Što je VPN poslužitelj i kako ga postaviti

Studentski tim: Dubravko Lukačević
Dominik Marjanović
Tomislav Markovac
Josip Trbuščić

Mentor: izv. prof. dr. sc. Miljenko Mikuc

Sadržaj

1 Uvod

Virtualna privatna mreža (engl. VPN, virtual private network) je tehnologija koja omogućava sigurno povezivanje privatnih mreža preko javne mrežne infrastrukture. VPN je razvijen kako bi se geografski udaljenim korisnicima omogućio siguran pristup privatnoj mreži. [1] Do potrebe za takvom tehnologijom je došlo devedestih godina te se ona u početku razvijala samo za velike organizacije koje su zahtjevale siguran prijenos osjetljivih podataka putem interneta. Kroz godine komercijalizacija interneta je omogućila većini država pristup najvećoj mreži što je drastično povećalo broj potencijalnih žrtava tadašnjih hakera. Nakon brojnih provala u sustave velikih tvrtki svakodnevni korisnici su postali svjesni loše sigurnosti interneta zbog čega raste potražnja tehnologija koje poboljšavaju mrežnu sigurnost.

Zaštita podataka se osigurava šifriranjem i dodavanjem posebnih zaglavlja na postojeći paket kako bi se osigurala njegova autentičnost, integritet i povjerljivost, koji su neki od osnovnih sigurnosnih zahtjeva. Šifriranje se odnosi na postupak pretvaranja izvornog teksta u šifrirani tekst pri čemu se koriste ključevi i prikladni algoritmi (npr. AES, RSA). Obrnuti proces, dešifriranje, provodi se kako bi samo korisnik koji posjeduje odgovarajući ključ mogao čitati izvoran tekst. U kontekstu mrežne sigurnosti šifriranje korstimo za zaštitu zaglavlja i podataka koji se nalaze unutar paketa. [2]

Jedan od najpoznatijih i najsigurnijiih skupova protokola koji se koristi u VPN tehnologijama je sigurni IP (engl. Internet Protocol Security, IPsec). IPsec uključuje protokole mrežnog sloja kako bi se omogućila sigurna razmjena podataka između parova mreža (engl. network-to-network), računala (engl. host-to-host) ili računala i mreža (netowrk-to-host). Neki od korištenih protokola su AH (engl. Authentication Header) kojim se postiže autentičnost paketa i ESP (engl. Encapsulating Security Payload) čija je zadaća da osigura povjerljivost podataka i informacija. Uz IPsec često korišteni skupovi protokola su: OpenVPN, PPTP, SoftEther i WireGuard.

U današnje vrijeme moguće je birati između mnogo pružatelja VPN usluga od kojih su neki besplatni dok su ostali dostupni kroz mjesečne ili godišnje pretplate. Besplatne VPN usluge se možda čine kao dobro rješenje za siguran prijenos podataka, ali pružatelje takvih usluga ništa ne sprječava od prodaje naših podataka ili korištenja istih u vlastitu korist. Još jedna opcija je postavljanje vlastitog VPN poslužitelja što može izgledati kao dugotrajan i naporan posao, ali ovakvo rješenje nam omogućava da sami odlučimo kako želimo zaštititi prijenos vlastitih podataka. U ostatku rada se nalazi pregled, usporedba i upute za instalaciju poznatijih VPN tehnologija na različitim platformama.

2 Windows

2.1 Windows 10 VPN

2.2 SoftEther VPN

3 FreeBSD

3.1 FreeBSD VPN over IPsec

Ovo poglavlje će uključivati postavljanje VPN-a na FreeBSD inačici operacijskog sustava BSD.

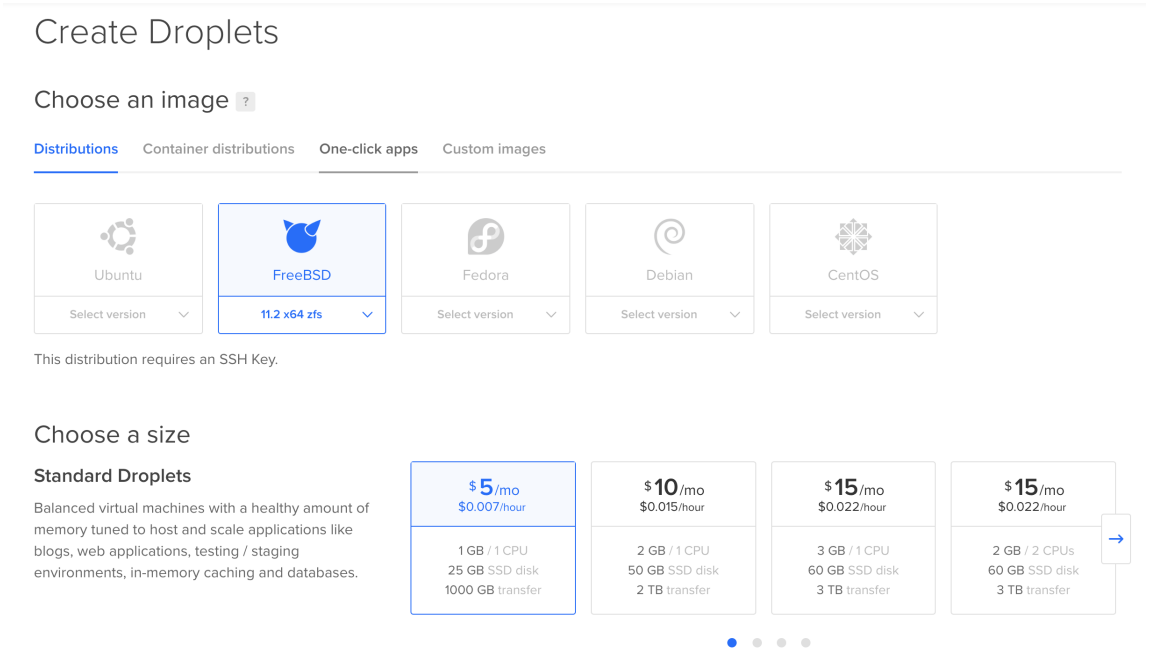
3.2 Postavljanje FreeBSD poslužitelja

Potreban nam je poslužitelj za po mogućnosti sa statičkom IP adresom. Ovdje ćemo koristiti platformu DigitalOcean koja omogućuje brzo i jednostavno podizanje i upravljanje poslužiteljem. Za pristup poslužitelju koristit ćemo ssh protokol za što nam je potreban par ključeva koje generiramo naredbom

```
$ ssh-keygen -t rsa -b 2048
```

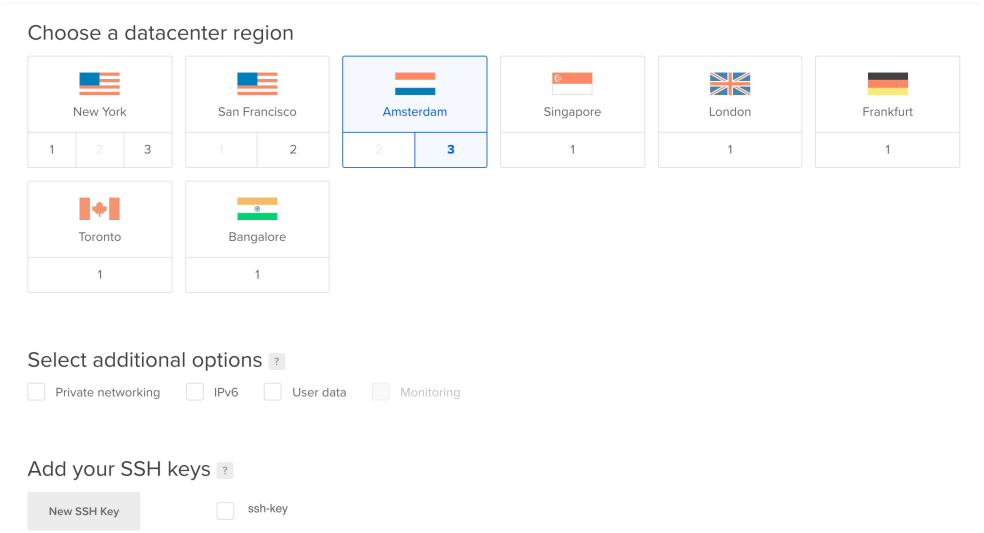
Javni ključ se nalazi u datoteci `/.ssh/id_rsa.pub`, a privatni, koji mora ostati tajan, u `/.ssh/id_rsa`.

Nakon registracije na Digital Ocean na svojem profilu možemo dodati javni ključ koji ćemo kasnije koristiti za pristup poslužitelju. Sada možemo stvoriti poslužitelja. Odabrat ćemo opciju *Create Droplet* i odabrati sljedeće postavke:



Slika 1: Postavke DigitalOcean poslužitelja

Digitlal Ocean nam također nudi opciju da odaberemo lokaciju našeg poslužitelja i pripremimo ssh ključeve kako bi si olakšali pristup



Slika 2: Odabir lokacije i ssh ključeva

Kako sada prvi puta pristupamo poslužitelju jedini korisnik koji postoji je root. Root je korisnik na unixoidima koji može izvršiti svaku naredbu i pristupiti svakoj datoteci. Njemu pristupamo naredbom

```
$ ssh root@139.59.159.111
```

Kada smo se prijavili na poslužitelja prvu stavr koju moramo napraviti je ažurirati sustav. To ćemo napraviti koristeći FreeBSD-ov upravitelj paketima `pkg` i njegove naredbe `update` i `upgrade`.

```
# pkg update
# pkg upgrade
```

Sada možemo instalirati OpenVPN.

```
# pkg install openvpn
```

Konfiguracijske datoteke ćemo smjestiti u direktorij `/usr/local/etc/openvpn` koji prvo moramo stvoriti.

```
# mkdir /usr/local/etc/openvpn
```

Openvpn nudi predloške konfiguracijskih datoteka stoga ćemo kopirati predložak za konfiguraciju poslužitelja u naš direktorij

```
# cp /usr/local/share/examples/openvpn/sample-config-files/server.conf \
    /usr/local/etc/openvpn/openvpn.conf
```

3.2.1 Konfiguracija

Kako bi mogli zaštititi našu vezu potrebno je šifrirati sav promet između poslužitelja i klijenta i osigurati inegritet svake poruke. Za šifriranje podataka ćemo koristiti simetrično šifriranje zbog svoje brzine, a za to nam je potreban simetrični ključ odnsno više ukoliko netko uspije dešifrirati jednu od naših poruka. Kako bi osigurali integritet poruka potrebno ih je potpisati i omogućiti provjeru potpisa. Ovaj problem ćemo rješiti digitalnim certifikatima koje ćemo sami napraviti. OpenVPN dolazi sa alatom Easy-RSA koji će nam poslužiti za izgradnju infrastrukture javnog ključa (*engl. PKI - public key infrastructure*). PKI služi kako bi se javni ključevi povezali s pripadajućim osobama ili oragnizacijama. Proces povezivanja izvršava tijelo za certificiranje (*engl. CA - certification authority.* CA također potvrđuje pripada li javni ključ osobi navedenoj u certifikatu. U praksi se CA nalazi na posebnom računalu, ali kako ovo radimo za privatnu uporabu naš CA će se nalaziti na poslužitelju.

Kako je Easy-RSA omotač oko složene programske knjižnice OpenSSL ona nam je jedini preduvjet te ćemo ju instalirati naredbom

```
# pkg install openssl
```

Nakon toga ćemo kopirati `easy-rsa` direktorij u naš direktorij sa svom konfiguracijom.

```
# cp -r /usr/local/share/easy-rsa /usr/local/etc/openvpn/easy-rsa
```

Sada ćemo se premjestiti u Easy-RSA direktoriji i urediti njegovu konfiguracijusku datoteku `vars`.

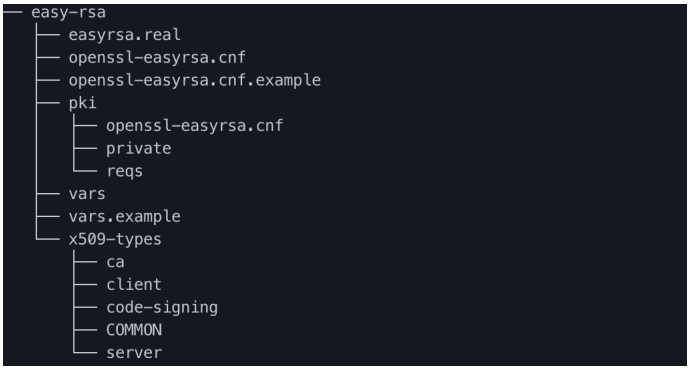
```
# cd /usr/local/etc/openvpn/easy-rsa
# vim vars
```

U nastavku su navedena polja koja je potrebno izmjeniti:

```
set_var EASYRSA_REQ_COUNTRY   "<ZEMLJA>"
set_var EASYRSA_REQ_PROVINCE "<ZUPANIJA>"
set_var EASYRSA_REQ_CITY      "<GRAD>"
set_var EASYRSA_REQ_ORG       "<ORGANIZACIJA>"
set_var EASYRSA_REQ_EMAIL     "<EMAIL>"
set_var EASYRSA_REQ_OU        "<ORGANIZACIJSKA JEDINICA>"
set_var EASYRSA_KEY_SIZE      <broj> # duljina rsa ključa u bitovima
set_var EASYRSA_CA_EXPIRE     <broj> # trajanje CA ključa u danima
set_var EASYRSA_CERT_EXPIRE   <broj> # trajanje certifikata u danima
```

Kako je `easy-rsa` skripta pisana za ljusku sh, dok FreeBSD koristi csh potrebno je naredbom `sh` pokrenuti sh ljsuksu. Sada možemo inicijalizirati PKI

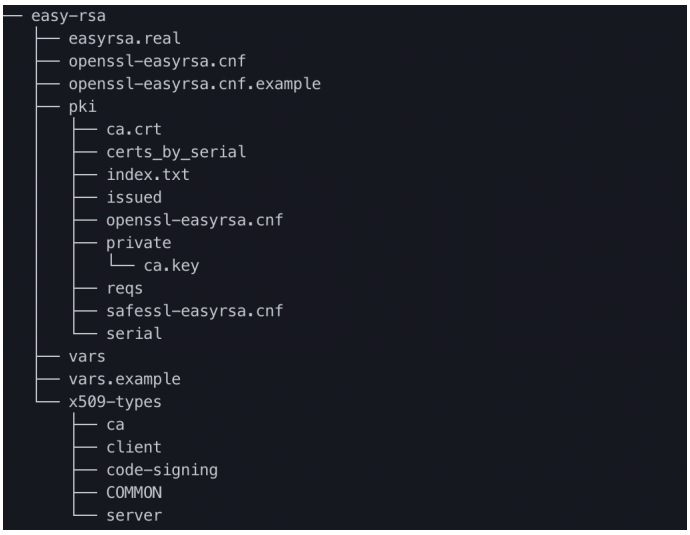
```
# ./easy-rsa.real init-pki
```



Slika 3: Struktura direktorija nakon inicijalizacije PKI

nakon čega ćemo stvoriti CA

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



Slika 4: Struktura direktorija nakon stvaranja korjenskog certifikata

Ovom naredbom smo stvorili par ključeva koji ćemo koristiti za potpisivanje izdanih certifikata.
Sada ćemo generirati serverov certifikat naredbom

```
# ./easy-rsa.real build-server-full <ime-server> nopass
```

gdje je `<ime-server>` ime certifikata, a s `nopass` opcijom ćemo generirati nešifrirani ključ kako bi mogli automatski pokrenuti OpenVPN uslugu prilikom pokretanja sustava bez upisivanja lozinke ključa.

Na sličan način ćemo generirati klijentove certifikate

```
# ./easy-rsa.real build-client-full <ime-klijent> nopass
```

```
# ./easyrsa.real build-client-full openvpn-client

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.0.2o-freebsd  27 Mar 2018
Generating a 2048 bit RSA private key
.....................................+++
.........+++
writing new private key to '/usr/local/etc/openvpn/easy-rsa/pki/private/openvpn-client.ke
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
Using configuration from /usr/local/etc/openvpn/easy-rsa/pki/safessl-easyrsa.cnf
Enter pass phrase for /usr/local/etc/openvpn/easy-rsa/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'openvpn-client'
Certificate is to be certified until Jan 10 20:59:06 2029 GMT (3650 days)

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

Slika 5: Stvaranje certifikata

```
├── easy-rsa
│   ├── easyrsa.real
│   ├── openssl-easyrsa.cnf
│   ├── openssl-easyrsa.cnf.example
│   ├── pki
│   │   ├── ca.crt
│   │   ├── certs_by_serial
│   │   │   ├── 3DD389B5B1D4BA7AAC8BB1978189DFF2.pem
│   │   │   └── 74385B1DA73C31029A193CF94361E839.pem
│   │   ├── dh.pem
│   │   ├── index.txt
│   │   ├── index.txt.attr
│   │   ├── index.txt.attr.old
│   │   ├── index.txt.old
│   │   ├── issued
│   │   │   ├── openvpn-client.crt
│   │   │   └── openvpn-server.crt
│   │   ├── openssl-easyrsa.cnf
│   │   ├── private
│   │   │   ├── ca.key
│   │   │   ├── openvpn-client.key
│   │   │   └── openvpn-server.key
│   │   ├── reqs
│   │   │   ├── openvpn-client.req
│   │   │   └── openvpn-server.req
│   │   ├── safessl-easyrsa.cnf
│   │   ├── serial
│   │   └── serial.old
│   ├── vars
│   ├── vars.example
│   └── x509-types
│       ├── ca
│       ├── client
│       ├── code-signing
│       ├── COMMON
│       └── server
```

Slika 6: Struktura direktorija nakon stvaranja klijentskog i poslužiteljevog certifikata

Za šifriranje same poruke koristit ćemo simetričan ključ generiran Diffie-Hellman razmjenom. Za to su nam potrebni Diffie-Hellman parametri koje stvaramo naredbom

```
# ./easyrsa.real gen-dh
```

Do sada smo sve naredbe izvršavali na poslužitelju te smo generirali velik broj datoteka od kojih ćemo neke morati premjestiti na klijentsko računalo. Kako bi znali koje datoteke

premejstiti potrebno je razumjeti čemu svaka od njih služi. Sve su datoteke stvorene u `easy-rsa/pki/` direktoriju pa ćemo se u njega pozicionirati.

- `ca.crt` - certifikat koji se koristi za validaciju ostalih certifikata, potrebno ga je kopirati na poslužitelja i sve klijente
- `ca.key` - ključ koji CA koristi za izdavanje certifikata
- `reqs/` - direktorij koji sadrži zahtjeve za izdajom certifikata
- `issued/<ime-server>.crt` - certifikat servera koji služi za provjeru potpisa na poruci, potrebno ga je prebaciti na poslužitelja
- `private/<ime-server>.key` - privatni ključ polužitelja koji se koristi za potpisivanje poruke, potrebno ga je prebaciti na poslužitelja
- `issued/<ime-klijent>.crt` - certifikat klijenta koji služi za provjeru potpisa na poruci, potrebno ga je prebaciti na klijentsko računalo
- `private/<ime-klijent>.key` - privatni ključ klijenta koji se koristi za potpisivanje poruke, potrebno ga je prebaciti na klijentsko računalo
- `dh.pem` - Diffie Hellman parametri, potrebno ih je prebaciti na poslužitelja

Ključeve poslužitelja ćemo premjestiti u poseban direktorij

```
# mkdir /usr/local/etc/openvpn/keys
# cp pki/dh.pem \
     pki/ca.crt \
     pki/issued/<ime-server>.crt \
     pki/private/<ime-server>.key \
     /usr/local/etc/openvpn/keys
```

```
keys
├── ca.crt
├── dh.pem
├── openvpn-server.crt
└── openvpn-server.key
```

Slika 7: Sadržaj `keys` direktorija na poslužitelju

Prije nego što počnemo konfigurirati klijentsko računalo, potrebno je u konfiguraciji poslužitelja navesti putanje do certifikata, ključeva i parametara. To ćemo napraviti u datoteci `openvpn.conf` koju smo na samom početku kopirali u `/usr/local/etc/openvpn`.

```
# vim /usr/local/etc/openvpn/openvpn.conf
```

Potrebno je urediti sljedeće linije

```
ca /usr/local/etc/openvpn/keys/ca.crt
cert /usr/local/etc/openvpn/keys/<ime-server>.crt
key /usr/local/etc/openvpn/keys/<ime-server>.key
dh /usr/local/etc/openvpn/keys/dh.pem
```

```
# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key).  Each client
# and the server must have their own cert and
# key file.  The server and all clients will
# use the same ca file.
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys.  Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca /usr/local/etc/openvpn/keys/ca.crt
cert /usr/local/etc/openvpn/keys/openvpn-server.crt
key /usr/local/etc/openvpn/keys/openvpn-server.key  # This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh2048.pem 2048
dh /usr/local/etc/openvpn/keys/dh.pem
```

Slika 8: Putanje do ključeva i certifikata u datoteci `openvpn.conf`

Sada možemo postaviti klijentsko računalo. Kako smo već pripremili većinu klijentovih datoteka na poslužitelju, potrebno ih je kopirati. Radi se o osjetljivim datotekama stoga nam je potreban siguran način slanja datoteka preko mreže za što ćemo koristiti program *secure copy*. Nakon što instaliramo openvpn isto kao i na poslužiteljskom računalu možemo kopirati predložak konfiguracije

```
# cp /usr/local/share/examles/openvpn/sample-config-files/client.config \
     /usr/local/etc/openvpn/openvpn.conf
```

Također stvorit ćemo direktorij u koji ćemo spremiti ključeve i certifikate

```
# mkdir /usr/local/etc/openvpn/keys
```

Sada možemo kopirati potrebne datoteke sa poslužitelja

```
$ scp root@<ip-server>:/usr/local/etc/openvpn/easy-rsa/pki/ca.crt keys
$ scp root@<ip-server>:\
      /usr/local/etc/openvpn/easy-rsa/pki/issued/<ime-klijent>.crt \
      keys
```

```
$ scp root@<ip-server>:\
      /usr/local/etc/openvpn/easy-rsa/pki/private/<ime-klijent>.key \
      keys
```

```
-- openvpn
   |-- keys
   |   |-- ca.crt
   |   |-- openvpn-client.crt
   |   `-- openvpn-client.key
   `-- openvpn.conf
```

Slika 9: Sadržaj `keys` direktorija na klijentu

U konfiguraciji (`openvpn.conf`) osim putanja do certifikata i ključeva potrebno je unjeti ip adresu poslužitelja

```
remote <ip-server> 1194
ca /usr/local/etc/openvpn/keys/ca.crt
cert /usr/local/etc/openvpn/keys/<ime-server>.crt
key /usr/local/etc/openvpn/keys/<ime-server>.key
```

Za upravljanje servisima koristimo naredbu `service`. Prvo ćemo njome pokrenuti OpenVPN servis na poslužitelju i nakon toga na klijentu

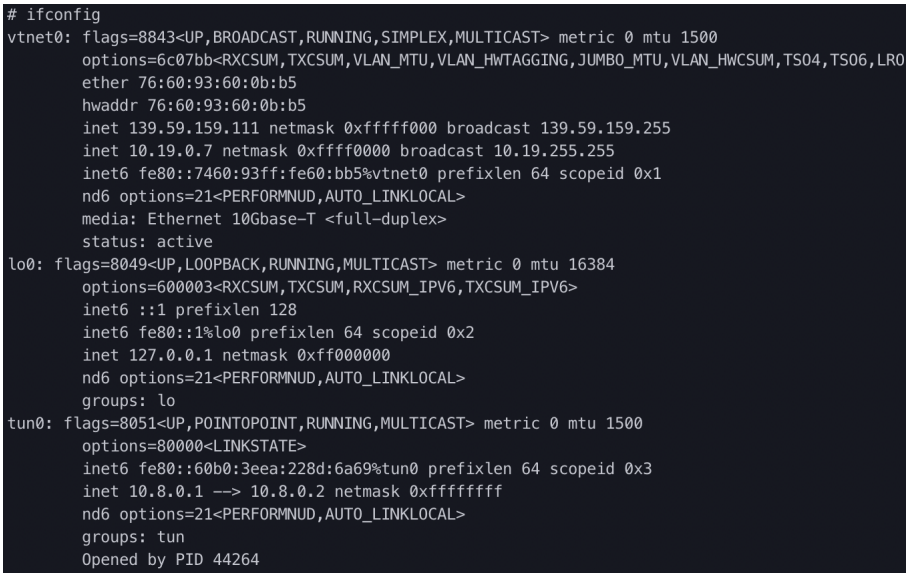
```
# service openvpn run
```

Sada možemo alatom `ifconfig` provjeriti stanje mrežnih sučelja
na klijentu:

```
root@:~ # ifconfig
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
        ether 08:00:27:17:df:37
        hwaddr 08:00:27:17:df:37
        inet6 fe80::a00:27ff:fe17:df37%em0 prefixlen 64 scopeid 0x1
        inet 10.0.2.15 netmask 0xffffff00 broadcast 10.0.2.255
        nd6 options=23<PERFORMNUD,ACCEPT_RTADV,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=600003<RXCSUM,TXCSUM,RXCSUM_IPV6,TXCSUM_IPV6>
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x2
        inet 127.0.0.1 netmask 0xff000000
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
        groups: lo
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> metric 0 mtu 1500
        options=80000<LINKSTATE>
        inet6 fe80::a00:27ff:fe17:df37%tun0 prefixlen 64 scopeid 0x3
        inet 10.8.0.10 --> 10.8.0.9 netmask 0xffffffff
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
        groups: tun
        Opened by PID 615
```

Slika 10: Mrežna sučelja klijenta

na poslužitelju:

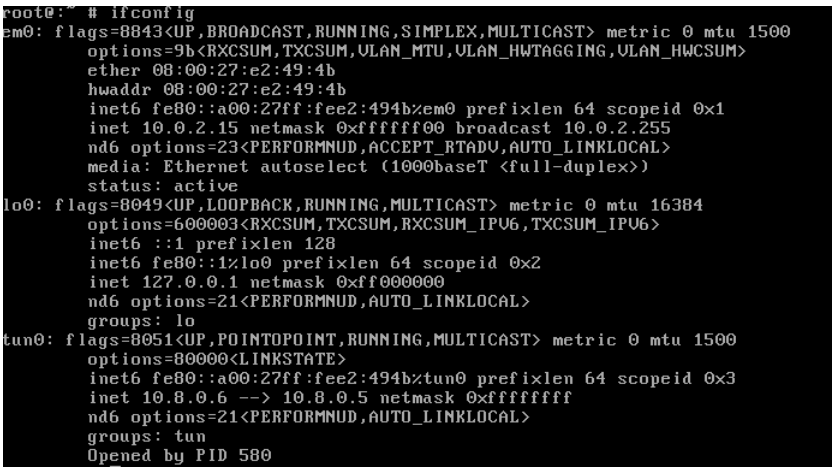


Slika 11: Mrežna sučelja poslužitelja

Možemo uočiti novo sučelje `tun0` koje predstavlja virtualno sučelje mrežnog sloja. Izvršavanjem naredbe `ping` možemo provjeriti je li klijentsko računalo stvarno povezano s poslužiteljem.

```
$ ping 10.8.0.1
```

Kako nam je cilj povezati dva klijenta koji se nalaze u različitim privatnim mrežama na isti ćemo pripremiti još jednog klijenta. Njegova konfiguracija će biti jednaka konfiguraciji prvog klijenta, a izlaz naredbe `ifconfig`:

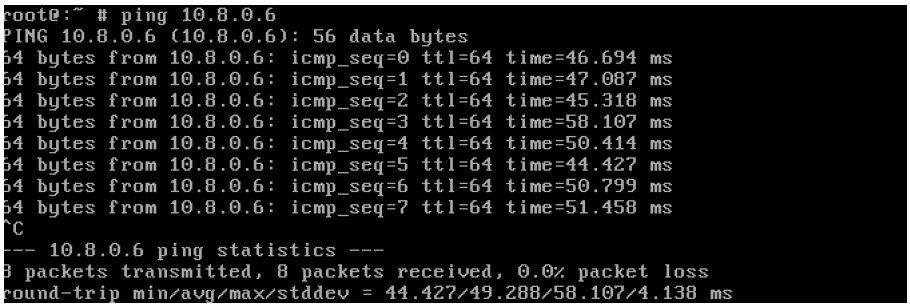


Slika 12: Mrežna sučelja drugog klijenta

Ako sada pokušamo naredbom `$ ping 10.8.0.6` provjeriti jesu li klijenti međusobno povezani nećemo dobiti nikakav rezultat. Razlog tome je što pretpostavljena konfiguracija poslužitelja ne dozvoljava komunikaciju između klijenata. Kako bi to omogućili potrebno je u konfiguracijskoj datoteci poslužitalja otkomentirati liniju
`client-to-client`

Sada ćemo ponovo pokrenuti OpenVPN i ispitati jesu li klijenti međusobno povezani

```
# service openvpn restart
$ ping 10.8.0.6
```



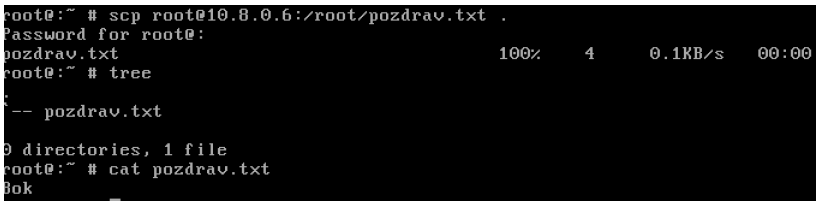
Slika 13: Izlaz naredbe ping

Za kraj možemo pokušati kopirati datoteku s jednog klijenta na drugi koristeći `tun0` sučelja. Na klijentu s adresom `10.8.0.6` stvorit ćemo datoteku `pozdrav.txt` i u nju nešto zapisati

```
$ touch pozdrav.txt
$ echo "Bok" > pozdrav.txt
```

Sada ćemo sa drugog klijenta (adresa `10.8.0.10`) kopirati `pozdrav.txt` datoteku i ispistai ju

```
$ scp root@10.8.0.6:/root/pozdrav.txt .
$ cat /root/pozdrav.txt
```



Slika 14: Kopiranje i ispis datoteke `pozdrav.txt`

4 Linux

4.1 OpenVPN za Ubuntu distribuciju Linux operacijskog sustava

Literatura

[1] CARNet CERT. Osnovni koncepti vpn tehnologije, 2003. https://www.cis.hr/www.edicija/LinkedDocuments/CCERT-PUBDOC-2003-02-05.pdf.

[2] James Henry Carmouche. *IPsec Virtual Private Network Fundumentals*. Cisco Press, 2006.

A Dodatak A: Indeks (slika, tablica, ispisa koda)