Wireguard

Wg-quick

`sudo ip link delete dev wg0`

Gestionar les interficies wireguard amb l'eina `wg` pot ser tediós.

Per aquest motiu es pot automatizar amb l'eina `wg-quick`.

Claus

Desde qualsevol màquina pots generar tres claus privades (i les públiques)

En el nostre cas començarem amb la màquina `net-1` :

```
box@net-1:~$ wg genkey
GLA3HBTZQKfu/xiJ6vDmQ87XKTclJrtt02rfLyJVXUw=
box@net-1:~$ echo GLA3HBTZQKfu/xiJ6vDmQ87XKTclJrtt02rfLyJVXUw= | wg pubkey
FIBBEb1DA/oFx5ZE0E4cdpidJxFEeLbblHh7vXIRZwo=
box@net-1:~$
```

Crea el fitxer `/etc/wireguard/wg0.conf`

```
  GNU nano 6.2                                                /etc/wireguard/wg0
[Interface]
PrivateKey = GLA3HBTZQKfu/xiJ6vDmQ87XKTclJrtt02rfLyJVXUw=
Address = 10.20.0.11/24
ListenPort = 51820

[Peer]
PublicKey = 418ghuxL3YSR2H7mZBOXwMS7Msa4i17uIoBvERW/uHU=
AllowedIPs = 10.20.0.12/32
Endpoint = 192.168.56.21:51820

[Peer]
PublicKey = LID4aZRmnwJ5ioVMJ3ZsxbbrlOLjm42FnGQ7un1J+ws=
AllowedIPs = 10.20.0.13/32
Endpoint = 192.168.56.22:51820

^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execut
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justif
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

box@net-1:~$ wg genkey
GLA3HBTZQKfu/xiJ6vDmQ87XKTclJrtt02rfLyJVXUw=
box@net-1:~$ echo GLA3HBTZQKfu/xiJ6vDmQ87XKTclJrtt02rfLyJVXUw= | wg pubkey
FIBBEb1DA/oFx5ZE0E4cdpidJxFEeLbblHh7vXIRZwo=
box@net-1:~$ wg genkey
kHuZvK94PA4cW/Qj6IXp8K3KyjfcPuEmg6LceTV8IkU=
box@net-1:~$ echo kHuZvK94PA4cW/Qj6IXp8K3KyjfcPuEmg6LceTV8IkU= | wg pubkey
418ghuxL3YSR2H7mZBOXwMS7Msa4i17uIoBvERW/uHU=
box@net-1:~$ wg genkey
SC3/885QNhda2yx8XWL8dFPENT1IRzufYNt7gtZ2gXU=
box@net-1:~$ echo SC3/885QNhda2yx8XWL8dFPENT1IRzufYNt7gtZ2gXU= | wg pubkey
LID4aZRmnwJ5ioVMJ3ZsxbbrlOLjm42FnGQ7un1J+ws=
box@net-1:~$
```

systemd

WireGuard es pot configurar per executar-se com a servei de sistema.

D'aquesta manera la inteficie wg es configura de manera automàtica al arrencar el sistema,

```
box@net-1:~$ sudo systemctl enable --now wg-quick@wg0.service
Created symlink /etc/systemd/system/multi-user.target.wants/wg-quick@wg0.ser
box@net-1:~$
box@net-1:~$ sudo systemctl status wg-quick@wg0.service /etc/wireguard/wg0.conf
● wg-quick@wg0.service - WireGuard via wg-quick(8) for wg0
     Loaded: loaded (/lib/systemd/system/wg-quick@.service; enabled; vendor
     Active: active (exited) since Thu 2023-11-23 09:34:07 UTC; 16s ago
       Docs: man:wg-quick(8)
             man:wg(8)
             https://www.wireguard.com/
             https://www.wireguard.com/quickstart/
```

Observa que l'ordre especifica el nom del nom del dispositiu del túnel `wg0` com a part del nom del servei.

Aquest nom s'assigna al fitxer de configuració `/etc/wireguard/wg0.conf`.

Per tant, pots crear tants túnels VPN separats com vulguis a la màquina.

Pots veure que la interficie `wg0` de `net-1` ha quedat configurada:

```
box@net-1:~$ sudo wg show wg0
interface: wg0
  public key: FIBBEb1DA/oFx5ZE0E4cdpidJxFEeLbblHh7vXIRZwo=
  private key: (hidden)
  listening port: 51820

peer: 418ghuxL3YSR2H7mZBOXwMS7Msa4i17uIoBvERW/uHU=
  endpoint: 192.168.56.21:51820
  allowed ips: 10.20.0.12/32

peer: LID4aZRmnwJ5ioVMJ3ZsxbbrlOLjm42FnGQ7un1J+ws=
  endpoint: 192.168.56.22:51820
  allowed ips: 10.20.0.13/32
box@net-1:~$
```

Activitat

Configura `net-2` i `net-3` amb `wg-quick` i verifica que la VPN funciona:

```
box@net-2:~$ sudo wg show wg0
interface: wg0
  public key: 418ghuxL3YSR2H7mZBOXwMS7Msa4i17uIoBvERW/uHU=
  private key: (hidden)
  listening port: 51820

peer: FIBBEb1DA/oFx5ZE0E4cdpidJxFEeLbblHh7vXIRZwo=
  endpoint: 192.168.56.20:51820
  allowed ips: 10.20.0.11/32
  latest handshake: 26 seconds ago
  transfer: 348 B received, 436 B sent

peer: LID4aZRmnwJ5ioVMJ3ZsxbbrlOLjm42FnGQ7un1J+ws=
  endpoint: 192.168.56.22:51820
  allowed ips: 10.20.0.13/32
box@net-2:~$
box@net-2:~$ ping -c 1 10.20.0.11
PING 10.20.0.11 (10.20.0.11) 56(84) bytes of data.
64 bytes from 10.20.0.11: icmp_seq=1 ttl=64 time=0.811 ms

--- 10.20.0.11 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.811/0.811/0.811/0.000 ms
box@net-2:~$
```

Routing

Wireguard també es pot fer servir per routing.

Connectar dos xarxes privades

1.- Para les màquines box-1 i net-1.

2.- Modifica l'adaptador 1 a mode pont perquè la interfície `enp0s3` tingui una IP de la xarxa local.

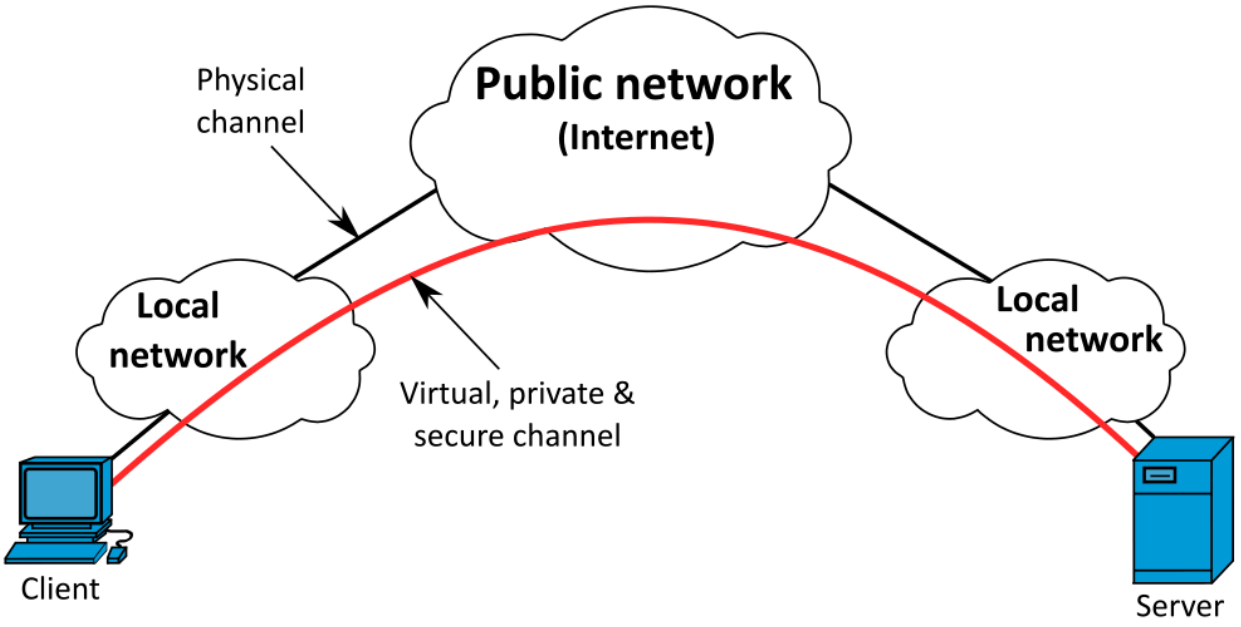
3.- Arrenca les màquines box-1 i net-1.

Atenció! Al parar `box-1` has perdut la configuració de la interficie `wg0`.

4.- Crea una xarxa virtual entre box-1 i net-1 (`wg1`)

5.- Configura box-1 i net-1 per fer routing de les xarxes privades tal com t'han ensenyat a M7.

Enrutar tràfic de clients per ocultar origen o accedir als nodes de la xarxa privada (opcional)



Fins ara hem fet servit una màscara `/32` en allowed ips per només pertmetre una connexió node a node.

Azure

En aquesta activitat utilitzarem una màquina amb IP pública al núvol d'Azure.

Abans de començar actualitza l'aplicació box.exe:

```
> sl ~
> rm .\box.exe
> curl.exe https://box.xtec.dev/box.exe -o box.exe

> .\box.exe -V
box 0.1.4
```

Recorda que al final de la pràctica has de borrar azure!

```
.\box.exe  azure remove
.\box.exe  azure disconnect
```

Recorda que has d'obrir el port pertinent a Azure.

Tutorial

Segueix aquest tutorial:

https://www.tangramvision.com/blog/what-they-dont-tell-you-about-setting-up-a-wireguard-vpn

TODO

Peer

Una interfície wireguard fa servir UDP (no es pot atacar amd DDOS) i és indetectable amb nmap.

Ja saps de ASIX-11-UF2-1 - Nmap que és molt difíil detectar ports UDP.

Una interfície wireguard només respon a un paquet UDP si el remitent és un node de confiança.

Un node és de confiança si has registrat la seva clau pública com a un peer.

Per tant, configura la màquina box-2 amb una interficie wireguard.

```
box@box-2:~$ ip --brief addr
lo               UNKNOWN        127.0.0.1/8 ::1/128
enp0s3           UP             10.0.2.15/24 metric 100 fe80::a00:27ff:fede
enp0s8           UP             192.168.56.18/24 fe80::a00:27ff:fef8:b961/6
wg0              UNKNOWN        10.10.0.12/24
box@box-2:~$
```

Mira quina és la id de la intefície `wg0` de `box-2` i en quin port està escoltant:

```
box@box-2:~$ sudo wg show wg0
interface: wg0
  public key: gm7ZXWBXCKD0kvJFqsLgfwsfqoGpPZHHDosuWDr3Uh4=
  private key: (hidden)
  listening port: 43381
box@box-2:~$
```

Ja pots configura a `box-2` com a peer de `box-1`:

```
interface: wg0
  public key: gm7ZXWBXCKD0kvJFqsLgfwsfqoGpPZHHDosuWDr3Uh4=
  private key: (hidden)
  listening port: 43381
box@box-2:~$ ip --brief addr
lo               UNKNOWN        127.0.0.1/8 ::1/128
enp0s3           UP             10.0.2.15/24 metric 100 fe80::a00:27ff:fec
enp0s8           UP             192.168.56.18/24 fe80::a00:27ff:fef8:b961/
wg0              UNKNOWN        10.10.0.12/24
box@box-2:~$
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

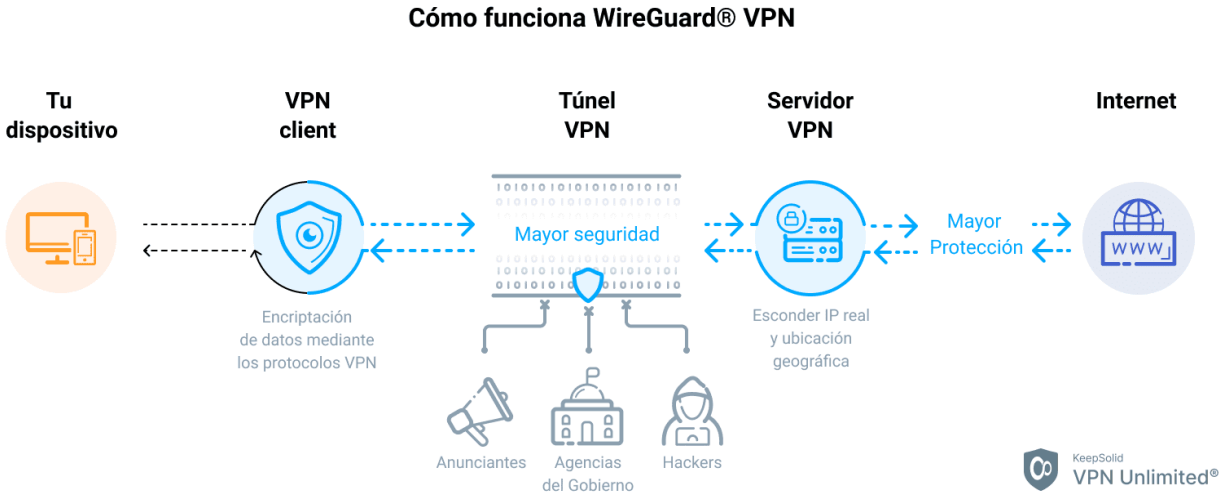
```
box@box-1:~$ sudo wg set wg0 \
> peer gm7ZXWBXCKD0kvJFqsLgfwsfqoGpPZHHDosuWDr3Uh4= \
> allowed-ips 10.10.0.12/32 \
> endpoint 192.168.56.18:43381
box@box-1:~$
box@box-1:~$ sudo wg show wg0
interface: wg0
  public key: tsWs/ezeFqOm1RCR8C1hqdadFKYFuu/9+2YwdMYVaSg=
  private key: (hidden)
  listening port: 51820

peer: gm7ZXWBXCKD0kvJFqsLgfwsfqoGpPZHHDosuWDr3Uh4=
  endpoint: 192.168.56.18:43381
  allowed ips: 10.10.0.12/32
box@box-1:~$
```

Torna a activar la interficie `wg0` de `box-1` :

```
box@box-1:~$ sudo ip link set wg0 up
box@box-1:~$
```

Fixa't que has de dir quina IP "real" ha de fer servir per connectar-se a l'altre node:



Wireguard crea un túnel encriptat sobre una altre connexioó IP/UDP.

Si proves si funciona la connexió entre `box-1` i `box-2` veuras que no funciona:

```
box@box-1:~$ ping 10.10.0.12
PING 10.10.0.12 (10.10.0.12) 56(84) bytes of data.
```

Peer vol dir parella, per tant hem d'aparellar els dos nodes fent que `box-2` accepti connexions de `box-1`:

Torna a activar la interficie `wg0` de `box-2` :

```
box@box-2:~$ sudo ip link set wg0 up
box@box-2:~$
```

Ja pots fer un ping entre les màquines:

```
box@box-1:~$ ping -c 1 10.10.0.12
PING 10.10.0.12 (10.10.0.12) 56(84) bytes of data.
64 bytes from 10.10.0.12: icmp_seq=1 ttl=64 time=0.941 ms

--- 10.10.0.12 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.941/0.941/0.941/0.000 ms
box@box-1:~$
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
box@box-2:~$ ping -c 1 10.10.0.11
PING 10.10.0.11 (10.10.0.11) 56(84) bytes of data.
64 bytes from 10.10.0.11: icmp_seq=1 ttl=64 time=0.858 ms

--- 10.10.0.11 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.858/0.858/0.858/0.000 ms
box@box-2:~$
```

Però com pots estar segur de que el peer està ben configurat?

Aixeca un servidor nginx a `box-2` que respongui amb el seu nom:

```
box@box-2:~$ sudo apt install -y nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.18.0-6ubuntu14.4).
0 upgraded, 0 newly installed, 0 to remove and 24 not upgraded.
box@box-2:~$
box@box-2:~$ echo "box-2" | sudo tee /var/www/html/index.html
box-2
box@box-2:~$
```

Verifica que et pots conectar al servidor web de l'altre node:

```
box@box-1:~$ curl 10.10.0.12
box-2
box@box-1:~$
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
box@box-2:~$ curl 10.10.0.11
box-1
box@box-2:~$
```

Peer

WireGuard associa adreces IP del túnel amb claus públiques (`public key`) i punts finals remots (`endpoint`).

		box-1	box-2
dev		`10.10.0.11/24`	`10.10.0.12/24`
wg0			
	public key	`gm7ZXWBXCKD0kvJFqsLgfwsfqoGpPZHHDosuWDr3Uh4=`	`tsWs/ezeFqOm1RCR8C1hqdadFKYFuu/9+2YwdMYVaSg=`
	listening	`51820`	`43381`
peer			
	pubic key	`tsWs/ezeFqOm1RCR8C1hqdadFKYFuu/9+2YwdMYVaSg=`	`gm7ZXWBXCKD0kvJFqsLgfwsfqoGpPZHHDosuWDr3Uh4=`
	endpoint	`192.168.56.18:43381`	`192.168.56.17:51820`
	allowed-ips	`10.10.0.12/32`	`10.10.0.11/32`

Exemple

Quan executes un `ping 10.10.10.12` desde `box-1` s'executa aquesta lògica:

1. L'adreça `10.10.0.12` pertany a la mateixa xarxa que `10.10.0.11/24`.
2. El paquet s'envia per la interficie `wg0`.
3. Existeix un peer amb allowed-ips `10.10.0.12/32` que fa match amb la IP de destí.
4. El paquet es xifra amb la clau pública del peer `tsWs/ezeFqOm1RCR8C1hqdadFKYFuu/9+2YwdMYVaSg=`
5. El paquet s'envia a l'endpoint `192.168.56.18:43381` mijantçant UDP

Quan `box-2` rep un paquet al port `43381` a nivell de kernel s'envia a la interficie `wg0` per ser processat:

1. El paquet té IP origen `10.10.0.11`.
2. Hi ha un peer configurat amb allowed-ips 10.10.0.11/32
3. El peer té clau pública `gm7ZXWBXCKD0kvJFqsLgfwsfqoGpPZHHDosuWDr3Uh4=`
4. Desxifro i autentico el paquet
5. Si tot és correcto continuo processant el paquet, i si no és correcte, elimino el paquet i no contesto (per això wireguard és indetectable per nmap)

6. Envio el paquet per la interficie `wg0` per ser processat per l'aplicació corresponent.

Seguretat

Amb wireguard:

1. Tots els paquets enviats a la interfície WireGuard estan xifrats i autenticats
2. Hi ha un acoblament molt estret entre la identitat d'un peer i l'adreça IP permesa d'un peer

Per tant:

1. No es necessita cap firewall
2. Pots estar segur de la identitat de l'altre node
3. Tot estar encriptat i protegit encara que les aplicacions no encriptin els seus missagtes.
4. Pots crear les teves xarxes privades com i quan vulguis.

Activitat

Has de configurar el reste de peers:

1. Entre `box-1` i `box-3`
2. Entre `box-2` i `box-3`