WireGuard VPN

Servidor VPN

- Instalación y configuración de WireGuard
- Añadir un cliente a WireGuard

Instalación y configuración de WireGuard

Para instalar WireGuard hacemos lo siguiente como cualquier paquete.

```
yoda@wireguard:~$ sudo apt install wireguard
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  wireguard-tools
Paquetes sugeridos:
  openresolv | resolvconf
Se instalarán los siguientes paquetes NUEVOS:
  wireguard wireguard-tools
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
Se necesita descargar 90,0 kB de archivos.
Se utilizarán 345 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Ahora vamos a generar directamente una clave privada y vamos a guardarla.

```
yoda@wireguard:~$ sudo wg genkey | sudo tee /etc/wireguard/serverpriv.key
[sudo] password for yoda:
[redacted]
yoda@wireguard:~$
```

Este comando nos dará una clave aleatoria por ejemplo "3JNHDCNSM3DHISN(&JD83ND3U"

Ahora vamos a cambiar los permisos de la clave para que solo pueda acceder el usuario creador.

```
yoda@wireguard:~$ sudo chmod go= /etc/wireguard/serverpriv.key
yoda@wireguard:~$
```

Ahora necesitamos una clave pública pero que este relacionada con la privada.

Vamos a suponer "874FD6V8S838YE8%&(8648367".

```
yoda@wireguard:~$ sudo cat /etc/wireguard/serverpriv.key | wg pubkey | sudo tee /etc/wireguard/serverpub.key
[redacted]
yoda@wireguard:~$
```

Ahora vamos a crear y modificar el archivo wg0.conf del directorio de configuración de wireguard.

```
yoda@wireguard:~$ sudo nano /etc/wireguard/wg0.conf
```

Y añadimos la siguiente configuración.

```
[Interface]
PrivateKey = [redacted]
Address = 10.8.0.1/24
ListenPort = 10296
SaveConfig = true
```

Donde pone privatekey, como el propio parámetro dice hay que poner la clave privada que fue la primera que creamos, para el tutorial vamos a coger el dicho anteriormente "3JNHDCNSM3DHISN(&JD83ND3U"

En dirección ponemos la ip que queramos que tenga el servidor vpn, no se refiere al rango.

El puerto que queramos poner, en este ejemplo he puesto 10296.

Y el parámetro saveconfig para que guarde la configuración cuando una interfaz de wireguard se apague.

Ahora vamos a ver que nombre de interfaz de red tenemos en el servidor.

```
yoda@wireguard:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether [redacted] brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 192.168.1.32/24 brd 192.168.1.255 scope global ens18
       valid_lft forever preferred_lft forever
    inet6 2a0c:5a84:f410:1800:be24:11ff:fe5f:183b/64 scope global mngtmpaddr noprefixroute
       valid_lft forever preferred_lft forever
    inet6 fe80::be24:11ff:fe5f:183b/64 scope link
       valid_lft forever preferred_lft forever
yoda@wireguard:~$
```

Vemos que es ens18 en este caso, nos vamos al archivo wg0 y añadimos dos reglas de iptables que hacen lo siguiente:

PostUp:

iptables -A FORWARD -i %i -j ACCEPT: Añade una regla a la cadena FORWARD para permitir el reenvío de paquetes provenientes de la interfaz especificada (%i).

iptables -t nat -A POSTROUTING -o [NOMBRE_INTERFAZ_INTERNET] -j MASQUERADE: Añade una regla a la tabla de nat para realizar el enmascaramiento de direcciones (NAT) en la interfaz de

salida especificada ([NOMBRE_INTERFAZ_INTERNET]), lo que permite que los paquetes parezcan provenir de la interfaz principal.

PostDown:

iptables -D FORWARD -i %i -j ACCEPT: Elimina la regla que permite el reenvío de paquetes provenientes de la interfaz especificada (%i).

iptables -t nat -D POSTROUTING -o [NOMBRE_INTERFAZ_INTERNET] -j MASQUERADE: Elimina la regla de enmascaramiento de direcciones para la interfaz de salida especificada ([NOMBRE_INTERFAZ_INTERNET]).

```
PostUp = iptables -A FORWARD -i ens18 -j ACCEPT; iptables -A FORWARD -o ens18 -j ACCEPT; iptables -t nat -A POSTROUTING -o ens18 -j MASQUERADE
PostDown = iptables -D FORWARD -i ens18 -j ACCEPT; iptables -D FORWARD -o ens18 -j ACCEPT; iptables -t nat -D POSTROUTING -o ens18 -j MASQUERADE
```

Ahora vamos a modificar los permisos del archivo wg0.

```
yoda@wireguard:~$ sudo chmod 600 /etc/wireguard/wg0.conf
yoda@wireguard:~$
```

Ahora vamos a probar a activar la interfaz.

```
yoda@wireguard:~$ sudo wg-quick up wg0
[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.8.0.1/24 dev wg0
[#] ip link set mtu 1420 up dev wg0
[#] iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o ens18 -j MASQUERADE
yoda@wireguard:~$
```

Y podemos comprobar el funcionamiento.

```
yoda@wireguard:~$ sudo wg show wg0
interface: wg0
  public key: [redacted]
  private key: (hidden)
  listening port: 10296
yoda@wireguard:~$
```

Ahora que hemos comprobado que funciona vamos a indicarle que se inicie automáticamente al iniciar el sistema.

```
yoda@wireguard:~$ sudo systemctl enable wg-quick@wg0
Created symlink /etc/systemd/system/multi-user.target.wants/wg-quick@wg0.service → /lib/systemd/system/wg-quick@.service.
yoda@wireguard:~$
```

Ahora vamos a modificar el sistema para permitir el reenvío del tráfico.

```
yoda@wireguard:~$ sudo nano /etc/sysctl.conf
yoda@wireguard:~$
```

Entramos y descomentamos la primera línea de la imagen, la segunda es por si lo estáis haciendo con ipv6.

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
net.ipv6.conf.all.forwarding=1
```

Ahora comprobamos que este bien.

```
yoda@wireguard:~$ sudo sysctl -p
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
yoda@wireguard:~$
```

Tenemos que activar el firewall y añadirle el puerto de la vpn.

```
yoda@wireguard:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
yoda@wireguard:~$ sudo ufw allow 10296/udp
Rule added
Rule added (v6)
yoda@wireguard:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
10296/udp                  ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
10296/udp (v6)             ALLOW       Anywhere (v6)

yoda@wireguard:~$
```

Y el ultimo paso es modificar el MTU para bajarlo a 1420.

sudo ip link set dev <nombre_interfaz> mtu 1420

Y con ello ya podríamos dar por terminado la instalación y configuración de wireguard, el siguiente tutorial es como añadir un cliente.

Añadir un cliente a WireGuard

Si ya esta el servidor encendido y no queremos pararlo, podemos añadir clientes/pares con el siguiente comando.

Creamos las claves privadas y publicas del cliente (necesitamos entrar en root)

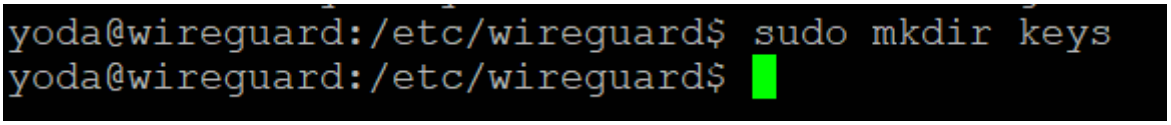
wg genkey > /etc/wireguard/keys/cliente.key

cat /etc/wireguard/keys/cliente.key | wg pubkey

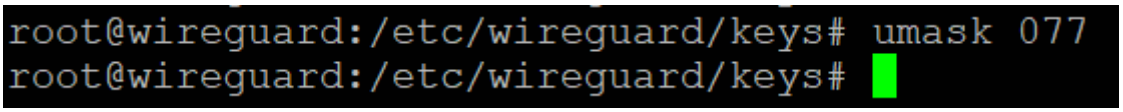
Y lo añadimos con lo siguiente.

sudo wg set wg0 peer [CLAVE_PUBLICA_CLIENTE] allowed-ips [IP_CLIENTE_VPN]

Para empezar vamos a crear una carpeta que almacene las claves.



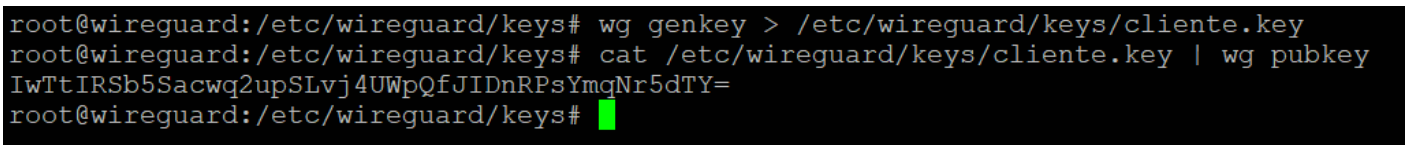
Y dentro del directorio le otorgamos permisos predeterminados para que solo el usuario pueda entrar a los documentos.



Creamos las claves privadas y publicas del cliente (necesitamos entrar en root)

wg genkey > /etc/wireguard/keys/cliente.key

cat /etc/wireguard/keys/cliente.key | wg pubkey



Ahora nos vamos al archivo wg0.conf y añadimos lo siguiente.

```
[Peer]
PublicKey = IwTtIRSb5Sacwq2upSLvj4UWpQfJIDnRPsYmqNr5dTY=
AllowedIPs = 10.8.0.8/32
```

Ahora encendemos el servidor y comprobamos el estado.

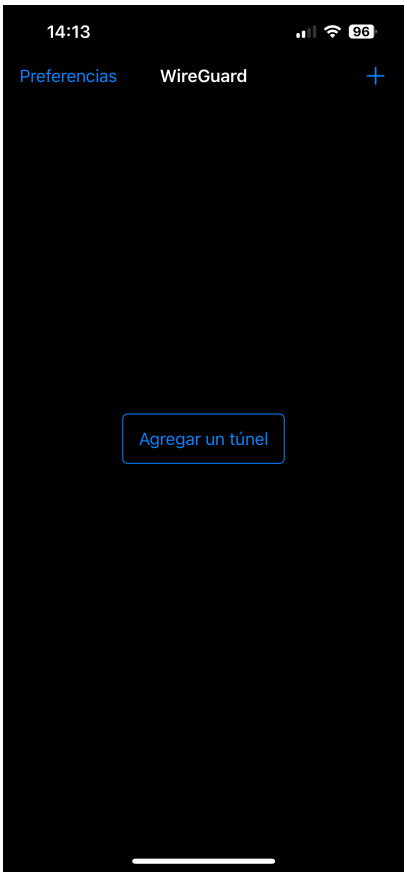
```
root@wireguard:/etc/wireguard# sudo wg-quick up wg0
[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.8.0.1/24 dev wg0
[#] ip link set mtu 1420 up dev wg0
[#] iptables -A FORWARD -i wg0 -j ACCEPT; iptables -A FORWARD -o wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o ens18 -j MASQUERADE
root@wireguard:/etc/wireguard# sudo wg show wg0
```

```
root@wireguard:/etc/wireguard# sudo wg show wg0
interface: wg0
  public key: [redacted]
  private key: (hidden)
  listening port: 10296
```

```
peer: IwTtIRSb5Sacwq2upSLvj4UWpQfJIDnRPsYmqNr5dTY=
  allowed ips: 10.8.0.8/32
root@wireguard:/etc/wireguard#
```

Ahora tenemos que irnos al cliente e instalar wireguard, en mi caso iré a un iphone.

Al instalar la aplicación y entrar nos dirá de agregar un tunel.

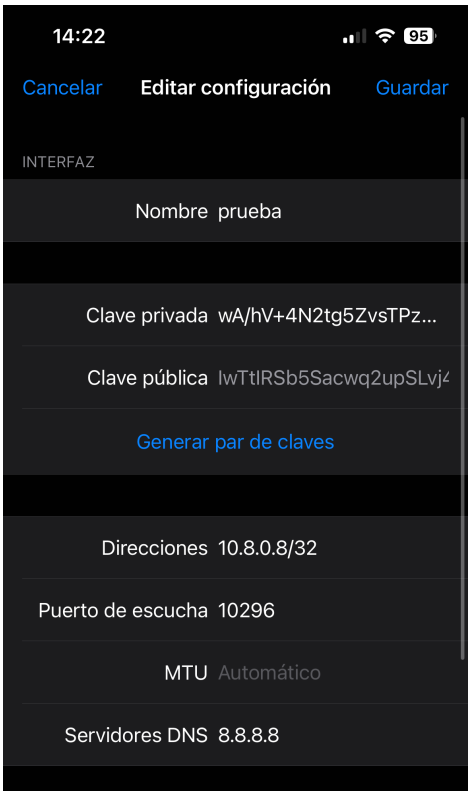


Ahora tenemos que rellenar los siguientes puntos.

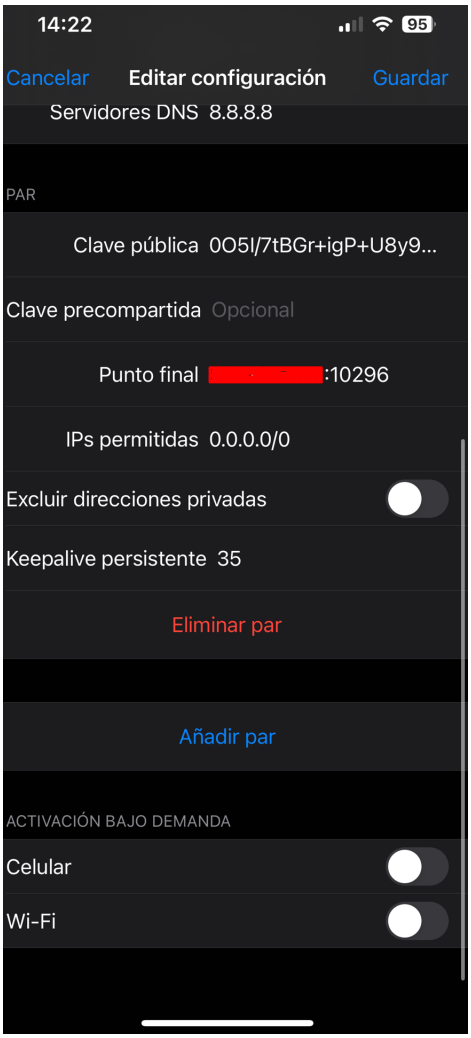
Las claves son las generadas en el servidor.

La dirección es la que le asignamos en el servidor.

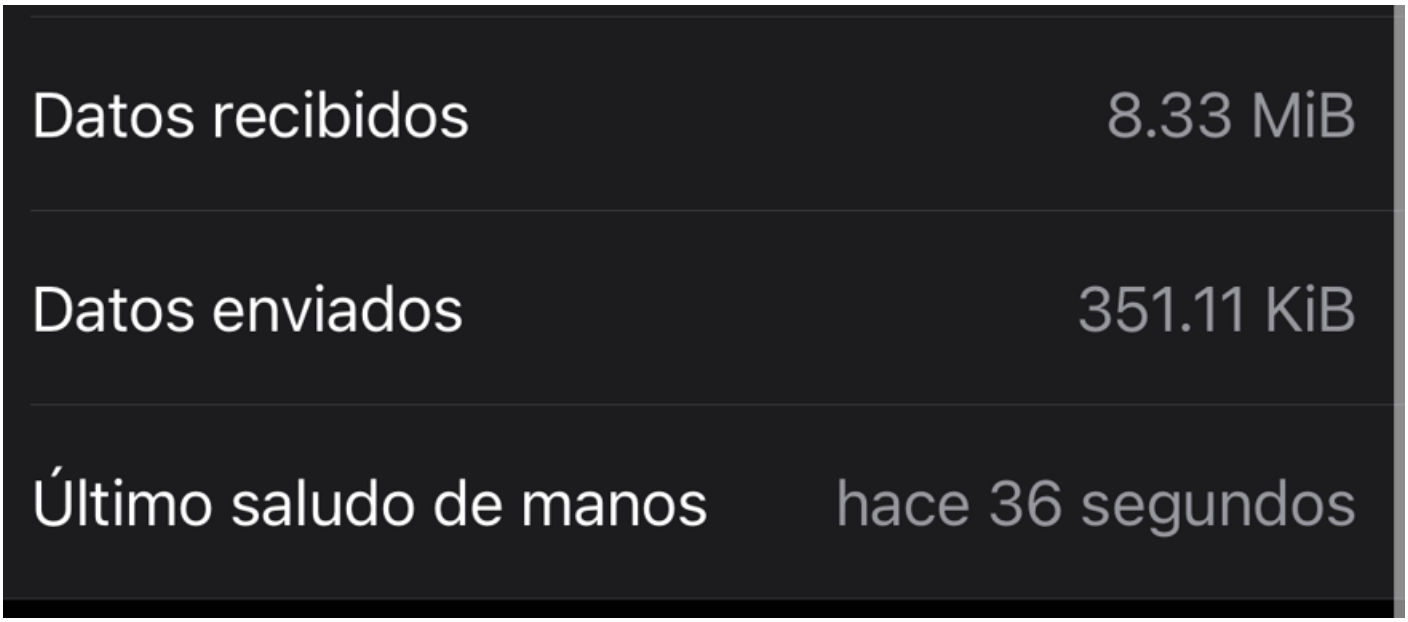
Puerto de escucha la misma que el servidor y en mi caso puse el DNS de google.



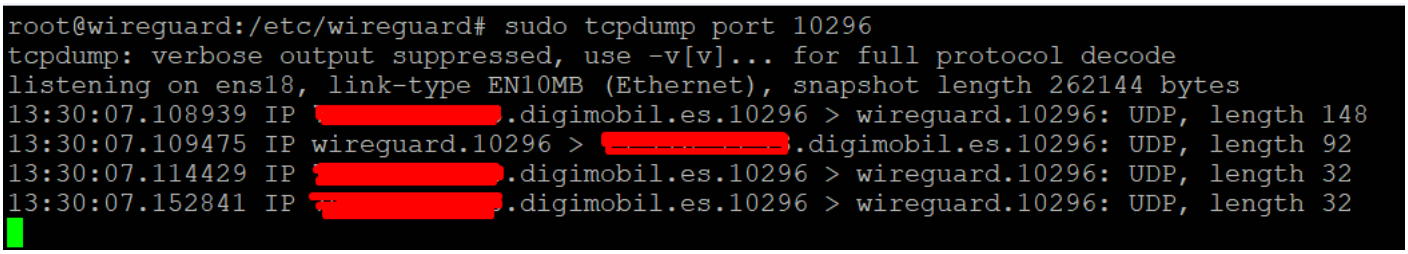
En la parte de abajo tenemos que poner la clave pública del servidor, donde pone punto final se refiere a la ip pública del servidor junto al puerto, el IPs permitidas recomiendo siempre dejarlo en 0.0.0.0/0 a no ser que queramos otra cosa y el Keepalive nos recomienda wireguard 25 segundos pero yo pongo 35, esto sirve para que cada 35 segundos compruebe si la conexión sigue activa. Las dos opciones ultimas "Celular" y "Wi-Fi" es para que pongamos si queremos que la vpn se active automaticamente al utilizar datos o wifi.



Una vez conectados veremos lo siguiente en el apartado par.



Y si en el servidor ponemos el siguiente comando veremos como llegan las solicitudes al servidor.



Y ya tendríamos todo listo, y si haces un script que automáticamente añada el par en el servidor y le mande al usuario un archivo zip para añadirlo a la aplicación sin tocar nada?