# 3. fejezet
# A biztonsági technológiák áttekintése

Az ember avégből utazik a világban, mert keresi szükségleteit, végül hazatér, hogy megtalálhassa azokat.

*(Ismeretlen szerző)*

***A fejezet végén az alábbi kérdésekre választ kell tudni adni:***

- ***Miként használható a csomagszűrés a hálózatot fenyegető egyszerűbb fenyegetések elhárítására?***
- ***Pontosan mit is jelent az állapotteljes csomagvizsgálat?***
- ***Mi a közvetítő (proxy) technológia szerepe?***
- ***Mit jelent a hálózaticím-fordítás?***
- ***Miért lehet képes a nyilvános kulcsú infrastruktúra az információfolyam globális védelmére?***
- ***Mi a hitelesítési technológiák szerepe a hálózati erőforrások védelmében?***

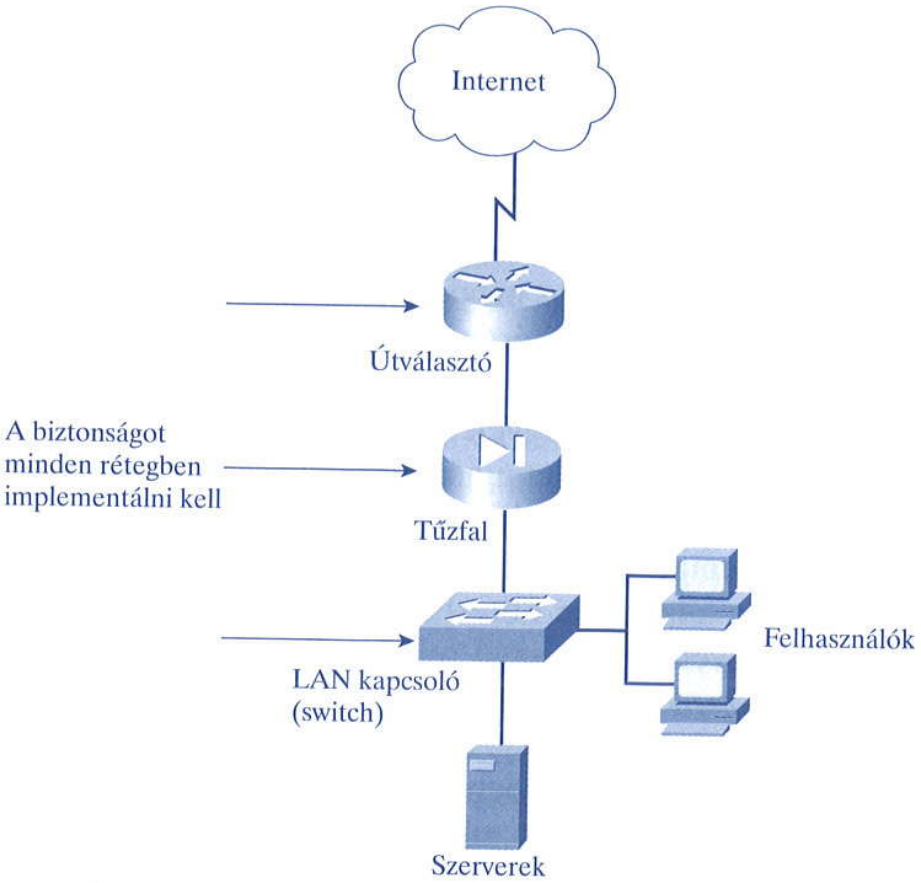
Mind ez ideig a könyv csupán durva ecsetvonásokkal vázolta fel, hogy a támadónak milyen lehetséges lépéseket kell tennie az érzékeny erőforrások elérése érdekében. Az erőforrások védelmére tett legelső lépés a 2. fejezetben tárgyalt számos különféle szempont vegyítésével létrehozott általános biztonságpolitika. Ebben a fejezetben a szélesebb körben használt biztonsági technológiák némelyikét igyekszünk bemutatni. Ezek mindegyike olyan alapelv köré épül, vagy olyan sajátos szerepe van, amely a rétegesen megtervezett és kiépített hálózat biztonságát van hivatva növelni.

## 3.1. A BIZTONSÁG FŐ TERVEZÉSI ELVEI

A hálózati biztonság a hálózat ellen irányuló fenyegetések és lehetséges támadások szempontjából többfejű sárkánynak tekinthető. Ebben a témakörben hihetetlen mennyiségű vélemény született, és hatalmas mennyiségű információ érhető el. Ezek a vélemények rendkívül eltérőek lehetnek attól függően, hogy ki alkotja meg őket. Ha az egyszerű Google keresőnek megadjuk a „biztonságos hálózat tervezése" kulcsszavakat, pontosabban angol megfelelőjüket *(designing a secure network),* akkor csaknem háromnegyed millió találattal tér vissza. Majdnem ugyanennyi könyv is található a témáról, s ha a fenti szavakat egy egységként kívánjuk megadni, még akkor is közel ezer találatot kapunk! Ha magyarul adjuk meg a kulcsszavakat, másfélezernél is több találatot kapunk.

Kijelenthető tehát, hogy a hálózati tervezés bármely területének szakértői olyan mennyiségben halmozták fel a hálózati biztonságról szóló írásokat, hogy ezek egyszerű megítélése is alaposan szétfeszítené e könyv kereteit. A témakörben már megtalálható írások a terület legkisebb részterületét is lefedik. Jelen könyv az olvasójában kialakítható lehető legszilárdabb alapok kiépítésére törekszik a helyénvaló hálózati biztonságtervezési alapelvek szemléltetésével. Egyes fontos tervezési alapelveket azonban az alábbiak figyelembe vételével ismerteti:

- **Többrétegű biztonság** – A biztonság réteges felépítésének alapelve abból a felismerésből származtatható, hogy az egyetlen ponton felállított védelem szükségszerűen kudarcra van kárhoztatva. Amint azt a 3.1. ábra is szemlélteti, a hálózat lehető legtöbb pontján konzisztens módon implementált biztonságot tekinthetjük helyes tervezésnek. A hálózati biztonság ezen rétegződése a jelen fejezet által közvetíteni kívánt legeslegfontosabb üzenet.
- **Hozzáférés ellenőrzése** – A hálózat végső soron az üzemeltető felelősségi körébe tartozik, s így rajta nyugszik annak eldöntése is, hogy mi engedélyezett a hálózaton belül. A legmelegebben ajánlott gyakorlat




3.1. ábra. Biztonsági rétegek

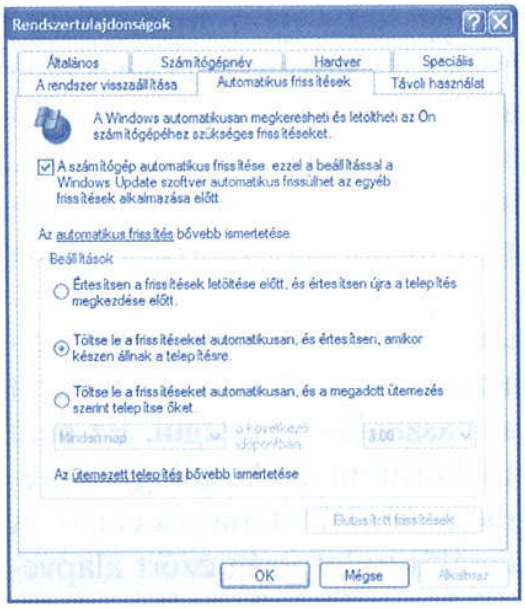
szerint a hozzáférést a „tilts meg mindent, és csak az üzleti tevékenység miatt szükséges néhány dolgot engedd meg külön" elv alapján kell szabályozni. Ezt az alapelvet nevezzük „legkisebb jogosultság elvének" *(POLP – Policy of Least Privilege)*. Például a Cisco-tűzfalak is ezen az alapelven működnek.

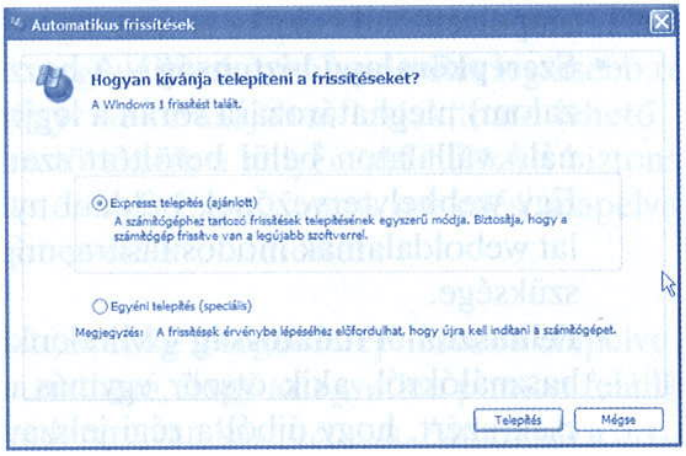
- **Szerepköralapú biztonság** – A hozzáférés és a jogosultság (vagyis a bizalom) meghatározása során a legjobban használható sablon a felhasználó vállalaton belül betöltött szerepének figyelembe vételét tanítja. Egy webhelytervezőnek például nyilván képesnek kell lennie a vállalat weboldalainak módosítására, míg egy titkárnőnek erre nyilván nincs szüksége.
- **Felhasználói tudatosság** – Mindenki hallott már történeteket olyan felhasználókról, akik ötször egymás után megváltoztatták a jelszavukat csak azért, hogy újból a régi jelszavukat használhassák végül. Ez nem azért történik, mert a felhasználók szándékosan meg akarják gyengíteni a biztonságot, csupán arról van szó, hogy nem értik meg a célját. A megfelelő képzéssel kialakítható felhasználói tudatosság ezért alapvető a biztonsági intézkedések fontosságának megértésében. Remek ötlet lehet például a felhasználók képzését annak bemutatásával kezdeni, hogy miért is fontos a fagylaltot teljes körítéssel felszolgálni – az előadó egy csapásra rendkívül népszerű lesz! Elengedhetetlen, hogy a felhasz-

nálók valóban figyelemmel legyenek a biztonságra, és támogassák a biztonsági szabályzatokat.

- **Figyelemmel kísérés** – A biztonság egyik, talán leggyakrabban elmulasztott összetevője a folyamatos megfigyelés. Számos szervezet hiszi elegendőnek a biztonság megteremtését. Elfeledkeznek rendszereik folyamatos figyelemmel kíséréséről, amely hitelt érdemlően bizonyíthatná a biztonság folyamatos meglétét, s hogy nem alanyai egyetlen támadásnak sem. A 9. fejezet tárgyalja azokat a módszereket, nevezetesen a behatolásjelző rendszereket *(IDS – Intrusion Detection System)*, amelyek a folyamatos megfigyelés segítségével felderítik a támadásokat. Forrón ajánlott gyakorlat a hálózatbiztonsági megoldások tervezése során gondoskodni az IDS implementálásáról is.
- **Rendszerek folyamatos frissítése** – A rendszerek folyamatos frissítését vagy javítását az állandóan túlterhelt rendszergazdák gyakran elfelejtik, jóllehet alapvető fontosságú feladat. Szerencsére a legtöbb újabb operációs rendszer képes automatikusan figyelmeztetni az újabb frissítések, javítások megjelenéséről. Például a fordító által is használt Windows XP operációs rendszerben is megtalálható az automatikus frissítés művelete, amely képes az új biztonsági javítások letöltésére és a rendszer frissítésére egyaránt (lásd 3.2. ábra).

  A fontos tehát az, hogy rendszeresen ellenőrizzük rendszereink frissességét, mivel a támadók állandóan igyekeznek felfedni a gyenge pontokat, és betörni a nem frissített rendszerekbe. Az Apple számítógépek OS X („Tiger") operációs rendszerében szintén megtalálható az automatikus frissítés lehetősége.




3.2. ábra. Windows XP automatikus frissítése

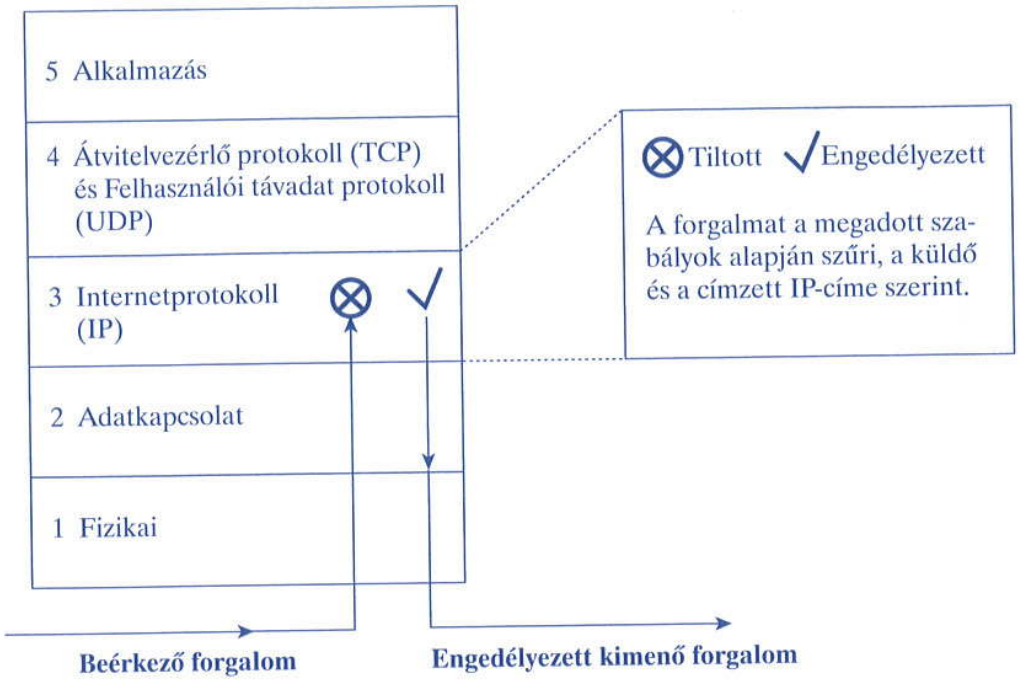
- **Beavatkozó szolgálat** – A biztonsággal kapcsolatos problémák előbb vagy utóbb, de nyilvánvalóan el fognak jutni hozzánk. Lehet, hogy rendszerünk támadás alatt áll, esetleg észrevesszük, hogy a rendszerbe behatoltak vagy már kárt is okoztak. A biztonsági terv beavatkozási képességet taglaló része azzal foglalkozik, hogy a szervezetnek miként kell válaszolnia egy támadásra, és hogyan kell a tapasztaltak alapján megtenni az óvintézkedéseket. A támadásra való reagálás leghatékonyabb módja az, ha megfontoljuk és a tervbe eleve bevesszük egy beavatkozó szolgálat életre hívását, valamint a tervezett válaszlépéseket még „békeidőben", s nem akkor, amikor a támadás miatt már egyébként is erős nyomás alatt állunk, esetleg a helyzet is kezd kicsúszni a kezünkből. A lehetséges helyzetek és a terv papíron történő elemzése segít felfedni azokat a hiányosságokat, amelyek felett a terv készítésekor esetleg elsiklottunk volna.

A biztonsági tervezéssel kapcsolatos, fentebb ismertetett „első lépcsős" megfontolások hozzásegítenek annak megértéséhez, hogy miként kezdhetünk hozzá bármely hálózat biztonságossá tételéhez. Ezzel a következő részben megkezdhetjük a biztonsági technológiák használatával kapcsolatos ismeretek, valamint a hálózat védelmében játszott szerepük bemutatását.

## 3.2. Csomagszűrés a hozzáférés-vezérlő lista segítségével

Amint az közismert, az interneten zajló forgalom a TCP/IP protokollt használja, amely az információt kicsiny részekre, úgynevezett csomagokra tördeli, s így küldi el. Az internet kezdeti időszakában a csomagokra alapuló szűrés elterjedt volt, s még ma is sok esetben számos hálózati útválasztó használja a csomagszűrést. A csomagszűrők más tűzfali technológiákkal kombinálva gyakran alkotják az első védelmi vonalat. Ma a technológia leginkább a hálózatok kapcsolódási pontjain található útválasztókban, a hozzáférés-vezérlő listákkal (ACL) kombinálva lelhető fel.

A **csomagszűrés** egyike a legrégebbi és legelterjedtebb csomagellenőrző technológiáknak. Úgy kezdődik, hogy megvizsgálja a csomag tartalmát, és az előre megírt szabályok alapján a tartalom ismeretében dönti el, hogy továbbengedi-e a csomagot, vagy eldobja. Jóllehet a TCP/IP-csomag fejlécében számos jellemző megtalálható (például protokoll, port stb.), a továbbiakban a 3.3. ábrán látható módon a küldő és a címzett IP-címére alapozott szűrést mutatjuk be.




3.3. ábra. A TCP/IP-modell harmadik rétegében alkalmazott csomagszűrés

A Cisco-útválasztók esetén a csomagszűrő implementálására és konfigurálására használt módszert **hozzáférés-vezérlő lista** *(ACL – Access Control List)* néven ismerjük.[1] Ennek két fő típusa az alap ACL, amelyek a szűrést kizárólag az IP-címek alapján végzik, és a bővített ACL, amelyek megfelelő konfigurálás esetén a csomagok fejlécében található más információkat is figyelembe vesznek.

*Az alap ACL a forrás címe alapján, míg a bővített ACL a forrás és a cél címe alapján képes szűrni. Közöttük elvi különbség nincs, ám az utóbbi további lehetőségekkel is rendelkezik, például képes figyelembe venni a portcímet vagy a protokollt is.*

A csomagszűrők az adat minden egyes csomagját külön ellenőrzik, megvizsgálva a forrás és a cél IP-címét és portcímét. A csomagoknak csak a legalapvetőbb jellemzőit vizsgálják, ezért az adatfolyamot gyorsan képesek továbbítani. Minden egyes ilyen ellenőrző pont eldönti, hogy a cso-

1 Az ACL megnevezést általában szélesebb értelemben, az IT különböző területein használjuk. A csomagszűrésre használt ACL a Cisco-terminológiának felel meg, más gyártótól származó tűzfalak ezt másként (például szabálylista) nevezhetik. *(A lektor megj.)*

magot továbbítja-e, vagy eldobja. Az ACL tehát az a módszer, amelynek segítségével a Cisco-eszközök megvalósítják a csomagszűrést, jelen részben ezért elsősorban ezekre összpontosítjuk figyelmünket.

Mivel minden egyes kapcsolat minden egyes csomagját ellenőrizzük a hozzáférés-vezérlő szabályok szerint, ezért a nagyobb, összetettebb csomagszűrő szabályok alkalmazása lerontja az ezen szabályok alapján ellenőrzést végző eszköz teljesítményét. Ráadásul a csomagszűrők csupán alacsony szintű ellenőrzésre képesek, így a más rétegekben elrejtett rosszindulatú támadásokkal szemben nem védenek.

A hozzáférés-ellenőrző lista sokak számára egyike a nehezen érthető témaköröknek. Amint azt a következő pontban látni fogjuk, a való életből vett példa segítségével sokkal könnyebben érthetjük meg a működését.

## 3.2.1. A BEVÁSÁRLÓLISTA ANALÓGIÁJA

Ez a példa csupán egy lehetséges módszer az ACL segítségével történő csomagszűrés elvének megismertetésére. Érdemes végiggondolni a bevásárlólista hasonlat során bemutatott kulcsfontosságú elveket. A 3.1. táblázatban azzal kezdjük a hasonlatot, hogy összehasonlítjuk az ACL használatával történő csomagszűrést a bevásárlólista összeállításával.

3.1. táblázat. Az ACL-hasonlat áttekintése

| ACL-jellemző | Hasonlóság a bevásárlólistával |
|---|---|
| Az ACL hatékony | A lista követése hatékony és pénzt takarít meg. |
| Felülről lefelé halad | A listán az egyes elemek fontossági sorrendben szerepelnek. |
| Elsőként a tiltások | Vannak a listán nem szereplő cikkek, ezeket ne vegyük meg. |
| Legyen engedélyezett | Mindig van olyan cikk a listán, amelyet meg akarunk venni. |
| Alapból minden tilos | Csak azt szabad megvásárolni, ami szerepel a listán. |

Egyszer vendégeket vártunk vacsorára. Pulykát akartunk készíteni, azonban felfedeztük, hogy a főzéshez még bizonyos hozzávalók hiányoznak. Készítettünk egy bevásárlólistát, így biztosan nem fogok elfelejteni semmit, amikor a boltban vásárolok. Egyes hozzávalók már megvannak, így ezeket nyilván nem írjuk fel a listára:

- pulyka,
- töltelék,
- kenyér,
- sajt.

Más szavakkal, ezeket a hozzávalókat nem vásárolhatom meg, mert a feleségem azt mondja, hogy ezek már megvannak. Amikor összeállítjuk a megvásárolandó cikkek listáját, egy meglehetősen általános listánk lesz. Ez azonban nem baj, ennyi bőven elegendő ahhoz, hogy elinduljak vele a boltba és megvegyem a hozzávalókat:

- tej,
- pástétom,
- krumpli,
- mártás,
- mást ne vegyél!!

A lista azért általános, mert sok különböző tej, és számos különféle mártás vagy pástétom létezik. Mivel szükség van ezekre a hozzávalókra, így ezeket meg szabad venni. Ez a bevásárló lista hasonlít az alábbi alap *(standard)* ACL-listához:

```
[alap ACL] Egy szokásos bevásárlólista
[tiltott] Pulyka
[tiltott] Töltelék
[tiltott] Kenyér
[tiltott] Sajt
[szabad] Tej
[szabad] Pástétom
[szabad] Krumpli
[szabad] Mártás
[fenntartás nélkül tiltott minden más] mást ne vegyél!!
```

Figyeljünk az utolsó sorra! A feleségem azért írta oda, mert tudja rólam, hogy nagyon szeretem a fagylaltot. Ma már nem feltétlenül kellene odaírnia, mert 16 évi házasság után már magamtól is tudom, hogy semmi mást nem szabad vennem.

A biztonság kedvéért elhatároztam, hogy megmutatom a listát a feleségemnek, nehogy valami kimaradjon. Miközben átnézte a listát, úgy vélte, inkább kiegészíti azt, hogy biztosan a „megfelelő" árucikket vegyem meg:

```
[bővített ACL] Feleségem által kiegészített bevásárlólista
[tiltott] Pulyka
[tiltott] Töltelék
[tiltott] Kenyér
[tiltott] Sajt
[szabad] Tej - 3,6%-os
```

```
[szabad] Pástétom - Mrs. Smiths Pumpkin féle
[tiltott] rózsakrumpli, mert a vendég utálja
[szabad] minden más krumpli, ha nem rózsakrumpli
[szabad] Mártás - White Country
[fenntartás nélkül tiltott minden más] mást ne vegyél!!
```

A lista a kiegészítésekkel sokkal pontosabban képes szűrni, illetve a vásárlás esetén a biztosan szükséges hozzávalókkal fogok hazatérni a segítségével. A két lista közötti eltérés nyilvánvaló, az első meglehetősen általános, egyáltalán nincsenek benne különleges utasítások, míg a második lista nagyon pontos volt, s nem csupán azt közölte, hogy mit vehetek meg, hanem azt is, hogy azon belül pontosan melyik fajtát. Végezetül persze mindkét lista úgy végződik, hogy semmi mást nem szabad vásárolni, csak a kifejezetten felsorolt cikkeket. A házasemberek biztosan tisztában vannak azzal, néha milyen kihívást jelent pontosan a feleség által kívánt holmik megvásárlása, s most nyilván azon tűnődnek, miként is kapcsolódik mindez az ACL-hez és a csomagszűréshez.

A csomagoknak vannak azonosítható jellemzőik, amelyek alapján a hozzáférés-vezérlő lista segítségével osztályokba sorolhatók, és így valamelyik művelet (a tiltás vagy az engedélyezés) végrehajtható. Figyeljük meg a 3.1. példát, amely azt mutatja be, hogy a fenti példán alapuló alap ACL-lista miként is nézne ki.

3.1. példa. Alap ACL-listaként felírt bevásárlólista

```
access-list 10 deny any pulyka
access-list 10 deny any töltelék
access-list 10 deny any kenyér
access-list 10 deny any sajt
access-list 10 permit any tej
access-list 10 permit any pástétom
access-list 10 permit any krumpli
access-list 10 permit any mártás
```

A Cisco-eszközökben az alap ACL-listát elsősorban arra használjuk, hogy IP-cím alapú szűrést végezhessünk velük. A számozás az alap ACL-listát azonosítja, ennek során az 1–99 és az 1300–1399 közötti számokat használhatjuk. Ha egy szinttel mélyebbre akarunk hatolni a technikai megvalósításba, és valós IP-címeket és alhálózati címeket akarunk használni, akkor a fenti lista egy Cisco-eszköz konfigurációs listájában a 3.2. példán látható formát öltené.

3.2. példa. Csomagszűrő alap ACL-lista

```
access-list 10 permit any 192.168.10.0
access-list 10 permit any 192.168.20.0
access-list 10 permit any 192.168.30.0
access-list 10 permit any 192.168.40.0
```

Valószínűleg feltűnt, hogy a tiltások nem szerepelnek a példában. Ennek oka az, hogy Cisco-eszközök esetén a külön nem engedélyezett csomagok automatikusan tiltásra kerülnek. Ezért nincs szükség a tiltások külön megadására. Az alap ACL-listát átalakíthatjuk bővített listává is, még pontosabban leírva ezáltal a teendőket. Ezt tette a feleségem is, amikor kiegészítette a listámat a még pontosabb utasításokkal.

Mivel a csomagok azonosítására tervezték őket, így a hozzáférés-vezérlő listák a hálózatok világában számos szerep betöltésére képesek. Miután sikerült egy csomagot azonosítani, valamilyen módon kezelhetővé válik. Ez a művelet lehet a csomag szűrése, de lehet egy másik, fontosabb csomag utánküldése is. A 3.4. ábra mutatja be, hol helyezkedik el az ACL a hálózati útvonalon.

3.4. ábra. A csomagszűrők helye a hálózaton

## 3.2.2. A CSOMAGSZŰRÉS KORLÁTAI

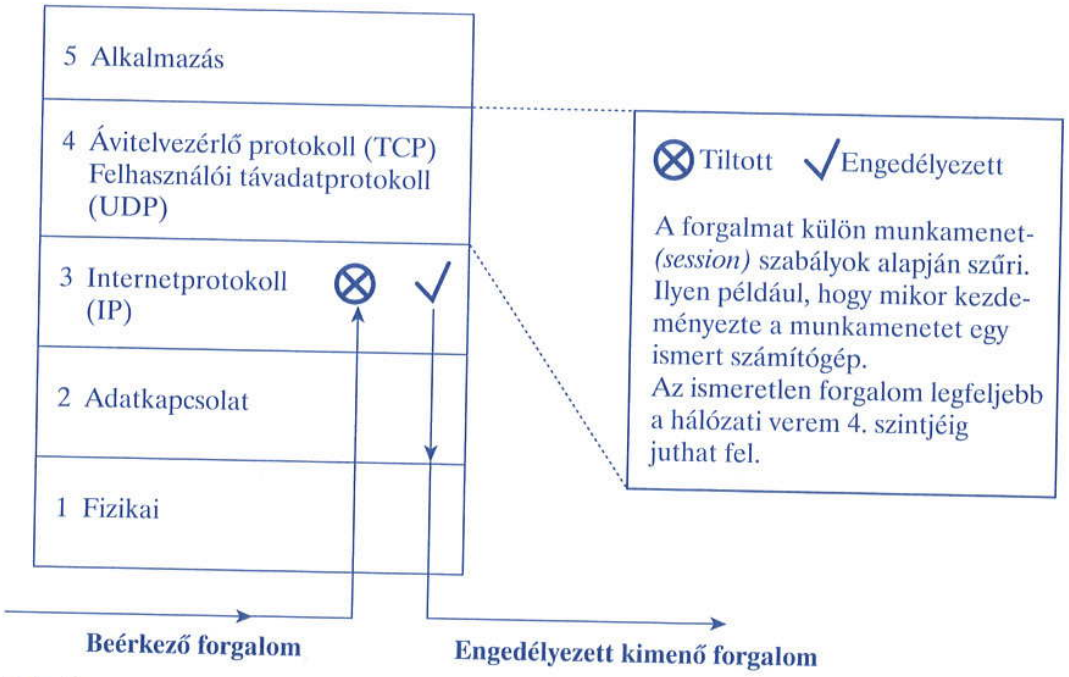
Elérkezett az ideje annak, hogy a csomagszűrés hátrányairól is szót ejtsünk. Használatukkal nyilván sok támadást meg lehet akadályozni. Tegyük fel, hogy van egy webszerverünk a demilitarizált zónában (DMZ). Nyilván minden HTTP- (web-) forgalomnak el kell érnie ezt a szervert. Fusson a szerveren mondjuk a Microsoft IIS webszerver szoftvere, egy támadó pedig határozzon úgy, hogy a HTTP-forgalmat használja a szerver megtámadására. Mivel a támadás csupán az IIS sebezhető pontjait célozza, szabályos forgalomnak álcázva azokat, így a csomagok keresztül-

jutnak a szűrőn. A csomagszűrés tehát önmagában még kevés biztonságot nyújt, így bizonyára szükségünk lesz olyan technológiákra is, amelyek erősítik a védelmet.

## 3.3. ÁLLAPOTTELJES CSOMAGVIZSGÁLAT

Ebben a részben a csomagok ellenőrzésének sokkal hatékonyabb technikáját, az állapotteljes csomagvizsgálatot *(SPI – stateful packet inspection)* ismerhetjük meg. Működésének megértéséhez azonban mindenekelőtt röviden át kell tekintenünk a TCP/IP-modellt.

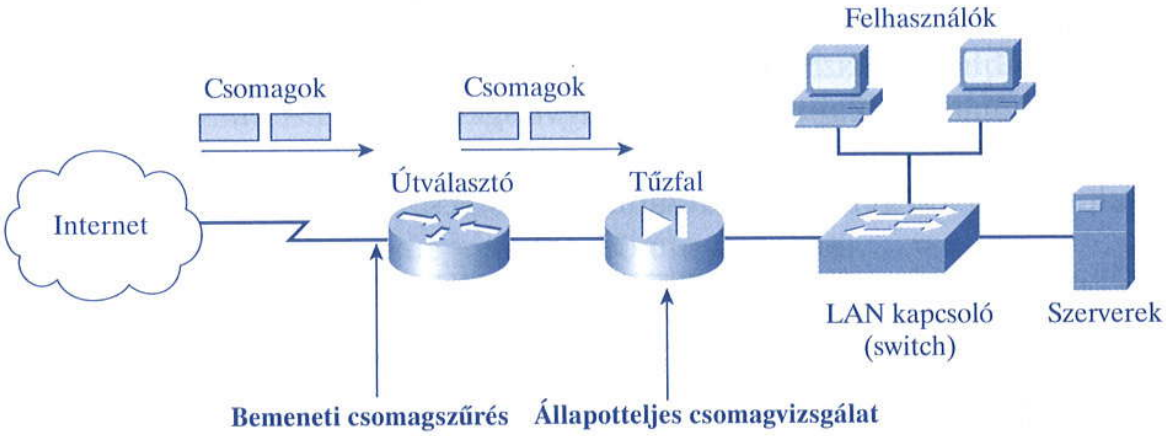
*Sokakat megzavar az OSI hivatkozási és a TCP/IP-modell közötti kapcsolat. Annyiban egyformák, hogy mindkettő réteges felépítésű, és az alsóbb rétegek mindig a felsőbbek számára nyújtanak szolgáltatásokat. Jelen könyvnek nem célja a különbségek bemutatása. Mivel a biztonsággal kapcsolatos témákat nem elvi megközelítésben, hanem gyakorlati példákon keresztül kívánjuk tárgyalni, így a továbbiakban a ténylegesen használt TCP/IP-modellt fogjuk alkalmazni.*




3.5. ábra. A TCP/IP-modell

A 3.5. ábrán a **TCP/IP-modell** öt rétege látható. Az állapotteljes vizsgálatot végző összetevő azt vizsgálja, hogy a TCP- (4. szállítási) réteg miként követi a kapcsolatokat. A TCP-kapcsolatok nyomon követését a TCP/IP-modell 4. rétegében végezzük.

A legtöbb esetben az állapotteljes csomagvizsgálat a tűzfalban történik, amely közvetlenül a hálózatunkat az internethez kötő útválasztó után található. Amennyiben az útválasztón első védelmi vonalként beüzemeltük az IP-csomagszűrést, akkor a második védelmi vonalunk a tűzfalon végzett SPI lesz, amint azt a 3.6. ábra is mutatja.




3.6. ábra. Az állapotteljes csomagszűrés helye

Ez az elhelyezkedés, és az így megnövelt biztonság teszi lehetővé a mélységi védelmi rétegek következő szintjének kiépítését azzal a céllal, hogy a többszörös rétegződésű védelemmel minél biztonságosabbá tegyük a hálózatunkat.

Az SPI általában a tűzfalakon kerül beüzemelésre, így a TCP/IP-kapcsolatokat tüzetesebben is ellenőrizhetjük. Ezt a technológiát ezért kapcsolatfigyelőnek tartjuk, mivel az SPI képes megfigyelni és követni a két számítógép között kiépített kapcsolat során általában mindkét irányban több csomagból álló adatfolyamokat. A csomagok állapotteljes vizsgálata a kapcsolatot kezdeményező első csomaggal kezdődik. Amikor a kapcsolatot megvizsgálja, egy táblába bejegyzi azt. Amikor a további csomagok megérkeznek, akkor ebben a táblában vizsgálja meg, hogy ezek egy már bejegyzett, létező kapcsolat részei-e. Amennyiben a csomagok átmennek ezen a vizsgálaton, akkor engedélyezett a továbbhaladásuk. Ezzel nagyjából ismertettük az SPI működését, a következő pontban azonban egy kicsit részletesebben is megvizsgáljuk azt.

### 3.3.1. Az SPI használatával kezelt bővített csomagfolyam

Mivel ebben a könyvben a hálózati biztonsággal és a hozzá tartozó technológiákkal kapcsolatban ragaszkodunk a legcélszerűbb módszerek ismertetéséhez, így ezen részletesebb ismertetőben foglaltakat is arra ala-

pozzuk, hogy a hálózatban megtalálható a külső útválasztó, amely a hálózatba irányuló forgalmat az IP-csomagszűrő technológia segítségével már előszűri. Így tehát a hálózatba érkező csomagok a csomagszűrésen már átjutottak, majd az alábbiak szerint megérkeznek a tűzfalra:

1. Amikor a csomag megérkezik a tűzfalhoz, el kell dönteni a sorsát – megengedett-e a belső hálózatba való továbbítása, vagy sem.
2. Az állapotteljes csomagvizsgálatot végrehajtó eszköz minden egyes csomagot megfog, és megvizsgálja a fejlécét, eldöntendő, hogy az megfelel-e az előre megadott, engedélyezett csomagokat leíró szabályoknak.
3. A csomag fejlécének ellenőrzése során a vizsgálat kiterjed a csomag feladójának és címzettjének az IP-címére, a protokoll (TCP, UDP, ICMP stb.) típusára, a forrás és cél végpontok (portok) címére, a csomaghoz beállított jelzők (SYN, ACK, FIN, RST stb.) értékére, és más alapvető fejléc-információkra. A beérkező csomagokat addig vizsgálja, ameddig a kapott csomagról elegendő információ gyűlik össze (felhasználva például a TCP-sorszámot és más hasonló információkat is) ahhoz, hogy meghatározható legyen az „állapota".
4. Az ellenőrzés során gyűjtött információkat összehasonlítja az engedélyezést és tiltást leíró szabályokkal. Előírható például, hogy a webszervernek címzett csomagok közül kizárólag a HTTP-forgalmat lebonyolítók kerüljenek továbbításra, és minden más oda irányuló forgalom legyen letiltva. Ez egyike a leggyakoribb szabályoknak, hogy egy adott szerverre kizárólag bizonyos típusú forgalom irányulhat, minden más típusú pedig tiltott.
5. A csatlakozás állapotától függően az ellenőrzés során gyűjtött információt ezután összehasonlítja az eszköz által engedélyezett valamennyi kapcsolatról információt tároló állapottáblával. A legtöbb eszköz például megengedi, hogy a hálózaton belül keletkező forgalom bármit elérhessen a külvilágban, így ez a kapcsolat bejegyzésre kerül az állapottáblában. Ettől kezdve azonban már nem engedhető ki valamennyi, a beállított követelménynek megfelelő csomag, csak a már megnyitott kapcsolat részét képező csomagok.
6. Végezetül vagy megadja az engedélyt a csomag továbbítására, vagy megtiltja azt, az eddigi ellenőrzési lépésektől függően. Mivel a szabályokat és a táblákat csak egyszer nézi végig, így az ellenőrzésnek nincs túlságosan nagy kihatása a teljesítményre.

Az SPI-szabályokat a megnövelt bonyolultság miatt nem annyira egyszerű létrehozni, mint a csomagszűrésnél használtakat. Mindazonáltal használatuk bőségesen megéri a költségeket és erőfeszítéseket, hiszen a hálózathoz így egy újabb védelmi réteget kapcsolhatunk. Működése ráadásul gyors, és nagyobb mennyiségű forgalmat is gond nélkül képes kezelni. Amennyiben a kapcsolathoz feljegyzett paraméterek nem egyeznek meg az állapottáblában tárolt értékekkel, a kapcsolatot eldobja.

*Jóllehet általában a tűzfalakat választjuk az állapotteljes csomagvizsgálatot végző eszköznek, ezt a szerepet azonban az útválasztók is betölthetik. Ez azonban mégsem tanácsos azért, mert ha összekeverjük a hálózati eszközök szerepköreit, akkor megváltoztatjuk azt a funkciójukat, amire tervezték. Lehet ugyan azzal érvelni, hogy sikeresen kombinálhatók a szerepkörök és az eszközök. Előfordulhat, hogy a távolabbi jövőben ez igaz lesz, manapság azonban, különösen azon hálózatok esetén, amelyek biztonságáért nekem kell viselnem a felelősséget, ezt mégsem javasoljuk.*

## 3.3.2. Az állapotteljes csomagvizsgálat korlátai

Jóllehet az SPI-eszközök jobban méretezhetők, és számos más előnyük is van a csomagszűrőkkel szemben, mégsem alkothatják a hálózat védelmének végső vonalát. Vizsgáljuk meg az állapotteljes csomagvizsgálat két fő hátrányát:

- **Nincs alkalmazásszintű ellenőrzés** – Az SPI-technológia nem képes a csomagokat a TCP/IP-modell negyedik rétegénél följebb vizsgálni. A gyakorlatban az állapotteljes csomagszűrést alkalmazó, tűzfallal védett, de valamilyen hozzáférést engedélyező szervereket a kalózok ezt kihasználva támadják.
- **Nem minden TCP/IP protokollnak van kapcsolati állapota** – A TCP/IP-modell egyes protokolljai nem teszik lehetővé a számítógépek közötti kapcsolat állapotának követését. Ilyen például az UDP és az ICMP egyaránt. A mélységi védelemnek ezért ezen protokollok esetén csak az első, csomagszűrő vonala működik, hiszen nincs nyomon követhető állapotuk.

Ebben a részben a tűzfalak és hasonló hálózati eszközök a kapcsolat állapotának, s így érvényességének felismerési képességéről volt szó, amelynek alapján eldönthető, hogy a vizsgált csomag beengedhető-e a védett belső hálózatba. A következő részben a hálózatba érkező csomagok érvényességének megállapítását lehetővé tevő további lehetőségeket vizsgálunk meg a TCP/IP-modell 4. (szállítási) rétegének szintjén, tovább növelve ezáltal a védett hálózat biztonságát.

## 3.4. A HÁLÓZATICÍM-FORDÍTÁS

Az internet sokkal nagyobbra nőtt, mint azt bárki is megjósolhatta volna. Jóllehet pontos mérete ismeretlen, az ezredfordulón mintegy 100 millió gép és 350 milliónál is több felhasználó használta aktívan az internetet, a legfrissebb adatok szerint pedig 180 milliónál több gépen legalább 800 millió ember csatlakozik hozzá. A lélekszámot tekintve ez már 2000-ben is több volt, mint az Egyesült Államok népessége, és mintegy háromnegyede a bővítés utáni Európai Unió lakosainak, ma pedig már több, mint e két régió lakossága összesen. A fenti számokból az is látható, hogy az internet gyors ütemben növekszik.

Amikor az IPv4 címzést bevezették, mindenki azt hitte, hogy az bőven fedezni fogja a szükségleteket. Elméletileg $2^{32}$, vagyis 4 294 967 296 nyilvános cím létezik. A ténylegesen használható címek száma ennél kevesebb (valahol 3,2 és 3,3 milliárd között). Ennek oka egyrészt maga az a mód, ahogy a címek kiosztásáért felelős Internet Mérnöki Feladatok Szolgálat *(IETF – Internet Engineering Task Force)* szervezet elérhetővé teszi a címeket (A, B és C osztályú címzések), másrészt a többesküldésű üzenetekhez, teszteléshez és más különleges feladatokra használt címek sem használhatók más célra (D osztályú címek).

*Az IPv4 címek osztályba sorolása ellenére még mindig többmilliónyi nyilvános IP-cím „tűnt el". Annak érdekében, hogy a saját IP-címeket igénylő hálózatok ténylegesen kiszolgálhatók legyenek, az IETF sok címtartományt fenntart a hálózaticím-fordítás néven ismert technológiával kiosztott belső hálózati címek számára. Ezen címek közül sokat* **magán IP-cím** (private IP address) *néven emlegetünk, mivel ezek közvetlenül nem érhetők el az interneten (nyilván ezért nevezzük őket „magán" vagy „privát" IP-címeknek). A magáncímeket bármely intézmény használhatja a saját hálózatában, de soha nem a nyilvánosan elérhető interneten. Az interneten elérhető címeket ezért nevezzük a saját cím ellentéteként* **nyilvános címek***nek.*

Az internet egyre gyorsuló növekedése és az otthoni és vállalati hálózatok egyre nagyobbá váló IP-cím igénye miatt az elérhető nyilvános IPv4 címek száma egyszerűen nem elegendő. A nyilvánvaló megoldás az újratervezés egy nagyobb számú lehetséges IP-címet biztosító címzési séma bevezetése megoldaná ezeket a problémákat. IPv6 néven[2] már zajlanak is

2 A jelenlegi IP-szabvány verziószáma 4, a következő generációs protokoll verziószáma pedig 6. Ebből ered az IPv4, illetve IPv6 elnevezés. *(A lektor megj.)*

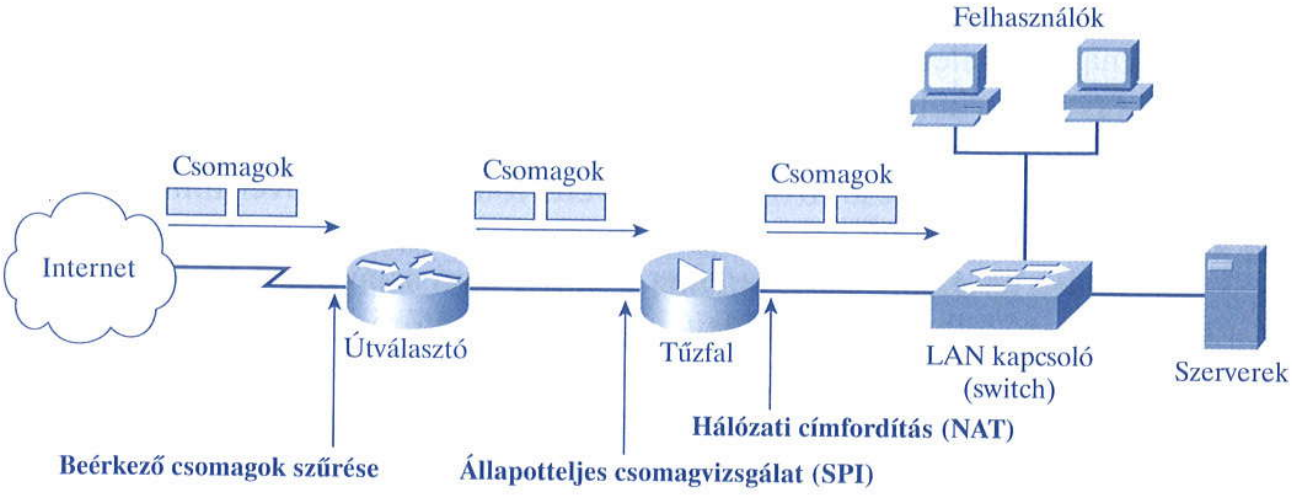
az ezzel kapcsolatos erőfeszítések, valójában a módszert már teljesen ki is dolgozták, bevezetése mégis jó esetben évekig fog tartani, hiszen ehhez az internet teljes infrastruktúráját módosítani kell. Ennek következtében az IPv4 címzésről IPv6 címzésre való átállás jelenleg nagyon lassan halad, és várhatóan továbbra is lassú marad, miközben a hálózaticím-fordítás *(NAT – network address translation)* technológia ráadásul tovább növeli az IPv4 címzés élettartamát.

A NAT lehetővé teszi a már létező címtartománnyal és hálózattal rendelkező szervezetek számára az IP-címek iránti igényük kielégítését. A Hálózati Információs Központnál *(NIC – Network Information Center)* bejegyzett IP-címmel nem rendelkező szervezeteknek elsőként az Internet Számkiosztó Szervezetnél *(IANA – Internet Assigned Numbers Authority)*, vagy az amerikai internetcím nyilvántartónál *(ARIN – American Registry for Internet Numbers)* saját címtartományt kell igényelniük (mindkettő nagy örömmel okoz némi bürokratikus késedelmet). Európában ugyanezt a tevékenységet a RIPE (Réseaux IP Européens, Európai IP-hálózat) nevű szervezet végzi, lásd http://www.ripe.net. Számos hely nem tudja átverekedni magát az általuk támasztott barátságtalan bürokratikus útvesztőn, s nem kapja meg az elegendő számú címet, így a legtöbb szervezetnek a NAT jelenti az egyetlen megoldást.

*Az IANA a saját hálózatok számára az alábbi címtartományokat tartja fenn:*
*10.0.0.0 – 10.255.255.255 (vagyis a 10.0.0.0/8 mintájú címek),*
*172.16.0.0 – 172.31.255.255 (vagyis 172.16.0.0/12 mintájúak),*
*192.168.0.0 – 192.168.255.255 (vagyis 192.168.0.0/16 mintájúak).*

A *nyilvános IP-címek a külső hálózaton* (vagyis az internethez közvetlenül kapcsolódó eszközökön) használhatók. Amint azt már említettük, ilyen címből valószínűleg nincs elegendő valamennyi hálózati nyomtató, munkaállomás, szerver, útválasztó, tűzfal, drót nélküli eszköz stb. számára. Ezen eszközöknek is szükségük van azonban IP-címre ahhoz, hogy a TCP/IP-modell szerinti kapcsolat kiépíthető legyen velük. Ilyenkor használunk *saját vagy magán IP-címeket* a belső (az internetről közvetlenül nem elérhető) hálózatunkon. Ezzel azt a célunkat elértük, hogy minden eszköznek legyen saját IP-címe, és így képes legyen a (belső) hálózaton kommunikálni, ezek a saját IP-címek azonban nem érhetők el az internetről, és ami még rosszabb, maguk sem érhetik el azt. Ezért van szükségünk a hálózaticím-fordításra.

A **hálózaticím-fordítás (NAT)** technológiát olyan eszközön (tűzfalon, útválasztón vagy számítógépen) üzemeljük be, amely a saját IP-címeket használó belső hálózat, és a nyilvános IP-címeket használó internet között helyezkedik el. A sajátról nyilvános címre történő **címfordítást** végző eszköz általában a tűzfal, vagy ritkábban az útválasztó. Ennek az eszköznek bizonyos hálózati interfészei a belső, más hálózati interfészei pedig a külső hálózathoz – például az internethez – csatlakoznak. A 3.7. ábrán láthatjuk a hálózaticím-fordításnak a többrétegű hálózati védelemben elfoglalt helyét.




3.7. ábra. A NAT elhelyezkedése a hálózaton

A hálózaticím-fordítás hálózatbiztonságot megnövelő lehetőségeit kicsit később, a következő (hálózatbiztonság növelése) pont tárgyalja. A címfordítás többféleképpen is működhet:

- **Statikus címfordítás** – A saját IP-címek nyilvános IP-címre fordítását egy-egyes hozzárendelés alapján végzi. Ez különösen akkor hasznos, ha a belső hálózaton elhelyezkedő eszközt a külső hálózaton is el kell érni. Ha a saját webszerverünknek a belső (saját) címe mondjuk 10.0.0.1, azonban az internetről is el kell tudni érni azt (végül is ezen a szerveren van a külvilág számára is elérhető webhelyünk!), akkor a NAT-ot statikus címfordításra kell beállítani, hogy a nyilvános címre hivatkozó külső megkereséseket mindig a 10.0.0.1 címre fordítsa le. A statikus címfordítást az internetről elérni kívánt szerverek (például webszerver) esetén alkalmazzuk a leggyakrabban.
- **Dinamikus címfordítás** – A saját IP-címeket a nyilvános címek egy adott készletéből kiválasztott címre fordítja. A címfordítás ebben az esetben is egy-egyes jellegű, minden egyes saját címhez pontosan egy, a többitől különböző nyilvános cím társul. Ha a saját gép mondjuk a 10.0.0.2, egy munkatárs gépe pedig a 10.0.0.3 címmel rendelkezik, akkor ezen gépek az internetre való kijutáskor, vagyis a tűzfalon való áthaladáskor saját,

csak hozzájuk rendelt nyilvános IP-címet kapnak a kapcsolat idejére. A dinamikus címfordítás sokat segíthet, számos esetben azonban mégsem jelentheti a jó megoldást. Mi történjék például akkor, ha a munkatárs gépe ki akar jutni az internetre, de a tűzfalon elfogytak a kiadható nyilvános címek? Kijutási kérelmét időlegesen ugyan, de nyilvánvalóan el kell utasítani. Ez nyilván komoly problémákhoz vezet, így kidolgozták a túlterheléses címfordítást.

- **Túlterheléses címfordítás** – A dinamikus címfordítás egy olyan típusa, amikor több saját IP-címet ugyanazon nyilvános címre fordít úgy, hogy mindegyikhez más kimenő TCP- (vagy UDP-) portszámot rendel, majd az ellenirányú forgalomban a címeket visszafordítja. Szokás ezt egycímű címfordításnak, vagy végpontcím címfordításnak *(PAT – port address translation)* is nevezni. Lényegtelen, hogy milyen néven emlegetjük, működése azonban alapvető fontosságú. Mivel minden egyes IP-címhez 64 000-nél több TCP-port is tartozhat, így a címfordítás ezen típusa hatékony eszköze lehet a nagyszámú saját belső IP-címmel rendelkező felhasználó egyidejű internet-hozzáférésének biztosításához. A leggyakrabban ezt a fajta címfordítást használják, mivel egyedül ez képes nagyszámú felhasználó egyidejű kiszolgálására.

## 3.4.1. A HÁLÓZATBIZTONSÁG NÖVELÉSE

A címfordítás kidolgozását elsősorban az IPv4 címek kiosztási nehézségei ösztönözték ugyan, azonban a technológia egy újabb védelmi vonalat is jelenthet a hálózatunkban. A címfordítás használata ugyanis a támadó számára általában nehezebbé teszi:

- a hálózati topológia felmérését és a csatlakozások meghatározását;
- a hálózatra csatlakozó rendszerek számának felismerését;
- a gépek és operációs rendszereik típusainak azonosítását;
- bizonyos típusú szolgáltatásmegtagadási támadások végrehajtását, mint például a SYN-csomagokkal való elöntés, a portletapogatás, vagy a csomagbeszúrás. (A fordító megjegyzése: *A túlterheléses címfordítás fontos jellemzője, hogy a védett hálózaton található gépeket kívülről nem lehet megcímezni, mert kizárólag a belülről kezdeményezett kapcsolatok idején van külső elérhetőségük [amit az állapotteljes szűréssel viszonylag hatékonyan lehet védeni]. Teljes biztonságot azonban ez sem jelenthet, csupán egy újabb védelmi vonalat ad. Bizonyos trójai programok például [amelyek már bejutottak, mondjuk egy e-levél csatolmányaként] megnyitnak kifelé egy kapcsolatot, s ezen keresztül a trójai által megcímzett gép, végső soron a támadó hozzáférhet a belső géphez.*)

## 3.4.2. A CÍMFORDÍTÁS KORLÁTAI

Az internet és a hálózat területén a címfordítás bevezetése nyilvánvalóan megoldotta, vagy legalábbis időben kitolta a címkiosztási gondokat. Sokan még azt is megkérdőjelezik, hogy egyáltalán be lesz-e vezetve valaha az IPv6, ha egyszer a címfordítás ennyire jól működik. A kérdés azonban valójában nem ez, csupán a pontos időpontját nehéz megjósolni. Jelenleg az ázsiai és csendes-óceáni térség vezeti az IPv6 bevezetési statisztikáit, beüzemelve azt számos már működő hálózaton.

A kapcsolatok száma egyre növekszik, így a további IP-címek iránti igény is egyre nagyobb lesz. Éppen ezért mindenképpen át kell majd térni az IPv6 címzési módra, a NAT csupán késlelteti az elkerülhetetlent.[3] A címfordítás hasznos, és új előnyökkel is jár, azonban vannak korlátai:

- **UDP-problémák** – A címfordítás a kapcsolatok állapotát is nyomon követi, s mint arról már volt szó, az UDP-kapcsolatok esetén eredendően nincs lehetőség az állapot meghatározására. Éppen ezért a címfordítónak nincs módja annak megállapítására, hogy egy UDP-csomag egy már zajló kommunikáció része, vagy csupán egy új kapcsolat kiépítési kísérlete. A csomagfordító eszköznek ezért valamilyen döntést kell hoznia arról, hogy az utolsó UDP-csomag elküldése után még mennyi ideig tartsa nyitva az UDP-portot, ezt nevezzük üresjárati időtartamnak *(idle time)*. A Cisco-tűzfalak esetén beállítható az üresjárati időtartam felhasználó által kívánt hossza.
- **Érzékeny protokollok** – Bizonyos protokoll elrejti, megváltoztatja, vagy másként elkendőzi a csomag olyan jellemzőjét, amelyre a címfordítás helyes végrehajtásához szükség lenne. Így például a Kerberos, X-Window, távoli parancsértelmező, és a munkamenet-kezdeményező protokoll *(SIP – session initiation protocol)* NAT-eszközön keresztül történő továbbítása nehézségekbe ütközhet. A gondot a bizonyos alkalmazások által a csomagok adatrészébe ágyazott IP-címek jelentik. A Cisco-tűzfalakhoz léteznek olyan szoftverkiegészítések, amelyek az ilyen alkalmazások használta protokollnak megfelelő javítás elvégzésére képesek.
- **Ütközések a titkosító és hitelesítő rendszerekkel** – Számos adattitkosító rendszer megkísérli a csomagok sértetlenségének ellenőrzését, így bizonyosodva meg arról, hogy a továbbítás során a csomagok nem lettek megváltoztatva. A NAT viszont tervezési elvénél fogva természetesen megváltoztatja a csomagokat, ami azt eredményezi, hogy a titkosító

3 A NAT bizonyos esetekben nem jó, és sok felhasználó esetén szűk keresztmetszetet jelent. Nem képes tehát teljes egészében helyettesíteni az IPv6 jelentette elegendő címet. *(A lektor megj.)*

és hitelesítő rendszerek alaphelyzetben nem működnek túlságosan jól együtt a címfordítással.[4]

- **Bonyolult naplózás** – Amikor egyes eszközök más eszközökön keresztül próbálnak naplózni, a naplók közötti kapcsolatok megértéséhez a felhasználónak a NAT által végzett változtatások (címfordítások) pontos ismeretére van szüksége. A rendszernaplók NAT-rendszerrel való megfeleltetése ezért meglehetősen bonyolulttá válik, és igen nehezen lehet kikövetkeztetni, hogy pontosan mely belső rendszerek voltak érintettek a forgalomban.
- **Uniformizálás** – Ha a PAT típusú címfordítással védett belső hálózat egyik felhasználója hozzáférést szerez egy külsőhálózat-beli erőforráshoz, akkor előfordulhat, hogy a belső hálózat összes többi felhasználója is megkapja a hozzáférést. Ne feledjük, a PAT esetén valamennyi saját belső cím kívül ugyanazon egyetlen címnek látszik, csupán más portokon keresztül éri el a külső hálózatot. A hitelesítést igénylő védett külső erőforrás ezért a belső hálózat felől érkező forgalmat ugyanazon IP-címről érkezőnek látja, és így engedélyezettnek tekintheti.[5]

Végezetül még egyszer szögezzük le, a NAT számos szempontból rendkívül hasznos, az egész vállalat számára lehetővé teszi a védelem egy újabb rétegének kiépítését, amint azt a 3.7. ábrán korábban már láthattuk.

A következő részben azt vizsgáljuk meg, miként mélyíthető tovább a védelem olyan eszközök és technikák segítségével, amelyek a TCP-csomagot részletesebben is megvizsgálják.

## 3.5. A KÖZVETÍTŐK ÉS AZ ALKALMAZÁSSZINTŰ VÉDELEM

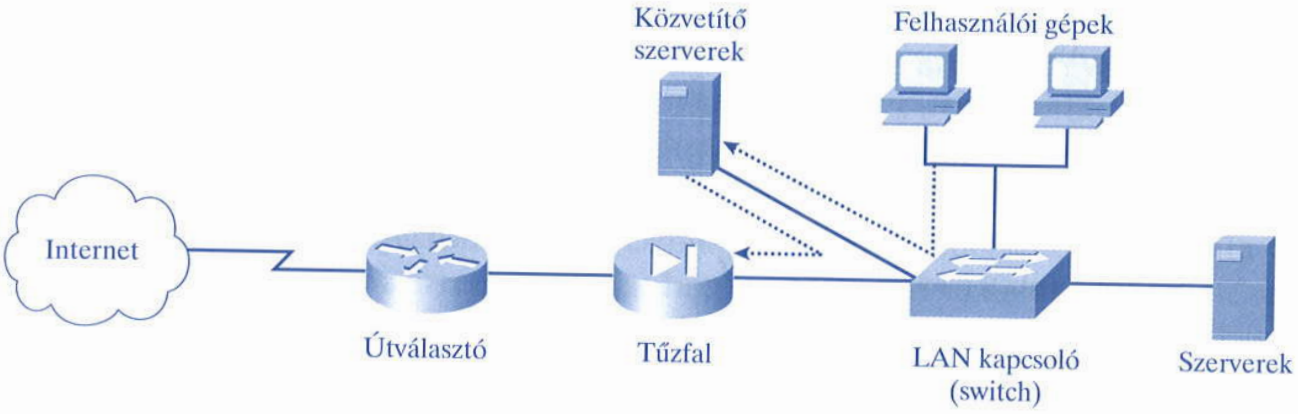
Az állapotteljes csomagvizsgálatot végző tűzfalak csupán a csomagszűrő javított változatai. A most ismertetésre kerülő eszközök ehhez még további intelligenciát adnak úgy, hogy a csomagokat a TCP/IP-modell 5., alkalmazási rétegében vizsgálják. Ilyen jellegű védelemre több különbö-

4 Amennyiben a titkosító vagy hitelesítő rendszer csupán az adatok sértetlenségét ellenőrzi, és nem kísérli meg a csomagok sértetlenségének biztosítását, illetve másként megfogalmazva, amennyiben az 5. rétegben elhelyezkedő alkalmazás nem próbálja meg a 4. réteg bizonyos jellemzőit (végső soron illegálisan) kihasználni, akkor ez a probléma nem merülhet fel, hiszen a címfordítás a 4. rétegben történik, arról az 5. rétegbeli alkalmazás már mit sem tud.

5 Meglehetősen sok kívánnivalót hagyna maga mögött annak a rendszernek a biztonsága, amely a hozzáférés engedélyezését kizárólag az IP-cím alapján végezné. Előfordulnak azonban egyes webszolgáltatások (például webes szavazás, fórumok stb.), amelyek az IP-cím alapján korlátozzák a többszörös hozzáférést. Ilyenkor problémát jelenthet, hogy több felhasználó ugyanazon a címen látszik. *(A lektor megj.)*

ző technológia használható, amelyeket számos elnevezéssel illethetünk. Jóllehet mindegyik ilyen technológia a többitől kismértékben különböző módon működik, céljuk azonban ugyanaz: a hálózat biztonságának további növelése.

Az **alkalmazásszintű tűzfalak** jelentik az adatkapcsolatok legbiztonságosabb fajtáját, hiszen az adatcsere folyamatában képesek az adatokat a TCP/IP-modell valamennyi szintjén ellenőrizni. A védelem ezen szintjének eléréséhez ezek a tűzfalak – **közvetítőnek** *(proxy)* is nevezzük őket – valójában beékelődnek a kapcsolatba, és vezérlik is azt azáltal, hogy elfogják és megvizsgálják azokat. Amennyiben a közvetítő úgy dönt, hogy a kapcsolat engedélyezett, akkor a szerver felé a kezdeményező gép nevében egy másik kapcsolatot épít fel, amint azt a 3.8. ábrán is láthatjuk.



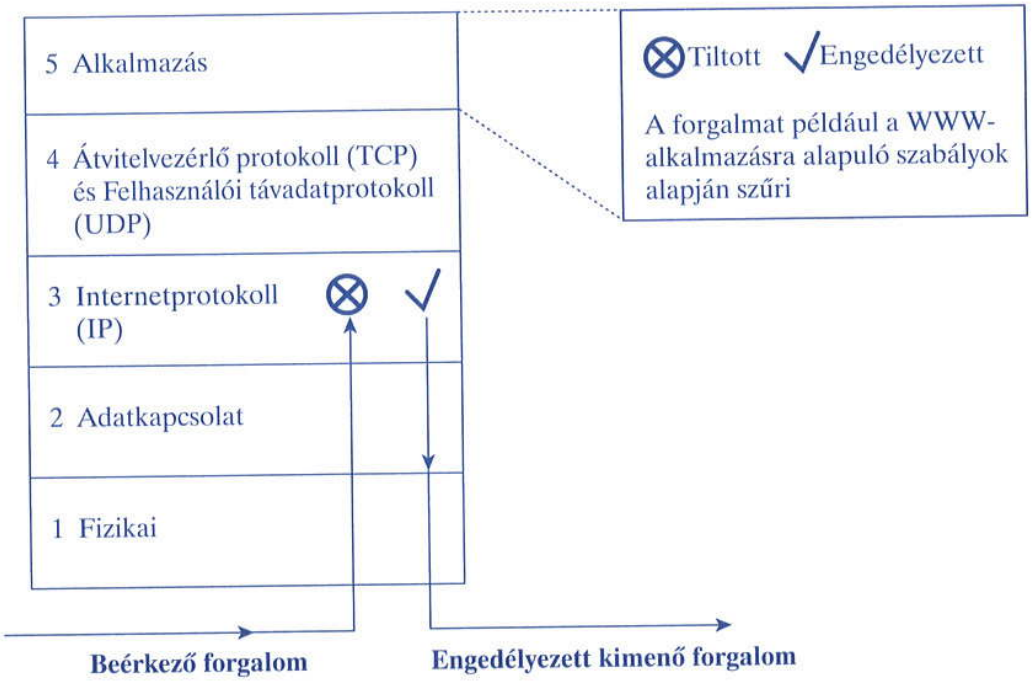

3.8. ábra. A közvetítő elhelyezkedése

Minden egyes csomagról le kell választani az adatot, meg kell vizsgálni, újra fel kell építeni egy másik csomagot, majd el kell azt küldeni a második kapcsolaton keresztül. Amint az alábbi listán látható, és amint arról később még szót ejtünk, különböző típusú tűzfalak képesek ezen feladat végrehajtására:

- **Alap közvetítő tűzfalak** – A közvetítő tűzfal nem irányítja a csomagokat, csupán továbbítja azokat. Ez a tűzfal a TCP/IP-modell 5. (alkalmazási) szintjén működik. Az ilyen tűzfal az egyik hálózati interfészen érkező adatfolyamokat fogadja, megvizsgálja azokat, majd az előre meghatározott szabályoknak megfelelőket, vagyis az engedélyezhetőket a másik hálózati interfészén továbbítja. A külső és a belső gépek között ilyenkor soha nincs közvetlen kapcsolat – a tűzfalon belüli gépek számára úgy tűnik, hogy minden információt kizárólag a közvetítő tűzfaltól kapnak.
- **Dinamikus közvetítő tűzfalak** – Az alap közvetítő tűzfalak továbbfejlesztéseként jöttek létre úgy, hogy megtartották annak valamennyi előnyét, és kiegészültek a csomagszűrés lehetőségével is. A dinamikus köz-

vetítő tűzfal a kapcsolat felépülésekor az adatfolyam első néhány csomagjának teljes vizsgálatát elvégzi, majd – ha a kapcsolat létrehozását jóváhagyta – az ugyanezen kapcsolat részét képező további csomagokat már csak a gyengébb, ám gyorsabb csomagszűrési technológia szerint vizsgálja. Összefoglalva, a kapcsolatot létesítésekor az 5. rétegben, a már létrejött kapcsolatot azonban a továbbiakban csupán a 4. (hálózati) rétegben vizsgálja.

Mivel ezen közvetítő tűzfalak a TCP/IP-modell 5. rétegében helyezkednek el, így az eddig tárgyalt technológiáknál sokkal pontosabban képesek az adatok vizsgálatára. Képesek például megkülönböztetni a csomagban található e-levelet és a Java-adatot, amint az a 3.9. ábrán is látható.




3.9. ábra. A közvetítő csomagvizsgálati módszere

Mire a megvizsgálandó csomag elérkezik a 3.9. ábrán látható közvetítőhöz, leválasztja a TCP/IP-fejlécinformációkat, és az adatokkal együtt vizsgálja. A *tényleges adatot* ezután a tűzfalszabályok alapján megvizsgálja, és az ellenőrzés végén a továbbítási engedélyt megadja vagy megtagadja. Ha az adat továbbítható, akkor a korábban eltávolított, ám megőrzött fejlécben található kapcsolati információkból újraírja a fejlécet, majd annak megfelelően elküldi a csomagot. A továbbítás megtagadása esetén a csomagot a bitkukába *(bit bucket)* dobja.

**Bitkukának** *tréfásan a szemétláda elektronikus megfelelőjét hívjuk. Amikor azt mondjuk, hogy a csomag a bitkukába került, akkor ezzel azt akarjuk kifejezni, hogy az útválasztó, tűzfal vagy közvetítő a csomagot eldobta.*



## 3.5.1. A KÖZVETÍTŐ KORLÁTAI

Mostanra már nyilvánvalónak tűnik, hogy minden eszköznek, technológiának vannak korlátai és hátrányai. A forgalmazók reklámosztálya persze nagyon szeretné elhitetni velünk, hogy az általuk kínált új biztonsági kütyü valamennyi problémánkat megoldja. A valóság azonban gyakran különbözik az általuk bemutatni kívánt rózsás képtől, s e tekintetben a közvetítő tűzfalak sem mások. Az alábbi hátrányuk mindenképpen megvan:

- **Csökkent teljesítmény** – A csomagok ennyire aprólékos vizsgálata és kezelése egyszerre jelenti azt, hogy a közvetítő tűzfalak meglehetősen biztonságosak és lassúak. A teljesítmény csökkenése azért következik be, mert továbbengedése előtt gyakorlatilag minden egyes csomag minden részét alaposan meg kell vizsgálni.
- **Nem mindig a legfrissebb** – Mivel állandóan új protokollokat és új alkalmazásokat fejlesztenek ki, így a közvetítő tűzfalakat is állandóan bővíteni kell, hogy az új adatformátumokat is felismerjék és vizsgálhassák. A bővítés azt jelenti, hogy állandóan új közvetítőket kell kifejleszteni és tesztelni. Ez viszont időbe telik, ami könnyen azt eredményezheti, hogy nem mindig a közvetítő legfrissebb változatát fogjuk birtokolni.



A védelem szempontjából legbiztonságosabb az alap közvetítő tűzfal, amely minden forgalmat alkalmazásszinten is vizsgál. Napjaink hálózatában azonban nem mindig ez a megoldás tekinthető a legpraktikusabbnak. A hálózat és a rajta zajló forgalom megértése és gondos tervezése ezért rendkívüli fontosságú az erős védelmi megoldás kidolgozásához. Egy parképítő vállalatnak például nyilván másfajta biztonsági igényei vannak, mint a hadiipar számára elektronikai termékeket fejlesztő és összeszerelő cégnek.

A tárgyalt kétféle – állapotteljes és közvetítő – tűzfalak közül legalább az egyiket ajánlatos használni a hálózat védelmére. Használjuk még a hálózati belépési ponton (útválasztón vagy tűzfalon) a címfordítást is végző csomagszűrést is, és máris elkezdtük a mélységi (többrétegű) védelem kifejlesztését. A következő részben azt vizsgáljuk meg, hogy miként lehet használni a tartalomszűrőket a hálózat és a felhasználók védelmére.

## 3.6. TARTALOMSZŰRÉS

A tartalomszűrés olyan hatalmas terület, hogy részleteivel és a lehetséges megoldásokkal számos cég foglalkozik, megkísérelve a számunkra szükséges „helyes" módszerek kidolgozását függetlenül attól, hogy otthoni felhasználóként vagy egy nagyvállalati rendszerben van-e rá szükségünk. Úgy tűnik, a hálózati csatlakozás mikéntjének valamennyi szintjén mindenki szembesül a tartalomszűrés valamilyen fajtája iránti igénnyel. Vegyük sorra az alábbi kihívásokat, amelyek mostanában kezdenek beszivárogni a politikába és a médiába:

- **Nyilvános könyvtárak és pornográfia** – Valamilyen oknál fogva létezik egy olyan embercsoport, akik úgy vélik, helyénvaló az adófizetők pénzéből fenntartott számítógépek használata különböző pornográf tartalmak böngészésére. A problémát még kényesebbé teszi az, hogy mindezt nyilvános könyvtárak közepén teszik – ugyanott, ahol egyébként gyermekek is olvasnak. (Az Egyesült Államokban viszonylag nagy vihart kavart ez a jelenség, itthon egyelőre nem nagyon került szóba.) A könyvtárak a jelenség ellen az ilyen típusú tartalmat elérhetetlenné tevő tartalomszűréssel védekezhetnek. Egyes vállalkozások szintén használhatnak tartalomszűrőket annak érdekében, hogy az interneten megtalálható bizonyos helyeket elérhetetlenekké tegyenek alkalmazottaik számára.

  A problémát sajnos nem csupán a pornográf oldalak jelentik – léteznek kábítószer-használatra, bűnözésre, terrorizmusra, erőszakra, gyermekek biztonságának fenyegetésére és gyűlöletkeltésre használt weboldalak is.
- **Szemétlevél (spam)** – Akinek van e-levélcíme, az naponta kap szemétlevelet *(spam)* is. A vállalkozások minden típusa harcol a szemétlevél ellen, egyre a detektálhatóságukért és megállításukért küzdve. Valahányszor egy jónak tűnő megoldás felbukkan, a levélszemét-küldők azonnal még találékonyabbá válnak, és megváltoztatják viselkedési mintájukat. Így például manapság sokan lebetűzik az e-levélcímüket (tomi pont tamas kukac bme pont hu) abban a reményben, hogy az e-levélcímek után kutató programokat megtévesztik vele. Egy ideig ez valóban sikerült is, azonban ez nem tarthat túl sokáig. A szemétlevél meggátolásában a tartalomszűrők például gátolhatják az alacsony jelzálogkamatot ígérő hitelekkel kapcsolatos bosszantó hirdetések postaládánkba kerülését. Ez egy különösen ostoba hirdetés (sajnos, csak egy a sok közül) – ugyan ki akarna hitelt egy olyan vállalattól, amelynek levélszemetet kell küldenie az üzletek nyélbeütéséhez? A vicces csak az, hogy a szemétlevél valamekkora bevételt mégis eredményez, különben már magától elhalt volna.

- **Vírusok és trójai falovak** – Az internet növekedésével párhuzamosan egyre jobban terjednek a vírusok is. Gyakorlatilag mindenki, aki az internethez kapcsolódik, rendelkezik e-levélcímmel, így a rosszindulatú csatolmányok e-levelekhez fűzése nagyon elterjedtté vált. A tartalomszűrők képesek ezen csatolmányok tartalmát ellenőrizni, s még azelőtt kiszűrni, hogy azok kárt okozhatnának.
- **Rosszindulatú weboldalak** – A támadók ma már képesek különféle kódokat elhelyezni a weboldalakon, amelyek segítségével ezen oldalak látogatóiról különféle információkat tudhatnak meg. Ennek számos módja létezik, a tartalomszűrők azonban képesek lehetnek a weboldalakat alkotó HTML-kód ellenőrzésére, és a rosszindulatú kódot tartalmazók kiszűrésére.
- **Üzleti siker megnövelése** – Első olvasásra talán valószínűtlennek tűnik, hogy a vállalkozások általános értelemben vett sikeréhez a tartalomszűrők is hozzájárulhatnak. Gondoljuk meg azonban, hogy a vállalatok és a kormányzati szervek a munkavállalóik viselkedése miatt komoly kockázatot vállalnak. Ha egy munkavállaló mondjuk erőszakos vagy illegális anyagot terjeszt egy vállalat hálózatán, annak bizonyára vannak kellemetlen kihatásai. A gyűlöletkeltésre alkalmas weboldalakat látogató munkavállalók, ha mást nem is, de annyit szinte bizonyosan elérnek, hogy velük szemben ellenséges munkahelyi környezetet alakítanak ki maguk körül, lerontják a munkahelyi morált, ami negatív módon hat a termelékenységre, ráadásul az esetleges következményként a sajtóban megjelenő ellenséges cikkek és/vagy jogi következmények költséges bírósági díjakat is jelenthetnek. Idézzük vissza az 1. fejezetben a közvetítői felelősséggel kapcsolatban elmondottakat. Ha egy alkalmazott mondjuk gyermekpornó anyagokhoz fér hozzá, akkor ezért akár a vállalat is felelőssé tehető, elvben elkobozható tehát a vagyona (a hálózat), ráadásul az ügy további negatív publicitást is jelenthet.

Az internet-hozzáférés a vállalatok egyre alapvetőbb szükségletévé válik, amely számos szervezet esetén komoly pozitív kihatással járhat. Gondok azonban a munkavállalók nyakló nélküli internet-hozzáférése kapcsán is felléphetnek, amint azt az előbb láttuk. Az eddig tárgyalt technológiák egyike sem képes ezen lehetséges kockázatok elhárítására. Persze helyesen vélhetjük úgy, hogy a saját vállalatunk esetén nem kell szembenézni az ilyen kockázatokkal – ez akár igaz is lehet. Ebben a részben a tartalomszűrést megvalósító technológiákat mutatjuk be, amelyek szükségleteinknek megfelelően számos különböző problémára nyújthatnak megoldást. A tartalomszűrés előnyei közé tartozik például:

- Csökkenti vagy megszünteti a jogi felelősséget azáltal, hogy a szervezet erőforrásainak illetéktelen vagy jogosulatlan használatát, illetve a bizalmas információk szándékolatlan csatolását nem teszi lehetővé.
- Optimalizálja a munkavállalók termelékenységét. Ki akarna azért fizetni, hogy a munkavállaló a saját szórakozására böngésszen az interneten?
- Javítja a munkavállalók internet-hozzáférésének nyomon követését. Ez azért fontos, mert ezáltal érezhetjük magunkat biztonságban. Semmiképpen sem tudhatnánk biztosan, hogy védettek vagyunk, ha nem figyelnénk állandóan, mi történik a hálózatunkon.
- Erővel betartatja a vállalat internetelérési szabályzatát, amelyet a 2. fejezetben ismertetett indokolható használati házirend tartalmaz:
  – lehetetlenné teszi az illegális vagy nem kívánatos anyagok elérését,
  – megakadályozza a jogosulatlan szoftverletöltést.

A hálózaton keresztül áramló csomagok tartalmának szűrésére számos lehetőség kínálkozik. A szűrési szolgáltatások ezernyi fajtáját kínáló vállalkozások serege és termékek özöne létezik. Mindegyikük ismertetése és megítélése meghaladja a fejezet kereteit. A kiválasztott terméktől függetlenül azonban ezeknek vannak közös vonásaik.

*A szervezet indokolható használati szabályzatának tartalmaznia kell, hogy mit várunk el tőle a hálózati erőforrások használatakor, valamint a tartalomszűréssel és -figyeléssel kapcsolatos intézkedéseket is meg kell adni.*

A tartalomszűréssel kapcsolatos megoldások kulcsa az internetről, csevegőszobákból, azonnali üzenetekből, e-levelekből és csatolmányaikból, a dokumentumokból, valamint a webböngészőkből származó adatok megfigyelésének és szűrésének lehetősége. A forgalom szűrésére számos módszer kínálkozik, amelyek azonban csupán két alapvető kategória valamelyikébe tartoznak:

- **Kliensalapú szűrés** – Ezen szűrési módszer esetén a meghatározott szabálykészlet alapján a tartalmat ellenőrző és szűrő szoftvert fel kell tölteni minden egyes ellenőrizni kívánt számítógépre. Otthonról dolgozó felhasználók esetén ez a megoldás a legelterjedtebb, és általában a frissítéseket tartalmazó szerverre való előfizetés formáját ölti.
- **Szerveralapú szűrés** – Ezen megoldás esetén az egyes kliensszámítógépeken nincs szükség külön szoftver installálására, mivel minden fel van töltve a szerverre, a kliensek pedig csak ezt a szervert érik el. Ezt a tí-

pusú szűrést például a levelezés, valamint a vírusellenőrzés esetén használjuk. Minden e-levél a központi levelezőszerverre érkezik, ami egyben az e-levelek tartalomszűrésének leglogikusabb helye is.

A tartalomszűréshez a weboldalakat megtekinteni kívánó felhasználók forgalmát egy közvetítő szerveren vagy Cisco-tartalomszűrőn keresztül kényszerítjük. A felhasználók is ellenőrizhetők annak eldöntésére, hogy a kérés engedélyezhető-e, vagy meg kell azt tiltani. A tartalomszűrést egy adatbázisban vagy könyvtárban található szóhasználati szabály, szó- vagy kifejezésgyűjtemény alapján lehet elvégezni.

Számos esetben a kérést és a választ is figyelni kell. A kérések bizonyos típusát például leírhatjuk a fenti adatbázisban vagy könyvtárban található szabályokkal (például hogy a http://www.showmeporno.com oldal elérését letiltsuk), más esetben azonban a kért weboldal tartalmát kell analizálni ahhoz, hogy a kérés engedélyezése vagy tiltása eldönthető legyen.

Az internet böngészését leíró ezen szabályok rendkívül hasonlóak a szemétlevél- és a vírusszűrés megvalósítására. Végső soron mindkettőben egy adatbázis tartalmazza a szűréshez használt mintákat vagy szabályokat, amelyek alapján eldönthető az engedély megtagadása vagy megadása. Amikor a forgalom beérkezik a hálózatba, ezen adatbázis alapján ellenőrzi azt. Például számos eszköz és termék használható szerverszinten a szemétlevél azonosítására és megállítására. Semmi sem működik persze százszázalékos biztonsággal, ezért sok e-levelező kliensben is megtalálható a felhasználók által beállítható további szemétlevél azonosítási eljárás.

## 3.6.1. A TARTALOMSZŰRÉS KORLÁTAI

A tartalomszűrésnek nagy szerepe lehet a hálózat védelmében, s a hálózati erőforrások megfelelő használatának biztosításában. Van azonban néhány olyan hátránya is, amelyek figyelembe vételével a szűrés műveletét még jobbá tehetjük:

- Az interneten minden héten nagyjából három-öt millió weboldal keletkezik vagy változtat nevet. Ez rendkívül nehézzé teszi a jó/rossz oldalak nyomon követését, és olyan dedikált szolgáltatás létezését követeli meg, amellyel biztosítható a szűrési szabályoknak állandóan a legfrissebb szinten tartása.
- A tartalmak folyamatosan változnak, az új weboldalak mellett állandóan új szemétlevél-küldési módszerek, új vírusok és terjedési módok és egyéb új fenyegetések jelennek meg, ami nagyon megnehezíti ezen változások állandó figyelemmel kísérését.

- Semmi sem tökéletes, így bizonyos mértékben számítani lehet a hamis találatokra is. A rendszer ezért nem tehető automatikussá, valamilyen szinten mindig kézben kell tartani a vezérlését, és nem célszerű vakon megbízni bármiféle külső osztályozásban. Csak egyetlen példaként figyeljük meg, hogy a www.msexchange.com a tartalomelemzés során értelmezhető „m sex change" kifejezésként is a helyes „ms exchange" helyett.

A tartalomszűrés valamilyen formája valószínűleg megtalálható a rendszerünkben. Alkalmazásának mértéke nagymértékben függ a védett vállalat méretétől és érzékenységétől.

## 3.7. Nyilvános kulcsú infrastruktúra

Ebben a részben megismerhetünk egy olyan technológiát, amellyel teljes mértékben biztonságossá tehető a hálózat: ez a nyilvános kulcsú infrastruktúra *(PKI – Public Key Infrastructure)*. Bizonyosan sokakkal előfordult már, hogy a hálózaton vásároltunk, vagy az interneten fellelhető valamilyen e-kereskedelmi tevékenységben vettünk részt. Ennek során valószínűleg mindenki észrevette a böngésző sarkában található kis jelölést, amely arra hívja fel a figyelmünket, hogy biztonságos tranzakcióban veszünk részt. A könyvben eddig elmondottak alapján van még olyan, aki ezt tényleg elhiszi?

A böngésző ablakán látható kis kulcs vagy zár azt jelenti, hogy **biztonságos csatlakozóponti rétegben** *(SSL – Secure Socket Layer)* használt tanúsítványt használó webhelyhez csatlakozunk, amely azt hivatott igazolni, hogy valóban ahhoz a webhelyhez csatlakoztunk, amelyhez akartunk. Nyugodtan vásárolhatunk, és megadhatjuk a hitelkártyánk számát.

*A kis zár (vagy kulcs) csupán azt jelzi, hogy a csatlakozott webhely SSL kapcsolati hitelesítést használ. Bárki létrehozhat ilyen hitelesítést használó webhelyet, ezért a jelzés ellenére ajánlatos továbbra is óvatosnak lenni!*

Valószínűleg azt is mindenki észrevette már, hogy az e-kereskedelmi ügyletben való részvétel során a http:// protokolljelző átváltozik http*s*:// jelzésre. Ennek az „s" betűnek a használata jelenti azt, hogy oda-vissza SSL fölötti HTTP-kommunikáció zajlik.

A háttérben az történik, hogy a webböngésző fogadja az SSL-tanúsítványt, felveszi a kapcsolatot a tanúsítvány kiállítójával, hogy ellenőrizze

az érvényességét, majd biztonságos módon folytatja a kommunikációt a szerverrel, hogy a tranzakciót teljes biztonságban lehessen lebonyolítani. Jó módszernek lehet ezt tartani?

Maga az SSL-tanúsítvány egy 40 (manapság inkább 128) bit hosszúságú adat, azonban hosszabb is lehet. Bizonyos jellemzőit a tanúsítványt birtokoló szerver tárolja 1024 bit hosszú adatként. Hasonlítsuk össze ezt a legalább 40 bit hosszúságú adatot az IP-címmel, amely ugye 32 bit hosszú – vagy a 128 bites 3DES titkosítással. Ennek fényében már látható, hogy az e-kereskedelmi tranzakció pusztán az SSL használatától még nem nagyon lesz biztonságos, mivel így önmagában nagyon sokat nem véd. Amint egyre növekszik az e-kereskedelem mértéke, a csalások száma még jobban nő. Emiatt persze az e-kereskedelembe, általában a vonali tranzakciókba vetett bizalom egyre csökken, mérsékelve így az elterjedését. Az üzleti és a hirdetői oldal persze ezt a tényt nem nagyon reklámozza. Nyugtalanságra azért persze nincs ok, mivel az e-kereskedelem biztonsága a nyilvános kulcsú infrastruktúrában gyökeredzik.

Valójában az SSL és a PKI nem egymás alternatívái, sőt az SSL maga is a nyilvános kulcsú infrastruktúrát használja. Mivel párhuzamosan léteznek és egyszerre használjuk őket, így tulajdonképpen egymást erősítik.

A **nyilvános kulcsú infrastruktúra** *(PKI – Public Key Infrastructure)* jelenleg fejlődő technológia, amely hamarosan nyilvánvalóan IETF-szabvánnyá fog válni. Célja egy olyan rendszer alapjainak megteremtése, amely a biztonsági szolgáltatások széles skáláját támogatja, beleértve az adat sértetlenségének, bizalmasságának és letagadhatatlanságának biztosítását is. Ezt a hardver, szoftver, eljárások és szabályok kombinációja segítségével lesz képes biztosítani, miáltal a felhasználók valóban biztonságosan kommunikálhatnak és cserélhetnek adatot attól függetlenül, hogy helyileg hol tartózkodnak.

A rendszer a hálózaton keresztüli tranzakció mindkét résztvevőjét ellenőrzi és azonosítja. Vegyük például a hálózatos hitelkártyacsalások emberekre és az üzleti életre gyakorolt hatását. Jelenleg mindenki veszít, ha csalás történik – az emberek azért, mert a hitelkártya-adataikat vagy azonosságukat ellopták, az üzlet pedig azért, mert így nehezebben kísérelhetik meg a szolgáltatás további nyújtását úgy, hogy profithoz is jussanak közben.

A PKI bővített digitális azonosítók és elismert hitelesítő szervezetek segítségével végzi el a hitelesítést, így ellenőrizve és hitelesítve a tranzakció mindkét résztvevőjét. Ez a tranzakció lehet olyannyira érzékeny ügylet, mint egy internetes vásárlás, vagy olyan nyilvánvaló, mint bizalmas információ e-levélben való kicserélése. A biztonságos kommunikáció és e-kereskedelem terén a PKI, mint a biztonság kulcsfontosságú összetevője, jelentheti a következő lépést.

További információk az alábbi helyeken találhatók:

- http://www.pki-page.org
- http://www.pkiforum.org

## 3.7.1. A PKI KORLÁTAI

Jelenleg nem tekinthető szabványnak, mivel az IETF-nek még jóvá kell hagynia a vele kapcsolatos RFC-ket. A PKI tanulmányozása során úgy tűnt, ez nagyon fontos következő lépés lesz a biztonság felé. Ez különösen igaz lett akkor, amikor a feleségem azonosságát ellopták,[6] amihez ráadásul a rendfenntartó hatóságok teljes érdektelensége társult. Lenyűgöző volt az a könnyedség, amellyel a bűnt elkövették. Ráadásként a különböző vállalkozások úgy gondolták, hogy ez egy olyan üzleti kockázat, amelynek a veszteségét egyszerűen le kell nyelniük. A PKI tehát valóban a biztonság irányába tett kitűnő lépés lesz, a jövőben azonban még meg kell küzdenie számos komoly kihívással:

- Az e-kereskedelem jelenleg is virágzik az interneten, függetlenül a hozzá társuló kockázat jelenlegi mértékétől.
- Az USA bizonyos államaiban (például Utah, Washington) már léteznek olyan törvények, amelyek szerint a kulcs feltörése vagy illegális használata esetén keletkezett adósságért a kulcs eredeti tulajdonosa továbbra is felelős. Látva azt a számlát, amelyet a tolvaj a feleségem azonosságával kifizetetlenül hagyott, nagyon félnék a PKI-technológiától, ha rá is kényszerítenének a használatára.
- Manapság a biztonság elsősorban a tanúsítvány tulajdonosának felelőssége. Ez előreláthatólag a PKI esetén is így marad. Így tehát nincs más választásunk, mint megbízni abban, hogy a tanúsítvány tulajdonosa minden lehetséges óvintézkedést megtett úgy, hogy nem gyártott közben újabb gyenge pontokat. A PKI bevezetése tehát egyre közelibb, azonban még számos nyitott kérdés van vele kapcsolatban.
- A PKI nem támogatja az egybejelentkezős infrastruktúrákat, és különböző erőforrások elérési kísérlete esetén a felhasználóknak többször kell igazolniuk azonosságukat, ez viszont biztos módja a katasztrófá-

6 Az Egyesült Államokban bizonyos e-kereskedelmi tranzakciókat a felhasználó által megadott néhány azonosító adat (amelyeket elvileg csak a jogosult felhasználó ismer) segítségével hajtanak végre. Ilyenkor a felhasználót az általa megadott adatok (ezt nevezzük „azonosságnak") hitelesítik, és egyben garantálják a későbbi fizetést. Hazai analógiája az interneten keresztüli kártyás vásárlás lehet, amikor a kártyát az általunk megadott néhány adat alapján fogja megterhelni a másik fél. *(A lektor megj.)*

nak. A felhasználók mindenféle módon megkísérlik majd „egyszerűsíteni" (vagyis meghiúsítani) a PKI jelentette biztonságot, aminek során számos hibát fognak elkövetni.

Jó vagy rossz tehát a PKI-hoz hasonló technológia? Erre a válaszra ma még nehezen adható kielégítő válasz, hiszen a módszer még nem eléggé felnőtt ahhoz, hogy szabvánnyá váljék. Bizonyos, hogy megnöveli a biztonságot, ami számos területen fontos lehet. A PKI-val kapcsolatos vélemény még a levegőben lóg, amely a technológia tényleges megvalósítóinak rigolyáitól, valamint majdani termékeik figyelemmel kísérésétől és kidolgozásától is függ.

A következő részben egy olyan módszert ismerünk meg, amely már ma is lehetővé teszi a hálózati hozzáférés során a hitelesítést.

## 3.8. AAA TECHNOLÓGIÁK

Manapság olyan világban élünk, ahol szinte mindent védenünk kell a viszszaéléstől és amelyben semmi sincs ingyen. Teljesen mindegy, hogy rendszergazdaként, igazgatóként, egyetemi hallgatóként vagy hálózati mérnökként dolgozunk. Amennyiben hálózaton keresztül kell szolgáltatásokat elérnünk, akkor három alapvető teendőt mindig el kell végeznünk:

- hitelesítés *(authentication)*,
- feljogosítás *(authorization)*,
- naplózás *(accounting)*.

E három teendőre együttesen **AAA** („három A", *triple A*) néven hivatkozhatunk. Amint a továbbiakban kifejtjük, mindegyiknek fontos szerepe van.

### 3.8.1. HITELESÍTÉS (AZONOSÍTÁS)

A hitelesítéssel biztosítható, hogy a hálózati felhasználó valóban az, akinek kiadja magát. Ez rendkívül fontos, hiszen nyilván nem akarjuk megengedni a jogosulatlanok számára a hálózatunkhoz való hozzáférést. Általában valamilyen közös titok ismerete, vagy egy megbízható másik szoftveralkalmazás hitelesít.

A hitelesítés teszi lehetővé a rendszergazda számára annak ellenőrzését, hogy ki csatlakozhat egy hálózatra kötött eszközhöz vagy érheti el az internetet úgy, hogy megalkotja a felhasználó azonosító nevét és jelszavát. Amennyiben egy felhasználó távolról akar csatlakozni egy útválasztóhoz

Telnet segítségével, akkor a hozzáféréshez csupán a jelszavát kell megadnia. Ez ugyan működőképes, de egyáltalán nem biztonságos megoldás, mivel – ha az útválasztó az internetről elérhető – egy támadó sorra próbálgathatja a jelszavakat a csatlakozási kísérletei során, amiről a rendszer fenntartója esetleg még csak nem is értesül. A támadónak tehát elegendő egyetlen jelszót kitalálnia ahhoz, hogy elérje az útválasztót. Mennyire van nehéz dolga, mikor a világ összes ideje a rendelkezésére áll ehhez?

Amikor valaki belép a hálózati eszközeink egyikére, és megváltoztat azon valamit, honnan fogjuk tudni, hogy ki ez a személy és mit csinált? Az AAA azonosítás esetén valahányszor a felhasználó belép, meg kell adnia mind a felhasználónevét, mind a jelszavát (mindkettőt a rendszergazdától kapta korábban). A terminálképernyő alábbi részletén látható, miként néz ki egy AAA hitelesítésre beállított Cisco-útválasztó távoli elérése:

```
User Access Verification
Username: tom_thomas
Password: xxxxxxxx
MyCiscoRouter>
```

Amint a példán látható, a felhasználónak érvényes felhasználónevet (témaszámot), valamint a csak ehhez a névhez tartozó érvényes jelszót is ismernie kell a hozzáféréshez. Az érvényes felhasználóneveket tároló adatbázis jellemzően vagy magán az elérni kívánt útválasztón, vagy egy távoli biztonsági szerveren helyezkedik el.

## 3.8.2. Feljogosítás

Az azonosításhoz kapcsolódik a feljogosítás, ami a hitelesítés befejeződése után zajlik le. Miután a felhasználót sikerült azonosítani, valamilyen módon azt is biztosítani kell, hogy az általa jogosan kívánt műveletekhez megkapja a kellő jogosultságokat. Ha valaki egyszerű felhasználó, akkor nyilván nem kaphat olyan jogokat, amelyek lehetővé tennék számára az állományrendszer bármely bejegyzésének elérését. A feljogosítást szabályok vagy hozzáférés-vezérlő listák *(ACL – Access Control List)* segítségével valósíthatjuk meg.

A feljogosítás teszi tehát lehetővé a rendszergazda számára az eszközhöz való hozzáférést az azonosítás során sikeresen elnyert felhasználó hozzáférési szintjének a beállítását. A Cisco IOS szoftvere támogat bizonyos felhasználói hozzáférési szinteket (úgynevezett *jogosultági szinteket*), amelyek azt vezérlik, hogy a felhasználónak milyen IOS-parancsok kiadására van joga. Például a legkisebb (0) jogosultsági szintű felhasználó egyáltalán nem adhat ki IOS-parancsot, míg a legnagyobb (15) jogosultsági

szintű felhasználó bármely érvényes parancsot használhatja. A hozzáférési jogosultságokat a helyi vagy távoli biztonsági szerver adja ki.

A Cisco-útválasztón a saját jogosultsági szintünket a *show privilege* parancs segítségével kérdezhetjük le, amint az alábbi terminálképernyő részletén is látható:

```
MyCiscoRouter#show privilege
Current privilege level is 15
MyCiscoRouter#
```

A feljogosítás előírhatja azt is, hogy milyen típusú protokollok használata engedélyezett a felhasználó számára. Megengedheti vagy megtilthatja például az FTP-, Telnet-, HTTP- vagy SSH-forgalmat. Soha ne feledjük, hogy minél magasabb számmal írható le a felhasználó jogosultsági szintje, az IOS-parancskészlet annál nagyobb részét érheti el.

## 3.8.3. NAPLÓZÁS

A naplózás akkor kezdődik, amikor már az azonosítás és a feljogosítás egyaránt megtörtént. A naplózás teszi lehetővé a rendszergazdák számára a felhasználók be- és kijelentkezéseinek, valamint az általuk végrehajtott műveleteknek a nyomon követését. A naplózás során gyűjtött információ kétségbevonhatatlan bizonyítékát adhatja a felhasználó rosszindulatú tevékenységének, hiszen rendelkezésünkre fog állni mind a felhasználó rendszerben töltött időpontjainak, mind a tevékenységének a listája. A rendszergazda nyilvántarthatja, hogy ki, mikor, melyik útválasztóra lépett be, milyen IOS-parancsokat adott akkor ki, és összesen hány bájtnyi adatot fogadott és küldött a benntartózkodás ideje alatt. A naplózás például lehetővé teszi a rendszergazda számára azt is, hogy megfigyelhesse azokat az útválasztókat, amelyeknek a konfigurációja megváltozott. A naplózási információkat gyűjtheti maga az útválasztó vagy egy távoli biztonsági szerver egyaránt.

*Amennyiben valaki az internetet telefonos behívó vonalon keresztül éri el, akkor AAA hitelesítés és feljogosítás segítségével éri el az internetszolgáltató hálózatát. A naplózás pedig az a folyamat, amellyel a szolgáltató például a számlázáshoz is szükséges időtartam információit (mennyi ideig voltunk belépve) gyűjti. Ugyancsak a naplózás segít a szükséges kapacitás felmérésében és tervezésében. Ezek mind fontos információk az internetszolgáltató számára.*

Miután az AAA-t beállítottuk, a külső biztonsági protokollok futtatására külső biztonsági szervereket is használhatunk (például RADIUS vagy TACACS), így nehezítve meg a hálózatunkhoz való illetéktelen hozzáférést. Mindkét szoftver implementálható Cisco hálózati eszközökön, s a későbbiekben még alaposabban is megvizsgáljuk őket.

*Amennyiben a RADIUS vagy TACACS biztonsági szerverprotokollok valamelyikét akarjuk használni, akkor kötelező az AAA beállítása is. Ahogy az AAA gyűjti az információkat, úgy küldi tovább azokat a biztonsági szerverekhez az AAA-val kapcsolatos bizonyos jellemzők meghatározásához.*

## 3.8.4. RADIUS

A **RADIUS** (*Remote Authentication Dial-In User Service,* távoli azonosítású behívó felhasználói szolgáltatás) egy kliens-szerver alapú rendszer, amely a behatolóktól védi a Cisco-rendszereket. Ugyanez a neve a használt protokollnak is, amelyet a Cisco IOS szoftver használ a hitelesítési kérések RADIUS-szerverhez történő továbbítására. A RADIUS szerver pedig egy olyan eszköz, amelyen a RADIUS-szolgáltatást nyújtó démon vagy alkalmazás fut. Együtt kell használni az AAA-szolgáltatással, hogy a távoli felhasználók azonosítása, feljogosítása és naplózása megtörténjék. Amikor a RADIUS-szerver azonosít egy felhasználót, a következő események következnek be:

1. A távoli felhasználótól megkérdezi a felhasználónevét és jelszavát.
2. A felhasználónevet és a jelszót titkosítja, és így küldi át a hálózaton.
3. A RADIUS-szerver vagy elfogadja, vagy elutasítja a felhasználónév/jelszó párost. Bizonyos esetekben a felhasználót arra is felszólíthatja, hogy további információkat adjon meg (ezt nevezik felszólítás/válasz módszernek). Ha például a felhasználó jelszavának érvényességi időtartama lejárt, akkor felszólítja őt egy új jelszó megadására.

*A hálózati hozzáférési szerver* (NAS – Network Access Server) *és a RADIUS közötti adatforgalom* ***nincs*** *titkosítva, mint például a TACACS esetén lenne, amely minden hálózati forgalmat titkosít.*

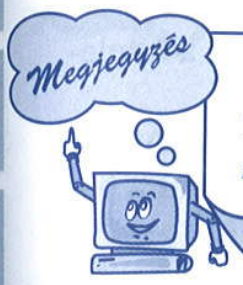
*A RADIUS-szerver a legtöbbször valamilyen platformon futó szoftver, amely platform lehet a Microsoft Windows szervercsalád tagja vagy egy UNIX/Linux-gép egyaránt. Használható az útválasztó felhasználóinak azonosítására, a szoftverkibocsátók hitelesítésére, de akár az IP-útvonalak érvényesítésére is.*

A Cisco-útválasztón az alábbi lépéseket kell megtenni a RADIUS-szerver engedélyezéséhez:

**1. lépés.** *Ki kell adni az* `aaa new-model` *parancsot. RADIUS esetén kötelező az AAA használata.*

**2. lépés.** *Meg kell adni a RADIUS-szervert a* `radius-server host` *paranccsal, amint az a 3-3 példán is látható.*

**3. lépés.** *Meg kell adni az útválasztó és a RADIUS-szerver között érvényes jelszót.*




*Természetesen arról is gondoskodni kell, hogy engedélyezése előtt már felvettük a felhasználónév/jelszó párokat a RADIUS-szerveren.*

A 3.3. példa azt mutatja be, miként kell egy Cisco-útválasztót úgy beállítani, hogy a 10.99.34.50 IP-című gépen futó RADIUS-szerver segítségével hitelesítse a felhasználókat.

3.3. példa. RADIUS konfiguráció

```
radius-server host 10.99.34.50
radius-server key <password>
```

Folytassuk most a vizsgálatunkat a RADIUS protokoll alternatívájával.

## 3.8.5. TACACS

A TACACS a *Terminal Access Controller Access Control System* rövidítése. A Cisco IOS három változatát támogatja: TACACS, bővített TACACS és TACACS+. Mindhárom a felhasználók hitelesítésére szolgál, és megtagadja a hozzáférést azon felhasználóktól, akik nem rendelkeznek érvényes felhasználónév/jelszó párral.

Az első változat csupán egyszerű jelszóellenőrzést és hitelesítést kínált. A naplózás meglehetősen korlátozott, mivel kizárólag a kéréseket és az elutasításokat naplózza. A bővített változat később az első lecserélésére jött létre. Az utolsó, a TACACS+ megnövelt naplózási lehetőségeket tesz lehetővé, s csak az AAA-val együtt működik (másként fogalmazva, a használata előtt ki kell adni az `aaa new-model` parancsot). Mindkét korábbi változatot felülmúlja. Általánosságban, a TACACS olyan központosított biztonsági rendszert nyújt, amely valamennyi távoli helyről azonosítja a bejelentkezni kívánókat. Jellemzően egy Windows-szerveren, vagy egy UNIX/Linux-gépen fut. Az azonosítás során a következő események játszódnak le:

1. A távoli felhasználótól megkérdezi a felhasználónevet és a jelszót.
2. A felhasználónevet és a jelszót átküldi a hálózaton és azonosítja azokat.
3. A szerver elfogadja vagy elutasítja a bejelentkezési kérést. Bizonyos esetekben a felhasználót további információk megadására is kötelezheti (felszólítás/válasz módszer).

A felszólítás/válasz módszert alkalmazhatja például akkor, ha a hitelesítés során valamilyen hiba következik be.

A Cisco-útválasztót az alábbiak szerint kell konfigurálni a TACACS+ használatához (jegyezzük meg, hogy az eredeti TACACS, valamint a bővített TACACS nem használják az AAA módszert):

**1. lépés.** *Ki kell adni az* `aaa new-model` *parancsot. TACACS+ esetén kötelező az AAA használata.*

**2. lépés.** *Meg kell adni a TACACS+ szervert a* `tacacs-server host` *parancs segítségével.*

**3. lépés.** *Meg kell adni az útválasztó és a TACACS+ szerver között érvényes jelszót.*

**4. lépés.** *Mivel az AAA-val együtt kell használni, így be kell állítani a TACACS+ hitelesítést, feljogosítást és naplózást.*

A 3.4. példán látható a Cisco-útválasztó beállítása a 10.99.34.50 címen található TACACS+ szerver használatához.

3.4. példa. TACACS konfigurálása

```
aaa new-model

! Beállítja az útválasztót, hogy hitelesítésre a tacacs+
! szervert használja
aaa authentication enable default tacacs+
```

```
! Legyen a tacacs+ szerver a parancsok végrehajtási
! jogosultságának ellenőrzője
aaa authorization exec tacacs+

! Gyűjtse a naplózási információkat az exec parancsokból
aaa accounting exec start-stop tacacs+

tacacs-server host 10.99.34.50
tacacs-server key <password>
```

A példán az alapbeállítások látszanak, azonban bonyolultabb `aaa` parancsok segítségével más konfiguráció is beállítható.

*Amikor engedélyezzük az AAA lehetőségét egy útválasztón, akkor figyelmetlenségből nagyon könnyen kizárhatjuk magunkat. Ha hebehurgyán beírunk néhány parancsot, majd kilépünk a konfigurálásból, könnyen előfordulhat, hogy nem leszünk képesek újból belépni.*

### 3.8.6. A TACACS+ és a RADIUS összehasonlítása

A hitelesítő szerver két megoldását összehasonlítva kitűnik, hogy mindkettőhöz engedélyezni kell a Cisco-útválasztón az AAA lehetőséget (hacsak nem a TACACS korábbi változatainak valamelyikét használjuk). A hozzáférési engedély megadásához mindkettő igényli a felhasználónév/jelszó pár helyes megadását. Közöttük a különbség egyrészt magában a protokollban van, másrészt a TACACS egy központi hitelesítő szoftver, míg a RADIUS a kliens-szerver technológián alapul.

## 3.9. Összefoglalás

A fejezet a hálózati védelem réteges felépítésének fontosságára hívta fel a figyelmet. A védelem mélységi kiépítése nyújtja ugyanis a hálózat számára a legnagyobb biztonságot. Nyilván el akarjuk kerülni a narancs szindrómát – a narancs esetén csupán egy réteget kell eltávolítanunk ahhoz, hogy hozzáférjünk a „becses dolgokhoz". Visszatérve a hálózatra, nyilván nem akarjuk azt, hogy a támadónak csupán egyetlen réteg feltörése után szabad hozzáférése legyen a rendszereinkhez.

A fejezetben röviden áttekintettünk a védelem mélységi kiépítéséhez szükséges több technológiát is:

- csomagszűrés hozzáférés-vezérlő lista segítségével,
- állapotteljes csomagvizsgálat,
- hálózaticím-fordítás,
- közvetítő és alkalmazásiréteg-szintű védelem,
- tartalomszűrés,
- nyilvános kulcsú infrastruktúra,
- AAA-technológiák.

A fentiek bármelyike önmagában csupán a védelem egy rétegét jelenti, együttes használatuk azonban ténylegesen többrétegű, mélységi védelmet nyújt, s így a „becses dolgokat" biztonságban tartja.

## 3.10. Összefoglaló kérdések

Az alábbi kérdések megválaszolásával ellenőrizhető a fejezetben tárgyalt elvek megértése.

1. Melyik hat, a hálózatbiztonsági technológiákkal kapcsolatos tervezési alapelvet érdemes szem előtt tartani a hálózat biztonságossá tételéhez?
2. Melyik az a szabály, amelyet a csomagszűrés során utolsóként automatikusan odaértünk a többi szabály után?
3. Az állapotteljes csomagvizsgálatot végrehajtó eszköz a csomag fejlécének mely jellemzőit vizsgálja, és miért fontosak ezek?
4. Sorolja fel az állapotteljes csomagvizsgálati technológia néhány hátrányát!
5. Fejtse ki, hogy mi a különbség a nyilvános és a saját IP-cím között!
6. Hasonlítsa össze a hálózati címfordítás három lehetséges típusát, sorolja fel közös vonásaikat és eltéréseiket, továbbá mondja meg, hogy közülük melyik a legelterjedtebben használt!
7. A közvetítő tűzfalnak melyik két típusa ismeretes?
8. Miért fontos a tartalomszűrés a hálózatokban?
9. Milyen lehetséges értékekkel bír a PKI a hálózat és az e-kereskedelem biztonságossá tétele szempontjából?
10. Az AAA a hálózat mely jellemzője szempontjából nyújt biztonságot?
11. Az interneten keressen három olyan forgalmazót, amelyek hatékony RADIUS-szerver megoldást nyújtanak!

# 4. fejezet Biztonsági protokollok



Még a legbölcsebb elmének is van mit tanulnia.
*(Ismeretlen szerző)*

***A fejezet elolvasása után érteni kell és el kell tudni magyarázni az alábbi témákat:***

- ***a DES és a 3DES mibenlétét, a közöttük fennálló különbséget, beleértve korlátaikat is,***
- ***az MD5 működését és a kapcsolat biztonságossá tételében játszott szerepét,***
- ***a PPTP és az L2TP közötti különbséget,***
- ***az SSH működési területét, s hogy mennyiben biztonságosabb a Telnetnél.***

A Cisco-eszközök világában jártas olvasónak talán meglepő lehet, hogy a fejezet címe „biztonsági protokollok", hiszen ott ezen fogalom alatt mindig valamilyen útválasztó protokollt értünk. Ezek közül talán a legjobb az „elsőként az elérhető legrövidebb út" *(OSPF – open shortest path first)* nevet viselő protokoll (az *open* a szabvány nyílt voltára utal, nem pedig a nevének a része), amelyet az útválasztókat jobban megismerni szándékozónak érdemes tanulmányoznia. Jelenleg azonban a biztonságra összpontosítjuk figyelmünket. A Newton-féle *távközlési szótár* szerint a protokoll „a számítógépek és az emberek közötti adatcsere részét képező üzenetek formátumát meghatározó szabálygyűjtemény".

A biztonság területén a **biztonsági protokoll** alatt a számítógépek közötti adatátvitel biztonságos kezelési eljárását értjük. Jelen fejezet a hálózati átvitelre kerülő adatok titkosítási eljárásait tárgyalja. Az adat biztonságos átvitelének eszközeit majd a további fejezetekben ismerhetjük meg.

Ebből a fejezetből tehát megérthetjük, hogy miként lehet az adatot biztonságossá tenni. Számos értelemben az adatok védelmére használt titkosítási képesség a hálózati biztonság egy újabb rétegét jelenti.

Nehéz elhinni, de mindennap kiadunk olyan információkat másoknak, amelyeket nem akarnánk a tudomásukra hozni. Az esetek többségében ezek bizalmas információk. Ez a legtöbbször nem szándékosan történik, semmi köze nincs bármiféle támadáshoz, főleg nem valamilyen bűncselekményhez. Túlságosan valószínűtlennek hangzik ez az állítás? Nos, vegyük figyelembe az alábbiakat:

- A bizalmas adatokat gyakran helyezzük el a hálózatról is elérhető szervereken, amelyhez így mások is hozzáférnek.
- A bizalmas adatokat gyakran másoljuk mágneslemezre, CD-re, más adathordozóra, vagy nyomtatjuk ki, így adva azt át másoknak.
- A bizalmas adatokat gyakran küldjük e-levélben, amely a hálózatunkon, vagy akár az interneten keresztül jut el a címzetthez.
- A bizalmas adatokat időnként más módon is közvetítjük elektronikusan.
- A bizalmas adatok időnként kikerülnek a webszerverre is.

Ezek a napi üzleti tevékenység részét képezik. Valamelyiket, vagy akár közülük többet is már mi magunk is elkövettük. Az ezzel kapcsolatos veszély csupán abban van, hogy a bizalmas adatot nyíltan közöltük, így a szándékosan vagy véletlenül (valószínűleg mindenki küldött már e-levelet rossz címre!) elfogott adatot bárki képes elolvasni. Most persze feltehető a kérdés, hogy milyen jellegű adatokat lehet rosszindulatúan felhasználni. Vegyük például az alábbi jellegű adatokat:

- **Személyi adatok** – Szinte mindenki megadta már a lakáscímét, telefonszámát, születési időpontját, egyéb személyes adatát e-levélben vagy helyezte el azt a honlapján.
- **Gazdasági adatok** – Sokan használnak valamilyen gazdasági vagy pénzkezelő szoftvert a saját gépükön. Ezen gépek többsége időnként csatlakozik a hálózatra. Gyakran előforduló tevékenység a bankszámla számítógépes ellenőrzése, részvényeink árfolyamának, tőzsdei információinknak a nyomon követése, sőt még a hitelkártya számának elektronikus vásárlási céllal történő megadása is.
- **Vásárlói adatok** – Számos vállalat tárolja a vásárlók adatait egy adatbázisban, esetleg fogad elektronikus megrendeléseket.
- **Orvosi adatok** – Manapság nem mehetünk be úgy az orvosi rendelőkbe, hogy ne látnánk ott legalább egy számítógépet. Helyenként már az is kezd általánossá válni, hogy az orvos kezében ott van az adatainkat tartalmazó tenyérszámítógép *(palmtop computer)!* Mi lesz, ha ezt a gépet elveszíti vagy ellopják tőle?

Itt csupán a legáltalánosabbnak számító bizalmas adatokat soroltuk fel, pedig ott van még a film, a zene, az új gyártmány tervei, az ügyvédek által készített különböző szerződéstervezetek, jogtárak, forráskódok, és számos más, szintén bizalmasan kezelendő adat is.

A legtöbbször persze semmilyen veszély nem leselkedik ezekre az adatokra, azonban egy esetleges hiba elkövetésének súlyos ára lehet. Mindenkinek és minden cégnek vannak tehát olyan fontos adatai, amelyeket nem akar másokkal megosztani. Ez a fejezet éppen azt tárgyalja, miként lehet megvédeni őket.

*Amikor titkosításról beszélünk, a jelszó helyett gyakran kulcsot mondunk. Jegyezzük meg, hogy ebben a fejezetben e két szó alatt pontosan ugyanazt értjük.*

## 4.1. A DES TITKOSÍTÁS

1972-ben a Nemzeti Technológiai és Szabványügyi Intézet *(NIST – National Institute of Standards and Technology),* amelyet akkoriban még Nemzeti Szabványügyi Hivatalnak hívtak, arra az elhatározásra jutott, hogy szükség van egy olyan erős titkosító algoritmusra, amellyel a minősítetlen adatok védhetők. Ez nagyjából ugyanakkor történt, amikor az általunk ma ismert internet kezdett egyre jobban terjedni az Egyesült Ál-

lamok és néhány más ország egyetemein és főiskoláin. A NIST döntését pontosan ez a tény befolyásolta, hiszen igény mutatkozott arra, hogy az emberek a saját adataikat megvédhessék az illetéktelenektől.

A kormányzatokra jellemző módon egy nyilvános előterjesztést dolgoztak ki, amely körülírta az új titkosítási algoritmus követelményeit. Olcsónak, könnyen terjeszthetőnek, mégis kellően rugalmasnak kellett lennie ahhoz, hogy számos alkalmazásba implementálható legyen, és nem utolsósorban nagyon biztonságosnak is.

1974-ben az IBM kiadta a Lucifer-algoritmust, amely – úgy tűnt – kielégítette az előterjesztésben meghatározott követelményeket. A Lucifer specifikációjában szerepelt az is, hogy 128 bit hosszú kulcsa van. A NIST nem rendelkezett kellő műszaki tudással a Lucifer megfelelő megítéléséhez, ezért felkérte a Nemzeti Biztonsági Ügynökséget *(NSA – National Security Agency)* a bírálatára.

A hetvenes évek közepén az NSA-ban nem nagyon bíztak meg a titkolódzó magatartása miatt. Ne feledjük, akkoriban javában tartott a hidegháború, és az NSA az Egyesült Államok biztonsága érdekében tevékenykedett. Az NSA ezért azt javasolta a NIST-nek, hogy a 128 bitet 56-ra csökkentse, jelentősen gyengítve ezzel az **adatkódolási szabvány** *(DES – Data Encryption Standard)* által nyújtott védelmet.

*Az NSA-t sokat vádolták azzal is, hogy úgy változtatta meg az algoritmust, hogy az ügynökeinek lehetővé tette bármilyen információ visszafejtését a titkosító kulcs ismerete nélkül is. Ezek a félelmek azonban igaztalannak bizonyultak.*

A NIST 1976. november 23-án fogadta el szövetségi szabványként a módosított Lucifer-algoritmust, és a nevét adatkódolási szabványra (DES) változtatta. Az algoritmus specifikációját 1977 januárjában tették közzé, és a kormányzat hivatalos támogatása mellett rövid időn belül széles körben alkalmazták.

A hetvenes évek végén, nyolcvanas évek elején, amikor a legjobb asztali számítógépek az Atari és a Commodore voltak, a DES nyújtotta biztonság ideálisnak tűnt. Amint azonban a számítógépek egyre gyorsabbá váltak, illetve egyre többen tanulmányozhatták az algoritmust, kiderültek a hiányosságai. Hamarosan meg kellett érteni, hogy az 56 bites kulcs nem elegendően hosszú ahhoz, hogy kielégítő biztonságot nyújthasson.

1997-ben ezért a NIST megvonta támogatását a DES-től, és egy helyettesítő titkosító algoritmuson kezdett dolgozni. Ez volt a **bővített titkosítási szabvány** *(AES – Advanced Encryption Standard)*. Azonban mindenki

kedveli a kihívásokat, és az RSA Security nevű cég elkezdett DES-feltörő versenyeket szponzorálni, amelyen a feladat az algoritmussal védett adat minél gyorsabb visszafejtése volt. A kódtörés eleinte napokig tartott, de a harmadik versenyen az Electronic Frontier Foundation „deep crack" nevű gépe – a distributed.net hálózat gépeivel együttműködve – már alig 22 óra 15 perc alatt fejtette meg a szöveget. Az alapítvány egy erre a célra tervezett gépet használt *(DES cracker)*, és a közel 100 000 számítógépet összekapcsoló distributed.net hálózat is segítette munkáját, együttes teljesítményükkel másodpercenként 245 milliárd kulcsot lehetett kipróbálni. Az RSA DES-versenyeiről bővebb információt az alábbi weboldalon találhatunk: http://www.rsasecurity.com/rsalabs/challenges/des3/index.html.

Jól érzékelhető tehát, hogy bármely cég, amelynek akár csak közepes nagyságú számítási kapacitás áll rendelkezésére, viszonylag rövid idő alatt képes lehet a DES-titkosítás feltörésére. A sérülékenységével kapcsolatos aggodalmak ellenére azonban még ma is számos gazdasági szolgáltató cég és más vállalat használja széles körben a DES-algoritmust érzékeny adatainak védelmére.



### 4.1.1. A TITKOSÍTÁS ERŐSSÉGE

A DES 56 bites, vagyis a titkosításra és visszafejtésre használt kulcs 56 bit hosszúságú. Alapfokú matematikai tudással felvértezve is hamar megmondható, hogy az 56 bit – 7 bit/bájtot feltételezve – nyolc bájtot jelent (ugye $7 \times 8 = 56$). Ha most valaki felkiált: de hiszen egy bájt nyolc bites!, akkor igaza van. A hagyományos ASCII-karakterkészlet 128 eleméhez tehát 128 különböző értéket lehet hozzárendelni. Az ASCII azonban 256 karaktert tartalmaz. Lévén ez pont kétszerese a DES kapacitásának, így minden egyes kulcselemhez két lehetséges karakter tartozhat (vagyis minden egyes bájt esetén két, egymással egyenértékű karakter létezik). A DES ráadásul minden kulcselem esetén egy bitet figyelmen kívül hagy.

A **titkosítási algoritmus** erőssége abban nyilvánul meg, hogy nagyon hosszú időbe telik a használt kulcs kitalálása. Egyszerű matematika segítségével meghatározható a lehetséges kombinációk száma:

$$128^8 = 128 \times 128 \times 128 \times 128 \times 128 \times 128 \times 128 \times 128 = 72\,057\,594\,037\,927\,936$$

Aki nem hiszi, számoljon utána! Így kimondva jól hangzik: hetvenkétbillió-ötvenhétezerötszázkilencvennégymilliárd-harminchétmillió-kilencszázhuszonhétezer-kilencszázharminchat. Minden egyes bitnek két állapota lehet: 0 vagy 1. Egy bájt nyolc bitet tartalmaz, s mivel nyolc bájt van,

így ez összesen 64 bit. A DES azonban csupán 7 bitet használ bájtonként, mivel minden egyes bájtban egy bitet figyelmen kívül hagy. Így jön ki az 56 bit. A fenti számítást tehát bitekben kifejezve az alábbi eredményt kapjuk: $128^8 = 2^{56}$. Ezért mondhatjuk azt, hogy a DES titkosítása 56 bit erősségű.

Fordítsuk le a fenti matematikai számítást a köznapi élet nyelvére. Tegyük fel, hogy valamilyen DES-sel titkosított adat került a birtokunkba, és szeretnénk belőle megfejteni a kulcsot. Amennyiben számítógépünk percenként egymillió lehetséges kulcsot képes kipróbálni, 72 billiónál kicsit több lehetséges kulcs van, vagyis a számítógépünk röpke 137 096 év alatt máris képes lesz megfejteni a kulcsot. Lenyűgözően hangzik, ugye? A hetvenes években a NIST és az NSA nyilván ezért kezdte támogatni a titkosító algoritmust. Azóta azonban a számítógépek fejlődtek egy kicsinykét – az RSA Biztonsági Vállalat versenyén használt technológiával másodpercenként 254 milliárd kulcsot lehetett kipróbálni.

### 4.1.2. A DES KORLÁTAI

A 2. fejezetben ismertetésre került jelszószabályzat a használt jelszavak (kulcsok) hatékonyságának növelése érdekében előírta bizonyos különleges karakterek és számok kötelező használatát. Tegyük most fel egy pillanatra, hogy az alkalmazottak szükségtelennek tartják ezt a szigorúságot, és nem tartják be az előírásait. Milyen kihatása van ennek a DES-titkosítással kódolt jelszó (kulcs) visszafejthetőségére?

Tegyük fel, hogy valamilyen fontos adatot csak normál karaktereket, mondjuk betűket tartalmazó kulcs segítségével kódol. Legyen ez az édesanyja neve, a kedvenc sportcsapata vagy a szülővárosának neve. Amennyiben csak betűket használunk egy jelszóban (kulcsban), akkor feltörhetőségét több százezerszeresével könnyebbé tettük.

Ez annyira fontos, hogy érdemes megismételni: csak betűk használata esetén nagyon gyorsan feltörhető jelszavunk lesz. Ez még a percenként egymillió művelet esetén is csupán 815 percbe, vagyis 14 óránál kevesebb időbe telik. A gyakorlatban azonban azok az emberek, akik csak betűket tartalmazó jelszavakat használnak, általában valamilyen értelmes szót választanak jelszó gyanánt, ezek a jelszavak pedig már percek alatt feltörhetők.

## 4.2. A TRIPLA DES (3DES) TITKOSÍTÁS

A DES a hetvenes években a problémára adott fantasztikus válasznak minősült, amely immár közel harminc éve hatékonyan működik. Tervezői azonban nyilván nem látták előre, hogy kevesebb mint 30 év alatt milyen nagyot fog változni a világ. Nem tudhatták, hogy az IT-ipar viharos fejlő-

désének ők csupán előfutárai. A technológia fejlődése végül azt eredményezte, hogy az üzleti élet szereplőinek a DES nyújtotta egyre gyengébb védelem helyett más megoldás után kellett nézniük.

A DES-algoritmus elavult. A helyén maradt űr kitöltésére kezdték használni a **tripla DES** (leírva **3DES**) módszert, amelyet a DES alapján fejlesztettek ki. Fejlesztése rendkívül gyors lehetett, hiszen a már működő megoldás (a DES) felhasználásával történt.

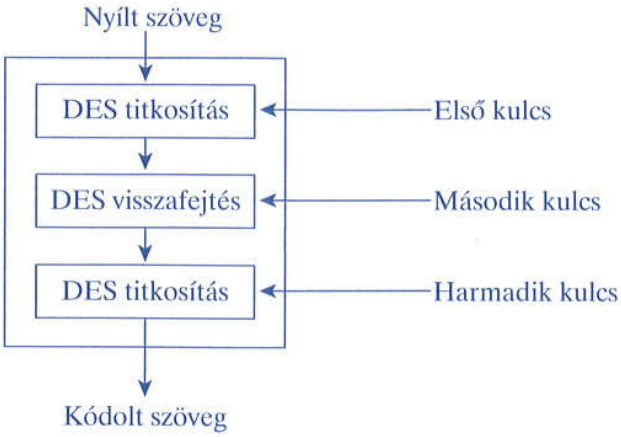
A két algoritmus nevének összehasonlításából azt a következtetést lehetne levonni, hogy a 3DES a kód feltörését háromszor nehezebbé teszi. A valóságban ez a szám némileg nagyobb, mintegy $5 \times 10^{33}$.

A 3DES-algoritmus ezt a kódolási művelet során elvileg három különböző kulcs alkalmazásával éri el. A három kulcs használata miatt a kulcs hosszúságát gyakorlatilag nyolcról huszonnégyre növeli, ami 168 bites titkosítási erősséget jelent. Matematikailag leírva a lehetséges kulcskombinációk számát: $2^{168} = 3{,}7 \times 10^{50}$.

Egy ötvenjegyű számot a csillagászok talán el tudnak képzelni, halandó emberek viszont aligha tudják azt kimondani. Korábban már volt szó arról, hogy percenként egymillió műveletet feltételezve mennyi ideig tart a DES-algoritmus kulcsának feltörése. A 3DES esetén ehhez $3{,}7 \times 10^{44}$ percre van szükség. Figyelembe véve, hogy egy év mintegy ötszázezer perc, ez úgy nagyjából $7 \times 10^{38}$ évet jelent. Fogalmam sincs arról, hogy ez milyen hosszú idő mondjuk a Tejútrendszer élettartamához képest, de az biztos, hogy jóval meghaladja a Nap becsült maradék életét. Bízvást nevezhetjük tehát a 3DES-titkosítást nagyon erősnek.[1] Ha valaki végig akarja követni, hogy miként lehetséges a pénzkiadó automatákba is beépített IBM 4758 processzor által is használt DES- és 3DES-titkosítás feltörése, és mire van hozzá szükség, annak javasolt a *http://public.planetmirror.com/pub/descrack/* cikk elolvasása, amely nagyban segíti az itt tárgyaltak jobb megértését is.

1 Valójában ennyire azért nem rózsás a kép. Az ANSI X5.92 néven szabványosított 3DES-algoritmus ugyanis csak két kulcsot használ. Az első kulccsal titkosítja az adatot, majd a második kulccsal „visszafejti", végül ismét az első kulccsal újra titkosítja. Mivel nem három, csupán két különböző kulcsot használ, így csupán 112 bites erősségű, nem pedig 168. Azonban ez is azt jelenti, ha a DES-algoritmussal titkosított adatot 1 másodperc alatt lennénk képesek megfejteni (ami azért még messze van a valóságtól), a 3DES-algoritmus feltörése még ekkor is 2,285 milliárd évet venne igénybe. Elméletileg persze lehetséges lenne három független kulcs használata is, azonban a 3DES-algoritmus jelenleg csupán kettőt használ (és mint látható, az is bőven elegendő). Ennek oka a kompatibilitás – ha a második kulcs megegyezik az elsővel, akkor tulajdonképpen DES-titkosítást használ, így a 3DES-eszköz kompatibilis lehet a csak DES-titkosítást ismerővel. *(A ford. megj.)*

## 4.2.1. A TITKOSÍTÁS ERŐSSÉGE



4.1. ábra. A 3DES-titkosítás lépései

A 3DES a DES olyan kiterjesztése, amely az adatok titkosításához három kulcsot használ, amint az a 4.1. ábrán is látható.

Maga a titkosítási eljárás pontosan ugyanaz, mint a DES esetén, azonban háromszor egymás után hajtja azt végre. A nyílt szöveget először az első kulcs segítségével kódolja. Az eredményt a második kulccsal ismét titkosítja (valójában ekkor a DES visszafejtő algoritmusát használja). A második menet után kapott adatot végül a harmadik kulccsal ismét kódolja. E három menet miatt nevezzük „tripla" DES-nek.

## 4.2.2. A 3DES KORLÁTAI

A szöveg minden darabjának háromszor egymás utáni titkosítása nyilván azt eredményezi, hogy a 3DES-algoritmus lassabb, mint a DES. Cserében – ha valóban három különböző kulccsal használjuk – több nagyságrenddel is erősebb titkosítást tesz lehetővé.

Érdemes mindhárom kulcsot különbözőnek választani. Ha az első két kulcs megegyezik, akkor csupán a DES egy lassabb változatára tettünk szert (hiszen a második lépésben visszafejti, amit az első lépésben kódolt).

*A jelenlegi gyakorlat szerint az első és a harmadik kulcs megegyezik. Előfordul, hogy a modernebb 3DES-titkosítást használó eszköznek egy régebbi, csak a DES-titkosítást ismerő eszközzel kell titkosított kommunikációs csatornát kiépítenie. Az ilyenkor szokásos eljárás, hogy a második kulcs is az elsővel egyezik meg.*

# 4.3. AZ MD5 ALGORITMUS

Az internet fejlődésével, valamint a világ adat- és hálózatközpontúvá válásával sajnos azt is meg kellett tanulnunk, hogy *„itt gépkalózok élnek"*. Ez oda vezet, hogy mindannyiunknak foglalkozni kell a biztonság, a hitelesség és az adatok sértetlenségének kérdésével.

Ezek a témakörök csaknem mindenki számára fontosak, kezdve a katonai/kormányzati felhasználástól az egészségügyi/személyes adatokon át egészen a gazdasági/pénzügyi vonatkozásúakig. Minden szervezet meg akarja akadályozni a titkos vagy bizalmas adataihoz való illetéktelen hozzáférést. A hitelesség és az adat sértetlensége az alábbi értelemben növeli a biztonságot:

- A **hitelesítés** során bizonyítást nyer az adat hitelessége, vagyis igazolható, hogy az adat feladója valóban az a személy vagy szervezet, akinek vagy aminek mondja magát. A hitelesség igazolásának fontosságát mi sem jelzi jobban, mint a digitális aláírások egyre nagyobb terjedése.
- A **sértetlenség** ellenőrzésével igazolható, hogy az adat az átvitel során nem változott meg, hanem pontosan megegyezik a feladott információval. Nyilván mindenki töltött már le az internetről szoftver- vagy operációsrendszer-frissítést. Ezek jól példázzák, mennyire fontos biztosítani az adatok sértetlenségét – a letöltött frissítések legyenek ugyanazok, mint amelyeket a frissítés készítője (akiben nyilván megbízunk) helyezett el a honlapján.

*Ide kívánkozik az adatvédelem még egy fontos összetevőjének ismertetése is, amelyre talán nincs olyan széles körben szükség, mint az ismertetett kettőre, azonban a maga helyén rendkívül fontossá is válhat. Ez pedig a* **letagadhatatlanság** *– egy hálózatos megrendelés esetén például fontos lehet a kereskedő számára, hogy a megrendelő később ne tagadhassa le a megrendelését, és ne állíthassa azt, hogy* „én nem ezt a lovat akartam".



Az **üzenetlenyomat 5** *(MD5, message digest 5)* nevű algoritmus az egyik legjobb olyan módszer, amely képes a fenti két követelmény kielégítésére. Ezt a kivonatoló algoritmust úgy tervezték, hogy a kivonatolt információ alapján elkészítsen egy rögzített hosszúságú adatot. Ezt az adatot ne-

*A „vagdalt" kifejezés nyilván a valós életből vett analógián alapszik, amely szerint „veszünk valamit, felvagdaljuk és összekeverjük". Veszünk valami felismerhető dolgot (mondjuk disznó- vagy marhahúst), összevagdossuk és összekeverjük. Az eredmény hasonlít az eredetire, azonban mégsem azonos vele. Hús esetén az eredmény a virsli vagy a hamburgerhús. Ahogy ebből sem lehet visszanyerni az eredeti húst, ugyanúgy a lenyomatból sem nyerhető vissza az eredeti adat.*

vezzük „vagdaltnak" *(hash value)*, lenyomatnak vagy üzenetkivonatnak. Ez az MD5 által nyújtott biztonság kulcsa.

Rivest 1994-ben dolgozta ki az MD5-algoritmust, amely olyan egyirányú kivonatoló algoritmus, amely bármilyen hosszúságú adatot képes fogadni, s eredményül egy 128 bites visszaállíthatatlan lenyomatot készít róla – ezt nevezzük lenyomatnak. (Hivatalosan az RFC 1321 írja le a módszert.) A lenyomat alapján nem lehet visszafejteni azt az adatot, amelyből készült. A gyakorlatban ez azt jelenti, hogy lehetetlen az MD5 lenyomatból visszaállítani az eredeti állomány tartalmát – ezért nevezzük egyirányúnak.

*Az egyirányú lenyomat egy olyan algoritmus eredményeként jön létre, amely bármilyen típusú adatból egy számsorozatot, ha úgy tetszik,* ***digitális aláírást*** *készít. Az adat hitelességének és sértetlenségének ellenőrzésére ezt a digitális aláírást lehet használni. Akárcsak a valódi aláírásnak, a digitálisnak is az a célja, hogy garantálható legyen segítségével az azt használó személy azonossága – ő ténylegesen az, akinek mondja magát.*

Fontos megemlíteni, hogy az MD5 nem titkosítja vagy változtatja meg az adatot. Ehelyett egy olyan lenyomatot készít belőle, amelynek segítségével az adat hitelessége és sértetlensége megállapítható. Mivel nem végez titkosítást, így az MD5 algoritmus szabadon használható és terjeszthető a világ minden részén.

**Hitelesítésnek** *azt az eljárást nevezzük, amelynek során egy személyről vagy egy eszközről a megadott felhasználónév és jelszó páros segítségével eldönthető az azonossága. A hitelesítés nem határozza meg, hogy a hitelesített személynek vagy eszköznek mihez van joga, csupán a személyazonosságát tisztázza.*
*A feljogosítás azon eljárás, amelynek során a személy vagy eszköz megkapja az őt megillető jogosultságokat – feltéve persze, hogy előtte már hitelesítve (azonosítva) lett.*

A következő fejezetben megvizsgáljuk az MD5 algoritmus működési alapelvét. Szót ejtünk arról is, hogy hány helyen találkozhattunk már az algoritmussal anélkül, hogy tudomásunk lett volna róla. Az MD5 matematikai háttere, azaz hogy milyen számításokkal képzi a lenyomatot,

meghaladja jelen könyv kereteit. Ha valaki többet is meg szeretne tudni, az ismereteket az RFC 1321 dokumentumban megtalálhatja (Az MD5 üzenetkivonat-képző algoritmus: http://www.isi.edu/in-notes/rfc1321.txt).



### 4.3.1. Az MD5 algoritmus működése

Aki használt már számítógépet, szinte bizonyosan találkozott már az MD5 algoritmussal, legfeljebb nem tudott róla. Az eljárásnak komoly szerepe van a hálózatok világában, amely számos módon alkalmazható:

- Amikor állományokat töltünk le az internetről, az MD5 lenyomatot használhatjuk a sértetlenség bizonyítására. Letöltés után egyszerűen ki kell számolni a saját gépünkön a lenyomatot, és össze kell azt hasonlítani az állományt elhelyező által szintén megadott értékkel. Egyezés esetén az állományt senki nem módosította.
- Használhatjuk a rendszerünk sértetlenségének ellenőrzésére is. Számos eszköz, például a később tárgyalt tripwire használja az MD5 algoritmust annak ellenőrzésére, hogy nem változtak-e meg a fontos rendszerállományok. Amennyiben valamelyik állomány megváltozna, a lenyomatok többé már nem egyeznek, s így a rendszergazda észlelheti a beavatkozást.

Az MD5-höz hasonló egyirányú kivonatoló eljárások esetén a sértetlenség ellenőrzésekor az általunk kiszámított lenyomat értékét kell összehasonlítani a kapott lenyomattal. Ezt az összehasonlítást nevezzük **lenyomat-ellenőrzésnek** *(hash check)*, a kapott értékre pedig **ellenőrző összeg** *(checksum)* néven is hivatkozhatunk.

Az MD5 ellenőrző összeget a szoftverfejlesztők gyakran használják annak ellenőrzésére, hogy a letöltendő állomány nem változik-e meg a letöltés során. A közzétett MD5 ellenőrző összeget össze kell hasonlítani a letöltés után kiszámítottal. Amennyiben a kettő megegyezik, a felhasználó biztos lehet abban, hogy a gépére letöltött szoftver vagy frissítés megegyezik a kibocsátó által közzétett állománnyal.

Az ellenőrzési folyamat védelmet nyújt a vírusokkal vagy a trójai kódokkal szemben, hiszen ha ilyenek fertőznék meg a letölthető kódot, akkor az ellenőrző összeg többé nem egyezne meg. Amint korábban is említettük, ezt tekinthetjük digitális aláírásnak is. Ezek pedig különösen fontos szerepet töltenek be az elektronikus kereskedelemben, és a legtöbb hitelesítő módszernek is kulcsösszetevői. Használhatóságukhoz szükség van egyrészt arra, hogy ne legyenek meghamisíthatók vagy feltörhetők. Ez a célkitűzés persze a maga teljességében megvalósíthatatlan. Elfogadható kompromisszumként azt írhatjuk elő, hogy legyenek önmagukban

ellenőrizhetők, bonyolult legyen a feltörésük, és olyan módon legyenek megtervezve, amely lehetővé teszi erősségük megnövelését vagy fejlesztését. Amint azt már a DES tárgyalásakor is láthattuk, a technológiai háttér erősödése könnyen elfogadhatatlanul alacsony szintre süllyesztheti a biztonságot, amennyiben nem teszünk megfelelő óvintézkedéseket vagy nem alkalmazkodunk a megnövelt biztonsági igényekhez.

## 4.4. A PONT-PONT KÖZÖTTI ALAGÚTPROTOKOLL (PPTP)

Ebben a részben a **pont-pont közötti alagútprotokollt** *(PPTP – Point-to-Point Tunneling Protocol)* ismertetjük, amelyet több cég (Ascend Communications, Microsoft Corporation, 3Com/Primary Access, ECI Telematics és az U.S. Robotics) együttes erőfeszítései hívtak életre.

A behívó hálózatok szabványos pont-pont protokolljára *(PPP – Point-to-Point Protocol)* épülő protokoll az OSI-modell második (adatkapcsolati) rétegében működik. (A PPP-klienst használó felhasználók egy ISP [internetszolgáltató] segítségével csatlakozhatnak a hálózatra.) A PPTP tartalmazza a hagyományos behívóhálózathoz tervezett PPP lehetőségeit, lehetővé téve a felhasználók (például üzleti partnerek vagy munkavállalók) számára egy biztonságos VPN-en (lásd 6. fejezet) keresztüli csatlakozást.

*Egy ilyen jellegű könyv megírásakor az egyik legnagyobb nehézséget az okozza, hogy mindig az ismertetni kívánt technológiával foglalkozzunk, és soha ne hagyjuk magunkat befolyásolni más technológiák által. A PPTP alapos átvizsgálásakor például felbukkant egy másik olyan technológia is, amelyet érdemes megemlíteni:*
*A postagalamb csomagátviteli protokoll* (PPTP – Pingeon Packet Transfer Protocol) *egy forradalmian új drót nélküli csomagátviteli technológia. Az átviteli kábel használata helyett ugyanis a csomagokat postagalambok hátára kell kötözni, és a végpontok között a levegőben kell továbbítani. Bővebb információk a http://www.ne2.com honlapon találhatók.*

A PPTP olyan protokoll (szabálygyűjtemény), amely lehetővé teszi a vállalatok számára a saját vállalati hálózatuknak a nyilvános interneten keresztüli saját „alagutak" segítségével történő kiterjesztését. A gyakorlatban tehát a vállalat a nagy kiterjedésű hálózatot *(WAN – Wide Area Network)* helyi hálózatként *(LAN – Local Area Network)* használhatja. A nagy kiterjedésű kommunikációhoz tehát nincs többé szükség saját bérelt

vonalakra, a kommunikációra biztonságosan használható a nyilvános hálózat is. Az ilyen jellegű összeköttetésen alapuló hálózatot nevezzük virtuális magánhálózatnak *(VPN – Virtual Private Network)*.



## 4.4.1. A PPTP MŰKÖDÉSE

A PPTP az adatokat PPP-csomagokba csomagolja, majd ezeket IP-csomagokban (távadatként, vagyis datagramként) továbbítja az internetalapú VPN-alagúton (csatornán) keresztül. A protokoll támogatja a csomagok tömörítését és titkosítását is. Az adat mindkét irányú továbbítása során használja továbbá az **általános alagúttechnika** *(GRE – Generic Routing Encapsulation)* egy formáját is.

A PPTP-alapú internet távoli elérésű VPN, messze a leggyakoribb formája a PPTP VPN-nek. A PPTP VPN azonban nem nevezhető a leggyakoribb VPN-formátumnak, mivel az IPSec sokkal biztonságosabb, és így népszerűbb is. A Cisco IOS szoftver lehetővé teszi a PPTP VPN használatát, azonban érdemes alaposan átgondolni a következő részben (4.4.2. A PPTP korlátai) elmondottakat. A PPTP-alagút két lépésben jön létre:

1. A PPTP-kliens segítségével csatlakozni kívánó felhasználó mindenekelőtt a behívó telefonvonalon bejelentkezik a saját internetszolgáltatójánál (modemes vagy ISDN-kapcsolaton) – esetleg a felhasználó egy kábelmodemen keresztül állandóan csatlakozva van.
2. Elindítja a PPTP-klienst, amely a szerverrel létrehoz egy vezérlő kapcsolatot (a 1723 porton), felállítva így az alagutat.

Miután az alagút létrehozása megtörtént, kétfajta csomag küldhető rajta keresztül: vezérlő üzenetek (amelyek az alagutat kezelik) és adatcsomagok.



A PPTP a PPP-től örökölt műveleteken alapszik. Ilyenek a kapcsolat kezelésére, a csomagok összeállítására és a felhasználók azonosítására való lehetőségek. A PPTP azonosításra a **felhívásos kézfogás hitelesítő protokollt** *(CHAP – Challenge Handshake Authentication Protocol)* vagy a **jelszóhitelesítő protokollt** *(PAP – Password Authentication Protocol)* használja.

A PPTP közvetlenül végzi a VPN-alagút fenntartását, és az adatokat ezen az alagúton keresztül továbbítja. A PPP által nyújtott mértéken felül is támogatja továbbá a VPN-adatok néhány további biztonsági lehetőségét.

A Microsoftnak köszönhetően a PPTP továbbra is népszerű választás a VPN-t használni kívánók körében. A PPTP-kliensek a Microsoft Windows valamennyi változatában ingyenesen megtalálhatók. A Windows-szerverek, valamint bizonyos Cisco-eszközök képesek PPTP-alapú VPN-szerverként viselkedni, és fogadni a PPTP-kliens kapcsolatokat.

## 4.4.2. A PPTP KORLÁTAI

Amint már említettük, a Microsoft-termékek elterjedtsége számos módon ösztönzi a PPTP használatát. Jóllehet más PPTP-gyártók is léteznek, azonban sokan természetesen azt használják, amivel már rendelkeznek – s számos szervezet rendelkezik Windows-szerverrel. A PPTP-nek azonban van számos korlátja és hátránya is, amelyek általános használatát visszavetik.

A PPTP-szabvány nem határozza meg, hogy a hitelesítést és az adattitkosítást miként kell megoldani. Ez azt jelenti, hogy két különböző gyártó előállíthat a szabványnak maradéktalanul megfelelő PPTP-eszközöket, amelyek mégsem képesek együttműködni. Ez bármely olyan szervezeten belül, amely különböző PPTP-implementációkat használ, komoly együttműködési problémákat jelent. Ha például az egyik gyártó a CHAP, a másik viszont a PAP hitelesítő protokollt választja, akkor a két eszköz képtelen lesz az együttműködésre.

A PPTP-kapcsolatok használatával összefüggésben komoly biztonsági aggályok is felmerülnek, főleg más lehetséges megoldásokkal összehasonlítva – különösen a Microsoft PPTP-implementációját övezi komoly kétely, amely pedig messze a legelterjedtebb.

A felügyeleti biztonsági szolgáltatásokat nyújtó Counterpane Internet Security vállalatot Bruce Schneider alapította, aki számos titkosítási technológia kifejlesztésében vett részt, beleértve a Blowfish és Twofish nevűeket is. A vállalat számos biztonsággal kapcsolatos kitűnő és alapos cikket publikált. Közösen a L0pht Heavy Industries vállalattal közzétette például a Microsoft PPTP-implementációjáról szóló kimerítő elemzését. Ebben többek között az alábbi vélemény is megtalálható:

*„A pont-pont közti alagútprotokollt (PPTP) arra tervezték, hogy az elterjedt PPP-protokollt felhasználva megoldja a nyilvános TCP/IP-hálózaton felépített VPN létrehozásával és kezelésével kapcsolatos problémát. Jóllehet a protokoll teret hagy bármilyen elképzelhető hitelesítő és titkosító eljárás alkalmazásának, a legtöbb kereskedelmi termék mégis a protokoll Microsoft Windows NT változatát használja. Ebben a cikkben ezért ezt a változatot fogjuk alaposan elemezni.*

*Azt találtuk, hogy a Microsoft-féle hitelesítési eljárás gyenge, és szótáralapú támadással könnyen feltörhető, a legtöbb jelszó órák alatt megfejthető. Úgy találtuk továbbá, hogy a titkosítás (mind a 40, mind a 128 bites) egyformán gyenge, és a titkosítás elleni más támadásokat lehetségessé tevő hibás tervezési döntések sorát fedeztük fel. Képesek voltunk a PPTP-kapcsolatfelvétel kijátszásával kapcsolatot nyitni a tűzfalon keresztül, és a Microsoft PPTP felhasználók bármelyike ellen számos komoly szolgáltatásmegtagadási támadást tudunk indítani."*

Amennyiben valakit az egész cikk érdekel, a következő webhelyen találhatja meg: http://www.counterpane.com/pptp-paper.html. A Counterpane Internet Security cég egy másik cikke a Microsoft CHAP-implementációjának (mind az MS-CHAP, mind az MS-CHAPv2) analízisét és annak eredményét tartalmazza: http://www.counterpane.com/pptpv2-paper.html. Az eredményekkel kapcsolatosan az alábbi humoros GYIK (FAQ) érdemes még elolvasásra: http://www.counterpane.com/pptp-faq.html.

Aki szakított némi időt arra, hogy elolvassa ezen írások némelyikét, most valószínűleg azon tűnődik, miként is implementálhatná a VPN-t, ha már egyszer a PPTP erősen ellenjavallt. A PPTP elsődleges alternatívája az IPSec-alapú VPN. Az IPSec nyílt szabvány, amelyet a hagyományos nyílt eljárás során az IETF (Internet Engineering Task Force – az internet technikai fejlesztését koordináló szervezet) vezérletével fejlesztettek ki, s amely nem tulajdona egyetlen vállalatnak sem. Ez nagyon fontos különbséget jelent, ugyanis a PPTP implementálásának a Microsoft által választott módszere azt eredményezte, hogy a protokoll a cég tulajdonává vált. A következő részben megvizsgáljuk a „második rétegbeli alagútprotokollt" (L2TP), amely szintén a PPP kiterjesztése, az IPSec-ről pedig a VPN-ről szóló fejezetben fogunk bővebben szót ejteni.

## 4.5. A MÁSODIK RÉTEGBELI ALAGÚTPROTOKOLL (L2TP)

A **második rétegbeli alagútprotokoll** *(L2TP – Layer Two Tunneling Protocol)* a PPTP kiterjesztése, amelyet az RFC 2661 ír le, és amelyet a virtuális magánhálózat (VPN) nyilvános interneten keresztüli működésének lehetővé tételére használunk. Az RFC 3193 azt definiálja, hogy miként lehet az L2TP-t használni a biztonságos IPSec átvitel segítségével – ekkor az L2TP-csomagokat a felhasználói távadat protokollal *(UDP – User Datagram Protocol)* a 1701-es porton továbbítjuk. Az IPSec „beágyazódó biztonsági rakomány" *(ESP – Encapsulating Security Payload)* módszere megvédi az UDP-információt, így biztosítva a biztonságos kommunikációt. A protokoll a Microsoft és a Cisco megegyezéséből született, örökölve a két alagútprotokoll legjobb tulajdonságait. A Microsoft adta a PPTP-t, míg a Cisco Systems adta a **második rétegbeli továbbítás** *(L2F – Layer 2 Forwarding)* technológiát.

A protokollt alkotó két fő komponens egyike az L2TP belépési koncentrátor *(LAC – L2TP access concentrator)*, amely a kapcsolat egyik végén elhelyezkedő eszköz, a másik pedig az L2TP hálózati szerver *(LNS – L2TP Network Server)*, amely az alagút másik oldali végződése (egyben valószínűleg a PPP-adatfolyam hitelesítője is).

Az L2TP tervezését és működését tekintve annyiban hasonlít a PPTP-re, hogy mindkettő a PPP-kapcsolatra épül. A két legnagyobb IT-vállalat együttműködéséből némely terület előnyt élvez – különösen az érzékeny adatok biztonságával kapcsolatos terület.

## 4.5.1. Az L2TP és a PPTP összehasonlítása

A két protokollnak számos egyforma olyan jellemzője és előnye van, amelyek a hálózati működésből és az eredeti tervezési elveikből következnek. Ezek a hasonlóságok a következők:

- Mindkettő logikus átviteli mechanizmust kínál a PPP-adatok küldésére.
- Az alagúttechnika és az adatok becsomagolása révén mindkettő képes a bármilyen protokollon alapuló PPP-adatok IP-hálózaton keresztüli átvitelére.
- A felhasználói hitelesítésre és a protokoll konfigurálására mindkettő a PPP kapcsolatfelvételi folyamatát használja.

Jóllehet sok hasonlóság lelhető fel, azonban számos különbség is van:

- A PPTP az adattitkosítást csak a PPP kapcsolatfelvételi folyamat (s így a PPP-hitelesítés) befejeződése után kezdi meg. Az L2TP/IPSec már a kapcsolatfelvétel során is titkosít.
- Az L2TP/IPSec használhatja a DES- vagy 3DES-protokollt egyaránt.
- A PPTP kizárólag felhasználói hitelesítést igényel, míg az L2TP ugyanezen felhasználói hitelesítésen kívül megköveteli a számítógép bizonyítványának bemutatásán alapuló számítógép-hitelesítést is.

A következőkben az L2TP néhány fontos előnyét tárgyaljuk, megvizsgálva azt is, hogy mennyiben nevezhető elődjénél (a PPTP-nél) biztonságosabbnak.

## 4.5.2. Az L2TP előnyei

Az internetszolgáltatók képessé váltak az L2TP-t használó virtuális magánhálózatok kiépítésére (mivel az L2TP internetes szabvánnyá vált), így felhasználóik a szolgáltató hálózatán keresztül élvezhetik a VPN előnyeit. További előnyök:

- Lévén szabványosított, az L2TP-eszközök együttműködési képessége nagymértékben megnövekedett.

- Az L2TP-alapú virtuális magánhálózat a szolgáltatók termékévé válhatott.
- A Cisco-eszközöket használó hálózatban a végpontok közötti szolgáltatás minősége *(QoS – Quality of Service)* előírható például a *DiffServ* vagy más QoS-technológia felhasználásával, amely a forgalmat ennek megfelelően válogatja és kezeli.
- A titkosítást az IPSec-technológia biztosítja, amely szintén szabványos (vagyis létezik RFC-definíciója). Az IPSec minden egyes csomag esetén ellenőrzi az adat származását (bizonyítandó, hogy a feljogosított felhasználó küldi a csomagot), a sértetlenségét (ellenőrizendő, hogy az adat a továbbítás során nem lett megváltoztatva), a visszajátszás elleni védelmet (megakadályozandó, hogy a korábban elfogott csomagok sorozatának újrajátszásával megtéveszthető legyen a címzett), és az adat bizalmasságát (a megfelelő kulcs ismeretének hiányában lehetetlenné téve az elfogott adatok megértését). A PPTP ezzel szemben kizárólag az adat bizalmasságát képes minden csomag esetén biztosítani.
- Több protokollt alkalmazó környezetben is használható, hiszen tervezésénél fogva az L2TP bármilyen irányított protokollt támogat, beleértve az IP-, az IPX- és az AppleTalk-protokollt is. Támogatja továbbá a távoli átvitelre használt nagy kiterjedésű hálózati *(WAN – Wide Area Network)* technológiákat (például Frame Relay, ATM, X.25 vagy SONET), akárcsak a helyi hálózati *(LAN – Local Area Network)* technológiákat (Ethernet, Fast Ethernet, Token Ring, FDDI).

Sok tekintetben az L2TP a legjobb mindkét nagy cég (Cisco és Microsoft) számára. Személy szerint úgy gondolom, hogy a Microsoft profitált belőle többet, mivel a korábbi PPTP nagyon sok kívánnivalót hagyott maga után. A továbbiakban megvizsgáljuk, miként működik az L2TP.

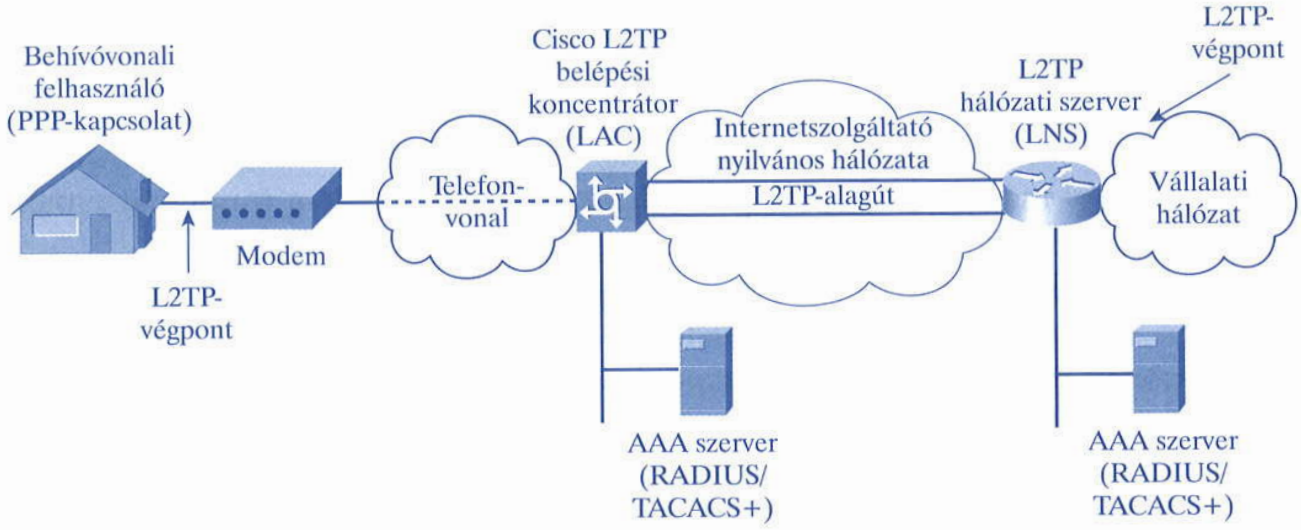


## 4.5.3. Az L2TP működése

Amint arról már esett szó, az L2TP támogatja a létező protokollok általános útvonalmegadás (GRE) segítségével történő, alagúton keresztüli továbbítását. Ez olyan architektúra kialakítását teszi lehetővé, amely alapján az L2TP-alagutak könnyen létrehozhatók a nyilvános interneten vagy egy behívószolgáltatáson keresztül.

A 4.2. ábrán az L2TP-hálózat kialakításának legelterjedtebb módszerét szemléltetjük. Azok az eszközök láthatók, amelyet egy internetszolgáltató használ-

*A hagyományos behívó hálózati szolgáltatások kizárólag a bejegyzett (nyilvános) IP-címek használatát támogatják, némileg korlátozva így a VPN-t használó alkalmazások fajtáit.*




4.2. ábra. L2TP-hálózati architektúra

na akkor, ha az L2TP-technológia valamennyi itt ismertetett előnyét valóban ki akarja használni. Gyakran előfordul, hogy a társaságok ennek csupán egy részét valósítják meg.

Az L2TP az internetet és annak hálózati kapcsolatait használja a különböző földrajzi helyen lévő végpontok összekapcsolására. A 4.2 ábrán a felhasználó számítógépe egy behívó-összeköttetést létesít (a második rétegben) az L2TP belépési koncentrátorral (LAC), amely egy AAA szerver segítségével hitelesíti, majd továbbítja a (titkosított) kapcsolatot az L2TP hálózati szervernek (LNS).

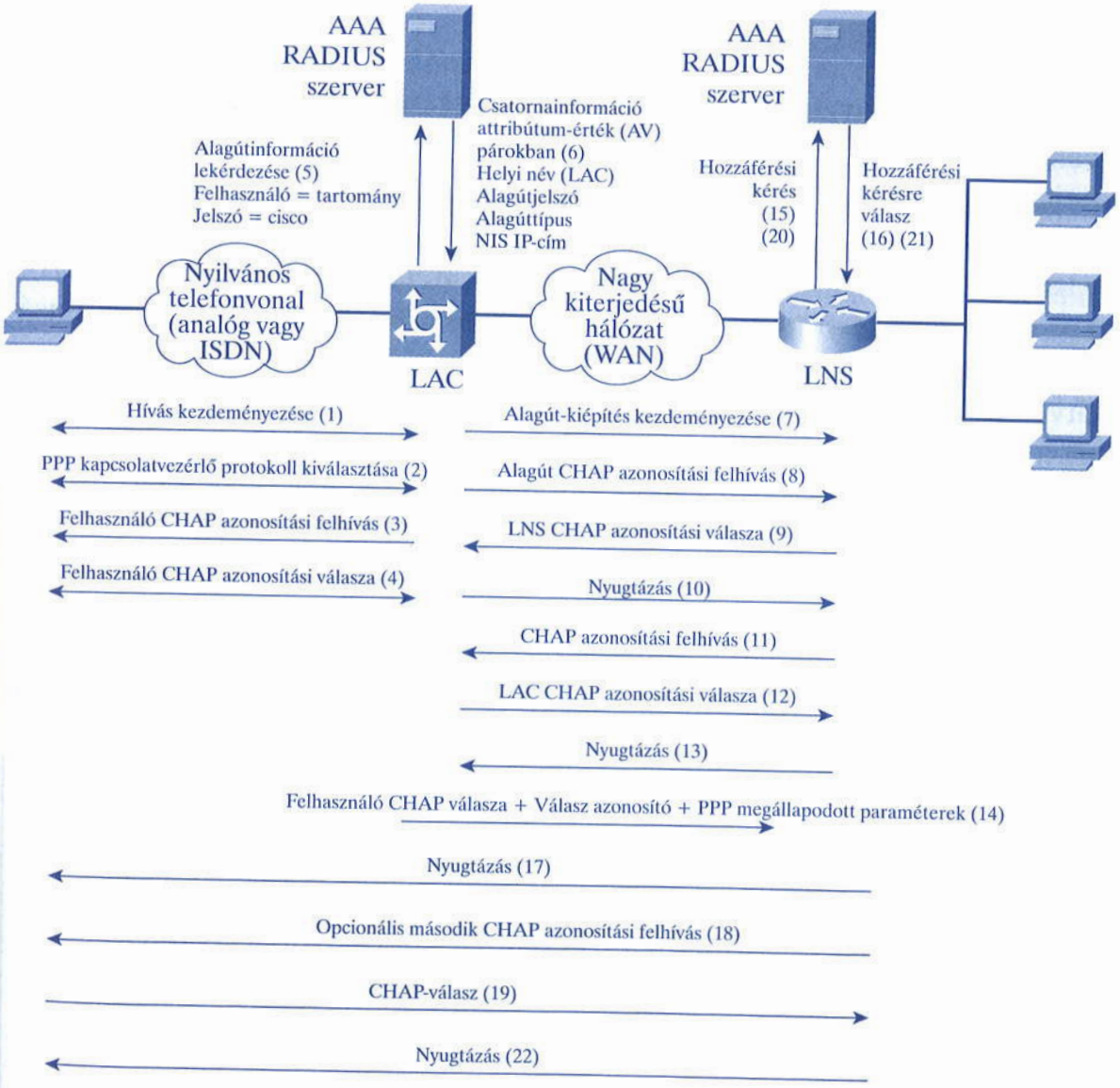
Az L2TP legnagyobb biztonsági erősségét az IPSec használata jelenti, amely a vezérlő és adatcsomagok továbbításának bizalmas, csomagonként hitelesített és visszajátszás elleni védelemmel ellátott módszere. Ezzel szemben a PPTP által használt Microsoft pont-pont titkosítás *(MPPE – Microsoft Point-to-Point Encryption)* csupán az adat titkosítását végzi el, de nem véd a módosítása vagy visszajátszása ellen.

A következő lista az internetszolgáltató behívóvonalán keresztül a vállalati hálózathoz való L2TP-csatlakozás tényleges lépéseit mutatja be az otthoni felhasználó felől:

1. A távoli felhasználó a nyilvános analóg vagy szélessávú telefonhálózaton keresztül kezdeményezi a PPP-kapcsolat kiépítését az internetszolgáltató behívóeszközével.
2. Az internetszolgáltató hálózatában lévő LAC a saját jelenléti helyén *(POP – Point of Presence)* fogadja a kapcsolatot, s ezzel a PPP-kapcsolat kiépült.
3. Miután a végfelhasználó gépe és az L2TP belépési koncentrátor (LAC) megegyeztek a kapcsolatvezérlő protokollban *(LCP – Link Control Protocol),* a koncentrátor a PAP vagy a CHAP hitelesítés segítségével részben azonosítja a felhasználót. A felhasználónév, a tartománynév, vagy a hívó-

szám-azonosítás *(DNIS – Dialed Number Identification Service)* segítségével meghatározza, hogy a felhasználó egy virtuális magán-behívóhálózat *(VPDN – Virtual Private Dialup Network)* tagja-e. Az internetszolgáltató csak így képes a VPN-szolgáltatás biztosítására, hiszen minden felhasználó és minden vállalat rendelkezik egyedi azonosítóval. A belépési koncentrátor által használt AAA szerver határozza meg a felhasználót.

4. Amennyiben a felhasználó nem egy (L2TP-t használó) VPN-kliens, a hitelesítés folytatódik, és a kliens hagyományos felhasználóként csatlakozik az internethez. Ha viszont VPN-kliens, akkor a kapcsolati kérése meghatároz egy adott végpontot (az L2TP hálózati szervert, vagyis az LNS-t), ahová a VPN kapcsolata csatlakozni fog. A felhasználói információkat a további azonosítás érdekében elküldi az ehhez az LNS-hez tartozó AAA szervernek.
5. Mielőtt a felhasználó bármilyen adatot is küldhetne a kiépülő alagúton, a két végpont (LAC és LNS) kölcsönösen azonosítják egymást.




4.3. ábra. L2TP-kapcsolat kiépítése

6. Miután az L2TP-t használó VPN-alagút kiépült, a felhasználó és a vállalati hálózat között létrejön egy L2TP-munkamenet.

A végeredmény tehát az, hogy az adatforgalom látszólag csak a behívó felhasználó és az L2TP hálózati szerver között zajlik, mintha semmiféle köztes eszköz (az LAC) nem is létezne. A 4.3. ábrán az L2TP beérkező hívása során kicserélt kapcsolatkiépítési üzenetek láthatók, időrendjük szerint megsorszámozva. Ezek a sorszámok az üzenetek időrendjét tükrözik, s nem kapcsolódnak a fenti leírásban szereplő sorszámokhoz.

A továbbiakban tárgyalásunkat a professzionális IT-felhasználók manapság egyik legkedveltebb protokolljával és eszközével folytatjuk: a biztonságos távelérés *(SSH – Secure Shell)* ismertetésével. Ez szintén nagyon jó biztonsági megoldásnak számít, így egyetlen hálózati biztonsággal foglalkozó könyv sem lehetne teljes nélküle.

## 4.6. A BIZTONSÁGOS TÁVELÉRÉS

A **biztonságos távelérés** *(Secure Shell)*, a közismert SSH a régebben azonos célra használt Telnethez hasonlóan a távoli számítógéprendszerekbe való bejelentkezésre használatos. Kettejük között azonban lényeges különbség van, mivel az SSH jelentős mértékben megnöveli a kapcsolat biztonságát. Az SSH-t azzal a céllal fejlesztették ki, hogy két számítógép között egy biztonsági szempontból nem megbízható, potenciálisan nem biztonságos hálózaton (például internet) keresztül egy titkosított kommunikációs útvonal legyen kiépíthető. Ennek megfelelően a protokoll megakadályozza a felhasználói jelszó és más érzékeny adatok egyszerű nyílt szövegként való átvitelét. Segítségével kiküszöbölhető az internet egyik legfontosabb biztonsági hiányossága: a jelszavak lehallgatása vagy feltörése.

A UNIX nem biztonságos r-parancsai (rlogin, rsh, rcp) lecserélésére lett kifejlesztve 1995-ben az SSH-1. Az r-parancsok a UNIX-felhasználók számára számos hasznos szolgáltatást nyújtottak, azonban komoly biztonsági kockázatot jelentettek. Az IETF 1997-ben kiadta az SSH-2 vál-

Megjegyzés

*Felmerülhet a kérdés, hogy mi a különbség az SSH-1 és az SSH-2 között, illetve hogy a két változat vajon csereszabatos-e. A kérdésre adható válasz röviden az, hogy nem csereszabatosak, az SSH-2 a korábbi változat teljes újraírása, egy teljesen más protokoll implementációja. A csomagokat biztonságosabban titkosítja, valamint a végpontok azonosítása is biztonságosabb, mivel csak lenyomatokat cserél.*

tozatát, amely az SSH-1 képességeit és biztonságosságát még tovább növelte. Napjainkban az SSH-1 lassanként átadja a helyét az egyre terjedő SSH-2 protokollnak.

Az SSH leggyakoribb használati módja a telnettel megegyező módon a távoli bejelentkezéssel kapott biztonságos parancsértelmező. Megőrzi a telnet alapvető funkcionalitását, sérülékeny megoldásait azonban egy olyan megoldással helyettesíti, amely az SSH-t a kapcsolatok valóságos szabványává emeli.

Az SSH ráadásul a telnethez képest megnövelt képességekkel és jellemzőkkel rendelkezik. Napjainkban az SSH gyakorlatilag minden platformon megtalálható: Macintosh, Microsoft Windows, UNIX, Linux és társaik.

Az SSH-alkalmazások jellemzően az interneten vagy más, nem biztonságos hálózaton keresztül biztosítják a távoli hozzáférést (bejelentkezést) bizonyos erőforrásokhoz, amelynek során az SSH alábbi három alapképességeinek egyikét kívánjuk használni:

- biztonságos távelérés,
- biztonságos állományátvitel,
- biztonságos végpontáthelyezés *(port forward)*.

Jóllehet az SSH-t elsődlegesen távoli bejelentkezésre használjuk, a protokoll használható általános célú titkosított csatorna kiépítésére is, amely alkalmas az állományok átmásolására, e-levélkapcsolatok titkosítására, valamint programok távoli végrehajtását is kiválthatja.

## 4.6.1. Az SSH és a telnet összehasonlítása

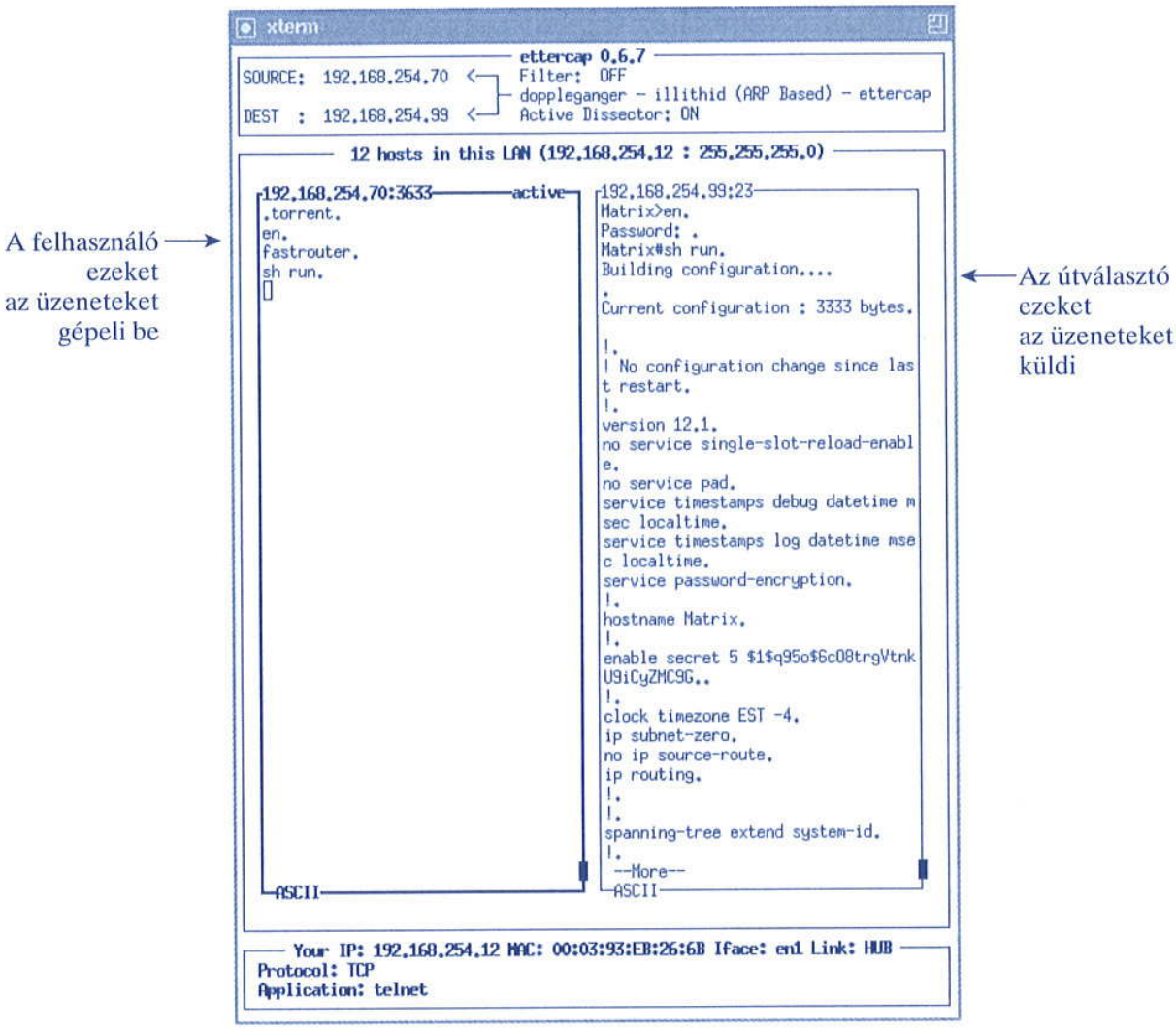
A telnet egyáltalán nem biztonságos, mivel nem védi és nem titkosítja a jelszavakat vagy a rajta keresztül zajló bármilyen tevékenységet. Ha rákeresünk a Google internetes keresőmotorral a „telnet vulnerabilities"[2] szavakra, akkor összesen 81 900 találatot kapunk, ha pedig a „telnet vulnerability" kifejezésre keresünk rá, akkor még mindig 966 találat közül válogathatunk.

Ha ennek ellenére kételkednénk abban, hogy a telnet egyáltalán nem biztonságos (végül is ha sokan hisznek valamiben, az csupán ettől még nem válik kétségtelenül igazzá), akkor vizsgáljuk meg a 4.4. ábrát. Ezen az ingyenesen elérhető `ettercap` program Apple Powerbook számítógépen történő elindítása utáni képernyőképe látható. A program két eszköz

2 Telnet sérülékenységek

közötti kommunikációt képes elfogni. Az ábrán egy másik számítógép és egy Cisco-útválasztó közötti üzenetváltások követhetők nyomon.

Az ábra jobb megértéséhez segítségként: az `ettercap` program elindításakor aktiválva lett az ARP-hamisító lehetőség *(ARP spoofing)*, így a programot futtató laptop beékelődött a 192.168.254.70 IP-című számítógép és a Cisco-útválasztó (192.168.254.99) kommunikációjába, amint az a képernyőkép tetején látható.

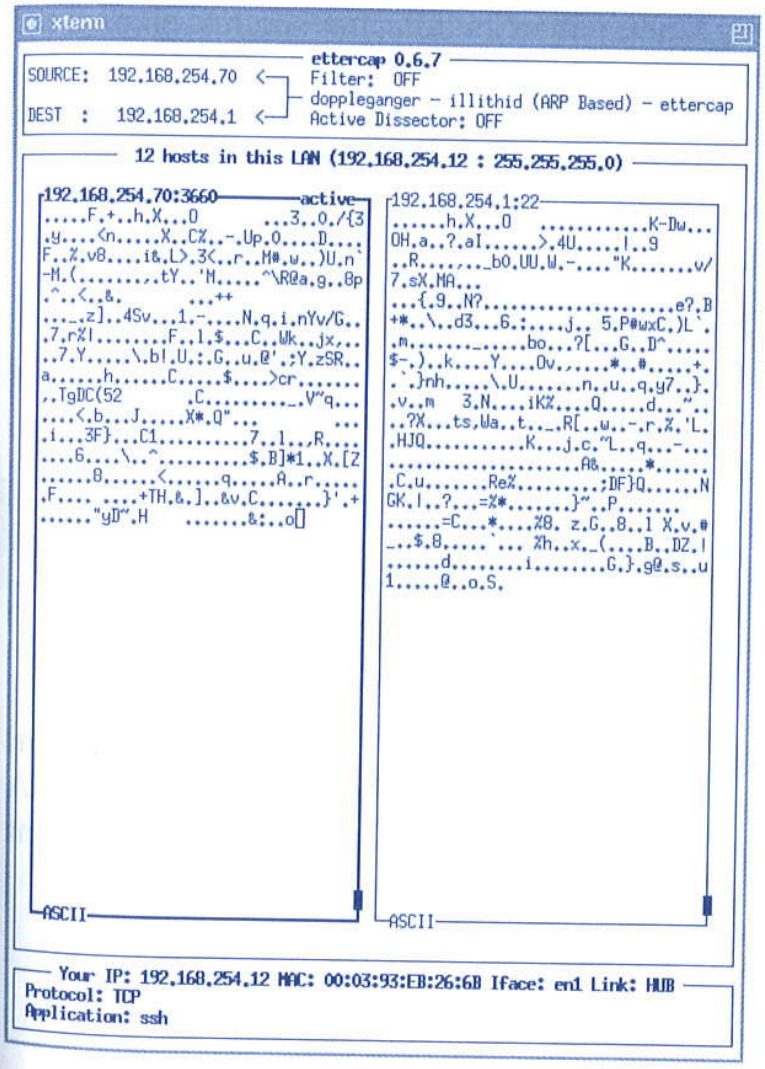



4.4. ábra. Az `ettercap` program szemlélteti a telnet program nyílt üzenetváltását

Észlelve és kiválasztva a két gép közötti telnet-kapcsolatot, a program elkezdte elfogni a mindkét irányba küldött ASCII-karaktereket. A bal oldali oszlop a PC-ről küldött karaktereket, a jobb oldali oszlop pedig az útválasztó által küldött jeleket mutatja. Látható, amint a felhasználó bejelentkezik, megadja a jelszavát, „engedélyező" *(enable)* módba kapcsolt, megadta az ehhez szükséges jelszót, végül végrehajtotta az **sh run** parancsot. Az útválasztó jelszó kérései és a konfigurációja szintén jól látható (a jobb oldali ablakban).

*Az* `ettercap` *a helyi hálózatban használható többcélú szaglászó, elfogó vagy naplózó program. Alapvetően számos aktív és passzív protokoll elemzésére használható (beleértve a kódoltakat is), azonban a hálózat és az eszközök analíziséhez is különböző módszereket kínál. Amennyiben többet szeretne megtudni az eszközről, a http://ettercap.sourceforge.net helyen megtalálja. A hozzá hasonló* `ethereal` *program, amelynek a legtöbb platformhoz létezik implementációja, ugyancsak a hálózati forgalom elemzésére szolgál. Nyílt forráskódú, szabadon felhasználható szoftver, amelyről további információk a http://www.ethereal.com webhelyen találhatók.*

A telnet nem biztonságos voltának bizonyítása persze nem lenne teljes akkor, ha nem mutatnánk meg az alternatívájaként rendelkezésre álló SSH-munkamenet során elfogott üzeneteket is. A 4.5. ábrán látható, amint az előbbi számítógép egy DES-titkosítású SSH-1 protokollt használó Cisco-tűzfalhoz csatlakozott (sajnálatos módon a Cisco-tűzfalak jelenleg csak az SSH-1 protokollt támogatják). Az ábrán látható, hogy sem a jelszavak, sem a tényleges tevékenység nem érthető.




4.5. ábra. Az SSH minden adatot kódolva küld

Az SSH tehát nyilván jobb a telnetnél, hiszen kódolja a forgalmat, előnyei azonban nem csupán a titkosításra korlátozódnak. További előnye és jellemzői:

- A csomagok IP-szintű hamisítását lehetetlenné teszi, így biztosítva a csomagokat küldő eszközök azonosságát.
- Megakadályozza a nyílt jelszavak vagy adatok közbülső eszközön történő lehallgatását, hiszen minden csomagot titkosít.

- IP-forrásazonosítás, amely megakadályozza, hogy egy rosszindulatú gép egy másik, megbízható gépnek kiadva magát küldhessen IP-csomagot.
- A csomagok átviteli útján bárhol lévő eszközöket uralma alatt tartó ember számára lehetetlenné teszi az áthaladó adatok megváltoztatását.

Tekinthető úgy is, hogy az SSH bármely más eszközben inkább megbízna, mint abban, amelyikkel éppen készül kiépíteni a biztonságos kapcsolatot. A telnet azonban bárkiben és gyakorlatilag bármiben megbízik. Arra is lett volna lehetőség, hogy az `ettercap` képernyőn látható telnet-kapcsolatot átvéve mondjuk töröljem az útválasztó konfigurációját. Elég riasztóan hangzik, igaz?

## 4.6.2. Az SSH működése

Röviden összefoglalva, az SSH két különböző végpont titkosított kommunikációs csatornával való összekötésére szolgál. Legegyszerűbb formájában a TCP segítségével kapcsolódik a géphez, felhasználónevet és jelszót kér az azonosításhoz. Amennyiben a hitelesítés sikeres, elkezdi titkosítani az adatokat.

Az SSH használt változatától függően a titkosítási módszerek serege áll rendelkezésre, amint azt a 4.1. táblázatban láthatjuk.

4.1. táblázat. Az SSH titkosítási lehetőségei

| Titkosítási eljárás | SSH-1 | SSH-2 |
|---|---|---|
| DES | igen | nem |
| 3DES | igen | igen |
| IDEA | igen | nem |
| BLOWFISH | igen | igen |
| TWOFISH | nem | igen |
| ARCFOUR | nem | igen |
| CAST-128-CBC | nem | igen |

Az SSH a hitelesítési fázis során választja ki és cseréli ki a kapcsolati kulcsokat. A használt SSH-változattól függően vagy az RSA, vagy a DSA kétkulcsú hitelesítést használja, egy nyilvános és egy titkos kulccsal, amint a 4.2. táblázatban látható.

4.2. táblázat. Az SSH-hitelesítés során használt titkosítás

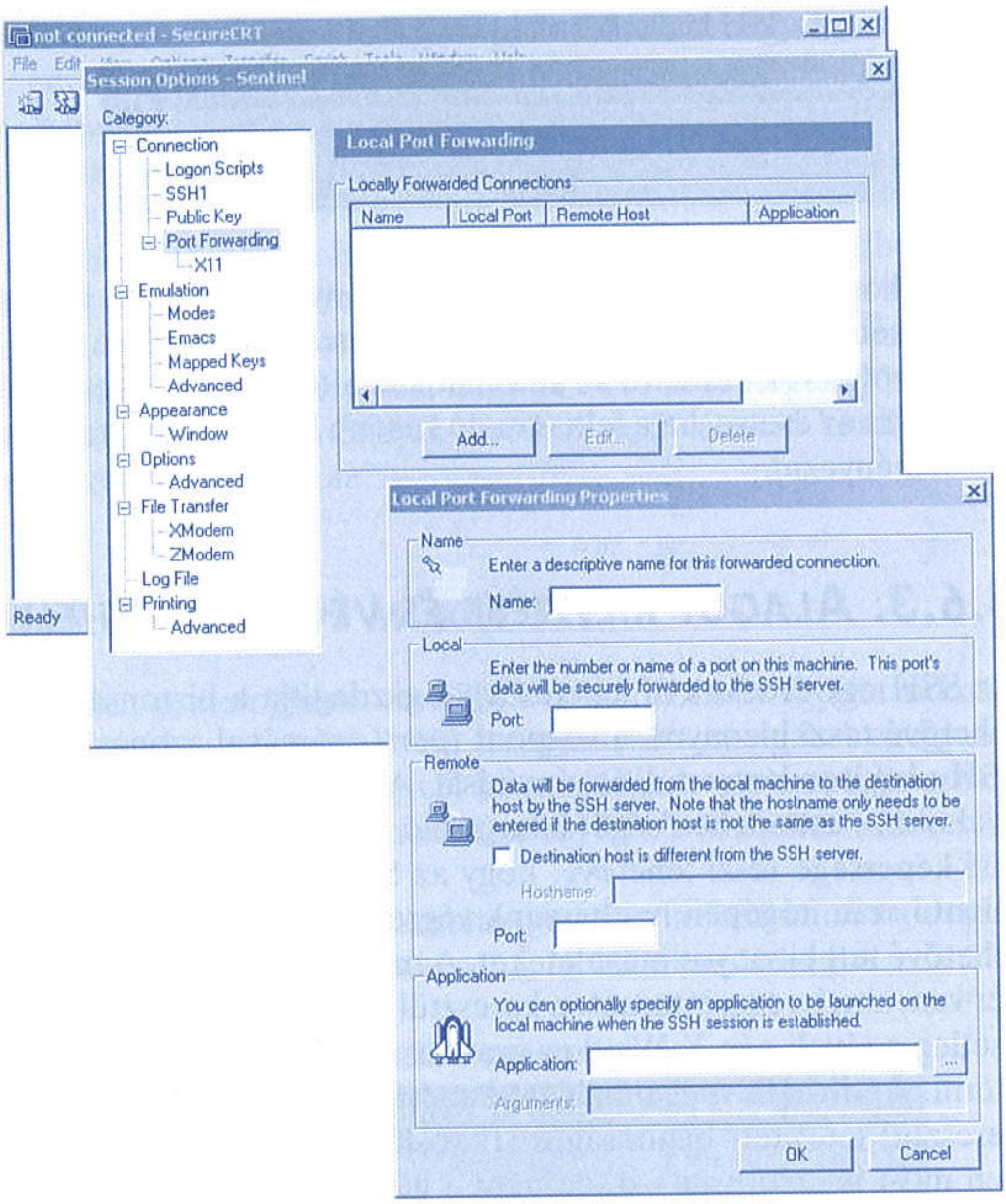
| Hitelesítő kódoló | SSH-1 | SSH-2 |
|---|---|---|
| RSA | igen | nem |
| DSA | nem | igen |

A titkos kulcsot kódolva tárolja, míg a nyilvános kulcs a megfelelően hitelesített felhasználó gépén kerül tárolásra. Ez lehetővé teszi az SSH-kliensszoftverek számára az automatikus hitelesítést, hiszen a kulcs tárolódik arra az esetre, ha a felhasználó később bármikor kapcsolatot akarna kezdeményezni.

## 4.6.3. Alagút kiépítése és végpont áthelyezése

Az SSH egy érdekes új lehetőséggel gazdagítja a biztonság birodalmát: lehetővé teszi bizonyos, a végpont *(port)* számával azonosított forgalom SSH-alagúton keresztüli átirányítását. Az FTP és az X Window a két leggyakoribb, ezt a lehetőséget kihasználó protokoll. Az SSH ezen átirányítási képessége teszi lehetővé, hogy az SSH-kapcsolat végződési pontját jelentő számítógépen hajthassunk végre az átirányított protokollok által lehetővé tett bizonyos műveleteket. A másik végén mondjuk egy webszerver van, amelyre az interneten keresztül akarunk állományokat feltölteni, esetleg a távoli gép X Window grafikus felületű asztalát akarjuk helyileg elérni. A virtuális magánhálózat kapcsán már volt szó a nyílt hálózaton keresztül létesített biztonságos (titkosított) kapcsolatról – tulajdonképpen most is erről van szó, de nem a gép teljes forgalmát, csak az adott kapcsolathoz (például X Window elérés) tartozót titkosítja.

*Az általam ajánlott legjobb SSH-kliens a VanDyke Software cég által árult SecureCRT (http://www.vandyke.com). Ez a grafikus felületű program számos hasznos beépített lehetőséggel rendelkezik, mint az automatikus naplózás, testre szabható parancsállományok és beállítható pufferek.*
*Sokan használják a szintén biztonságos PuTTY alkalmazást is, amely ingyenes, tág keretek között konfigurálható, és többféle csatlakozási lehetőséget kínál: a http://www.chiark.greenend.org.uk/~sgtatham/putty webhelyen lehet róla többet megtudni.*




4.6. ábra. A SecureCRT is lehetővé teszi az SSH-alagút használatát

A 4.6. ábrán a SecureCRT program egyszerű végpont-átirányítási beállítását szemlélteti. Megjegyzésre érdemes azonban, hogy az X Window könnyen átirányítható, de nem teljesen biztonságos, s ezért használják inkább az SSH-val együtt (vagyis titkosított kapcsolaton keresztül).

## 4.6.4. Az SSH korlátai

Az SSH-1 változatnak számos hibája és gyengesége volt, ezért hacsak lehet, tanácsos az SSH-2 változatot választani, így a legtöbb hiányosság egy csapásra megszűnik. Az SSH továbbá nem képes a belső rendszerek (számítógépek, szerverek stb.) védelmére, hiszen nem ez a feladata. Ha tehát

egy támadó hozzáfér ezek bármelyikéhez, akkor hozzáfér az SSH-hoz is,[3] amelyet szintén saját céljaira fordíthat.

Az SSH-alagúton keresztüli kapcsolat létrehozásának lehetősége azt sugallja, hogy a VPN-nek jó alternatívája lehet. Ez a feltételezés nem állja meg a helyét. Kitűnő megoldás azonban, ha a VPN-kapcsolatban használunk SSH-alagutat – ez ám az igazán biztonságos összeköttetés!

A fejezet ezen részében az SSH-t csak nagyvonalakban ismertettük, amely mindössze arra elég, hogy nagyjából megérthető legyen, mi is történik a háttérben. Ha az SSH nagyobb mélységű megértésére is szükség lenne, a [Barrett és Silverman, 2001] könyv elolvasása ajánlható.

Összefoglalva, az SSH a hálózaton keresztüli TCP-munkamenetek titkosításának hatékony és népszerű eszköze. Leggyakrabban távoli bejelentkezésre használjuk, azonban a hálózati biztonság növelése érdekében további célokra is használható.



## 4.7. Összefoglalás

A fejezet a DES- és a 3DES-titkosítási technológiák fontosságának és működésének ismertetésével kezdődött. Vethettünk egy pillantást a titkosítási módszerek matematikai hátterére, így szemléltetve feltörésük nehézségét – amelyet a hibásan megválasztott, például csak betűkből álló jelszó rendkívül megkönnyíthet.

Számos módja van a hálózati forgalom megvédésének és alagúton keresztüli továbbításának. A fejezetben két módszert ismerhettünk meg: a PPTP- és az L2TP-technológiákat. Megvizsgáltuk előnyeiket, s ezek alapján az L2TP-módszert ajánlottuk, mivel ez egyesíti magában a Microsoft és a Cisco legjobb képességeit. A fejezet végül az SSH ismertetésével zárult, bemutatva a hálózat számára jelentett értékeit.



3 Az SSH képes az automatikus csatlakozásra. Ehhez az kell, hogy a kliensen helyileg tároljuk a szerveren korábban előállított azonosító kódot. Amennyiben egy támadó képes ennek a kódnak az ellopására, akkor képessé válik a szerveren a kódot eredetileg birtokló felhasználó megszemélyesítésére, és a nevében bármilyen cselekedetre. *(A ford. megj.)*

## 4.8. Összefoglaló kérdések

Az alábbi kérdések megválaszolásával ellenőrizhető a fejezetben foglalt elvek megértése.

1. Hány bit hosszúságú a DES-kulcs?
2. A 3DES-módszer szerint ugyanazt a titkosítási kulcsot kell használni háromszor egymás után. Igaz vagy hamis ez az állítás?
3. Saját szavaival magyarázza el, mit értünk a „lenyomat" *(hash)* kifejezés alatt!
4. Mit használunk a digitális aláírás létrehozására?
5. Definiálja a hitelesítés fogalmát, és mondjon rá egy példát is!
6. Definiálja a feljogosítás fogalmát, és mondjon rá egy példát is!
7. Az MD5 működési folyamatának melyik pontján ellenőrzi a lenyomatot?
8. A fejezetben ismertetett biztonsági protokollok közül melyek használják az általános útvonalmegadás (GRE) módszerét?
9. Ismertesse az L2TP különböző biztonsági előnyeit!
10. Mit tartunk az SSH három alaplehetőségének?

# 5. fejezet Tűzfalak

A bátorság a félelem uralása,
a vele szembeni ellenállás,
nem pedig a félelem hiánya.
*(Mark Twain)*

***A fejezet elolvasása után érteni kell, és el kell tudni magyarázni az alábbi témákat:***

- ***Kinek van szüksége tűzfalra, és miért használunk tűzfalat a hálózati erőforrások védelmére?***
- ***A tűzfal mennyiben nevezhető a vállalati írott biztonsági politika technológiai kifejezésének?***
- ***Mikor van szükség a DMZ-re, és mik a tűzfal DMZ-t támogató beállításának az előnyei?***