digitális jólét
nonprofit kft.

Az OpenVPN privát hálózat kiszolgáló szerver telepítése és konfigurálása

Tartalomjegyzék

1 Bevezetés

Az OpenVPN nyílt forráskódú szoftver egy TLS/SSL (Transport Layer Security, vagy Secure Socket Layer) titkosítási csatornát használó, virtuális privát hálózat létrehozására alkalmas eszköz, melyet minden vállalkozás megvalósíthat saját infrastruktúráján belül és ezáltal biztonságos csatornát biztosíthat a távmunkában dolgozó munkavállalók számára. A VPN hálózat előnyeit kombinálva https alapú webes szolgáltatásokkal (mint például fájlszerver, webszerver, projektmenedzsment rendszer, webes email szolgáltatás, stb.) biztosítható a biztonságos rendszerhasználat vezeték nélküli csatlakozáskor is.

Ebben a dokumentumban bemutatjuk, hogy mely lépésekkel lehet létrehozni Ubuntu 18.04 szerveren az OpenVPN szolgáltatást.[1] Az eljárást a Digitális Jólét Program szakértői állították össze és tesztelték egy saját kiszolgáló tesztszerver létrehozásával. Egyéb Debian alapú Linux szervereken a telepítés azonos lépésekkel megvalósítható, azonban a kiadott rendszerutasítások eltérhetnek az itt tárgyalt szintaxisoktól. Windows szerverre is telepíthető a szolgáltatás, ennek menetét több angol nyelvű útmutató is taglalja, példaként említhetjük az OpenVPN hivatalos telepítési útmutatóját is, mely itt érhető el.

A telepített OpenVPN szerver elérhető tetszőleges platformról (ún. kliens eszközökről), melyek estében a kapcsolódási eljárást külön dokumentumban tárgyalunk.

2 Első lépések, előfeltételek

2.1 Felhasználói jogosultságok, és tűzfal

Az OpenVPN szerver telepítéséhez szükség van **rendszergazdai** hozzáférésre a konfigurálandó szerveren (például `sudo` jogosultságok). A szükséges jogosultságok biztosításáról egy frissen telepített Ubuntu 18.04 szerveren ez az útmutató ad tájékoztatást. Az útmutató tárgyalja a **tűzfal** kialakításának módját is. (A továbbiakban feltesszük, hogy a szerveren létrehozott tűzfal mindvégig aktív.)

2.2 SSL tanúsítványok hitelesítése

A VPN hálózatba való belépésekkor a kliensek (azaz a hálózatba bekapcsolódni kívánó eszközök) SSL alapú tanúsítványokkal hitelesítik magukat a VPN hálózatot létrehozó szerverrel. Ezáltal a kliens gépek a közösségi hálózatról biztonságosan tudnak belépni a vállalti VPN hálózatba a kiszolgáló szerveren keresztül mely elérhető a nyilvános Internet hálózaton. Az SSL hitelesítéshez egy különálló számítógép szükséges (ami nem szerver gép), ahol az SSL tanúsítványok, illetve azok aláírása történik. Az SSL tanúsítványok tárolásának és aláírásának elkülönítése biztonsági okok miatt ajánlott. A feladatot ellátó számítógépnek (továbbiakban: CA szerver, mely Certificate Authority angol kifejezésre utal) csupán a tanúsítványok létrehozása közben és aláírásakor kell aktívnak, azaz működőnek lennie. **Amennyiben éppen nem hozunk létre új hozzáféréseket felhasználók számára, a CA szerver kikapcsolt állapotban lehet.** (Biztonsági okokból a CA szerver kikapcsolása ajánlott is, hogy illetéktelen személyek ne tudjanak semmilyen módon létrehozni újabb hozzáférési tanúsítványokat. Ebből kifolyólag a feladat ellátható egy erre a feladatra dedikált virtuális géppel is.) A továbbiakban feltesszük, hogy a CA szerverhez is rendszergazdai jogosultságokkal rendelkezünk, valamint a gép aktív tűzfallal van ellátva, ahogy azt a 2.1 fejezetben tárgyaltuk.

1 A telepítési útmutató forrása: https://www.digitalocean.com/community/tutorials/how-to-set-up-an-openvpn-server-on-ubuntu-18-04

2.3 A VPN szerver és a CA szerver közti kommunikációja (OPCIONÁLIS)

Bár a VPN szerver telepítése során nem feltétel, azonban jelentősen megkönnyíthetjük a telepítés során a VPN szerver és CA szerver közti kommunikációt és fájlmásolást, hogy ne kelljen minden egyes ilyen lépés során beírni bejelentkezési jelszavainkat. A kommunikációt SSH kapcsolattal javasoljuk megvalósítani. Ehhez létrehozunk egy-egy SSH kulcspárt a VPN és CA szerverek számára. (Az SSH kulcspárok fogalma nem azonos a fentiekben említett SSL tanúsítvánnyal. Ez előbbi csupán a géppárok SSH összekapcsolódását segíti elő, míg a SSL tanúsítvány a VPN hálózatra való bejelentkezéshez és a VPN hálózaton belüli adatforgalom titkosításához szükséges.) Az SSH kulcspár létrehozásához adjuk ki például a VPN szerveren az

```
$ ssh-keygen
```

parancsot, mely alapértelmezetten létrehoz egy 2048 bites RSA kulcs párt. A kulcsgenerálás során megadható opcionálisan egy jelszó is, mely feloldja minden bejelentkezéskor a kulcsok párosítását.

```
Enter passphrase (empty for no passphrase):
```

Mivel pont a jelszavak ismételt beírását szeretnénk elkerülni, így javasolt, hogy ne adjunk meg jelszót, csupán nyomjunk egy `enter`-t. (Mivel a CA szerver általában kikapcsolt állapotban lesz, így ez nem jelent biztonsági kockázatot.)

A generált kulcspár közül a publikusat át kell másolni arra szerverre, melyre be szeretnénk lépni SSH-val. Ez megtehető manuálisan (azaz a ~/.ssh könyvtárban létrehozott *.pub fájl tartalmának folytatólagos átmásolásával a bejelentkezni kívánt szerveren lévő ~/.ssh/authorized_keys fájlba), vagy az ssh-copy-id parancssoros applikáció felhasználásával. Ehhez az alábbi parancsot kell futtatni azon a gépen, melyen létrehoztuk az SSH kulcsokat:

```
$ ssh-copy-id username@remote_host
```

A `username` és `remote_host` paraméterek megegyeznek a VPN szerverről a CA szerverre történő SSH bejelentkezéshez használt paraméterekkel. További tájékoztatás az SSH kulcsokkal kapcsolatban itt érhető el.

Az eljárást meg kell ísmételni fordított irányban is, azaz létrehozzuk az SSH kulcspárt a CA szerverenen és átmásoljuk annak publikus részét a VPN szerverre. Így biztosítható, hogy a két gép között mindkét irányban megvalósítható a jelszómentes, ám hitelesített adatátvitel.

3 Az OpenVPN szoftver letöltése és telepítése

A soron következő lépéseket a VPN szerveren végezzük el. Először ajánlott az elérhető Linux csomagok adatbázisának frissítése:

```
$ sudo apt update
```

A frissítés után az OpenVPN szoftver az alábbi paranccsal telepíthető:

```
$ sudo apt install openvpn
```

4 Az SSL tanúsítványok létrehozása és hitelesítése

4.1 A tanúsítványokat létrehozó applikáció telepítése

Az OpenVPN által létrehozott TLS/SSL csatornák titkosításához szükséges a kapcsolódó tanúsítványok létrehozása, hogy biztosítsuk a VPN szerver és a hozzá csatlakozó kliens felhasználók közti kapcsolat biztonságát. A tanúsítványok létrehozására a DJP szakértői az ún. EasyRSA applikáció letöltését és telepítését ajánlják. Ennek segítségével a különálló CA szerverünkön létrehozhatjuk a hitelesített felhasználók tanúsítványainak tárolására és kezelésére alkalmas PKI (public key infrastructure) rendszert.

Ahogy korábban említettük a tanúsítványok kezelésére egy különálló CA szerver alkalmazása ajánlott. Ha CA és VPN szerver ugyanazon gépen kerülne kiépítésre, akkor egy esetleges betörés esetében elérhetővé válna a CA privát kulcsa is, mellyel további hitelesített tanúsítványok lennének korlátlanul generálhatóak új VPN belépők számára.

A PKI rendszer CA szerveren történő kiépítéséhez első lépésben letöltjük az EasyRSA nyílt forráskódú szoftvert (az alábbi két sort a CA szerveren egy sorban kell kiadni):

```
$ wget -P ~/ https://github.com/OpenVPN/easy-rsa/
releases/download/v3.0.4/EasyRSA-3.0.4.tgz
```

A csomag letöltése után kicsomagoljuk azt:

```
$ cd ~
```

```
$ tar xvf EasyRSA-3.0.4.tgz
```

Az EasyRSA csomag nem igényel további telepítést, mivel csupán bash szkripteket és konfigurációs fájlokat tartalmaz. Végül az EasyRSA szoftvert az előző 3 lépéssel a VPN szerveren is telepítjük.

4.2 A CA szerver konfigurálása

A CA szerveren nyissuk meg a kicsomagolt EasyRSA könyvtárat:

```
$ cd ~/EasyRSA-3.0.4/
```

A könyvtárban készítsük el a `vars.example` konfigurációs fájl másolatát:

```
$ cp vars.example vars
```

Nyissuk meg ezt a fájlt kedvenc szerkesztőnkkel:

```
$ nano vars
```

Majd pedig keressük meg az alábbi sorokat:

```
#set_var EASYRSA_REQ_COUNTRY    "US"
```

```
#set_var EASYRSA_REQ_PROVINCE   "California"
```

```
#set_var EASYRSA_REQ_CITY       "San Francisco"
```

```
#set_var EASYRSA_REQ_ORG        "Copyleft Certificate Co"
#set_var EASYRSA_REQ_EMAIL      "me@example.net"
#set_var EASYRSA_REQ_OU         "My Organizational Unit"
```

A sorok elejéről töröljük a # karaktert és adjuk meg a paraméterek értékeit. (Fontos, hogy ne hagyjuk üresen a paraméterek értékeit) Például az új beállítások:

```
set_var EASYRSA_REQ_COUNTRY    "HU"
set_var EASYRSA_REQ_PROVINCE   "Pest Megye"
set_var EASYRSA_REQ_CITY       "Budapest"
set_var EASYRSA_REQ_ORG        "Cégnév"
set_var EASYRSA_REQ_EMAIL      "admin@example.com"
set_var EASYRSA_REQ_OU         "Community"
```

A paraméterek megváltoztatása után zárjuk be a fájlt. Az EasyRSA könytárában található egy `easyrsa` szkript, melynek futtatásával különböző CA-hoz köthető feladatokat hajthatunk végre automatizált módon. Például az init-pki opcióval létrehozhatjuk a PKI rendszert a CA szerveren:

```
$ ./easyrsa init-pki
```

Ezután újra meghívjuk az `easyrsa` szkriptet, ezúttal a build-ca opcióval, mely a CA SSL tanúsítványának publikus és privát részeit hozza létre. (Ez a tanúsítvány szükséges majd a kliensek SSL tanúsítványainak hitelesítéséhez):

```
$ ./easyrsa build-ca nopass
```

A nopass paraméter megadásával olyan tanúsítványok keletkeznek, melyek feloldásához nem lesz szükség jelszóra. (Mivel a CA szerver általában kikapcsolt állapotban lesz, ezért ez nem jelent biztonsági kockázatot) A szkript futtatásával (`entert` nyomva minden alkalommal az alapértelmezett konfigurációk megtartásához) két fájl keletkezik:

- `ca.crt:` a CA szerverünk tanúsítványának publikus része. Ezt a fájlt használja a VPN szerver a kliensekkel való kommunikáció során annak alátámasztása érdekében, hogy a kliens és a szerver egy megbízható hálózaton belül kommunikálnak egymással.
- `ca.key`: a CA szerverünk tanúsítványának privát része. Ezt használja a CA szerver a VPN hálózatba belépő kliensek SSH tanúsítványainak aláírására (avagy hitelesítésére). Ez a fájl szükséges ahhoz, hogy érvényes SSH tanúsítványok legyenek generálhatóak a kliens felhasználók számára, ezért ajánlott ezen fájlt kizárólagosan a CA szerveren tárolni.

4.3 A VPN Szerver SSL tanúsítványának létrehozása és hitelesítése

A CA szerver felkonfigurálása után létrehozzuk a VPN szerverünk SSL tanúsítványát és hitelesítjük azt a CA szerverrel. (Az SSL tanúsítvány a VPN szerver és a kliensek közti titkosított

kommunikációhoz szükségesek.) Ehhez megnyitjuk az EasyRSA csomag könyvtárát a **VPN szerveren** (az EasyRSA csomagot ugyanúgy töltjük le, mint a CA szerver esetében):

```
$ cd EasyRSA-3.0.4/
```

A könyvtárban meghívjuk az `easyrsa` szkriptet az `init-pki` opcióval. (A VPN szerveren és a CA szerveren is szükségünk van egy-egy tanúsítványokat kezelő rendszerre)

```
$ ./easyrsa init-pki
```

Ezután újra meghívjuk az `easyrsa` szkriptet, most a `gen-req` opcióval. Az `gen-req` opciót a VPN szerver általunk választott neve követi (például `VPNserver`), illetve a `nopass` opció. Ha kihagyjuk a `nopass` opciót, később fájlok szerkesztésével kapcsolatos jogosultsági problémákba ütközhetünk. Felhívjuk a figyelmet, hogy ha a szerver elnevezésére a `VPNserver`-től különböző elnevezést használunk, azt konzisztensen kell majd alkalmazni a soron következő összes parancsban, tehát a `VPNserver` elnevezés helyett mindenhol az általunk használt elnevezést kell használni.

```
$ ./easyrsa gen-req VPNserver nopass
```

A szkript meghívásával létrejön a VPN szerver „kérvénye” az SSL tanúsítvány létrehozásához, valamit az SSH tanúsítvány privát része. A tanúsítvány privát részét (`VPNserver.key`) másoljuk a VPN szerver OpenVPN könyvtárába:

```
$ sudo cp ~/EasyRSA-3.0.4/pki/private/VPNserver.key /etc/openvpn/
```

Illetve másoljuk át a CA szerverre a tanúsítvány „jóváhagyási kérelmét”, azaz a generált VPNserver.req fájlt:

```
$ scp ~/EasyRSA-3.0.4/pki/reqs/server.req username@your_CA_ip:/tmp
```

A parancsban a `username` és a `your_CA_ip` helyére helyettesítsük a CA szerverre való SSH bejelentkezéshez szükséges felhasználó és domain értékeket.

A fájl átmásolása után jelentkezzünk be a CA szerverre, hogy hitelesíthessük a VPN szerver tanúsítványát. A CA szerveren navigáljunk az EasyRSA csomag könyvtárába:

```
$ cd EasyRSA-3.0.4/
```

Az `easyrsa` szkript felhasználásával importáljuk a VPN szerver tanúsítványának adatait a hitelesítéshez:

```
$ ./easyrsa import-req /tmp/VPNserver.req VPNserver
```

Az importált SSL tanúsítványt ugyancsak az `easyrsa` szkripttel hitelesíthetjük. (A hitelesítést `server` és `client` megkülönböztetett opciókkal kell elvégezni a központi VPN szerver és a hozzá csatlakozó kliensek esetében):

```
$ ./easyrsa sign-req server VPNserver
```

A szkript futása közben fel leszünk szólítva a hitelesítés jóváhagyására. A kérdésre a `yes` szó begépelésével és az `enter` lenyomásával válaszoljunk. Ezután másoljuk át a tanúsítvány publikus részének hitelesített változatát a VPN szerverre (titkosított csatornán):

```
$ scp pki/issued/VPNserver.crt username@your_VPNserver_ip:/tmp
```

Ebben az esetben is helyettesítsük a `username` és a `your_VPNserver_ip` változókat a VPN szerverre SSH-val történő bejelentkezéshez szükséges adatokkal. Mielőtt kijelentkeznénk a CA szerverről, másoljuk át a VPN szerverre a CA szerver SSL tanúsítványának publikus részét is:

```
$ scp pki/ca.crt username@your_VPNserver_ip:/tmp
```

Ezután jelentkezzünk be a VPN szerverre és másoljuk át a `VPNserver.crt` és `ca.crt` fájlokat az OpenVPN szoftver `/etc/openvpn/` könyvtárába:

```
$ sudo cp /tmp/{VPNserver.crt,ca.crt} /etc/openvpn/
```

Ezután navigáljunk át az EasyRSA csomag könyvtárába:

```
$ cd EasyRSA-3.0.4/
```

és hozzunk létre egy erős Diffie-Hellman kulcsot, melyet a kulcsok kicserélése során alkalmaz majd a VPN szerver.

```
$ ./easyrsa gen-dh
```

A szkript ezúttal számításigényes feladatot végez, ezért várni kell egy-két percet míg elkészül a kulcs. Miután elkészült a kulcs, generáljunk egy HMAC aláírást, hogy megerősítsük a VPN szerver TLS kommunikáció-integritásának ellenőrzését.

```
$ openvpn --genkey --secret ta.key
```

Az elkészült két fájlt másoljuk az OpenVPN szoftver `/etc/openvpn/` mappájába:

```
$ sudo cp ~/EasyRSA-3.0.4/ta.key /etc/openvpn/
```

```
$ sudo cp ~/EasyRSA-3.0.4/pki/dh.pem /etc/openvpn/
```

Ezzel minden szükséges tanúsítvány és kulcs fájlok a megfelelő helyre kerültek. Ezen fájlok szükségesek lesznek ahhoz, hogy a VPN hálózatra bejelentkező kliensek számára is SSL tanúsítványokat hozzunk létre.

5 Az OpenVPN szerver konfigurálása

A tanúsítványok elkészítése után rátérhetünk az OpenVPN szerver konfigurálására. Ehhez szövegszerkesztők segítségével módosításokat hajtunk végre az OpenVPN szoftver konfigurációs fájljaiban. Ezért először bemásoljuk az alapértelmezett konfigurációs fájlokat az OpenVPN szoftver könyvtárába. Az alapértelmezett konfigurációs fájlok egy tömörített állományban vannak tárolva. Ezt bemásoljuk a `/etc/openvpn` könyvtárba és kitömörítjük.:

```
$ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz
/etc/openvpn/
```

```
$ sudo gzip -d /etc/openvpn/server.conf.gz
```

Végül megnyitjuk az alapértelmezett konfigurációs fájlok egyikét:

```
$ sudo nano /etc/openvpn/server.conf
```

A megnyitott fájlban keressünk rá a HMAC aláírásra vonatkozó részt. Amennyiben a `tls-auth` direktíva előtt pontosvessző található, távolítsuk azt el:

```
tls-auth ta.key 0 # This file is secret
```

Ezután keressük meg a titkosítási rejtjelezéssel foglalkozó részt a fájlban. Az AES-256-CBC rejtjelezés megfelelő biztonságot biztosít, így távolítsuk el a pontosvesszőt a

```
cipher AES-256-CBC
```

elejéről. Ez alá a sor alá adjuk ki az `auth` direktíva segítségével, hogy a HMAC aláírás által használt algoritmust:

```
auth SHA256
```

Ezután keressük meg a fájlban a `dh` direktívát, mely a Diffie-Hellman paramétereket szabja meg. A direktíva után meg kell adnunk annak a fájlnak a nevét, melyben létrehoztuk a Diffie-Hellman kulcsot. Ebben az útmutatásban ez a `dh.pem` fájl volt, így a kérdéses sor a fájlban

```
dh dh.pem
```

Végül keressünk rá a `user` és `group` direktívákra is a fájlban és távolítsuk el a sorok elejéről a pontosvesszőt:

```
user nobody
```

```
group nogroup
```

Az eddigiekben elvégzett változtatások a `server.conf` fájlban szükségesek voltak a szerver működéséhez. A soron következő beállítások azonban opcionálisak annak függvényében, hogy milyen viselkedést várunk el a VPN szervertől.

5.1 A kliensek összes adatforgalmának VPN-re terelése (OPCIONÁLIS)

Amennyiben végrehajtjuk az itt taglalt konfigurációs lépéseket, a VPN-re csatlakozó kliensek minden adatforgalma a VPN-en keresztül lesz terelve. Ez azt jelenti, hogy a kliensek annyit fognak látni a külső közösségi hálózatokból, amennyit a VPN-be kapcsolódó gépek. Függetlenül attól, hogy földrajzilag hol kapcsolódik rá a kliens a VPN hálózatra, a hálózatának a működése olyan lesz, mintha fizikailag is egy zárt hálózatra kapcsolódna rá. (Azaz a kliens nem fog tudni internetet böngészni, vagy egyéb hálózati adatátvitel létesíteni, amennyiben a VPN szerver ezt nem engedélyezi.) Ez a konfiguráció akkor előnyös, ha a munkáltatót szigorú adatvédelmi korlátok kötik, vagy nem szeretné, hogy a munkavállalók a munkavégzésen egyéb dolgokkal is foglalkozzanak. (A VPN szerver tűzfalán keresztül elérhetővé tehetők bizonyos kapcsolati formák, vagy korlátozások léptethetőek érvénybe.)

A funkció konfigurálásához ugyancsak a `server.conf` fájlban keressük meg a `redirect-gateway` direktívát és távolítsuk el a sor elejéről a pontosvesszőt:

```
push "redirect-gateway def1 bypass-dhcp"
```

Közvetlenül a sor alatt keressük meg a `dhcp-option` direktívát tartalmazó sorokat és ugyancsak távolítsuk el a pontosvesszőt a sor elejéről:

```
push "dhcp-option DNS 208.67.222.222"

push "dhcp-option DNS 208.67.220.220"
```

Ez a két sor a kliens gépek számára nyújt majd háttér információt DNS beállításaikhoz a VPN alagútra történő csatlakozáskor.

5.2 A VPN szolgáltatás portjának és protokolljának átkonfigurálása (OPCIONÁLIS)

Az OpenVPN szerver alapértelmezett csatlakozási portja az `1194`, melyen keresztül UDP protokollal fogadja a kliensek kapcsolódását. A fogadó port tetszés szerint megváltoztatható (ugyancsak óvintézkedésképp) és az adatátvitel protokollját is meg lehet választani TCP vagy UDP között. Ezen beállítások érvényre juttatásához az előzőekben szerkesztett `server.conf` fájlban az alábbi módosítások szükségesek, ha például a HTTPS által használt `443`-as portra szeretnénk átkonfigurálni a fogadó portot:

```
port 443
```

Az adatátvitel protokolljának megváltoztatásához TCP-re pedig az alábbi módosítás szükséges a fájlban:

```
proto tcp
```

Amennyiben az adatátviteli protokollt megváltoztattuk TCP-re, úgy a `explicit-exit-notify` direktíva értékét is meg kell változtatni:

```
explicit-exit-notify 0
```

Megjegyzendő, hogy az átállított portok és protokollok függvényében a tűzfal beállításain is változtatni kell, hogy a kliensek elérhessék a VPN szervert. A tűzfal beállításáról itt lehet bővebben olvasni.

5.3 A TSL 1.0 és 1.1 verziójának blokkolása

Az OpenVPN kommunikációs protokoll a TSL 1.2 verzióját (illetve korábbi 1.1 és 1.0 verzióit) támogatja. A TSL 1.0 és 1.1 korábbi években felfedezett sérülékenységeiből kifolyólag ajánlott a VPN szerverrel való kommunikációt a TSL 1.2 verziójára korlátozni. Ezt megtehetjük a `server.conf` fájlban az alábbi direktíva beírásával:

```
# allow only TLS version 1.2

tls-version-min 1.2
```

5.4 A VPN szerver tanúsítványaihoz tartozó elérési útvonalak beállítása

A 4.3-as fejezetben létrehoztuk a VPN szerver SSL tanúsítványit, azonban a *server.conf* fájlban meg kell adni azok elérési útvonalait. Ehhez megkeressük a *server.conf* fájlban a *cert* és *key* direktívákat és megadjuk hozzájuk az elérési útvonalakat. Amennyiben a *VPNserver.crt* és *VPNserver.key* fájlokat a */etc/openvpn* könyvtárba másoltuk, elég megadni a relatív elérési útvonalat is:

```
cert VPNserver.crt
```

```
key VPNserver.key
```

A változtatás befejeztével bezárjuk a fájlt.

6 A VPN szerver hálózati beállításai

6.1 Port továbbításának engedélyezése

A VPN szerver előző fejezetekben felkonfigurált beállításaihoz a VPN szerver hálózati beállításait is módosítani kell. Az egyik legfontosabb hálózati beállítás a hálózat port továbbítási funkciójának engedélyezése. Ez a */etc/sysctl.conf* fájl módosításával tehető meg:

```
$ sudo nano /etc/sysctl.conf
```

Ebben a fájlban keressük meg a *net.ipv4.ip_forward* sort és távolítsuk el a sor elejéről a # karaktert:

```
net.ipv4.ip_forward=1
```

A változtatás után zárjuk be a szerkesztett fájlt. A változtatásokat a

```
$ sudo sysctl -p
```

```
$ sudo sysctl –system
```

parancsokkal tudjuk érvényre juttatni és ellenőrizni.

6.2 Tűzfal beállítása

A tűzfal alapértelmezett beállítása során a VPN szerver kommunikációs portja nagy valószínűséggel zárva van. Ezért a VPN szerver eléréséhez szükséges portokat és kommunikációs protokollokat engedélyezni kell. Ugyancsak biztosítani kell az IP címek álcázását (angol szakkifejezéssel: masquerading), valamint módosítani kell a NAT (network address translation) beállításokat is, hogy a kliensek adatcsatornái megfelelően legyenek irányítva. Mielőtt megváltoztatnánk a hálózati konfigurációs fájlokat, állapítsuk meg az VPN szerver publikusan elérhető hálózati interfészét:

```
$ ip route | grep default
```

A publikusan elérhető hálózati interfész azonosítója a válaszként adott sorban található meg:

```
default via 203.0.113.1 dev enp0s3 proto static
```

A vastagon szedett rész jelöli a publikus hálózati interfész azonosítóját. Ennek ismeretében nyissuk meg a tűzfal `/etc/ufw/before.rules` konfigurációs fájlját (amennyiben az `ufw` csomagot használtuk a tűzfal konfigurálására):

```
$ sudo nano /etc/ufw/before.rules
```

A `before.rules` fájlban lévő beállítások a szerver bootolása során a tűzfal felállása előtt töltődnek be. A fájl első részében adjuk hozzá az alábbi utasításokat:

```
#
# rules.before
#
# Rules that should be run before the ufw command line added rules. Custom
# rules should be added to one of these chains:
#   ufw-before-input
#   ufw-before-output
#   ufw-before-forward
#


# START OPENVPN RULES
# NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to enp0s3 (change to the interface you
discovered!)
-A POSTROUTING -s 10.8.0.0/8 -o enp0s3 -j MASQUERADE
COMMIT
# END OPENVPN RULES


# Don't delete these required lines, otherwise there will be errors
*filter
. . .
```

A fájl szerkesztésének végeztével zárjuk be a fájlt. Ezután még engedélyezni kell a tűzfalban az adatcsomagok továbbítását alapértelmezett viselkedésként. Ehhez nyissuk meg a `/etc/default/ufw` fájlt:

```
$ sudo nano /etc/default/ufw
```

A fájlban keressük meg a `DEFAULT_FORWARD_POLICY` direktívát és módosítsuk `ACCEPT` értékre:

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

A módosítások mentése után zárjuk be a fájlt. Végül pedig meg kell nyitni a tűzfalban a VPN szerver külső eléréséhez szükséges csatornákat. Ezt az alábbi parancsokkal tehetjük meg:

```
$ sudo ufw allow 1194/udp
```

A kiadott parancsban módosítsuk az engedélyezett port értékét és a kommunikációs protokollt a szerver beállításainak függvényében (lásd a 5.2 fejezetet).

A módosítások érvénybe juttatásához a tűzfalat újra kell indítani:

```
$ sudo ufw disable
```

```
$ sudo ufw enable
```

6.3 Statikus IP címek kiosztása a klienseknek (OPCIONÁLIS)

Amennyiben hálózati kiszolgálókat (azaz szervereket) is szeretnénk üzemeltetni a VPN hálózatban (mint például projektmenedzser rendszer, vagy online meeting kiszolgáló), célszerű statikus IP címeket kiosztani ezen szerverek számára. Ezt az alábbi beállítások segítségével tehetjük meg:

Az OpenVPN szoftver telepítési könyvtárában (`/etc/openvpn`) hozzunk létre egy `ccd` mappát, melyben a kiosztásra szánt statikus IP címek konfigurációs fájljait fogjuk tárolni:

```
$ sudo mkdir /etc/openvpn/ccd
```

Ezután nyissuk meg a `/etc/openvpn` mappában a `szerver.conf` fájlt és egy tetszőleges helyen adjuk hozzá a `client-config-dir ccd` direktívát:

```
…
# specific settings for the clients are stores in deirctory ccd
client-config-dir ccd
…
```

Ezzel a VPN szerverrel tudattuk, hol tároljuk a statikus IP címek listáját. A változtatások mentése után lépjünk ki a fájlból.

Ezután lépjünk be a `ccd` könyvtárba, és hozzunk létre egy szerkeszthető fájlt minden statikus IP címmel ellátandó kliens számára. Esetünkben tegyük fel, hogy a client1 végfelhasználó számára szeretnénk statikus IP címet kiosztani a VPN hálózaton belül, vagyis hozzuk létre a `client1` fájlt:

```
$ sudo touch /etc/openvpn/ccd/client1
```

Ha a `client1` felhasználónak a `10.8.0.2` statikus IP címet szeretnénk kiosztani, akkor a létrehozott fájlba az alábbi sort illesszük be:

```
ifconfig-push 10.8.0.2 255.255.255.255
```

A változtatások érvénybe juttatásához (amennyiben már aktív volt a VPN hálózat) a VPN hálózatot indítsuk újra. Ennek módját a 7. fejezetben tárgyaljuk

6.4 A VPN hálózathoz csatlakozott kliensek közti kommunikáció

Az OpenVPN szerverhez kapcsolódott kliensek közti kommunikáció alapértelmezett esetben nincs engedélyezve a VPN hálózaton. Ez problémát jelent, ha például a klienseknek a VPN hálózathoz csatlakozott kiszolgálókat (például videokonferencia szervert) kell elérniük. A kleinsek közti kommunikáció engedélyezéséhez nyissuk meg a VPN szerver konfigurációs fájlját:

```
$ sudo nano /etc/openvpn/server.conf
```

A fájlban keressük meg `client-to-client` direktívát és távolítsuk el a sor elejéről a pontos vesszőt:

```
...
# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
client-to-client
```

…

A változtatások mentése után indítsuk újra a VPN szervert a 7.1 fejezetben leírtak alapján.

7 Az OpenVPN szerver indítása

Az eddigi konfigurációs lépések végeztével végre elindíthatjuk a VPN szerverünket. Ezt a operációs rendszer megfelelő eszközével tehetjük meg, Ubuntu esetében:

```
$ sudo systemctl start openvpn@server
```

Az utasítás végén azon fájl neve szerepel, melyben az OpenVPN konfigurációs adatait mentettük. A mi esetünkben ez a `/etc/openvpn/server.conf` fájl, ezért az utasítás végére a `@server` sztringet írjuk. Az OpenVPN szerver állapotára a

```
$ sudo systemctl status openvpn@server
```

paranccsal kérdezhetünk rá. Amennyiben a VPN szerver jól működik, az alábbiakhoz hasonló választ kapunk a szervertől:

- `openvpn@server.service - OpenVPN connection to server`

```
     Loaded: loaded (/lib/systemd/system/openvpn@.service; disabled; vendor
preset: enabled)
   Active: active (running) since Tue 2016-05-03 15:30:05 EDT; 47s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn23ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
  . . .
```

Ugyancsak ellenőrizhetjük a VPN hálózat interfészének elérhetőségét:

```
$  ip addr show tun0
```

A rendszer válasza az utasításra az alábbihoz hasonló:

```
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UNKNOWN group default qlen 100
    link/none
    inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
       valid_lft forever preferred_lft forever
```

Az OpenVPN szolgáltatás elindítása után a

```
$ sudo systemctl enable openvpn@server
```

utasítással rögzítjük, hogy a VPN szerver a szerver újraindítása után automatikusan elinduljon.

7.1 Az OpenVPN szerver leállítása és újraindítása

Az elindított OpenVPN szervet a

```
$ sudo systemctl stop openvpn@server
```

paranccsal tudjuk leállítani, majd a `start` direktívával újra indíthatjuk, amennyiben változtatásokat hajtottunk végre a beállításokban.

8 Kliens felhasználók hozzáadása a hálózathoz

A VPN szerver sikeres konfigurálása és elindítása után gondoskodnunk kell a kliens felhasználók hálózathoz történő csatlakozásáról. Az alábbiakban bemutatjuk, hogyan hozhatunk létre egy szkript fájlt, mellyel félautomatikusan adhatjuk hozzá a felhasználókat privát hálózatunkhoz. A folyamat megértéséhez tudnunk kell, hogy minden felhasználó saját konfigurációs fájllal tud majd csatlakozni a VPN hálózathoz, mely konfigurációs fájl tartalmazza a kliensek számára generált egyedi SSL tanúsítványok publikus részét is.

Először hozzuk létre a VPN szerveren a `client-configs` mappát, melyben tárolni fogjuk a kliensek létrehozott konfigurációs fájljait.

```
$ mkdir -p ~/client-configs/files
```

Ezután másoljuk a létrehozott mappába az OpenVPN szoftver által javasolt alapértelmezett kliens konfigurációs fájlt:

```
$ cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf ~/client-configs/base.conf
```

Nyissuk meg ezt a fájlt kedvenc szerkesztőnkkel:

```
$ nano ~/client-configs/base.conf
```

A fájlban keressük meg a `remote` direktívát, mely az OpenVPN szerverünk címére kell hogy mutasson:

```
. . .

# The hostname/IP and port of the server.

# You can have multiple remote entries

# to load balance between the servers.

remote your_VPNserver_ip 1194

. . .
```

A `remote` sorban írjuk be a VPN szerver publikusan elérhető IP címét (`your_VPNserver_ip` helyébe), valamint a szolgáltatás számára megnyitott portot, melyet a 5.2 fejezetben beállítottunk és melyet a 6.2 fejezetben megnyitottunk a tűzfalban. Arra is ügyeljünk, hogy a fájlban az adatátvitel protokollja megegyezzen a `server.conf` konfigurációs fájlban megadottal (UDP vagy TCP). UDP protokoll esetében tehát a `base.conf` fájlban:

```
proto udp
```

A protokoll beállítása után töröljük a pontosvesszőket a `user` és `group` direktívák elől:

```
# Downgrade privileges after initialization (non-Windows only)

user nobody

group nogroup
```

Ezután keressük meg a `ca, cert` és `key` direktívákat a fájlban. Ezen sorok elejéről távolítsuk el a `#` karaktert. (A tanúsítványok elérési útvonalait a szkriptünk automatikusan fogja frissíteni.)

```
 # SSL/TLS parms.

# See the server config file for more

# description.  It's best to use

# a separate .crt/.key file pair

# for each client.  A single ca

# file can be used for all clients.

ca ca.crt
```

```
cert client.crt

key client.key
```

Hasonlóan távolítsuk el a `#` karaktert a `tls-auth` direktíva elől is, mivel a `ta.key` kulcsot is közvetlenül a konfigurációs fájlba fogjuk menteni:

```
# If a tls-auth key is used on the server

# then every client must also have the key.

tls-auth ta.key 1
```

Ezen felül másoljuk be a szerkesztett `~/client-configs/base.conf` fájlba a `cipher` és `auth` direktívákat is a server.conf fájlból:

```
cipher AES-256-CBC

auth SHA256
```

Majd adjuk írjuk be a `base.conf` fájlba egy tetszőleges helyre a `key-direction` direktívát is. Ennek értékét `1`-re kell állítani:

```
key-direction 1
```

Végül adjunk hozzá a `base.conf` fájlhoz három inaktív sort is, melyeket abban az esetben szükséges aktívvá tenni, ha a Linux klienseken az OpenVPN szoftver tartalmaz `/etc/openvpn/update-resolv-conf` fájlt is.

```
# script-security 2

# up /etc/openvpn/update-resolv-conf

# down /etc/openvpn/update-resolv-conf
```

A változtatások mentésével lépjünk ki a szerkesztett fájlból. A következő lépésben létrehozzuk a szkript fájlt, mellyel automatizálhatjuk a kliens felhasználók felvételét a VPN hálózatba.

8.1 Szkript a kliensek konfigurációs fájljainak létrehozásához

A szkript elkészítéséhez hozzuk létre a `make_config.sh` fájlt a `~/client-configs` mappában:

```
$ nano ~/client-configs/make_config.sh
```

A fájlba az alábbi sorokat másoljuk be:

```
#!/bin/bash


# First argument: Client identifier


KEY_DIR=~/client-configs/keys

OUTPUT_DIR=~/client-configs/files
```

```
BASE_CONFIG=~/client-configs/base.conf


cat ${BASE_CONFIG} \
    <(echo -e '<ca>') \
    ${KEY_DIR}/ca.crt \
    <(echo -e '</ca>\n<cert>') \
    ${KEY_DIR}/${1}.crt \
    <(echo -e '</cert>\n<key>') \
    ${KEY_DIR}/${1}.key \
    <(echo -e '</key>\n<tls-auth>') \
    ${KEY_DIR}/ta.key \
    <(echo -e '</tls-auth>') \
    > ${OUTPUT_DIR}/${1}.ovpn
```

A fájl tartalmát mentsük, majd pedig lépjünk ki belőle. E kilépés után állítsuk a fájlt futtathatóvá:

```
$ chmod 700 ~/client-configs/make_config.sh
```

A létrehozott szkript fájl

1. létrehoz egy másolatot az alapértelmezett kliens konfigurációs fájlból (`base.conf`)
2. összegyűjti az új klienshez tartozó SSH tanúsítványok publikus és privát részeit és tartalmukat bemásolja a kliens újonnan létrehozott konfigurációs fájljába. (Tehát nem lesz szükség több fájl mozgatására a VPN szerver és kliensek között, csupán a konfigurációs fájlt kell eljuttatni biztonságos módon a klienshez.)

Ezen a ponton megjegyezzük, hogy minden új felhasználó számára külön kell generálni az SSL tanúsítványokat a korábban létrehozott CA szerver bevonásával, ahogy ezt a soron következő fájlban bemutatjuk.

8.2 Hitelesített SSL tanúsítványok generálása a kliensek számára

Bár az SSL tanúsítványokat a kliens felhasználók gépein is lehet generálni, majd pedig a CA szerverünkkel hitelesíthetjük azokat, ebben az útmutatóban azt javasoljuk biztonsági és praktikussági okok miatt, hogy a kliensek SSL tanúsítványai a 4.3 fejezetben bemutatott, a VPN szerver esetében eredményesen alkalmazott módszerrel történjen.

A soron következő lépésekben egy kliens felhasználó részére hozzuk létre a tanúsítványokat (a klienst jelen esetben nevezzünk `client1`-nek), több felhasználó esetében a lépéseket minden felhasználó esetében meg kell ismételni.

Először a **VPN szerveren** hozzuk létre a kliensek titkos kulcsok tárolására szolgáló mappát:

```
$ mkdir -p ~/client-configs/keys
```

A létrehozott mappát lássuk el megfelelő biztonsági korlátozásokkal:

```
$ chmod -R 700 ~/client-configs
```

Ezután navigáljunk el a tanúsítványokat létrehozó EasyRSA csomag könyvtárába és hozzunk létre jelszómentes SSL tanúsítványt a `client1` részére:

```
$ cd ~/EasyRSA-3.0.4/
```

```
$ ./easyrsa gen-req client1 nopass
```

A tanúsítvány generálása során szimplán nyomjuk meg az `enter` gombot, hogy a tanúsítvány a megadott `client1` fájlnévvel jöjjön létre. Ezután másoljuk át biztonságos csatornán a CA szerverünkre a `client1.req` fájlt.

```
$ scp pki/reqs/client1.req usernname@your_CA_ip:/tmp
```

(a `username` és a `your_CA_ip` helyére helyettesítsük be a CA szerverünkre való bejelentkezéshez szükséges adatokat), illetve a titkos kulcsot másoljuk be a kliensek nyilvántartására létrehozott mappába a VPN szerveren:

```
$ cp pki/private/client1.key ~/client-configs/keys/
```

Ezután lépjünk be a CA szerverünkbe, és navigáljunk át az EasyRSA mappába:

```
$ ssh username@your_CA_ip
```

```
$ cd EasyRSA-3.0.4/
```

```
$ ./easyrsa import-req /tmp/client1.req client1
```

Ezután hitelesíthetjük a CA szerverrel a `client1` tanúsítványát:

```
$ ./easyrsa sign-req client client1
```

A szerver rákérdez hitelesítési szándékunkra. Válaszoljunk a `yes` szócska begépelésével és nyomjuk meg az `enter` gombot. A hitelesítéssel létrejön egy `client1.crt` fájl, melyet másoljunk vissza a VPN szerverre:

```
$ scp pki/issued/client1.crt username@your_VPNserver_ip:/tmp
```

Lépjünk vissza a VPN szerverünbe és másoljuk át a jóváhagyott publikus tanúsítványt a `/client-configs/keys/` mappába:

```
$ cp /tmp/client1.crt ~/client-configs/keys/
```

Végül másoljuk a CA szerver publikus tanúsítványát, valamint a HMAC kulcsot is a kliensek nyilvántartását szolgáló mappába:

```
$ cp ~/EasyRSA-3.0.4/ta.key ~/client-configs/keys/
```

```
$ sudo cp /etc/openvpn/ca.crt ~/client-configs/keys/
```

Ezzel minden tanúsítvány a helyén van ahhoz, hogy létrehozzuk a kliens belépési adatait tartalmazó konfigurációs fájlt. (A továbbiakban új kliensek hozzáadásakor értelemszerűen elegendő az új kliens felhasználók tanúsítványainak a mozgatása, a CA szerver tanúsítványának másolását kihagyhatjuk a lépések közül.)

8.3 Az új kliens felhasználók konfigurációs fájljának létrehozása

Az új kliens tanúsítványának létrehozása után a 8.1 fejezetben létrehozott szkript segítségével létrehozhatjuk a kliens VPN hálózatba való belépéséhez szükséges konfigurációs fájlt, melyet majd biztonságos úton el kell juttatni a klienshez.

```
$ cd ~/client-configs
```

```
$ sudo ./make_config.sh client1
```

Ez utóbbi utasítás létrehozza a `client1.ovpn` fájlt a `~/client-configs/files` mappában. Ezt a fájlt felhasználva tudja a kliens felhasználó bekonfigurálni saját VPN kliens szoftverét, mellyel be tud lépni a VPN hálózatba. A kliens oldali szoftverek telepítésével és a VPN hálózatba való belépéssel egy különálló útmutatóban foglalkozunk.

8.4 Kliensek belépési jogosultságának megvonása

A VPN szerver működése során olykor igény merülhet fel egyes kliensek hozzáférési jogának megvonása. Ehhez navigáljunk a CA szerverünkön telepített EasyRSA csomag mappájába:

```
$ cd EasyRSA-3.0.4/
```

Ezután hívjuk meg az `easyrsa` szkriptet a `revoke` opcióval, valamint a hozzáférési jogosultság megvonásával sújtott kliens felhasználó (például `client2`) azonosítójával:

```
./easyrsa revoke client2
```

A megvonás megerősítése céljából gépeljük be a szerver kérésére a `yes` szócskát és nyomjuk meg az `enter` gombot. A `client2` felhasználó tanúsítványának megvonásáról a VPN szervert is értesíteni kell. Ennek érdekében hozzuk létre a megvont tanúsítványok listáját:

```
$ ./easyrsa gen-crl
```

Az utasítással létrehozott `crl.pem` fájlt másoljuk át a VPN szerverre:

```
$ scp ~/EasyRSA-3.0.4/pki/crl.pem username@your_VPNserver_ip:/tmp
```

A VPN szerveren másoljuk át a fájlt az OpenVPN szoftver mappájába:

```
$ sudo cp /tmp/crl.pem /etc/openvpn
```

Ezután nyissuk meg a VPN szerver konfigurációs fájlját:

```
$ sudo nano /etc/openvpn/server.conf
```

A megnyitott fájl végére illesszük be a `crl-verify` opciót, mely a VPN szervert arra utasítja, hogy nézze át a visszavont tanúsítványok listáját minden alkalommal, amikor valaki be szeretne jelentkezni.

```
crl-verify crl.pem
```

A fájlban mentsük a változtatásokat, majd pedig lépjünk ki belőle és indítsuk újra a VPN szerverünket:

```
$ sudo systemctl restart openvpn@server
```

A `client2` felhasználó ezután már nem fog tudni bejelentkezni a VPN szerverre. További kliensek belépési jogosultságának megvonásához az alábbi lépéseket végezzük el:

1. A CA szerveren vonjuk vissza az érintett kliens tanúsítványának hitelességét.
2. Generáljuk le a visszavont tanúsítványok listáját.
3. Másoljuk át a generált listát a VPN szerver `/etc/openvpn` mappájába felülírva az előző listát.
4. Indítsuk újra a VPN szervert.