FRI NET

Navodila za postavitev

Metod Medja

[mm5484@student.uni-lj.si](mailto:mm5484@student.uni-lj.si)

Kazalo vsebine

[**Opis omrežja**](#_9etp0m7jdd50) **3**

[**Sistemske zahteve**](#_es7mccz747zb) **4**

[OpenSSL](#_1wb1c2w96kwp) 4

[OpenVPN](#_38k8ics2g8u5) 4

[Posredovanje IPv4 prometa](#_7qmv1wwft6n8) 5

[**Pisanje poročila**](#_2xfqtyvyl9en) **6**

[Delovanje certifikatne agencije](#_gomx31ie2qro) 6

[Različne vrste certifikatov](#_r8w663tb4kl3) 6

[Ustvarjanje ključev in certifikatov](#_6r31ucacafxs) 6

[Verige ključev](#_b0wig7pynn0c) 8

[Delovanje navideznega omrežja](#_t89oordqjxau) 9

[Usmerjanje po omrežjih](#_zc12w9x8v4eh) 9

[**Certifikatna agencija**](#_80cnb3by4xyo) **11**

[Namestitev certifikatne agencije](#_nacr2xtdifi8) 11

[Generiranje zahtev za podpis](#_odr565982buq) 11

[Podpisovanje zahtev za podpis](#_nfqabvty4nmv) 11

[**VPN Klient**](#_n62u9nprhy4t) **12**

[Konfiguracija klienta](#_qp6c7fqecjyl) 12

[Zagon klienta](#_8ej93u7vtxki) 12

[**VPN Strežnik**](#_2p4h9fqzz87p) **13**

[Konfiguracija strežnik](#_tpnbj5orqnad) 13

[Zagon strežnika](#_mivrvshvx54) 13

[**Pogoste napake in vprašanja**](#_y3lj5c2hhqr1) **14**

# Opis omrežja

Struktura omrežja: <https://fri.medja.net/fri-net.svg>

Omrežje FRI NET je sestavljeno kot drevo. Ideja izhaja iz poenostalvjene strukure spleta. V sredšču se nahaja nekaj močnejših usmerjevalnikov, ki omogočajo komunikacijo med celo kopico različnimi omrežji. Na robovih (listih drevesa) pa so manjša omrežja, do katerih je, kot že omenjeno, omogoče dostop preko jedra omrežja.

Kot je navadno za taka večja omrežja, tudi tukaj, manjša omrežja v listih ne poznajo vseh ostalih. Zanašajo se na to, da bo ves promet namenjen ostalim omrežjem usmerjan do pravih naprav v korenu omrežja.

A ker je bil projekt orientiran okoli virtualnih omrežji to niso fizično povezana omrežja. Vsa so navidezna in za povezavo na njih je potreben nek dokaz identitete. Temu služi ena velika certifikatna agnecija, ki pokriva celotno omrežje.

Ta certifikatna agencija je ponovno sestavlje iz večih vozlišč. Glavno je korensko vozlišče, kateremu zaupajo vsi strežniki in uporabniki. Z dokazovanjem, da je ta zaupal ostlaim, si lahko kdorkoli poveže na katerokoli omrežje. Deluje tako, da korensko vozlišče podpiše certifikate vozlišč, ki so pod njim. S tem potrdi, da jim zaupa. Ta vozlišča lahko potem naprej podpisujejo druge certifikate. In ko nekdo želi preveriti, ali je nek končni certifikat res vreden zaupanja, enostavno preveri kdo ga je podpisal. Če vidi, da je to korenski certifikat, ki mu zaupa, je problem rešen. Če ni, pa v datoteki poišče naslednji certifikat, torej certifikat enega vozlišča višje v drevesu in preveri če je ta res podpisal tega, ki ga preverja. Na ta način se premika skozi celotno verigo, dokler ali ne pride do certifikata, ki mu zaupa, ali pa ugotovi, da ta certifikat ni vredu.

Na ta način se je torej mogoče povezati na katerokoli glavno ali podomrežje celotnega sistema. Ni pa to dovolj. Navadno po spletu komuniciramo tako, da se povezujemo na javne naslove strežnikov, neposredno ali preko domene. To omrežje ponuja še eno možnost. Tukaj namreč naslovi korenskega omrežja pomenijo našemu sistemu to, kar pomnijo javni spletni naslovi spletu. Torej neko “javno” dosegljivo točko. To pomeni, da se je mogoče na neko strežnik povezati tudi preko glavnega dela omrežja, četudi da nima javnega naslova na spletu. Edini predpogoj za to, je da sta in uporabnik, ki se povezuje in strežnik, na katerega se bo povezal, oba povezana na glavni del omrežja. Kjer je glavni del omrežja v točno tem primeru, katerikoli izmed korenskih strežnikov.

# 

# Sistemske zahteve

Uporaba sistema Windows ni priporočena, zlasti zaradi skript, ki so napisane za Bash.

V primeru da ne gre drugače, niti z virtualko, je priporočena uporaba MinGW ali pa prijateljevega računalnika. To velja lasti za delo s certifikati. OpenVPN je mogoče postaviti tudi brez.

## OpenSSL

**Samo za Windows, če se ne uporablja MinGW.**

1. Prenesi OpenSSL (najbolje 1.1.x).
   * <https://slproweb.com/products/Win32OpenSSL.html>
2. Namesti OpenSSL v poljuben direktorij.
3. Dodaj PATH do *bin* mape znotraj OpenSSL namestitve.
   * Odpri:
     1. Control Panel
     2. System and Security
     3. System
     4. Advanced system settings
     5. Environment Variables…
   * Uredi PATH.
   * Izvedi ali reboot ali ponovni vpis ali pa uporabljaj PowerShell.

## OpenVPN

* Windows
  + <https://openvpn.net/index.php/open-source/downloads.html>
  + Prenesi installer, zadnji v tabeli.
* OS X
  + <https://tunnelblick.net/downloads.html>
  + Prenesi in namesti.
* Linux (Debian distribucije):
  + sudo apt-get install openvpn

## 

## 

## Posredovanje IPv4 prometa

To je potrebno za uspešno usmerjanje med omrežji.

Mora biti vklopljeno na računalniku, ne na usmerjevalniku ali pa v nastavitvah VirtualBox-a!

* Windows
  + [http://www.wikihow.com/Enable-IP-Routing](http://www.wikihow.com/Enable-IP-Routing#Another_easier_method_for_Windows_7)
* OS X
  + Omogoči se posredovanje:
    - sudo sysctl net.ipv4.ip\_forward=1
  + Po potrebi se lahko net.ipv4.ip\_forward=1 dopiše oz. popravi tudi v /etc/sysctl.conf, kar bo povzročilo, da bo nastavitev preživela ponovne zagone **(ni preverjeno!)**.
* Linux
  + Začasna omogočitev:
    - sudo sysctl -w net.ipv4.ip\_forward=1
  + Dolgotrajna omogočitev:
    - Popravi se datoteka /etc/sysctl.conf z vrednostjo net.ipv4.ip\_forward=1
    - Izvede se ozvežitev nastavitev: sudo sysctl -p

# 

# Pisanje poročila

Ta navodila so namenjena postavitvi omrežja. Njihova vloga je voditi uporabnika skozi celoten postopek in s tem minimizirati število zapletov, do katerih lahko pride.

Za pisanje poročila je še vedno potrebno poznati delovanje certifikatne agencije in OpenVPN. Za bolj podrobno razumevanje skript in konfiguracijskih datotek je priporočeno branje njihove vsebine. Te vsebujejo komentarje, ki poskušajo opisati, kaj se na določenem delu skripte oz. konfiguracije tudi dogaja.

**V obseg tega omrežja spada le postavitv certifikatnih agencij in VPN omrežja. Vse ostalo morate storiti sami.**

## Delovanje certifikatne agencije

Certifikatna agencija ni nič drugega kot neka oseba ali naprava s certifikatom (in parom ključev). Privatni ključ je vedno shranjen v posebni datoteki medtem ko je javni pogosto shranjen le znotraj certifikata. Vedno pa ga je mogoče obnoviti s pomočjo privatenga ključa. Kar torej loči navadnega uporabnika s certifikatom od certifikatne agencije so lastnosti njegovega certifikata in njegova uporaba.

### Različne vrste certifikatov

Certifikat lahko vsebujejo različne razširitve in defacto standard tu je x509v3. Ta razširitev omogoča navajanje uporabe in omejitev certifikata. In s pomočjo nje je tudi mogoče ločiti med certifikatom neke certifikatne agencije, strežnika ali uporabnika.

Glavna razlika med certifikatno agencijo in ostalimi je, da ta nima onemogočeno podpisovanje certifikatov. Ta omejitev spada pod: X509v3 Basic Constraints. Druga je, da je pod njegov namen uporabe nevedeno podpisovanje drugih certifikatov.

Ogled vseh teh lastnosti certifikata je mogoče s pomočjo ukaza:

openssl x509 -in <pot\_do\_crt> -text -noout

* x509 je OpenSSL program, ki ga poganjamo
* -in poda pot do certifikata, alternativno ga je mogoče tudi preusmeriti s <
* -noout določa, da ne želimo dobiti pod vsebino še certifikata

Če želimo dobiti podoben pregled nad zahtevo za podpis lahko izvedemo isti ukaz, le da namesto x509 uporabimo program req.

### 

### 

### Ustvarjanje ključev in certifikatov

Generiranje parov ključev OpenSSL omogoča ukaz:

openssl genrsa -out <pot\_do\_key> <velikost>

* genrsa je OpenSSL program, ki ga poganjamo
* -out podat pot do privatnega ključa, ki bo ustvarjen; alternativno se lahko uporabi >
* velikost je število bitov ključa, navadno ali 2048 ali 4096

A ta ključ še ni certifikat. Certifikat je dokument, ki povezuje javni ključ z nekimi (npr. osebnimi) podatki. Da pa vemo, da je ta verodostojen, je ta dodatno podpisan s strani certifikatne agencije. Za zaupanje certifikatu je torej treba najprej že prej zaupati certifikatni agenciji. To se navadno naredi tako, da se že prej na končno napravo namesti glavni (root) certifikat agencije. Veliko število teh certifikatov pride vnaprej nameščenih s operacijskim sistemom in brskalniki.

Za ustvarjanje certifikata obstajata dva načina. Prvi je da tega podpišemo sami. Pri tem načinu OpenSSL podamo nastavitve za certifikat in ta ga ustvari tako, da javni ključ podpiše s privatnim. Ukaz:

openssl req -config <pot\_do\_cnf> -new -x509 -key <pot\_do\_key> -days <veljavnost> -extensions <nastavitve> -out <pot\_do\_crt>

* req je OpenSSL program, ki ga poganjamo
* -config poda pot do konfiguracijske datoteke z nastavitvami
* -new določa, naj se ustvari nova zahteva za podpis
* -x509 določa, da se ustvari certifikat in ne zahteva za podpis
* -key poda pot do privatnega ključa
  + Z njim se izdela javni ključ in podpiše certifikat
* -days določi koliko dni bo veljaven ključ
* -extensions določi kateri del nastavitev naj se uporabi za certifikat
* -out določa pot do certifikata, ki bo ustvarjen; lahko se uporabi tudi >

Drugi način generiranja certifikata pa je s pomočjo zahteve za podpis. Ta omogoča, da certifikat podpiše nekdo drug, npr. certifikatna agencija, kateri zaupa več uporabnikov. Pri tem načinu je potrebno najprej ustvariti zahtevo za podpis:

openssl req -config <pot\_do\_cnf> -new -key <pot\_do\_key> -out <pot\_do\_crt>

* req je OpenSSL program, ki ga poganjamo
* -config poda pot do konfiguracijske datoteke z nastavitvami
* -new določa, naj se ustvari nova zahteva za podpis
* -key poda pot do privatnega ključa s katerim se bo izdelal javni ključ
* -out določa pot do zahteve za podpis, ki bo ustvarjena; lahko se uporabi tudi >

Ostane le še podpis certifikate na strani neke certifikatne agencije:

openssl ca -config <pot\_do\_cnf> -extensions <nastavitve> -notext -in <pot\_do\_csr> -out <pot\_do\_crt>

* ca je OpenSSL program, ki ga poganjamo
* -config poda pot do konfiguracijske datoteke z nastavitvami
* -extensions določi kateri del nastavitev naj se uporabi za certifikat
* -notext onemogoči besedilni izpis opisa certifikata
* -out določa pot do certifikata, ki bo ustvarjen; lahko se uporabi tudi >

S tem je certifikat končno ustvarjen. V resnici sta ta dva ukaza samo 2 dela ukaza za izdelavno lastno podpisanega certifikata. Uporaba teh dveh korakov s lastnim zasebnim ključem torej prav tako naredi lastno-podpisan certifikat.

### Verige ključev

Kako certifikatne agencije delujejo je pribljižno razloženo zgoraj. A ta primer vsebuje le eno agencijo, s katero so direktno podpisani uporabniki. Problem nastane, ko je uporabniški certifikat podpisan s strani vmesne certifikatne agencije (torej agencije, katere certifikat je podpisan s strani glavne ali root agencije). Ta certifikat seveda omogoča podpisovanje a predstavlja težave. Ko bo neka tretja oseba želela preveriti verodostojnost z vmesno podpisanim certifikatom, to ne bo delovalo, tudi če pozna glavni certifikat. Ta se namreč zaveda le vmesnega, tega pa nima.

Ta problem se rešuje z verigami (ang. chain). Ta lahko enostavno vsebuje vse certifikate med glavnim in uporabnikovim, včasih pa tudi glavnega (full chain). Preverjanje certifikata keteremu je priložena veriga poteka tako, da se najprej preveri če je certifikat podpisan s prvim iz verige, nato če je ta podpisan z drugim itd.

Verige se torej navadno podaja posebaj. Lahko se ga tudi združi skupaj z glavnim certifikatom in uporabi na mestu, kjer se pričakuje certifikatna agencija. Ali pa, kot je v primeru OpenVPN, združi skupaj s certifikatom uporabnika, saj lahko tako popolnoma vsi uporabljajo enak glavni certifikat.

Realizacija same verige pa je silno preprosta. Gre enostavn za datoteko v kateri so eden za drugim napisani certifikati, urejeni tako, da spodnji vedno podpisuje enega višje.

## Delovanje navideznega omrežja

Navidezno omrežje deluje tako, da simulira omrežje. Bolj natančno:

* TAP: Simulira na nivoju povezavne plasti
* TUN: Simulira na nivoju omrežne plasti

Ko ste povezani na VPN, je kot da bi računalnik s kablom povezali na nek nov usmerjevalnik. Na ta način ste lahko s tem navideznim “usmerjevalnikom” povezani skupaj s ljudmi, ki so lahko na popolnoma drugi strani sveta.

Drugače povedano je VPN: tunel za promet + virtualizacija omrežja

To pomeni, da lahko s tem omrežjem delate karkoli. Npr. nastavljate usmejevalne tabela, ki določajo kam bodo šli paketki za določene naslove. Pri tem je pametno poznati le še eno omejitev. Čeprav TAP deluje na njižji plasti kot TUN, programi kot so OpenVPN, na njih pogosto ne omogočajo tako dobrega nadzora IP prometa kot s TUN adapterji. Ti se pri njih navadno omejijo bolj na podporo povezovanja na povezavni plasti.

### Usmerjanje po omrežjih

Usmerjanje po vrtualnih omrežjih je podobno kot pri pravih omrežjih. Ideja tega je, da vse paketke, ki pridejo na našo napravo na enem vmesniku preusmerimo na drug vmesnik.

* Nekdo, ki je le povezan na neko mrežje mora poskrbeti le, da ves promet za tisto omrežje potuje do omrežnega vmesnika tistega omrežja (OpenVPN klienta).
* Nekdo, ki je le strežnik za neko omrežje, mora poskrbeti, da OpenVPN pozna naslove vseh klientov, tako da lahko vse paketke posreduje pravemu uporabniku.
* Nekdo, ki ima na napravi tako strežnik kot klient in ju želi povezati pa mora poskrbeti, da, ves promet za omrežje, na katerem je odjemalec, teče na vmesnik tistega omrežja (torej iz serverja na klienta). In da ves promet iz klienta potuje tudi proti strežniku (ter da ta ve kje je vsak uporabnik in mu to lahko posreduje).

To razdeli usmejanj na par odgovornosti:

* Usmerjanje proti klientu (ves promet za tisto omrežje, ne glede na izvor - naj bo to Firefox ali pa OpenVPN strežnik) - za to poskrbi strežnik tistega omrežja
* Usmerjanje proti strežniku (ves promet za omrežje, ki ga ta tvori) - za to skrbi OpenVPN samodejno
* Usmerjanje do pravega uporabnika na omrežju, ko paket prispe do strežnika OpenVPN (za to skrbi OpenVPN napol samodejno)
  + Če je promet namenjen nekomu na istemu omrežju, kot je strežnik, je usmerjanje samodejno
  + Če je promet namenjen na neko drugo omrežje, ki del strežnikovega omrežja, je potrebno za klienta, ki nadzoruje to omrežje, ustvariti posebno konfugracijsko datoteko in notri navesti to omrežje. *To se v našem omrežju dela le na korenskih strežnikih.*

Kako zagotoviti, da klienti nekega strežnika posredujejo ves promet za tisto omrežje nazaj proti strežniku:

* Če želimo posredovati samo promet za točno tisto omrežje, na katerem je strežnik, se to ponovno zgodi samodejno.
* Če želimo posredovati več (npr. strežnik z omrežjem 10.1.1.0/24 želi prejeti vse namenjeno za omrežje 10.0.0.0/8) pa je potrebno navesti dodatno push nastavitev. S push nastavitvijo se v tem primeru na vseh odjemalcih nastavi route nastavitev, ta pa ne naredi nič drugega, kot doda vnos v odjemalčevo usmerjavalno tabelo.

Ostane le še ena težava. Kako pa bodo paketki, ki pridejo iz enega vmensika (npr. OpenVPN odjemalec) sploh prišli do drugega vmesnika (npr. OpenVPN strežnik). Ne bo odjemalec enostavno zavrgel te paketke, ker niso namenjeni zanj? Problem se reši z omogočanjem posredovanja IP prometa. To pove operacijskemu sistemu, naj se vsi paketki posredujejo preko usmerjevalne tabele. Ta rešitev deluje, ker v resnici odjemalci navadno ne zavržejo paketkov sami, temveč to za njih naredi OS.

# 

# Certifikatna agencija

**To poglavje (z izjemo zadnjega dela) je namenjeno popolnoma vsem.**

## Namestitev certifikatne agencije

1. Prenesi ca.zip in ga razširi v poljubnem direktoriju, lahko kar na namizju.
   * Brez skrbi, zip vsebuje poddirektorij v katerem so vse datoteke.
2. Poženi ./config.sh in namesti certifikatno agencijo:
   * Za CA: ./config.sh intermediate oz. ./config.sh i
   * Za vse ostal: ./config.sh client oz. ./config.sh c
   * Strežniki za ukaz dopošite še +dh za izdelavo Diffie-Hellman parametrov.
3. Prvi pogon konfiguracije je kar dolg, pojdi na kavo.

## Generiranje zahtev za podpis

Ta korak je ponovno namenjen vsem, tudi certifikatne agencije so le vmesni člen verige, zato te ne izdelajo svojega certifikata temveč zanj zaprosijo glavni strežnik.

1. Premakni se v direktorij s certifikatno agencijo.Poženi ./generate.sh
2. Pri generiranju bo OpenSSL postavil prav prašanj:
   * Prvih par nastavitev ima privzete vrednosti, ki so pravilne.
   * Skupina naj bo enaka kot v razpredelnici, napisana z velikimi črkami.
   * Student id naj bo **vaša vpisna številka** (ne od strežnika!).
3. Posreduj zahtevo za podpis svoji certifikatni agenciji.

Po končanem postopku se bo zahteva za podpis nahajala v certs/cert.csr, zasebni ključ pa v private/privkey.key.

## Podpisovanje zahtev za podpis

Ta korak je namenjen izključno certifikatnim agencijam.

1. Premakni zahtevo za podpis (.csr) v poljuben direktorij.
2. Poženi ./sign.sh za podpis certifikata:
   * Za strežniški certifikat poženi eno izmed:
     1. ./sign.sh server <cert.csr>
     2. ./sign.sh s <cert.csr>
   * Za uporabniški certifikat poženi eno izmed:
     1. ./sign.sh client <cert.csr>
     2. ./sign.sh c <cert.csr>
3. Poleg zahteve za podpis je sedaj certifikat z istim imenom (.crt).
4. Posreduj podpisani in root certifikat nazaj uporabniku.

**Strežnik bo vedno dobil nazaj 2 certifikata (cert.client.crt in cert.server.crt). Potreboval bo oba. client za povezavo na korenski strežnik, server za ustvarjanje strežnika.**

# VPN Klient

**To poglavje je namenjeno vsem.**

Če želite biti povezani na omrežje mora namreč tudi strežnik podomrežja (omrežja skupine) delovati kot klient korenskega omrežja. Povezava na korensko omrežje je priporočena tudi vsem ostalim, a v primeru, da ste le odjemalec in imate možnost direktne povezave do vašega podomrežja (mogoče ima javen naslov), lahko to povezavo izpustite. V primeru, da ste odjemalci povezani le na strežnik vaše ekipe, in je ta postavljen po navodilih, boste kljub temu imeli dostop do celotnega omrežja.

## Konfiguracija klienta

Konfiguracija klienta ni nič posebnga, v resnici je zares pomembnih le par nastavitev. Večina ostalega pa je neobveznega. Paziti je treba le, ker nekatere nastavitve morajo biti enake in in strežniku in na klientu.

Predloga konfiguracije klienta, ki bo delovala za korenski strežnik, se nahaja v client.conf.

V datoteki so podani tudi veliko bolj podrobni opisi nastavitev. Hkrati pa je izpiščenih cel kup nastavitev, ki jih v večina ne bo potrebovala.

Nastavitve, ki jih nujno potrebujete, ko boste gradili svojo konfiguracijo znotraj skupine:

* topology - določi vrste naslovo v konfiguraciji
* client - to je konfiguracija klienta
* proto - kateri protokol se uporablja, TCP ali UDP (isti kot na strežniku)
* dev - kakšne vrste naprava naj se uporabi za vmesnik (tun če želite usmerjanje)
* ca - root certifikat (isti popolnoma povsod)
* cert - uporabnikov certifikat
* key - uporabnikov privatni ključ
* remote - naslov na katerem je dostopen VPN strežnik (**glej tabelo**)

Nastavitve, ki jih nastavite le, če so nastavljene na strežniku:

* cipher - kriptografska funkcija za komunikacijo
* auth - zgoščevalna funkcija za komunikacijo
* comp-lzo - omogoči kompresijo povezave

Več neobveznih nastavitev pa je mogoče najti v predlogi in na [uradni strani OpenVPN](https://openvpn.net/index.php/open-source/documentation/howto.html#client).

## Zagon klienta

Na Windows in OS X shranite vse datoteke v en direktorij in uporabite grafični vmesnik.

Na Linux premaknite vse datoteke v /etc/openvpn, nato poženite program kot storitev:

sudo service openvpn@<ime\_config\_brez\_koncnice> start

Npr.: sudo service openvpn@client start

# VPN Strežnik

**To poglavje je namenjeno le tem, ki bodo postavljali OpenVPN strežnik.**

Za strežnik je priporočeno delo na enem izmed Debian sistemov.

## Konfiguracija strežnik

Konfiguracija strežnika poteka podobno kot konfiguracija klienta. Če bo računalnik na katerem teče strežnik hkrati priklopljen na pravilno korensko omrežje in bo imel omogočeno posredovanje, bo podomrežje, ki ga ustvarja, dostopno povsod. Predloga zanj se nahaja v datoteki server.conf.

Nastavitve, ki jih nujno potrebujete, ko boste gradili svojo konfiguracijo znotraj skupine:

* topology - določi vrste naslovov v konfiguraciji
* proto - kateri protokol se uporablja, TCP ali UDP
* dev - kakšne vrste naprava naj se uporabi za vmesnik (tun če želite usmerjanje)
* ca - root certifikat (isti popolnoma povsod)
* cert - uporabnikov certifikat
* key - uporabnikov privatni ključ
* server - omrežje znotraj katerega bo deloval strežnik (**glej tabelo**)
* client-to-client - omogoča prepoznavanje med klienti

Nastavitve potrebne za usmerjanje oz. posredovanje po omrežjih:

* push(route) - posreduje route nastavitev uporabniku (**tu popravite svoj naslov**)

Določanje statičnih naslovov za kliente:

1. Dodati je treba nastavitev client-config-dir
2. Narediti datoteko poimenovano kot je vpisna številka nekega študenta
3. V to datoteko dodati nastavitev ifconfig-push

Neobvezne a uporabne nastavitve:

* ifconfig-pool-persist - omogoča ohranjevanje IP naslovov klientov
  + To se lahko uporabi namesto statičnih naslovov
* keepalive - preprečuje ohranjevanje prekinjenih povezav
* cipher - kriptografska funkcija za komunikacijo
* auth - zgoščevalna funkcija za komunikacijo
* comp-lzo - omogoči kompresijo povezave

Več neobveznih nastavitev pa je mogoče najti v predlogi in na [uradni strani OpenVPN](https://openvpn.net/index.php/open-source/documentation/howto.html#client).

## Zagon strežnika

Je edentičen zagonu klienta, torej Windows in OS X: vse v isti direktorij in grafični vmesnik.

Na OS X samo ne omogočiti down-root plugina!

Linux pa požene sudo service openvpn@<ime\_config\_brez\_koncnice> start

# Pogoste napake in vprašanja

1. Pri podpisovanju ključev dobim napako: “failed to update database”:
   1. Posodobiti je potrebno ca (z novim ca.zip)
   2. Obvezno je treba izbrisati internal mapo
2. Povezava s korenskim strežnikom ni uspela:
   1. Prenesi in uporabi novo root.crt (zna biti da je bil še stari)
   2. (če si strežnik) Preveri, da uporabljaš cert.client.crt za povezavo
3. Povezava na strežnik skupine ne deluje:
   1. Preveri, da vaš strežnik uporablja cert.server.crt
   2. Preveri, da je in na strežniku in pri uporabnikih uporabljen najnovejši root.crt
4. Kako nadgradim ca (z novim ca.zip):
   1. Izbriši openssl.cnf in openssl.orig.cnf
   2. Izbriši vse .sh skripte
   3. Izbriši mapo internal
   4. Skopiraj novo openssl.orig.cnf in skripte iz ca.zip
   5. Poženi ./config.sh (kot je navedeno zgoraj)
5. Ostali člani skupine se ne morejo povezati na moj strežnik:
   1. Preveri, da ima strežnik tudi klienta, ki je povezan na pravi korenski strežnik
   2. Poskrbi, da se klienti najprej povežejo na korenski strežnik nato na tvojega
   3. Poskrbi, da je povezava klientov na strežnik preko remote IP od strežnikove povezave na korenski strežnik
6. Ne morem se povezati ker imam Windows, dobim obvestilo ki omenja nekaj glede tun, P2P in da nek naslov izgleda kot maska:
   1. V konfiguracijsko datoteko, takoj pod client dodajte topology subnet
   2. Če ste strežnik to dodajte na prvo vrstico
      1. Za njo lahko dodate tudi push “topology subnet”
   3. S tem Windows clientom poveste da je dodeljevanje /30 blokov za vsakega klienta zastarelo in naj se začnejo obnašati kot se zagre in preklopijo na nekaj malo bolj standardnega.
7. Zdi se mi, da usmerjanje ne deluje tako kot se zagre:
   1. Preveri ali na strežniku konfiguracija push “route …” vsebuje na koncu se interni naslov strežnika (ki se konca na .1)
8. Moram spreminjati vmesnike/interfaces:
   1. Ne, to OpenVPN stori samodejno, celo ustvari jih.
9. Generiranje Diffie-Hellman parametrov se izvaja že 5 ur in še vedno ni konec:
   1. Dobra vila jih je naredila za vas, najdete jih v mapi dhparams-skupine.
10. Ping na ostale člane moje skupine ne dela:
    1. Preveri, da je vklopljeno posredovanje IP paketkov (IP forwarding)
    2. Preveri, da je vklopljeno na vseh napravah in sicer na računalnikih (ne router, ne VirtualBox, prav na računalniku - tudi če je virtualen)