Wireguard

Wireguard

Instalar wireguard

Métodos de instalación

Para instala Wireguard, podemos realizarlo bien desde la instalación tradicional, instalando todos los paquetes necesarios y después ejecutando la configuración de forma manual, o bien usar un script de instalación/mantenimiento que nos permite realizarlo de forma sencilla.

Script

Vamos a instalar wireguard desde el script.

Asegúrate de que el sistema esté actualizado.

```
apt update && apt dist-upgrade
```

Descarga la última versión del script para ello ejecutaremos

```
curl https://raw.githubusercontent.com/complexorganizations/wireguard-manager/main/wireguard-man
```

A continuación procederemos a ejecutar el script

```
bash /usr/local/bin/wireguard-manager.sh
```

El script te guía a través de la instalación.

En primer lugar, el script te pregunta sobre las direcciones IP y las herramientas para detectar las direcciones IP externas. La configuración recomendada debería funcionar bien. Cuando ejecutes este script, la detección de la dirección IP externa con curl no funcionó. Tuve que seleccionar la opción 2 (IP) para IPv4 y la opción 3 (Personalizado) para IPv6. La opción 3 le permite ingresar la dirección IP externa manualmente si ninguna de las detecciones automáticas funciona.

Las preguntas que realiza son las siguientes:

```
What ipv4 subnet do you want to use?
  1) 10.8.0.0/24 (Recommended)
  2) 10.0.0.0/24
  3) Custom (Advanced)
Subnetwork Choice [1-3]: 1

What ipv6 subnet do you want to use?
  1) fd42:42:42::0/64 (Recommended)
  2) fd86:ea04:1115::0/64
```

```
  3) Custom (Advanced)
Subnetwork Choice [1-3]: 1

How would you like to detect IPv4?
  1) Curl (Recommended)
  2) IP (Advanced)
  3) Custom (Advanced)
IPv4 Choice [1-3]: 2

How would you like to detect IPv6?
  1) Curl (Recommended)
  2) IP (Advanced)
  3) Custom (Advanced)
IPv6 Choice [1-3]: 3
Custom IPv6: 2a01:4f9:c010:2bff::/64

How would you like to detect NIC?
  1) IP (Recommended)
  2) Custom (Advanced)
nic Choice [1-2]: 1
```

A continuación, te pregunta sobre el puerto que deseas que WireGuard escuche. Es recomendable usar un puerto personalizado (opción 2). Para este ejemplo, usaremos 55443

```
What port do you want WireGuard server to listen to?
  1) 51820 (Recommended)
  2) Custom (Advanced)
  3) Random [1024-65535]
Port Choice [1-3]: 2
Custom port [1024-65535]: 55443
```

El uso de un puerto personalizado o aleatorio en comparación con el puerto 51820 predefinido no aumenta la seguridad, pero evita que los escaneos de puertos para detectar puertos abiertos en los well-konw ports. Un ataque dirigido contra el servidor con un escaneo completo de puertos revelará todos los puertos abiertos. A continuación, la secuencia de comandos pregunta sobre el intervalo keepalive y MTU. Te recomiendo que confirmes los valores predeterminados.

```
What do you want your keepalive interval to be?
  1) 25 (Default)
  2) Custom (Advanced)
  3) Random [1-25]
Nat Choice [1-3]: 1

What MTU do you want to use?
  1) 1280 (Recommended)
  2) 1420
  3) Custom (Advanced)
MTU Choice [1-3]: 1
```

Después de eso, el script pregunta qué versión de IP deben usar los clientes para conectarse al servidor WireGuard. Aquí siempre elegiría la opción 1 a menos que todos sus clientes solo usen

IPv6 pero no direcciones IPv4.

```
What IPv do you want to use to connect to WireGuard server?
  1) IPv4 (Recommended)
  2) IPv6
  3) Custom (Advanced)
IP Choice [1-3]: 1
```

A continuación, puede desactivar uno de los protocolos IP. Útil si está seguro de que solo va a utilizar uno de los protocolos IP, pero normalmente seleccionaría la opción 1. Aunque si solo se va a usar IPV4, usaremos la opción 3

```
Do you want to disable IPv4 or IPv6 on the server?
  1) No (Recommended)
  2) Disable IPV4
  3) Disable IPV6
Disable Host Choice [1-3]: 1
```

La siguiente opción se refiere al cliente de conexión VPN. El script le pregunta si el cliente debe reenviar todo el tráfico a través de la conexión VPN o excluir las direcciones IP privadas. Las direcciones IP privadas se encuentran en estos rangos 192.168.0.0 - 192.168.255.255, 172.16.0.0 - 172.31.255.255 y 10.0.0.0 - 10.255.255.255 y se utilizan para LAN internas. Si usa una conexión WireGuard y, al mismo tiempo, desea conectarse a su LAN, seleccione la opción 2. Si por ejemplo, todos los clientes no están conectados a una LAN, así que selecciono la opción 1. De lo contrario, si son clientes que están usando servicios locales LAN además del túnel, seleccionaremos la opción 2

```
What traffic do you want the client to forward to wireguard?
  1) Everything (Recommended)
  2) Exclude Private IPs
  3) Custom (Advanced)
Client Allowed IP Choice [1-3]: 1
```

A continuación, el script le pregunta sobre la instalación de un servidor DNS. El servidor WireGuard puede actuar como un DNS resolver. En esta configuración, todas las solicitudes de DNS de los clientes se enrutan a través de la VPN y se ocultan al ISP. Tiene la opción entre Unbound y PiHole. Si solo deseas un sistema de resolución de DNS simple, elija Unbound. Si deseas, además del DNS resolver, un bloqueador de anuncios, elija PiHole. En caso de usar una red con Active directory, seleccionaríamos custom (3) e introduciremos las direcciones IP de los DNS del Active Directory.

```
Which DNS provider would you like to use?
  1) Unbound (Recommended)
  2) PiHole
  3) Custom (Advanced)
DNS provider [1-3]: 1
```

Cuando eliges PiHole, el script de instalación inicia la instalación de PiHole al final del script. Visita la página de inicio del proyecto para obtener más información sobre PiHole: https://pi-hole.net/
Cuando elige la opción 3, el script de instalación presenta una lista de servicios DNS externos más

adelante en el proceso de instalación. A continuación, sus clientes enviarán solicitudes de DNS al servicio seleccionado.

```
Which DNS do you want to use with the VPN?
  1) Google (Recommended)
  2) AdGuard
  3) NextDNS
  4) OpenDNS
  5) Cloudflare
  6) Verisign
  7) Quad9
  8) FDN
  9) Custom (Advanced)
DNS [1-9]: 9
```

Por último, el script pregunta por un nombre para la configuración del cliente. El script no solo instala el servidor WireGuard, sino que también crea por defecto una configuración de cliente. Para esta instalación de demostración, llamo a esta configuración "miconfigvpn"

```
Lets name the WireGuard Peer, Use one word only, no special characters. (No Spaces)
Client name: miconfigvpn
```

Después de esto, el script comienza a instalar WireGuard y todas las bibliotecas dependientes. Como último paso, crea la configuración del cliente y muestra un código QR en la pantalla. Puedes escanear este código con su aplicación WireGuard para iOS y Android. Los scripts escriben la configuración del servidor en el archivo.

```
/etc/wireguard/wg0.conf
```

Y las configuraciones de los clientes es la carpeta

```
/etc/wireguard/clients/
```

Agregar más clientes

El script no solo es útil para la instalación inicial; también puede servir para crear más configuraciones de cliente. Cuando ejecuta el script por segunda vez, reconoce que el servidor WireGuard ya está instalado y presenta un menú diferente

```
root@vpnwireguard:~# bash /usr/local/bin/wireguard-manager.sh
What do you want to do?
   1) Show WireGuard Interface
   2) Start WireGuard Interface
   3) Stop WireGuard Interface
   4) Restart WireGuard Interface
   5) Add WireGuard Peer
   6) Remove WireGuard Peer
   7) Reinstall WireGuard Interface
   8) Uninstall WireGuard Interface
```

```
   9) Update this script
Select an Option [1-9]: 1
```

Routing en Wireguard

Configuración con varias interfaces

En esta sección configuraremos un wireguard que nos sirva de router para varias redes internas y /o externas

Para ello necesitamos que nuestro servidor Wireguard disponga de varias tarjetas de red

Supongamos que nuestras tarjetas de red son las ens18 (la tarjeta asociada a nuestra IP pública), la wg0, que como hemos visto es la interfaz de nuestro Wireguard, y varias tarjetas o VLANs adicionales. en nuestro caso serán por ejemplo en este primer caso tarjetas (ens19,ens20,ens21, ens22)

Editaremos la configuración de red de nuestro servidor Wireguard en Debian

Aparecerá como sigue:

```
allow-hotplug ens18
iface ens18 inet static
 address A.B.C.D/24
 gateway A.B.C.X
 # dns-* options are implemented by the resolvconf package, if installed
 dns-nameservers 1.1.1.1 9.9.9.9 8.8.8.8
 dns-search ateinco.net
```

Agregaremos las interfaces que tenemos creadas, con lo que nuestro fichero quedará como sigue,

```
allow-hotplug ens18 ens19 ens20 ens21 ens22
iface ens18 inet static
 address A.B.C.D/24
 gateway A.B.C.X
 # dns-* options are implemented by the resolvconf package, if installed
 dns-nameservers 1.1.1.1 9.9.9.9 8.8.8.8
 dns-search ateinco.net

iface ens19 inet static
 address E.F.G.H/24

iface ens20 inet static
 address I.J.K.L/24

iface ens21 inet static
 address 192.168.30.254/24
```

```
iface ens22 inet static
address 192.168.40.254/24
```

En nuestra interfaz wg0, ahora, veremos que en la parte de PostUP, aparece asi

```
[Interface]
Address = 10.9.0.1/24, fd86:ea04:1115::1/64
ListenPort = 51820
PrivateKey = XXXXXXXXXXXXXXXXXXXXXXXXXXXX
PostUp = iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o ens18 -j MASQUE
```

Comentaremos la parte de PostUp, e insertaremos un enlace a un script

```
#PostUp = iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o ens18 -j MASQU
PostUp = /etc/wireguard/iptables/ipup.sh
```

Crearemos el fichero

```
nano /etc/wireguard/iptables/ipup.sh
```

Y escribiremos lo siguiente

```
iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o ens18 -j MASQUERADE
iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o ens19 -j MASQUERADE
iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o ens20 -j MASQUERADE
iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o ens21 -j MASQUERADE
iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o ens22 -j MASQUERADE

iptables -A INPUT -s 10.9.0.0/24 -p udp -m udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT
ip6tables -A FORWARD -i wg0 -j ACCEPT; ip6tables -t nat -A POSTROUTING -o ens18 -j MASQUERADE;
ip6tables -A INPUT -s fd86:ea04:1115::0/64 -p udp -m udp --dport 53 -m conntrack --ctstate NEW -
```

Como podemos ver, ahora las interfaces ens18, ens19, ens20, ens21 y ens22, están en masquerade Ejecutamos un chmod +755 a ipup.sh

```
chmod +755 ipup.sh
```

Comentaremos el PostDown y agregaremos lo siguiente:

```
#PostDown = iptables -D FORWARD -i wg0 -j ACCEPT; iptables -t nat -D POSTROUTING -o ens18 -j MAS
PostDown = /etc/wireguard/iptables/ipdown.sh
```

Ahora crearemos el fichero para el PostDown

```
nano /etc/wireguard/iptables/ipdown.sh
```

E insertaremos el siguiente contenido

```
iptables -D FORWARD -i wg0 -j ACCEPT; iptables -t nat -D POSTROUTING -o ens18 -j MASQUERADE
iptables -D FORWARD -i wg0 -j ACCEPT; iptables -t nat -D POSTROUTING -o ens19 -j MASQUERADE
iptables -D FORWARD -i wg0 -j ACCEPT; iptables -t nat -D POSTROUTING -o ens20 -j MASQUERADE
iptables -D FORWARD -i wg0 -j ACCEPT; iptables -t nat -D POSTROUTING -o ens21 -j MASQUERADE
iptables -D FORWARD -i wg0 -j ACCEPT; iptables -t nat -D POSTROUTING -o ens22 -j MASQUERADE

iptables -D INPUT -s 10.9.0.0/24 -p udp -m udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT
ip6tables -D FORWARD -i wg0 -j ACCEPT; ip6tables -t nat -D POSTROUTING -o ens18 -j MASQUERADE;
ip6tables -D INPUT -s fd86:ea04:1115::0/64 -p udp -m udp --dport 53 -m conntrack --ctstate NEW -
```

Y procederemos a cambiar los permisos en dicho fichero

```
chmod +755 ipdown.sh
```

Configuración con otras conexiones

Ahora vamos a suponer que tenemos redes detrás de nuestras ip 192.168.30.0 y 192.168.40.0 que serán respectivamente las redes 192.168.31.0/24 y 192.168.41.0/24

En primer lugar modificaremos las interfaces de red, para ello modificaremos el archivo interfaces

```
nano /etc/network/interfaces
```

Y agregaremos las rutas detrás de las interfaces correspondientes

```
allow-hotplug ens18 ens19 ens20 ens21 ens22
iface ens18 inet static
address A.B.C.D/24
gateway A.B.C.X
# dns-* options are implemented by the resolvconf package, if installed
dns-nameservers 1.1.1.1 9.9.9.9 8.8.8.8
dns-search ateinco.net

iface ens19 inet static
address E.F.G.H/24

iface ens20 inet static
address I.J.K.L/24

iface ens21 inet static
address 192.168.30.254/24
up /bin/ip route add 192.168.31.0/24 via 192.168.30.1


iface ens22 inet static
address 192.168.40.254/24
up /bin/ip route add 192.168.41.0/24 via 192.168.40.1
```

Suponiendo que las redes se encuentren detrás de un router o un firewall en las ip 192.168.30.1 y 192.168.40.1 respectivamente.