OpenVPN

Domenico Iezzi

domenico.iezzi.201@gmail.com

Cos'è una VPN

VPN (Virtual Private Network) è un insieme di tool che permette la connessione di più reti private in maniera sicura, attraverso una rete pubblica.

virtuale: utenti possono comunicare tra di loro come se fossero connessi in una LAN

privata: accessibile in maniera sicura solo dagli utenti autorizzati

Utilizzi più comuni

- Connettersi a internet in maniera sicura (es. wifi pubblica)
- Bypassare eventuali restrizioni di rete (es. firewall)
- Esporre tutte le risorse di una rete remota (es. ufficio) in modo sicuro ed efficente
- Connettere reti di un organizzazione situate in diverse parti del mondo

IPSec

Primo standard per il networking sicuro (1995)

- Protocollo implementato nel layer 3 (network)
- Richiede varie funzioni per poter verificare integrità e autenticità di un pacchetto
- Richiede un'implementazione del protocollo nei vari OS

Progetto open source sviluppato da James Yonan nel 2001 come alternativa ad IPSec.

- Riutilizza protocolli già esistenti
  - UDP/TCP
  - application level security tramite SSL/TLS
- interfacce di rete virtuali per il tunneling del traffico

Transport protocols

OpenVPN può utilizzare due protocolli per il trasporto dei dati: **TCP** o **UDP**.

TCP

Il protocollo TCP è un protocollo affidabile che utilizza un network non sempre affidabile. In caso un pacchetto non arrivi a destinazione, viene ritrasmesso con un timeout che aumenta esponenzialmente ad ogni retry

Non è considerato una buona opzione per una VPN in quanto

- Incapsulando il TCP su TCP, stiamo generando uno strato aggiuntivo di inaffidabilità
- TCP meltdown

UDP

Considerato un protocollo più adatto per una VPN, in quanto

- connectionless: dati vanno in una sola direzione
- unreliable: spetta all'applicazione gestire eventuali problemi con i pacchetti
- unordered: il protocollo non gestisce il caso in cui i pacchetti arrivino in ordine sparso rispetto a come sono stati inviati

Virtual network interfaces

Interfaccia di rete **virtuale**, cioè non associata direttamente ad un'interfaccia di rete fisica, ma astratta dal kernel.

Dati in arrivo su tale interfaccia vengono inviati al demone OpenVPN che si occupa di cifrare e incapsulare questi dati nel protocollo scelto

Due tipologie principali sono supportate dal kernel linux:

- **TAP**
- **TUN**

TAP

TAP viene utilizzato con il layer 2, quindi si comporta come una vera e propia scheda di rete

- layer 2 → ethernet frames
- supporta qualsiasi network protocol
- è un'interfaccia di rete "full-fledged"

Svantaggi:

- Overhead maggiore rispetto a TUN

TUN

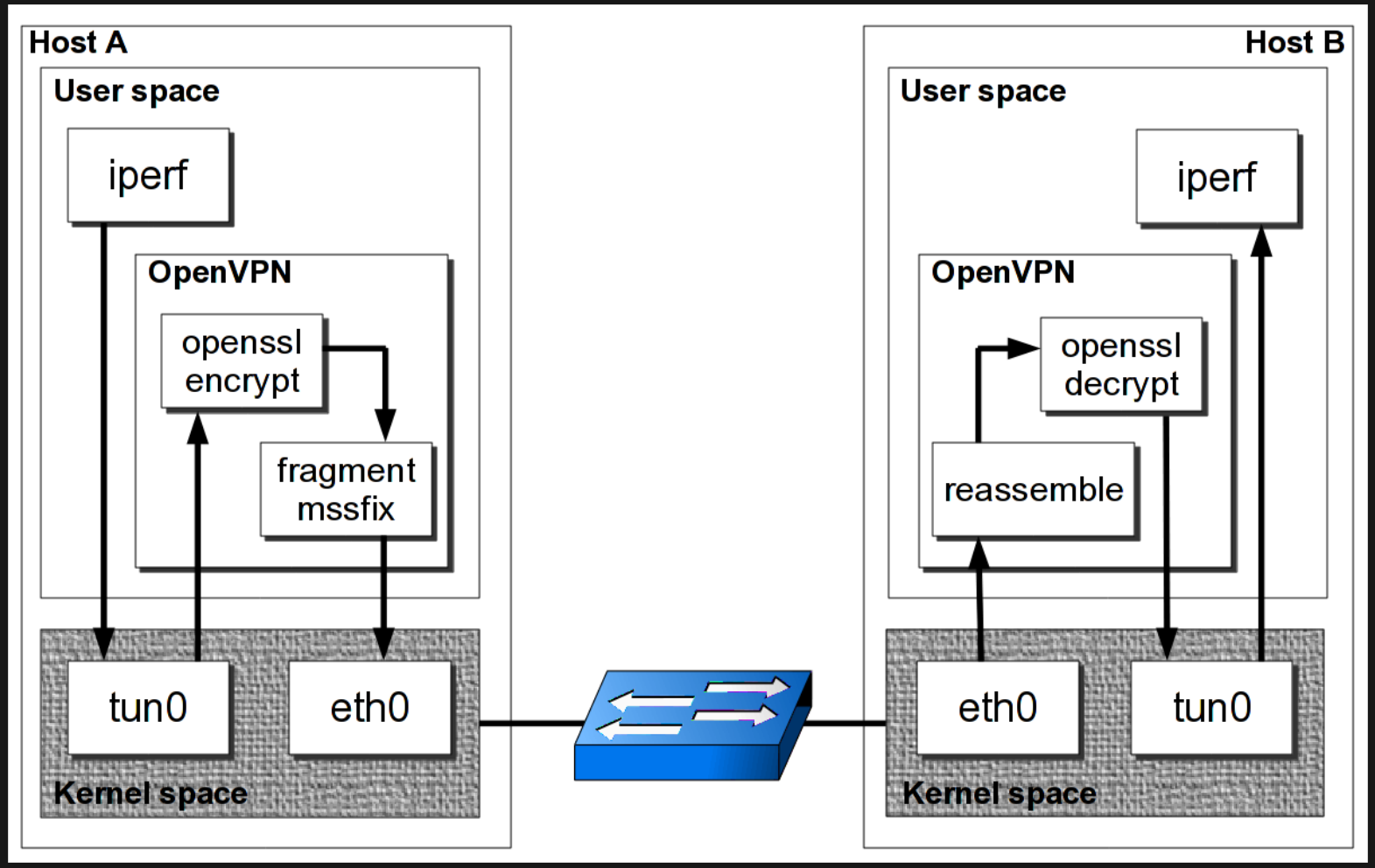
TUN invece opera in layer 3 e permette il transito di pacchetti IPV4/6

- overhead minore dato che non deve mandare frame ethernet ma pacchetti IP
- Mantiene comunque le possibilità di routing con iptables
- Utilizzato nella maggior parte dei casi a meno di condizioni particolari

Svantaggi:

- Solo pacchetti IP
- No broadcast

packet flow


Host A
User space
iperf
OpenVPN
openssl
encrypt
fragment
mssfix
tun0
eth0
Kernel space
Host B
User space
iperf
OpenVPN
openssl
decrypt
reassemble
eth0
tun0
Kernel space

Security

OpenVPN implementa due metodi per cifrare la connessione:

- **Static key**: utilizza una chiave pre-condivisa
- **TLS**: SSL/TLS + certificati per autenticazione e scambio di chiavi

The easy way

OpenVPN è in grado di generare una chiave statica per cifrare la connessione.

```
openvpn --genkey --secret secret.key
```

Un file di configurazione minimale per il server:

```
dev tun
ifconfig 10.8.0.1 10.8.0.2
secret static.key
```

Semplice e veloce, comodo per fare testing ma in generale sconsigliato per motivi di sicurezza (more)

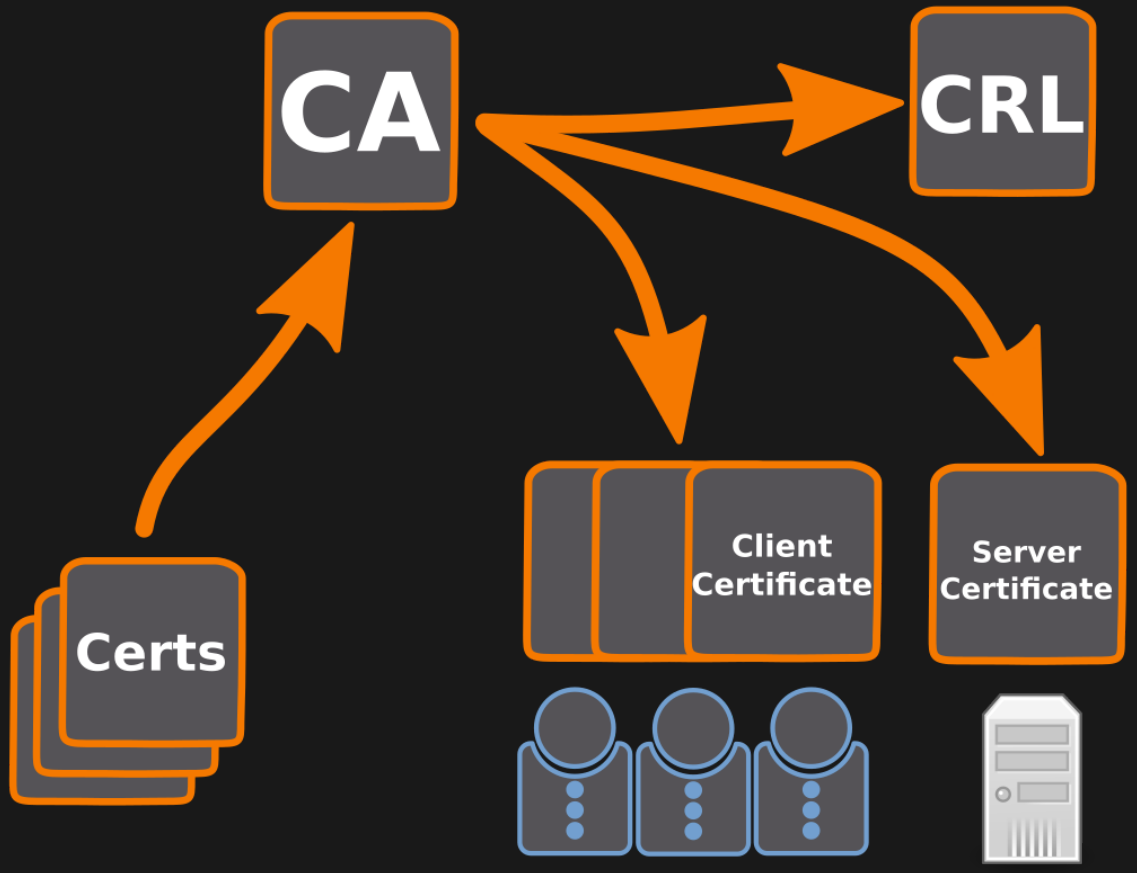
TLS

La modalità TLS consiste in una autenticazione bidirezionale tramite certificato. Una volta autenticati, client e server possono scambiarsi chiavi in maniera sicura

Vantaggi di questo approccio:

- È un sistema più robusto dal punto di vista della sicurezza
- Ogni utente è identificato da un certificato
- È possibile revocare i certificati individualmente
- Si possono creare delle CA intermedie

PKI



PKI

Gli autori di openvpn hanno sviluppato un comodo tool per permettere il setup di una PKI, la generazione e la firma di certificati, chiamato **EasyRSA**

EasyRSA 3 è la versione più recente. La versione presente nei repository delle maggiori distro GNU/Linux potrebbe essere piuttosto datata (ad esempio 2.2.2 in Ubuntu 16.04), tuttavia è possibile clonare il repository git e utilizzare direttamente il tool senza dover installare nulla

```
git clone https://github.com/OpenVPN/easy-rsa.git
cd easy-rsa/easyrsa3
```

Configurare una PKI

Inizializziamo la PKI:

```
./easyrsa init-pki

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /your/path/to/easyrsa3/pki
```

Creiamo il root certificate:

```
./easyrsa build-ca
Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
Generating RSA private key, 2048 bit long modulus

[..]

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/your/path/to/easyrsa3/pki/ca.crt
```

Configurare una PKI

Creiamo un certificato per il nostro server:

```
./easyrsa build-server-full server
Generating a 2048 bit RSA private key
writing new private key to '/path/to/your/easyrsa3/pki/private/pre.key.o4vxT8HKMX'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:

[..]

The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'server'
Certificate is to be certified until Mar 27 17:33:28 2028 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated
```

Analogamente usiamo `./easyrsa build-client-full client` per creare un certificato per il client

Concludiamo creando i parametri per lo scambio di chiavi Diffie-Helman

```
./easyrsa gen-dh
[..]
DH parameters of size 2048 created at /your/path/to/easyrsa3/pki/dh.pem
```

Dove metto cosa?

```
pki
|-- ca.crt
|-- dh.pem
|-- issued
|   |-- client.crt
|   `-- server.crt
`-- private
    |-- ca.key
    |-- client.key
    `-- server.key
```

- ca.crt può essere dato a chiunque dato che serve per verificare la catena di certificati (serve sia al server che al client)
- client.key, client.crt vanno dati al client (reminder: ogni client ha bisogno del propio certificato!)
- server.key, server.crt, dh.pem sono utilizzati dal server
- ca.key deve essere riposto al sicuro

Hands On

E' possibile configurare il demone OpenVPN tramite parametri a riga di comando:

openvpn [--askpass [file]] [--auth-nocache] [--auth-retry type] [--auth-user-pass-verify script] [--auth -user-pass up] [--auth alg] [--bcast-buffers n] [--ca file] [--ccd-exclusive] [--cd dir] [--cert file] [--chroot dir] [--cipher alg] [--client-cert-not-required] [--client-config-dir dir] [--client-connect script] [--client-disconnect] [--client-to-client] [--client] [--comp-lzo] [--comp-noadapt] [--config file] [--connect-freq n sec] [--connect-retry n] [--crl-verify crl] [--cryptoapicert select-string] [--daemon [progname]] [--dev-node node] [--dev-type device-type] [--dev tunX | tapX | null] [--dev tunX | tapX] [--dh file] [--dhcp-option type [parm]] [-- dhcp-release] [--dhcp-renew] [--disable-occ] [--disable] [--down-pre] [--down cmd] [--duplicate-cn] [--echo [parms...]] [--engine [engine-name]] [--explicit-exit-notify [n]] [--fast-io] [--float] [--fragment max] [--genkey] [--group group] [--hand-window n] [--hash-size r v] [--help] [--http-proxy-option type [parm]] [--http-proxy-retry] [--http-proxy-timeout n] [--http -proxy server port [authfile] [auth-method]] [--ifconfig-noexec] [--ifconfig-nowarn] [--ifconfig-pool-linear] [--ifconfig-pool-persist file [seconds]] [--ifconfig-pool start-IP end-IP [netmask]] [--ifconfig-push local remote-netmask] [--ifconfig l rn] [--inactive n] [--inetd [wait|nowait] [progname]] [--ip-win32 method] [--ipchange cmd] [--iroute network [netmask]] [--keepalive n m] [--key-method m] [--key file] [--keysize n] [--learn-address cmd] [--link-mtu n] [--local host] [--log-append file] [--log file] [--suppress-timestamps] [--lport port] [--management-hold] [--management-log-cache n] [--management-query-passwords] [--management IP port [pw-file]] [--max-clients n] [--max-routes-per-client n] [--mktun] [--mlock] [--mode m] [--mssfix max] [-- mtu-disc type] [--mtu-test] [--mute-replay-warnings] [--mute n] [--nice n] [--no-iv] [--no-replay] [--nobind] [--ns-cert-type client|server] [--passtos] [--pause-exit] [--persist-key] [--persist-local-ip] [--persist-remote-ip] [--persist-tun] [--ping-exit n] [--ping-restart n] [--ping-timer-rem] [--ping n] [--pkcs12 file] [--plugin module-pathname init-string] [--port p ort] [--proto p] [--pull] [--push-reset] [--push "option"] [--rcvbuf size] [--redirect-gateway ["local"] ["def1"]] [--remap-usr1 signal] [--remote-random] [--remote host [port]] [--reneg- bytes n] [--reneg-pkts n] [--reneg-sec n] [--replay-persist file] [--replay-window n [t]] [--resolv-retry n] [--rmtun] [--route-delay [n] [w]] [--route-gateway gw] [--route-method m] [--r oute-noexec] [--route-up cmd] [--route network [netmask] [gateway] [metric]] [--rport port] [--secret file [direction]] [--secret file] [--server-bridge gateway netmask pool-start-IP pool-end-IP] [--server network netmask] [--service exit-event [0|1]] [--setenv name value] [--shaper n] [--show-adapters] [--show-ciphers] [--show-digests] [--show-engines] [--show-net-up] [--show-ne t] [--show-tls] [--show-valid-subnets] [--single-session] [--sndbuf size] [--socks-proxy-retry] [--socks-proxy server [port]] [--status file [n]] [--status-version n] [--syslog [progname]] [--tap-sleep n] [--tcp-queue-limit n] [--test-crypto] [--tls-auth file [direction]] [--tls-cipher l] [--tls-client] [--tls-exit] [--tls-remote x509name] [--tls-server] [--tls-timeout n] [--tls-verify cmd] [--tmp-dir dir] [--tran-window n] [--tun-ipv6] [--tun-mtu-extra n] [--tun-mtu n] [--txqueuelen n] [--up-delay] [--up-restart] [--up cmd] [--user user] [--username-as -common-name] [--verb n] [--writepid file]

Fortunatamente è possibile specificare le opzioni in un comodo file di configurazione. La directory di default utilizzata per i file di configurazione e certificati è `/etc/openvpn`.

Inoltre è possibile specificare una subdirectory per ogni configurazione. Ad esempio in Debian 9 avremo due subdirectory di default:

- `/etc/openvpn/client`
- `/etc/openvpn/server`

Dopo aver installato openvpn dai repository della propria distro, è possibile trovare degli esempi di configurazione ben documentati nei seguenti path:

- Debian: `/usr/share/doc/openvpn/examples/`
- Archlinux: `/usr/share/openvpn/examples/`

Questi file costituiscono un ottimo punto di partenza per la configurazione del proprio client/server OpenVPN.

Configurazione Server

Copiamo i file precedentemente creati tramite EasyRSA nella directory openvpn

```
# cp pki/issued/server.crt pki/dh.pem pki/ca.crt pki/private/server.k
```

Copiamo il file di configurazione di esempio:

```
# cd /etc/openvpn/server
# gzip -dck /usr/share/doc/openvpn/examples/sample-config-files/serve
# vim server.conf
```

Oppure creiamone uno da zero

```
# cd /etc/openvpn
# vim server.conf
```

Innanzitutto specifichiamo la porta su cui restare in ascolto (default 1194), protocollo e tipo di interfaccia virtuale da utilizzare

```
port 1194
proto udp
dev tun
```

Specifichiamo i file contenenti chiavi e certificati

```
ca ca.crt
cert server.crt
key server.key
dh dh.pem
```

N.B. sono stati utilizzati path relativi in quanto abbiamo copiato tutti i file necessari nella working directory di OpenVPN

server

Crea un pool di indirizzi ip da assegnare ai client che si connettono

```
server 192.168.2.0 255.255.255.0
```

```
server-ipv6 fd00::/64
```

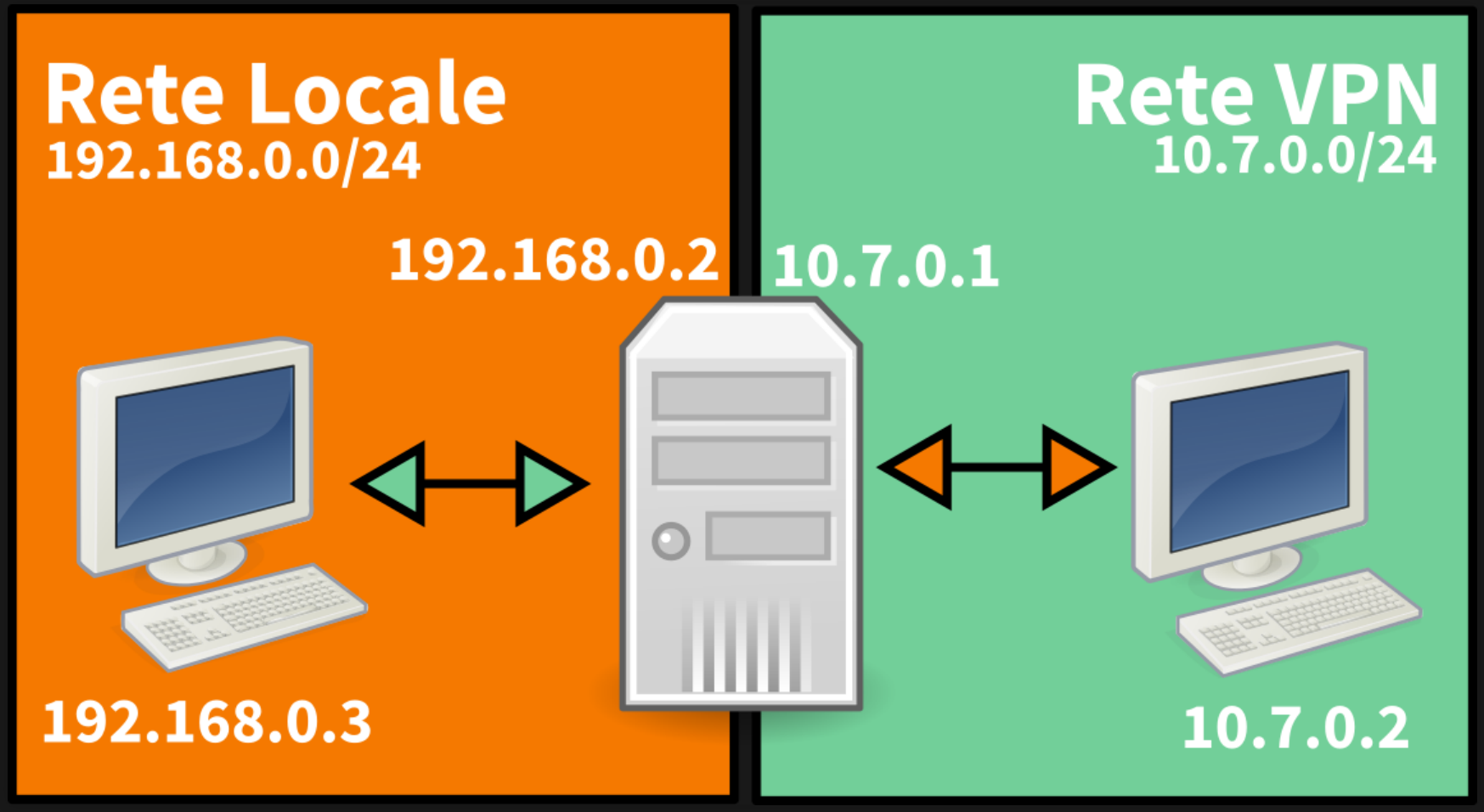
push

Push permette di inviare impostazioni ai client che si connettono, in questo modo si possono comunicare informazioni su:

- rotte
- se utilizzare o meno la compressione
- impostazioni dhcp

N.B.: i client che si connettono devono avere l'opzione `pull` (o `client`) per ricevere le configurazioni tramite push

esempio rotte




```
push "route 192.168.0.0 255.255.255.0"
```

redirect-gateway

redirect-gateway: è una direttiva che redirige tutto il traffico sulla vpn, questo lo fa in 3 passi:

1. crea una rotta statica per il server remoto attraverso il default gatweay pre-esistente
2. cancellare la rotta al default gateway
3. mettere l'ip (virtuale) del server vpn come gateway di default

compressione

Abilitare la compressione permette di risparmiare banda aggiungendo dell'overhead e sacrificando un po' di RAM e CPU

Utilizzare l'opzione `comp-lzo` (deprecata) se si utilizzano client la cui versione è minore di 2.4

Altrimenti le opzioni consigliate sono

```
compress lz4-v2
push "compress lz4-v2"
```

esempio

```
server 10.11.0.0 255.255.255.0
push "redirect-gateway def1 bypass-dhcp"
push "route 192.168.20.0 255.255.255.0"
push "dhcp-option DNS 8.8.8.8"
push "dhcp-option DNS 8.8.4.4"
compress lz4-v2
push "compress lz4-v2"
```

altri parametri consigliati

messaggio tipo ping da client a server ogni 10 secondi. Niente risposta in 120s = offline.

```
keepalive 10 120
```

Configurazione TLS consigliata

```
tls-version-min 1.2
tls-cipher TLS-DHE-RSA-WITH-AES-256-GCM-SHA384:TLS-DHE-RSA-WITH-AES-128-GCM-SHA256:TLS-DHE-RS
cipher AES-256-CBC
auth SHA512
```

altri parametri consigliati

La chiave statica utilizzata per il setup "easy" può essere utilizzata per aggiungere una firma HMAC ad ogni pacchetto. Ogni pacchetto che non presenta una firma HMAC corretta verrà droppato. Può essere generata tramite comando

```
# openvpn --genkey --secret /etc/openvpn/server/ta.key
```

Sulla configurazione inseriremo (0 se configurazione server, 1 se configurazione client)

```
tls-auth ta.key 0
```

N.B. usare lo stesso file su server e tutti i client

Permette ai client connessi alla VPN di poter comunicare tra loro

```
client-to-client
```

Altri parametri consigliati

Mantenere un record client - virtual address. Se un client si riconnette, gli verrà assegnato l'IP precedente

```
ifconfig-pool-persist ipp.txt
```

Ridurre privilegi del demone openvpn

```
user nobody
group nogroup
```

firewall

Se il vostro server è protetto da firewall, è necessario aggiungere delle regole per il corretto forwarding dei pacchetti tra le varie interfacce.

Abilitare pacchetti UDP in input nella porta scelta

```
# iptables -A INPUT -p udp --dport 1194 -j ACCEPT
```

Abilitare il forwarding tra l'interfaccia virtuale e quella fisica

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

NAT

```
# iptables -t nat -A POSTROUTING -s 10.11.0.0/24 -o eth0 -j MASQUERADE
```

avviare il demone

In genere il pacchetto OpenVPN presente in tutte le distro contiene anche il relativo service file di systemd. In caso avessimo utilizzato una subdirectory, possiamo lanciare il demone con:

```
sudo systemctl start openvpn-(client|server)@<conf-file-name>.service
```

In caso avessimo copiato i file direttamente in `/etc/openvpn`:

```
sudo systemctl start openvpn@<conf-file-name>.service
```

Nel nostro caso:

```
sudo systemctl start openvpn-server@server.service
```

configurazione client

La configurazione client è molto simile a quella server. In questo caso chiameremo il nostro file di configurazione `client.conf`, e sposteremo tutti i file necessari nella directory `/etc/openvpn/client`. Per connetterci al nostro server sono necessarie le seguenti opzioni

```
client
remote myvpnserver.com 1194
ca ca.crt
cert client.crt
key client.key
tls-auth ta.key 1
comp-lzo            # se lo avete abilitato su server
auth SHA512         # se lo avete specificato su server
cipher AES-256-CBC  # se lo avete specificato su server
```

Per connetterci alla VPN:

```
sudo systemctl start openvpn-client@client.service
```

profili .ovpn

E' possibile concatenare il file di configurazione con i relativi certificati/chiavi client in un unico file tramite il tool ovpngen

```
# ls
ca.crt  client.crt  client.key ta.key
# /path/to/ovpngen myvpnhost.com ca.crt client.crt client.key ta.key > client.ovpn
# vim client.ovpn
```

Una volta generato è possibile importare il profile nelle app OpenVPN di vari OS, in modo da potersi connettere senza dover configurare nulla

Fonti e link utili

- ArchWiki OpenVPN page
- ArchWiki Easy-RSA page
- Corsi Linux Avanzati 2017
- man page
- OpenVPN protocol
- crypto101
- RFC 1541 2132 3315
- manpage dei relativi programmi

Grazie per l'attenzione!

https://www.poul.org/