Universidad Nacional de Quilmes

Laboratorio de Sistemas Operativos

Trabajo Práctico Final

WIREGUARD VPN

Integrantes

Acosta, Matías Ariel

Gallitelli, Nehuen

Profesor

José Luis Di Biase

2026

Índice

Contents

1 Introducción

1.1 ¿Qué es una VPN?

Una **VPN** (Virtual Private Network) es una tecnología que cifra tu conexión a Internet y oculta tu dirección IP.

1.2 ¿Para qué sirve una VPN?

Una VPN sirve principalmente para:

- **Privacidad y Seguridad**: Al ocultar tu dirección IP y cifrar tus datos, evitas que tu proveedor de Internet (ISP), empresas de publicidad o hackers en redes públicas rastreen tu actividad.
- **Superar bloqueos geográficos**: Si te conectas a un servidor VPN ubicado en otro país, los sitios web pensarán que estás físicamente allí. Esto permite acceder a contenidos, plataformas de streaming o servicios que no están disponibles en tu región.
- **Acceso remoto seguro**: Es muy común en el ámbito laboral o académico. Permite a los empleados o estudiantes conectarse a la red interna de una empresa o universidad desde sus casas de forma totalmente segura, como si estuvieran sentados en la oficina.

1.3 ¿Tiene alguna desventaja?

- **Pueden reducir un poco la velocidad**: Al tener que cifrar los datos y desviarlos a través de otro servidor, la velocidad de conexión puede disminuir ligeramente.
- **La confianza con el proveedor**: El proveedor de la VPN sí puede ver tu tráfico. Por eso, es fundamental elegir un servicio confiable y evitar VPNs de dudosa procedencia o un historial de datos comprometido.

1.4 ¿Cómo funciona una VPN?

En pocas palabras, se crea un "túnel" seguro para transmitir tus datos, protegiendo tu privacidad y seguridad frente a terceros, y permitiéndote simular una ubicación geográfica diferente.

---

2 WireGuard

2.1 ¿Qué es WireGuard?

WireGuard es un protocolo de red seguro para crear conexiones VPN. Es de código abierto y se destaca por estar diseñado para ser más rápido, más seguro y mucho más sencillo que alternativas tradicionales como IPsec u OpenVPN.

2.2 ¿Cómo funciona WireGuard?

WireGuard funciona bajo un concepto de intercambio de claves públicas y claves privadas.

Cada dispositivo en la red (pares o peers) tiene su propia clave. Para establecer el túnel, simplemente se configura la clave pública del servidor en el cliente y viceversa.

WireGuard trabaja principalmente en la capa de red, por lo que encapsula paquetes IP y los envía dentro de un túnel cifrado. Durante el handshake inicial, los peers se autentican con sus claves y generan claves de sesión temporales para cifrar el tráfico. Esto hace que la conexión sea rápida de levantar y que el intercambio de datos quede protegido desde el inicio.

Además, WireGuard usa la directiva `AllowedIPs` como una mezcla de tabla de enrutamiento y control de acceso. Esa configuración define qué direcciones se aceptan o se envían por cada peer, así que el servidor sabe por dónde reenviar cada paquete y qué cliente puede usar cada subred.

El servidor posee una tabla de enrutamiento que define cómo se redirigen los paquetes a través del túnel. Si un cliente cambia de red o de IP pública, WireGuard mantiene la asociación con su clave pública y puede seguir comunicándose sin tener que renegociar toda la conexión.

2.3 Características

- **Velocidad y rendimiento brutales**: Al estar integrado directamente en el kernel (el núcleo) de sistemas operativos como Linux, procesa los datos de manera nativa y rapidísima. Consume mucha menos CPU y batería en dispositivos móviles, y ofrece velocidades de descarga mucho mayores que OpenVPN.
- **Simplicidad de código**: Mientras que OpenVPN tiene alrededor de 400.000 líneas de código e IPsec supera las 600.000, WireGuard tiene menos de 4.000 líneas. Esto hace que sea más fácil de auditar para encontrar fallos o vulnerabilidades.
- **Conexión instantánea y resiliencia**: Se detectan cambios de red (por ejemplo, de WiFi a datos móviles), y WireGuard actualiza automáticamente la IP de origen del peer sin cortar la conexión, ya que asocia la sesión a la clave pública, no a la IP externa.
- **Criptografía moderna por defecto**: está implementado con lo más moderno y seguro de las tecnologías criptográficas actuales (como ChaCha20 y Poly1305).

3 Instalación Servidor

3.1 Requisitos previos

- Tener acceso de administrador o privilegios `root` en el servidor y en los clientes.
- Contar con `wireguard` y `wireguard-tools` instalados en cada equipo.
- Disponer de una IP pública en el servidor, o configurar redirección de puertos si está detrás de un router.
- Abrir el puerto UDP `51820` en el firewall y en el router donde esté alojado el servidor.
- Definir previamente la red privada de la VPN, por ejemplo `10.0.0.0/24`, para asignar una IP distinta a cada peer.
- Verificar que el kernel o sistema operativo tenga soporte para WireGuard.
- Tener conectividad a Internet en todos los dispositivos que formarán parte de la red.

3.2 Relevamiento de hardware

Gallitelli Nehuen

Dispositivo 1 (Servidor) - OS: Linux Mint 22.3 - Kernel: 6.14 - CPU: AMD Ryzen 7 - RAM: 16 GB #### Dispositivo 2 - OS: iOS

Acosta Matías

Dispositivo 3 - OS: Arch Linux x86_64 - Kernel: Linux 7.1 - CPU: Intel Celeron N3050 - RAM: 4 GB #### Dispositivo 4 - OS: Android 16

3.3 1: Instalar WireGuard en el Host

Varía según el sistema operativo y la distribución.

En Linux (Arch Linux): El módulo WireGuard ya viene incluido en el kernel de Arch (desde la versión 5.6). Solo se necesitan las herramientas de usuario:

```
sudo pacman -S wireguard-tools
```

En Debian/Ubuntu:

```
sudo apt install wireguard wireguard-tools
```

3.4 2: Generar las claves pública/privada

```
# Entrar al directorio de configuración
cd /etc/wireguard/

# Generar clave privada
wg genkey | sudo tee privatekey | wg pubkey | sudo tee publickey
```

Para ver las claves generadas:

```
cat privatekey
cat publickey
```

3.5 3: Configurar WireGuard (Servidor)

WireGuard trabaja añadiendo una interfaz de red virtual a tu sistema, que se conecta a tu red física y permite el enrutamiento de tráfico VPN.

Por defecto, el nombre de la interfaz es `wg0` y debemos configurarla, para eso debemos crear el archivo `/etc/wireguard/wg0.conf` con el siguiente contenido:

```
[Interface]
# Dirección IP que tendrá el servidor dentro de la red VPN
Address = 10.0.0.1/24
# Puerto UDP donde escucha wireguard
ListenPort = 51820
# Clave privada del servidor
PrivateKey = <CONTENIDO_DE_SERVER.PRIVATE>

# Reenvío de tráfico (NAT) para darles salida a Internet a los clientes (reenvío de tráfico)
# Para este caso, la interfaz eth0 es la de salida a Internet
PostUp = iptables -A FORWARD -i %i -j ACCEPT;
iptables -A FORWARD -o %i -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
PostDown = iptables -D FORWARD -i %i -j ACCEPT;
iptables -D FORWARD -o %i -j ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE
```

3.6 4. Activar el reenvío de IP (IP Forwarding) en el Servidor

Para que los clientes puedan navegar por Internet usando tu PC como gateway, el servidor debe permitir el reenvío de paquetes:

```
# Temporalmente
sudo sysctl -w net.ipv4.ip_forward=1

# Persistente
echo "net.ipv4.ip_forward=1" | sudo tee -a /etc/sysctl.conf
```

3.7 5. Levantar la VPN

```
# Levantar la interfaz
sudo wg-quick up wg0

# Ver el estado (conexiones, transferencia, handshake)
sudo wg show

# Ver la interfaz de red
ip a show wg0

# Para apagarla
sudo wg-quick down wg0
```

3.8 6. Agregar un Cliente

Para agregar un cliente, se debe agregar la configuración del cliente en el archivo `wg0.conf` del servidor.

```
[Peer]
# Cliente 1
PublicKey = <PUBLIC_KEY_DEL_CLIENTE_1>
# IPs que este cliente podrá usar dentro de la VPN
AllowedIPs = 10.0.0.2/24

[Peer]
# Cliente 2
PublicKey = <PUBLIC_KEY_DEL_CLIENTE_2>
AllowedIPs = 10.0.0.3/24
```

Importante reiniciar el servicio ante cambios en el archivo de configuración

`sudo wg-quick down wg0 && sudo wg-quick up wg0`

4 Instalación Cliente

4.1 1. Instalación del Cliente

En Arch Linux:

```
sudo pacman -S wireguard-tools
```

En Debian/Ubuntu:

```
sudo apt install wireguard wireguard-tools
```

4.2 2. Generar claves del Cliente

Cada cliente debe tener su propio par de claves.

```
cd /etc/wireguard/
wg genkey | sudo tee privatekey | wg pubkey | sudo tee publickey

# Clave pública
sudo cat publickey
```

4.3 3. Configurar el Cliente WireGuard

El cliente debe crear un archivo:

```
sudo nano /etc/wireguard/wg0.conf
```

Con la siguiente estructura:

```
[Interface]
# Clave PRIVADA del cliente
PrivateKey =
# Dirección IP asignada por el servidor para el cliente
Address =

[Peer]
# Clave PÚBLICA del SERVIDOR
PublicKey = <PUBLIC_KEY_DEL_SERVER>

# Dirección IP pública o dominio de tu servidor
Endpoint = <IP_PUBLICA_DEL_SERVER>:51820

# Qué destinos van encriptados a través del túnel.
AllowedIPs = 0.0.0.0/0 # Todos los destinos
```

4.4 4. Iniciar conexión

Para iniciar la conexión, se debe ejecutar:

```
sudo wg-quick up wg0
```

5 Dentro de la VPN

Decidimos simular un entorno empresarial/académico para mostrar un caso de uso real de la VPN. Dentro del túnel montamos una pequeña red interna con una página web local accesible solo para los clientes conectados y con algunos archivos compartidos para trabajo colaborativo.

La idea fue representar un escenario similar al de una intranet: el servidor cumple el rol de puerta de acceso a recursos internos, y los clientes se conectan como si estuvieran dentro de la misma red física. De esta forma, la VPN no solo cifra el tráfico, sino que también permite acceder a servicios privados sin exponerlos directamente a Internet.

En la práctica, esto sirve para probar que un equipo remoto puede ver la página interna, consultar documentos compartidos y utilizar recursos de la red privada mediante las IP asignadas por WireGuard.

5.1 Aclaraciones técnicas

- **Puerto en el Router (NAT):** Si el servidor está detrás de un router doméstico, se debe configurar el router para redirigir el puerto UDP 51820 hacia la IP local del servidor (ej: 192.168.1.10). Si no se hace esto, los clientes no podrán conectarse desde fuera de la red local.
  - **Firewall local (UFW/iptables):** Asegurar que el puerto UDP 51820 esté abierto en el firewall del servidor.

Cómo redirigir puerto

Para que una conexión desde afuera de la red local llegue al servidor, el router debe traducir el tráfico entrante hacia la IP privada de la máquina donde corre WireGuard. En la mayoría de los casos esto se configura con una regla de port forwarding que redirige UDP `51820` a la IP local del servidor VPN.

Si el servidor cambia de dirección dentro de la red local, conviene reservarle una IP fija o una reserva DHCP para no tener que modificar la regla del router cada vez.

6 Problemáticas encontradas

1. Conexión a la VPN dentro de la red local

- Descripción: Tras la configuración inicial, no había conectividad entre clientes y servidor dentro de la misma red local.
- Causa: El puerto UDP `51820` estaba cerrado por el firewall del servidor.
- Solución: Abrir el puerto en UFW: `sudo ufw allow 51820/udp`.

2. Acceso desde fuera de la red local

- Descripción: No se logró conectar desde Internet pese a configurar el reenvío de puertos en el router.
- Acciones realizadas:
  - Verificamos que la IP pública fuera estática y que no estuviéramos detrás de CGNAT.
  - Configuramos el port forwarding del UDP `51820` en el router.
  - Probamos habilitar DMZ en el router para exponer el servidor.
- Resultado: El acceso externo continuó fallando. Posibles causas: bloqueo por parte del proveedor de Internet, restricción de NAT no documentada o configuración adicional del router.
- Decisión: Se limitaron las pruebas a conexiones desde la red local para avanzar con el resto del trabajo.

3. Errores menores y permisos

- Problemas: Errores tipográficos en archivos de configuración y dificultades para acceder a carpetas que requieren privilegios `root`.
- Recomendaciones: Revisar y validar configuraciones antes de aplicar cambios, usar `sudo` cuando sea necesario o ajustar permisos, y documentar los comandos y pasos críticos.