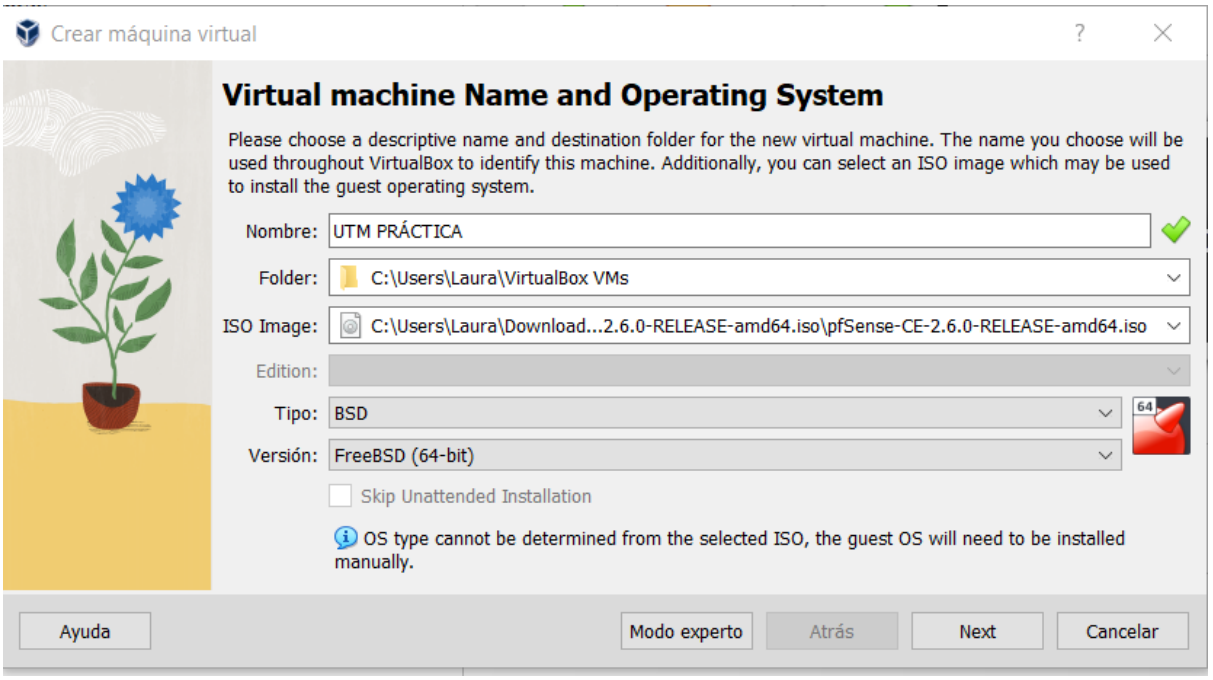
PRÁCTICA BLUETEAM

Laura Quijorna Velázquez
Abril de 2023

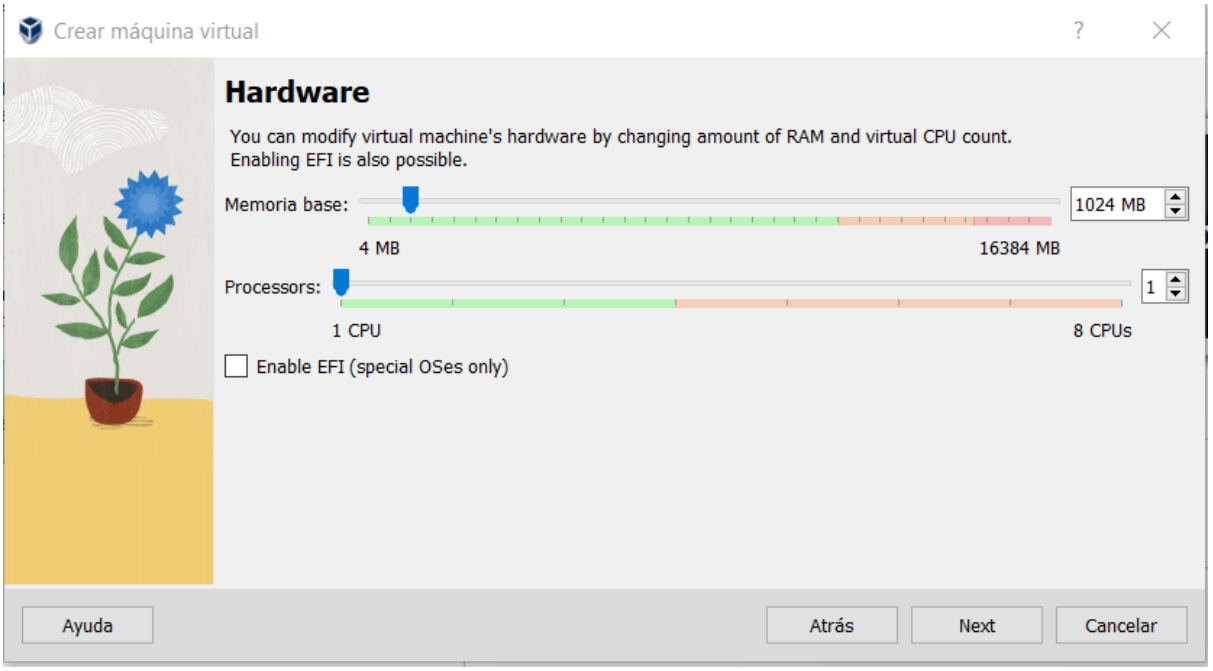
La práctica que se presenta a continuación describe el montaje de una infraestructura de red con PFSense y creación de un servidor VPN para posteriormente a través de un honeypot dar entrada a conexiones de ciberatacantes que serán registradas a través de un suricata y visualizadas a través de un Elastic Agent.

Creación de mi infraestructura en PFSENSE

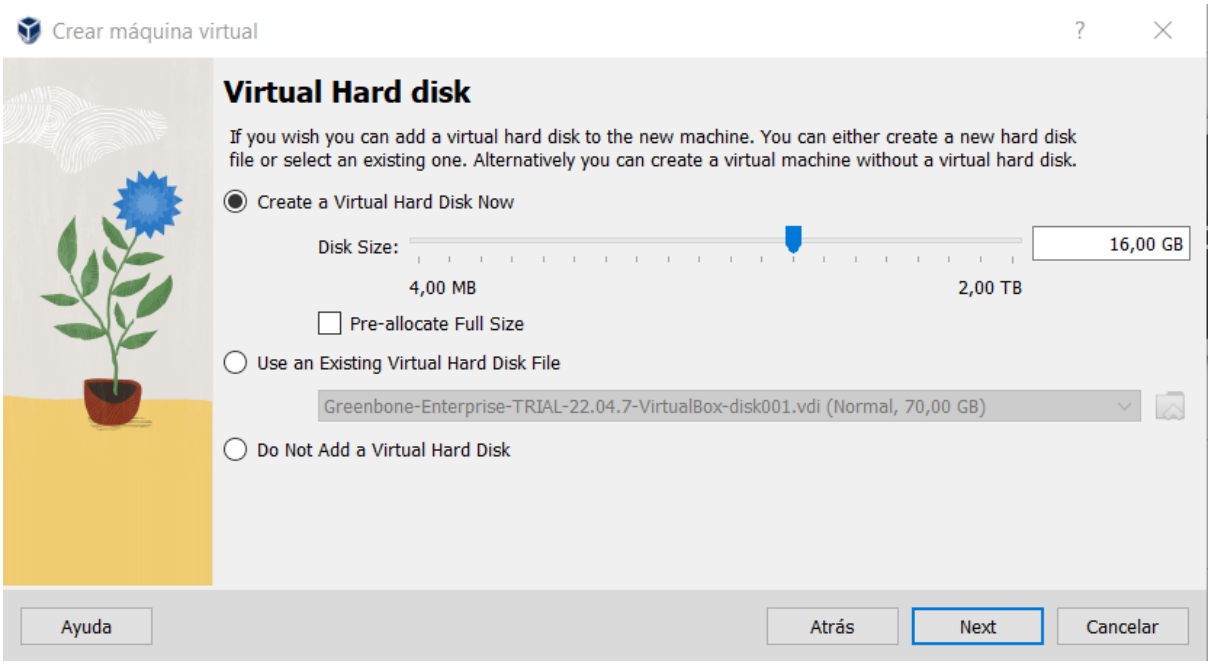
Primero descargo pfsense y creo mi máquina en Virtual Box. Para ello, pincho en “nueva”, pongo el nombre de la máquina, en mi caso la llamaré UTM PRÁCTICA. Después selecciono donde lo voy a guardar, cargo la iso, selecciono en tipo la opción BSD y en versión la FreeBSD (64-bit). Hago click en Next.



Ahora selecciono cuánto le voy a dar de memoria ram y procesamiento. Lo dejaré como viene por defecto. Pincho Next.

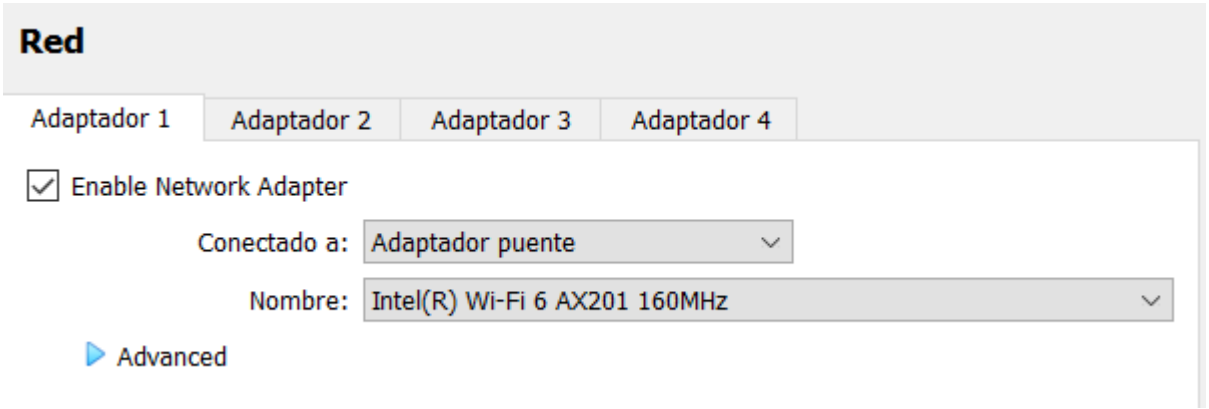


Igualmente selecciono cuánto le voy a dar de disco duro. Lo dejaré tal cual viene por defecto. Click en Next.

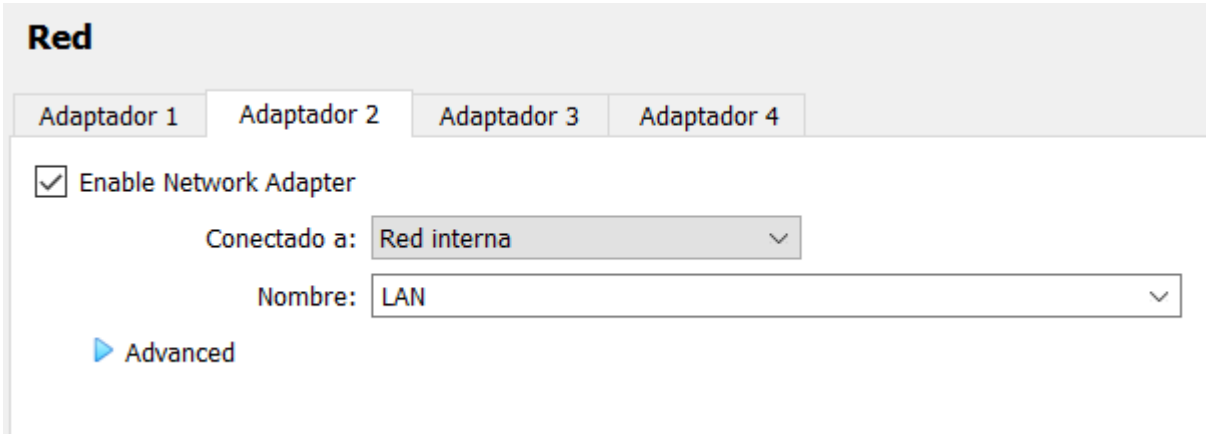


En la siguiente pantalla que aparece pincho en “Terminar” y seguidamente ya aparecerá mi máquina UTM PRÁCTICA en mi Virtual Box.

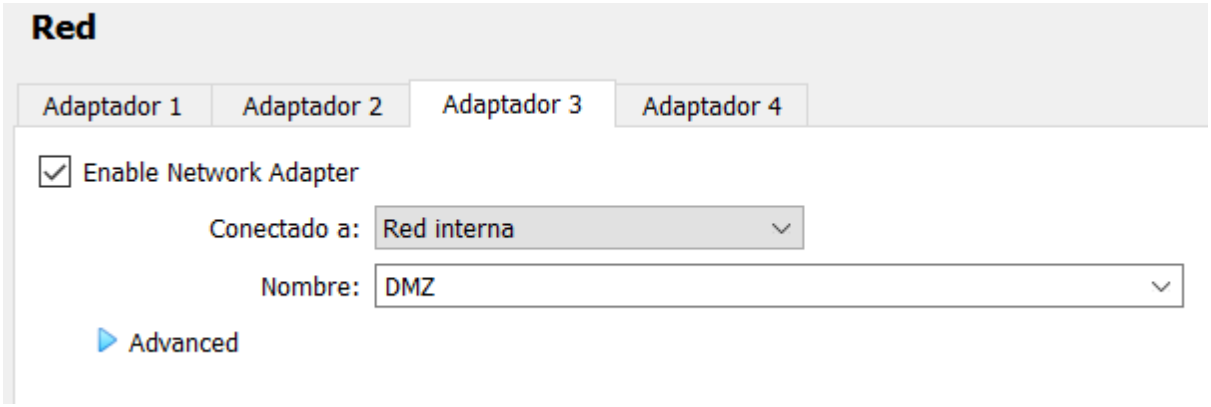
A continuación, procederé a crear las redes WAN (mi red, mi UTM), LAN, DMZ y DMZ_2. Para ello, con mi máquina UTM PRÁCTICA seleccionada, pincho en Configuración y voy al apartado redes. Crearé primero la red WAN en modo bridge (Adaptador puente).



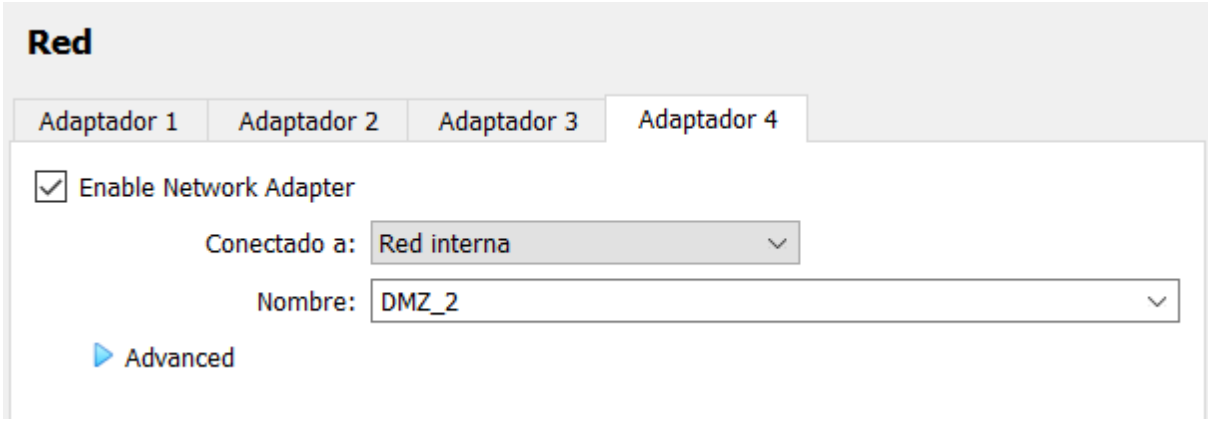
Después me voy a la pestaña de Adaptador 2 y crearé mi red con nombre de etiqueta LAN.



Ahora crearé la DMZ, igualmente red interna, en la pestaña Adaptador 3.



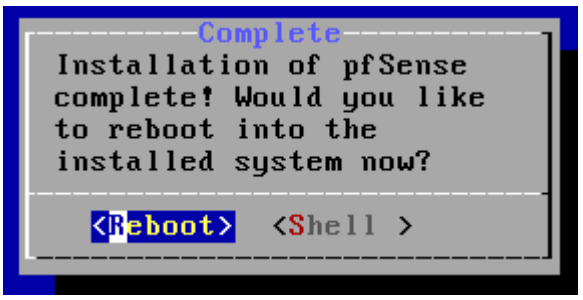
Y para terminar mi infraestructura, añado DMZ_2 como red interna también.



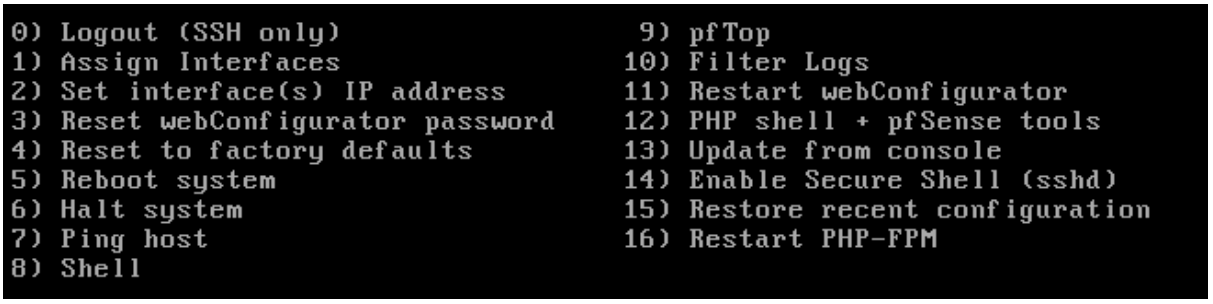
También hay que configurar en “Almacenamiento”, en el discoLA casilla “CD/DVD vivo”

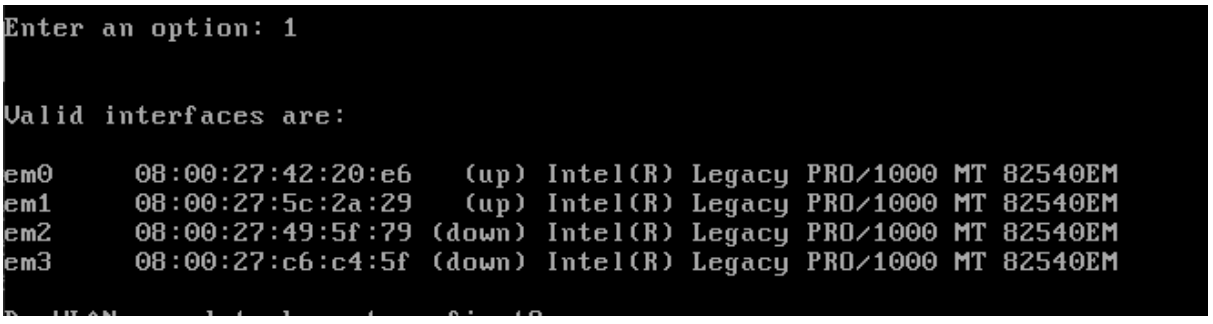
Una vez creada la infraestructura, encenderé la máquina virtual y procederé a seguir los pasos de la instalación de pfSense. Todo lo dejaré como viene por defecto menos el idioma, que seleccionaré Spanish (key accents), el disco duro, que lo seleccionaré en la opción que sale y diré “yes” en cuanto a la destrucción de los datos.

Cuando salga la pantalla que muestro a continuación, daremos en Reboot e inmediatamente iremos a la pestaña de Dispositivo - Unidades Ópticas - Eliminar disco de la unidad virtual. Esto es para que no me arranque con el disco.



Para continuar con la configuración voy a comprobar que las mac de mis redes creadas corresponden con las que salen en la terminal pfsense. Como puedo observar, tengo una lista de acciones enumeradas. Para poner ver las mac por terminal, escribo la opción “1” en la consola, . Después compruebo en el apartado de Configuración de la interfaz gráfica, en Red si coinciden.





He podido comprobar que todas coinciden.

En el siguiente paso, voy a comprobar las ips y las interfaces escribiendo la Opción “2”. Me salen dos interfaces, la LAN y la WAN y esta opción me pregunta que cuál es el número de interfaces que deseo configurar. A ésto respondo 2 y configuro ambas interfaces.
La primera a configurar es la LAN y me dice que le asigne la ip la cual me dará acceso al router y por tanto me servirá como puerta de enlace. Le pondré la 192.168.100.254.
A continuación, me pide la máscara de subred a la cual pondré /24. Después me dice si quiero asignar una puerta de enlace. A esto diré que no ya que la misma ip de la LAN será la puerta de enlace, por lo que no necesito ninguna. También me pide que introduzca la ipv6 la cual tampoco introduciré nada.

```
Available interfaces:

1 - WAN (em0 - dhcp, dhcp6)
2 - LAN (em1 - static)
3 - OPT1 (em2)
4 - OPT2 (em3)

Enter the number of the interface you wish to configure: 2

Enter the new LAN IPv4 address.  Press <ENTER> for none:
> 192.168.100.254

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
     255.255.0.0   = 16
     255.0.0.0     = 8

Enter the new LAN IPv4 subnet bit count (1 to 32):
> 24
```

Ahora me pregunta si quiero activar el DHCP en la LAN, a lo que diré que sí y a lo que tendré que introducir el rango de ips que irá de la 192.168.100.100 a la 192.168.100.200

```
For a WAN, enter the new LAN IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
>

Enter the new LAN IPv6 address.  Press <ENTER> for none:
>

Do you want to enable the DHCP server on LAN? (y/n)

Do you want to enable the DHCP server on LAN? (y/n) y
Enter the start address of the IPv4 client address range: 192.168.100.100
Enter the end address of the IPv4 client address range: 192.168.100.200
Disabling IPv6 DHCPD...
```

Después me pregunta si quiero activar el panel web en el pfsense, a lo que responderé que Sí.

```
Do you want to revert to HTTP as the webConfigurator protocol? (y/n) y

Please wait while the changes are saved to LAN...
 Reloading filter...
 Reloading routing configuration...
 DHCPD...
 Restarting webConfigurator...
```

```
The IPv4 LAN address has been set to 192.168.100.254/24
You can now access the webConfigurator by opening the following URL in your web
browser:
                http://192.168.100.254/
```

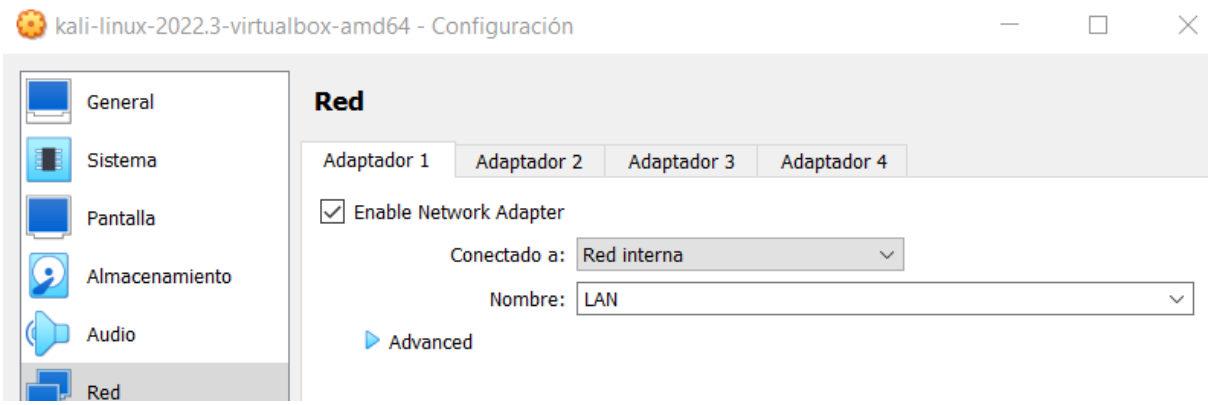
Asignación de interfaces.

Con la opción 1, asignaré las interfaces. Le diré que no a las vpn y a continuación me irá preguntando como aparece en la siguiente imagen.

```
Enter the WAN interface name or 'a' for auto-detection
(em0 em1 em2 em3 or a): em0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(em1 em2 em3 a or nothing if finished): em1

Enter the Optional 1 interface name or 'a' for auto-detection
(em2 em3 a or nothing if finished): em2

Enter the Optional 2 interface name or 'a' for auto-detection
(em3 a or nothing if finished): em3

The interfaces will be assigned as follows:

WAN  -> em0
LAN  -> em1
OPT1 -> em2
OPT2 -> em3

Do you want to proceed [y|n]? y

Writing configuration...done.
One moment while the settings are reloading... done!
```

Configuro Kali Linux en Virtual Box.

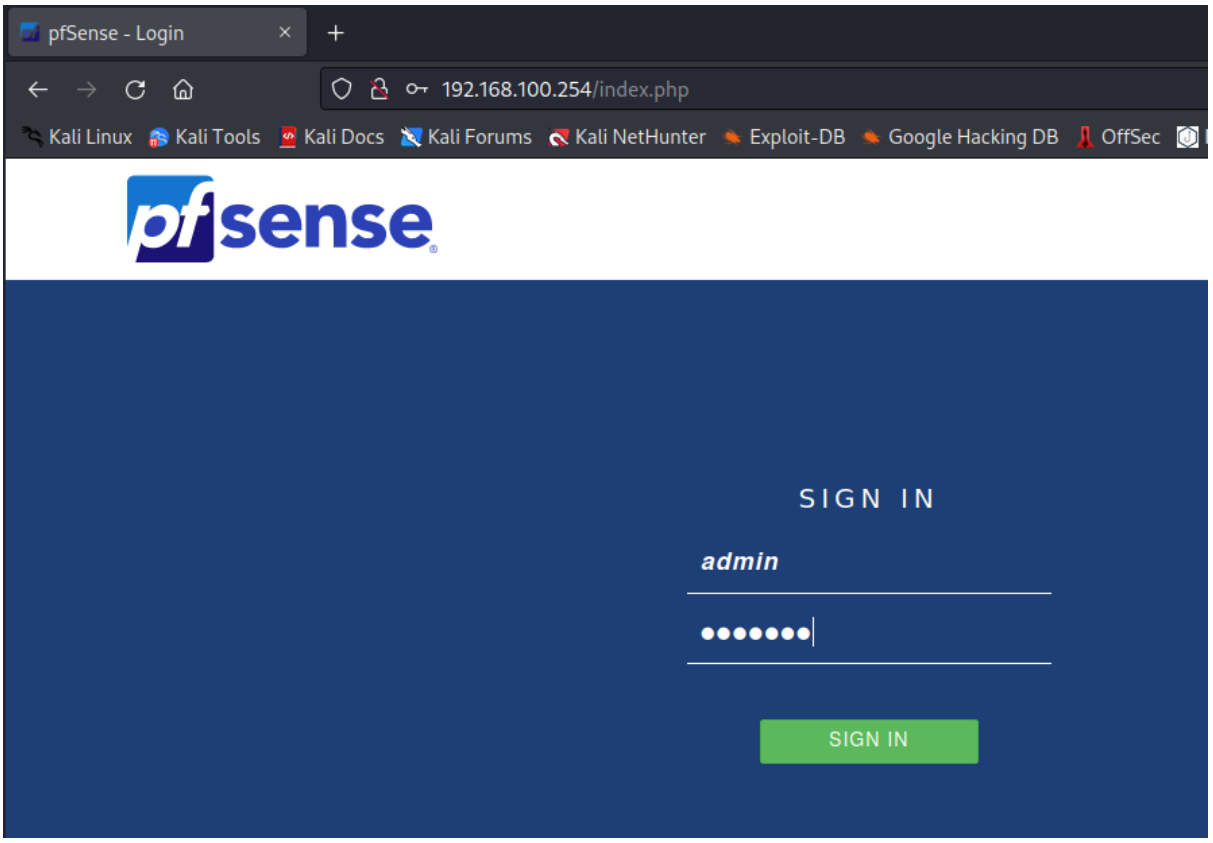
Ahora, sin apagar mi máquina de pfsense, iré al Virtual Box y selecciono mi máquina de Kali. Voy a Configuración - Red y selecciono Red interna y LAN. Esto es para que todo el tráfico pase por la red LAN.



A continuación, inicio mi máquina de Kali y abro mi consola para hacer “ifconfig”. Ahí comprobaré que mi ip sea donde comienza mi DHCP.

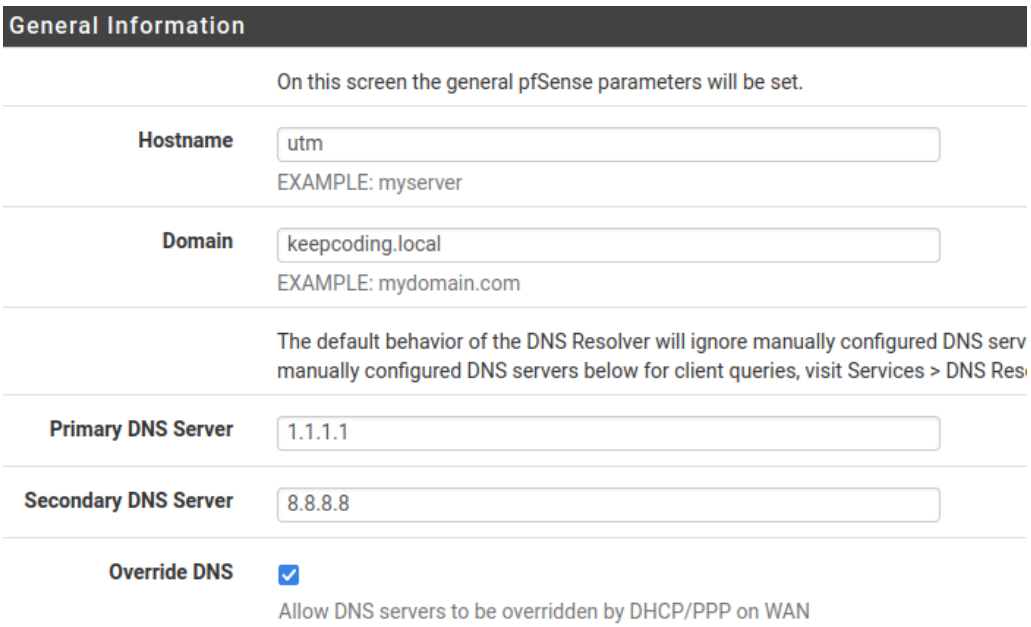
```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.100  netmask 255.255.255.0  broadcast 192.168.100.255
        inet6 fe80::c80c:3096:fa1d:101a  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:22:46:4f  txqueuelen 1000  (Ethernet)
        RX packets 1  bytes 342 (342.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 28  bytes 3434 (3.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

También inicio el pfsense en mi navegador con la dirección ip 192.168.100.254 e introduzco el usuario “admin” y la contraseña “pfsense”.

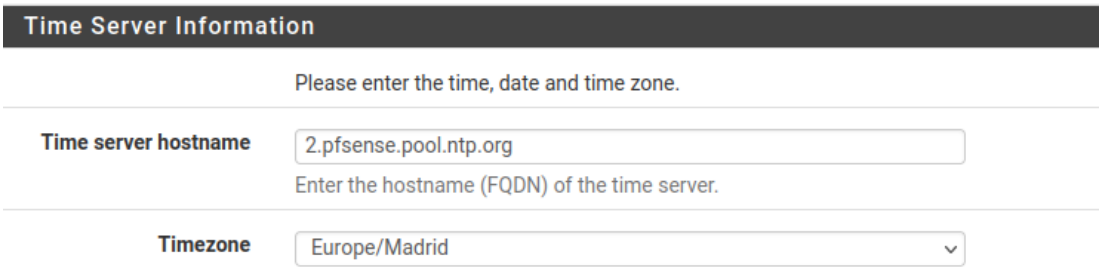


Configuración inicial en pfsense en web:

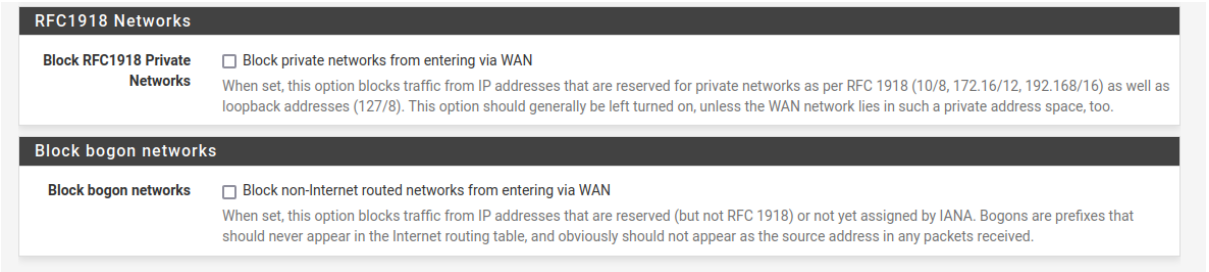
Introduzco nombre del hostname, del dominio, las dns y permito que los dns sean anulados por DHCP en Wan. Clic en Next.



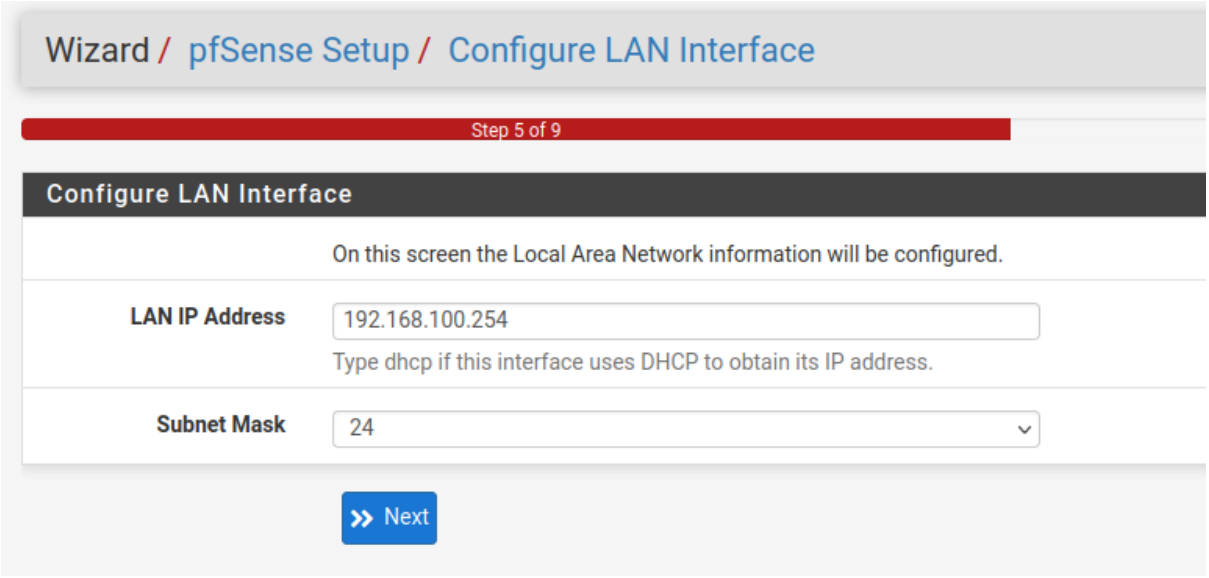
Después ponemos la zona horaria. Clic en Next



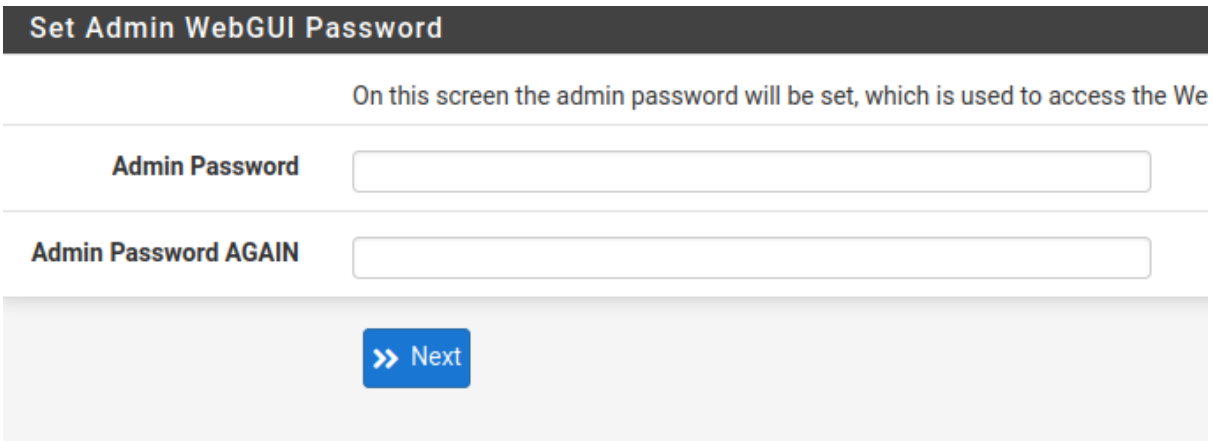
Luego configuro la interfaz WAN que la dejaré en DHCP y desbloqueo el tráfico de redes privadas y redes bogon. Lo demás lo dejo como está por defecto. Hago clic en Next.



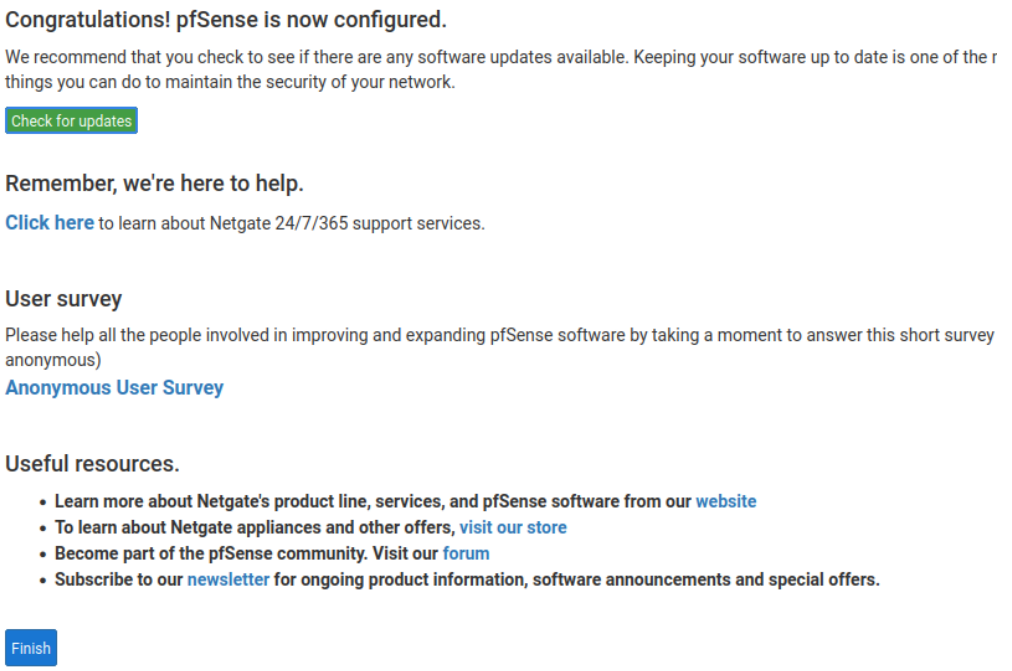
La siguiente pantalla que me sale la dejo tal cual. Clic en Next.



La siguiente pantalla me pide cambiar la contraseña, pero no la voy a cambiar, por lo que hago clic en Next.



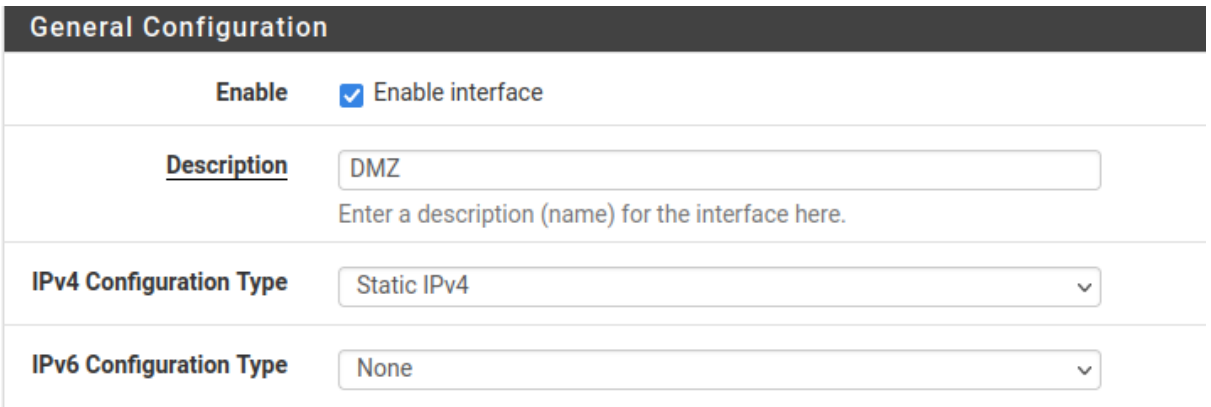
A continuación, me saldrá la última pantalla de finalización de la configuración donde haré clic en “Finished”.

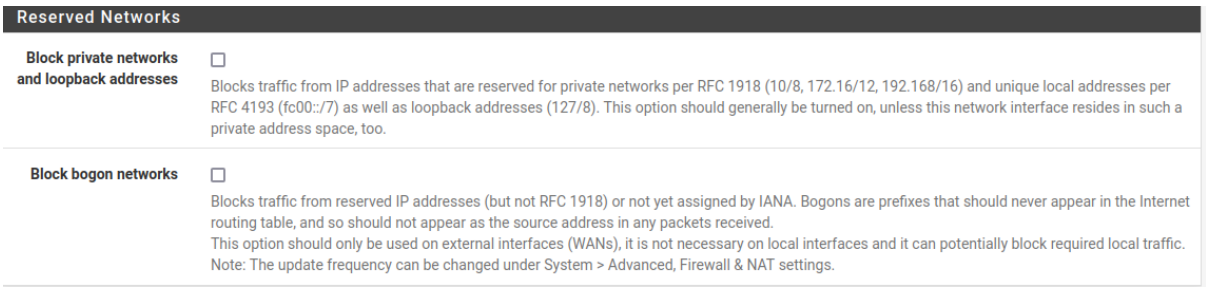


Una vez finalizado, se iniciará el pfsense y me saldrá una pantalla para aceptar el contrato de condiciones del uso del programa, el cual tendré que aceptar.

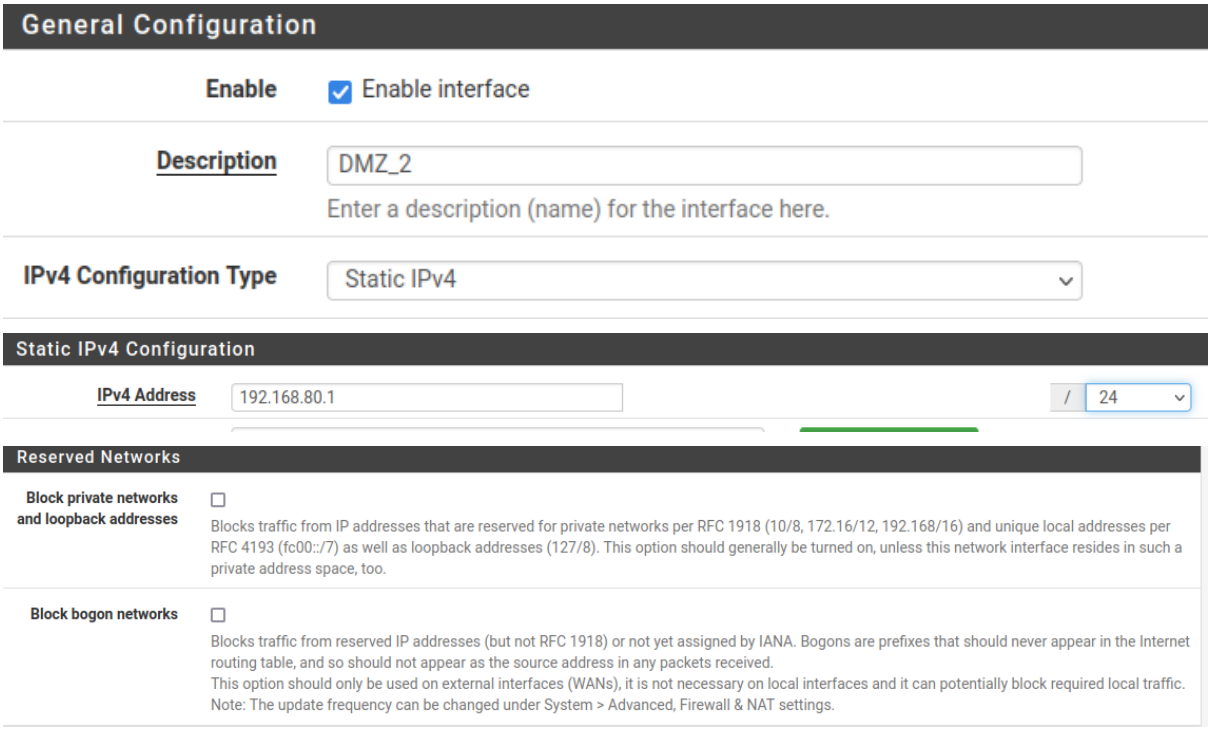
Habilitar interfaces

El siguiente paso será habilitar las redes DMZ y DMZ_2. Para ello, me voy a interfaces, a OPT1 primero. Selecciono “Enable interface”, en descripción porque el nombre de DMZ, en configuración Static IPv4 y después le pongo la ip en la configuración de la Static Ipv4, que será la 192.168.90.1 / 24. También quitaré el bloqueo de RFC 1918 y bogons. Después le damos a guardar y aplicar cambios.

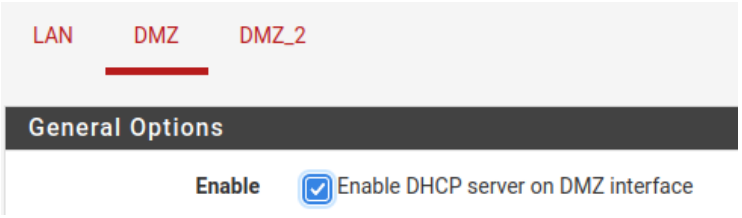


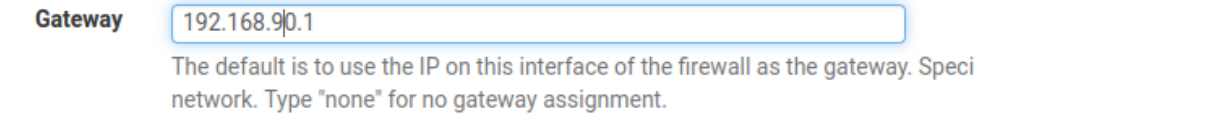


Ahora, realizaré lo mismo para DMZ_2, poniendo el nombre de DMZ_2. Para ello en interfaces me voy a OPT2 y haré el mismo procedimiento.

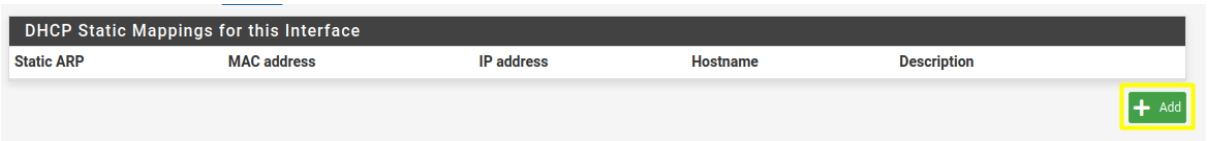


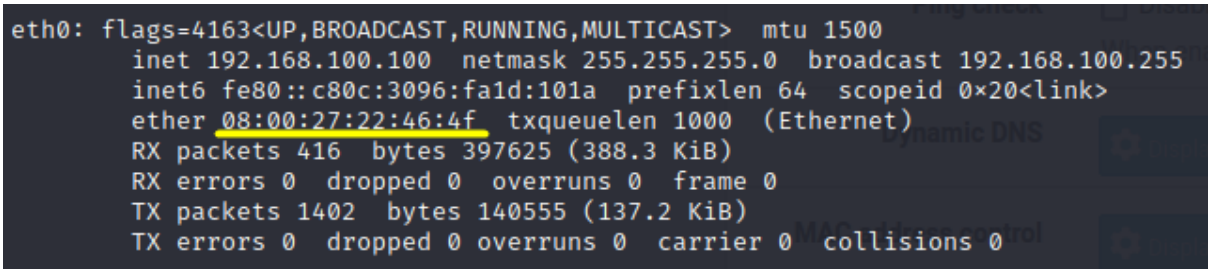
Habilitamos en DMZ y DMZ_2 el dhcp para tener ips de manera automática. Para ello, voy a Servicios - DHCP Server - DMZ y activo el DHCP server y después asigno el rango de ips que dará (para asignar estas ips concretas tendré que configurarlo con el mac de LAN). También añadiré las dns y la puerta de enlace.

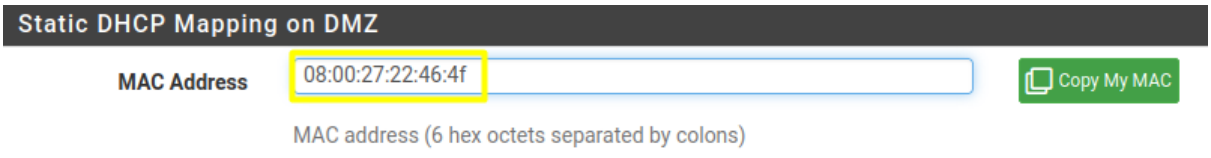




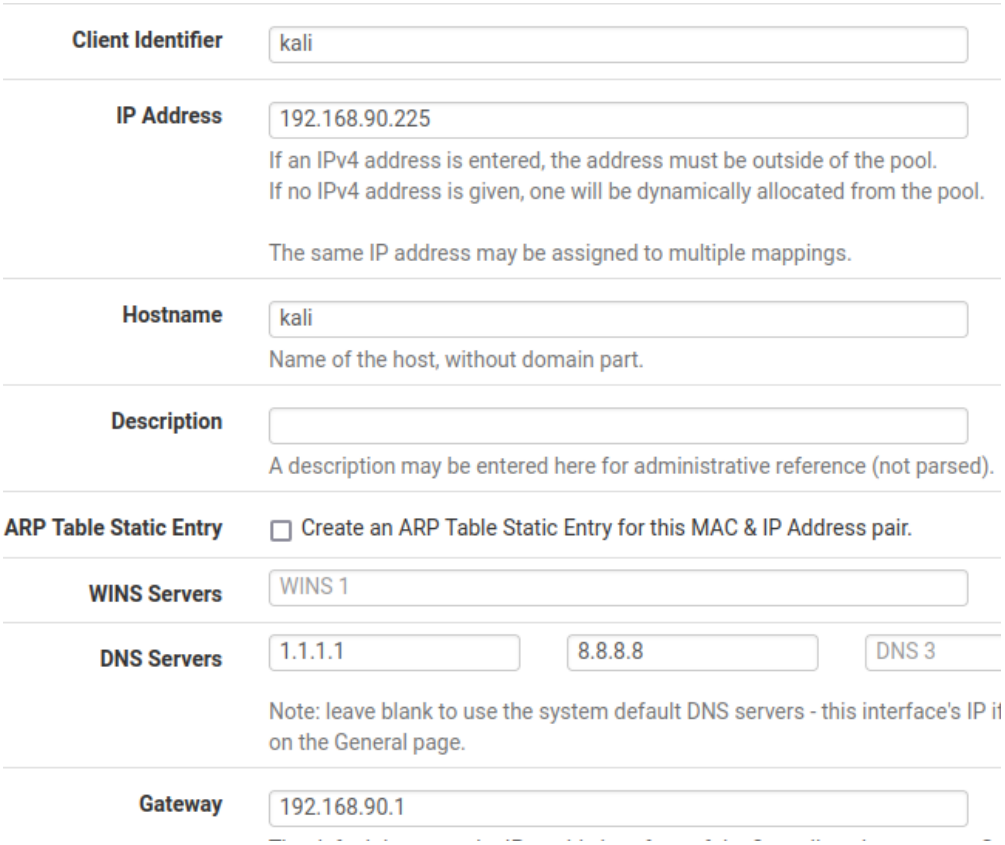
Para que las ips sean asignadas de manera concreta habrá que introducir el mac del servidor. Para ello añadiré la mac en DHCP Static Mappings. Para buscar la mac, con "ifconfig" en la consola de kali me aparecerá y la copiaré y pegaré en MAC Address.







Seguiremos rellenando datos necesarios.



Después daremos a "Guardar" y "Aplicar cambios". Ahora toca probar si funciona. Me voy Dispositivos - Red - Preferencias de red… y estando conectado a Red Interna, voy a cambiar de LAN a DMZ. Probaré lo mismo con DMZ_2. Haré "ifconfig" en la consola, y me tendrá que salir la ip de DMZ.

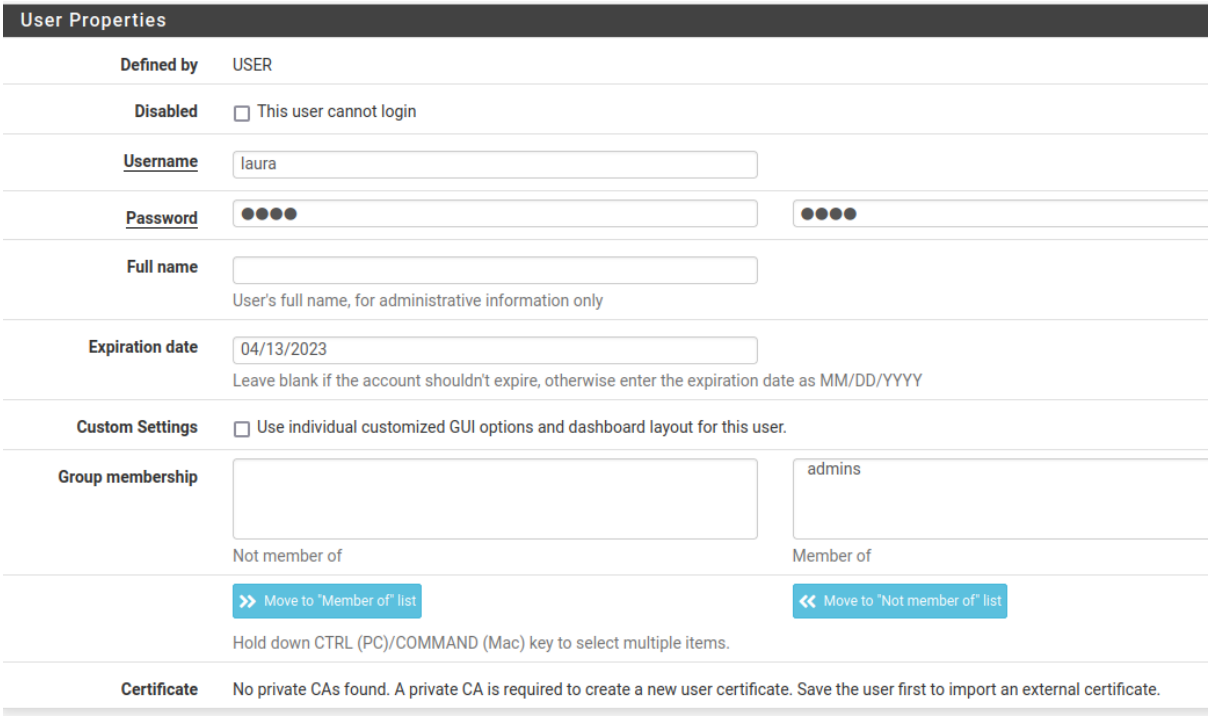
```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.90.225  netmask 255.255.255.0  broadcast 192.168.90.255
        inet6 fe80::c80c:3096:fa1d:101a  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:22:46:4f  txqueuelen 1000  (Ethernet)
        RX packets 4909  bytes 1027915 (1003.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4987  bytes 486419 (475.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

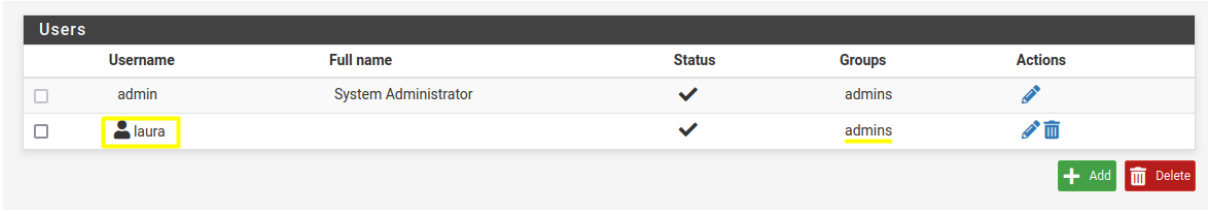
Como he podido comprobar, me ha salido la ip de DMZ.

Creación de usuario

System - User Manager - Users - +Add

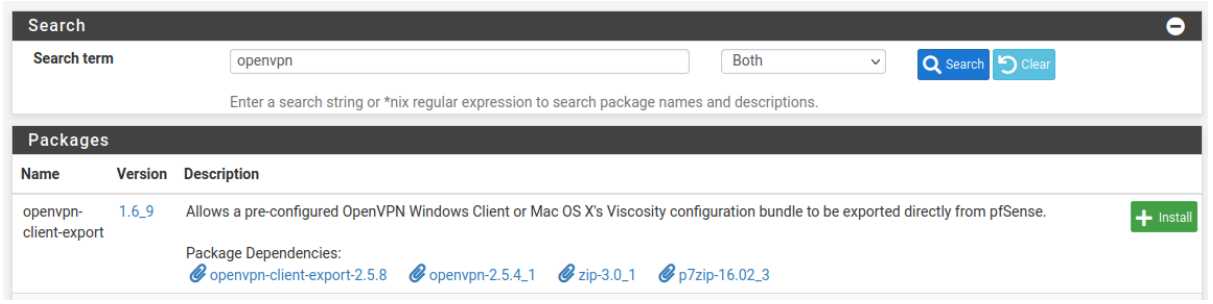
Añado un nuevo usuario. Pongo el nombre, password, fecha de expiración de la cuenta y selecciono "admins" y lo paso a la lista de "miembros de". Por último, doy a guardar.

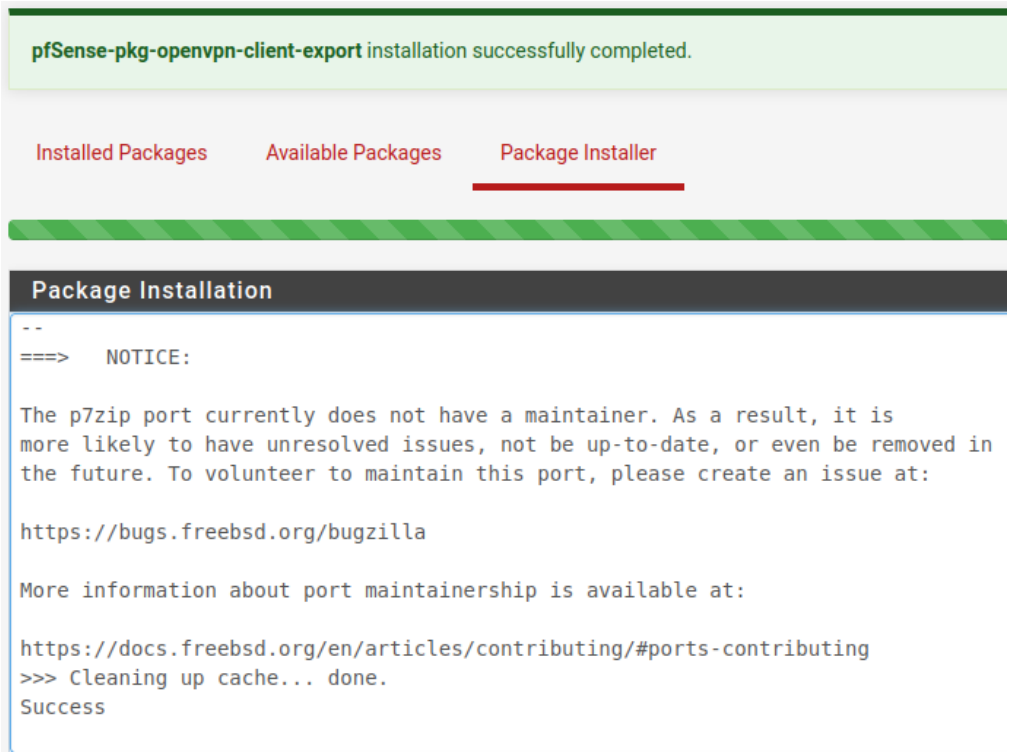




Instalar paquete openvpn-client-export.

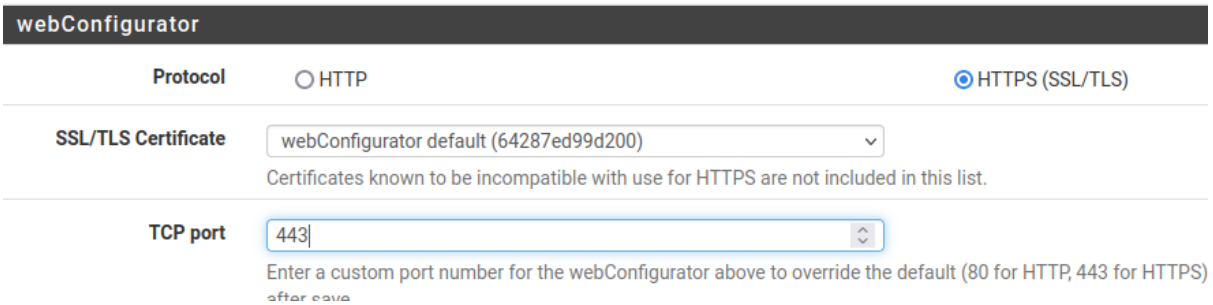
Para ello, en “System - Package manager” busco el nombre del paquete y hago clic en install.



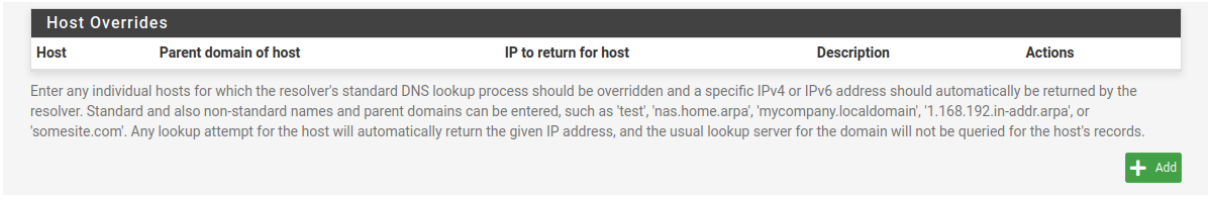


Activación de https

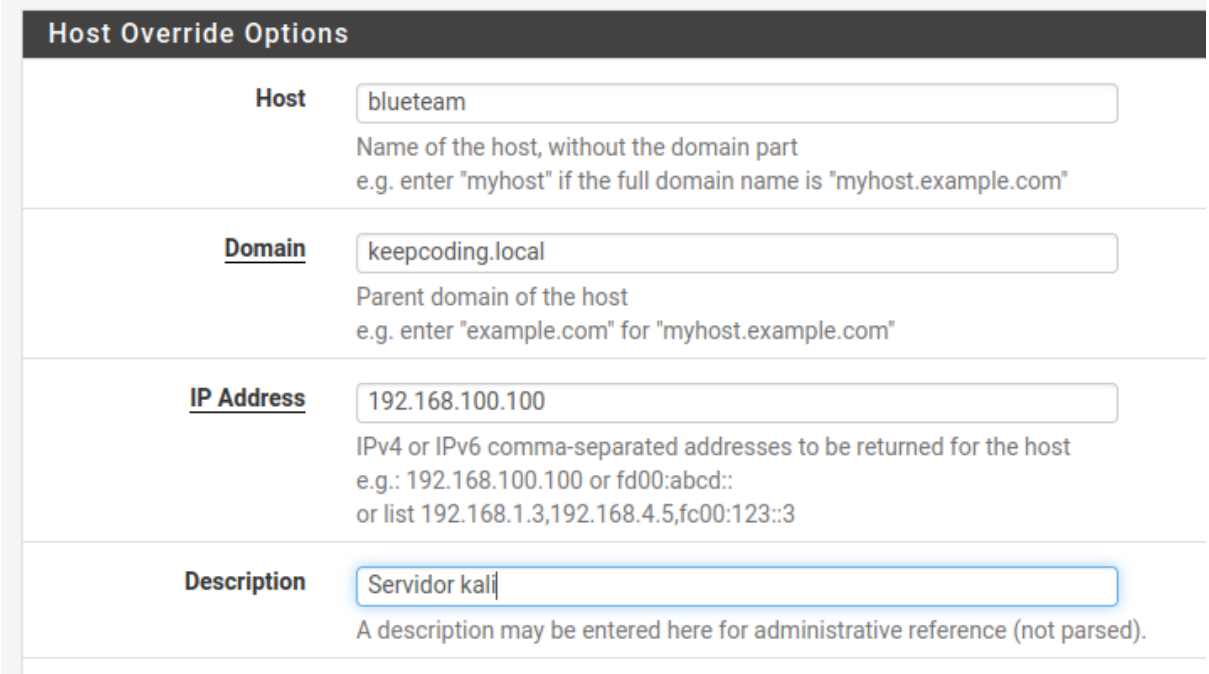
En “System - Advanced - Admin Access” activo el protocolo “https” dejando por defecto el certificado SSL/TLS que aparece y añadiendo el puerto 443. Lo demás lo dejamos por defecto como está. Hago clic en “guardar”.



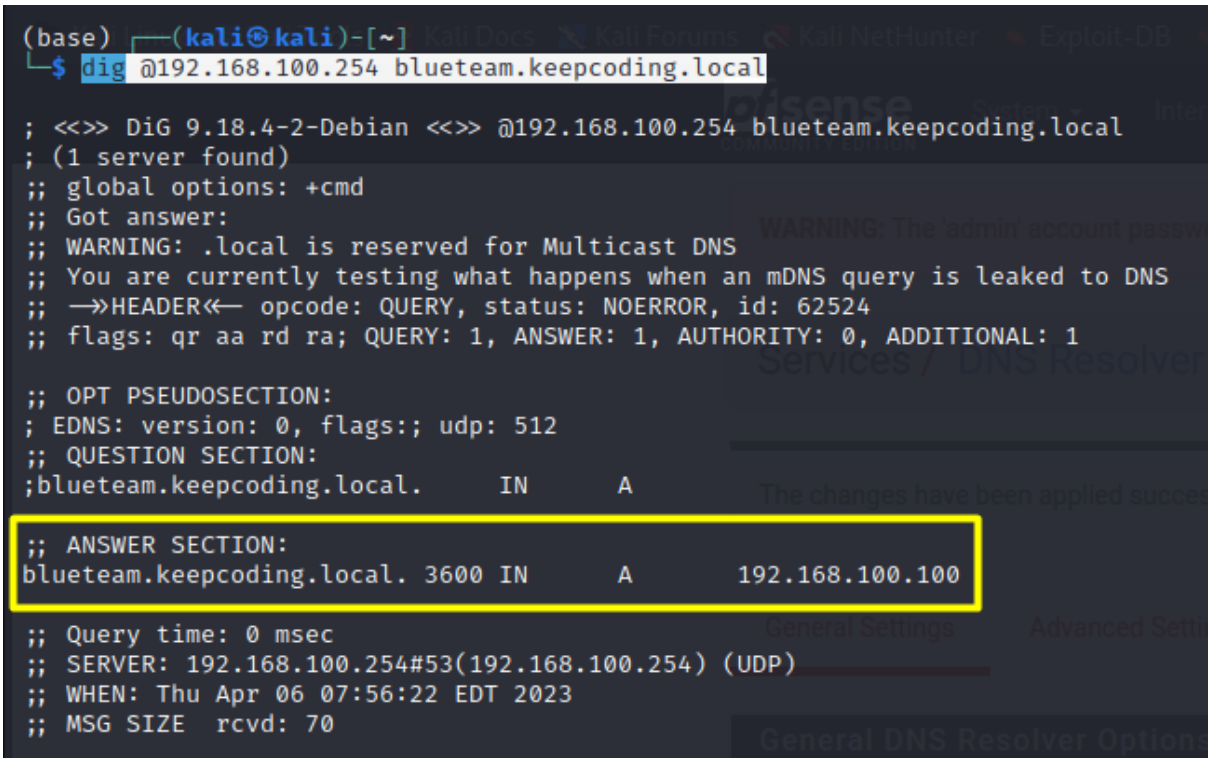
Creación en el servidor dns en un dominio que va a ser la dirección ip. En “Services - DNS Resolver - General Settings”Para añadir un servicio dns, iré a “Host Overrides” y clicaré en “Add”



Rellenaremos los datos del nombre del host, del dominio y la ip con la que se corresponde. Después hago clic en “guardar” y “aplicar cambios”.



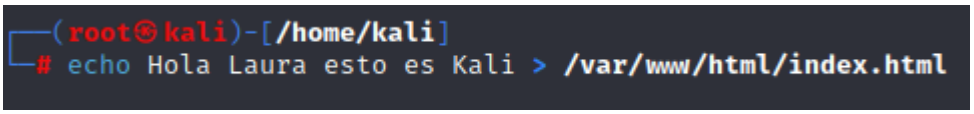
Luego voy a comprobar que la ip está asociada al dominio indicado. Para ello, en la consola haré “dig @192.168.100.254 blueteam.keepcoding.local” y me debería dar la ip 192.168.100.100.

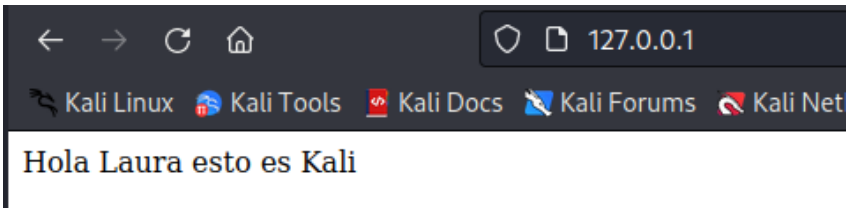


Levantar servidor apache

En la consola de Kali, inicio el servidor apache2 con “sudo service apache2 start” y me meto en la ip de apache 127.0.0.1. Después modifico la página de Apache con “echo Hola Laura

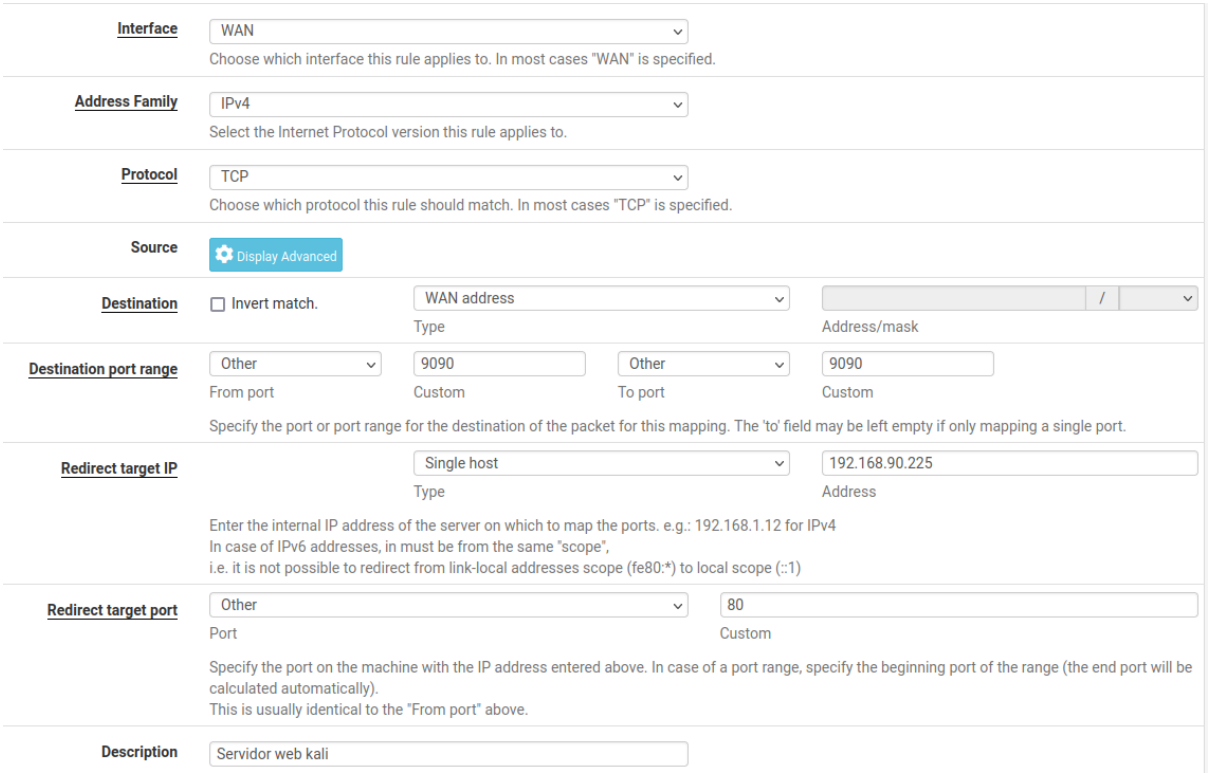
esto es Kali > /var/www/html/index.html" y actualizo la página. Como puedo comprobar, puedo acceder a mi página desde mi ordenador, pero no me puedo conectar con nadie. Para ello hay que abrir los puertos.





Abrir puertos.

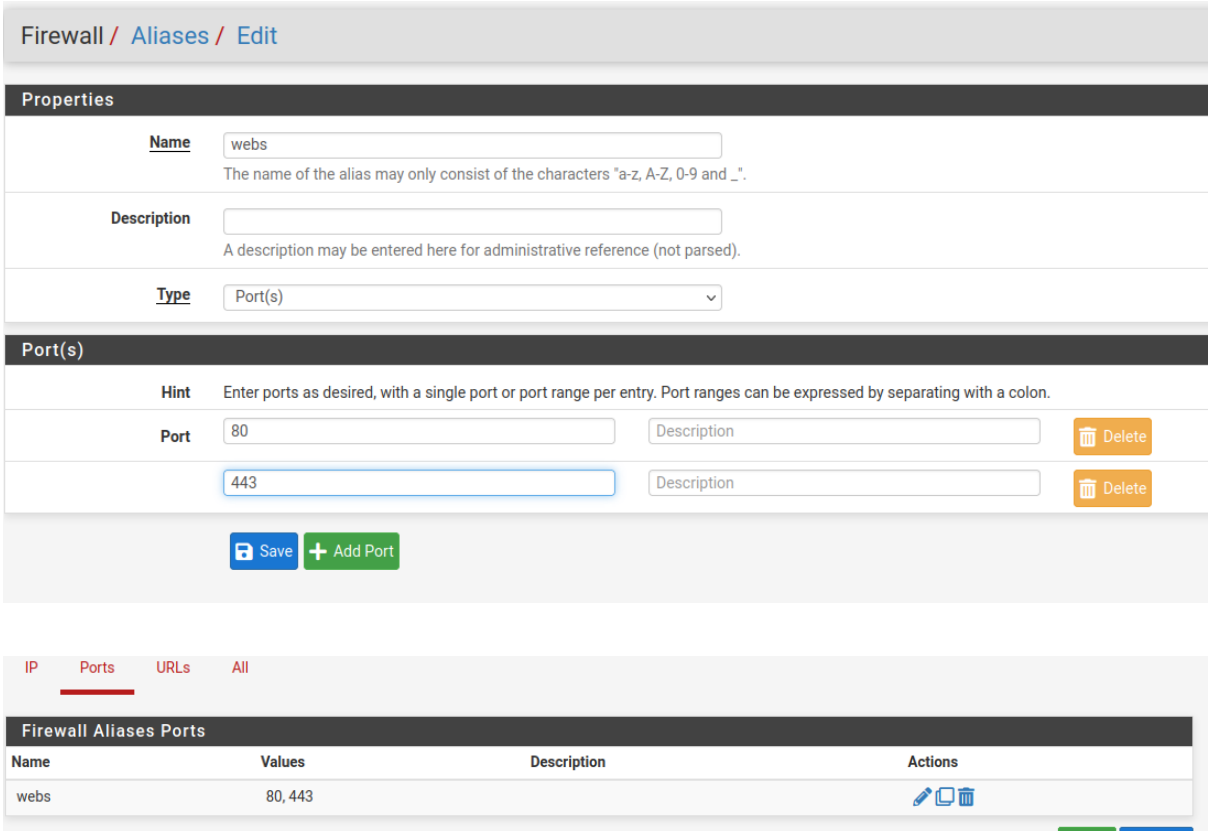
Firewall - NAT - Port Forward. Mi ip pública (wan) en ipv4 por el protocolo TCP. Siempre que vaya alguien con destino WAN (IP pública), por el puerto 9090 al 9090. lo mandaré a un equipo con host 192.168.90.225. Por el pùerto 80. Descripción "servidor web kali".



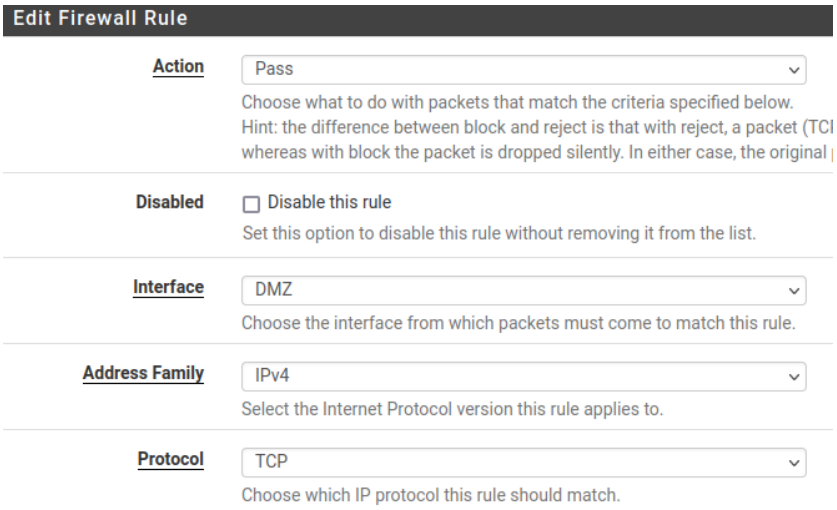
Creación de alias y creación de reglas

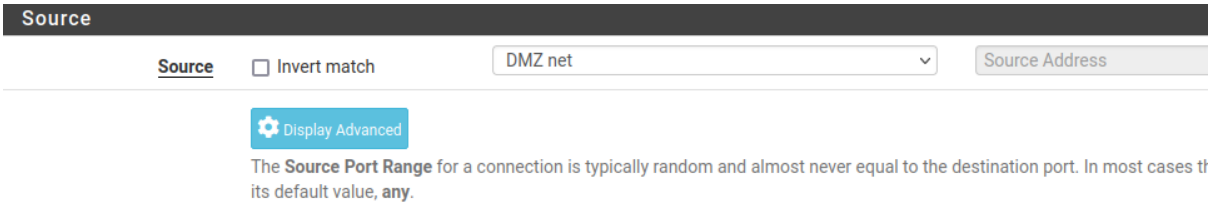
Primero crearé un alias para que me ayude en la creación de reglas. Esto me agrupará los puertos en dicho alias que cree, de manera que al tener que introducir estos datos, pondré directamente el alias.

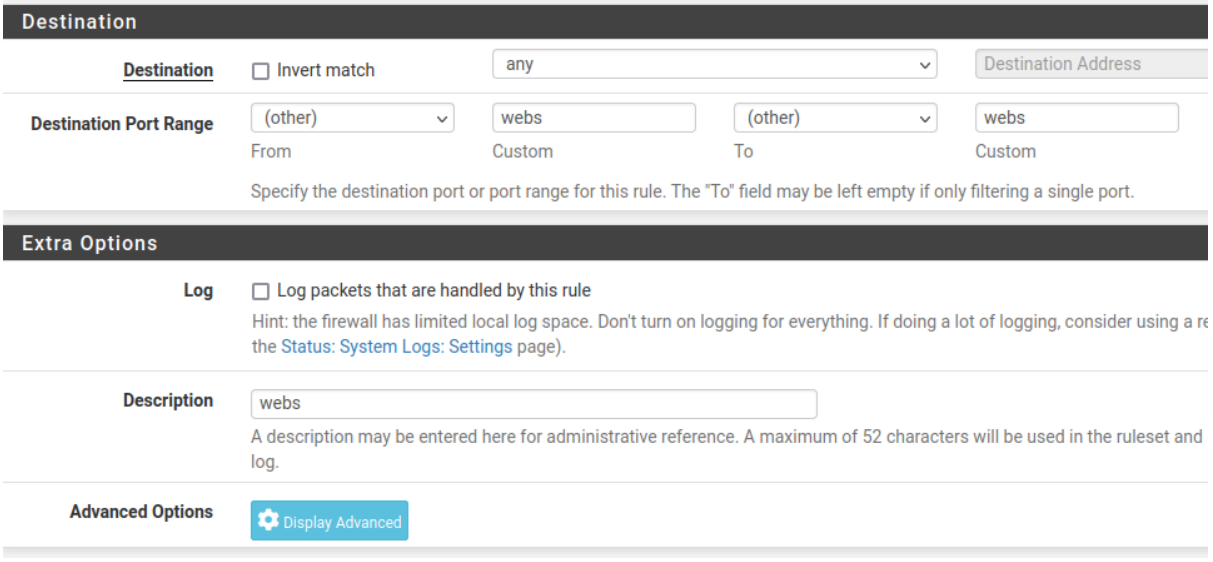
Para ello voy a Firewall - Alises - Ports - +Add. Después pongo el nombre del alias que será "webs" y añado los puertos que serán el 80 y el 443.

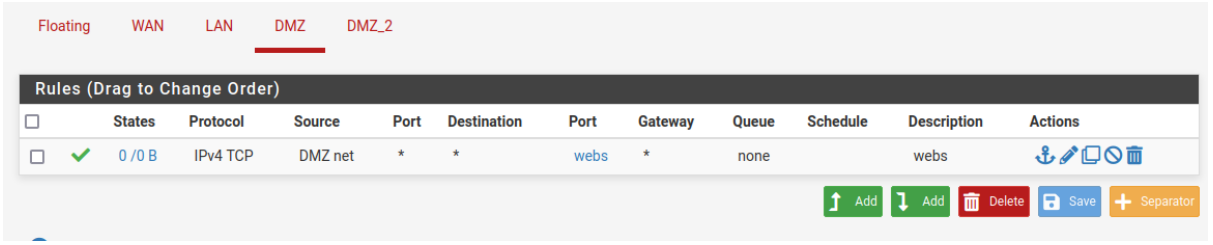


Ahora crearé las reglas. Para ello voy a Firewall - Rules - DMZ - Add. Selecciono en Action la opción “Pass”, en “Interface” dejo “DMZ” que es a la cual estamos haciendo la regla. El protocolo de internet dejo el “ipv4” y con el protocolo “TCP”. En “Source” selecciono “DMZ net” y en los puertos de destino pongo el alias creado, que será de “webs” a “webs”, es decir 80 y 443. También añadiré la descripción a la que he puesto “webs”. Para terminar esta regla, hago clic en “guardar”.

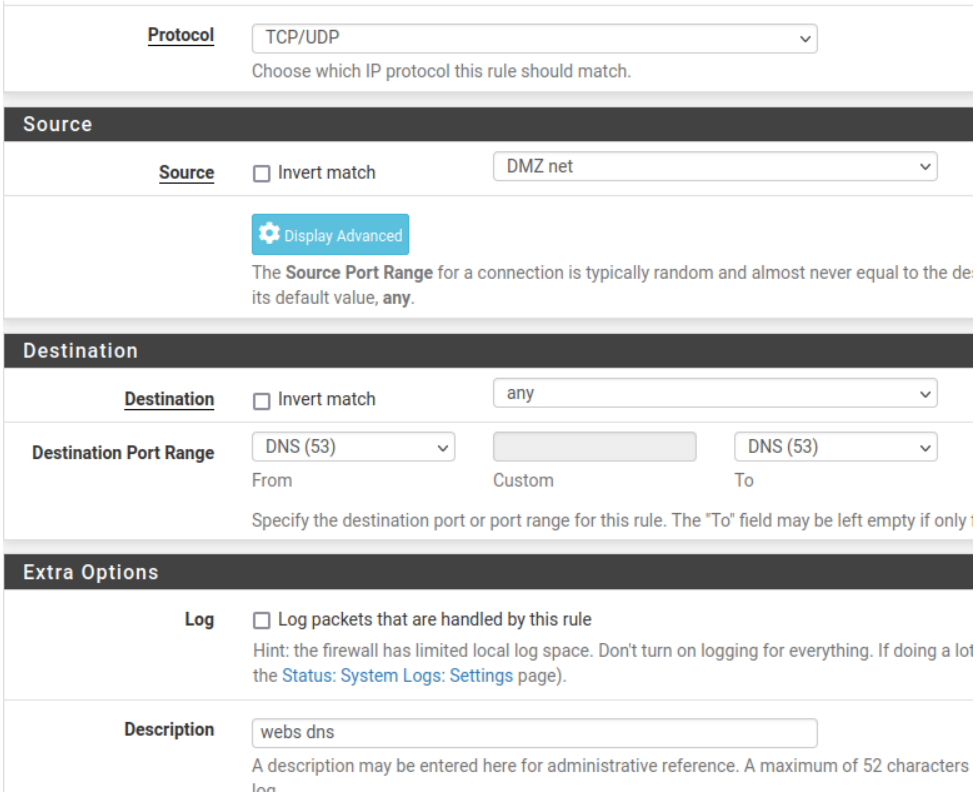


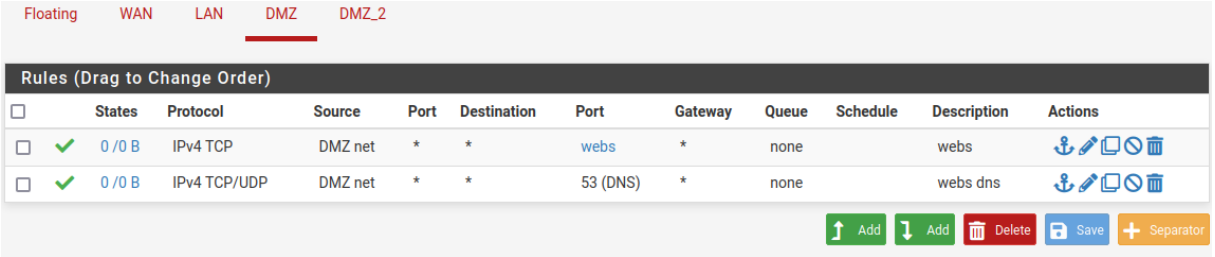






Seguidamente, creo una nueva regla, que es lo mismo pero en vez de poner “webs”, va a poner “DNS(53)”, que selecciono del listado que me aparece. Para el protocolo selecciono “TCP/UDP” y en descripción “webs dns”.

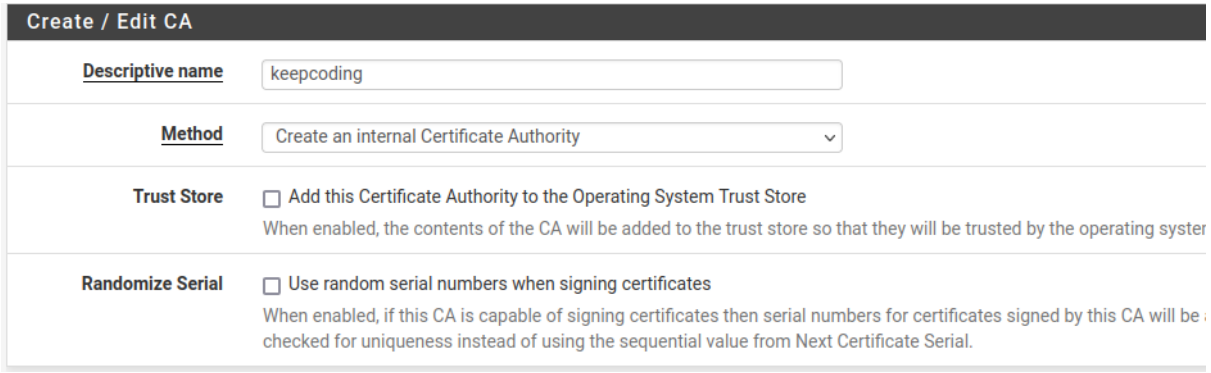


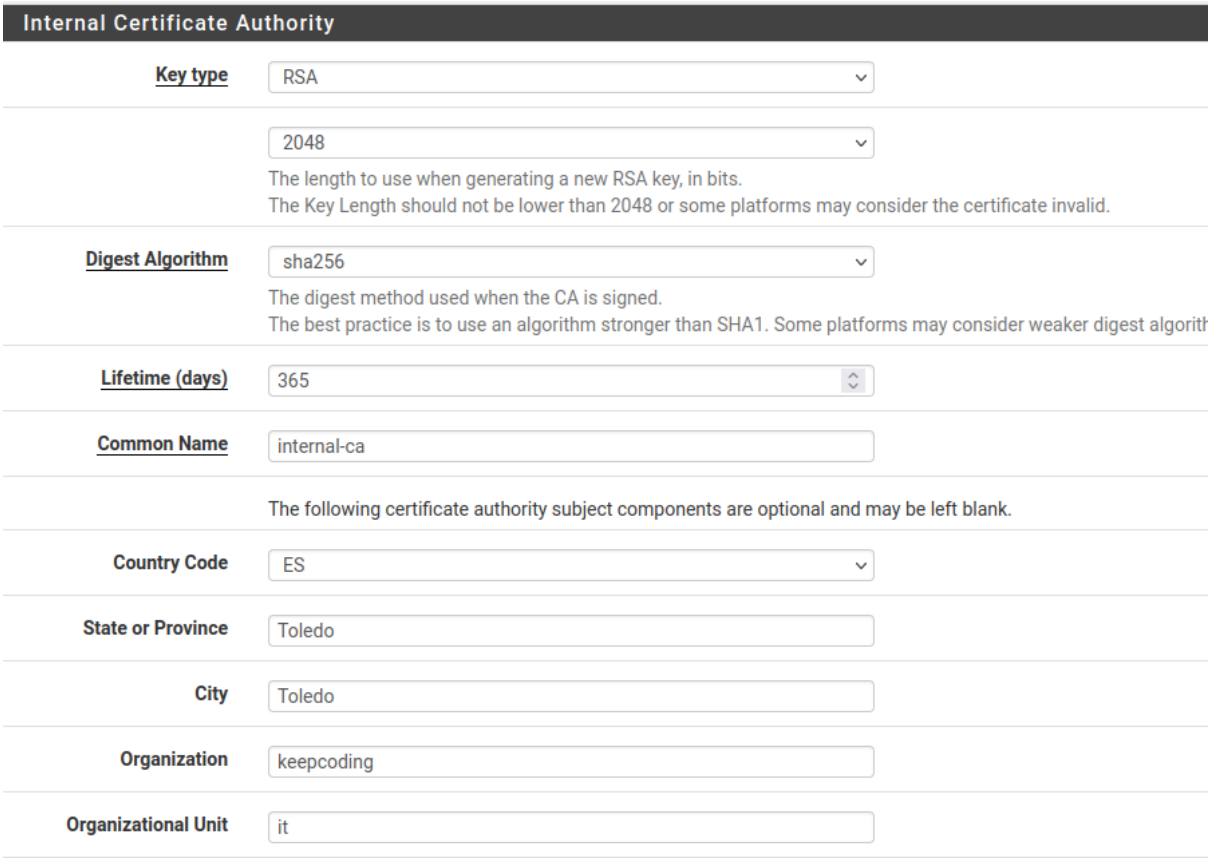


Creación de CAs

System - Certificate Manager - CAs - Add

Crearé un certificado de autenticación para poder certificar que soy yo cuando me quiera conectar a LAN, DMZ o DMZ_2.

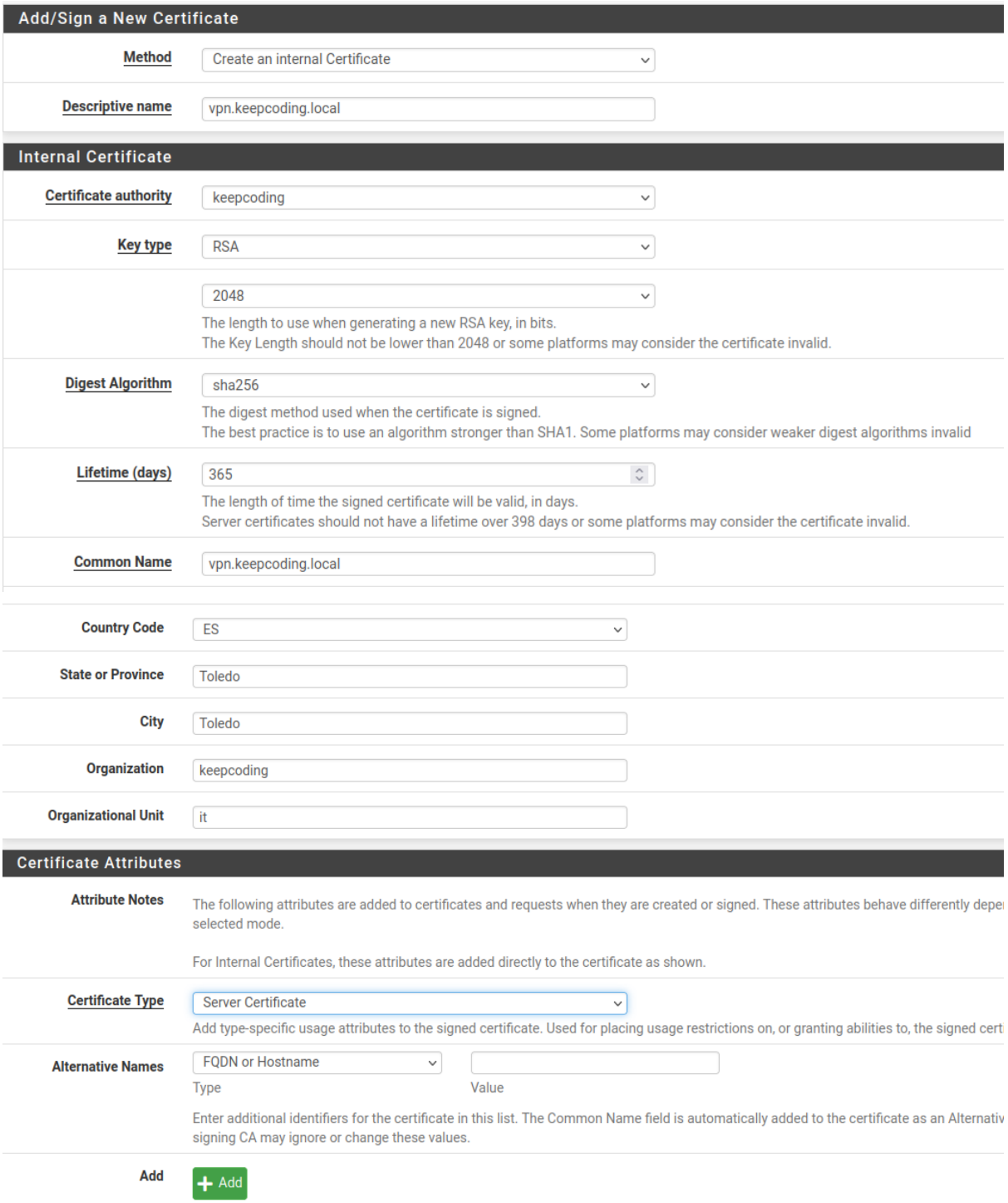




Creación de certificados

System - Certificate Manager - Certificates - Add

Crearé un certificado de servidor para la conexión a través de las vpn.

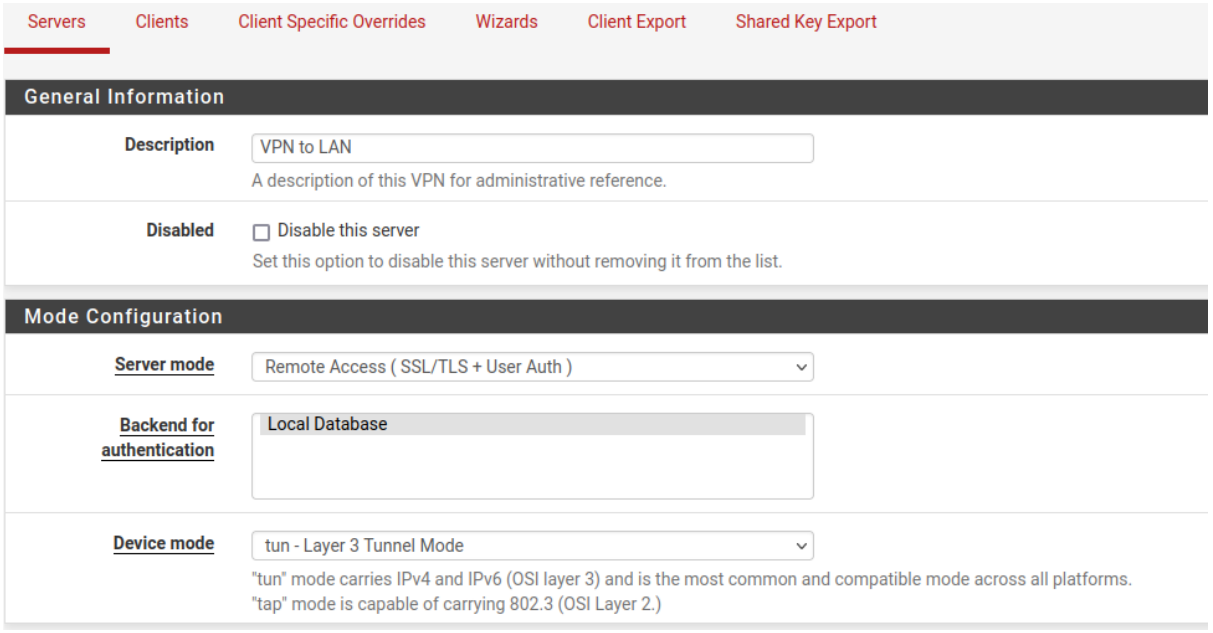


Creación servidor vpn

VPN - Open VPN - Server - Add

Crearé un servidor VPN que me servirá para conectarme desde mi WAN a LAN, DMZ y DMZ_2. Para ello en descripción pongo VPN to LAN, en “server mode” selecciono “SSL/TLS+User Auth” para que conecte de sede a sede pero con autenticación de usuario,

lo que requerirá un certificado por parte del usuario. En “Device Mode” elijo el modo layer3, que es la entrada al router que nos conecta al tunel y nos dará acceso tanto a LAN como a DMZ y DMZ_2. El protocolo por el que se hará será el TCP. En “server certificate” selecciono el creado de “vpn.keepcoding.local” que fue creado como certificado de servidor. En “Hardware crypto” elijo mi tarjeta “INTEL RDRAND engine - RAND”. Añadiré la ip del tunel y los rangos de ips que serán accesibles como punto remoto. Por último, doy a guardar.



Endpoint Configuration

Protocol: TCP on IPv4 only

Interface: WAN

The interface or Virtual IP address where OpenVPN will receive client connections.

Local port: 1194

The port used by OpenVPN to receive client connections.

Cryptographic Settings

TLS Configuration: ☑ Use a TLS Key

A TLS key enhances security of an OpenVPN connection by requiring both parties to have a common key before a peer can
This layer of HMAC authentication allows control channel packets without the proper key to be dropped, protecting the pee
unauthorized connections.The TLS Key does not have any effect on tunnel data.

☑ Automatically generate a TLS Key.

Peer Certificate Authority: keepcoding

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: vpn.keepcoding.local (Server: Yes, CA: keepcoding)

DH Parameter Length: 2048 bit

Diffie-Hellman (DH) parameter set used for key exchange.

ECDH Curve: Use Default

Data Encryption Negotiation: ☑ Enable Data Encryption Negotiation

This option allows OpenVPN clients and servers to negotiate a compatible set of acceptable cryptographic data encryption algo
selected in the Data Encryption Algorithms list below. Disabling this feature is deprecated.

Data Encryption Algorithms

Available Data Encryption Algorithms	Allowed Data Encryption Algorithms
AES-256-CBC (256 bit key, 128 bit block) AES-256-CFB (256 bit key, 128 bit block) AES-256-CFB1 (256 bit key, 128 bit block) AES-256-CFB8 (256 bit key, 128 bit block) AES-256-GCM (256 bit key, 128 bit block) AES-256-OFB (256 bit key, 128 bit block) ARIA-128-CBC (128 bit key, 128 bit block) ARIA-128-CFB (128 bit key, 128 bit block) ARIA-128-CFB1 (128 bit key, 128 bit block) ARIA-128-CFB8 (128 bit key, 128 bit block)	AES-256-GCM AES-128-GCM CHACHA20-POLY1305
Click to add or remove an algorithm from the list	Allowed Data Encryption Algorithms. Click an alg it from the list

The order of the selected Data Encryption Algorithms is respected by OpenVPN. This list is ignored in Shared Key mode.

Fallback Data Encryption Algorithm: 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 da
negotiation (e.g. Shared Key). This algorithm is automatically included in the Data Encryption Algorithms list.

Auth digest algorithm: 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 da
The server and all clients must have the same setting. While SHA1 is the default for OpenVPN, this algorithm is insecure.

Hardware Crypto: Intel RDRAND engine - RAND

Certificate Depth: One (Client+Server)

When a certificate-based client logs in, do not accept certificates below this depth. Useful for denying certificates made with int
generated from the same CA as the server.

Strict User-CN Matching: ☐ Enforce match

Client Certificate Key Usage Validation

☑ Enforce key usage

Verify that only hosts with a client certificate can connect (EKU: "TLS Web Client Authentication").

Tunnel Settings

IPv4 Tunnel Network

192.168.225.0/24

This is the IPv4 virtual network or network type alias with a single entry used for private communications between this server and cl
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 interf
usable addresses will be assigned to connecting clients.

IPv6 Tunnel Network

This is the IPv6 virtual network or network type alias with a single entry used for private communications between this server and cl
expressed using CIDR notation (e.g. fe80::/64). The ::1 address in the network will be assigned to the server virtual interface. The re
will be assigned to connecting clients.

Redirect IPv4 Gateway

☐ Force all client-generated IPv4 traffic through the tunnel.

Redirect IPv6 Gateway

☐ Force all client-generated IPv6 traffic through the tunnel.

IPv4 Local network(s)

192.168.90.0/24, 192.168.80.0/24, 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
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 gen
LAN network.

IPv6 Local network(s)

IPv6 networks that will be accessible from the remote endpoint. Expressed as a comma-separated list of one or more IP/PREFIX or
aliases. This may be left blank if not adding a route to the local network through this tunnel on the remote machine. This is generall
network.

Concurrent connections

Allow Compression

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 com
VPN (e.g. HTTP). Before enabling compression, consult information about the VORACLE, CRIME, TIME, and BREACH
the use case for this specific VPN is vulnerable to attack.

Asymmetric compression allows an easier transition when connecting with older peers.

Push Compression

☐ Push the selected Compression setting to connecting clients.

Type-of-Service

☐ Set the TOS IP header value of tunnel packets to match the encapsulated packet value.

Inter-client communication

☐ Allow communication between clients connected to this server

Duplicate Connection

☐ Allow multiple concurrent connections from the same user

When set, the same user may connect multiple times. When unset, a new connection from a user will disconnect the

Users are identified by their username or certificate properties, depending on the VPN configuration. This practice is
may be necessary in some environments.

Client Settings

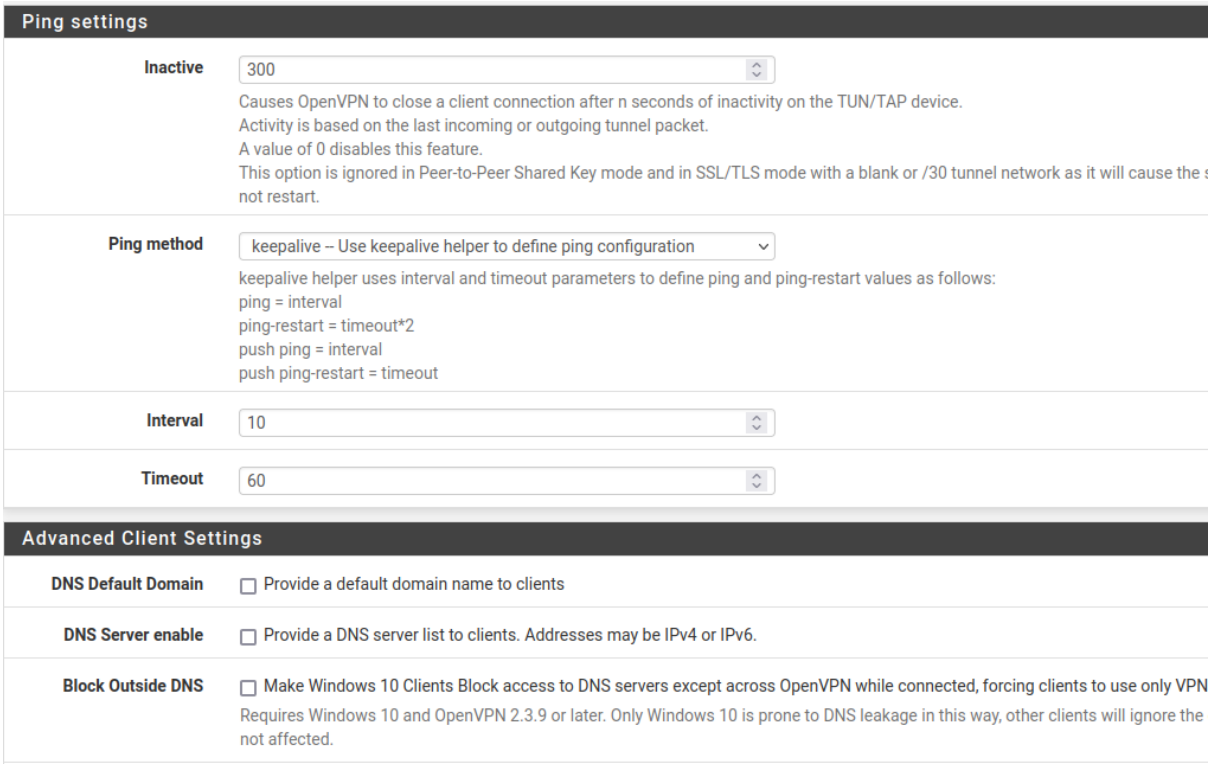
Dynamic IP

☐ Allow connected clients to retain their connections if their IP address changes.

Topology

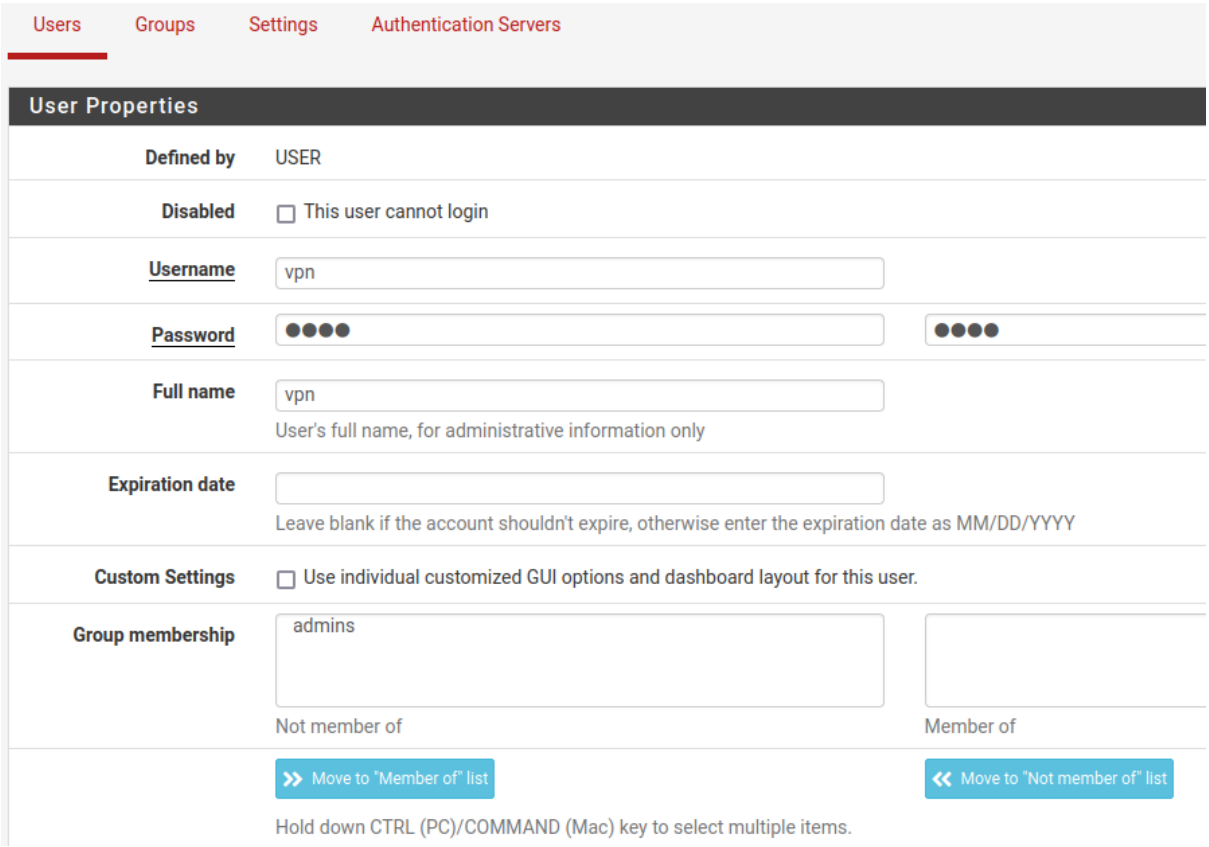
Subnet -- One IP address per client in a common subnet

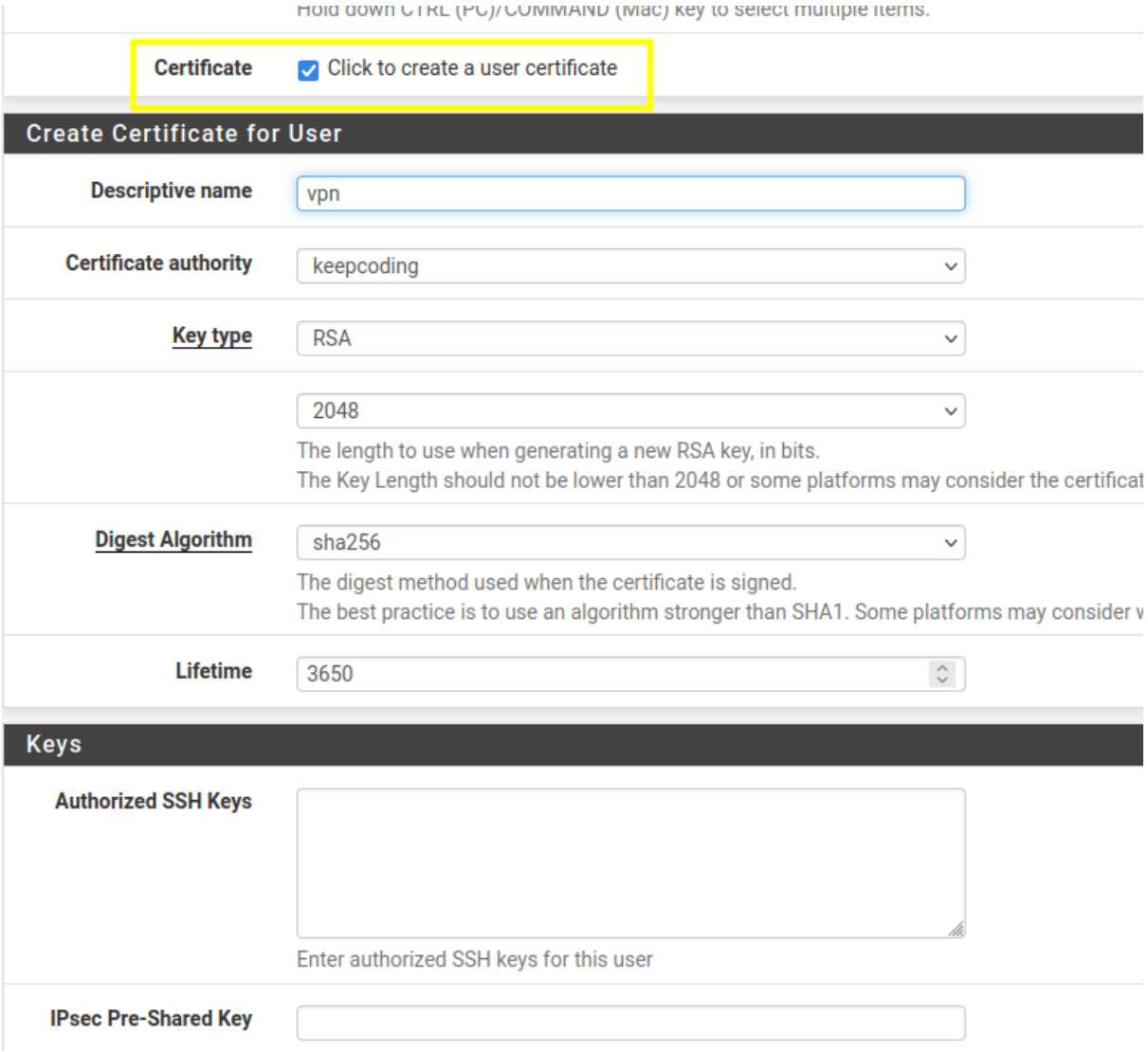
Specifies the method used to supply a virtual adapter IP address to clients when using TUN mode on IPv4.
Some clients may require this be set to "subnet" even for IPv6, such as OpenVPN Connect (iOS/Android). Older versio
clients such as Yealink phones may require "net30".

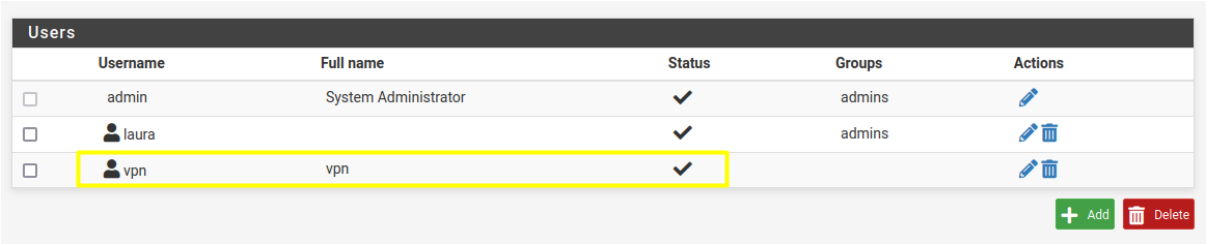


Crear usuarios

En “System - User Manager - Users - Add”. Pondré el nombre de usuario y contraseña (vpn:1234). En “Fullname” pondré vpn y después haré clic en “Crear certificado”. Para terminar pondré el “nombre de descripción” que será “vpn” y la información que aparece a continuación la dejaré tal y como sale por defecto. Para terminar la creación de usuario daré a guardar.


Users
Groups
Settings
Authentication Servers
User Properties
Defined by
USER
Disabled
This user cannot login
Username
vpn
Password
Full name
vpn
User's full name, for administrative information only
Expiration date
Leave blank if the account shouldn't expire, otherwise enter the expiration date as MM/DD/YYYY
Custom Settings
Use individual customized GUI options and dashboard layout for this user.
Group membership
admins
Not member of
Member of
Move to "Member of" list
Move to "Not member of" list
Hold down CTRL (PC)/COMMAND (Mac) key to select multiple items.



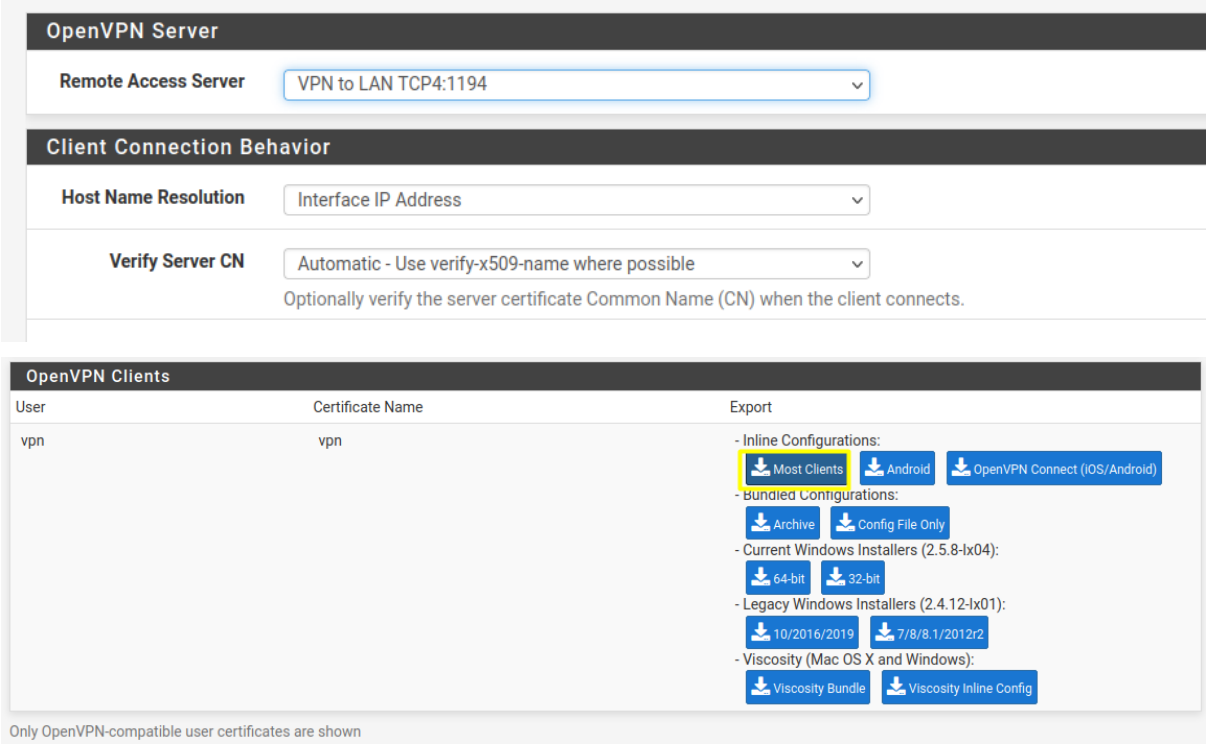


Exportar cliente

En “VPN - OpenVPN - Client Export”. Como ya tenemos certificado y usuario y contraseña, ahora hay que exportar el cliente. Con el plugin de “vpn client export” instalado exportaré el cliente.

Una vez dentro de “VPN - OpenVPN - Client Export” primero seleccionaré la vpn que vamos a usar que es la que me sale por defecto. Después seleccionaré la ip de WAN (Interface IP

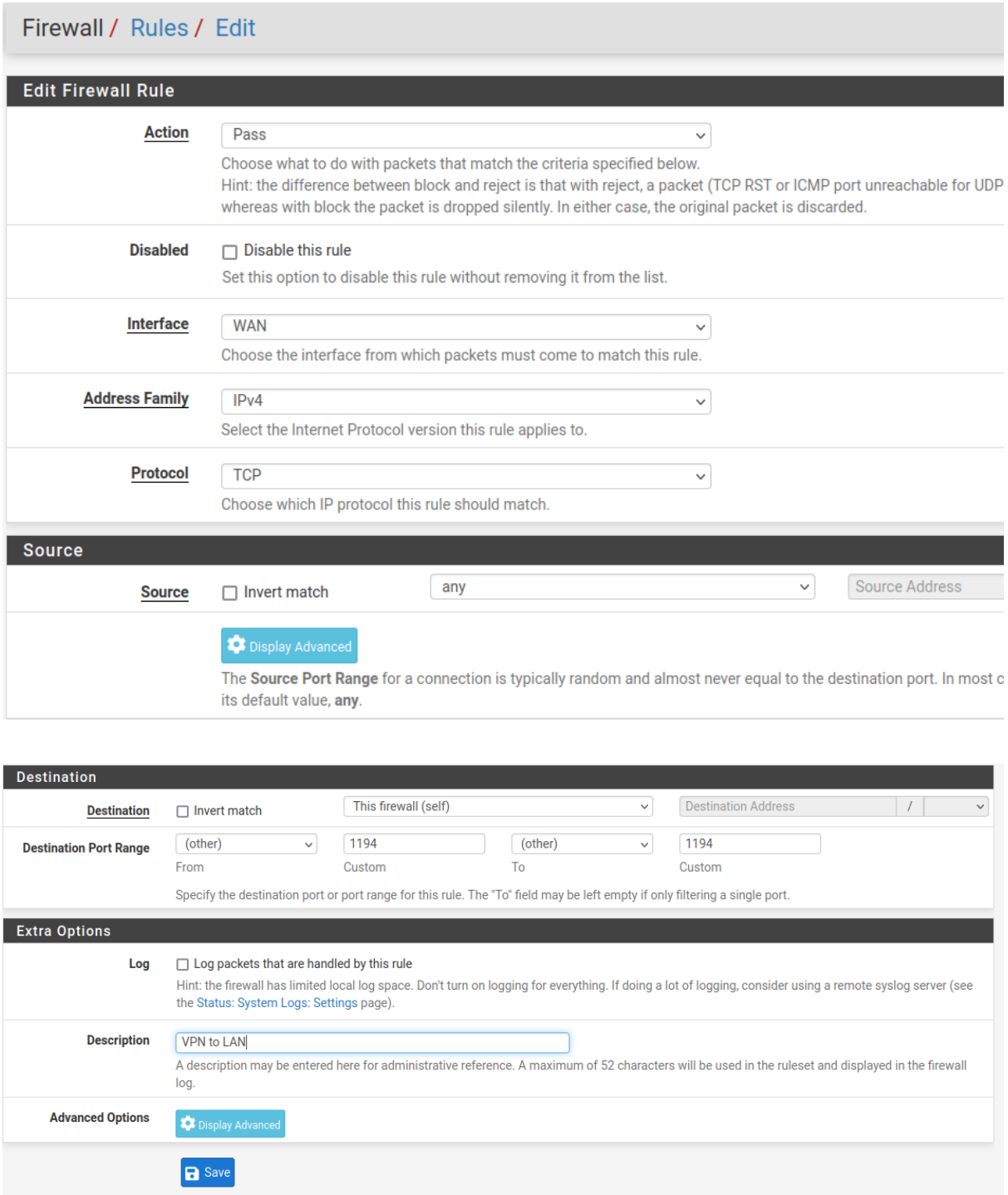
address). En la parte de “OpenVPN clients” seleccionaré “Most clients” y descargaré el certificado. Lo demás lo dejo por defecto.



Una vez descargado lo copio y pego en mi servidor local, para poder utilizarlo en la conexión con LAN.

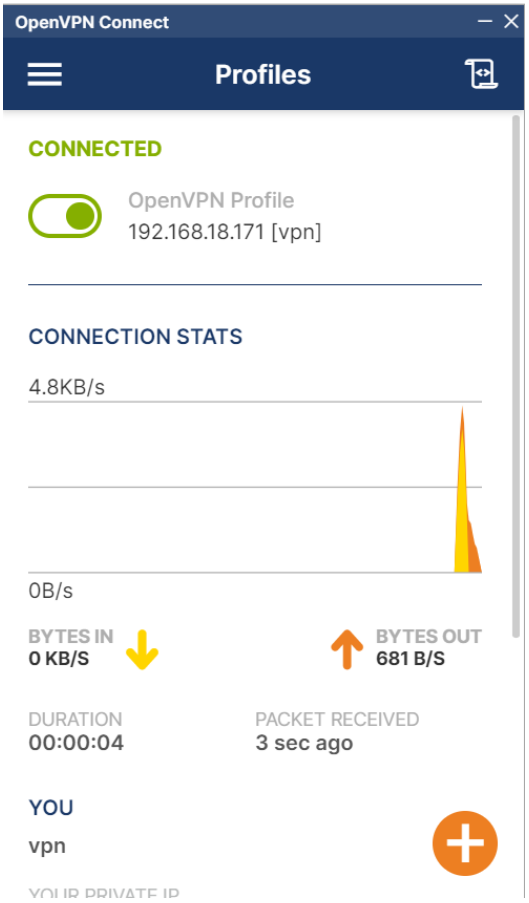
Creación de una nueva regla

En “Firewall - Rules - WAN - Add” crearé una nueva regla para que me permita la conexión VPN en WAN. A esta nueva regla le pediré que vaya a WAN por Ipv4, por el protocolo UDP, con origen a cualquier sitio y con destino a nuestro Firewall. Pongo el puerto seleccionado que es el 1194. Para terminar guardo y aplico cambios.



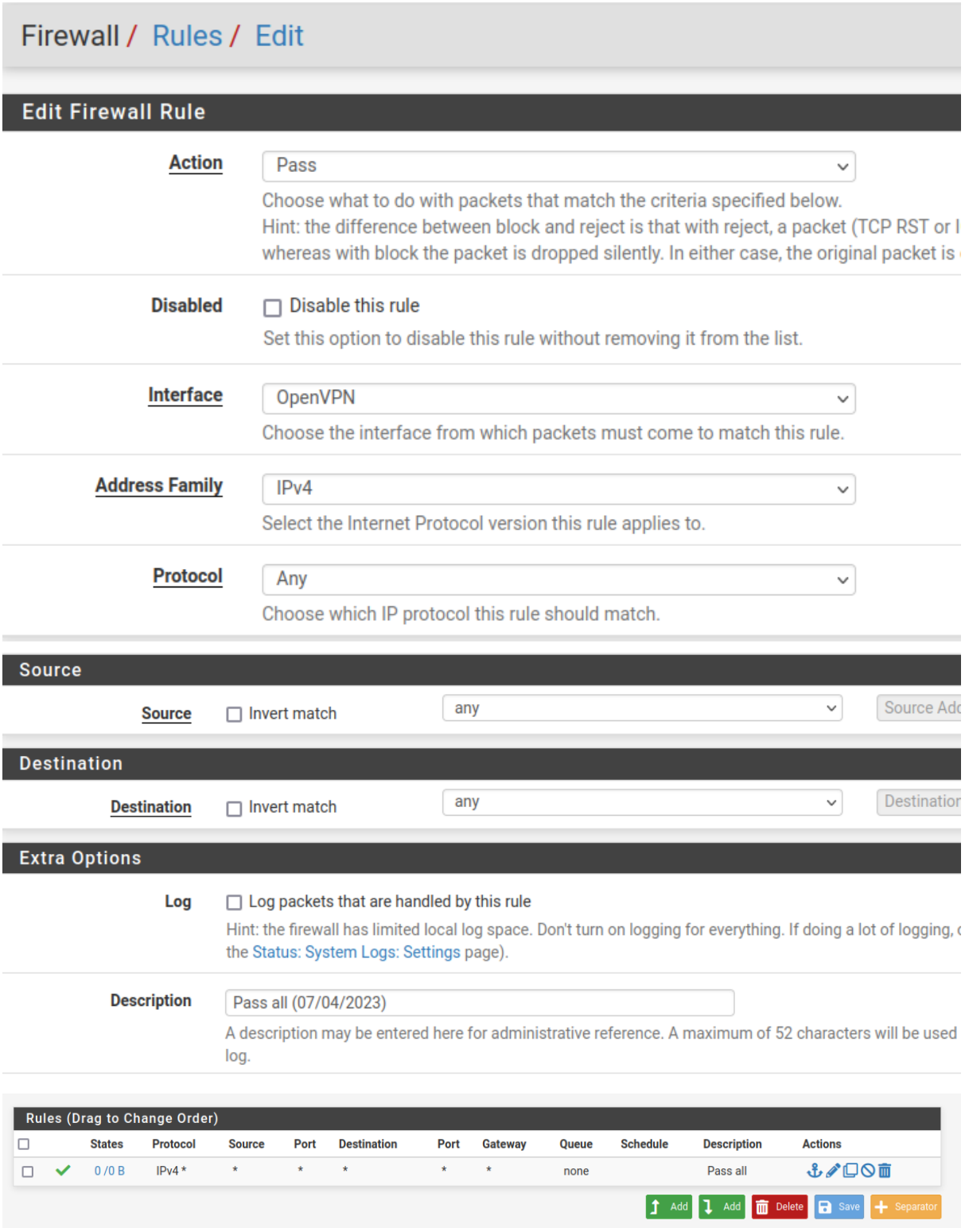
Compruebo conexión.

Con OpenVPN Connect subiré el certificado descargado previamente y pondré el nombre de usuario y contraseña que asigné en la creación de usuario (vpn:1234). Puedo comprobar que tengo conexión.



Creo nueva regla para acceder al apache desde WAN a través de VPN.

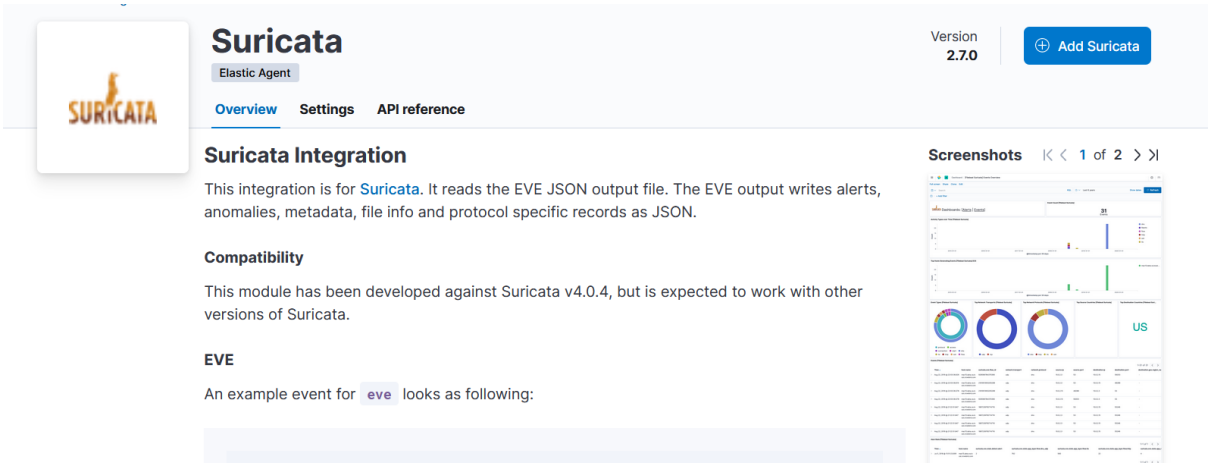
En “Firewall - Rules - OpenVPN - Add” creo regla que permita pasar la interfaz openvpn por cualquier protocolo, a cualquier sitio. En descripción, pondré “Pass all”. Para terminar, guardo y aplico cambios.



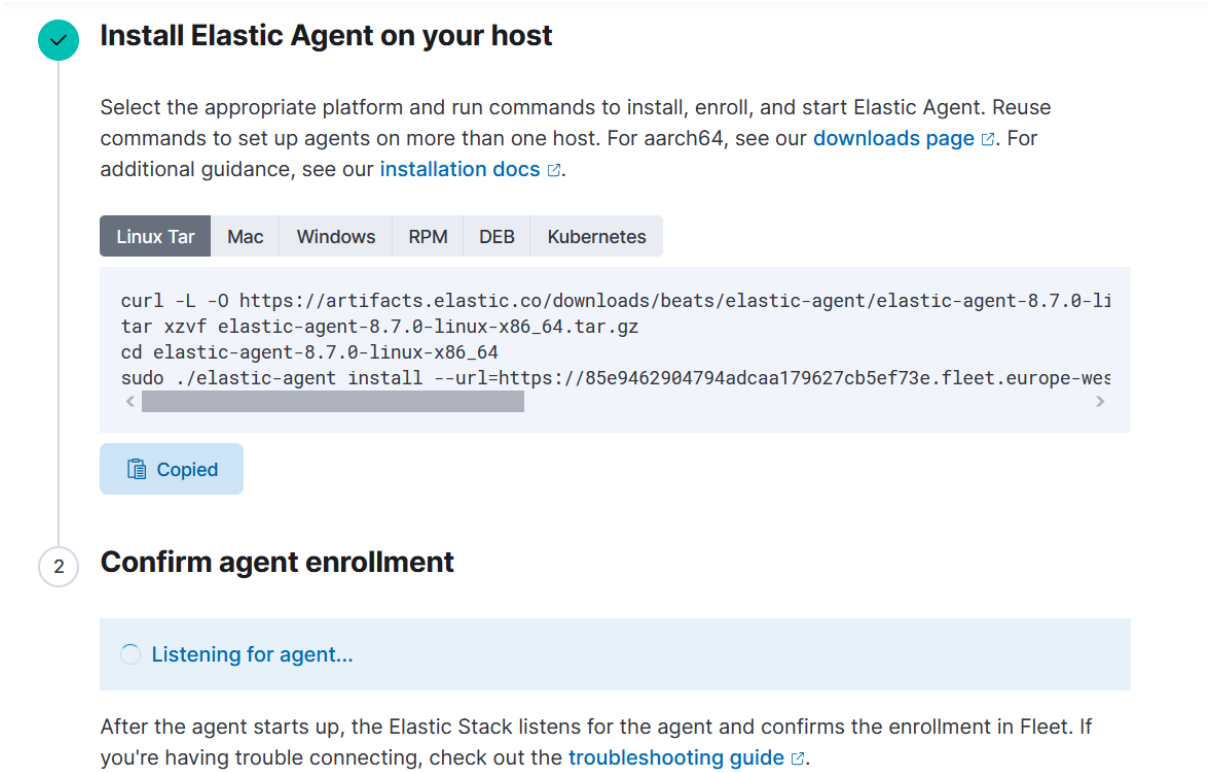
ELASTIC SEARCH, HONEYPOT Y SURICATA

Instalación de elastic search y suricata

Me registro en la página web de elastic search (https://www.elastic.co) pero en la opción de "Cloud" (https://cloud.elastic.co). Una vez registrado, cargo la configuración y una vez finalizado, busco y añado Suricata .



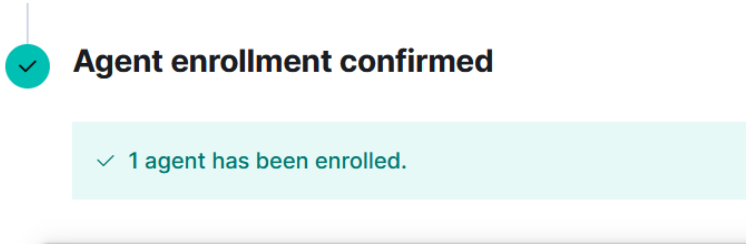
Me saldrá una ventanita en la parte inferior para instalar el “Elastic Agent”. Hago clic y aparecerá el instalador, el cual tendré que copiar lo que aparece en la primera parte y pegarlo en mi kali.



```
┌──(root㉿kali)-[/var/www/html]
└─# curl -L -O https://artifacts.elastic.co/downloads/beats/elastic-agent/elastic-agent-8.7.0-linux-x86_64.tar.gz
tar xzvf elastic-agent-8.7.0-linux-x86_64.tar.gz
cd elastic-agent-8.7.0-linux-x86_64
sudo ./elastic-agent install --url=https://85e9462904794adcaa179627cb5ef73e.fleet.europe-west2.gcp.elastic-cloud.c
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  407M  100  407M    0     0  11.4M      0  0:00:35  0:00:35 --:--:-- 11.7M
elastic-agent-8.7.0-linux-x86_64/LICENSE.txt
elastic-agent-8.7.0-linux-x86_64/elastic-agent.yml
elastic-agent-8.7.0-linux-x86_64/elastic-agent.reference.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/elastic-agent
elastic-agent-8.7.0-linux-x86_64/NOTICE.txt
elastic-agent-8.7.0-linux-x86_64/.build_hash.txt
elastic-agent-8.7.0-linux-x86_64/README.md
elastic-agent-8.7.0-linux-x86_64/.elastic-agent.active.commit
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/.build_hash.txt
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/LICENSE.txt
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/NOTICE.txt
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/README.md
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/apm-server
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/apm-server.spec.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/apm-server.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/audit.rules.d/
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/audit.rules.d/sample-rules.conf.disabled
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/auditbeat
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/auditbeat.elastic-agent.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/auditbeat.reference.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/auditbeat.spec.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/auditbeat.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/bundle.tar.gz
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/certs/
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/certs/certs.pem
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/checksum.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/cloudbeat
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/cloudbeat.spec.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/cloudbeat.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/elastic-agent-shipper
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/elastic-agent-shipper.spec.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/elastic-agent-shipper.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/endpoint-security
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/endpoint-security-resources.zip
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/endpoint-security.spec.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/fields.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/filebeat
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/filebeat.reference.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/filebeat.spec.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/filebeat.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/fleet-server
```

Cuando finalice aparecerá ésto.

En interfaz:



En consola de kali:

```
Successfully enrolled the Elastic Agent.
Elastic Agent has been successfully installed.
```

Después me aparecerá las siguientes ventanas en la parte inferior:

Y ya estaría mi elastic y suricata instalado. Elastic estaría en Cloud, y el Suricata en DMZ_2. De esta manera, cuando el honeypot permita una conexión será registrado por el suricata, y el elastic visualizará los logs. Ahora procederé a la creación de nuevas reglas para que se puedan permitir estas conexiones entrantes.

CREACIÓN DE NUEVAS REGLAS

NAT

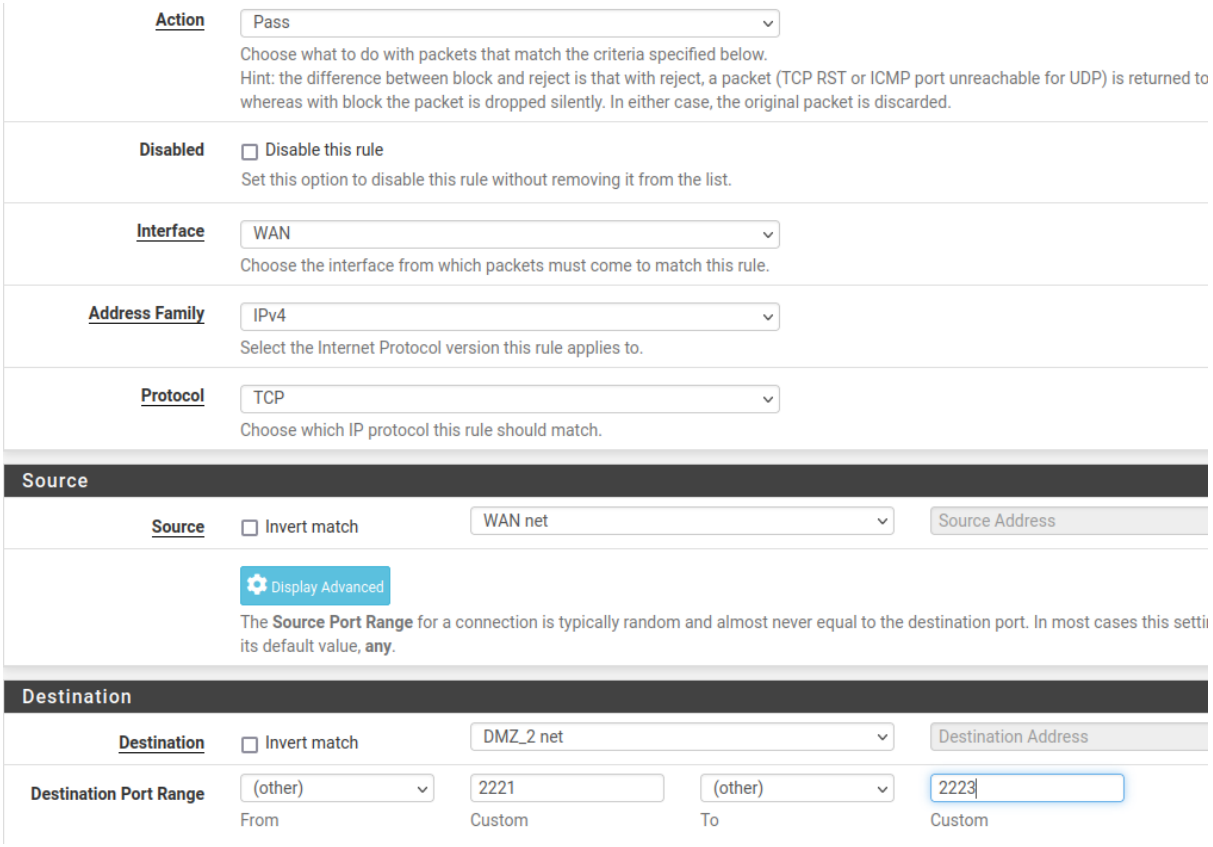
En Nat crearé una regla para que las conexiones que entren por la interfaz WAN, por el protocolo TCP, con origen desde cualquier lugar con destino a mi WAN al puerto 22 (ssh), puedan entrar y sean redirigidos al puerto 2222. (Se crea regla del puerto 22, el ssh, para dar paso a aquellos ciberdelincuentes que ataquen a través de ese puerto, y que sean redirigidos al honey de DMZ_2 por el puerto 2222).

Rules

	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions
☐	WAN	TCP	*	*	WAN address	9090	192.168.90.225	80 (HTTP)	Servidor web kali	
☐	WAN	TCP	*	*	WAN address	9091	192.168.80.225	80 (HTTP)	Test	
☐	WAN	TCP	*	*	WAN address	22 (SSH)	192.168.80.101	2222	ssh honey	

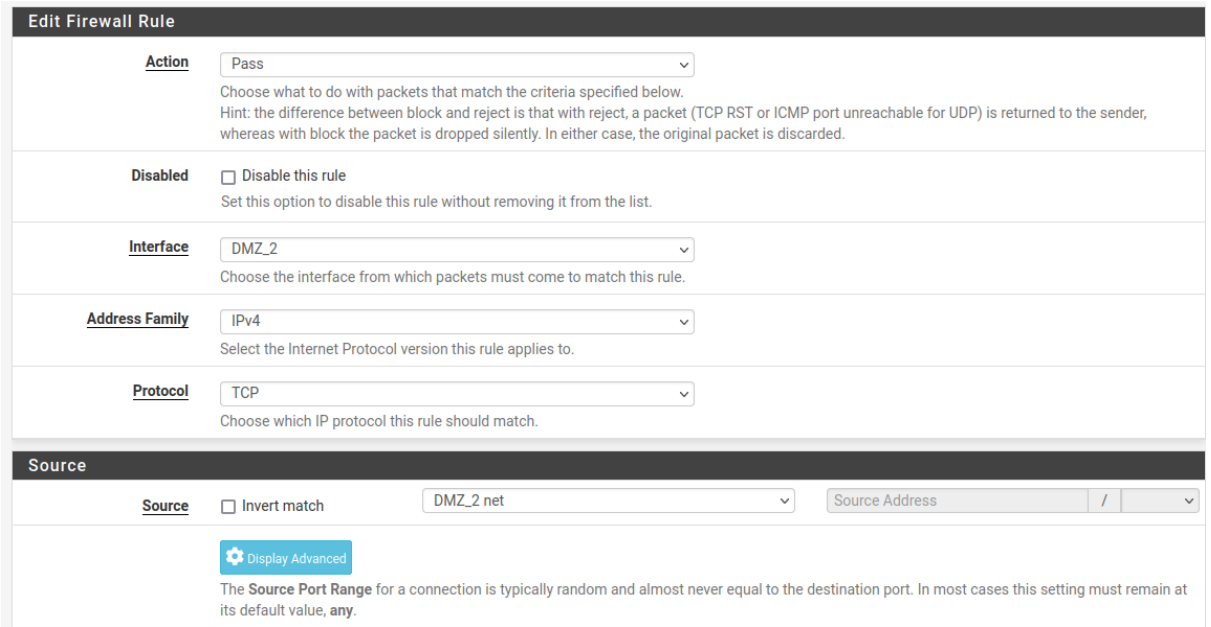
WAN

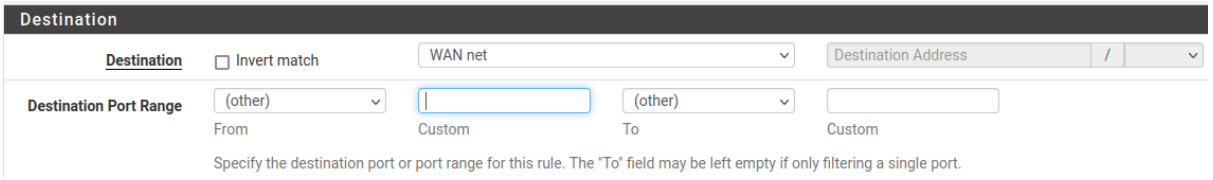
En WAN crearé otra regla para que por la interface WAN, por el protocolo TCP, con origen desde WAN con destino a DMZ_2 y en un rango de puertos de 2221-2223 puedan entrar. (Esto es para los atacantes que hayan accedido a mi WAN se redirijan a DMZ_2).



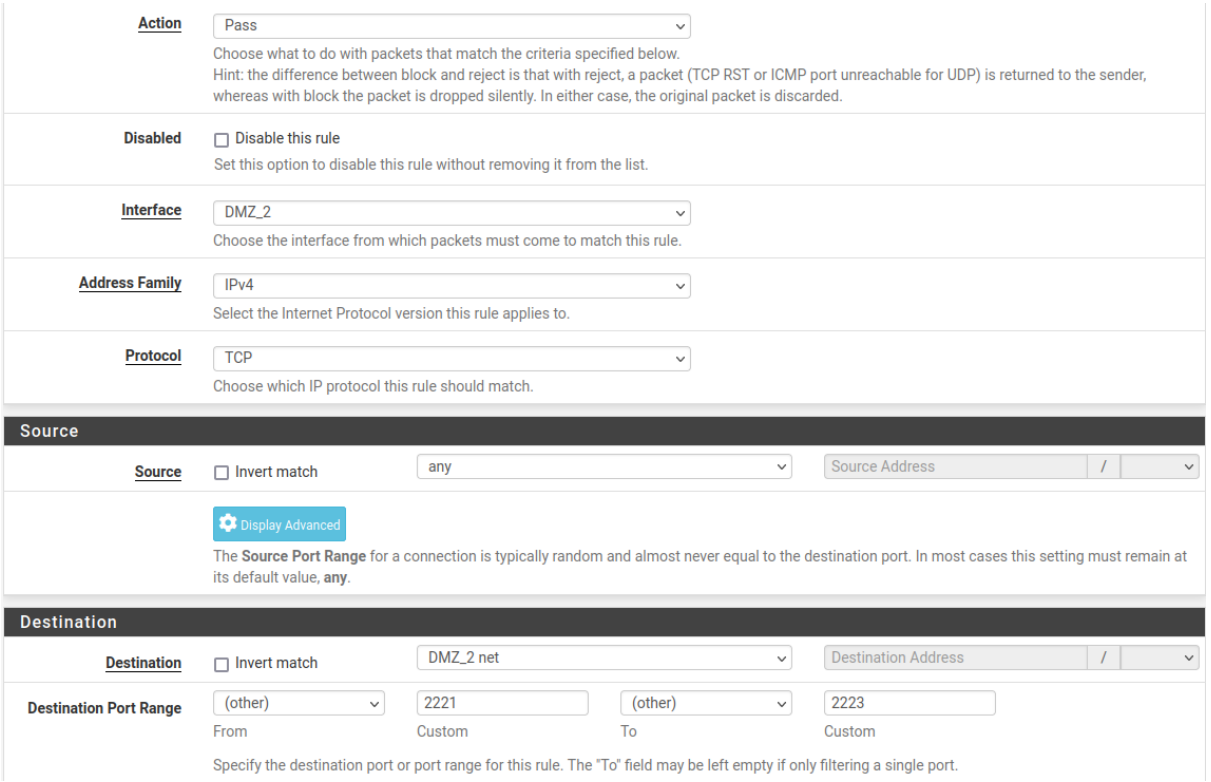
DMZ_2

En DMZ_2 crearé dos reglas. La primera será para que por la interface DMZ_2 por el protocolo TCP, con origen DMZ_2 con destino a WAN por cualquier puerto, se les permita pasar. (Esto es para que llegue al elastic los logs de honey).





Y otra regla que por la interface de DMZ_2, por el protocolo TCP, con origen desde cualquier lugar con destino a DMZ_2, en un rango de puertos del 2221-2223, se les permita el paso. (Esto es para permitir el paso de los atacantes a DMZ_2)



CREACIÓN DE REGLA EN SURICATA

En el suricata crearé una regla por consola para que puedan registrarse las conexiones entrantes a través del puerto ssh, que serán redirigidos por el puerto 2222, a través del protocolo TCP desde cualquier lugar. Para ello primero creo el fichero de la regla que llamaré “suricata.rules”, lo editaré con “nano” y meteré la regla.

```
┌──(root㉿kali)-[/home/kali/elastic-agent-8.7.0-linux-x86_64]
└─# cd /etc/suricata/rules

┌──(root㉿kali)-[/etc/suricata/rules]
└─# ls
app-layer-events.rules  http-events.rules       smb-events.rules
decoder-events.rules    ipsec-events.rules      smtp-events.rules
dhcp-events.rules       kerberos-events.rules   ssh-events.rules
dnp3-events.rules       modbus-events.rules     stream-events.rules
dns-events.rules        mqtt-events.rules       tls-events.rules
files.rules             nfs-events.rules
http2-events.rules      ntp-events.rules

┌──(root㉿kali)-[/etc/suricata/rules]
└─# nano suricata.rules
```

Añado la regla.

```
  GNU nano 7.2                    suricata.rules
alert tcp  any any → any 2222 (msg: “ssh”; sid:100; priority:1;)
```

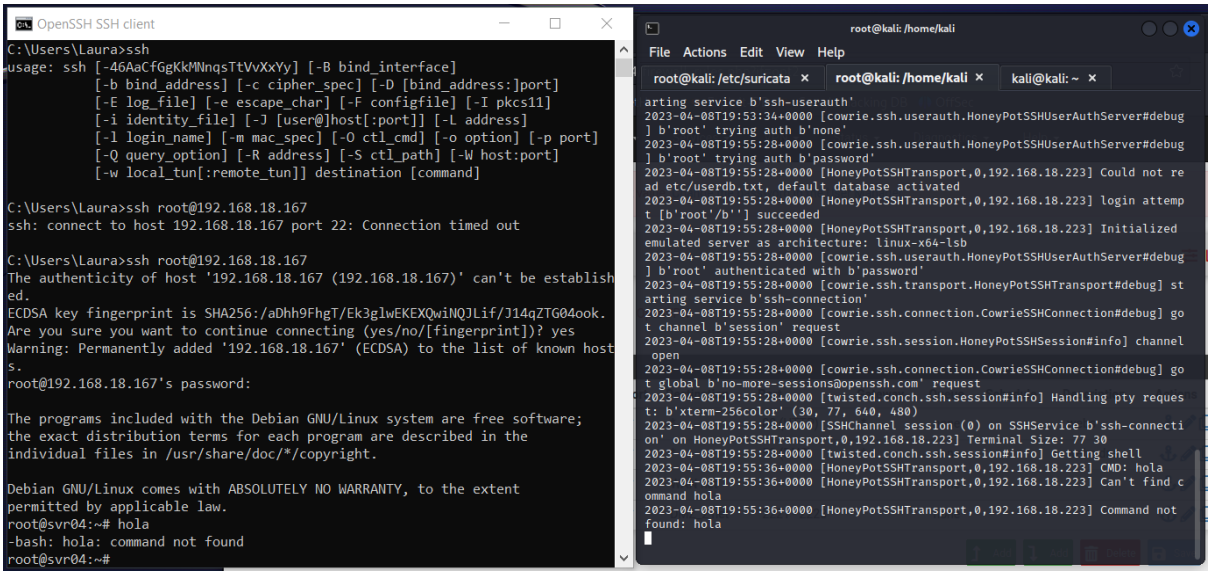
Ahora ejecuto suricata y después el honeypot con docker.

```
┌──(root㉿kali)-[/etc/suricata]
└─# suricata -c suricata.yaml -i eth1
8/4/2023 -- 15:37:53 - <Notice> - This is Suricata version 6.0.10 RELEASE run
ning in SYSTEM mode
8/4/2023 -- 15:37:53 - <Notice> - all 2 packet processing threads, 4 manageme
nt threads initialized, engine started.
```

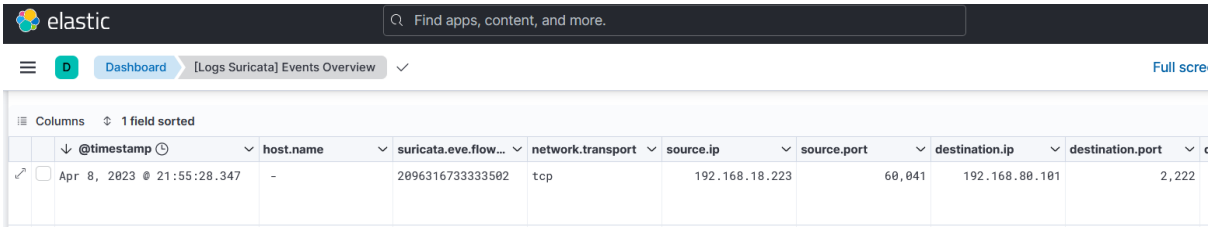
```
┌──(root㉿kali)-[/home/kali]
└─# docker run -p 2222:2222 cowrie/cowrie
Unable to find image 'cowrie/cowrie:latest' locally
latest: Pulling from cowrie/cowrie
fc251a6e7981: Pull complete
7be4d3667295: Pull complete
a1f1879bb7de: Pull complete
7eb7c5946a58: Pull complete
1817c8a12818: Pull complete
581833d6638a: Pull complete
277414dc2707: Downloading  61.93MB/79.04MB
f8b7231d72a2: Download complete
4f4fb700ef54: Download complete
```

Después simulo ser una ciberatacante desde WAN y soy interceptada por el honeypot de mi DMZ_2.

```
C:\Users\Laura>ssh root@192.168.18.167
The authenticity of host '192.168.18.167 (192.168.18.167)' can't be established.
ECDSA key fingerprint is SHA256:/aDhh9FhgT/Ek3glwEKEXQwiNQJLif/J14qZTG04ook.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.18.167' (ECDSA) to the list of known hosts.
root@192.168.18.167's password:
```



Si me voy a mi elastic search, podré comprobar que se puede visualizar el log y también lo almacena.

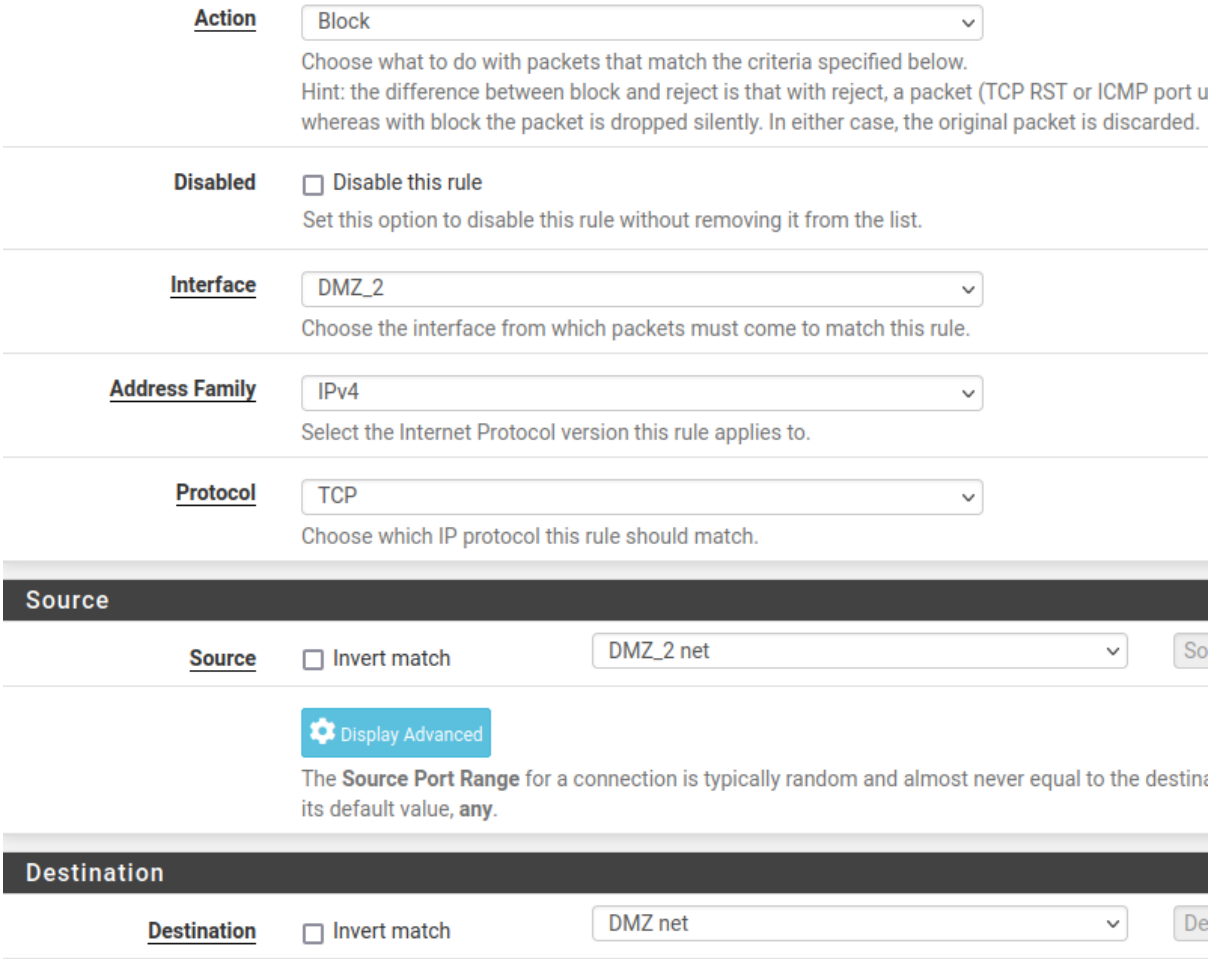


CREACIÓN DE NUEVAS REGLAS. BLOQUEOS.

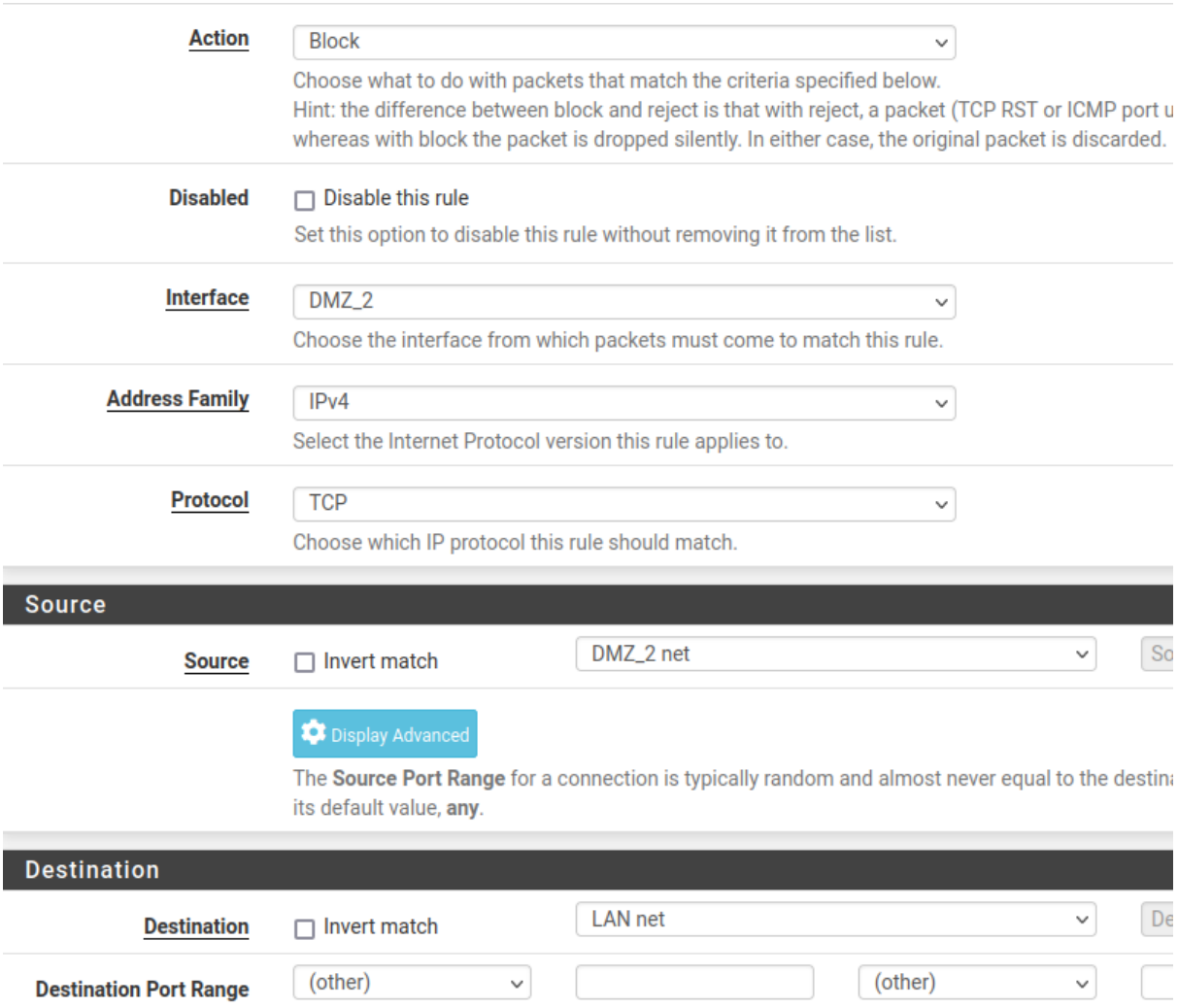
A continuación, crearé reglas para que DMZ_2 solo tenga salida al elastic search que está en la WAN y entrada de posibles ciberatacantes desde la WAN. De esta manera se bloqueará el acceso de DMZ_2 a DMZ y LAN, tanto desde su interfaz como desde la interfaz de éstas.

En DMZ_2:

- Creo regla para bloquear salida de DMZ_2 con destino DMZ.

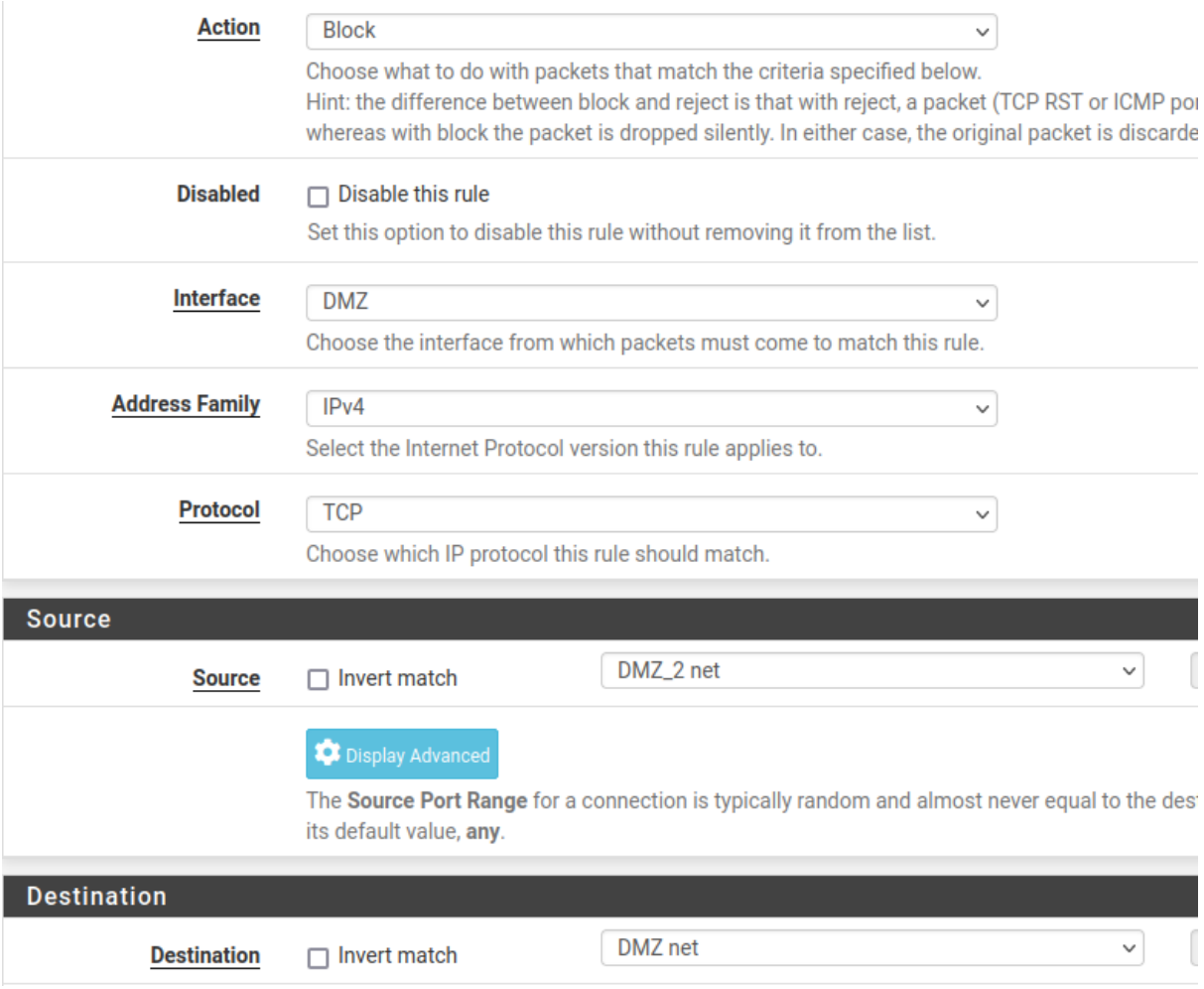


- Creo regla para bloquear salida de DMZ_2 con destino LAN.


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 u
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
DMZ_2
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
TCP
Choose which IP protocol this rule should match.
Source
Source
Invert match
DMZ_2 net
Display Advanced
The Source Port Range for a connection is typically random and almost never equal to the destina
its default value, any.
Destination
Destination
Invert match
LAN net
Destination Port Range
(other)
(other)

En DMZ:

- Bloqueo la conexión de DMZ_2 a DMZ en la interfaz DMZ.



En LAN:

Bloqueo salida de DMZ_2 con destino LAN por la interfaz LAN.