Wireguard

E' stato installato su una VM debian su hyper-v un server wireguard, sull'indirizzo rttswg.units.it. La VM ha una interfaccia gigabit ethernet. Risponde al nome rttswg.units.it (router trieste wireguard).

Wireguard

Wireguard è un protocollo per VPN sviluppato recentemente (2019) con la semplicità e l'efficienza in mente. Non ha possibilità di scegliere il tipo di crittografia e scambio chiavi, punta ad essere sicuro "by default". Non è quantum safe di default, ma con l'aggiunta di una chiave condivisa riesce ad esserlo.

A livello di rete, è basato su UDP, di default in porta 51820

Essendo basato su UDP abbiamo dovuto dire al fortinet di "rilassare" alcune protezioni verso l'host.

Configurazione server

La configurazione è stata fatta alla via debian classica, in /etc/network/interfaces

```
iface wg0 inet static
    address 10.180.0.254
    netmask    255.255.255.0
    pre-up ip link add $IFACE type wireguard
    pre-up wg setconf $IFACE /etc/wireguard/wg0.conf
    post-up ip route add 172.30.218.0/24 via 10.180.0.16 dev wg0
    post-up ip route add 172.30.219.0/24 via 10.180.0.12 dev wg0
    post-down ip link del $IFACE
    mtu 1420
```

Da notare: se è necessario instradare reti aggiuntive è necessario aggiungere le rotte sia in questo file di configurazione, sia a livello di infrastruttura (Fortinet).

I dettagli sul protocollo sono presenti in /etc/wireguard/wg0.conf , lo riporto per commentarlo senza le chiavi.

```
[Interface]
ListenPort = 51820
PrivateKey = [chiaveprivatadelserver]

[Peer]
PublicKey = [chiavepubblicadelserver]
AllowedIPs = 10.180.0.X/32, [indirizzodieventualiretiremote/mask]
PresharedKey = [presharedkeycondivisatraserveresingoloclient]
```

E' importante precisare sul server le reti per ogni singolo peer senza sovrapposizioni, pena ogni sorta

di peste bubbonica...

Dopo eventuali cambi di configurazione è necessario un

```
ifdown wg0; ifup wg0
```

Se qualcosa fosse andato storto, e quindi non torna su il server, prima di riprovare dare un

```
ip link delete wg0
```

Gestione chiavi

Per ogni peer son necessarie 3 chiavi:

- chiave privata, da cui deriva la
- chiave pubblica
- preshared key, indipendente dalle altre 2

Per generare le chiavi ho usato un simpatico tool https://github.com/warner/wireguard-vanity-address che consente di fare sì che le chiavi pubbliche abbiano un pezzo della stringa che rimanda al nome del client. Non è installato sul server, l'ho usato dalla mia workstation, per esempio:

```
wireguard-vanity-address --in 3 ln1
```

Non esagerate col numero di caratteri, se no ci potete passare le giornate ad aspettare un output.

Per generare le psk invece

```
wg genpsk
```

Per essere almeno vagamente ordinati, questi li salviamo in /etc/wireguard in sottocartelle dai nomi significativi.

Troubleshooting

Il primo strumento, da root su rttswg è il comando wg

```
interface: wg0
  public key: RttSVBFHOCRlcRUWpjAthEbpG1v+Q3VixM3kxX3EMU4=
  private key: (hidden)
  listening port: 51820

peer: LN1N/peG592O2MNzBMqA6JzEpmLpMBGcBZirx2/syV4=
  preshared key: (hidden)
  endpoint: 5.91.30.169:15054
  allowed ips: 192.168.8.0/24, 10.180.0.232/32
  latest handshake: 1 minute, 47 seconds ago
```

```
  transfer: 33.70 MiB received, 21.98 MiB sent

peer: wrt3Vr+Yiiw7qKtHv7qqr+TGWzUNuqiHoqTOwQbg9gg=
  preshared key: (hidden)
  allowed ips: 192.168.3.0/24, 10.180.0.248/32
```

Riporto solo 2 entry a titolo di esempio. La prima è una connessione attiva, la seconda è disattiva.

Se qualcosa non va, e le connessioni sono su, è altamente probabile uno scazzo di routing, o sul server o a livello di infrastruttura, o il NAT sull'endpoint che rompe le scatole.

Configurazione client

Riporto a titolo di esempio una configurazione funzionante per un client:

```
[Interface]
Address = 10.180.0.X/32
PrivateKey = [chiaveprivatadelclient]
DNS = 140.105.114.55, 140.105.114.66
MTU = 1420

[Peer]
PublicKey = RttSVBFHOCRlcRUWpjAthEbpG1v+Q3VixM3kxX3EMU4=
PresharedKey = [presharedkeycondivisatraserveresingoloclient]
Endpoint = rttswg.units.it:51820
AllowedIPs = 0.0.0.0/0
PersistentKeepalive = 21
```

I Gl.Inet non hanno un comportamento coerente nella gestione del passaggio della configurazione. Le macchine col modem 4G richiedono di passargli proprio un file di configurazione. Le macchine solo ethernet è possibile inserire la configurazione “al volo”.