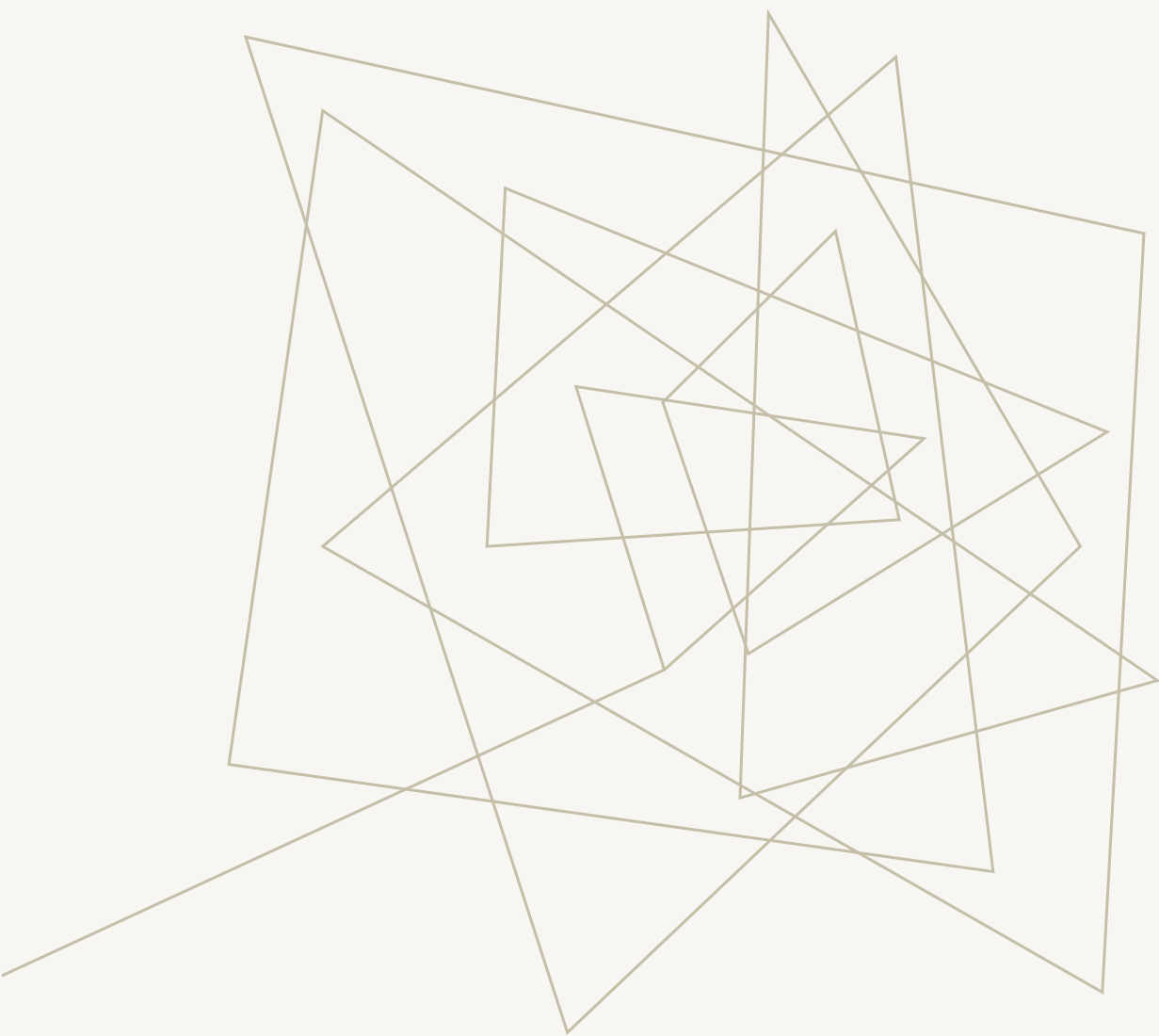




PRÁCTICA 1 VPN

Paloma Pérez de Madrid e Inés del Río



Nota: El apartado de la configuración del cliente y servidor no se adecua a lo que pone en el examen. Si ves “apuntes paloma” está la configuración del cliente y del servidor de forma más precisa.

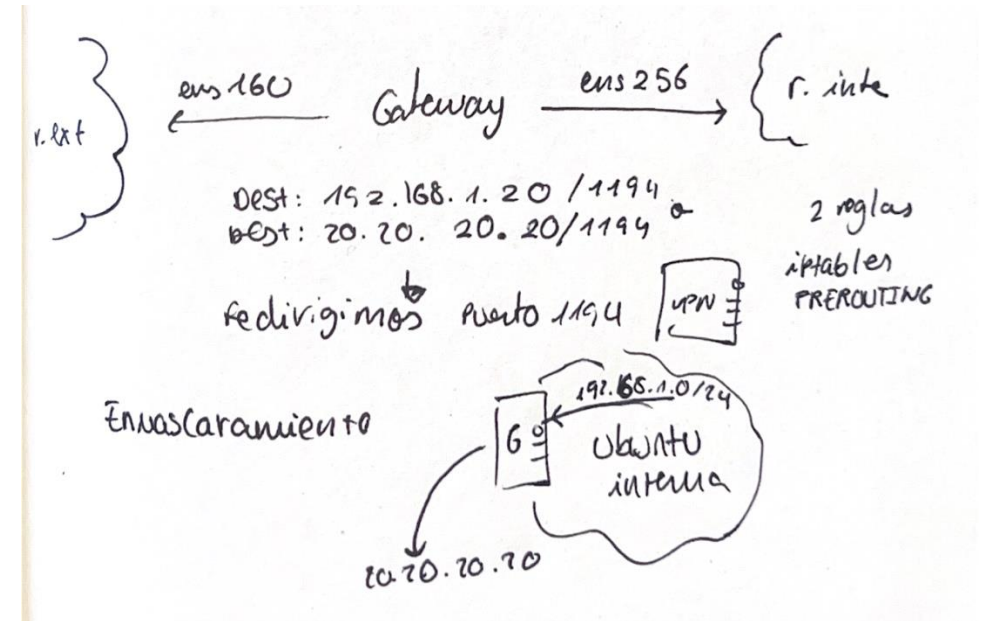
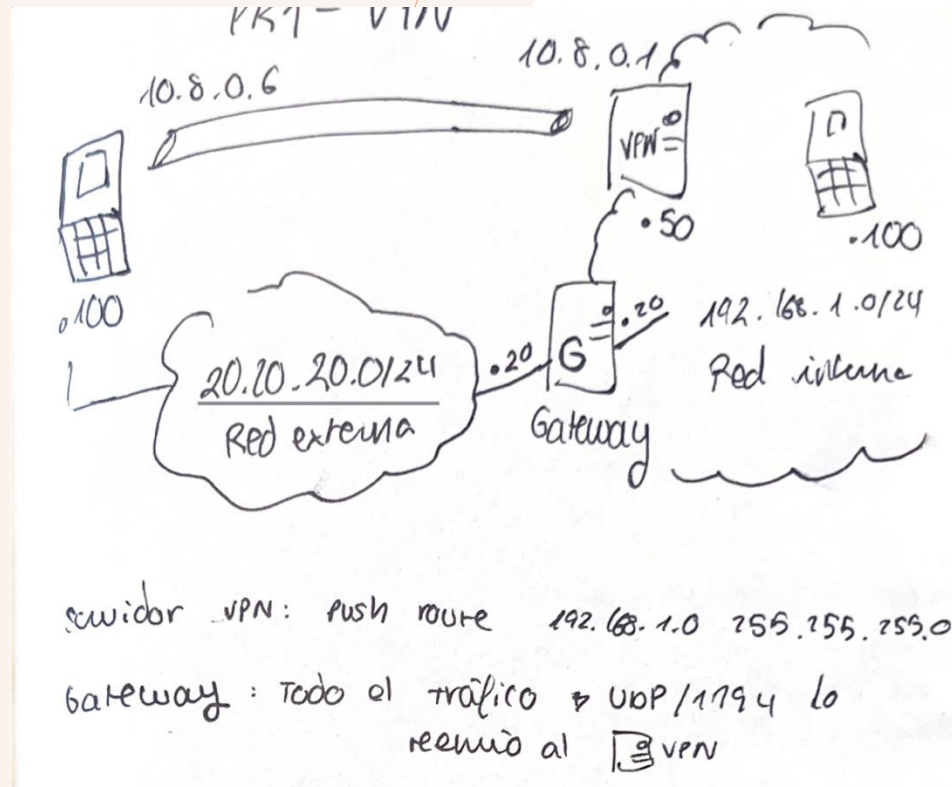
ÍNDICE

1. Creación de maqueta propuesta
2. Creación de Autoridad de Certificación
3. Configuración servidor OpenVPN modo TUN
4. Configuración Gateway port-forwarding
5. Configuración Gateway enmascaramiento
6. Configuración OpenVPN cliente PC1
7. Verificación funcionamiento
8. Comprobación alcance PC2 a PC1
9. Modificación PC2 para que pueda comunicar con PC1 a través de la VPN
10. (Extra) Configuración modo TAP del servidor
11. (Extra) Estudiar características de la solución VPN WireGuard

ÍNDICE

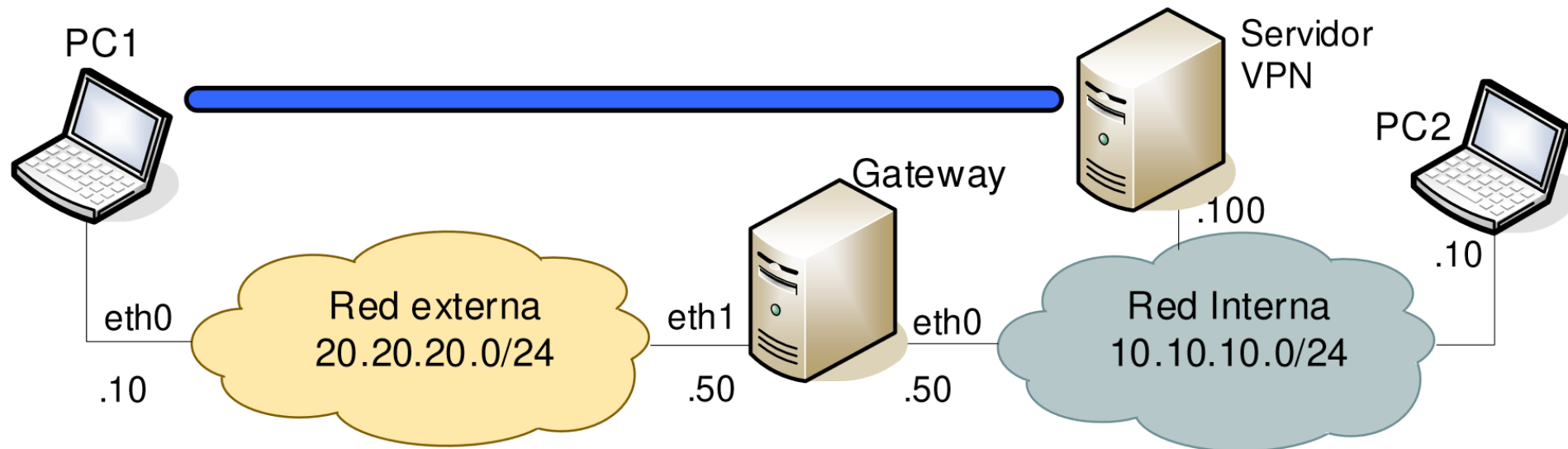
1. **Creación de maqueta propuesta**
2. Creación de Autoridad de Certificación
3. Configuración servidor OpenVPN modo TUN
4. Configuración Gateway port-forwarding
5. Configuración Gateway enmascaramiento
6. Configuración OpenVPN cliente PC1
7. Verificación funcionamiento
8. Comprobación alcance PC2 a PC1
9. Modificación PC2 para que pueda comunicar con PC1 a través de la VPN
10. (Extra) Configuración modo TAP del servidor
11. (Extra) Estudiar características de la solución VPN WireGuard

Esquema de "Apuntes Paloma"

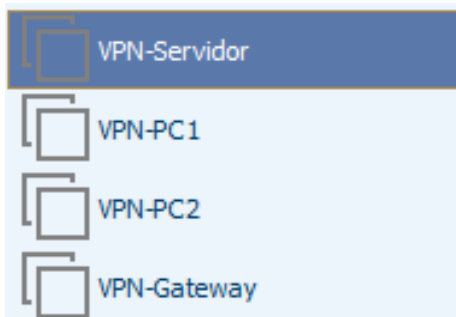
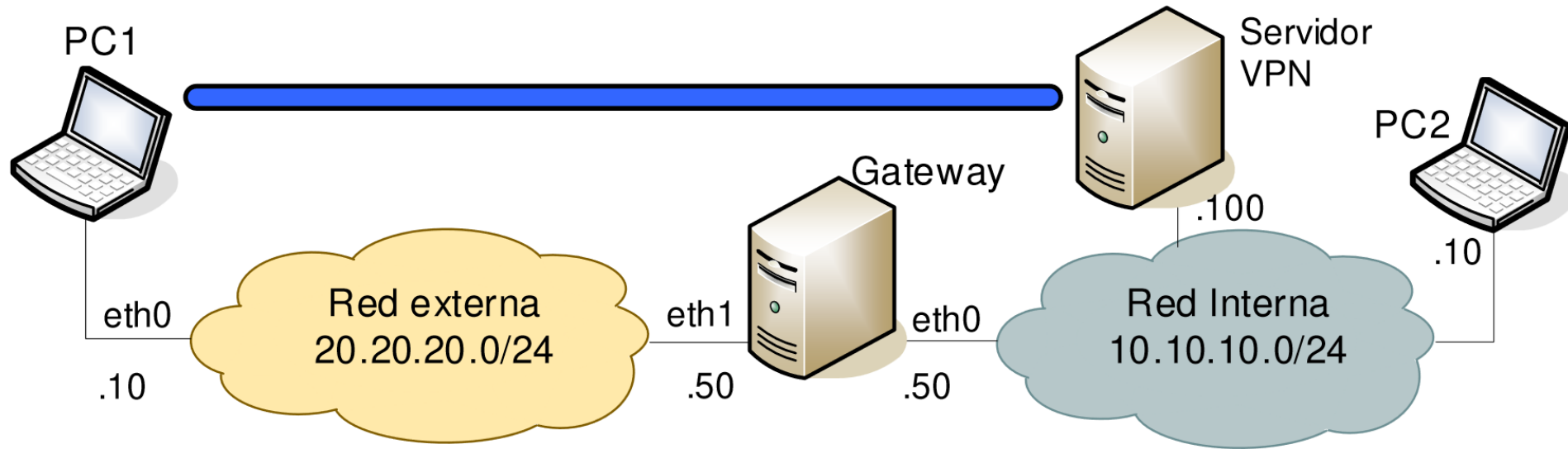


1. CREACIÓN DE MAQUETA PROPUESTA

Crear el escenario virtualizado propuesto. Configurar las direcciones IP de todos los interfaces y activar el forwarding en el Gateway. Establecer como router por defecto el Gateway en los restantes equipos y comprobar la conectividad. Instalar algún servicio en los equipos PC1, PC2 y Servidor (servidor SSH, Apache, ...).



1. CREACIÓN DE MAQUETA PROPUESTA



Comenzamos creando 4 máquinas virtuales y añadiéndole un interfaz adicional a la máquina Gateway

VPN-Gateway, segundo adaptador en modo Host-Only (eth0)

The screenshot shows the 'Device status' and 'Network connection' settings for a VM. In the 'Device status' section, 'Network Adapter 2' is listed with a status of 'NAT'. In the 'Network connection' section, 'Bridged: Connected directly to the physical network' is selected, and 'Replicate physical network connection state' is unchecked. Below this, 'NAT: Used to share the host's IP address' is selected, and 'Host-only: A private network shared with the host' is unselected.

1. CREACIÓN DE MAQUETA PROPUESTA

Configurar las interfaces de VPN-Gateway

`ip link show`: ver direcciones actuales

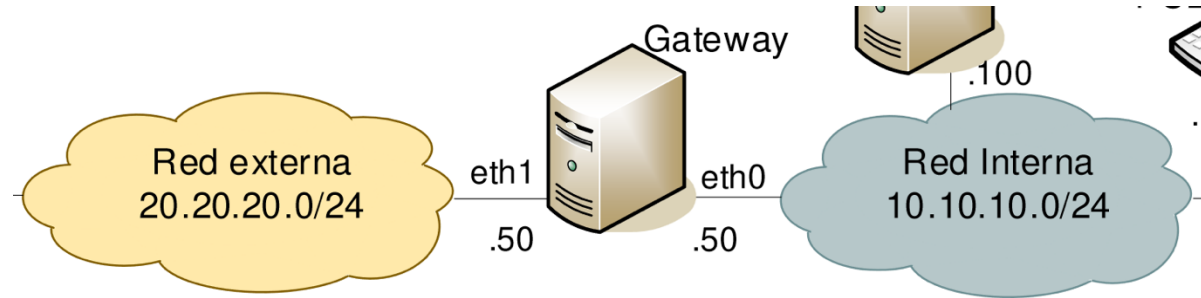
Interfaz eth0 (10.10.10.50):

`vim /etc/netplan/00-installer-config.yaml`

```
network:
  version: 2
  ethernets:
    eth0:
      # dhcp4: true
      dhcp4: true
      addresses: [10.10.10.50/24]
      nameservers:
        addresses: [8.8.8.8]
    eth1:
      dhcp4: true
      addresses: [20.20.20.50/24]
      nameservers:
        addresses: [8.8.8.8]
```

Guardamos los cambios:

`netplan apply`



Configuramos el Reenvío de IP

`vim /etc/sysctl.conf`

Añadimos "`net.ipv4.ip_forward=1`"

Aplicamos el cambio con:

`sysctl -p`

Reenvío IP activado
(forwarding)

```
net.ipv4.ip_forward=1
"/etc/sysctl.conf" 72L, 2382B escritos
root@server:/home# sysctl -p
net.ipv4.ip_forward = 1
```


1. CREACIÓN DE MAQUETA PROPUESTA

Configurar las interfaces de VPN-Servidor

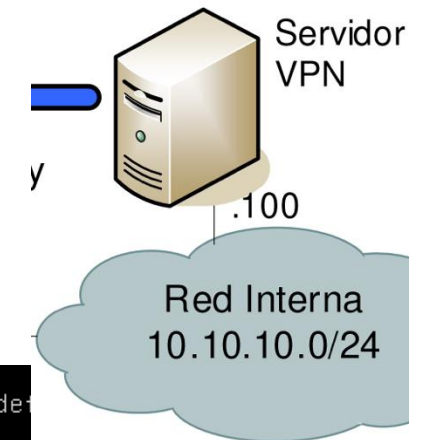
Interfaz eth0 (10.10.10.10):

vim /etc/netplan/00-installer-config.yaml

```
network:
  version: 2
  ethernets:
    eth0:
      dhcp4: true
      addresses: [20.20.20.10/24]
      routes:
        - to: default
          via: 20.20.20.50
      nameservers:
        _ addresses: [8.8.8.8]
```

```
root@server:~# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def
    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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP grou
    link/ether 00:0c:29:57:8d:fc brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    altname ens33
    inet 10.10.10.100/24 brd 10.10.10.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe57:8dfc/64 scope link
        valid_lft forever preferred_lft forever
```

netplan apply



1. CREACIÓN DE MAQUETA PROPUESTA

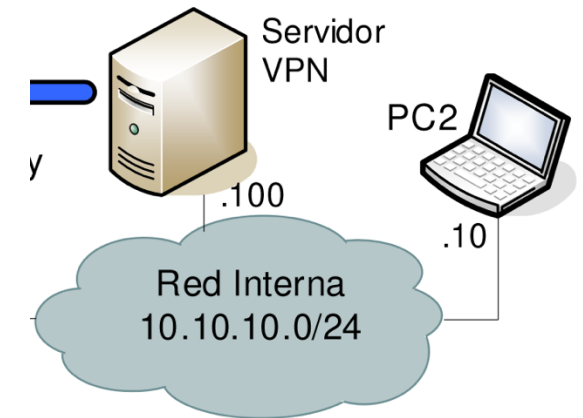
Configurar las interfaces de VPN-PC2

Interfaz eth0 (10.10.10.10):

```
# vim /etc/netplan/00-installer-config.yaml
```

```
network:
  version: 2
  ethernets:
    eth0:
      dhcp4: true
      addresses: [20.20.20.10/24]
      routes:
        - to: default
          via: 20.20.20.50
      # nameservers:
      #   addresses: [8.8.8.8]
```

```
# netplan apply
```



```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
p default qlen 1000
    link/ether 00:0c:29:3c:2e:a1 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    altname ens33
    inet 10.10.10.10/24 brd 10.10.10.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe3c:2ea1/64 scope link
        valid_lft forever preferred_lft forever
```

1. CREACIÓN DE MAQUETA PROPUESTA

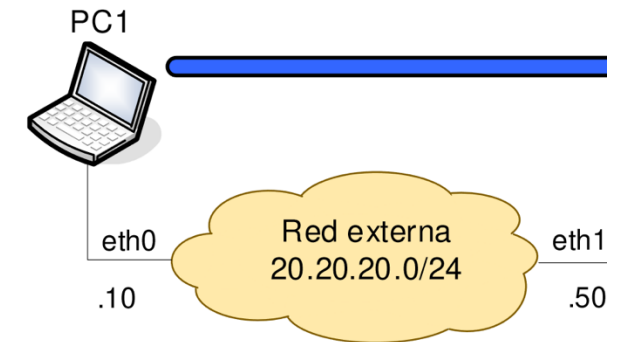
Configurar las interfaces de VPN-PC1

Interfaz eth0 (10.10.10.10):

```
# vim /etc/netplan/00-installer-config.yaml
```

```
network:
  version: 2
  ethernets:
    eth0:
      _dhcp4: true
      # dhcp4: no
      addresses: [10.10.10.10/24]
      routes:
        - to: default
          via: 10.10.10.50
      nameservers:
        addresses: [8.8.8.8]
```

```
# netplan apply
```



```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gro
p default qlen 1000
    link/ether 00:0c:29:de:7e:36 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    altname ens33
    inet 20.20.20.10/24 brd 20.20.20.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fedc:7e36/64 scope link
        valid_lft forever preferred_lft forever
```

1. CREACIÓN DE MAQUETA PROPUESTA

Instalar Servicios en PC1, PC2 y Servidor

PC1, PC2 , servidor → Servidores Apache

apt update

apt install apache2

systemctl start apache2

systemctl enable apache2

Cambiamos el archivo index.html y
comprobamos que el servidor funciona:

curl localhost

```
root@server:/var/www/html# curl localhost
Este es el Servidor VPN (10.10.10.100)
root@server:/var/www/html#
```

```
root@server:/var/www/html# curl localhost
Este es el PC2 (10.10.10.10)
root@server:/var/www/html#
```

```
index.html - [nuevo] LL, 888 caracteres
root@server:/var/www/html# curl localhost
Este es el PC1 (20.20.20.10)

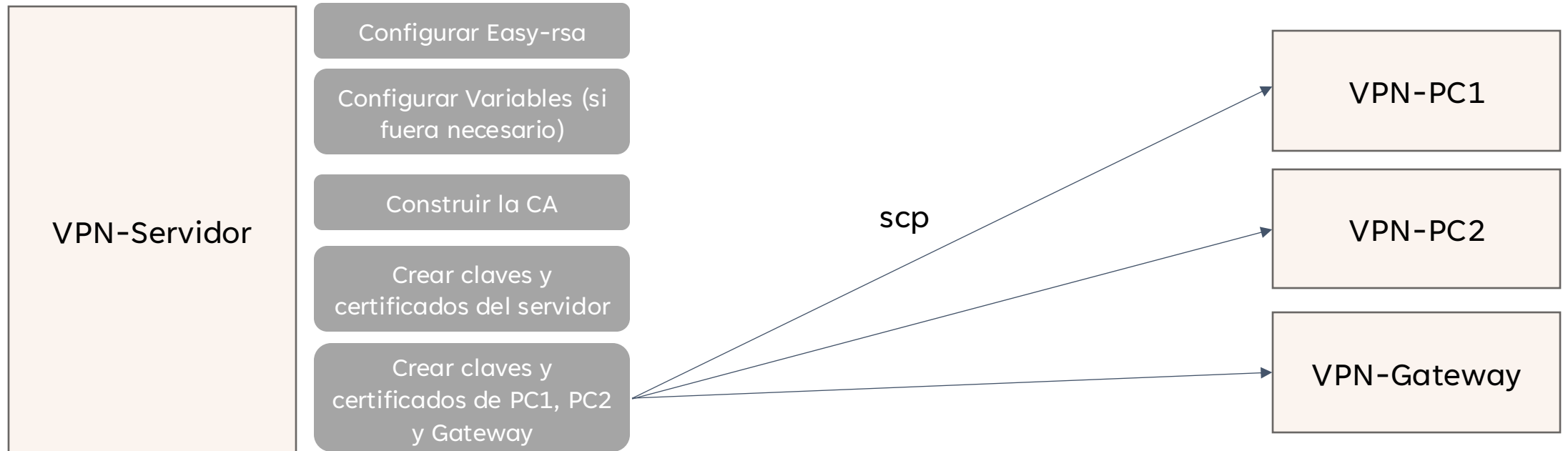
root@server:/var/www/html#
```

ÍNDICE

1. Creación de maqueta propuesta
2. **Creación de Autoridad de Certificación**
3. Configuración servidor OpenVPN modo TUN
4. Configuración Gateway port-forwarding
5. Configuración Gateway enmascaramiento
6. Configuración OpenVPN cliente PC1
7. Verificación funcionamiento
8. Comprobación alcance PC2 a PC1
9. Modificación PC2 para que pueda comunicar con PC1 a través de la VPN
10. (Extra) Configuración modo TAP del servidor
11. (Extra) Estudiar características de la solución VPN WireGuard

2. CREACIÓN DE AUTORIDAD DE CERTIFICACIÓN

Crear una Autoridad de Certificación en VPN-Servidor y emitir los elementos necesarios para nuestra configuración (valores DH y claves/certificados para el servidor y cliente)



2. CREACIÓN DE AUTORIDAD DE CERTIFICACIÓN

VPN-Servidor

Configurar Easy-rsa

```
# apt install easy-rsa
```

Vamos a cambiar ubicación para trabajar con más comodidad

```
# mkdir -p ~/easy-rsa
```

```
# cp -r /usr/share/easy-rsa/* ~/easy-rsa/
```

```
# cd ~/easy-rsa
```

```
# ./easyrsa init-pki
```

```
root@server:~/easy-rsa# ls
easyrsa  openssl-easyrsa.cnf  vars.example  x509-types
root@server:~/easy-rsa# ./easyrsa init-pki

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /root/easy-rsa/pki
```

2. CREACIÓN DE AUTORIDAD DE CERTIFICACIÓN

VPN-Servidor

Construir la CA

./easysrsa build-ca
(Key Passphrase: pr1)

./easysrsa gen-dh

Archivo de certificado de intercambio DH
(Diffie-Hellman)

- ca.crt → Certificado de la CA
- ca.key → Clave privada CA (tiene que estar muy protegida)

```
root@server:~/easy-rsa# ./easysrsa build-ca
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)

Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:pr1

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/root/easy-rsa/pki/ca.crt
```

```
+*****+
+*****+
+*****+
DH parameters of size 2048 created at /root/easy-rsa/pki/dh.pem
```


2. CREACIÓN DE AUTORIDAD DE CERTIFICACIÓN

VPN-Servidor

Crear claves y certificados del servidor

Crear claves y certificados de PC1, PC2 y Gateway

```
# ./easyrsa build-server-full vpnservidor nopass
```

```
# ./easyrsa build-client-full pc1 nopass
```

```
# ./easyrsa build-client-full pc2 nopass
```

```
# ./easysrsa build-client-full Gateway nopass
```

Yo no he puesto “nopass”, la “PEM Pass” será

- PC1: clientenum1
- PC2: clientenum2
- Gateway: clientgateway

[illegible]

pr1

[illegible]

PR1-VPN

2. CREACIÓN DE AUTORIDAD DE CERTIFICACIÓN

VPN-Servidor

Crear claves y
certificados del servidor

Crear claves y certificados
de PC1, PC2 y Gateway

Archivos generados:

```
root@server:~/easy-rsa# cd pki
root@server:~/easy-rsa/pki# cd issued
root@server:~/easy-rsa/pki/issued# ls
gateway.crt pc1.crt pc2.crt vpnservidor.crt
root@server:~/easy-rsa/pki/issued#
```

Pc1.crt → certificado del cliente que se utiliza
para autenticar al cliente ante el servidor VPN

```
root@server:~/easy-rsa/pki# ls
ca.crt          index.txt.attr      openssl-easyrsa.cnf  revoked
certs_by_serial index.txt.attr.old  private              safessl-easyrsa.cnf
dh.pem          index.txt.old       renewed              serial
index.txt       issued              reqs                 serial.old
root@server:~/easy-rsa/pki# cd private/
root@server:~/easy-rsa/pki/private# ls
ca.key gateway.key pc1.key pc2.key vpnservidor.key
```

1. Instalar OpenVPN en VPN y PC1 → `# apt install openvpn`
2. Copiar las claves (*.crt ,*.key) de cada cliente en su carpeta
/etc/openvpn

Ejemplo con Gateway:

```
# scp pki/private/gateway.key root@10.10.10.50:/etc/openvpn/
# scp pki/issued/gateway.crt root@10.10.10.50:/etc/openvpn/
```

Haremos lo mismo con PC1 y PC2

```
root@server:~/easy-rsa# scp pki/private/gateway.key root@10.10.10.50:/etc/openvpn/
The authenticity of host '10.10.10.50 (10.10.10.50)' can't be established.
ED25519 key fingerprint is SHA256:BcEa6rJBR3Q0fIe/YtgqRyPIbcm7w2lUjLQqiQ/Z59Y.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.50' (ED25519) to the list of known hosts.
root@10.10.10.50's password:
gateway.key                                100% 1854    213.6KB/s   00:00
root@server:~/easy-rsa# scp pki/issued/gateway.crt root@10.10.10.50:/etc/openvpn/
root@10.10.10.50's password:
gateway.crt                                100% 4458    420.7KB/s   00:00
root@server:~/easy-rsa# _
```

2. CREACIÓN DE AUTORIDAD DE CERTIFICACIÓN

VPN-Servidor

```
root@server:~/easy-rsa# scp pki/private/pc1.key root@20.20.20.10:/etc/openvpn
root@20.20.20.10's password:
pc1.key                                100% 1854    166.2KB/s   00:00
root@server:~/easy-rsa# scp pki/issued/pc1.crt root@20.20.20.10:/etc/openvpn
root@20.20.20.10's password:
pc1.crt                                100% 4450    245.0KB/s   00:00
root@server:~/easy-rsa# scp pki/issued/pc2.crt root@10.10.10.10:/etc/openvpn
root@10.10.10.10's password:
pc2.crt                                100% 4446    311.2KB/s   00:00
root@server:~/easy-rsa# scp pki/private/pc2.key root@10.10.10.10:/etc/openvpn
root@10.10.10.10's password:
pc2.key                                100% 1854    202.0KB/s   00:00
```

Nota: también copiamos a la carpeta /etc/openvpn los siguientes archivos:

- ca.crt
- vpnservidor.crt
- vpnservidor.key
- dh.pem

Nota: copiar ca.crt en los clientes

PC1

```
root@server:/etc/openvpn# ls
client  pc1.crt  pc1.key  server  update-resolv-conf
```

PC2

```
root@server:/etc/openvpn# ls
client  pc2.crt  pc2.key  server  update-resolv-conf
```

Gateway

```
root@server:/etc/openvpn# ls
client  gateway.crt  gateway.key  server  update-res
```

ÍNDICE

1. Creación de maqueta propuesta
2. Creación de Autoridad de Certificación
3. **Configuración servidor OpenVPN modo TUN**
4. Configuración Gateway port-forwarding
5. Configuración Gateway enmascaramiento
6. Configuración OpenVPN cliente PC1
7. Verificación funcionamiento
8. Comprobación alcance PC2 a PC1
9. Modificación PC2 para que pueda comunicar con PC1 a través de la VPN
10. (Extra) Configuración modo TAP del servidor
11. (Extra) Estudiar características de la solución VPN WireGuard

3. CONFIGURACIÓN SERVIDOR OPENVPN MODO TUN

Configurar el servidor OpenVPN en modo TUN (actuará como router), usando el archivo server.conf). Usaremos UDP como protocolo de transporte. Configurar su arranque automático al inicio y arrancar el servicio. También habilitaremos la función de forwarding en este equipo Servidor, para que pueda encaminar el tráfico entre sus interfaces.

vim /etc/openvpn/server/servidorvpn.conf

```
port 1194
proto udp
dev tun      # modo TUN

ca /etc/openvpn/ca.crt
cert /etc/openvpn/vpnservidor.crt
key /etc/openvpn/vpnservidor.key
dh /etc/openvpn/dh.pem

# subred VPN
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
```

```
# Rutas y reenvío de tráfico a la red local (10.10.10.0/24)
push "route 10.10.10.0 255.255.255.0"

keepalive 10 120

data-ciphers AES-256-CBC
tls-auth ta.key 0
key-direction 0

persist-key
persist-tun
```

cuando un cliente se conecta a la VPN, el servidor le indicará que todo el tráfico destinado a la red **10.10.10.0/24** (la red interna) debe enviarse a través del túnel VPN.

openvpn --genkey --secret ta.key

Nota: Me ha fallado porq no he metido el archivo de config en la carpeta server de openvpn

3. CONFIGURACIÓN SERVIDOR OPENVPN MODO TUN

Configurar el servidor OpenVPN en modo TUN (actuará como router), usando el archivo server.conf). Usaremos UDP como protocolo de transporte. Configurar su arranque automático al inicio y arrancar el servicio. También habilitaremos la función de forwarding en este equipo Servidor, para que pueda encaminar el tráfico entre sus interfaces.

Configurar el arranque automático al inicio y al arrancar el servicio.

`systemctl start openvpn-server@servidorvpn` (“servidorvpn” es el nombre de “servidorvpn.conf”)

`systemctl enable openvpn-server@servidorvpn`

```
root@server:/etc/openvpn/server# systemctl status openvpn-server@servidorvpn
● openvpn-server@servidorvpn.service - OpenVPN service for servidorvpn
   Loaded: loaded (/lib/systemd/system/openvpn-server@.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-09-18 06:18:06 UTC; 1min 23s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
  Main PID: 45943 (openvpn)
    Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 970)
    Memory: 2.8M
       CPU: 165ms
   CGroup: /system.slice/system-openvpn\x2dslice/openvpn-server@servidorvpn.service
           └─45943 /usr/sbin/openvpn --status /run/openvpn-server/status-servidorvpn.log --statu
```

3. CONFIGURACIÓN SERVIDOR OPENVPN MODO TUN

Configurar el servidor OpenVPN en modo TUN (actuará como router), usando el archivo server.conf). Usaremos UDP como protocolo de transporte. Configurar su arranque automático al inicio y arrancar el servicio. También habilitaremos la función de forwarding en este equipo Servidor, para que pueda encaminar el tráfico entre sus interfaces.

Configuramos el Reenvío de IP

```
# vim /etc/sysctl.conf
```

Añadimos “net.ipv4.ip_forward=1”

Aplicamos el cambio con:

```
# sysctl -p
```

```
#net.ipv6.conf.all.forwarding=1  
net.ipv4.ip_forward=1
```

```
root@server:/etc/openvpn/server# sysctl -p  
net.ipv4.ip_forward = 1
```

ÍNDICE

1. Creación de maqueta propuesta
2. Creación de Autoridad de Certificación
3. Configuración servidor OpenVPN modo TUN
4. **Configuración Gateway port-forwarding**
5. Configuración Gateway enmascaramiento
6. Configuración OpenVPN cliente PC1
7. Verificación funcionamiento
8. Comprobación alcance PC2 a PC1
9. Modificación PC2 para que pueda comunicar con PC1 a través de la VPN
10. (Extra) Configuración modo TAP del servidor
11. (Extra) Estudiar características de la solución VPN WireGuard

4. CONFIGURACIÓN GATEWAY PORT-FORWARDING

Configurar en el Gateway un port-forwarding para reenviar al servidor interno todo el tráfico recibido en el puerto udp/1194 de su interface externo (es necesaria una regla iptables nat en PREROUTING).

Configuramos regla de iptables:

```
# iptables -t nat -A PREROUTING -p udp --dport 1194 -j DNAT --to-destination 10.10.10.100:1194
```

Desglose:

- -t nat: Especifica que estamos trabajando con la tabla nat.
- -A PREROUTING: Agrega la regla a la cadena PREROUTING.
- -p udp: Indica que la regla se aplica al tráfico UDP.
- --dport 1194: Especifica el puerto de destino en el tráfico entrante.
- -j DNAT: Redirige el tráfico al destino especificado.
- --to-destination 10.10.10.10:1194: Redirige el tráfico al servidor interno en la dirección 10.10.10.10 y puerto 1194.

4. CONFIGURACIÓN GATEWAY PORT-FORWARDING

Configurar en el Gateway un port-forwarding para reenviar al servidor interno todo el tráfico recibido en el puerto udp/1194 de su interface externo (es necesaria una regla iptables nat en PREROUTING).

```
# iptables -t nat -A PREROUTING -p udp -d 20.20.20.50 --dport 1194 -j DNAT --to-destination 10.10.10.100:1194
```

```
# iptables -t nat -L -v (verificar reglas)
```

```
root@server:~# iptables -t nat -L -v
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source    destination
    6   174 DNAT      udp  --  eth1   any     anywhere  anywhere
    udp dpt:openvpn to:10.10.10.100:1194
```

Lo hacemos persistentes:

```
# apt-get install iptables-persistent
```

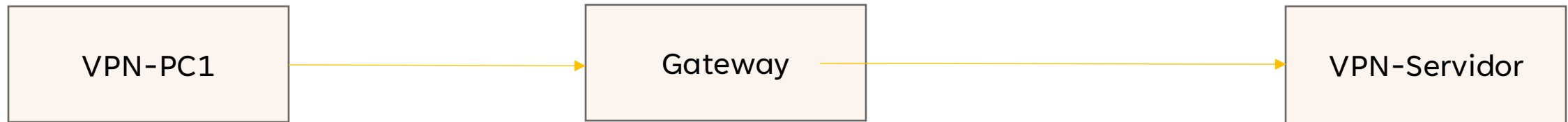
```
# iptables-save > /etc/iptables-rules
```

```
Restore → iptables-restore < /etc/iptables-rules
```

Alternativa:

```
# netfilter-persistent save | reload
```

4. CONFIGURACIÓN GATEWAY PORT-FORWARDING



nc -vzv 20.20.20.50 1194



tcpdump -i eth1 udp port 1194

Capturar tráfico en interfaz interna



```
root@server:~# nc -vzu 20.20.20.50 1194
Connection to 20.20.20.50 1194 port [udp/openvpn] succeeded!
root@server:~# |
```

```
root@server:~# tcpdump -i eth1 udp port 1194
tcpdump: verbose output suppressed, use -v[v]... for full protocol decoding
listening on eth1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
19:09:42.755952 IP 20.20.20.10.40039 > server.openvpn: UDP, length 1
19:09:42.755981 IP 20.20.20.10.40039 > server.openvpn: UDP, length 1
19:09:42.756888 IP 20.20.20.10.40039 > 10.10.10.100.openvpn: UDP, length 1
19:09:42.756916 IP 20.20.20.10.40039 > 10.10.10.100.openvpn: UDP, length 1
19:09:43.758228 IP 20.20.20.10.40039 > server.openvpn: UDP, length 1
19:09:43.758997 IP 20.20.20.10.40039 > 10.10.10.100.openvpn: UDP, length 1
19:09:44.759777 IP 20.20.20.10.40039 > server.openvpn: UDP, length 1
19:09:44.760738 IP 20.20.20.10.40039 > 10.10.10.100.openvpn: UDP, length 1
19:09:45.761561 IP 20.20.20.10.40039 > server.openvpn: UDP, length 1
19:09:45.762225 IP 20.20.20.10.40039 > 10.10.10.100.openvpn: UDP, length 1
```

ÍNDICE

1. Creación de maqueta propuesta
2. Creación de Autoridad de Certificación
3. Configuración servidor OpenVPN modo TUN
4. Configuración Gateway port-forwarding
5. **Configuración Gateway enmascaramiento**
6. Configuración OpenVPN cliente PC1
7. Verificación funcionamiento
8. Comprobación alcance PC2 a PC1
9. Modificación PC2 para que pueda comunicar con PC1 a través de la VPN
10. (Extra) Configuración modo TAP del servidor
11. (Extra) Estudiar características de la solución VPN WireGuard

5. CONFIGURACIÓN GATEWAY ENMASCARAMIENTO

Configurar en el Gateway el enmascaramiento para todo el tráfico saliente Comprobar que siguen pudiendo alcanzarse recursos externos desde la red interna (PC2 a PC1). Verificar que se realiza el enmascaramiento usando tcpdump.

iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE

- -A POSTROUTING: Agrega una regla a la cadena POSTROUTING.
- -j MASQUERADE: Realiza el enmascaramiento de las direcciones IP de origen de los paquetes salientes.

```
root@server:/etc/openvpn/server# iptables -t nat -L -v -n
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source               destination
     4  116 DNAT      udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp dpt:1194 to:10.10.10.10:1194

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source               destination

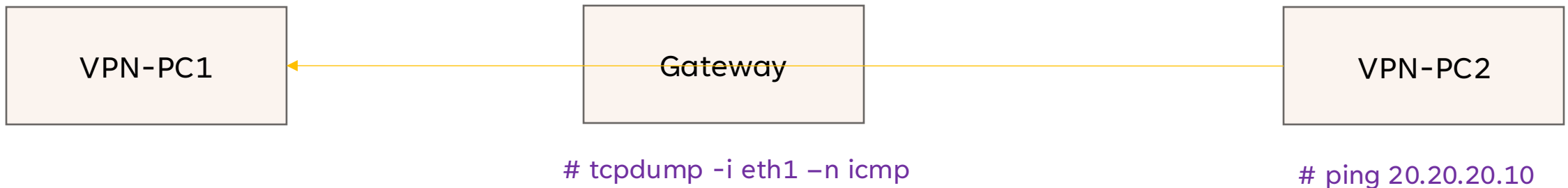
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source               destination
     0     0 MASQUERADE all  --  *      eth1    0.0.0.0/0            0.0.0.0/0
```

5. CONFIGURACIÓN GATEWAY ENMASCARAMIENTO

Comprobar el acceso a la red externa (PC2) desde la red interna (PC1)

En el tcpdump, los paquetes que provengan de la IP interna pero que tengan la IP del Gateway (20.20.20.50) como dirección IP de origen indicarán que el enmascaramiento está funcionando.



```
root@server:/etc/openvpn/server# tcpdump -i eth1 -n icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol details
listening on eth1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
07:20:09.302442 IP 10.10.10.10 > 20.20.20.10: ICMP echo request, id 11, seq 1, length 64
07:20:09.302965 IP 20.20.20.50 > 20.20.20.10: ICMP echo request, id 11, seq 1, length 64
07:20:09.304903 IP 20.20.20.10 > 20.20.20.50: ICMP echo reply, id 11, seq 1, length 64
07:20:09.305665 IP 20.20.20.10 > 10.10.10.10: ICMP echo reply, id 11, seq 1, length 64
07:20:10.305049 IP 10.10.10.10 > 20.20.20.10: ICMP echo request, id 11, seq 2, length 64
07:20:10.305299 IP 20.20.20.50 > 20.20.20.10: ICMP echo request, id 11, seq 2, length 64
```

```
root@server:~# ping 20.20.20.10
PING 20.20.20.10 (20.20.20.10) 56(84) bytes of data:
64 bytes from 20.20.20.10: icmp_seq=1 ttl=63 time=3.76 ms
64 bytes from 20.20.20.10: icmp_seq=2 ttl=63 time=3.29 ms
^C
--- 20.20.20.10 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 3.291/3.524/3.758/0.233 ms
root@server:~#
```

ÍNDICE

1. Creación de maqueta propuesta
2. Creación de Autoridad de Certificación
3. Configuración servidor OpenVPN modo TUN
4. Configuración Gateway port-forwarding
5. Configuración Gateway enmascaramiento
- 6. Configuración OpenVPN cliente PC1**
7. Verificación funcionamiento
8. Comprobación alcance PC2 a PC1
9. Modificación PC2 para que pueda comunicar con PC1 a través de la VPN
10. (Extra) Configuración modo TAP del servidor
11. (Extra) Estudiar características de la solución VPN WireGuard

6. CONFIGURACIÓN OPENVPN CLIENTE PC1

Configurar el cliente OpenVPN en PC1 (client.conf) para conectar al servidor (a través de la IP del Gateway). Para completar esta tarea será necesario copiar los archivos necesarios obtenidos en el apartado 2.

vim /etc/openvpn/client/clipc1.conf

```
client
dev tun
proto udp
remote 20.20.20.50 1194 # Gateway y puerto del servidor OpenVPN
resolv-retry infinite

persist-key
persist-tun
remote-cert-tls server
data-ciphers AES-256-CBC

ca /etc/openvpn/client/ca.crt
cert /etc/openvpn/client/pc1.crt
key /etc/openvpn/client/pc1.key

tls-auth ta.key 1
key-direction 1
```

```
root@server:/etc/openvpn# ls
client  pc1.crt  pc1.key  server  update-resolv-conf
root@server:/etc/openvpn# mv pc1.crt client/
root@server:/etc/openvpn# mv pc1.key client/
root@server:/etc/openvpn# vim client/clipc1.conf
root@server:/etc/openvpn# |
```

Nota: copiar ca.crt en los clientes

```
root@server:/etc/openvpn/client# ls
ca.crt  clipc1.conf  password.txt  pc1.crt  pc1.key
root@server:/etc/openvpn/client# |
```

systemctl start openvpn-client@clipc1

systemctl enable openvpn-client@clipc1

systemctl status openvpn-client@clipc1

APUNTES PALOMA

¿Qué me ha fallado?

- El data-ciphers lo he tenido que cambiar a cipher y otro cifrado
- Las rutas absolutas no me iban, no se pq
- - No hay que poner cifrado en el servidor

```
client
dev tun
proto udp
remote 20.20.20.20 1194 # Gateway y puerto del servidor
OpenVPN
resolv-retry infinite
persist-key
persist-tun
ca ca.crt
cert pc_externo.crt
key pc_externo.key
# tls-auth /home/ubuntu/ta.key 1
tls-auth ta.key 1
key-direction 1
remote-cert-tls server
# cipher AES-256-CBC
cipher AES-256-GCM
```

```
(SERVIDOR VPN)
port 1194
proto udp
dev tun # modo TUN
ca /etc/openvpn/ca.crt
cert /etc/openvpn/vpnserver.crt
key /etc/openvpn/vpnserver.key
dh /etc/openvpn/dh.pem

# subred VPN
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt

# Rutas y reenvío de tráfico a la red
local (192.168.1.0/24)
push "route 192.168.1.0 255.255.255.0"
keepalive 10 120
# cipher AES-256-CBC
tls-auth /etc/openvpn/ta.key 0
key-direction 0
persist-key
persist-tun
```

6. CONFIGURACIÓN OPENVPN CLIENTE PC1

Configurar el cliente OpenVPN en PC1 (client.conf) para conectar al servidor (a través de la IP del Gateway). Para completar esta tarea será necesario copiar los archivos necesarios obtenidos en el apartado 2.

```
sep 18 19:19:53 server openvpn[17970]: Initialization Sequence Complet>
lines 1-24/24 (END)...skipping...
● openvpn-client@clipc1.service - OpenVPN tunnel for clipc1
   Loaded: loaded (/lib/systemd/system/openvpn-client@.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-09-18 19:19:53 UTC; 4s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 17970 (openvpn)
    Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 968)
    Memory: 1.9M
       CPU: 98ms
    CGroup: /system.slice/system-openvpn\x2dclient.slice/openvpn-client@clipc1.service
            └─17970 /usr/sbin/openvpn --suppress-timestamps --nobind --config clipc1.conf

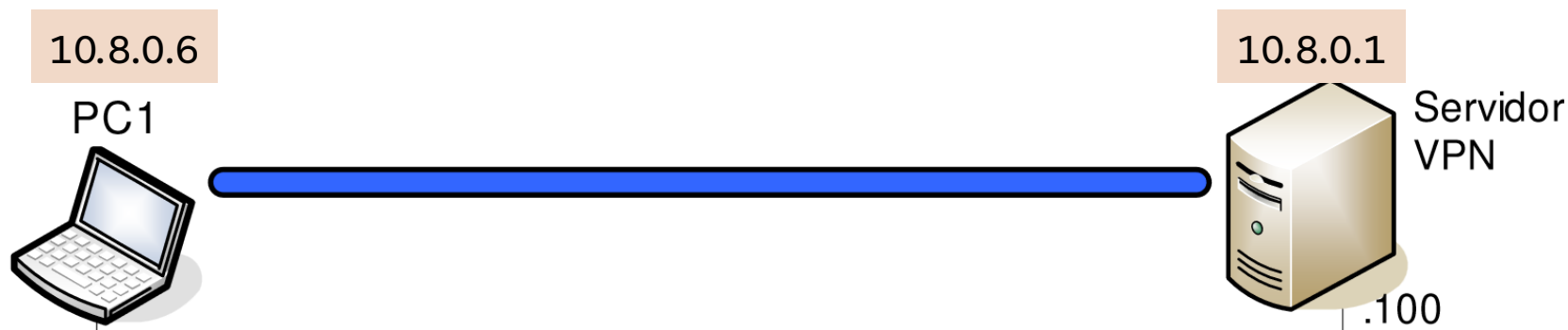
sep 18 19:19:53 server openvpn[17970]: WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
sep 18 19:19:53 server openvpn[17970]: TCP/UDP: Preserving recently used remote address: [AF_INET]20.20.20.50:1194
sep 18 19:19:53 server openvpn[17970]: UDP link local: (not bound)
sep 18 19:19:53 server openvpn[17970]: UDP link remote: [AF_INET]20.20.20.50:1194
sep 18 19:19:53 server openvpn[17970]: [vpnservidor] Peer Connection Initiated with [AF_INET]20.20.20.50:1194
sep 18 19:19:53 server openvpn[17970]: TUN/TAP device tun0 opened
sep 18 19:19:53 server openvpn[17970]: net_iface_mtu_set: mtu 1500 for tun0
sep 18 19:19:53 server openvpn[17970]: net_iface_up: set tun0 up
sep 18 19:19:53 server openvpn[17970]: net_addr_ptp_v4_add: 10.8.0.6 peer 10.8.0.5 dev tun0
sep 18 19:19:53 server openvpn[17970]: Initialization Sequence Completed
~
```

ÍNDICE

1. Creación de maqueta propuesta
2. Creación de Autoridad de Certificación
3. Configuración servidor OpenVPN modo TUN
4. Configuración Gateway port-forwarding
5. Configuración Gateway enmascaramiento
6. Configuración OpenVPN cliente PC1
7. **Verificación funcionamiento**
8. Comprobación alcance PC2 a PC1
9. Modificación PC2 para que pueda comunicar con PC1 a través de la VPN
10. (Extra) Configuración modo TAP del servidor
11. (Extra) Estudiar características de la solución VPN WireGuard

7. VERIFICACIÓN

Iniciar el cliente y comprobar que se conecta al servidor. Verificar la configuración que se establece en el interface virtual tun del cliente, su tabla de encaminamiento y que es posible alcanzar desde PC1 los recursos ofrecidos por el Servidor a través de la VPN (puede hacer una captura del tráfico intercambiado para verificar que los paquetes se transportan sobre tun0, que a su vez se envía encriptado sobre eth0). Desactivar el router por defecto en PC1 y comprobar que sigue siendo posible acceder al Servidor.



```
valid_lft forever preferred_lft forever
7: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.8.0.6 peer 10.8.0.5/32 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::e978:1da0:a244:47c6/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
root@server:/etc/openvpn/client# curl localhost
Este es el PC1 (20.20.20.10)
root@server:/etc/openvpn/client#
```

```
inet6 fe80::20c:29ff:fe37:8a9c/64 scope link
    valid_lft forever preferred_lft forever
6: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::f2be:caf0:4736:711f/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
root@server:/# curl localhost
Este es el Servidor VPN (10.10.10.100)
root@server:/#
```

7. VERIFICACIÓN

Iniciar el cliente y comprobar que se conecta al servidor. Verificar la configuración que se establece en el interface virtual tun del cliente, su tabla de encaminamiento y que es posible alcanzar desde PC1 los recursos ofrecidos por el Servidor a través de la VPN (puede hacer una captura del tráfico intercambiado para verificar que los paquetes se transportan sobre tun0, que a su vez se envía encriptado sobre eth0). Desactivar el router por defecto en PC1 y comprobar que sigue siendo posible acceder al Servidor.

Tabla de Encaminamiento:

```
root@server:/etc/openvpn/client# ip route
default via 20.20.20.50 dev eth0 proto static
default via 192.168.119.2 dev eth0 proto dhcp src 192.168.119.161 metric 100
10.8.0.1 via 10.8.0.5 dev tun0
10.8.0.5 dev tun0 proto kernel scope link src 10.8.0.6
10.10.10.0/24 via 10.8.0.5 dev tun0
20.20.20.0/24 dev eth0 proto kernel scope link src 20.20.20.10
192.168.119.0/24 dev eth0 proto kernel scope link src 192.168.119.161 metric 100
192.168.119.2 dev eth0 proto dhcp scope link src 192.168.119.161 metric 100
```

Conectividad PC1 ↔ Servidor:

```
root@server:/etc/openvpn/client# ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=7.72 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=6.08 ms
64 bytes from 10.8.0.1: icmp_seq=3 ttl=64 time=3.33 ms
^C
--- 10.8.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 3.325/5.709/7.723/1.814 ms
root@server:/etc/openvpn/client# curl 10.8.0.1
Este es el Servidor VPN (10.10.10.100)
```

PC1

```
root@server:/# curl 10.8.0.6
Este es el PC1 (20.20.20.10)

root@server:/# curl localhost
Este es el Servidor VPN (10.10.10.100)
root@server:/#
```

VPN-Servidor

7. VERIFICACIÓN

Envío por VPN:

```
root@server:/etc/openvpn/client# curl 10.8.0.1
Este es el Servidor VPN (10.10.10.100)
root@server:/etc/openvpn/client# curl 10.8.0.1
Este es el Servidor VPN (10.10.10.100)
root@server:/etc/openvpn/client#
```

Paquetes encriptados:

```
root@server:/# tcpdump -i tun0 src 10.8.0.6
tcpdump: verbose output suppressed, use -v[v]... for full protocol decoding
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
05:44:08.043864 IP 10.8.0.6.34802 > server.http: Flags [S], seq 4017144617, win 64240, options [mss 1350,sackOK,TS val 2126941786 ecr 0,nop,wscale 7], length 0
05:44:08.049716 IP 10.8.0.6.34802 > server.http: Flags [.], ack 1321081525, win 502, options [nop,nop,TS val 2126941791 ecr 2117941352], length 0
05:44:08.050347 IP 10.8.0.6.34802 > server.http: Flags [P.], seq 0:72, ack 1, win 502, options [nop,nop,TS val 2126941792 ecr 2117941352], length 72: HTTP: GET / HTTP/1.1
05:44:08.056045 IP 10.8.0.6.34802 > server.http: Flags [.], ack 267, win 501, options [nop,nop,TS val 2126941799 ecr 2117941359], length 0
05:44:08.058034 IP 10.8.0.6.34802 > server.http: Flags [F.], seq 72, ack 267, win 501, options [nop,nop,TS val 2126941801 ecr 2117941359], length 0
05:44:08.062541 IP 10.8.0.6.34802 > server.http: Flags [.], ack 268, win 501, options [nop,nop,TS val 2126941805 ecr 2117941366], length 0
^C
```



7. VERIFICACIÓN

Envío por gateway:

Paquetes no encriptados

<pre>root@server:/etc/openvpn/client# curl 10.10.10.100 Este es el Servidor VPN (10.10.10.100) root@server:/etc/openvpn/client# </pre>	<pre>root@server:/# tcpdump -i eth0 src 20.20.20.10 tcpdump: verbose output suppressed, use -v[v]... for full protocol decoding listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes 05:45:04.402429 IP 20.20.20.10.35708 > 20.20.20.50.openvpn: UDP, length 72 05:45:04.403964 IP 20.20.20.10.35708 > server.openvpn: UDP, length 72 05:45:08.828745 IP 20.20.20.10.35708 > 20.20.20.50.openvpn: UDP, length 120 05:45:08.829313 IP 20.20.20.10.35708 > server.openvpn: UDP, length 120 05:45:08.832722 IP 20.20.20.10.35708 > 20.20.20.50.openvpn: UDP, length 104 05:45:08.833324 IP 20.20.20.10.35708 > server.openvpn: UDP, length 104 05:45:08.834054 IP 20.20.20.10.35708 > 20.20.20.50.openvpn: UDP, length 184 05:45:08.835034 IP 20.20.20.10.35708 > server.openvpn: UDP, length 184 05:45:08.840968 IP 20.20.20.10.35708 > 20.20.20.50.openvpn: UDP, length 104 05:45:08.841664 IP 20.20.20.10.35708 > server.openvpn: UDP, length 104 05:45:08.843319 IP 20.20.20.10.35708 > 20.20.20.50.openvpn: UDP, length 104 05:45:08.843752 IP 20.20.20.10.35708 > server.openvpn: UDP, length 104 05:45:08.848908 IP 20.20.20.10.35708 > 20.20.20.50.openvpn: UDP, length 104 05:45:08.849320 IP 20.20.20.10.35708 > server.openvpn: UDP, length 104 </pre>
---	---

PC1	VPN-Servidor
-----	--------------

7. VERIFICACIÓN

Desactivar el router por defecto en PC1 y comprobar que sigue siendo posible acceder al Servidor:

```
root@server:/etc/openvpn/client# ip route
default via 20.20.20.50 dev eth0 proto static
default via 192.168.119.2 dev eth0 proto dhcp src 192.168.119.161 metric
100
10.8.0.1 via 10.8.0.5 dev tun0
10.8.0.5 dev tun0 proto kernel scope link src 10.8.0.6
10.10.10.0/24 via 10.8.0.5 dev tun0
20.20.20.0/24 dev eth0 proto kernel scope link src 20.20.20.10
192.168.119.0/24 dev eth0 proto kernel scope link src 192.168.119.161 me
tric 100
192.168.119.2 dev eth0 proto dhcp scope link src 192.168.119.161 metric
100
```

```
root@server:/etc/openvpn/client# ip route del default
root@server:/etc/openvpn/client# ip route
default via 192.168.119.2 dev eth0 proto dhcp src 192.168.119.161 metric
100
10.8.0.1 via 10.8.0.5 dev tun0
10.8.0.5 dev tun0 proto kernel scope link src 10.8.0.6
10.10.10.0/24 via 10.8.0.5 dev tun0
20.20.20.0/24 dev eth0 proto kernel scope link src 20.20.20.10
192.168.119.0/24 dev eth0 proto kernel scope link src 192.168.119.161 me
tric 100
192.168.119.2 dev eth0 proto dhcp scope link src 192.168.119.161 metric
100
```

Eliminar router por defecto:

ip route del default

Comprobar acceso al servidor:

```
root@server:/etc/openvpn/client# curl 10.8.0.1
Este es el Servidor VPN (10.10.10.100)
root@server:/etc/openvpn/client# curl 10.10.10.100
Este es el Servidor VPN (10.10.10.100)
root@server:/etc/openvpn/client#
```

PC1

7. VERIFICACIÓN

Iniciar el cliente y comprobar que se conecta al servidor. Verificar la configuración que se establece en el interface virtual tun del cliente, su tabla de encaminamiento y que es posible alcanzar desde PC1 los recursos ofrecidos por el Servidor a través de la VPN (puede hacer una captura del tráfico intercambiado para verificar que los paquetes se transportan sobre tun0, que a su vez se envía encriptado sobre eth0). Desactivar el router por defecto en PC1 y comprobar que sigue siendo posible acceder al Servidor.

```
root@server:/etc/openvpn/server# ip addr show
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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:57:8d:fc brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    altname ens33
    inet 10.10.10.100/24 brd 10.10.10.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet 192.168.119.160/24 metric 100 brd 192.168.119.255 scope global dynamic eth0
        valid_lft 1756sec preferred_lft 1756sec
    inet6 fe80::20c:29ff:fe57:8dfc/64 scope link
        valid_lft forever preferred_lft forever
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::9fc1:4b3:e876:6cc2/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
```

VPN-Servidor

ÍNDICE

1. Creación de maqueta propuesta
2. Creación de Autoridad de Certificación
3. Configuración servidor OpenVPN modo TUN
4. Configuración Gateway port-forwarding
5. Configuración Gateway enmascaramiento
6. Configuración OpenVPN cliente PC1
7. Verificación funcionamiento
- 8. Comprobación alcance PC2 a PC1**
9. Modificación PC2 para que pueda comunicar con PC1 a través de la VPN
10. (Extra) Configuración modo TAP del servidor
11. (Extra) Estudiar características de la solución VPN WireGuard

8. COMPROBACIÓN ALCANCE PC2 A PC1

Compruebe si es posible alcanzar el equipo PC2 desde PC1 a través de la conexión VPN. ¿Qué está sucediendo?

```
PC1
root@server:/etc/openvpn/client# ping 10.10.10.10
PING 10.10.10.10 (10.10.10.10) 56(84) bytes of data.

--- 10.10.10.10 ping statistics ---
74 packets transmitted, 0 received, 100% packet loss, time 74694ms

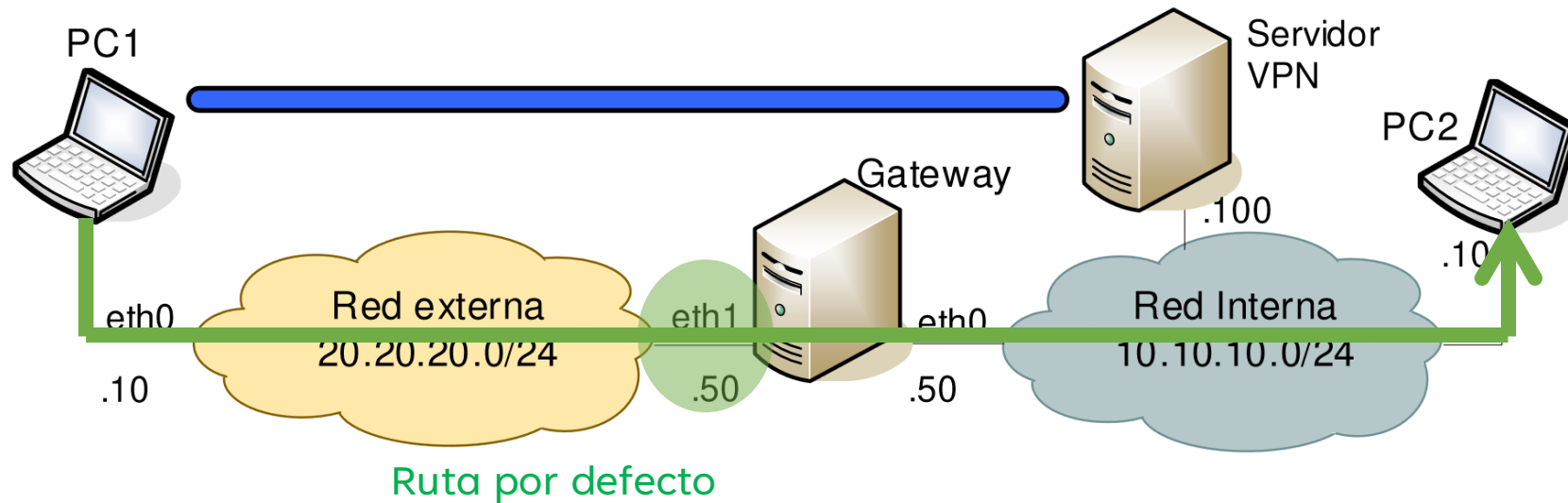
^Croot@server:/etc/openvpn/client# ip route
default via 192.168.119.2 dev eth0 proto dhcp src 192.168.119.161 metric
100
10.8.0.1 via 10.8.0.5 dev tun0
10.8.0.5 dev tun0 proto kernel scope link src 10.8.0.6
10.10.10.0/24 via 10.8.0.5 dev tun0
20.20.20.0/24 dev eth0 proto kernel scope link src 20.20.20.10
192.168.119.0/24 dev eth0 proto kernel scope link src 192.168.119.161 me
tric 100
192.168.119.2 dev eth0 proto dhcp scope link src 192.168.119.161 metric
100
root@server:/etc/openvpn/client# |

Servidor
root@server:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN grou
p 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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000
    link/ether 00:0c:29:3c:2e:a1 brd ff:ff:ff:ff:ff:ff
    altnam enp2s1
    altnam ens33
    inet 10.10.10.10/24 brd 10.10.10.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet 192.168.119.158/24 metric 100 brd 192.168.119.255 scope global
dynamic eth0
        valid_lft 1231sec preferred_lft 1231sec
    inet6 fe80::20c:29ff:fe3c:2ea1/64 scope link
        valid_lft forever preferred_lft forever
```

No es posible

8. COMPROBACIÓN ALCANCE PC2 A PC1

Compruebe si es posible alcanzar el equipo PC2 desde PC1 a través de la conexión VPN. ¿Qué está sucediendo?

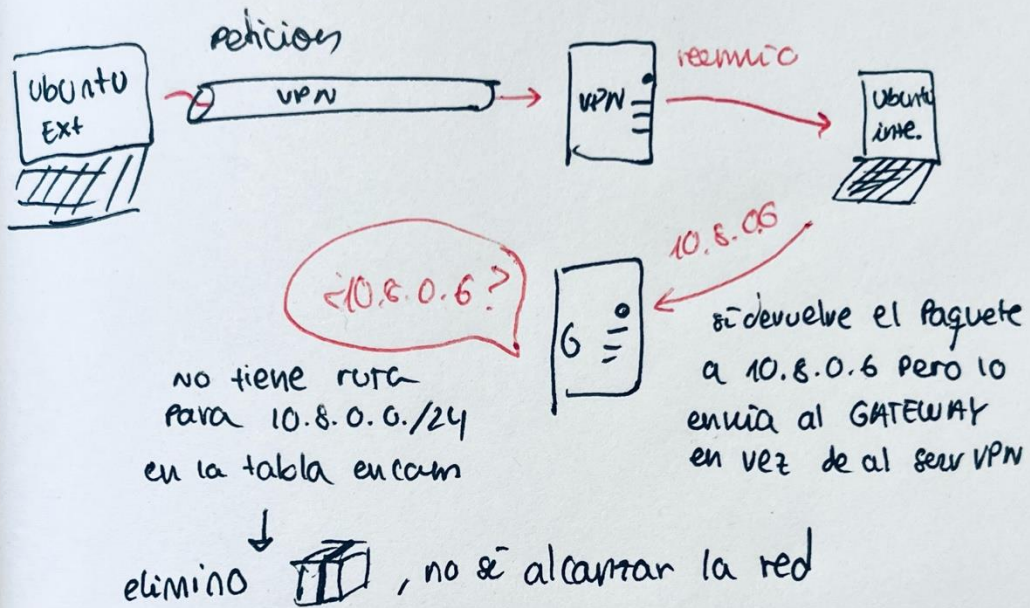


- Teo: “Lo que ocurre es que PC1 envía la petición por la VPN (tiene una ruta para alcanzar 10.10.10.0/24 por tun0). El servidor recibe el mensaje y lo reenvía a PC2. PC2 sí devuelve el paquete a 10.8.0.6, pero no lo envía al servidor sino al Gateway, como indica su tabla de encaminamiento (no tiene ruta para 10.8.0.0/24 y usa su ruta por defecto). Y el Gateway elimina el paquete, al no saber cómo alcanzar la red 10.8.0.0/24.”

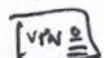
8. COMPROBACIÓN ALCANCE PC2 A PC1

Eliminar en Ubuntu Ext el Router por defecto:
ip route del default

¿Qué ocurre?



¿Qué cambiamos en (PC2) para evitar esto?

Añadiendo una ruta para q todo el tráfico destinado a la red VPN se enrute a través 

```
# ip route add 10.8.0.0/24 via 192.168.1.100
```

¿Y en el Gateway q cambiamos?

```
# iptables -A FORWARD -i tun0 -o eth0 -m state --state RELATED, ESTABLISHED -j ACCEPT
```

```
# iptables -A FORWARD -i eth0 -o tun0 -j ACCEPT
```

```
# ip route add 10.8.0.0/24 via 192.168.1.100
```

ÍNDICE

1. Creación de maqueta propuesta
2. Creación de Autoridad de Certificación
3. Configuración servidor OpenVPN modo TUN
4. Configuración Gateway port-forwarding
5. Configuración Gateway enmascaramiento
6. Configuración OpenVPN cliente PC1
7. Verificación funcionamiento
8. Comprobación alcance PC2 a PC1
9. **Modificación PC2 para que pueda comunicar con PC1 a través de la VPN**
10. (Extra) Configuración modo TAP del servidor
11. (Extra) Estudiar características de la solución VPN WireGuard

9. MODIFICACIÓN PC2 PARA QUE PUEDA COMUNICAR CON PC1 A TRAVÉS DE LA VPN

Modificar la configuración de PC2 para que pueda comunicar con PC1 a través de la VPN. Nota: también es posible configurar el Gateway para el retorno de los paquetes a través de la VPN, sin modificar el encaminamiento en PC2. Compruebe ambas opciones.

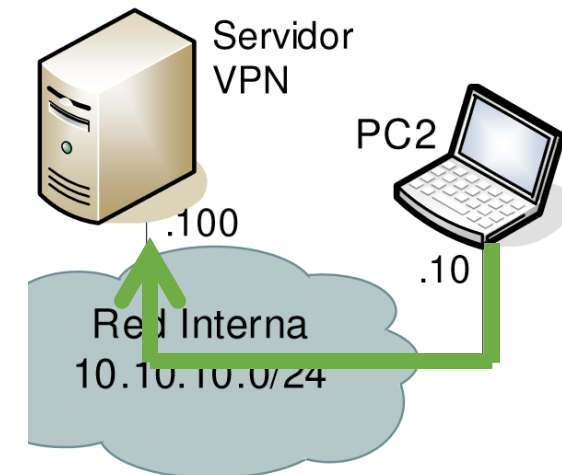
Cambiar configuración en PC2

- PC2 necesita saber cómo enviar el tráfico de vuelta a PC1 (10.8.0.6), que está en la red VPN.
- Añadir una ruta en PC2 para que todo el tráfico destinado a la red VPN se enrute a través del VPN-Servidor.

ip route add 10.8.0.0/24 via 10.10.10.100 [Para ir a la red de la VPN (10.8.0.0/24) pasa por la VPN (10.10.10.100)]

```
root@server:~# ip route add 10.8.0.0/24 via 10.10.10.100
root@server:~# curl 10.8.0.6
Este es el PC1 (20.20.20.10)

root@server:~# ip route
default via 10.10.10.50 dev eth0 proto static
default via 192.168.119.2 dev eth0 proto dhcp src 192.168.119.158 metric 100
10.8.0.0/24 via 10.10.10.100 dev eth0
10.10.10.0/24 dev eth0 proto kernel scope link src 10.10.10.10
192.168.119.0/24 dev eth0 proto kernel scope link src 192.168.119.158 metric 100
192.168.119.2 dev eth0 proto dhcp scope link src 192.168.119.158 metric 100
```



Borrar la regla:

ip route del 10.8.0.0/24 via 10.10.10.100

9. MODIFICACIÓN PC2 PARA QUE PUEDA COMUNICAR CON PC1 A TRAVÉS DE LA VPN

Cambiar configuración en Gateway

```
# iptables -A FORWARD -i tun0 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

- Permite que el tráfico relacionado y establecido que proviene de la VPN (tun0) y va hacia la red interna (eth0) pase correctamente.

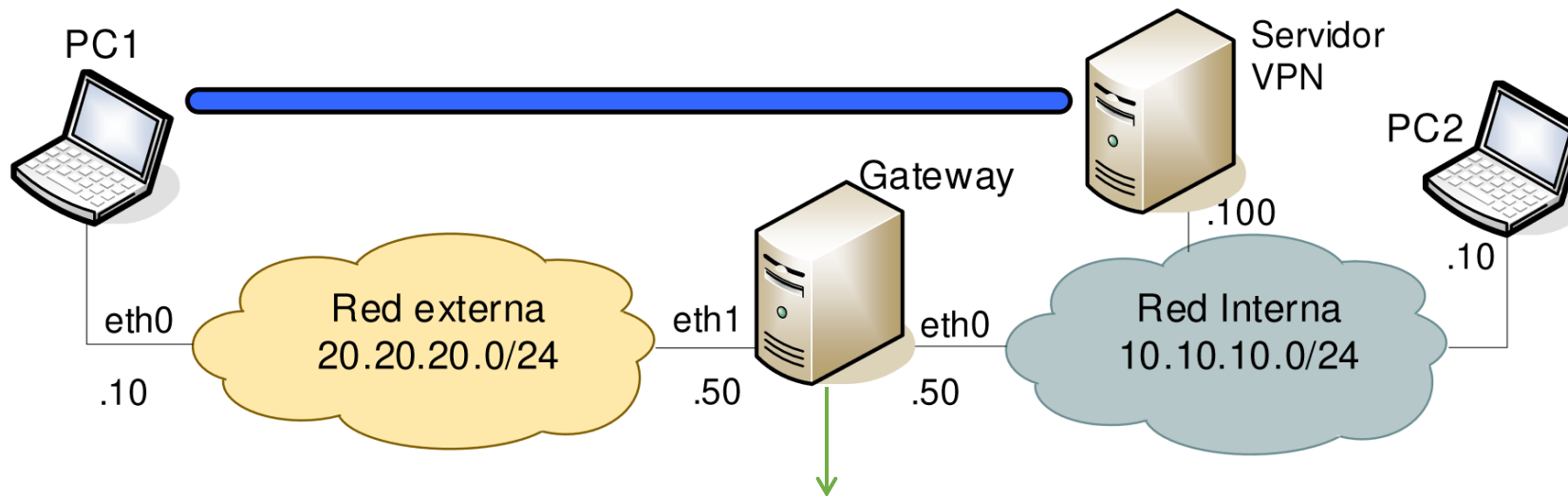
```
# iptables -A FORWARD -i eth0 -o tun0 -j ACCEPT
```

- Permite que el tráfico nuevo desde la red interna (eth0) hacia la VPN (tun0) fluya sin problemas.

9. MODIFICACIÓN PC2 PARA QUE PUEDA COMUNICAR CON PC1 A TRAVÉS DE LA VPN

Cambiar configuración en Gateway

```
# ip route add 10.8.0.0/24 via 10.10.10.100
```



Si le llega tráfico hacia 10.8.0.6 → lo redirige al servidor VPN para que envíe el tráfico por el túnel

9. MODIFICACIÓN PC2 PARA QUE PUEDA COMUNICAR CON PC1 A TRAVÉS DE LA VPN

Cambiar configuración en Gateway

Gateway:

PC1:

Gateway:

```
root@server:~# ip route add 10.8.0.0/24 via 10.10.10.50
root@server:~# ip route
default via 192.168.119.2 dev eth0 proto dhcp src 192.168.119.162 metric
100
default via 192.168.119.2 dev eth1 proto dhcp src 192.168.119.163 metric
100
10.8.0.0/24 via 10.10.10.50 dev eth0
10.10.10.0/24 dev eth0 proto kernel scope link src 10.10.10.50
20.20.20.0/24 dev eth1 proto kernel scope link src 20.20.20.50
192.168.119.0/24 dev eth0 proto kernel scope link src 192.168.119.162 me
tric 100
192.168.119.0/24 dev eth1 proto kernel scope link src 192.168.119.163 me
tric 100
192.168.119.2 dev eth0 proto dhcp scope link src 192.168.119.162 metric
100
192.168.119.2 dev eth1 proto dhcp scope link src 192.168.119.163 metric
100
```

PC1:

```
root@server:/etc/openvpn/client# curl 10.10.10.10
Este es el PC2 (10.10.10.10)
root@server:/etc/openvpn/client# ping 10.10.10.10
PING 10.10.10.10 (10.10.10.10) 56(84) bytes of data.
64 bytes from 10.10.10.10: icmp_seq=1 ttl=64 time=4.21 ms
64 bytes from 10.10.10.10: icmp_seq=2 ttl=64 time=4.71 ms
64 bytes from 10.10.10.10: icmp_seq=3 ttl=64 time=4.82 ms
^C
--- 10.10.10.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 4.211/4.579/4.820/0.264 ms
root@server:/etc/openvpn/client#
```

9. MODIFICACIÓN PC2 PARA QUE PUEDA COMUNICAR CON PC1 A TRAVÉS DE LA VPN

Cambiar configuración en Gateway

```
Gateway PC1
root@server:~# ip route
default via 192.168.119.2 dev eth1 proto dhcp src 192.168.119.163 metric
100
default via 192.168.119.2 dev eth0 proto dhcp src 192.168.119.162 metric
100
10.10.10.0/24 dev eth0 proto kernel scope link src 10.10.10.50
20.20.20.0/24 dev eth1 proto kernel scope link src 20.20.20.50
192.168.119.0/24 dev eth1 proto kernel scope link src 192.168.119.163 me
tric 100
192.168.119.0/24 dev eth0 proto kernel scope link src 192.168.119.162 me
tric 100
192.168.119.2 dev eth1 proto dhcp scope link src 192.168.119.163 metric
100
192.168.119.2 dev eth0 proto dhcp scope link src 192.168.119.162 metric
100
root@server:~# ip route add 10.8.0.0/24 via 10.10.10.100
root@server:~# ip route
default via 192.168.119.2 dev eth1 proto dhcp src 192.168.119.163 metric
100
default via 192.168.119.2 dev eth0 proto dhcp src 192.168.119.162 metric
100
10.8.0.0/24 via 10.10.10.100 dev eth0
10.10.10.0/24 dev eth0 proto kernel scope link src 10.10.10.50
20.20.20.0/24 dev eth1 proto kernel scope link src 20.20.20.50

PC2 Servidor VPN
root@server:~# ping 10.8.0.6
PING 10.8.0.6 (10.8.0.6) 56(84) bytes of data.
^C
--- 10.8.0.6 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2032ms

root@server:~# ping 10.8.0.6
PING 10.8.0.6 (10.8.0.6) 56(84) bytes of data.
64 bytes from 10.8.0.6: icmp_seq=1 ttl=63 time=5.95 ms
From 10.10.10.50 icmp_seq=2 Redirect Host(New nexthop: 10.10.10.100)
64 bytes from 10.8.0.6: icmp_seq=2 ttl=63 time=5.62 ms
64 bytes from 10.8.0.6: icmp_seq=3 ttl=63 time=4.93 ms
64 bytes from 10.8.0.6: icmp_seq=4 ttl=63 time=5.52 ms
^C
--- 10.8.0.6 ping statistics ---
4 packets transmitted, 4 received, +1 errors, 0% packet loss, time 3006m
s
rtt min/avg/max/mdev = 4.930/5.505/5.945/0.366 ms
root@server:~# curl 10.8.0.6
Este es el PC1 (20.20.20.10)

root@server:~#
```

9. MODIFICACIÓN PC2 PARA QUE PUEDA COMUNICAR CON PC1 A TRAVÉS DE LA VPN

Cambiar configuración en Gateway

```
root@server:~# tcpdump -i eth0 src 10.10.10.10
tcpdump: verbose output suppressed, use -v[v]... for full protocol decoding
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
07:32:31.895680 IP 10.10.10.10.43730 > server.http: Flags [S], seq 2377268605, win 64240, options [mss 1460,sackOK,TS val 3633177382 ecr 0,nop,wscale 7], length 0
07:32:31.898138 IP 10.10.10.10.43730 > server.http: Flags [.] , ack 1782298973, win 502, options [nop,nop,TS val 3633177385 ecr 301218512], length 0
07:32:31.898721 IP 10.10.10.10.43730 > server.http: Flags [P.] , seq 0:75, ack 1, win 502, options [nop,nop,TS val 3633177385 ecr 301218512], length 75: HTTP: GET / HTTP/1.1
07:32:31.901782 IP 10.10.10.10.43730 > server.http: Flags [.] , ack 261, win 501, options [nop,nop,TS val 3633177388 ecr 301218517], length 0
07:32:31.903716 IP 10.10.10.10.43730 > server.http: Flags [F.] , seq 75, ack 261, win 501, options [nop,nop,TS val 3633177390 ecr 301218517], length 0
07:32:31.906367 IP 10.10.10.10.43730 > server.http: Flags [.] , ack 262, win 501, options [nop,nop,TS val 3633177393 ecr 301218521], length 0
```

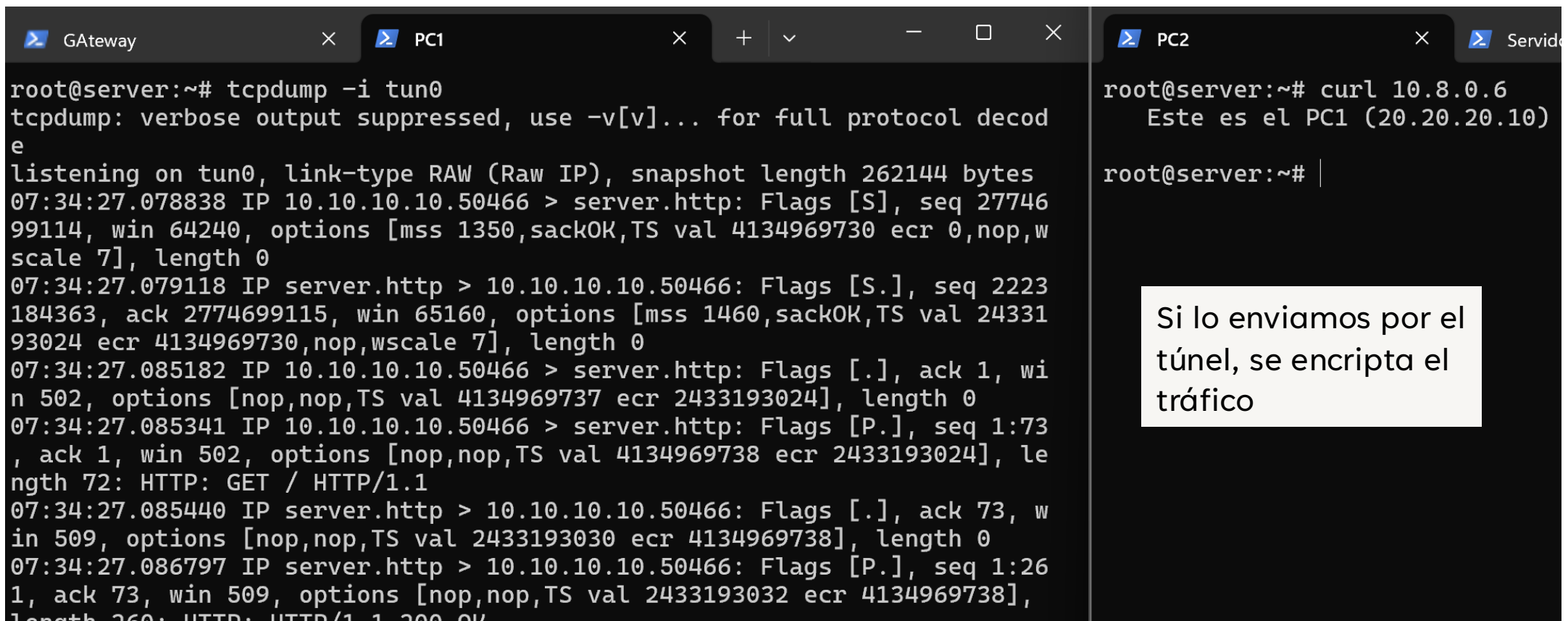
```
root@server:~# curl 20.20.20.10
Este es el PC1 (20.20.20.10)

root@server:~#
```

Si lo mandamos por el Gateway **no** se encripta el tráfico

9. MODIFICACIÓN PC2 PARA QUE PUEDA COMUNICAR CON PC1 A TRAVÉS DE LA VPN

Cambiar configuración en Gateway



The image shows two terminal windows side-by-side. The left window, titled 'Gateway', displays the output of a tcpdump command on the tun0 interface, showing an HTTP GET request from 10.10.10.10.50466 to server.http. The right window, titled 'PC2', shows a curl command being executed to connect to 10.8.0.6, with a comment 'Este es el PC1 (20.20.20.10)'. Below the curl command, there is a text box with the instruction: 'Si lo enviamos por el túnel, se encripta el tráfico'.

```
root@server:~# tcpdump -i tun0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decoding
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
07:34:27.078838 IP 10.10.10.10.50466 > server.http: Flags [S], seq 2774699114, win 64240, options [mss 1350,sackOK,TS val 4134969730 ecr 0,nop,wscale 7], length 0
07:34:27.079118 IP server.http > 10.10.10.10.50466: Flags [S.], seq 2223184363, ack 2774699115, win 65160, options [mss 1460,sackOK,TS val 2433193024 ecr 4134969730,nop,wscale 7], length 0
07:34:27.085182 IP 10.10.10.10.50466 > server.http: Flags [.], ack 1, win 502, options [nop,nop,TS val 4134969737 ecr 2433193024], length 0
07:34:27.085341 IP 10.10.10.10.50466 > server.http: Flags [P.], seq 1:73, ack 1, win 502, options [nop,nop,TS val 4134969738 ecr 2433193024], length 72: HTTP: GET / HTTP/1.1
07:34:27.085440 IP server.http > 10.10.10.10.50466: Flags [.], ack 73, win 509, options [nop,nop,TS val 2433193030 ecr 4134969738], length 0
07:34:27.086797 IP server.http > 10.10.10.10.50466: Flags [P.], seq 1:261, ack 73, win 509, options [nop,nop,TS val 2433193032 ecr 4134969738], length 260: HTTP: HTTP/1.1 200 OK
```

```
root@server:~# curl 10.8.0.6
Este es el PC1 (20.20.20.10)

root@server:~#
```

Si lo enviamos por el túnel, se encripta el tráfico

```

root@server:~# tcpdump -i tun0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decoding
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
07:34:27.078838 IP 10.10.10.10.50466 > server.http: Flags [S], seq 2774699114, win 64240, options [mss 1350,sackOK,TS val 4134969730 ecr 0,nop,wscale 7], length 0
07:34:27.079118 IP server.http > 10.10.10.10.50466: Flags [S.], seq 2223184363, ack 2774699115, win 65160, options [mss 1460,sackOK,TS val 2433193024 ecr 4134969730,nop,wscale 7], length 0
07:34:27.085182 IP 10.10.10.10.50466 > server.http: Flags [.], ack 1, win 502, options [nop,nop,TS val 4134969737 ecr 2433193024], length 0
07:34:27.085341 IP 10.10.10.10.50466 > server.http: Flags [P.], seq 1:73, ack 1, win 502, options [nop,nop,TS val 4134969738 ecr 2433193024], length 72: HTTP: GET / HTTP/1.1
07:34:27.085440 IP server.http > 10.10.10.10.50466: Flags [.], ack 73, win 509, options [nop,nop,TS val 2433193030 ecr 4134969738], length 0
07:34:27.086797 IP server.http > 10.10.10.10.50466: Flags [P.], seq 1:261, ack 73, win 509, options [nop,nop,TS val 2433193032 ecr 4134969738], length 260: HTTP: HTTP/1.1 200 OK
07:34:27.091878 IP 10.10.10.10.50466 > server.http: Flags [.], ack 261, win 501, options [nop,nop,TS val 4134969743 ecr 2433193032], length 0
07:34:27.094153 IP 10.10.10.10.50466 > server.http: Flags [F.], seq 73, ack 261, win 501, options [nop,nop,TS val 4134969747 ecr 2433193032], length 0
07:34:27.095112 IP server.http > 10.10.10.10.50466: Flags [F.], seq 261, ack 74, win 509, options [nop,nop,TS val 2433193040 ecr 4134969747], length 0
07:34:27.101387 IP 10.10.10.10.50466 > server.http: Flags [.], ack 262, win 501, options [nop,nop,TS val 4134969753 ecr 2433193040], length 0

```

Enviado por el túnel

```

root@server:~# tcpdump -i eth0 src 10.10.10.10
tcpdump: verbose output suppressed, use -v[v]... for full protocol decoding
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
07:37:43.744989 IP 10.10.10.10.39000 > server.http: Flags [S], seq 449328341, win 64240, options [mss 1460,sackOK,TS val 3633489229 ecr 0,nop,wscale 7], length 0
07:37:43.747409 IP 10.10.10.10.39000 > server.http: Flags [.], ack 309752341, win 502, options [nop,nop,TS val 3633489231 ecr 301530361], length 0
07:37:43.747781 IP 10.10.10.10.39000 > server.http: Flags [P.], seq 0:75, ack 1, win 502, options [nop,nop,TS val 3633489232 ecr 301530361], length 75: HTTP: GET / HTTP/1.1
07:37:43.752340 IP 10.10.10.10.39000 > server.http: Flags [.], ack 261, win 501, options [nop,nop,TS val 3633489236 ecr 301530366], length 0
07:37:43.754771 IP 10.10.10.10.39000 > server.http: Flags [F.], seq 75, ack 261, win 501, options [nop,nop,TS val 3633489239 ecr 301530366], length 0
07:37:43.757894 IP 10.10.10.10.39000 > server.http: Flags [.], ack 262, win 501, options [nop,nop,TS val 3633489242 ecr 301530372], length 0

```

Enviado por el Gateway



FIN PARTE OBLIGATORIA

Paloma Pérez de Madrid e Inés del Río

PARTE OPCIONAL

Estudiar las características de la solución VPN WireGuard (www.wireguard.com). Preparar un escenario para poner en marcha una configuración básica de esta VPN (puede usarse como base la maqueta anterior).

WireGuard es un protocolo VPN de última generación diseñado para ser más rápido, seguro y fácil de usar que los protocolos tradicionales como OpenVPN o IPsec.

Características principales de WireGuard:

1. **Criptografía moderna:** Utiliza algoritmos avanzados como ChaCha20 para el cifrado y Poly1305 para la autenticación de mensajes, lo que garantiza una alta seguridad.
2. **Rendimiento superior:** Está optimizado para ofrecer altas velocidades y baja latencia, siendo capaz de manejar múltiples núcleos de CPU simultáneamente.
3. **Facilidad de configuración:** La configuración de WireGuard es sencilla y rápida, lo que lo hace accesible tanto para usuarios novatos como para expertos.

PARTE OPCIONAL

WireGuard

	OpenVPN	WireGuard
Seguridad	Utiliza AES-256 para el cifrado y soporta múltiples algoritmos de autenticación.	Algoritmos ChaCha20 para el cifrado y Poly1305 para la autenticación.
Velocidad	Puede ser más lento, especialmente cuando se usa con TCP, aunque es más estable en redes con alta latencia.	Alta velocidad y baja latencia debido a su diseño eficiente y uso de UDP.
Privacidad	No guarda registros de IPs y es más adecuado para entornos donde la privacidad es una prioridad	Mantiene una lista de IPs autorizadas durante la sesión, lo que puede ser un problema en entornos donde el anonimato es crucial.

PARTE OPCIONAL

Configuración básica de WireGuard

- **Servidor WireGuard:** Configurar la interfaz wg0 para crear la VPN y enrutar tráfico hacia los clientes.
 - **Cliente PC1 WireGuard:** Configurar la interfaz wg0 y conectarse al servidor.
-
1. Instalar WireGuard en el VPN-Servidor y PC1
 2. Configuración VPN-Servidor:
 1. Generar claves de WireGuard
 2. Configurar archivo de WireGuard (/etc/wireguard/wg0.conf)
 3. Habilitar el reenvío de IP en el servidor (hecho en apartados anteriores)
 4. Iniciar la interfaz de WireGuard
 3. Configuración PC1:
 1. Generar claves de WireGuard
 2. Configurar archivo de WireGuard (/etc/wireguard/wg0.conf)
 3. Iniciar interfaz
 4. Enrutar el tráfico en el Gateway (ip route add 10.8.0.0/24 via 10.10.10.100, hecho en apartados anteriores)

PARTE OPCIONAL

Nota: antes de empezar, desactivar openvpn en el Servidor y en el Cliente (para hacer un ejemplo con las mismas IPs que en OpenVPN)

```
PC1
root@server:~# systemctl status openvpn-client@clipc1
○ openvpn-client@clipc1.service - OpenVPN tunnel for c
  Loaded: loaded (/lib/systemd/system/openvpn-clien
  Active: inactive (dead) since Fri 2024-09-27 07:3
  Docs: man:openvpn(8)
        https://community.openvpn.net/openvpn/wik
        https://community.openvpn.net/openvpn/wik
  Process: 818 ExecStart=/usr/sbin/openvpn --suppres
  Main PID: 818 (code=exited, status=0/SUCCESS)
  Status: "Initialization Sequence Completed"
  CPU: 163ms

sep 27 07:35:57 server openvpn[818]: SIGUSR1[soft,ping
sep 27 07:36:02 server openvpn[818]: TCP/UDP: Preservi
sep 27 07:36:02 server openvpn[818]: UDP link local: (>
sep 27 07:36:02 server openvpn[818]: UDP link remote: >
sep 27 07:36:51 server systemd[1]: Stopping OpenVPN tu
sep 27 07:36:51 server openvpn[818]: event_wait : Inte
sep 27 07:36:51 server openvpn[818]: net_addr_ptp_v4_d
sep 27 07:36:51 server openvpn[818]: SIGTERM[hard,] re
sep 27 07:36:51 server systemd[1]: openvpn-client@clip
sep 27 07:36:51 server systemd[1]: Stopped OpenVPN tun
lines 1-21/21 (END)

Servidor
root@server:~# systemctl status openvpn-server@servidorvpn
○ openvpn-server@servidorvpn.service - OpenVPN service for servidorvpn
  Loaded: loaded (/lib/systemd/system/openvpn-server@.service; enabled; vendor preset
  Active: inactive (dead) since Fri 2024-09-27 07:34:02 UTC; 30min ago
  Docs: man:openvpn(8)
        https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
        https://community.openvpn.net/openvpn/wiki/HOWTO
  Process: 748 ExecStart=/usr/sbin/openvpn --status /run/openvpn-server/status-servido
  Main PID: 748 (code=exited, status=0/SUCCESS)
  Status: "Initialization Sequence Completed"
  CPU: 191ms

sep 27 07:21:39 server openvpn[748]: 20.20.20.10:39254 peer info: IV_COMP_STUBv2=1
sep 27 07:21:39 server openvpn[748]: 20.20.20.10:39254 peer info: IV_TCPNL=1
sep 27 07:21:39 server openvpn[748]: 20.20.20.10:39254 [pc1] Peer Connection Initiated w
sep 27 07:21:39 server openvpn[748]: pc1/20.20.20.10:39254 MULTI_sva: pool returned IPv4
sep 27 07:34:02 server openvpn[748]: event_wait : Interrupted system call (code=4)
sep 27 07:34:02 server openvpn[748]: net_addr_ptp_v4_del: 10.8.0.1 dev tun0
sep 27 07:34:02 server systemd[1]: Stopping OpenVPN service for servidorvpn...
sep 27 07:34:02 server openvpn[748]: SIGTERM[hard,] received, process exiting
sep 27 07:34:02 server systemd[1]: openvpn-server@servidorvpn.service: Deactivated succe
sep 27 07:34:02 server systemd[1]: Stopped OpenVPN service for servidorvpn.
lines 1-21/21 (END)

Gateway
PC2
```

PARTE OPCIONAL

Configuración básica de WireGuard

1. Instalar WireGuard en el VPN-Servidor y PC1

```
# apt install wireguard
```

Configuración VPN-Servidor:

1. Generar claves de WireGuard

```
root@server:~# wg genkey | tee privatekey | wg pubkey > publickey
root@server:~# ls
easy-rsa privatekey publickey snap
```

```
root@server:~# cat privatekey
OPX25WruEwG/UZmFLSZvR2s8MNlnvravjxIFRCtvT3o=
root@server:~# cat publickey
zmk8sITH4WkA3a1dk05PgQX8BFZvmJ+xxwrjdnveSFI=
root@server:~#
```

Generar también las claves en el PC1:

```
root@server:~# wg genkey | tee privatekey | wg pubkey > publickey
root@server:~# cat publickey
178uQQiOr1TglBkoHwF7WCp1KhFwgWcy86iY9WGTCyY=
root@server:~# cat privatekey
aFVpwmytz1n00LZcMWlUEF/INpEXnmMIiBocI7fU1Fg=
```

PARTE OPCIONAL


Configuración básica de WireGuard

2. Configurar archivo de WireGuard
(/etc/wireguard/wg0.conf)

4. Iniciar la interfaz de WireGuard
`wg-quick up wg0`

Claves públicas y privadas en PC1:

```
root@server:~# wg genkey | tee privatekey | wg pubkey > publickey
root@server:~# cat publickey
178uQQiOr1TglBkoHwF7WCp1KhFwgWcy86iY9WGTCyY=
root@server:~# cat privatekey
aFVpwwytz1n00LZcMWlUEF/INpEXnmMIiBocI7fU1Fg=
```



VPN-Server

[Interface]

Address = 10.8.0.1/24

ListenPort = 51820

PrivateKey = OPX25WruEwG/UZmFLSZvR2s8MNlnvrajxIFRCtvT3o=

PostUp = iptables -A FORWARD -i wg0 -j ACCEPT

PostUp = iptables -A FORWARD -o wg0 -j ACCEPT

PostUp = iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

PostDown = iptables -D FORWARD -i wg0 -j ACCEPT

PostDown = iptables -D FORWARD -o wg0 -j ACCEPT

PostDown = iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE

[Peer]

PublicKey = 178uQQiOr1TglBkoHwF7WCp1KhFwgWcy86iY9WGTCyY=

AllowedIPs = 10.8.0.6/24

PARTE OPCIONAL

Configuración básica de WireGuard

En PC1:

1. Generar claves de WireGuard
2. Configurar archivo de WireGuard (/etc/wireguard/wg0.conf)
3. Iniciar interfaz

VPN-PC1

[Interface]

Address = 10.8.0.6/24

PrivateKey = aFVpwmytz1n0OlZcMWIUEF/INpEXnmMliBocl7fU1Fg=

[Peer]

PublicKey = zmk8sITH4WkA3a1dkO5PgQX8BFZvmJ+xxwrjdnveSFI=

Endpoint = 10.10.10.100:51820

AllowedIPs = 10.8.0.0/24

PersistentKeepalive = 25

PARTE OPCIONAL

Configuración básica de WireGuard

VPN-Servidor:

```
11: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN group default qlen 1000
    link/none
    inet 10.8.0.1/24 scope global wg0
        valid_lft forever preferred_lft forever
```

```
root@server:~# wg
interface: wg0
  public key: zmk8sITH4WkA3a1dk05PgQX8BFZvmJ+xxwrjdnveSFI=
  private key: (hidden)
  listening port: 51820

peer: 178uQQi0r1TglBkoHwF7WCp1KhFwgWcy86iY9WGTCyY=
  endpoint: 20.20.20.10:48551
  allowed ips: 10.8.0.0/24
  latest handshake: 1 minute, 49 seconds ago
  transfer: 1.01 KiB received, 568 B sent
```

→ Clave pública PC1

PARTE OPCIONAL

Configuración básica de WireGuard

VPN-PC1:

```
: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN group default qlen 1000
  link/none
  inet 10.8.0.6/24 scope global wg0
    valid_lft forever preferred_lft forever
```

```
root@server:~# wg
interface: wg0
  public key: 178uQQiOr1TglBkoHwF7WCp1KhFwgWcy86iY9WGTCyY=
  private key: (hidden)
  listening port: 48551

peer: zmk8sITH4WkA3a1dkO5PgQX8BFZvmJ+xxwrjdnveSFI=
  endpoint: 10.10.10.100:51820
  allowed ips: 10.8.0.0/24
  latest handshake: 38 seconds ago
  transfer: 2.41 KiB received, 4.67 KiB sent
  persistent keepalive: every 25 seconds
```

→ Clave pública Servidor

PARTE OPCIONAL

Configuración básica de WireGuard

VPN-Gateway:

```
# iptables -A INPUT -i wg0 -j ACCEPT
# iptables -A OUTPUT -o wg0 -j ACCEPT
# iptables -A FORWARD -i wg0 -j ACCEPT
# iptables -A FORWARD -o wg0 -j ACCEPT
# iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

PARTE OPCIONAL

Configuración básica de WireGuard

Comprobación

PC1

```
root@server:~# curl 10.8.0.1
Este es el Servidor VPN (10.10.10.100)
root@server:~# |
```

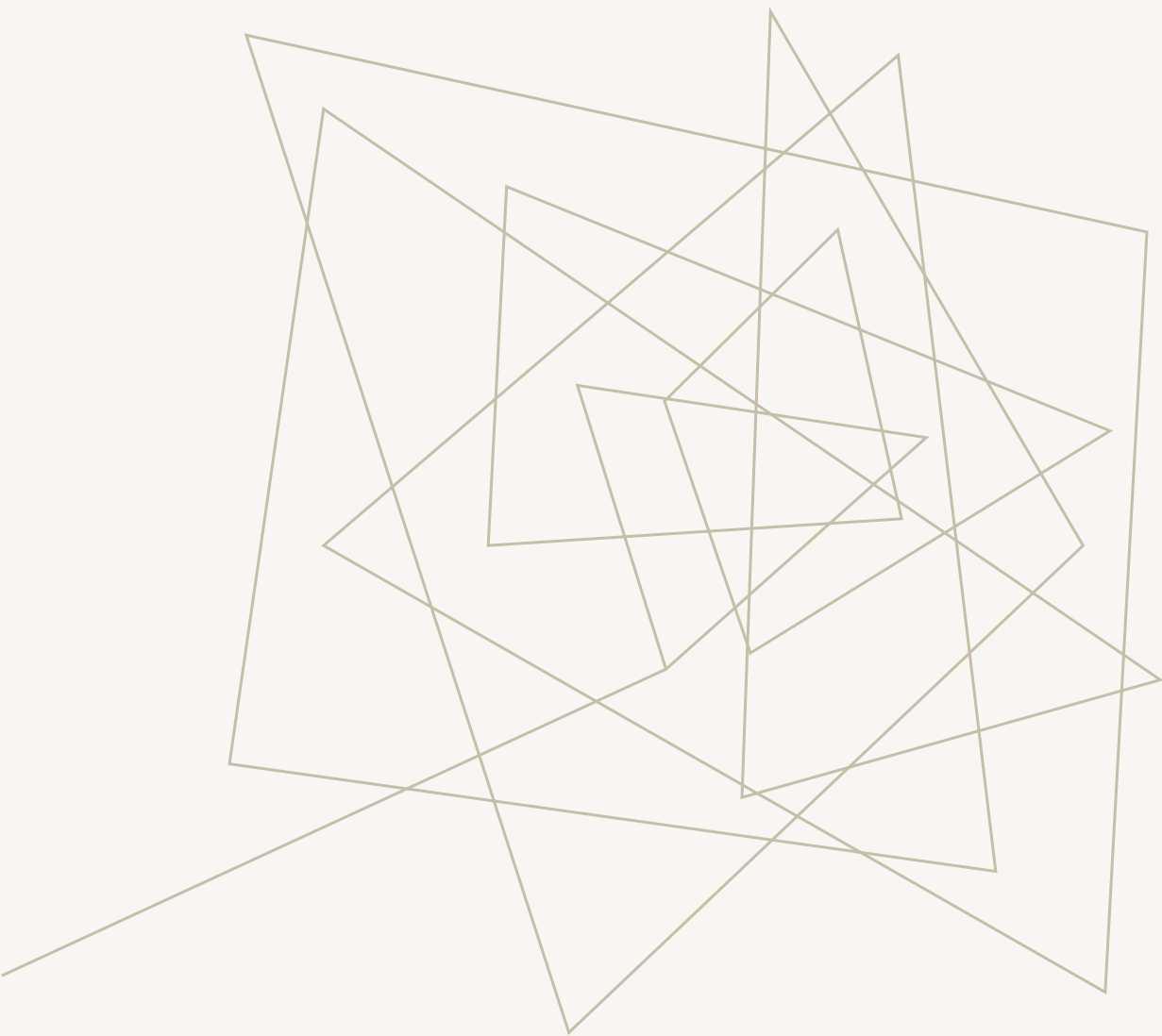
VPN-Servidor

```
root@server:~# tcpdump -i wg0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decoding
listening on wg0, link-type RAW (Raw IP), snapshot length 262144 bytes
08:02:11.239688 IP 10.8.0.6.53144 > server.http: Flags [S], seq 2310459115, win 64860, options [mss 1380,sackOK,TS val 1108976486 ecr 0,nop,wscale 7], length 0
08:02:11.240222 IP server.http > 10.8.0.6.53144: Flags [S.], seq 3990411270, ack 2310459116, win 64296, options [mss 1380,sackOK,TS val 999440617 ecr 1108976486,nop,wscale 7], length 0
08:02:11.244688 IP 10.8.0.6.53144 > server.http: Flags [S.], seq 3990411270, ack 2310459116, win 64296, options [mss 1380,sackOK,TS val 999440617 ecr 1108976486,nop,wscale 7], length 0
08:02:11.244724 IP 10.8.0.6.53144 > server.http: Flags [P.], seq 1:73, ack 1, win 507, options [nop,nop,TS val 1108976491 ecr 999440617], length 72: HTTP: GET / HTTP/1.1
08:02:11.245063 IP server.http > 10.8.0.6.53144: Flags [S.], seq 3990411270, ack 2310459116, win 64296, options [mss 1380,sackOK,TS val 999440617 ecr 1108976486,nop,wscale 7], length 0
08:02:11.262345 IP server.http > 10.8.0.6.53144: Flags [P.], seq 1:267, ack 73, win 502, options [nop,nop,TS val 999440639 ecr 1108976491], length 266: HTTP: HTTP/1.1 200 OK
08:02:11.268122 IP 10.8.0.6.53144 > server.http: Flags [S.], seq 3990411270, ack 2310459116, win 64296, options [mss 1380,sackOK,TS val 999440617 ecr 1108976486,nop,wscale 7], length 0
08:02:11.268163 IP 10.8.0.6.53144 > server.http: Flags [F.], seq 73, ack 267, win 505, options [nop,nop,TS val 1108976513 ecr 999440639], length 0
08:02:11.268407 IP server.http > 10.8.0.6.53144: Flags [F.], seq 267, ack 74, win 502, options [nop,nop,TS val 999440645 ecr 1108976513], length 0
08:02:11.271987 IP 10.8.0.6.53144 > server.http: Flags [S.], seq 3990411270, ack 2310459116, win 64296, options [mss 1380,sackOK,TS val 999440617 ecr 1108976486,nop,wscale 7], length 0
```

PARTE OPCIONAL

Bibliografía:

- [¿Qué es WireGuard? Todo lo que necesita saber \(softwarelab.org\)](https://softwarelab.org/whatis/wireguard/)
- [WireGuard vs OpenVPN: ¿qué VPN deberíamos usar? \(profesionalreview.com\)](https://profesionalreview.com/wireguard-vs-openvpn/)



COMENTARIOS TEO

COMENTARIOS TEO

- **Apdo 1.** En nuestra maqueta sólo usaremos un cliente openvpn en PC1 (no es necesario en PC2 ni en el Gateway). El paquete openvpn sólo debe instalarse en Servidor y PC1.
- **Apdo. 8.** Lo que ocurre es que PC1 envía la petición por la VPN (tiene una ruta para alcanzar 10.10.10.0/24 por tun0). El servidor recibe el mensaje y lo reenvía a PC2. PC2 sí devuelve el paquete a 10.8.0.6, pero no lo envía al servidor sino al Gateway, como indica su tabla de encaminamiento (no tiene ruta para 10.8.0.0/24 y usa su ruta por defecto). Y el Gateway elimina el paquete, al no saber cómo alcanzar la red 10.8.0.0/24.
- **Apdo 9.** Las reglas iptables de FORWARD que usáis en el Gateway (pg. 46) no son necesarias en nuestro caso (en nuestros equipos está habilitado todo el tráfico de forward). Pero tampoco hacen daño. Todo lo demás es correcto.