Beállítás (gyakorlat)

Előkészületek

Ennél a példánál a site-to-site példában megadott VPN környezetet feltételezzük. Ide értve az ott létrehozott tanúsítványokat is.

Az office router-en fogunk kialakítani egy olyan VPN szerver beállítást, road warrior kliensek csatlakozását is fogadni tudja. Ahogy ott, itt is nagyon fontos az NTP kliens helyes beállítása.

Tanúsítványok

Arra készülünk, hogy sok road warrior kliens fog csatlakozni. Ezért egy úgynevezet tanúsítvány sablont hozunk létre. Ebből később könnyebb lesz származtatni az egyes kliens tanúsítványokat.

```
/certificate
add name="~client-template@office.myserver.hu"  \
    country="HU" state="Heves" locality="Eger" \
    organization="office.myserver.hu"  \
    key-size=4096 days-valid=1095 \
    common-name="~client-template@office.myserver.hu" \
    subject-alt-name="email:~client-template@office.myserver.hu" \
    trusted=yes key-usage=tls-client
```

A template-et nem írjuk alá! Minden klienshez külön kliens tanúsítványt generálunk. Kliens alatt nem felhasználót kell érteni, hanem VPN kliens eszközt. Ha egy felhasználó több számítógépről is be akar jelentkezni (pl. a telefonjáról és a laptopjáról), akkor ezekhez külön tanúsítványt kell létrehozni. Mi most egy gandalf nevű felhasználóhoz és egy vivobook laptophoz tartozó tanúsítványt generálunk a template-ből:

```
/certificate
add copy-from=~client-template@office.myserver.hu \
    name="gandalf-vivobook@office.myserver.hu" \
    common-name="gandalf-vivobook@office.myserver.hu" \
```

```
    subject-alt-name=email:gandalf-vivobook@office.myserver.hu
sign gandalf-vivobook@office.myserver.hu ca=ca.office.myserver.hu
```

Ezután exportáljuk a kliens és a CA cert-et. A kliens cert-hez a privát kulcsot is.

```
/certificate
export-certificate gandalf-vivobook@myserver.hu type=pkcs12 export-
passphrase="not_telling_you"
export-certificate ca.office.myserver.hu
```

VPN kliensek alhálózata

A VPN klienseknek egy saját címtartományt adunk. A router-en fölveszünk egy hidat, és adunk neki egy címet ezen az alhálózaton belül. A kapcsolódó kliensek ebből a tartományból fognak címeket kapni.

```
/interface bridge
add name=bridge-vpn-rw comment="VPN Road Warrior"
/ip address
add address=10.10.10.254/24 comment="VPN Road-Warrior" name=pool-vpn-rw
```

mode-config

Az IKE/ISAKMP protokoll nem csak kulcsok és titkosítási algoritmusok egyeztetésére használható. Ezen keresztül lehet IP címet beállítani, közölni a másik féllel az elérhető alhálózatokat ("route push") stb. Ezen paraméterek megadására való a mode-config.

```
/ip ipsec mode-config
add address-pool=pool-vpn-rw address-prefix-length=32 \
    name=modeconf-vpn-rw split-include=172.16.1.0/24,172.16.2.0/24 \
    static-dns=172.16.1.1 system-dns=no responder=yes
```

- Az address-pool mondja meg, hogy melyik IP pool-ból osztunk címeket
- Az address-prefix-length mondha meg, hogy mennyi címet osztunk. (Nem csak egyedi címet lehet átadni, hanem teljes címtartományokat is.)
- A split-include segítségével lehet megadni egy vagy több helyi alhálózatot. A másik peer megkapja ezeket a hálózatokat. Értesül arról, hogy ezen a peer-en keresztül milyen hálózati címtartományok érhetők el. Ennek a policy generálásnál lesz szerepe (erről később írok)

- A static-dns megadásával lehet átküldeni azt a DNS szervert, ami a helyi hálózat címeit tudja feloldani.
- A system-dns=no beállítás akadályozza meg azt, hogy a helyi router DNS-t (/ip dns és /ip dns static) adja át a másik félnek
- A responder=yes beállítás hatására ez a peer küldi át a mode-config beállításokat a távoli peer-nek. Ha ez no-ra van állítva, akkor ez az initiator oldal.

IKE Phase 1 (ipsec profile) és phase 2 (ipsec proposal)

Hogy ez pontosan mit jelent, azt a site-to-site VPN résznél már leírtam. Ami ahhoz képest eltérés, az az algoritmusok megválasztása. Itt többféle kliensre számítunk: Windows 10, Linux, iOS, különböző android verziók stb. Próbálunk beállítani biztonságos algoritmusokat. De figyelembe kell vennünk azt, hogy várhatóan milyen típusú klienseket szeretnénk támogatni, és hogy ezek a kliensek milyen algoritmusokat támogatnak, plusz még azt is, hogy milyen algoritmusokhoz van hardveres gyorsítás a router-ben.

```
/ip ipsec profile
add dh-group=modp2048,modp1536,modp1024 \
    enc-algorithm=aes-256,aes-192,aes-128 \
    hash-algorithm=sha256 \
    name=profile-vpn-rw \
    nat-traversal=yes proposal-check=claim

/ip ipsec proposal
add auth-algorithms=sha512,sha256,sha1 \
    enc-algorithms=aes-256-cbc,aes-256-ctr,aes-192-cbc,aes-192-ctr,aes-128-cbc,aes-128-ctr \
    lifetime=8h name=proposal-vpn-rw pfs-group=none
```

Hogy melyik OS milyen algoritmusokat támogat, arról például itt is tájékozódhatsz:

- Windows
- macOS
- IOS
- Android (StrongSwan)
- Linux (StrongSwan)

Érdekes, hogy például a Windows 10 kliensek az első IKE fázisban támogatják a modp1024 + SHA256 beállítást, de a második fázisban hash algoritmusnak kizárólag az SHA1-et támogatja, pfs-t pedig egyáltalán nem tud használni. (Ezért engedélyeztem az SHA1-et, máskülönben Windows-ból nem lehetne csatlakozni.)

Figyelem! Ha mégis megpróbálod beállítani a pfs-group értéket valami none-tól különböző értékre, akkor először azt fogod tapasztalni hogy működik. De amint letelt a megadott lejárati idő, a MikroTik oldal meg fogja követelni a kulcs lecserélését. Azonban a Windows ezt nem fogja megtenni (mert ezt nem támogatja). Ha például 4 óra a lejárat, akkor 4 óra folyamatos kapcsolat után fog "megszakadni" a kapcsolat. Ráadásul úgy, hogy MikroTik-ből connected-nek látszódik, csak épp elérhetetlen lesz a másik oldal. Ilyenkor egy gyors megoldás a kapcsolat újraépítésére az, hogy az identity-t disabled-re állítod, majd újra enabled-re. Ez a MikroTik oldalon törli a phase1 -et is, és így a kapcsolat újra ki tud épülni.

Policy group

A policy group egy névvel ellátott csoportot határoz meg. A nevén kívül semmit nem lehet itt megadni. Ennek az a jelentősége, hogy ezzel lehet később összekapcsolni az identity-t és a policy template-eket. (Hogy ez miért fontos, azt később fogom leírni.)

```
/ip ipsec policy group
add name=group-vpn-rw
```

Policy template

A site-to-site VPN példában kézzel, statikus policy -kat adtunk hozzá. Ezt azért tehettük meg, mert előre ismertük az alhálózatok címeit, és ezeket néhány telephelyen meg tudtuk adni kézzel. Ez az út nem járható a road warrior kliensek esetén, sok okból:

1. Ha ezeket kézzel akarnánk fölvenni a klienseken, akkor a kliensek nagy száma miatt ez túl nagy munka lenne.
2. Ráadásul a VPN-en keresztül elérhető alhálózatok listája bármikor megváltozhat a VPN szerveren, és ilyenkor nem szeretnénk, ha az összes road warrior klienst kézzel újra kellene konfigurálni.
3. Ráadásul a kliensek általában erre nincsenek fölkészülve. Egy Windows 10 vagy egy macOS kliens alapértelmezésben a szervertől kéri le a split-include -ban megadott alhálózatok listáját, és ezekhez automatikusan generálja a policy-ket. Bár elvileg lehetséges ezeket kézzel előre hozzáadni, de ezek a kliensek alapvetően nem erre vannak kitalálva, és ez nem véletlen.

A VPN szerver oldalán hasonló a helyzet. Bár megadhatnánk kézzel a policy-ket, de akkor duplán végeznénk el a munkát. Ha a mode-config -ban már megadtuk a split-include direktívával a kliensek számára elérhető alhálózatokat, akkor ez alapján automatikusan generálhatunk policy-ket a szerver oldalon is. (A kapcsolat sikeres felépülésekor.)

```
/ip ipsec policy
add dst-address=10.10.10.0/24 \
group=group-vpn-rw proposal=proposal-vpn-rw \
```

```
    src-address=0.0.0.0/0 template=yes \
    ipsec-protocols=esp level=require \
    protocol=all action=encrypt
```

Ipsec peer

```
/ip ipsec peer
add exchange-mode=ike2 \
    address=0.0.0.0/0 \
    local-address=89.134.227.161 \
    name=peer-vpn-rw \
    passive=yes send-initial-contact=yes \
    profile=profile-vpn-rw
```

Részletek:

- az address=0.0.0.0/0 miatt bármilyen címről fogadunk beérkező kapcsolatokat (nem tudjuk előre megmondani, hogy a road warrior honnan fog kapcsolódni)
- a local-address adja meg azt a címet, ahol az IKE démon figyel (ehhez a peer-hez!)
- a passive=yes, send-initial-contact=yes beállítás az ami miatt ebből egy olyan responder lesz, ami várja a bejövő kapcsolatokat
- a profile az nyilván az IKE kulcs-csere első fázisának beállításait tartalmazza.

Ipsec indentity

Ez szolgál a kliensek azonosítására. Minden klienshez külön identity-t kell létrehozni!

```
/ip ipsec identity
add auth-method=digital-signature \
    certificate=ca.myserver.hu \
    remote-certificate=gandalf-vivobook@myserver.hu \
    generate-policy=port-strict \
    match-by=certificate \
    mode-config=modeconf-vpn-rw \
    peer=peer-vpn-rw \
    policy-template-group=group-vpn-rw \
    remote-id=user-fqdn:gandalf-vivobook@myserver.hu
```

Itt most csak a site-to-site VPN -hez képesti eltéréseket írom le:

- meg van adva a mode-config, ez tartalmazza a split-include beállítást, a VPN szerver ezt veszi alapul

- meg van adva a generate-policy=port-strict - ezt használjuk a policy generáláshoz.

A policy generálás folyamata:

- A kliens csatlakozik, azonosítja magát az identity-ben megadott certificate-tel.
- Elküldi az általa elfogadhatónak ítélt címtartományokat, amik erre a kapcsolatra vonatkoznak. (Ez egyébként a kliens oldali policy template-nek felel meg!)
- A szerver ezt összeveti azokkal a policy template-ekkel, amik ahhoz a policy template group-hoz tartoznak, amiket az identity-ben megadtunk. (Ez a szerver oldali policy template.)
- A kettő összevetése után határozza meg azokat a tartományokat amiket mindkét oldal elfogadhatónak talál.
- Ha nincsen közös rész, akkor abból kapcsolódási hiba lesz.
- Ha van közös rész, akkor a helyi policy template-ek alapján konkrét, dinamikus policy-ket generál.
- Ezek a dinamikus policy-k csak addig élnek, amíg a kapcsolat él. A kapcsolat megszakadásakor ezek a policy-k automatikusan törlésre kerülnek.

Ezek alapján most már jobban érthető az, amit a site-to-site kapcsolatnál a package leaking-ről írtunk. A kapcsolat bármikor megszakadhat, és emiatt a policy-k egy része bármikor eltűnhet. Fokozottan fontos hogy a router-t úgy konfiguráljuk, hogy titkosítatlan, privát címtartományhoz tartozó csomagok véletlenül se kerüljenek ki az internetre.

Tűzfal beállítások

Az IKE és IPSEC -hez itt is szükség van az 500 és 4500 -as UDP port megnyitására, valamint az esp csomagok fogadására.

```
/ip firewall filter
add place-before=0 \
    protocol=udp dst-port=500,4500 \
    dst-address=myserver.hu.public.ip \
    action=accept \
    chain=input \
    comment="Allow UDP 500,4500 IPSec for myserver.hu.public.ip"
add action=accept chain=input comment="Allow IPSEC/ESP" \
    dst-address=myserver.hu.public.ip \
protocol=ipsec-esp \
    place-before=1
```

Ezután engedélyezünk minden forgalmat a VPN klienseknek kiosztott IP címtartomány felől **mindenhová**.

```
/ip firewall filter
add chain=input src-address=10.10.10.0/24 \
    ipsec-policy=in,ipsec action=accept \
    place-before=[ find where comment~"defconf: drop all not coming from LAN" ] \
    disabled=no \
    comment="IKE2: Allow ALL incoming traffic from vpn-rw client to this router"

add chain=forward \
    src-address=10.10.10.0/24 \
    ipsec-policy=in,ipsec action=accept \
    place-before=[ find where comment~"defconf:  drop all from WAN not DSTNATed" ] \
    disabled=no \
    comment="IKE2: Allow ALL incoming traffic from vpn-rw client to ANY"
```

Ha nem szeretnéd, hogy a kliensek ezen keresztül internetezzenek, akkor a második szabály helyett beírhatsz egy olyat, ahol a dst-address -ben a helyi office privát LAN szerepel. Illetve természetesen ha akarod, akkor hozzáadhatod az összes többi branch címét is.

```
add chain=forward \
    src-address=10.10.10.0/24 \
    dst-address=office.lan.subnet/24 \
    ipsec-policy=in,ipsec action=accept \
    place-before=[ find where comment~"defconf:  drop all from WAN not DSTNATed" ] \
    disabled=no \
    comment="IKE2: Allow ALL incoming traffic from vpn-rw client to office LAN"

 add chain=forward \
    src-address=10.10.10.0/24 \
    dst-address=branch01.lan.subnet/24 \
    ipsec-policy=in,ipsec action=accept \
    place-before=[ find where comment~"defconf:  drop all from WAN not DSTNATed" ] \
    disabled=no \
    comment="IKE2: Allow ALL incoming traffic from vpn-rw client to branch01 LAN"

  add chain=forward \
    src-address=10.10.10.0/24 \
    dst-address=branch02.lan.subnet/24 \
    ipsec-policy=in,ipsec action=accept \
    place-before=[ find where comment~"defconf:  drop all from WAN not DSTNATed" ] \
```

```
    disabled=no \
    comment="IKE2: Allow ALL incoming traffic from vpn-rw client to branch02 LAN"
```

A mi példánkban az office router már tartalmaz egy általános masquerade szabályt, ami átírja az internet irányába menő összes csomag forráscímét az office rotuer WAN címére. Ide értve azokat is, amik a VPN kliensek felől érkeznek az alagúton át.

Általános esetben, ha nincsen beállítva ilyen szabály, akkor azt pluszban kézzel hozzá kell adni. (A mi példánkban erre nincs szükség, csak a teljesség kedvéért írtam ki.)

```
/ip firewall nat
add place-before=0 chain=srcnat src-address=10.10.10.0/24 \
out-interface-list=WAN \
    ipsec-policy=out,none \
    action=masquerade \
    comment="Masquerade vpn-rw clients --> WAN traffic"
```

Ezen felül van arra is lehetőség, hogy a VPN kliensek felől az office LAN (illetve bármely más branch) irányába menő csomagok címét is átfordítsuk.

```
/ip firewall nat
add place-before=0 chain=srcnat \
src-address=10.10.10.0/24 \
    dst-address=office.lan.subnet/24 \
    ipsec-policy=out,none action=masquerade \
    comment="SRC-NAT vpn-rw client -> office LAN "
add place-before=0 chain=srcnat \
src-address=10.10.10.0/24 \
    dst-address=branch01.lan.subnet/24 \
    ipsec-policy=out,none action=masquerade \
    comment="SRC-NAT vpn-rw client -> branch01 LAN "
add place-before=0 chain=srcnat \
src-address=10.10.10.0/24 \
    dst-address=branch02.lan.subnet/24 \
    ipsec-policy=out,none action=masquerade \
    comment="SRC-NAT vpn-rw client -> branch02 LAN "
```

Ezek közül az első opcionális, mert az office LAN-ján belülről a default route arra a router-re vezet, amelyikre a road warrior kliensek is csatlakoznak. A több branch esetében ez nem opcionális, mivel a branch01 LAN-ján belülről a road warrior kliensek 10.10.10.0/24 címéhez nincs statikus route megadva, a default route-on keresztül pedig kimegy az internetre titkosítatlanul a válasz, és ott

elakad... Alternatív megoldásként kézzel hozzá lehet adni route-okat és ipsec policy-ket a vpn-rw alhálózatra a branch-eken is, de ez elég fáradságos munka (minden branch-en +1 route és +1 policy, és az office-ban is...)

MTU TCP MSS

Ez szinte ugyan az mint ami a site-to-site leírásban volt, csak a szabályban a forráscím más.

```
add action=change-mss chain=forward \
    comment="Clamp TCP MSS from vpn-rw clients to ANY" \
    ipsec-policy=in,ipsec new-mss=1360 \
    passthrough=yes protocol=tcp \
    src-address=10.10.10.0/24 \
    tcp-flags=syn tcp-mss=!0-1360
```

Revision #20
Created 10 January 2021 16:36:25 by Gandalf
Updated 2 April 2021 18:52:06 by Gandalf