

# Crear puntos de acceso AP con VPN

este capítulo trata de como crear puntos de acceso (AP) para compartir una red wifi y enrutar el tráfico por una VPN

- Punto de acceso portable con Raspberry pi y Wireguard VPN

# Punto de acceso portable con Raspberry pi y Wireguard VPN

## Índice

- Introducción
- Que es Wireguard VPN?
- Requisitos
- Configurando la SD para usar Rpi en modo “headless”
- Asegurando la Rpi
- Instalando y configurando Wireguard
- Configurando la Rpi como AP o “Hotspot”
- Configurando DHCP y DNS para servir a los clientes que se conectan por wifi
- Configurando Iptables para crear las reglas NAT y de redireccionamiento necesarias
- Arrancar la red y servicios
- Conectándonos a la red con distintos dispositivos y haciendo comprobaciones

## Introducción

Como sabemos las redes Wifi son consideradas inseguras, especialmente aquellas de lugares públicos como bibliotecas, bares, estaciones de tren, etc. Las redes domésticas también se pueden considerar inseguras según su configuración y son potencialmente espiables por parte de nuestro proveedor ISP. Por esto es recomendable cifrar el trafico de datos a través de la red. Una buena opción para ello es usar una VPN.

En este articulo vamos a ver como conectar nuestra raspberry pi a una instancia de Wireguard VPN bien sea en un VPS propio o a la que nos ofrezca un proveedor de este servicio. También

veremos como convertir nuestra RPi en un punto de acceso portátil que enrute el tráfico de los clientes, que se conecten a este, por la VPN. También veremos durante el proceso maneras de hacer más segura nuestra RPi.

## Que es Wireguard VPN?

WireGuard es una aplicación de software libre y de código abierto y un protocolo de comunicación que implementa técnicas de red privada virtual (VPN) para crear conexiones seguras punto a punto en configuraciones enrutadas o puentes. Se ejecuta como un módulo dentro del kernel de Linux y tiene como objetivo un mejor rendimiento que los protocolos de tunelización IPsec y OpenVPN. Fue escrito por Jason A. Donenfeld y se publica bajo la segunda versión de la GNU General Public License (GPL). [Más información](#)

Recientemente Wireguard ha sido incluido como módulo en el kernel 5.6 en adelante, [más información](#)

Entre otras característica Wireguard es reconocido por:

- Rápida y fácil implementación.
- Código base reducido a 4.000 líneas, lo que lo hace fácil de auditar. A diferencia OpenVPN usa unas 400.000 líneas de código lo
- que hace más complejo encontrar “bugs”.
- WireGuard utiliza Curve25519 para el intercambio de claves, ChaCha20 para la encriptación, Poly1305 para la autenticación de datos, SipHash para claves de hashtables y BLAKE2s para el hashing.
- Soporta la capa 3 para IPv4 e IPv6 y puede encapsular v4 en v6 y viceversa.
- Permite el cambio de redes sin interrupción, “roaming” entre conexiones
- Velocidades muy rápidas
- Bajo consumo de batería y energía en los dispositivos
- Open Source

# Requisitos

- Configurar Wireguard VPN en un VPS o en un servidor
- Imagen de Raspbian Buster en el PC o otra distribución al gusto
- Tener un programa para transferir la imagen a la microSD
- Raspberry pi modelos 2, 3 y 4 (puede funcionar con Rpi 1 y Archlinux o otra distribución compatible con armv6) También si se puede es recomendable usar una placa de hardware libre como por ejemplo OrangePi
- tarjeta microSD
- 2 tarjetas de red wifi, la integrada y una extra (opcional)
- nociones básicas de la línea de comandos y redes
- nociones básicas de iptables

## Configurando la SD para usar Rpi en modo “headless”

Hay varias opciones para usar y configurar una Raspberrypi, en este caso vamos a configurar y usar la raspberrypi en modo “headless”. Es decir, sin teclado ni monitor. Vamos a interactuar con ella a través de nuestro PC usando SSH.

1. Descargar la imagen **Raspbian Buster** y transferirla a nuestra SD por ejemplo con gnome-disks o otras como etcher o incluso con la herramienta dd desde la línea de comandos
2. Montar la partición boot de la SD en nuestro PC bien desde el administrador de archivos o por línea de comandos
3. desde nuestra terminal acceder a la partición boot y crear el archivo ssh que nos permitirá acceder a la Rpi

	cd /
	media
	/
1	tu_usuario
2	/
3	boot
	/
	touch
	ssh

4. Paso opcional, solo necesario si vamos a conectar la Rpi por wifi con una de las tarjetas y sin cable ethernet

	nano
1	wpa_supplicant
	.
	conf

y añadir los datos de la red wifi a la que queremos que se conecte la Rpi

	ctrl_interface
	=
	DIR
	=/
	var/
	run/
	wpa_supplicant
	GROUP
	=
1	netdev
2	update_config
3	=1
4	country
5	=ES
6	network
7	={
8	ssid
	=
	"nombre_de_la_red"
	psk
	=
	"contraseña
	de
	la
	red"
	}

5. salir del directorio /media/tu\_usuario/boot/ para poder desmontar la SD en el siguiente paso

1	cd --
---	-------

6. Desmontar la microSD del PC, bien desde el gestor de archivos o bien con el comando umount

7. montar la tarjeta SD en la Rpi y enchufarla a la corriente

8. Conectar a la Rpi por ssh

1	ssh pi@ raspberrypi . local
---	---

el password por defecto es: raspberry

## Asegurando la Rpi

En este apartado veremos como asegurar un poco nuestra Rpi, los pasos siguientes son opcionales aunque muy recomendables

1. vamos a reforzar el acceso SSH para que sea solo accesible con sshkey en nuestro PC si no la tenemos vamos a generar un sshkey

1	ssh- keygen -t rsa - b 4096 -C "tu_email@example.com"
---	--

Esto crea una nueva clave ssh usando el correo electrónico proporcionado como etiqueta. Cuando se te indique “Ingresar un archivo donde guardar la clave”, presiona Intro. Al hacerlo aceptas la ubicación predeterminada del archivo.

A continuación te va a pedir que ingreses una contraseña, que debe ser segura y no olvidada:

1	Enter
2	passphrase
	(
	empty
	for
	no
	passphrase
	): [
	Type
	a
	passphrase
	]
	Enter
	same
	passphrase
	again
	: [
	Type
	passphrase
	again
	]

2. Una vez creada la sshkey la transferimos a nuestra raspberrypi

1	ssh- copy -id pi@ raspberrypi . local
---	---

3. Entramos de nuevo a la Rpi, esta vez introduciremos nuestra “passphrase” en vez de la contraseña por defecto de la RPi “raspberrypi”

1	ssh pi@ raspberrypi . local
---	---



4. Si todo va bien es momento de limitar el acceso a la Rpi con sshkey y reforzar otros aspectos de la conexión SSH. Para ello en la RPi editamos el archivo `/etc/ssh/sshd_config` y hacemos los siguientes cambios:

	PubkeyAuthentication
	yes
1	PermitRootLogin
2	no
3	PermitEmptyPasswords
4	no
	PasswordAuthentication
	no

Guardamos y cerramos

Aunque hay otros parámetros para reforzar la conexión SSH, no los vamos a tratar aquí. Pero por el momento con esta configuración ya está bastante bien

5.Reiniciamos el servicio SSH para que los cambios se hagan efectivos

1	sudo service ssh restart
---	-----------------------------------

6. Como hemos observado al usar “sudo” en la RPi no nos pide contraseña, lo cual no es nada conveniente. Para resolverlo editar el archivo `/etc/sudoers.d/010_pi-nopasswd` y modificamos lo siguiente

1	pi ALL =( ALL) NOPASSWD : ALL
---	---

cambiarlo por

1	pi ALL =( ALL) ALL
---	--------------------------------

cerramos y guardamos

## Instalando y configurando Wireguard

1. Actualizar la Rpi

1	sudo apt update && sudo apt upgrade -y
---	---

2. Añadir repositorio de Wireguard





	echo
	"deb
	http://deb.debian.org/debian/
	unstable
	main"
	sudo
	tee
	--
	append
	/etc/
	apt/
	sources
	.list.
	d/
	unstable
	.list
	sudo
	apt-
	key
	adv
	--
	keyserver
1	keyserver
2	.
3	ubuntu
4	.
5	com
	--
	recv
	-
	keys
	04EE7237B7D453EC
	printf
	'Package:
	*\nPin:
	release
	a=unstable\nPin-
	Priority:
	150\n'
	sudo
	tee
	--
	append
	/etc/
	apt/
	preferences

### 3. Sincronizar repositorios y instalar Wireguard

	sudo
	apt
	update
1	
2	sudo
3	apt
	install
	wireguard
	-y

4. Configurar la conexión Wireguard con los datos que hemos generado instalando Wireguard en nuestro VPS o con los datos que nos da nuestro proveedor. Editar el archivo `/etc/wireguard/wg0.conf`

	[
	Interface
	]
	Address
	=
	ip_peer_rpi
	/32
	DNS
	=
	dns_del_servidor
	PrivateKey
	=
1	clave_privada_rpi
2	
3	[
4	Peer
5	]
6	PublicKey
7	=
8	clave_p
9	ú
10	blica_servidor
	AllowedIPs
	=
	0.0.0.0
	/0
	Endpoint
	=
	ip_servidor
	:
	puerto_servidor
	PersistentKeepalive
	=
	21

Para desglosar un poco en la sección [Interface] configuramos lo relacionado a la Rpi:

Address = Aquí es la dirección VPN para la Rpi configurada en el “servidor” Wlreguard.

DNS = las DNS del servidor o bien usar las que se consideren

PrivateKey = es la clave privada para conectarse a la VPN y configurada en el servidor. Se puede generar en la propia Rpi o bien generarla en el servidor. Si usamos un proveedor también nos la

pueden mandar

Para desglosar un poco en la sección [Interface] configuramos lo relacionado al servidor:

PublicKey = es la clave pública para la conexión VPN del servidor

AllowedIPs = Aquí se determina que tráfico va enrutado por la VPN, en este caso todo

Endpoint = es la IP del servidor y el puerto de conexión

PersistentKeepalive = Este parámetro es necesario si nos encontramos detrás de una conexión NAT o CGNAT. Que son la mayoría de conexiones domésticas, bares, bibliotecas, etc

6. Permitimos el forward en los parámetros del kernel para que pueda pasar el tráfico ipv4 desde el archivo /etc/sysctl.conf descomentar la línea

1	net. ipv4 . ip_forward =1
---	---------------------------------------

Si usamos el protocolo ipv6 podemos descomentar también la línea referido a este. Guardamos y cerramos

7. aplicamos la regla anterior

1	sudo sysctl -p
---	----------------------

8. Instalamos paquetes necesarios para la persistencia de Iptables, para la configuración de las DNS, y para crear el punto de acceso:



1	<pre> sudo apt install hostapd dnsmasq dnsutils bc iptables - persistent </pre>
---	---

9. En este punto podemos ya probar la conexión VPN en nuestra RPi. Para ello podemos comprobar la IP externa de esta previamente a levantar la VPN

1	<pre> curl ip. stigok . com </pre>
---	------------------------------------

10. Levantamos la VPN y volvemos a comprobar y debería darnos la ip del servidor

1	<pre> sudo wg- quick up wg0 </pre>
---	------------------------------------

para comprobar nuestra VPN podemos también hacer ping a la ip VPN del servidor

## Configurando la Rpi como AP o “Hotspot”

1. Editamos el archivo `/etc/default/hostapd` para configurar el paquete `hostapd` instalado anteriormente y añadimos lo siguiente

	interface
	=
	wlan0
	hw_mode
	=g
	channel
	=10
	ieee80211d
	=1
1	country_code
2	=ES
3	ieee80211n
4	=1
5	wmm_enabled
6	=1
7	ssid
8	=
9	nombre_de_la_red
10	auth_algs
11	=1
12	wpa
13	=2
14	wpa_key_mgmt
	=
	WPA
	-PSK
	rsn_pairwise
	=
	CCMP
	wpa_passphrase
	=
	password_de_la_red

Elegir en “country\_code” el código de nuestro país, en “ssid” el nombre que queramos ponerle a nuestro Rpi “hotspot” y en “wpa\_passphrase” la contraseña de nuestra elección para el punto de acceso

2. modificamos el archivo /etc/network/interfaces para configurar las distintas interfaces de red en este caso eth0, wlan0, wlan1 y usb





	auto
	wlan0
	iface
	wlan0
	inet
	static
	address
	10.100.100.1
	netmask
	24
	allow
	-
	hotplug
	eth0
	iface
	eth0
	inet
1	dhcp
2	
3	
4	allow
5	-
6	hotplug
7	usb0
8	iface
9	usb0
10	inet
11	dhcp
12	
13	allow
14	-
15	hotplug
	wlan1
	iface
	wlan1
	inet
	dhcp
	wpa
	-
	driver
	wext
	wpa
	-
	conf
	/etc/
	wpa_supplicant

en este caso wlan0 será la interfaz que actúe como punto de acceso con la IP estática 10.100.100.1/24 y que actuará como “gateway” para los clientes que se conecten a la Rpi. Las otras interfaces de red quedan configuradas como posibles interfaces para la conexión a internet según cual se usada. EN este caso podemos usar eth0, una tarjeta de red externa por USB o wlan1 a las que se le asignará una dirección dhcp servida por el router al que se conecte la Rpi.

3. Para añadir nuevas redes wifi (debajo de la que hemos configurado en el primer apartado) y que la Rpi sea lo más portable posible podemos añadir las que necesitemos en el archivo /etc/wpa\_supplicant/wpa\_supplicant.conf con el siguiente formato:

	network
	= {
	ssid
	=
	"network
	1"
	psk
	=
	"password
	to
	network
1	1"
2	id_str
3	=
4	"w"
5	}
6	
7	network
8	= {
9	ssid
10	=
11	"network
	2"
	psk
	=
	"password
	to
	network
	2"
	id_str
	="z"
	}

# Configurando DHCP y DNS para servir a los clientes que se conectan por wifi

1. Vamos a usar “dnsmasq” así que primero desactivamos el servidor dhcp por defecto de Raspbian para la interfaz wlan0. Editar el archivo /etc/dhcpd.conf y añadir la siguiente línea

1	denyinterfaces wlan0
---	-------------------------

2. Vamos a hacer un backup de la actual configuración de dnsmasq antes de editar el archivo

1	sudo mv / etc/ dnsmasq . conf /etc/ dnsmasq . conf .orig
---	--

3. Editamos el archivo /etc/dnsmasq.conf y añadimos lo siguiente cambiando los parámetros a nuestro gusto



	dhcp
	-
	authoritative
	interface
	=
	wlan0
	listen
	-
	address
	=
	10.100.100.1
	dhcp
	-
	range
	=
1	10.100.100.50
2	,
3	10.100.100.150
4	,12h
5	read
6	-
7	ethers
8	bogus
	-priv
	domain
	-
	needed
	dhcp
	-
	option
	=
	option
	:dns
	-
	server
	,
	10.200.200.1

El parámetro dhcp-range determina el rango de IPs de los clientes que se conectan a la Rpi, así que modifica según tus necesidades. Importante que el parámetro dns-server sea igual al del VPN Server (Gateway) que hemos configurado anteriormente.

# Configurando Iptables para crear las reglas NAT y de redireccionamiento necesarias

Aquí lo recomendable sería añadir algunas reglas más que nos puedan interesar, eso lo dejo a cargo del lector.

	sudo
	iptables
	-t
	nat -
	A
	POSTROUTING
	-o
	wg0
	-j
	MASQUERADE
	sudo
	iptables
	-A
	FORWARD
	-i
	wlan0
	-o
1	wg0
2	-j
3	ACCEPT
	sudo
	iptables
	-A
	FORWARD
	-i
	wg0
	-o
	wlan0
	-m
	state
	--
	state
	RELATED
	,
	ESTABLISHED
	-j
	ACCEPT

Guardamos las reglas para que sean persistentes

1	sudo iptables - legacy - save
---	--

## Arrancar la red y servicios

1. Si tenemos otra tarjeta de red

1	sudo ifup wlan0
---	-----------------------

2. Arrancando los servicios dnsmasq y hostapd

1	sudo service dnsmasq start
2	sudo systemctl unmask hostapd
3	sudo service hostapd start
4	sudo update -rc.d hostapd enable

3. Arrancar el servicio Wireguard al inicio

1	<code>sudo systemctl enable wg- quick @ wg0 . service</code>
---	--

#### 4. Reiniciar la RPi

1	<code>sudo reboot</code>
---	------------------------------

## Conectándonos a la red con distintos dispositivos y haciendo comprobaciones

1. Conectarse a la red que hemos creado desde el PC y comprobar nuestra ip externa.

1	<code>curl ip. stigok . com</code>
---	--

También lo podemos hacer a través del navegador a través de distintos servicios web específicos para el caso

2. Opcionalmente para comprobar las DNS podemos instalar el paquete dnsutils y para comprobamos

	apt install dnsutils
1	nslookup
2	www
3	. startpage . com . 10.200.200.1

los resultados serian algo así según la ip que hayamos configurado en el servidor DNS:

	Server
	:
	10.200.200.1
	Address
	:
	10.200.200.1
	#53
	Non-
	authoritative
	answer
	:
1	www
2	.
3	startpage
4	.
5	com
6	canonical
7	name
	=
	startpage
	.
	com
	.
	Name
	:
	startpage
	.
	com
	Address
	:
	37.0.87.23

3. Comprobar si se quiere si tenemos filtración por DNS en sitios como por ejemplo  
[dnsleaktest.com](https://dnsleaktest.com)