Passo 1 - Generare la master Certificate Authority (CA)

(https://openvpn.net/index.php/open-source/documentation/howto.html)

Recuperare easy-rsa (ver 2) o cercarla nell'installazione dei propri pacchetti binari
(in linux **/usr/share/doc/packages/openvpn** or **/usr/share/doc/openvpn)** meglio copiare il tutto in **/etc/openvpn**

Edit file: **`vars`**
settare le varaibili KEY_SIZE, KEY_COUNTRY, KEY_PROVINCE, KEY_CITY, KEY_ORG, and KEY_EMAIL
non lasciare nulla in bianco !!!

initializzre la PKI, per Linux/BSD/Unix:

```
# source ./vars
# ./clean-all
# ./build-ca
```

build-ca crea la certificate authority (CA) usando openssl

```
easy-rsa # ./build-ca
```

```
Generating a 1024 bit RSA private key
............++++++
...........++++++
writing new private key to 'ca.key'
-----
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.
-----
Country Name (2 letter code) [KG]:
State or Province Name (full name) [NA]:
Locality Name (eg, city) [BISHKEK]:
Organization Name (eg, company) [OpenVPN-TEST]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:OpenVPN-CA
Email Address [me@myhost.mydomain]:
```

Passo 2 – Generare certificato e key per server

Generare un certificato e una key privata per il serverOn Linux/BSD/Unix:

```
# ./build-key-server server
```

con il passaggio precedente molti parametri sono stati definiti di default.
Quando il **Common Name** viene richiesto inserire **server** e rispondere Y alle altre due domande:
"Sign the certificate? [y/n]"
"1 out of 1 certificate requests certified, commit? [y/n]"

Passo 3 - Generate chiave pre-condivisa (PSK)

L'opzione --tls-auth utilizza una chiave pre-condivisa (PSK) statica che deve essere generata in anticipo e condivisa tra tutti i peer. Questa funzionalità aggiunge una "protezione extra" al canale TLS richiedendo che i pacchetti in entrata abbiano una firma valida generata utilizzando la chiave PSK.

Il vantaggio principale è che un client non autenticato non può causare lo stesso carico CPU / crittografia contro un server poiché il traffico indesiderato può essere eliminato molto prima. Questo può aiutare a mitigare i tentativi di negazione del servizio.

```
# openvpn --genkey --secret keys/ta.key
```

Passo 4 - Generate Diffie Hellman parameters

Diffie Hellman può essere generato per l'OpenVPN server, su Linux/BSD/Unix:

```
# ./build-dh
```

```
easy-rsa # ./build-dh
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
..................+.........................................
...................+............+.................+.........
..............................
```

Passo 5 - Generare certificati & keys for 2 clients

Generare i certificati lato client, su Linux/BSD/Unix:

```
# ./build-key client1
# ./build-key client2
```

se si vuole usare una protezione con password usare il comando script **build-key-pass**

Attenzione, usare per ogni client l'appropriata "Common Name" quando richiesta i.e. "client1", "client2", "client3" il nome deve essere univoco!

Panoramica dei Key Files generati:

keys e certificati in the **keys** subdirectory.

Filename	Richiesto da	Scopo	Segreto
ca.crt	server + all clients	Root CA certificate	NO
ca.key	key signing machine only	Root CA key	YES
dh{n}.pem	server only	Diffie Hellman parameters	NO
server.crt	server only	Server Certificate	NO
server.key	server only	Server Key	YES
client1.crt	client1 only	Client1 Certificate	NO
client1.key	client1 only	Client1 Key	YES

Esempio di nomina con gestine di vpn multiple:

```
ca.crt → ca_miavpn.crt
ca.key → ca_miavpn.key
dh1024.pem
server.crt → server_miavpn.crt
server.key → server_miavpn.key
```

Composizione del certificato per il client

Al client vanno consegnati:
- file configurazione .ovpn
- client1.crt
- client1.key
- ca.crt
- ta.key

Revoca di un certificato

```
# source .vars
# ./revoke-full nomecertificato
```

```
Revoking Certificate 08.
Data Base Updated
Using configuration from /etc/openvpn/rsa/openssl-1.0.0.cnf
prova.crt: C = IT, ST = IT, L = Milano, O = Fort-Funston, OU = changeme, CN =
prova, name = changeme, emailAddress = mail@host.domain
error 23 at 0 depth lookup:certificate revoked
```

Non badare al messaggio "error 23"

Possiamo in ogni momento verificare la situazione dei certificati:

```
# cat /etc/openvpn/rsa/keys/index.txt
```

```
R 241120112146Z 141123112216Z 08 unknown
/C=IT/ST=IT/L=Milano/O=Fort-Funston/OU=changeme/CN=prova/name=changeme/
emailAddress=mail@host.domain
```

Notiamo che il primo flag "*R*" sta per "*Revoked*".

Ora dobbiamo configurare OPENVPN per fare in modo che vengano controllati i certificati alla connessione del client, Per far ciò aggiungiamo/verifichiamo la presenza nel file di configurazione la seguente riga:

```
crl-verify /etc/openvpn/rsa/keys/crl.pem
```

Definizioni dei device

Device VPN:

TAP: device è un virtual ethernet adapter, (ovvero può Trasmette Frames Ethernet)
TUN device è un virtual point-to-point IP link (ovvero può Trasmette pacchetti TCP/IP)

Non e possibile mischiare device tun/tap per la stessa connessione. (in pratica un TUN non si collega a un TAP e viceversa)

Se si intende collegare dispositivi mobili (iOS o Android) utilizzando OpenVPN, è necessario utilizzare TUN poiché attualmente TAP non è nativamente supportato (è possibile usare App esterne, ma con Rooting del device)

Regole pratiche sui device:

TUN
Fa passare meno dati attraverso la connessione internet
se si ha solo bisogno di accedere alle risorse connesse direttamente alla macchina server OpenVPN dall'altra parte, e non ci sono problemi con Windows.

TAP
Fa passare qualunque protocollo
Permette di fare un bridge tra le interfacce
Permette di usare VLAN
se è necessario accedere a più risorse (macchine, storage, stampanti, dispositivi) connessi tramite la rete all'altra estremità. TAP può anche essere richiesto per alcune applicazioni Windows.

Vantaggi/Svantaggi:

TUN
Si perdono tutti i vantaggi del TAP e si può usare solo il TCP/IP

Normalmente limita l'accesso VPN a una singola macchina (indirizzo IP) e quindi (presumibilmente) una migliore sicurezza attraverso una limitata connettività alla rete remota. La connessione TUN creerà meno carico sul tunnel VPN e, a sua volta, sulla rete remota perché solo il traffico verso / dal singolo indirizzo IP attraverserà la VPN dall'altra parte. I percorsi IP verso altre stazioni nella sottorete non sono inclusi, quindi il traffico non viene inviato attraverso il tunnel VPN e sono possibili poche o nessuna comunicazione oltre al server OpenVPN.

TAP
Traffico di rete leggermente superiore rispetto a TUN

In genere consente ai pacchetti di scorrere liberamente tra i punti finali. Ciò offre la flessibilità di comunicazione con altre stazioni sulla rete remota, inclusi alcuni metodi utilizzati dai software Microsoft meno recenti. TAP ha le implicazioni di sicurezza inerenti alla concessione dell'accesso esterno "dietro il firewall". Permetterà a più pacchetti di traffico di fluire attraverso il tunnel VPN. Ciò apre anche la possibilità di conflitti di indirizzi tra gli endpoint.

Esistono differenze di latenza a causa del livello di stack, ma nella maggior parte degli scenari per gli utenti finali la velocità di connessione degli endpoint è probabilmente un contributo più significativo alla latenza rispetto al particolare livello di stack della trasmissione. Se la latenza è in discussione, potrebbe essere una buona idea considerare altre alternative. Gli attuali multiprocessore a livello di GHz superano di solito il collo di bottiglia della trasmissione via Internet.