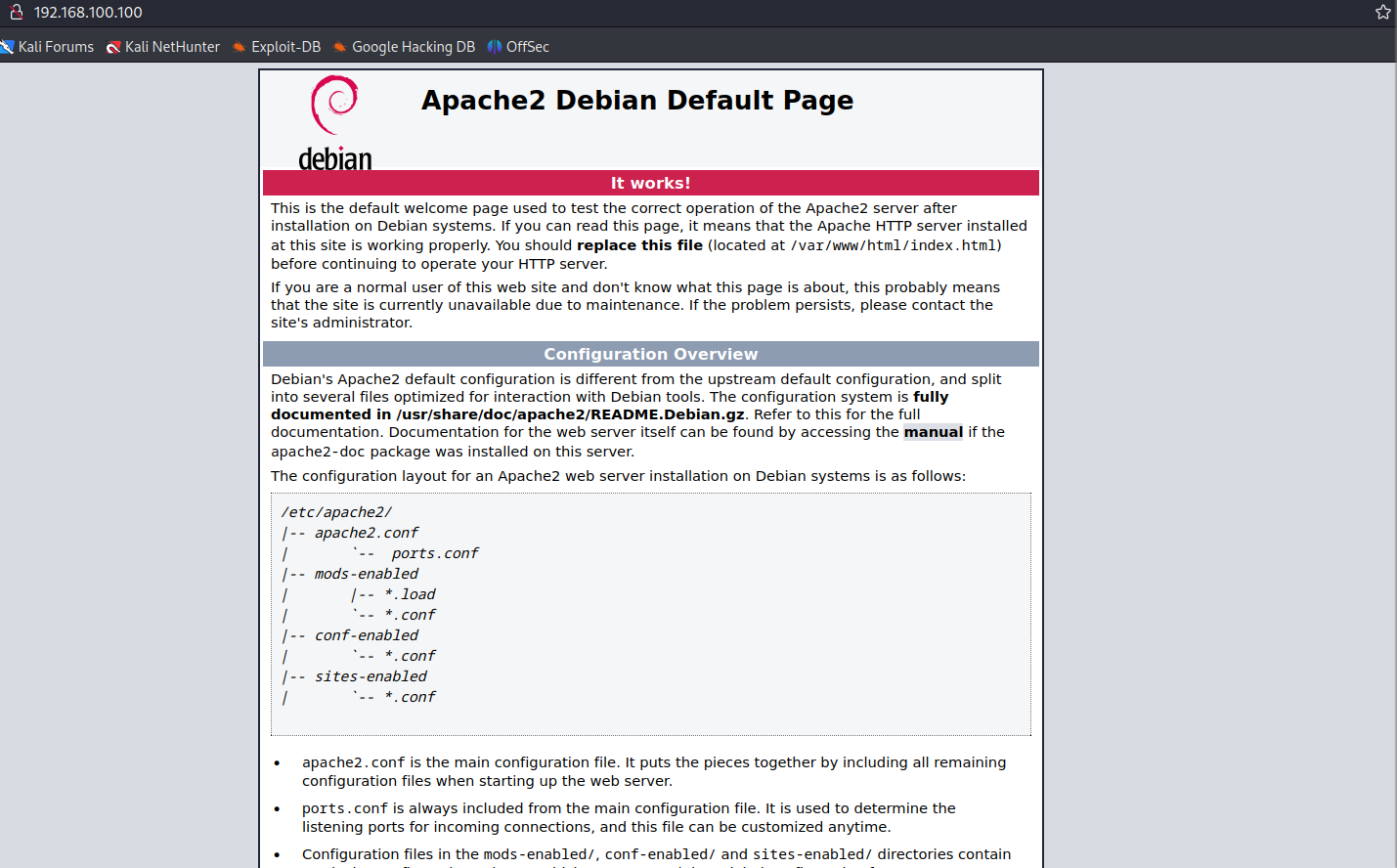
Los pasos a seguir serían:

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 80, 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 por un puerto concreto, en el que está corriendo el servidor web apache2.



Con esto conseguimos que nos reenvié a la página de apache



* 1. ***Creación de OpenVPN***

Vamos a crear una VPN para que las conexiones entrantes y salientes del WAN

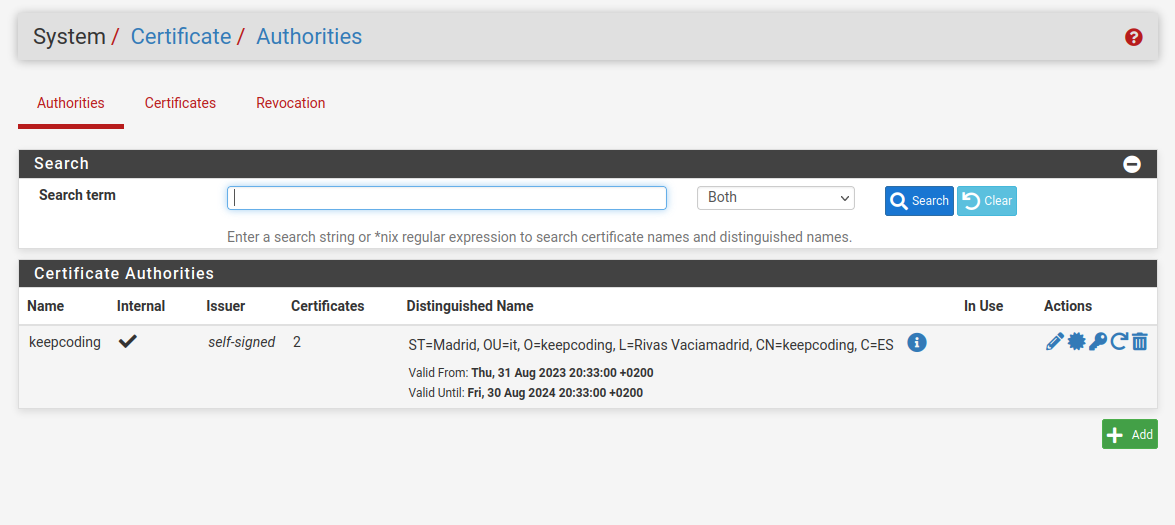
a la red interna y viceversa pasen por el UTM y estén controladas.

En pfSense, entramos en System/Package Manager/Available Packages

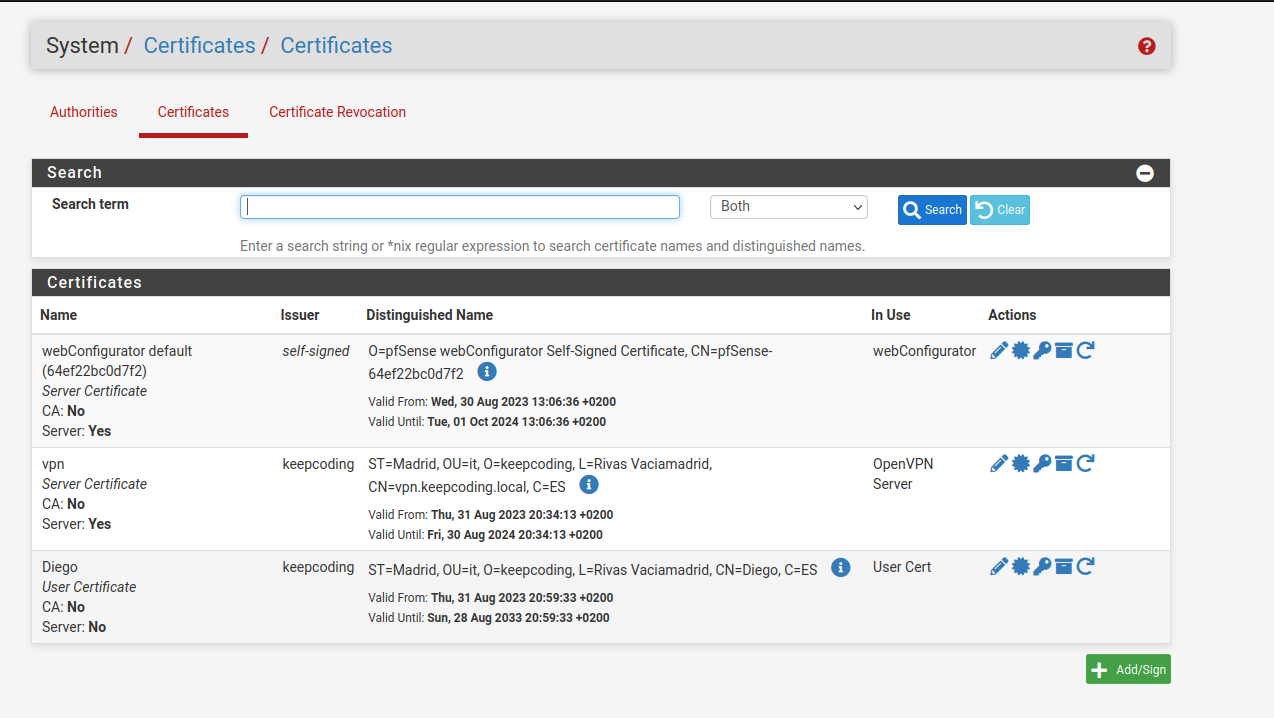
Buscamos e instalamos el paquete openvpn-client-export

Seguidamente vamos a System/Certificate Manager/CAs. Vamos a crear una entidad certificadora dentro de pfSense.

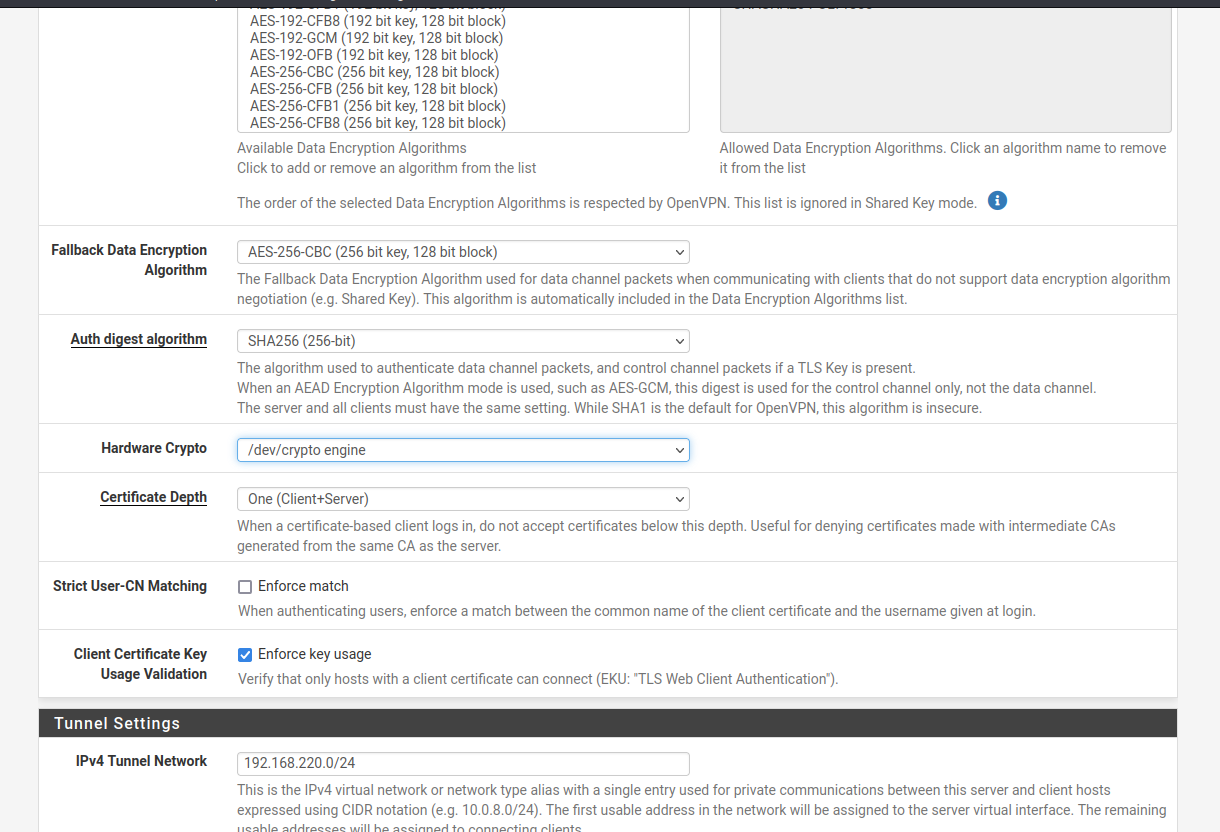
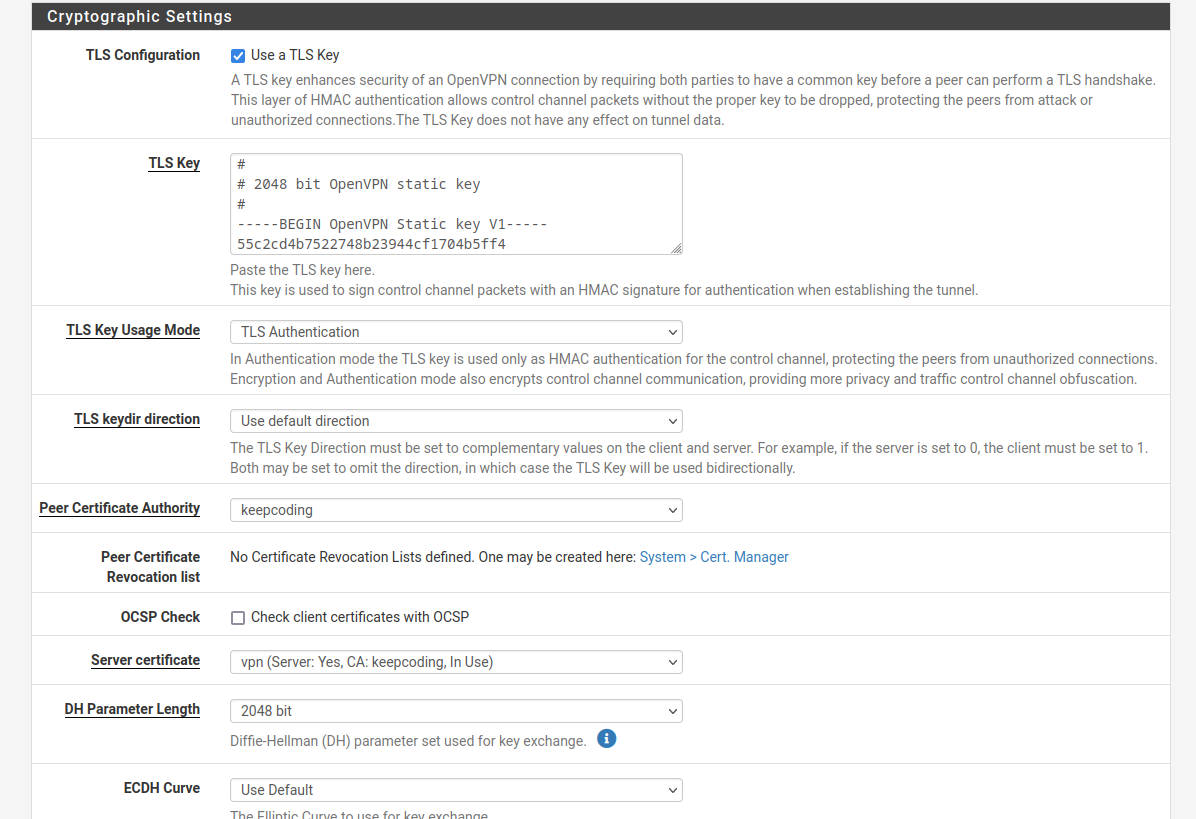
Añadimos una nueva autoridad certificadora con esta configuración llamada keepcoding



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



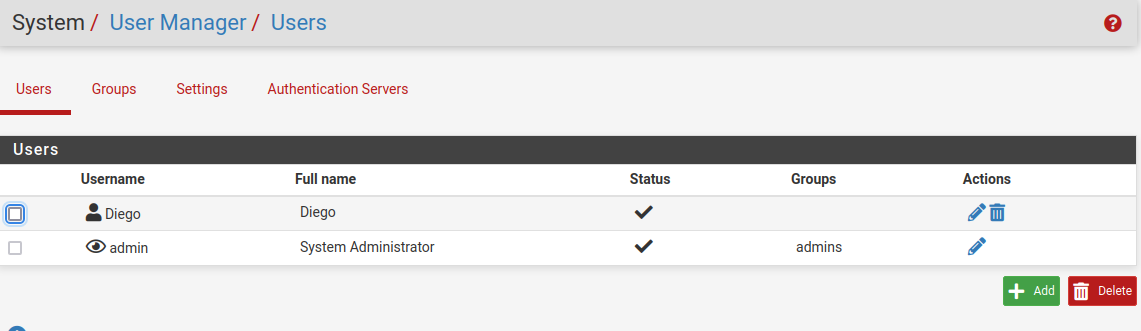
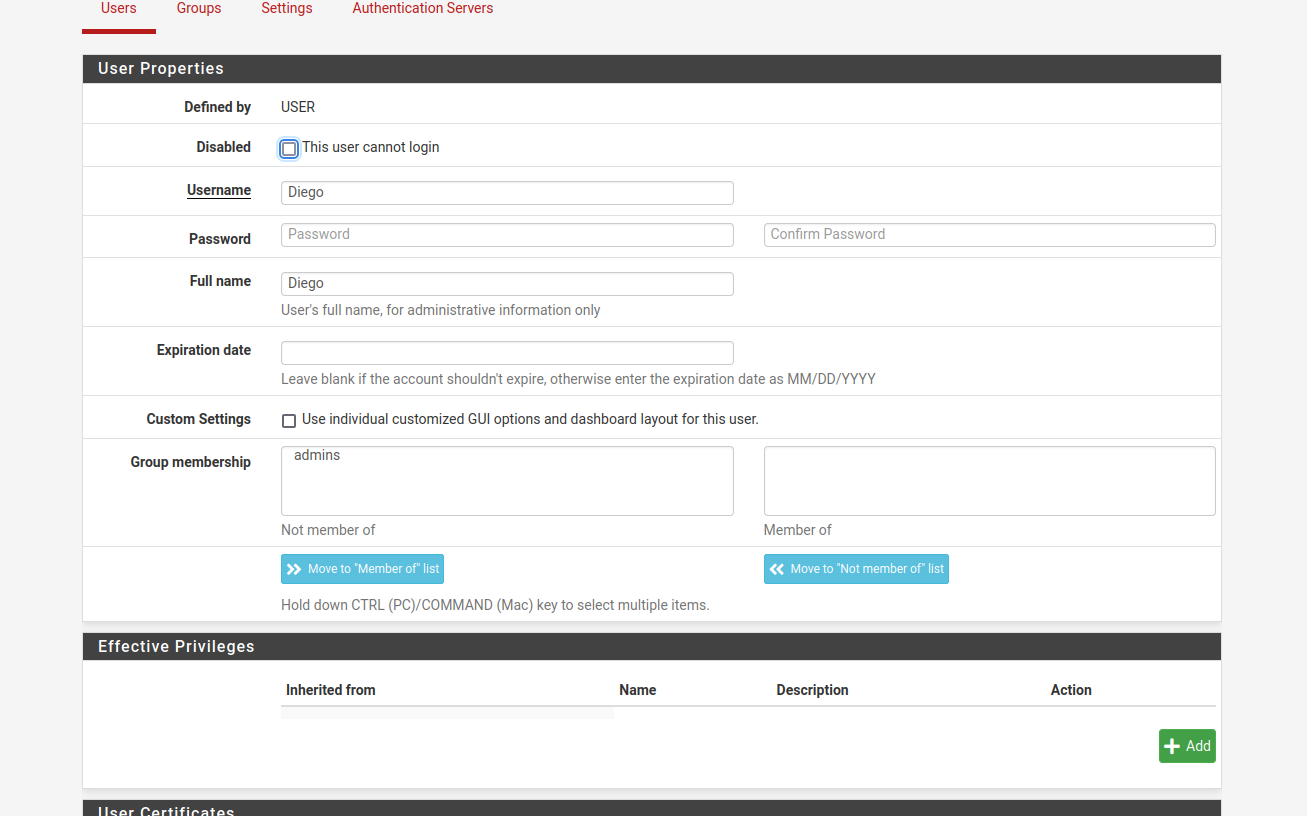
Seguidamente vamos a VPN/Open VPN/Servers para crear un servidor que va a ir a IT por la VPN. Seleccionamos Remote Access, doble factor de autenticación, por certificado y usuario y el server certificate UTM que hemos creado antes.



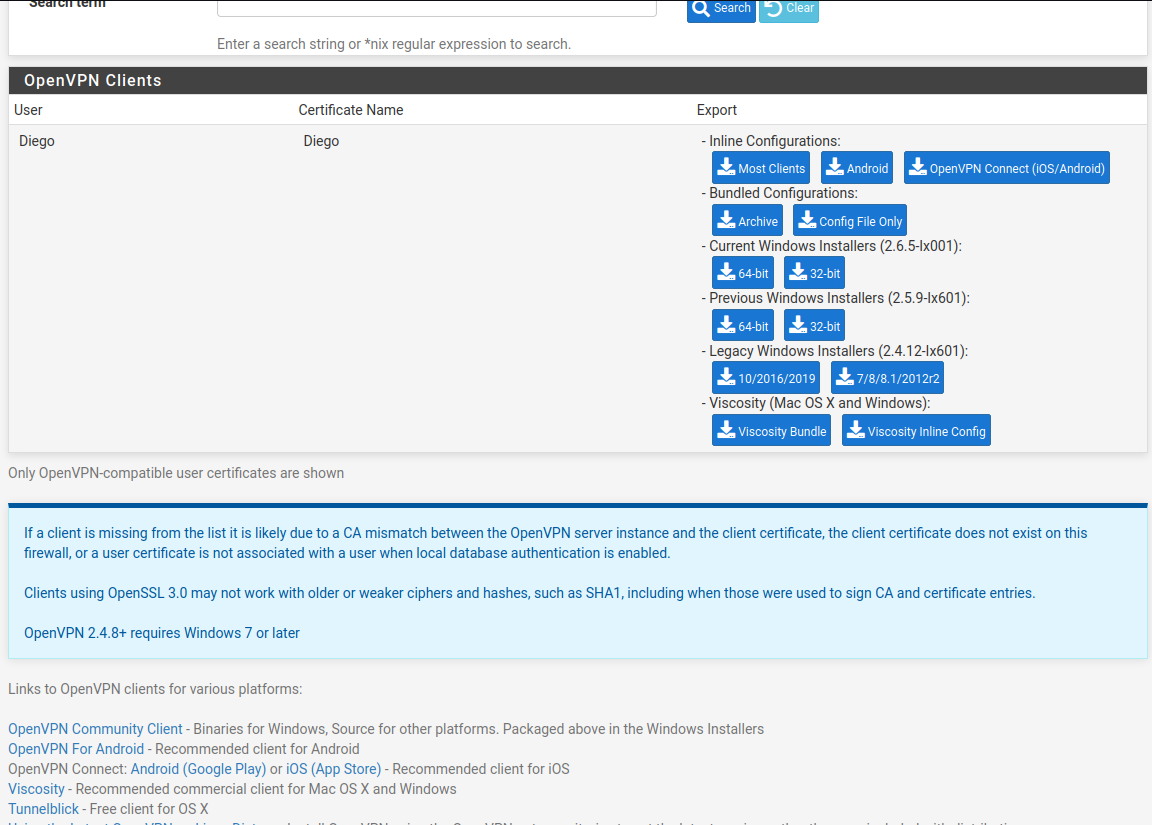
También es importante la configuración de Tunnel Settings. Si se habilita aquí la opción Redirect IPv4/IPv6 Gateway, todo el tráfico que salga de localhost va a pasar a través del UTM, y si localhost sale a internet, lo hará con la IP publica de UTM. En este caso, se deshabilita y solo se incluye la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



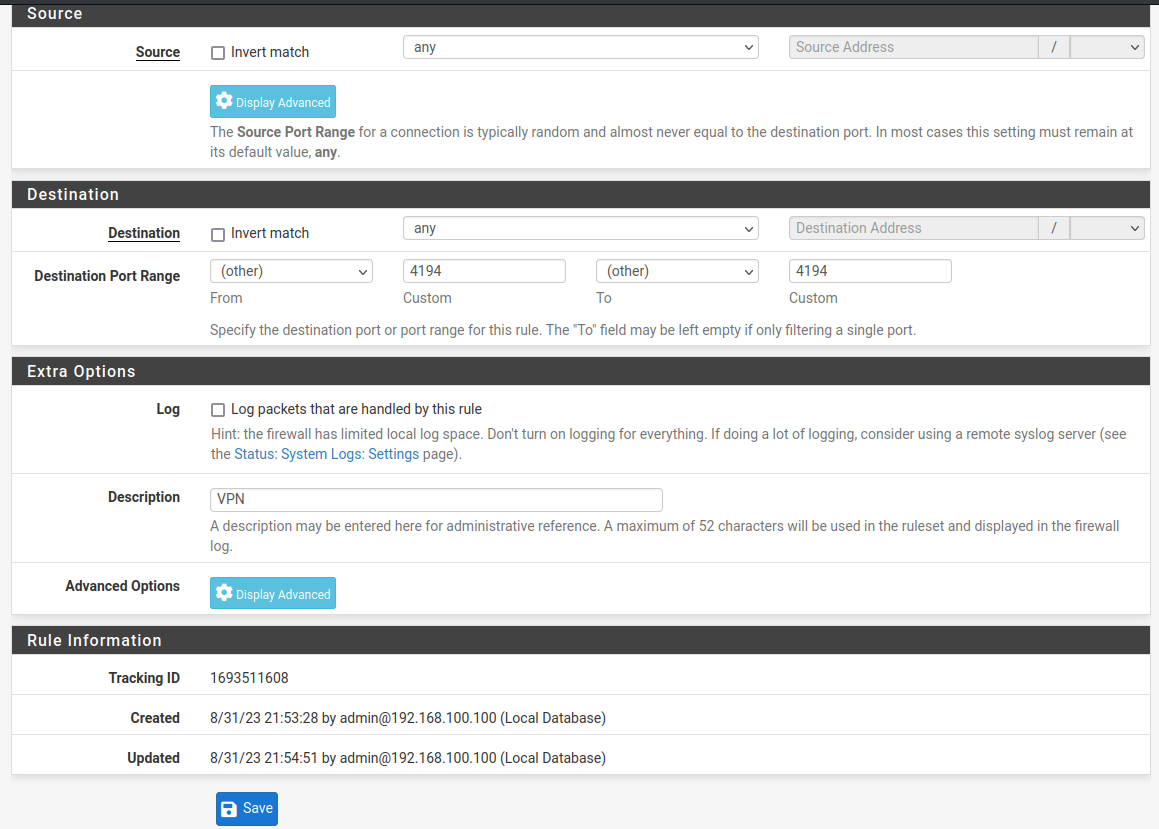
Seguidamente vamos a crear un usuario que se pueda conectar al tunnel network. Vamos a System/User Manager/Users y añadimos nuevo usuario donde lo importante es crear un certificado de usuario con el CA creado anteriormente. Comprobamos después en System/Certificate Manager/ Certificates que este certificado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



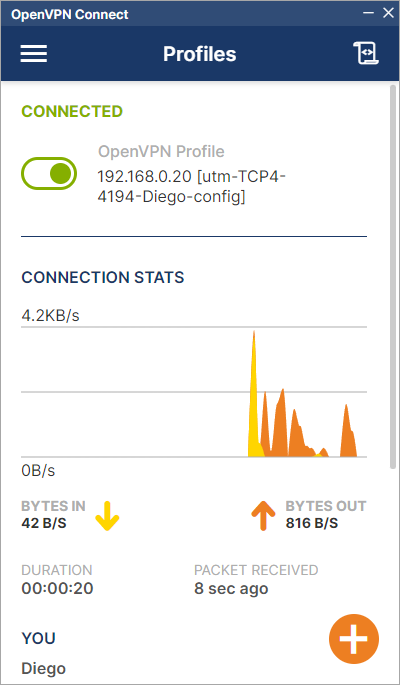
Después vamos o OpenVPN/Client export. Descargamos el archivo Inline configurations/Most clients y lo llevamos y abrimos en nuestro localhost.



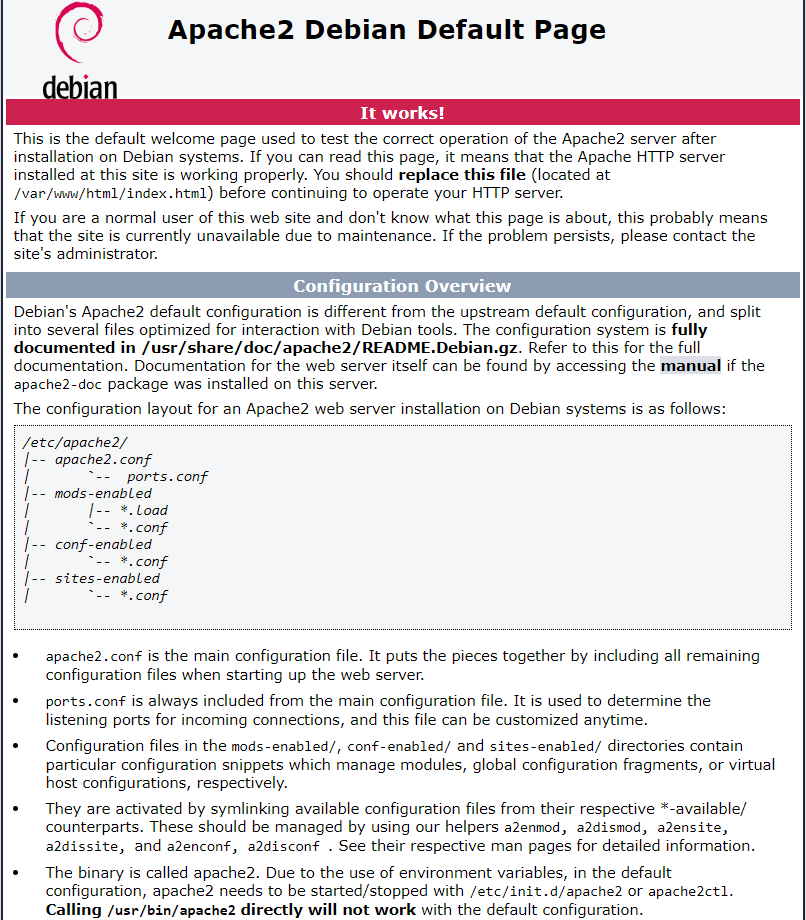
Antes de conectar nuestro localhost a la VPN, hay que crear una nueva regla en Firewall/Rules/WAN. Añadimos nueva regla, dejando todas las opciones por defecto excepto Destination y Extra options, que las configuramos así:



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



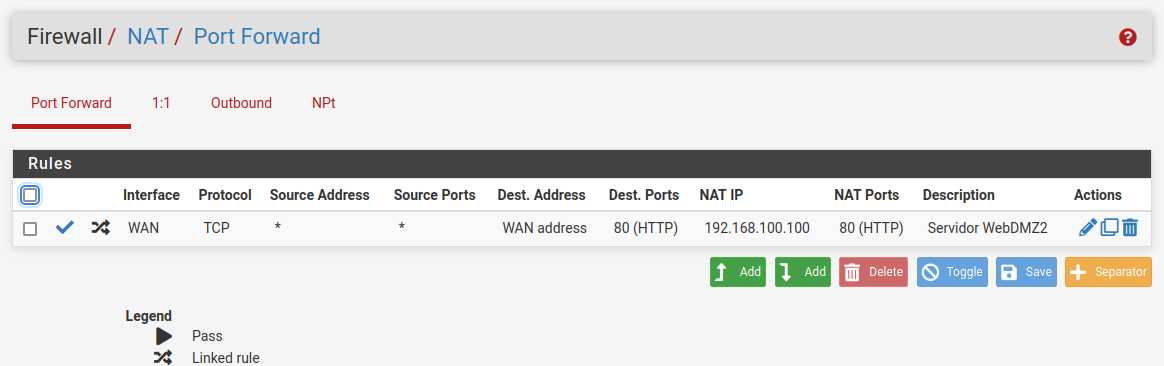
Comprobamos después que desde el navegador de localhost podemos alcanzar el servidor apache2 activo en la IP del IT (nuestra Kali en este caso)



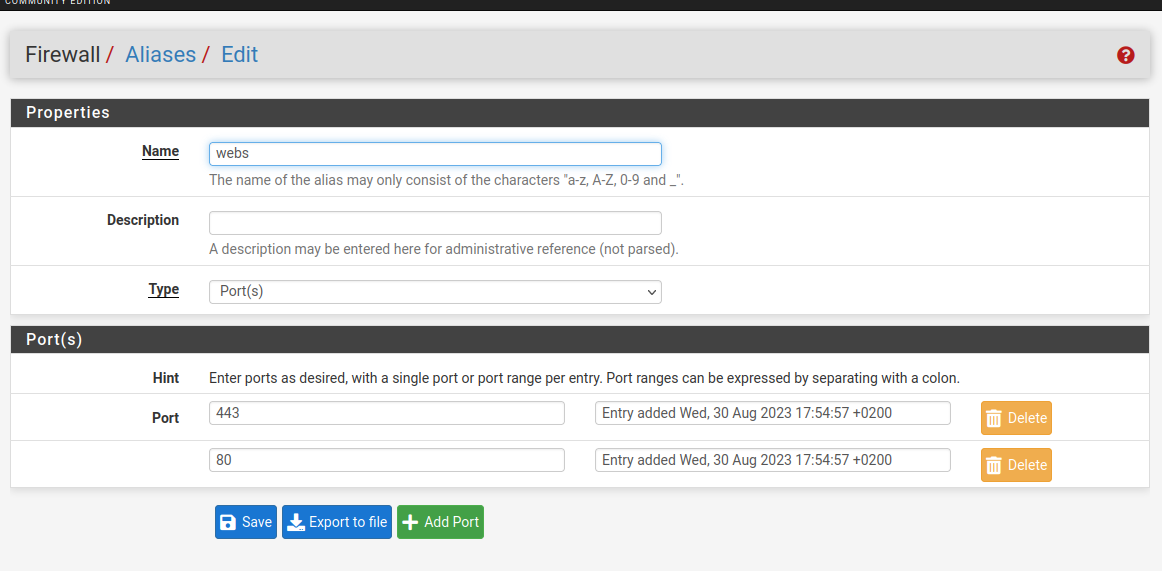
* 1. ***Reglas DMZ***

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

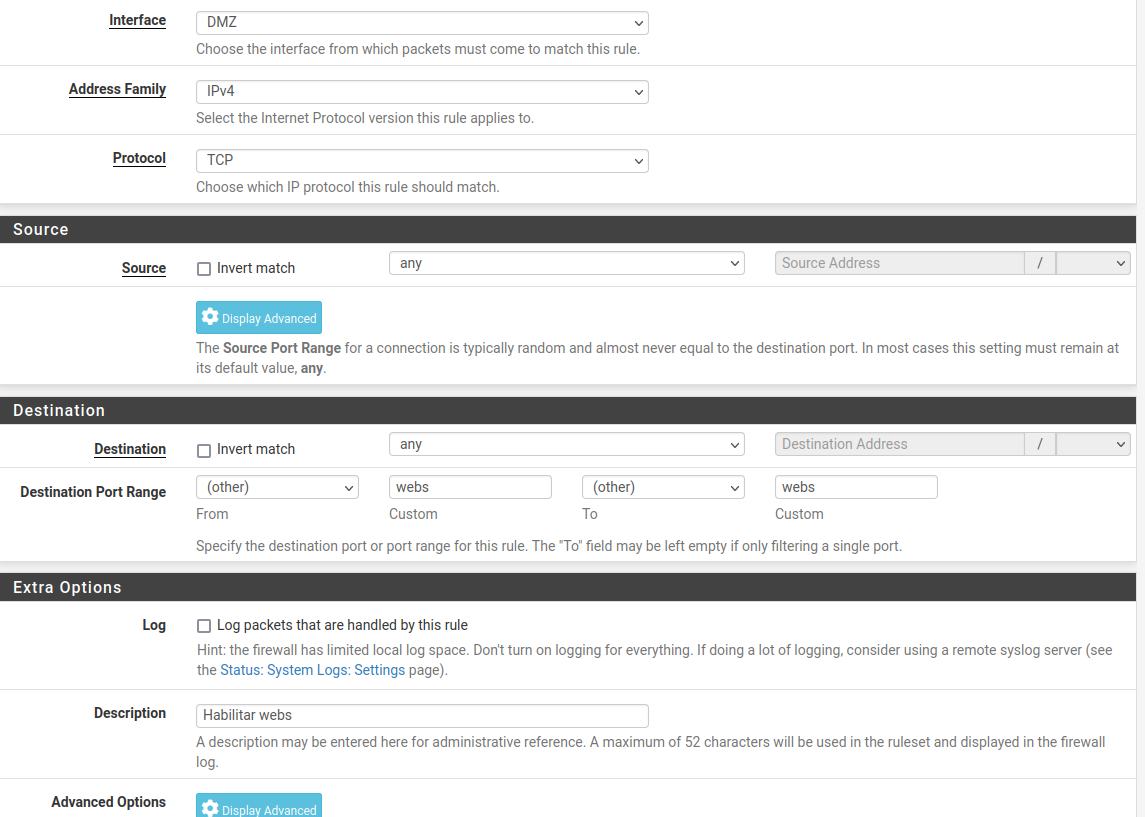
Primero creamos un nuevo NAT en Firewall/NAT/Port Forward con la siguiente configuración. Desde la interfaz WAN con destino a la IP de WAN y tráfico HTTPS, va a reenviar el tráfico al puerto 80. La IP a la que va a redirigir es la 192.168.200.1, que es la que habíamos creado para DMZ.



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.

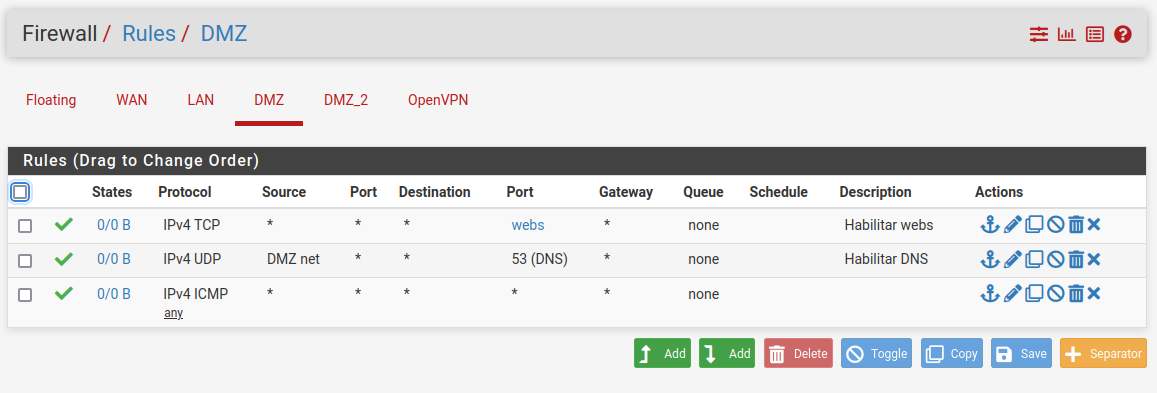


Vamos luego a Firewall/Rules/DMZ y aplicamos este alias a las rules de DMZ



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

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.



Instalacion de herramientas

Instalacion de suricata

1. ***Instalación de herramientas***

***2.1. Suricata***

Suricata es una herramienta IDS (Intrusión 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.

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

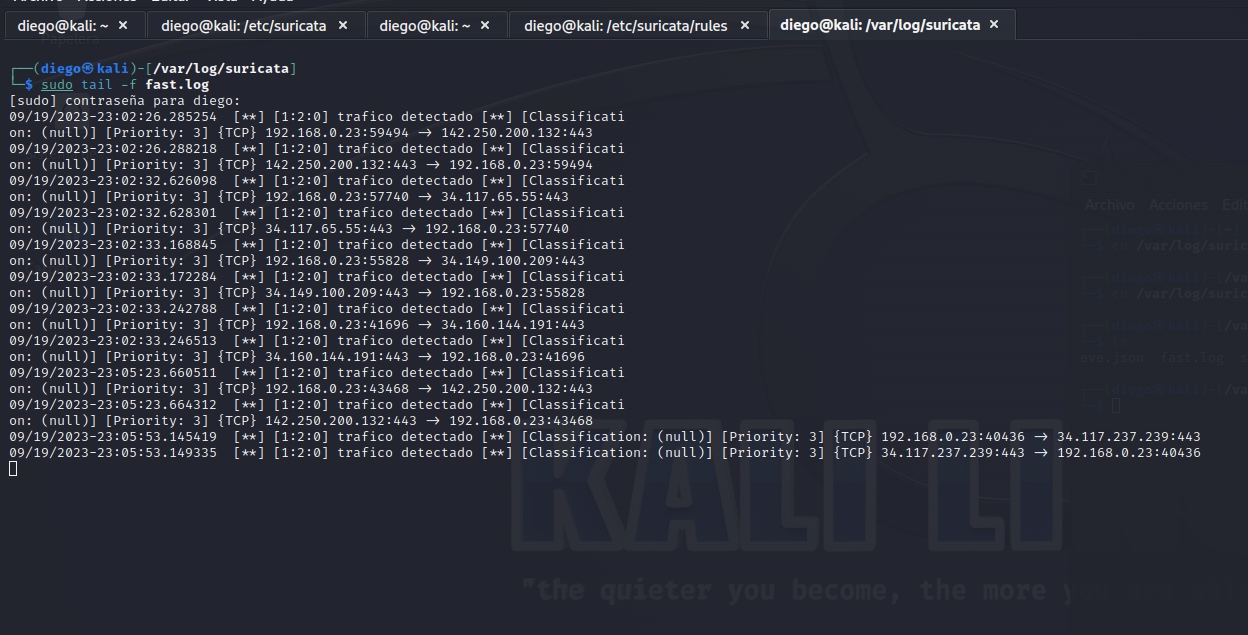
apt install suricata

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



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.

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



***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.

Vamos a realizar la instalación de Elastic Stack (ELK) y de Kibana mediante los comandos en la terminal

sudo docker run -d --name elasticsearch -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node" docker.elastic.co/elasticsearch/elasticsearch:7.14.0

docker run -d --name kibana --link elasticsearch:elasticsearch -p 5601:5601 docker.elastic.co/kibana/kibana:7.14.0

Conexión entre equipos de la red interna

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

• La LAN 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.

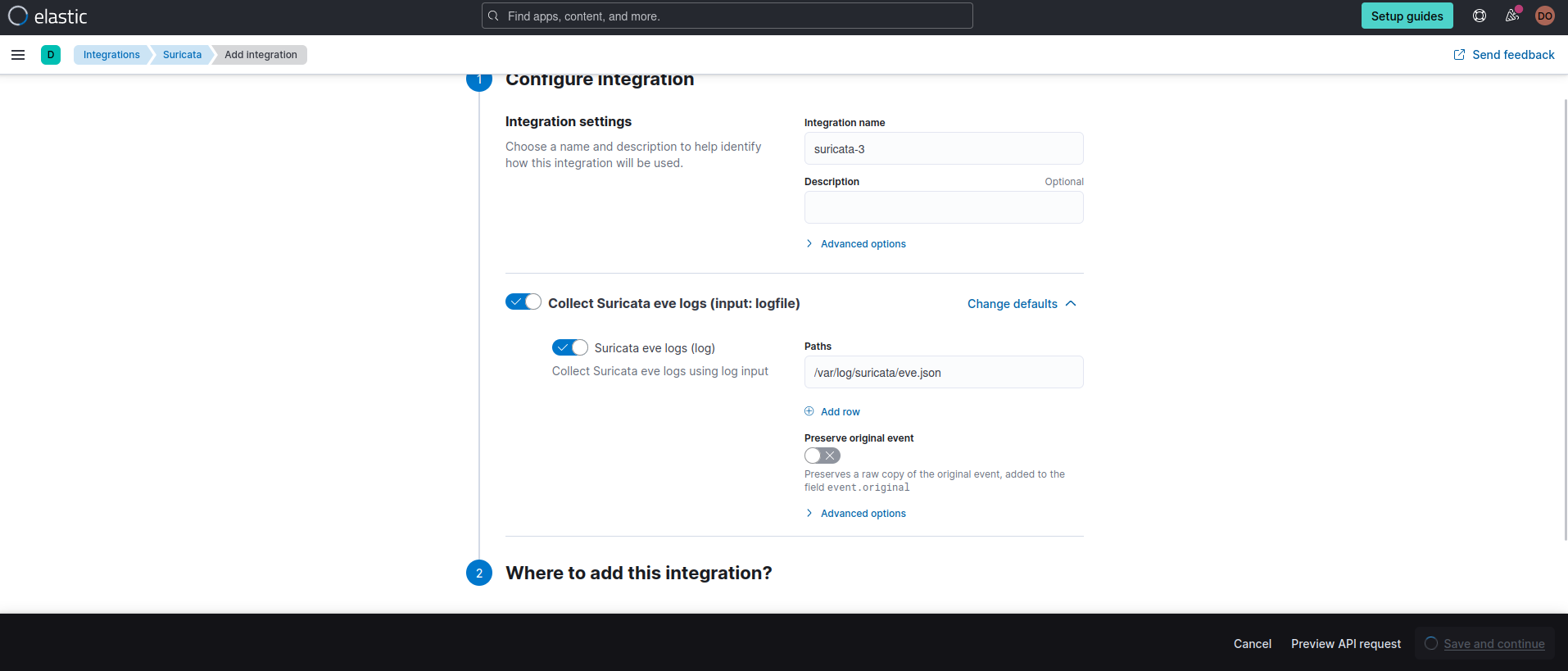
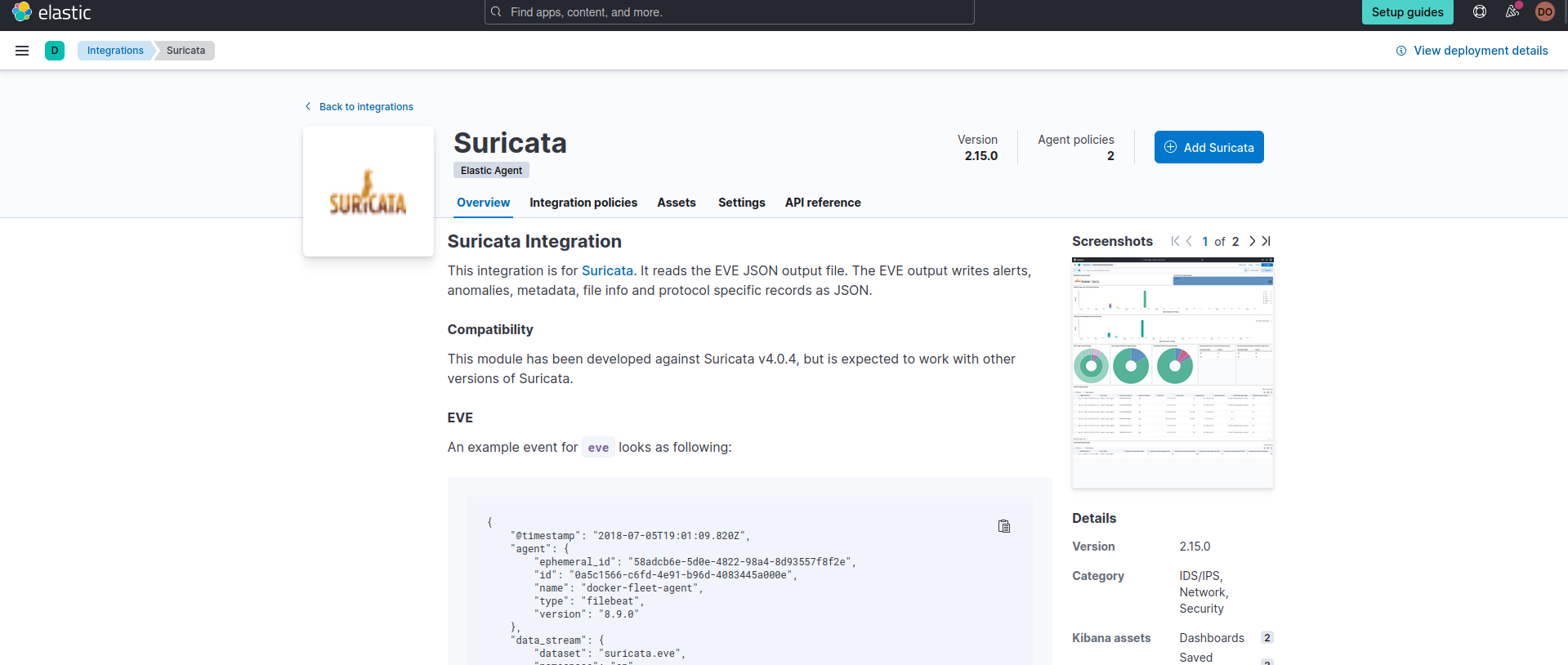
• El servidor ELK debe almacenar y mostrar los logs del honeypot

1. **Conexión entre equipos de la red interna**

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

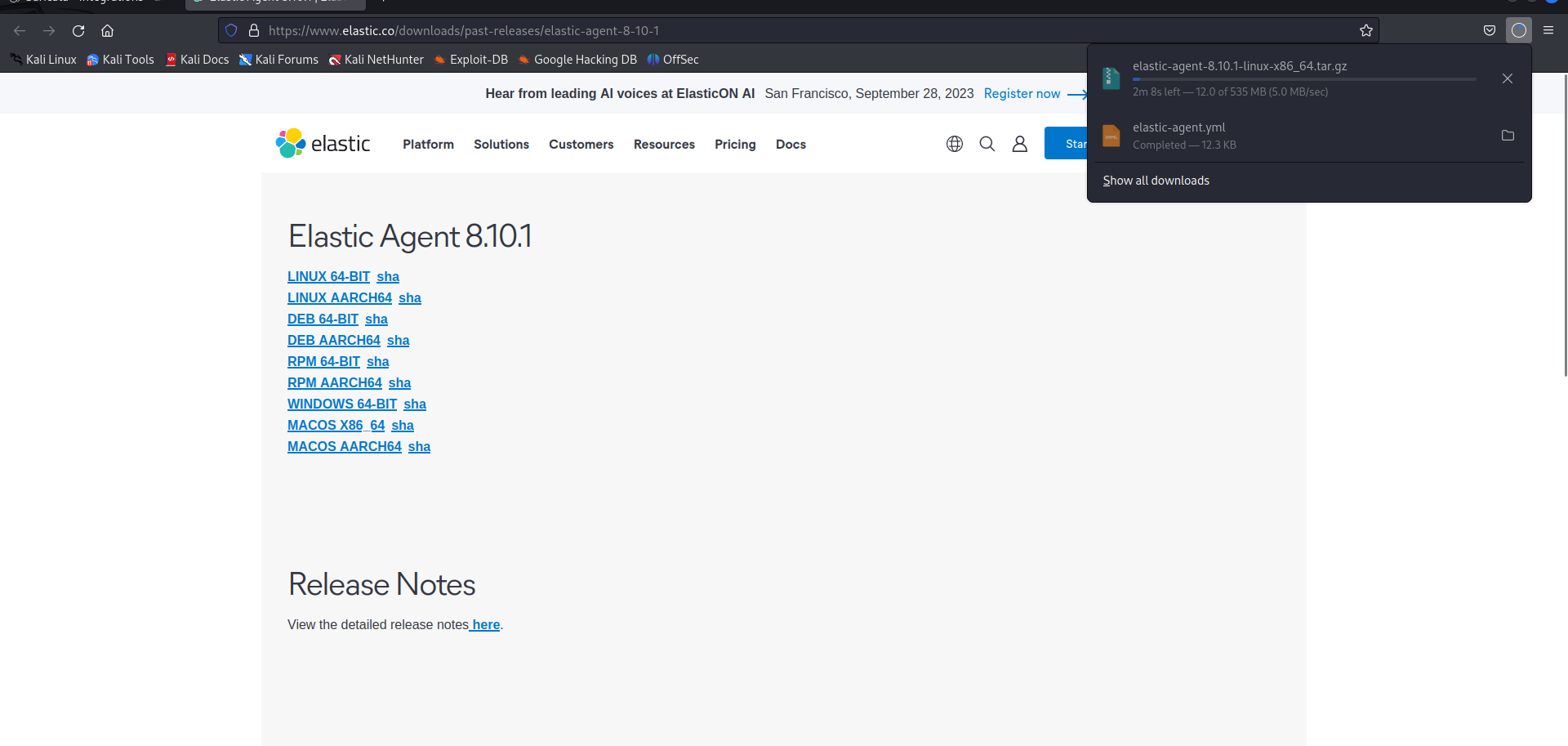
En primer lugar vamos a añadir la integración con Suricata en el ELK.

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:



Abrimos la pagina de descarga del agente en otra pestaña del navegador y

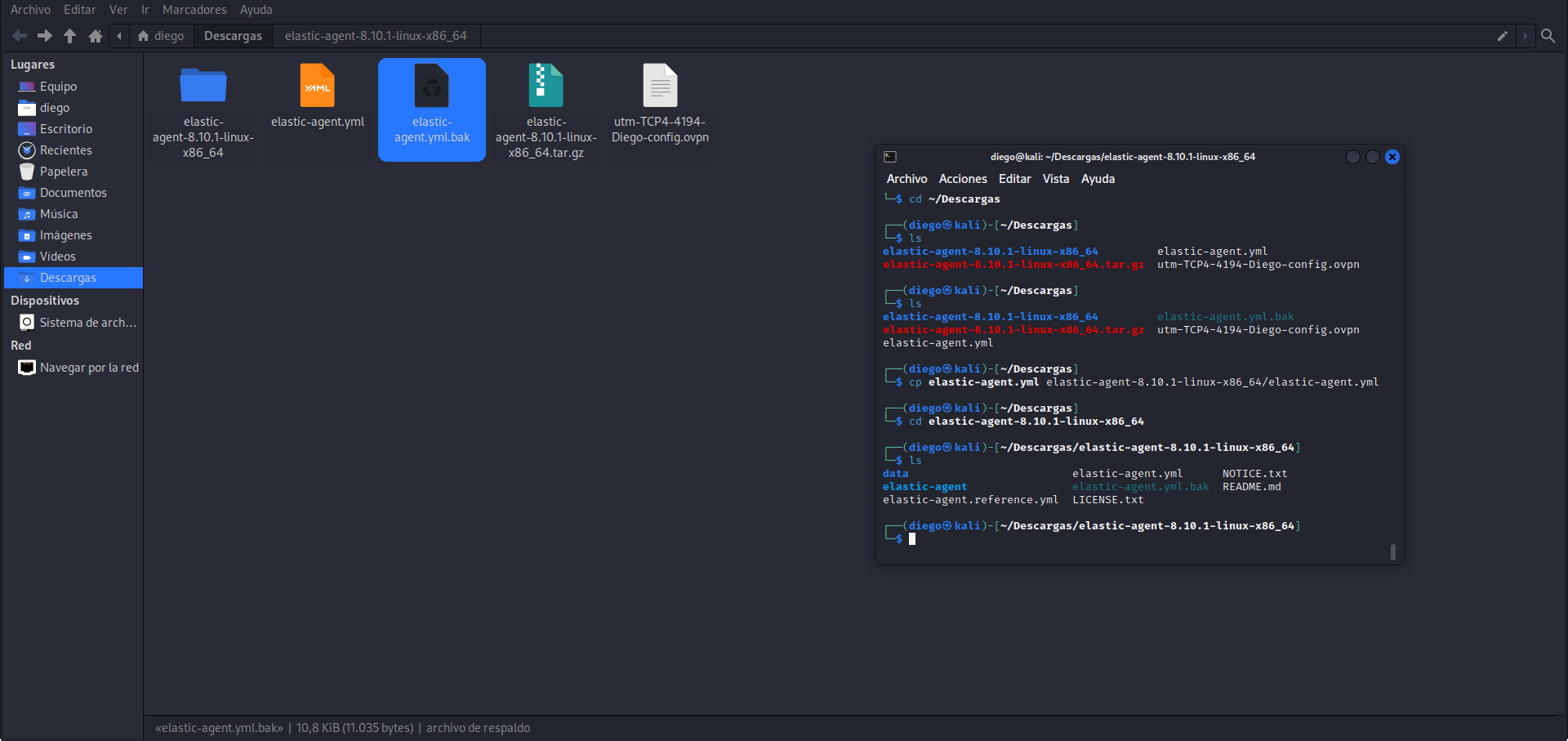
descargamos la Policy (archivo .yml)



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

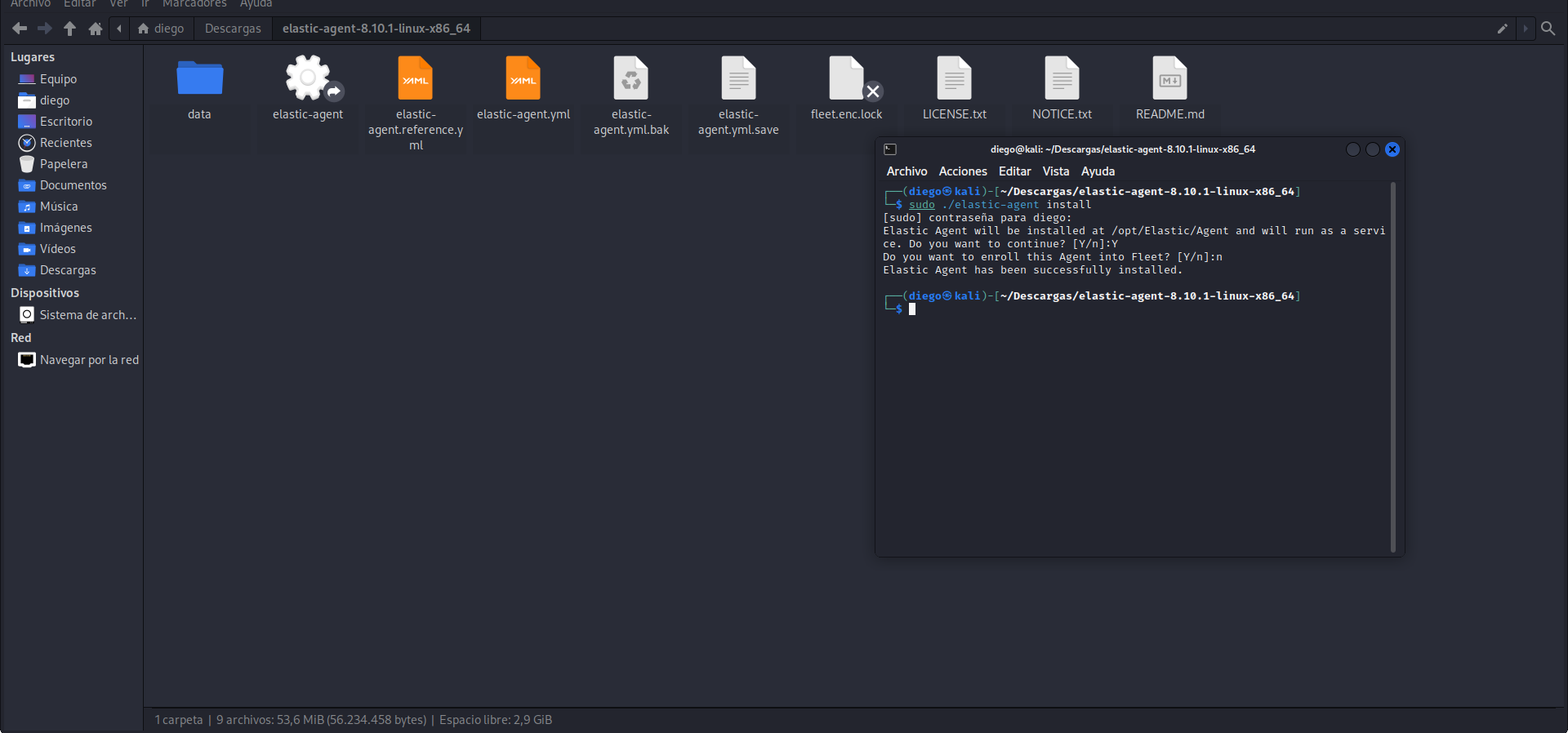
Descomprimimos el archivo Elastic Agent 8.1.0

El archivo .yml de la Policy lo copiamos dentro del directorio del elasticagent

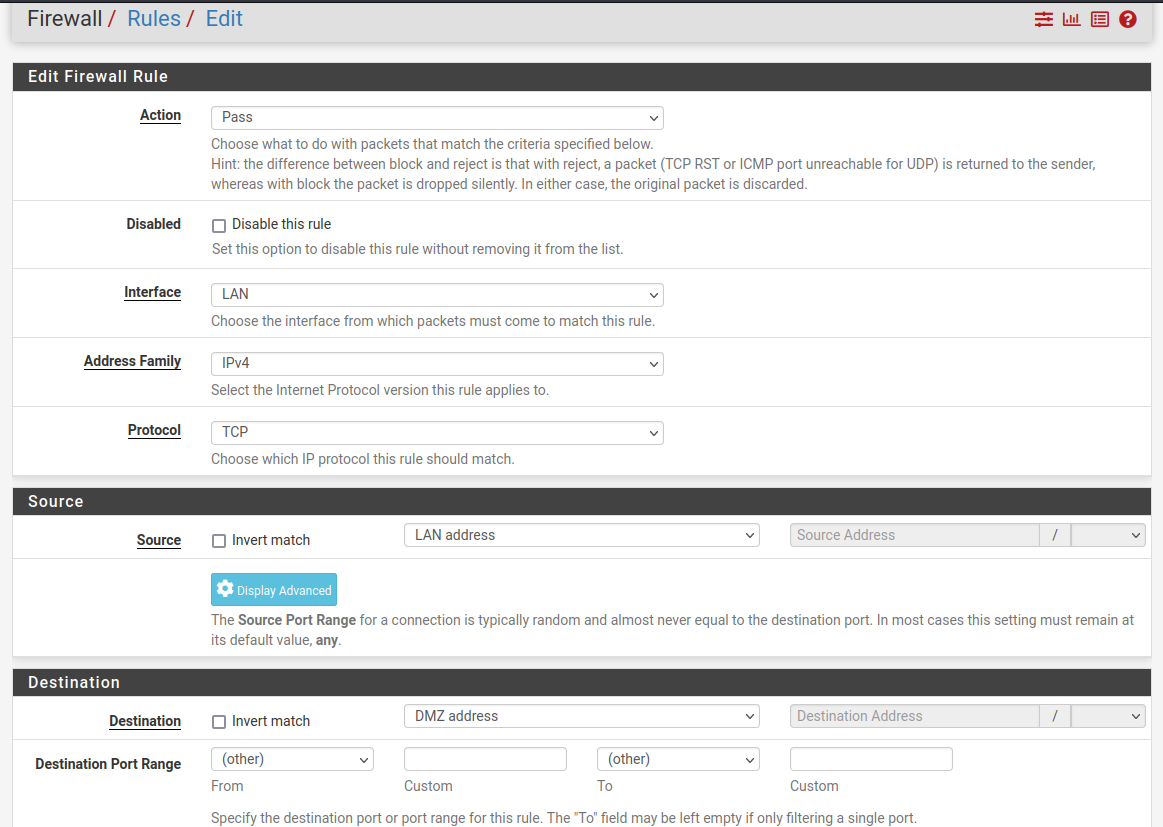


Con usuario root vamos a /opt/Elastic/Agent y modificamos el archivo elastic-agent.yml para introducir la IP de la máquina Kali (DMZ) que aloja el ELK y el usuario y contraseña de inicio de sesión del Stack.

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



Ahora vamos a pfSense y creamos una regla en el firewall (Firewall/Rules/ LAN) para permitir las conexiones TCP desde la IP del LAN (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 LAN 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.

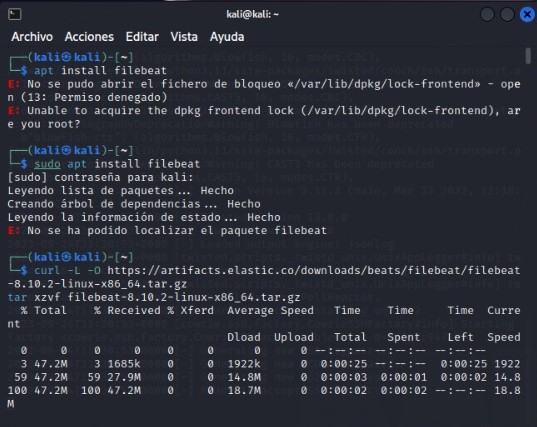


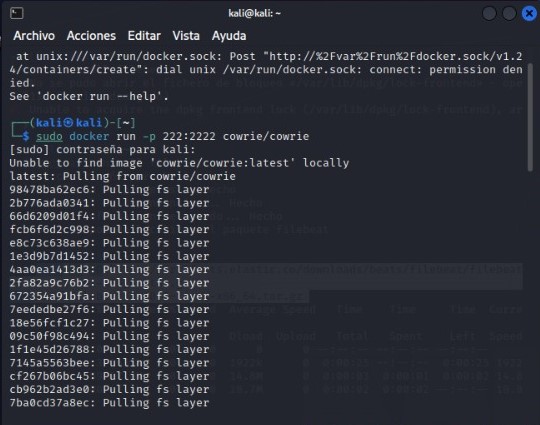
Nos vamos a la interfaz del ELK en la Kali DMZ, cerramos las ventanas de integraciones y vamos al dashboard principal pulsando el logo de Elastic arriba a la izquierda en la interfaz. Nos movemos a Analytics/Discover desde el menú lateral izquierdo y podremos ver los eventos de tráfico registrados desde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

* 1. ***Creación e integración de honeypot en Elastic Stack.***

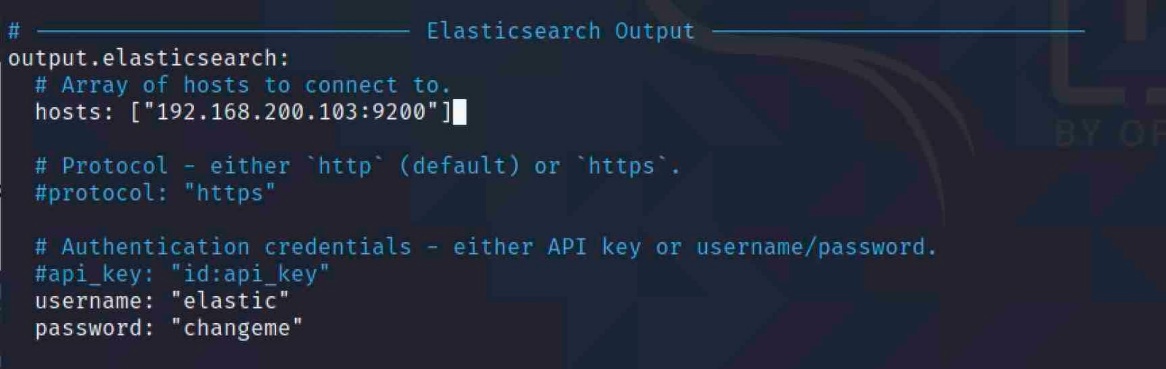
Un honeypot es un sistema o equipo informático que se configura como si fuese un sistema real con vulnerabilidades que pueden ser explotadas. Se encuentra aislado del resto de la red y monitorizado para poder observar y analizar cuáles son los procedimientos que están siguiendo los posibles atacantes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

Instalamos filebeat para enviar los logs y también instalamos el cowrie





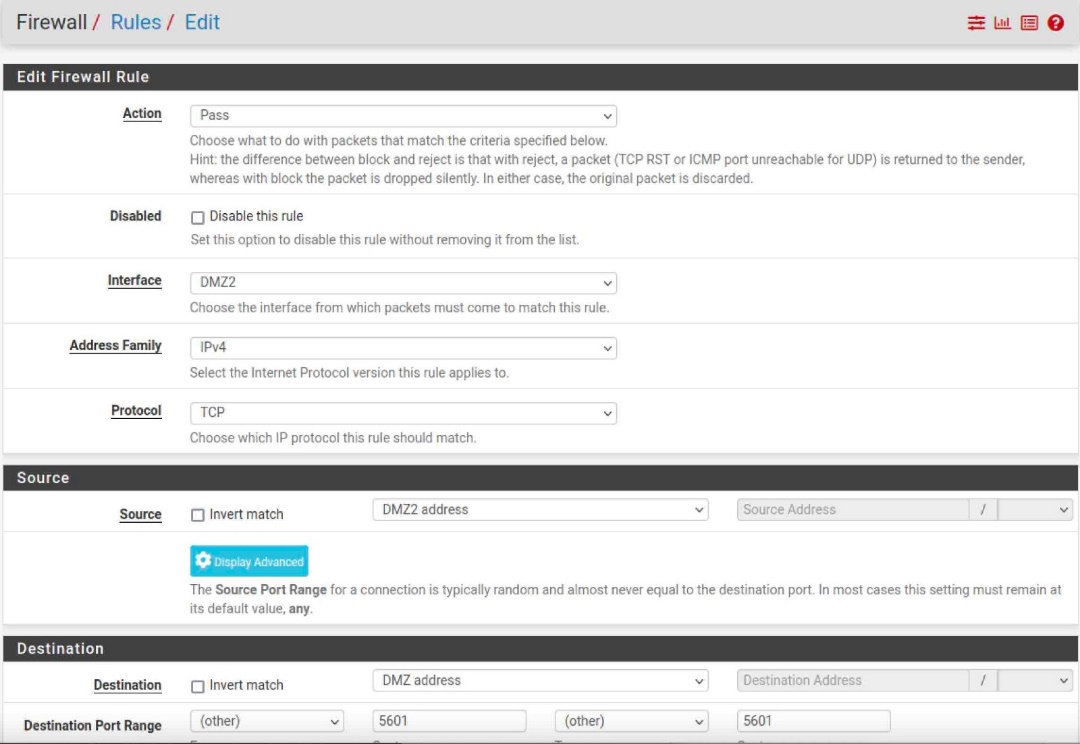
Configuramos el archivo filebeat.yml ubicado en /etc/filebeat

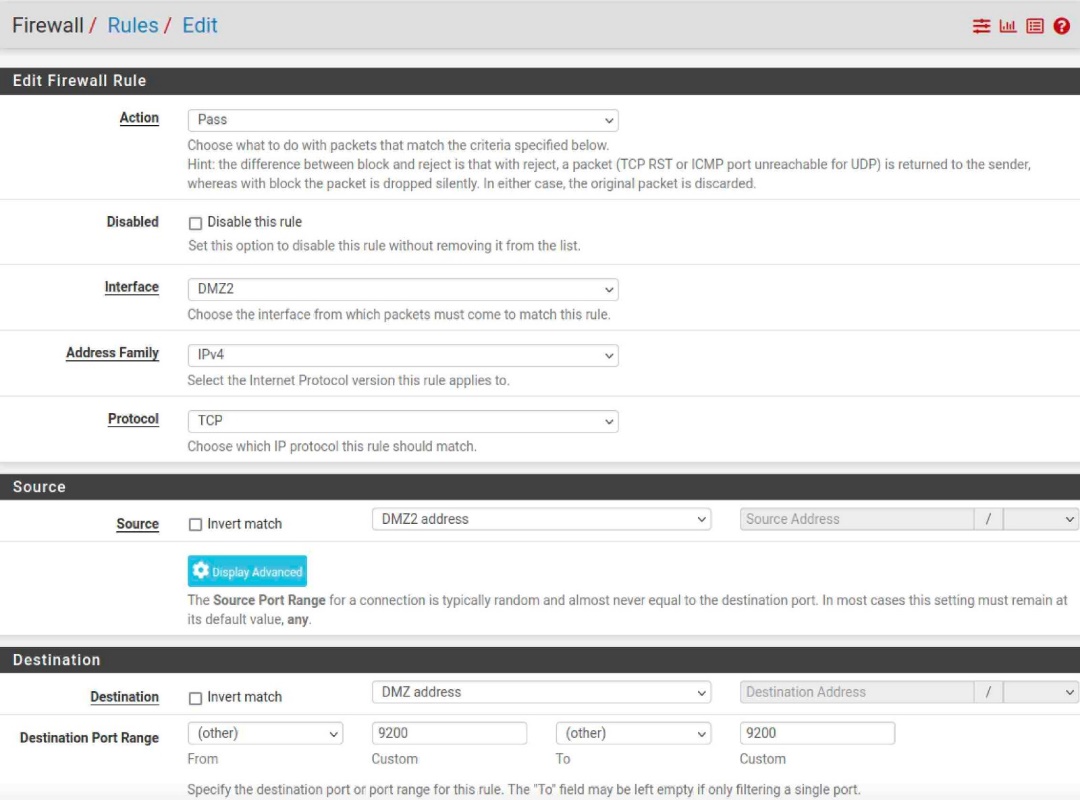


En el apartado Filebeat inputs hemos habilitado y establecido la ruta de la cual Filebeat recogerá los logs de cowrie. Después, en Elasticsearch Output indicamos la IP y puerto de la maquina DMZ donde tenemos alojado Elastic Stack y el usuario y password de acceso.

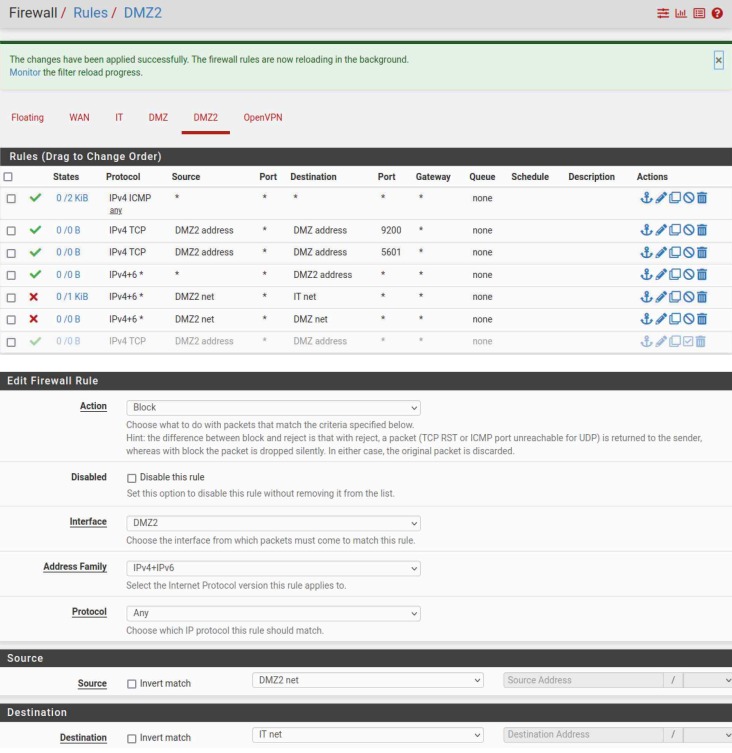
Seguidamente vamos a crear las reglas necesarias en el pfSense para que DMZ2 (que

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:





Después creamos reglas Block para que no permitan el tráfico proveniente de DMZ2 al resto de equipos de la red (LAN y DMZ). Lo único que permitimos es el tráfico 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:



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

Habilitamos la regla Block y ya la conexión DMZ2-LAN 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.

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

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