# Alapfogalmak

Most egy kis elméleti rész következik. Ahol lehet, ott be fogom írni hogy RouterOs-ben az adott rész beállításai melyik menüpont alatt érhetők el. Ezen felül gyakorlati, konkrét javaslatokat teszek az egyes beállításokra a mi konkrét példáinkra (két telephely folyamatos összeköttetése VPN alagúton, illetve road warrior setup).

Ez a protokoll több fő részre bontható:

- Az `IKE` felelős azért, hogy a kommunikációs csatorna két oldalán levő felek közösen és biztonságosan megegyezzenek olyan kriptografikus kulcsokban, amiket később a kommunikáció során föl lehet használni az adatforgalom titkosítására. Az `IKE` az Internet Key Exchange rövidítése. Két fő verziója van. Az egyes verziót már nem nagyon használjuk. A kettes verziót úgy alakították ki, hogy kiküszöbölje az első verzió használata közben tapasztalt hibákat és hiányosságokat. Az IKEv2 normál UDP csomagokat és alapértelmezésben az 500-as portot használja.
- A titkosított adatforgalom úgy zajlik, hogy a tunnel két végén levő eszközök az egymás irányába küldött csomagokat a küldés előtt enkapszulálják (azaz beágyazzák) `IPSEC` csomagok belsejébe, a beérkező csomagokat pedig dekapszulálják (kicsomagolják).
- Az enkapszulációra kétféle szabvány terjedt el: az Authentication Header (`AH`) és az Encapsulating Security Payload (`ESP`). Az `AH` arra alkalmas, hogy garantálja a csomagok eredetét. (Harmadik fél nem képes meghamisítani őket.) Az `ESP` ezen felül titkosítást is végez - harmadik fél nem képes hozzájutni az eredeti csomagok adattartalmához akkor sem, ha a két fél közötti csatorna összes `IPSEC` csomagját el tudja olvasni.

Bár erről később még részletesen írok, de elöljáróban megemlítek két fogalmat. Ezek szükségesek ahhoz, hogy elkerüljük az előre hivatkozásokat, és úgy előre-hátra ugrálás nélkül lehessen olvasni ezt a cikket.

## IPSEC Peer

Az IPSEC protokoll használatakor mindig van két kitüntetett csomópont. Az egyik a csomagok beágyazását (enkapszuláció) és titkosítását végzi. A másik a csomagok kicsomagolását (dekapszuláció), a titkosítás feloldását és az eredetiség vizsgálatot végzi. A gyakorlatban a kommunikáció kétirányú szokott lenni, ezért mindkét fél párhuzamosan végzi mindkét feladatot. Ezeknek a feleknek a neve `ipsec peer`. Fontos látni, hogy az enkapszulációt és dekapszulációt végző felek nem feltétlenül ugyan azok, mint akik a csomagokat eredetileg küldik vagy fogadják. A mi site-to-site példánkban az internet és a telephely (`branch01` és `office`) határán álló router-ek a `peer`-ek, mivel ők végzik a csomagok be- és kicsomagolását. Ugyanakkor a csomagok feladói és címzettjei a telephelyen belül található gépek. A gyakorlatban előfordulhatnak olyan hálózati topológiák is, ahol a csomagok  olyan útvonalon utaznak, ahol az út egyes szakaszait IPSEC

csomagokba ágyazva, egy másik részét normál módon teszik meg. Mi most csak a feladatleírásban szereplő problémára koncentrálunk.

> RouterOS esetén az ehhez tartozó menüpont a `/ip ipsec peer`. Itt lehet megadni az IPSEC kommunikációhoz a távoli peer-eket. A helyi peer-t nem kell külön megadni, mert az maga a router.

## IPSEC Policy

A másik fogalom az `ipsec policy`. Amikor egy csomagot továbbítani kell egy forrás címről egy cél címre, akkor a router elsősorban forgalomirányítási táblázatokat (routing table) használ annak megállapítására, hogy melyik csomagot melyik interface-en keresztül küldje ki. A csomagokat tűzfal szabályokkal tudjuk szűrni és manipulálni. A szűrés azt jelenti, hogy bizonyos csomagokat eldobunk ahelyett, hogy továbbítanánk őket. A manipuláció azt jelenti, hogy módosítjuk őket a továbbítás előtt. Ilyen manipuláció lehet például a forrás- vagy célcím megváltoztatása (srcnat és dstnat), a csomagok sorba állítása, várakoztatás és priorizálás stb. Bizonyos módosítások automatikusak, minden router a rajta áthaladó összes csomagot módosítja az IP protokoll szabályainak megfelelően. (Pl. TTL csökkentése) Az IPSEC protokoll a csomagok feldolgozását a fentiektől jól elkülöníthető rétegben végzi. A csomagok feldolgozása ebben a rétegben úgynevezett ipsec szabályok, szakszóval `ipsec policy`-k segítségével történik. Egy ilyen policy kicsit hasonlít egy tűzfal szabályra:

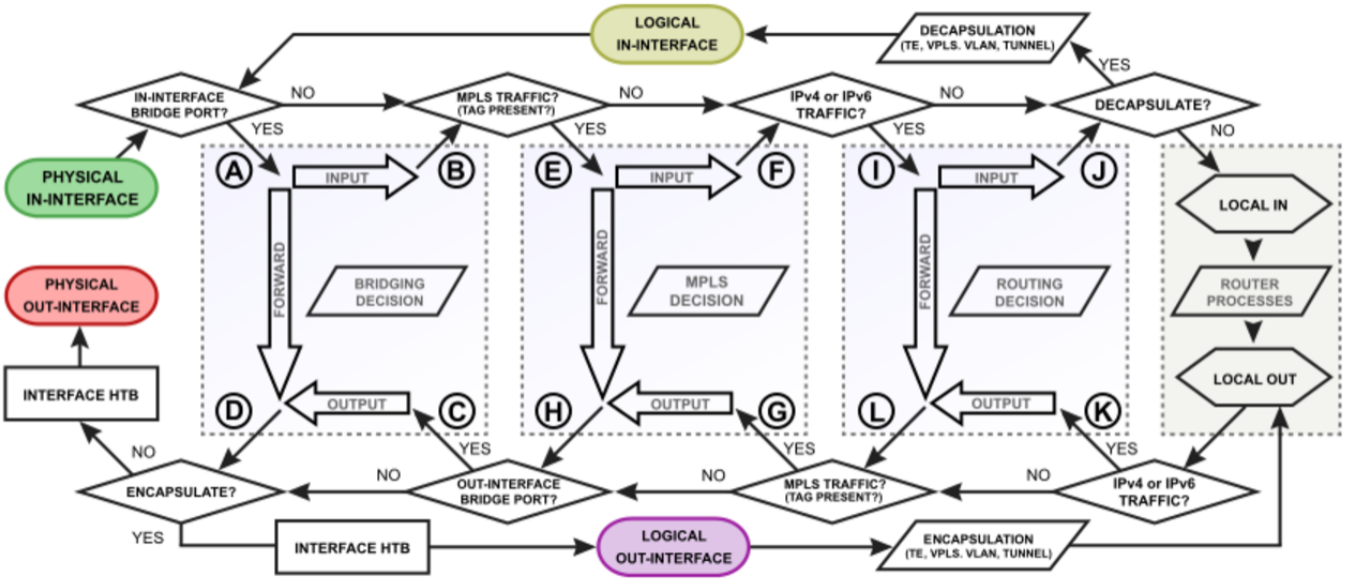
- IP csomagokon működik
- Minden szabályban feltételeket adunk meg. Ezek a feltételek vonatkozhatnak a csomag cél címére, forrás címére, az csomag típusára (TCP, UDP) stb.
- A feltételeken felül meg van adva egy művelet (`action`) is. Ez RouterOS esetén lehet `discard` (csomag eldobása) `encrypt` (csomag enkapszulálása és titkosítása) vagy `none` (ne csináljon vele semmit). Megjegyzés: az `encrypt` művelet egy kicsit megtévesztő lehet, mivel `AH` esetében nem történik titkosítás. Ami mindig megtörténik az az enkapszuláció.
- Amikor egy csomag illeszkedik a policy-ben megadott szabályokra, akkor a policy-ben megadott művelet végrehajtódik. Az `encrypt` művelet a csomagot beágyazza (enkapszulálja) egy IPSEC csomag belsejébe. Ennek során titkosíthat és ellenőrző összegeket írhat be az így előállított új csomagba.
- Fontos látni, hogy a beágyazás során az eredeti csomag "elhasználódik". Helyette egy teljesen új csomag képződik. Ez az új csomag szintén egy rendes IP csomag, de a protokoll száma már nem az eredeti (pl. UDP vagy TCP) hanem `IPSEC AH` vagy `IPSEC ESP`. Van rendes forrás- és célcíme, és a router az eredeti csomag helyett ezt a csomagot továbbítja. A fogadó oldal a nála megtalálható kulcs segítségével tudja kicsomagolni az eredeti csomagot. A kicsomagoláskor szükség lehet a titkosítás feloldására és az ellenőrző összeg ellenőrzésére. Az `AH` és `ESP` csomagok működés szempontjából datagram típusúak. A továbbítás nem feltétlenül sorrendhelyes. Ha a csomag valami miatt elveszik, akkor az IPSEC nem biztosít erről visszajelzést, és nem próbálja meg az újraküldést. Az eredeti feladó és címzett számára az IPSEC transzparens, nem látható. Ha az eredeti csomag kapcsolatállapot alapú (TCP) volt, akkor az esetleges csomag újraküldéseket is az

erdeti (pl. TCP) protokoll valósítja meg függetlenül attól, hogy a csomagok az út egyrészében IPSEC csomagokba vannak ágyazva.
- Nem létezik `decrypt` művelet. A bejövő csomagok esetén a RouterOS automatikusan felismeri az IPSEC csomagokat, és megpróbálja kicsomagolni őket. Más szóval a bejövő csomagokra nem kell külön policy-kat megadni.
- Több ilyen szabályt (`ipsec policy`) meg lehet adni. Ezek egymás után lefutnak, és ha van olyan ami illeszkedik, akkor a később következő szabályok már nem futnak le.
- A tűzfal szabályokkal ellentétben az `ipsec policy`-nál nincsenek `chain`-ek.

> RouterOS esetében a policy-k az `/ip ipsec policy` menüpont alatt találhatók.

## Enkapszuláció és dekapszuláció

A ki- és becsomagolási folyamatot jobban nyomon lehet követni ezen az ábrán:




Forrás: https://wiki.mikrotik.com/wiki/Manual:Packet_Flow

A három nagy szürke doboz balról jobbra sorrendben:

- Switch (layer 2 routing), ebben vannak az A,B,C,D virtuális portok
- MPLS, ebben vannak az E,F,G,H virtuális portok
- Router (layer 3 routing), ebben vannak az I,J,K,L virtuális portok
- Belső RouterOS folyamatok (local in, local out, router processes)

Látható, hogy a `decapsulate?` és `encapsulate?` döntést jelképező dobozok a switching és routing részeken kívül helyezkednek el. A kimenő csomagok közvetlenül a fizikai interfész elérése előtt érik el az `encapsulate?` dobozt. Tehát az enkapszuláció a switching és routing után történik. Ennek során az eredeti (enkapszulált) csomag "elhasználódik". Helyette egy új csomag jön létre, ami a `local out` belső processzben képződik, és innen kezdődik a feldolgozása. Hasonlóan, a

dekapszulációról való döntés is a switching és routing után történik. Ha dekapszulációra van szükség, akkor ezt a `local in` belső RouterOS processz végzi el. Ennek hatására az eredeti IPSEC csomag "elhasználódik", helyette megjelenik az eredeti (dekapszulált) csomag a `local out` processz kimenetén. Ezután megkezdődik a feldolgozása. Az IPSEC csomag és a dekapszulált csomag is áteshet routing/switching műveleteken. Sőt, ha többszörös beágyazás van, akkor ezek többször is megismétlődhetnek. Ha például az internet kapcsolat PPPoE protokollt használ, és ezen keresztül érkezik be egy IPSEC csomag, akkor az eredeti csomagot két dekapszuláció után kapjuk meg.

## Policy template és generált policy-k

A legegyszerűbb esetben mindkét peer-en kézzel adjuk meg a policy-kat. Tehát kézzel írjuk elő hogy mik azok a csomagok, amiket IPSEC csomagokba kell ágyazni, és biztonságosan továbbítani a másik peer-hez. Ezt a módszert akkor lehet használni, ha kevés számú, előre megismerhető peer-t használunk, és kevés számú policy-t. Van azonban egy másik módja is a policy-k előállításának, ez pedig a `policy template` ("szabály sablon") megadása. RouterOS-ben úgy tudunk ilyeneket létrehozni, hogy policy `template` attribútumát `yes` értékre állítjuk. RouterOS-ben ezek a policy-k úgy jelennek meg a `/ip ipsec policy` menü alatt, hogy egy `T` betű van a státusz oszlopban. Egyelőre ezekről elegendő tudni annyit, hogy a rendszer a policy template-ek alapján konkrét dinamikus policy-kat generál a kapcsolat létrehozásakor. Ezeket automatikusan kitörli akkor, amikor a kapcsolat befejeződik. Ezek működését egy másik cikkben (road warrior VPN) fogom elmagyarázni.

---

Revision #7
Created 2 January 2021 18:18:01 by Gandalf
Updated 9 January 2021 12:29:46 by Gandalf