TÁMOP-4.1.1.F-14/1/KONV-2015-0009
„A GÉPÉSZETI ÉS INFORMATIKAI ÁGAZATOK DUÁLIS ÉS MODULÁRIS KÉPZÉSEINEK KIALAKÍTÁSA A PÉCSI TUDOMÁNYEGYETEMEN „

Gyurák Gábor

Informatikabiztonság I.

Pécs
2015

A tananyag a TÁMOP-4.1.1.F-14/1/KONV-2015-0009 azonosító számú, „A gépészeti és informatikai ágazatok duális és moduláris képzéseinek kialakítása a Pécsi Tudományegyetemen” című projekt keretében valósul meg.

PÉCSI TUDOMÁNYEGYETEM
UNIVERSITY OF PÉCS

Európai Unió
Európai Szociális Alap
MAGYARORSZÁG KORMÁNYA
BEFEKTETÉS A JÖVŐBE

TÁMOP-4.1.1.F-14/1/KONV-2015-0009
„A GÉPÉSZETI ÉS INFORMATIKAI ÁGAZATOK DUÁLIS ÉS MODULÁRIS KÉPZÉSEINEK KIALAKÍTÁSA A PÉCSI TUDOMÁNYEGYETEMEN „

Informatikabiztonság I.

Gyurák Gábor
Szakmai lektor: dr. Muha Lajos
Nyelvi lektor: Veres Mária

Pollack Press
7624 Pécs, Boszorkány u. 2.

Felelős kiadó:

ISBN szám

Pécsi Tudományegyetem
Műszaki és Informatikai Kar

Pécs, 2015

PÉCSI TUDOMÁNYEGYETEM
UNIVERSITY OF PÉCS

Európai Unió
Európai Szociális Alap
MAGYARORSZÁG KORMÁNYA
BEFEKTETÉS A JÖVŐBE

TARTALOMJEGYZÉK

ÁBRÁK JEGYZÉKE

1. Előszó

Az utóbbi két évtizedben az információs technológiák alapjaiban megváltoztatták az életünket és a számítógép-hálózatok – különösképpen az internet – a mindennapi életünk részévé váltak. Ezek a változások új kihívások elé állítanak minket, és a kihívások között különös figyelmet kell szentelnünk a biztonságra.

A könyv célja alapvető ismeretek nyújtása az informatikabiztonság területén. A témakör jelen könyv terjedelmi korlátait lényegesen meghaladja, ezért csak olyan ismeretek bemutatására törekszünk, amelyek a mérnök-informatikusok képzésében részt vevő fiatalok számára elengedhetetlenek.

A könyv alapjául szolgál a Pécsi Tudományegyetem Műszaki és Informatikai Kar mérnök-informatikus szak „Az informatikabiztonság alapjai” című tantárgy oktatásának.

A szerző

2. Bevezetés

Az informatika vívmányai mindenhol körülvesznek minket. Társadalmunk működésének alapvető mozgatórugója lett az informatika. Akár az egyén, akár egy egész társadalom életét tekintjük, nyugodtan kijelenthetjük, hogy függővé váltunk ezektől a rendszerektől. [20] Ahhoz, hogy nyugodt szívvel bízzuk az életünket az informatikai rendszerekre, fontos, hogy megbízzunk bennük. Szeretnénk, ha az informatikai rendszerek biztonságosak, a rájuk bízott információk pedig biztonságban lennének. Ebben a bevezető fejezetben olyan alapvető kérdésekre keressük a választ, mint:

- Mit értünk informatikai rendszer alatt?
- Mi a biztonság?
- Mi az informatikabiztonság?

Az alapfogalmak tisztázása után áttekintjük a legfontosabb védelmi alapelveket, valamint kitérünk a rendszereinket veszélyeztető tényezőkre is.

2.1. Alapfogalmak

2.1.1. Informatikai rendszer

A könyv címe informatikabiztonság. Már az elején fontos rögzítenünk, hogy mit értünk informatika és informatikai rendszer alatt. A 20. század közepe óta a számítástechnika olyan mértékű változásokon ment keresztül és olyan sok területre begyűrűzött, hogy kihívást jelent a vele kapcsolatos fogalmak definiálása. Gondoljunk bele abba, hogy manapság különböző méretű és funkciójú számítógépekkel találkozunk a hagyományos személyi számítógépekben (PC), az okostelefonokban (smartphone), a gépjárművekben, az atomerőművek vezérlőrendszereiben – és még sorolhatnánk. Ezenkívül számtalan olyan felhasználási területe van az információs technológiáknak, amelyekre nem is gondolnánk. A könyvben informatikai rendszerek biztonságáról lesz szó. Az informatikai rendszert mint fogalmat, véleményem szerint a törvényben[1] is használt definíció írja le a legjobban:

„Elektronikus információs rendszer az adatok, információk kezelésére használt eszközök (környezeti infrastruktúra, hardver, hálózat és adathordozók), eljárások (szabályozás, szoftver és kapcsolódó folyamatok), valamint az ezeket kezelő személyek együttese.”

Az irodalom [1] szerint ide tartoznak:

- a számítástechnikai rendszerek és hálózatok;

[1] 2013. évi L. törvény az állami és önkormányzati szervek elektronikus információbiztonságáról.

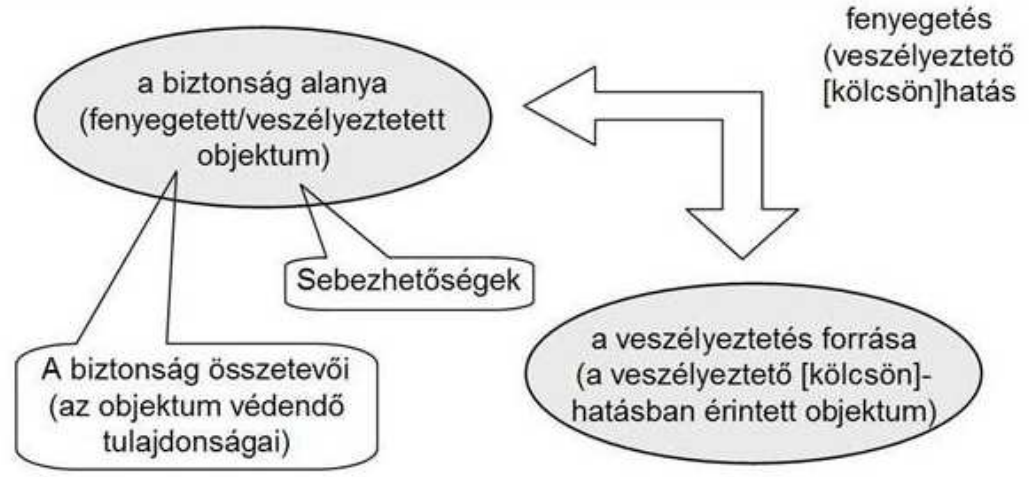
- a helyhez kötött, mobil és egyéb rádiófrekvenciás, valamint műholdas elektronikus hírközlési hálózatok, szolgáltatások;
- a rádiós vagy műholdas navigáció;
- az automatizálási, vezérlési és ellenőrzési rendszerek (vezérlő és adatgyűjtő, távmérő, távérzékelő és telemetriai rendszerek stb.);
- a fentiek felderítéséhez, lehallgatásához vagy zavarásához használható rendszerek.

2.1.2. Biztonság

Egy általános definíció szerint „a védelem tevékenység vagy tevékenységsorozat, amely arra irányul, hogy megteremtse, fejlessze vagy szinten tartsa azt az állapotot, amit biztonságnak nevezünk”. [1] Ezen definíció alapján a védelem egy tevékenység, míg a biztonság egy állapot.

A fenti definíció után tekintsük a biztonság alapmodelljét [2], amely a következő összetevőkből áll:

- *a biztonság alanya*: a fenyegetés által veszélyeztetett objektum;
- *a biztonság összetevői*: az alany azon tulajdonságai, amelyek megváltozása sértheti a biztonságot;
- *fenyegetések*: károsan befolyásolják a védendő objektumot;
- *sebezhetőségek*: az alany hiányossága, amely lehetővé teszi a fenyegetést;
- *veszélyeztetés forrása*: a biztonság alanyát veszélyeztető objektumok.



1. ábra A biztonság alapmodellje (forrás: [2])

A modell áttekintése után egy fokkal közelebb kerültünk a biztonság fogalmának egzakt meghatározásához. A definícióval kapcsolatban elvárásunk, hogy kedvező állapotot írjon le, akadályt biztosítson a támadónak, minimalizálja a támadás következtében elszenvedett

károkat és utaljon az időbeliségre is. Ismét a törvényben[2] találjuk azt a definíciót, amely mindezeknek megfelel:

„A biztonság a rendszer olyan – az érintett számára kielégítő – állapota, amelyben zárt, teljes körű, folytonos és kockázatokkal arányos védelem valósul meg.”

A zárt tulajdonság azt jelenti, hogy az összes releváns fenyegetést figyelembe kell venni. Teljes körű, folytonos védelem akkor valósul meg, ha megszakítások nélkül a rendszer összes elemére kiterjedő a védelmünk. Ha kihagyjuk valamelyik rendszerkomponenst a védelemből, akkor már nem lesz teljes körű. Végül, de nem utolsósorban olyan védelmet kell megvalósítani, amely a kockázatokkal arányos. Ez azt jelenti, hogy kicsi kockázat esetén szükségtelen túlzott erőfeszítéseket tenni a védelmi oldalon. Például általában felesleges a védendő objektum értékénél nagyobb összeget fordítani a védelemre. [11][1]

2.1.3. Informatikabiztonság

Ha az előzőekben bemutatott biztonság definíciót átültetjük az informatikai rendszerek világába, akkor a törvényben is szereplő definíciót kapjuk:

„Az elektronikus információs rendszer biztonsága az elektronikus információs rendszer olyan állapota, amelyben annak védelme az elektronikus információs rendszerben kezelt adatok *bizalmassága*, *sértetlensége* és *rendelkezésre állása*, valamint az elektronikus információs rendszer elemeinek sértetlensége és rendelkezésre állása szempontjából zárt, teljes körű, folytonos és a kockázatokkal arányos.” [3]

2.2. Modellek

Ebben a fejezetben az informatikabiztonság kapcsán alkalmazott modelleket tekintjük át.

2.2.4. CIA modell

A definíció kulcsszavait kiemelve eljutunk a szakmában jól ismert CIA háromszög (CIA triad) fogalmáig.

A rövidítés az alábbi fogalmak angol kezdőbetűiből adódik:

- Bizalmasság (Confidentiality)
- Sértetlenség (Integrity)
- Rendelkezésre állás vagy elérhetőség (Availability).

Ez a három követelmény, amelyet elvárunk az informatikai rendszerben tárolt adatokkal szemben, illetve a bizalmasság kivételével elvárjuk az informatikai rendszerrel szemben is.

[2] 2013. évi L. törvény az állami és önkormányzati szervek elektronikus információbiztonságáról.

2. ábra CIA modell

Az alábbiakban e három kulcsfontosságú kifejezés magyarázatát olvashatjuk:

- *Bizalmasság*

  A bizalmasság a leggyakrabban említett komponense az informatikabiztonságnak. Röviden megfogalmazva arról van szó, hogy nem akarjuk, hogy a bizalmas információk illetéktelenekhez jussanak. A védelem rengeteg pénzt és időt fektet olyan rendszerek fejlesztésébe, amelyek garantálják az adatok bizalmasságát. A következő fejezetekben ismertetjük a hozzáférés-szabályozást, amely megakadályozza, hogy jogosulatlanok hozzáférjenek adatokhoz. A kriptográfiai fejezetben arról is szó lesz, hogy különböző matematikai eszközökkel (például titkosítással) miként biztosítható a távoli felek hálózaton keresztüli kommunikációja úgy, hogy a hálózatra küldött adatok bizalmassága ne sérüljön.

- *Sértetlenség*

  A sértetlenség biztosítása azt jelenti, hogy a rendszerben és az adatokban nem lehet jogosulatlanul módosításokat végezni. Védelmet jelent például az alábbiakkal szemben:

  - jogosulatlan személy (például egy hacker) módosításokat végez egy adatbázis rekordjaiban,
  - egy jogosult személy jogosulatlan műveleteket hajt végre,
  - az adat megváltozik egy rendszerhiba (például egy hirtelen túlfeszültség) miatt.

A sértetlenség jelenti továbbá az adatok hitelességét (az adatot tényleg az készítette, aki mondja) és az adatok letagadhatatlanságát (a dokumentum készítője később nem tudja letagadni, hogy ő volt annak szerzője) is. Ezeket a megoldásokat szintén a kriptográfia fejezetben tárgyaljuk, digitális aláírás címszó alatt.

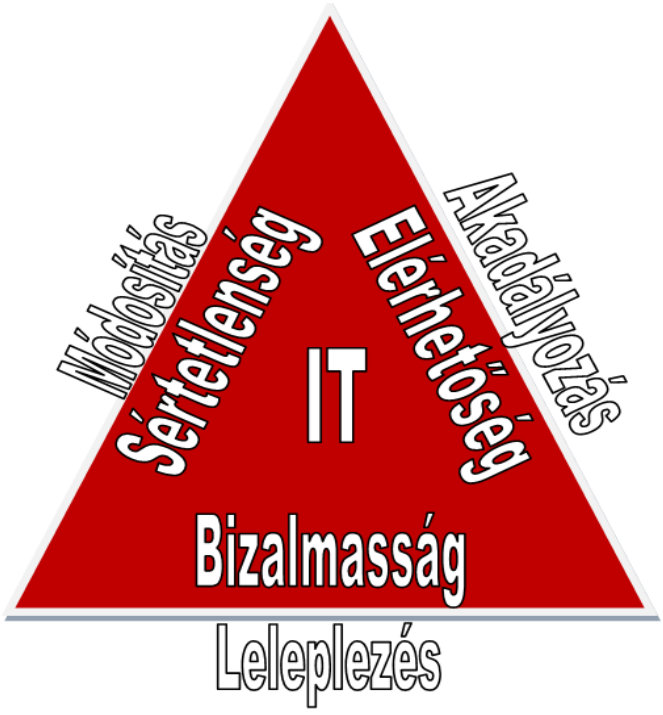
- *Rendelkezésre állás*

  Az IT biztonsági rendszerek harmadik feladata annak biztosítása, hogy a jogosult felhasználók hozzáférhetnek adataikhoz. Hatalmas károkat okozhat egy vállalat életében, ha a szükséges adatok nem érhetők el. Egy támadás, amelynek során a támadó elérhetetlenné tette az adatokat, éppen olyan sikeresnek tekinthető, mintha ellopta volna azokat. Napjaink legveszélyesebb támadásai, a DDoS (Distributed Denial of Service) támadások, pontosan ezt célozzák meg.

A CIA háromszög tulajdonságaira úgy kell tekintenünk, mint egy háromlábú szék lábaira. A háromlábú szék eldől, ha egyik lába hiányzik. Itt is hasonló a helyzet. Mindhárom tulajdonságot meg kell valósítani a biztonság fenntartásához.

2.2.5. DAD modell

A CIA modell azokat a tulajdonságokat írja le, amelyeket a védelmi oldalnak meg kell valósítania a biztonság érdekében. Ezzel párhuzamosan a támadói oldalon megalkották a DAD modellt, amely a sikeres támadás tulajdonságait nevesíti. [7]



3. ábra DAD modell

A rövidítés az alábbi fogalmak angol kezdőbetűiből adódik:

- *Leleplezés* (Disclosure)

A leleplezés akkor valósul meg, amikor egy jogosulatlan személy illetéktelenül fér hozzá adatokhoz. Ebben az esetben a rendszer bizalmassága sérül.

- *Módosítás* (Alteration)

  Amikor a biztonsági megoldások nem tudják biztosítani az adatok, illetve a rendszer integritását, akkor módosítás történik. Ennél a pontnál meg kell említenünk, hogy engedély nélküli módosítás történhet a rendszerben (a) szándékosan és (b) véletlenül is. Például:

  - az egyik alkalmazott (esetleg képzetlensége miatt) véletlenül rekordokat töröl ki az adatbázisból (a),

  - egy hacker módosítja az általunk elküldött elektronikus levelek tartalmát (b).

  Természetesen a szándékos és a nem szándékos, engedély nélküli módosítások ellen is védekezni kell. Utóbbi esetet megelőzhetjük az alkalmazottak oktatásával, illetve körültekintő hozzáférés-szabályozás implementálásával.

- *Akadályozás* (Denial)

  Akadályozás olyan esetben történik, amikor egyes események megakadályozzák a jogosult felhasználókat az adatokhoz, illetve az informatikai rendszerhez való hozzáférésben. Ilyen esemény lehet egy egyszerű áramszünet vagy egy számítógép meghibásodása, de ide tartozik egy rosszindulatú, szolgáltatásmegtagadásos DoS (Denial of Service) támadás is.

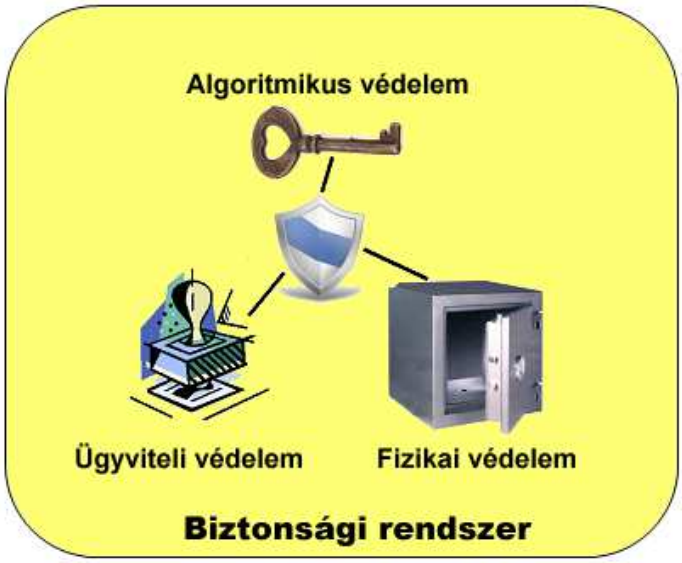
2.2.6. A biztonság három alappillére

A biztonság az információs technikák alkalmazhatóságának alapvető kritériumává vált. E biztonság megteremtését az logikai (algoritmikus), fizikai, illetve adminisztratív (ügyviteli) védelem kombinálásával érhetjük el.

Ebbe a három csoportba sorolhatók a védelmi technikák legkülönbözőbb lehetőségei, ezért ezeket nevezzük a biztonság három alappillérének.

- *Algoritmikus védelem*

  Szokás még logikai védelemnek is nevezni. Ide tartoznak mindazon szoftverkomponensek, amelyek a védelem során felhasználhatók. Különböző kriptográfiai módszerek, amelyek biztosítják az adatok titkosságát, hitelességét úgy, hogy a védtelen közegben elhelyezkedő adatokat titkosítják, illetve a védett entitások védtelen közegen keresztüli kommunikációját biztosítják. E védelmi pillér alapeszközei a szimmetrikus kulcsú és nyilvános kulcsú rejtjelezők, kriptográfiai hash függvények. Ide tartoznak a tűzfalak, a behatolásjelző és megelőző rendszerek, AAA keretrendszer, hozzáférés-szabályozás stb. Ezekről a technikákról a későbbi fejezetekben részletesebben is olvashatunk.



4. ábra A biztonság alappillérei

- *Fizikai védelem*

  A fizikai védelem**Hiba! A könyvjelző nem létezik.** alatt az informatikai rendszer szoftvereinek, hardvereinek és teljes infrastruktúrájának fizikai védelmét értjük. Az eszközöket védeni kell a természeti és humán behatásoktól. Természeti behatások például a villámcsapás, de a szerverterem árvíz elleni védelme is ide tartozik. Humán behatásnak tekinthető minden olyan emberi beavatkozás, ami fizikailag veszélyezteti a rendszert. Ilyen lehet például a betöréses lopás vagy akár egy kommunikációs kábel elvágása is. Ma már a fizikai védekezésre nagyon különleges megoldások is léteznek (például egyes katonai rendszerekben a kommunikációs kábeleket gázzal töltött védőcsővel veszik körül, amelyek felvágása nyomáseséssel jelzi a behatolási kísérletet), de olyan kézenfekvő lehetőségekről sem szabad megfeledkezni, mint például erősített páncélajtó használata a szerverszobákban vagy biztonsági őrök alkalmazása.

- *Adminisztratív védelem*

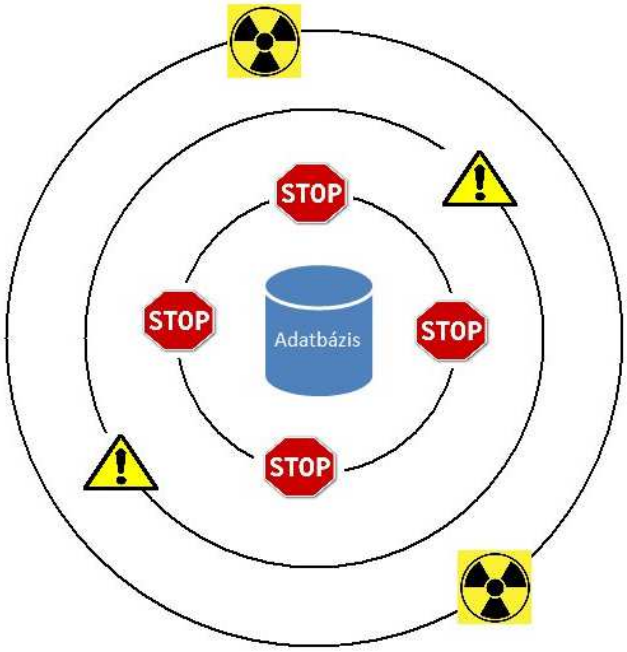
  A harmadik védelmi pillér az ügyviteli védelem**Hiba! A könyvjelző nem létezik.**, amelyre méltatlanul kevesebb hangsúlyt fektetnek, mint a fenti kettőre. Ügyviteli védelem alatt egy szervezet működési szabályainak összességét értjük. Összekapcsolja a fizikai védelmet az algoritmikus védelemmel, és ez a biztonsági pillér**Hiba! A könyvjelző nem létezik.** felelős a rendszerben jogosult entitások (emberek vagy akár programok) tevékenységeinek dokumentálására. A rendszabályokban határozzák meg a felelősségi köröket és a jogosultságokat.

Egy rendszer akkor tekinthető biztonságosnak, ha a biztonság mindhárom alappillérét körültekintően megvalósítja. Ha egy pillér hiányzik, akkor már borul az egész biztonság, hiszen lehet egy rendszer „atombiztosra” tervezve fizikai és algoritmikus szempontból, ha egy jogosult rendszergazda nehézségek nélkül módosíthat szenzitív adatokat.

2.2.7. Defense-in-depth stratégia

Úgy gondolom, a korábbi fejezetek alapján senkinek sincsenek kétségei – és ha voltak, akkor már szertefoszlottak – az adatok védelmének fontosságáról. Egy informatikai rendszer legfontosabb értékei az adatok. Elsősorban ezek védelmére kell koncentrálni.

A védelmi stratégia kidolgozásakor az adatokat nem tekinthetjük külvilágtól elszigetelt objektumnak, hiszen azok kapcsolatban állnak más rendszerekkel, alkalmazásokkal, amelyek sokszor hálózati kapcsolaton keresztül veszik igénybe a szolgáltatásokat. Az adatok legtöbbször valamilyen adatbázisban vagy fájlrendszerben tárolódnak, amelyek természetesen operációs rendszerek felügyelete alatt állnak. Nyilvánvaló, hogy az adatok biztonságát nem lehet függetleníteni az adatbázis operációs rendszerének biztonságától. Számos olyan rendszer van, amely kihatással van egy adatbázis működésére, ezért a hatékony védekezéshez integrálni kell a biztonsági stratégiát. Azt is tudomásul kell venni továbbá, hogy tökéletes biztonság nem létezik. Amíg az emberi tényező szerepet játszik az informatikai rendszerek tervezésében, megvalósításában és üzemeltetésében, addig hibák is lesznek, amelyeket a támadók megpróbálnak kihasználni.



5. ábra Defense-in-depth stratégia

A fenti okok miatt ma már nem egy „szuper biztos” rendszer építése a cél, hanem helyette az úgynevezett *mélységi védelem**Hiba! A könyvjelző nem létezik.*** (defense-in-depth**Hiba! A könyvjelző nem létezik.**) stratégiája vált meghatározó iránnyá. Ez azt jelenti, hogy a védelmet egy többrétegű biztonsági rendszeren keresztül valósítjuk meg.

Ahogyan az ábrán is látható, az adatokat a biztonsági rétegek magjában kell elhelyezni. A magot övező rétegek a legkülönfélébb biztonsági eszközök megvalósításával védelmezik az adatokat. Egy ilyen stratégia új kihívások elé állítja a hackereket, hiszen egy biztonsági réteg feltörésével a rendszer még nem válik védtelenné. Ha a biztonsági stratégia része egy jól

működő audit rendszer, akkor betörési kísérlet gyanúja esetén időben – még a károkozás előtt – megtörténhet az illetékes személyek értesítése.

A biztonsági rétegek számát és funkcióját a konkrét informatikai rendszerhez kell igazítani, de általában az alább felsorolt eszközök állnak rendelkezésre:

- *Felhasználó azonosítása (authentication**Hiba! A könyvjelző nem létezik.**)*

  Célja megbizonyosodni arról, hogy a rendszerrel kapcsolatban álló felhasználó valóban az, akinek mondja magát. Erre alapvetően három lehetőség van. Az első a tudásalapú azonosítás, amely olyan információra épül, amelyet csak egy jogosult személy ismerhet (pl.: jelszó). A második a birtokalapú azonosítás, ami feltételezi, hogy a jogosult felhasználó birtokában van egy fizikai vagy logikai kulcs (pl.: mágneskártya). Végül napjainkban egyre népszerűbb a biometriai azonosítás, ami az ember biológiai jellemzői alapján ellenőrzi a felhasználót (pl.: ujjlenyomat, írisz). Kritikus biztonságú rendszerekben a fentiek közül legalább két módszer egyidejű alkalmazása javasolt.

- *Hozzáférés engedélyezése (authorization)*

  Feladata annak biztosítása, hogy az egyes erőforrásokhoz csak a megfelelő jogosultsággal rendelkező felhasználók férjenek hozzá.

- *Audit*

  A biztonsági rétegek összefogásáért és a tevékenységek naplózásáért felelős réteg.

- *Tűzfal (firewall**Hiba! A könyvjelző nem létezik.**)*

  A tűzfalak egy szervezet informatikai rendszerének határán működnek, elválasztva a külvilágot a belső rendszertől. Feladatuk megakadályozni a külvilágból (pl.: internetről) érkező fenyegetéseket.

- *Antivírus*

  Az antivírusok az informatikai rendszerek szoftveres és hardveres eszközeit védik a rosszindulatú kódoktól (pl.: vírusok, kémprogramok).

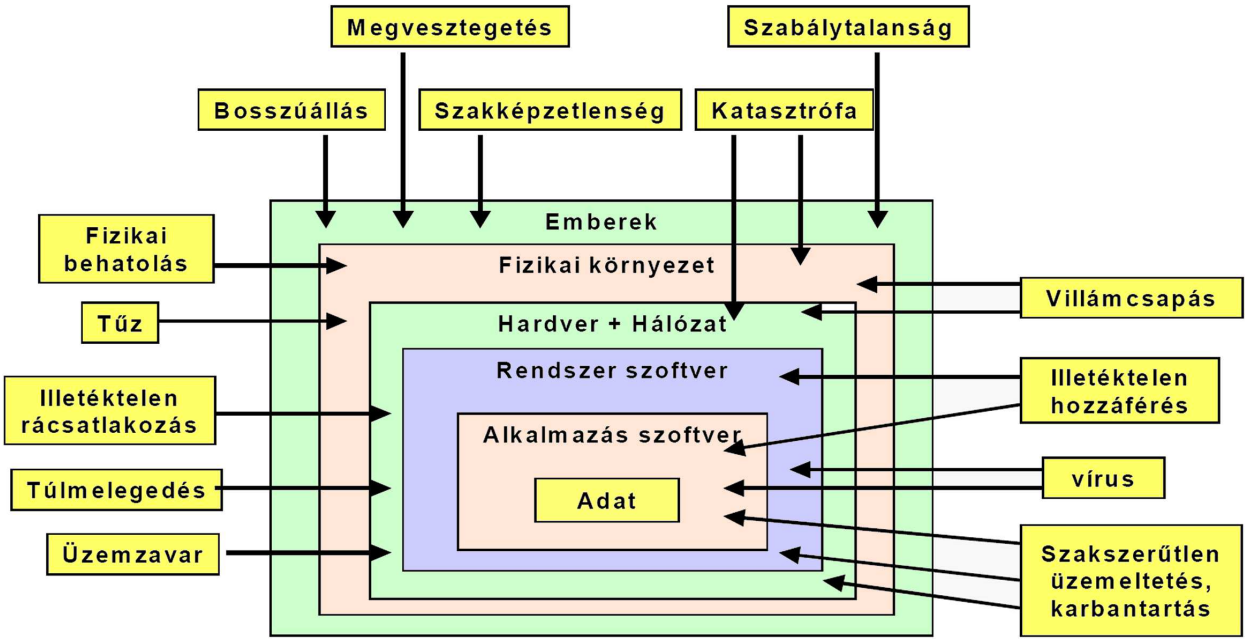
- *Behatolásjelző rendszer (IDS**Hiba! A könyvjelző nem létezik.**, intrusion detection system)*

  Feladata, hogy a rendszer folyamatos vizsgálatával olyan jeleket keressen, amelyek behatolásra utalhatnak. Technikáik között szerepelnek a különféle naplók vizsgálatai, de képesek akár a hálózati forgalom csomagszintű monitorozására is (hálózati behatolás jelzése).

2.3. *Veszélyek*

Ahhoz, hogy biztosítani tudjuk az informatikai rendszert és az abban tárolt adatokat, tudnunk kell, hogy milyen veszélyekkel kell szembenéznünk. A „legizgalmasabb” veszélyt a szándékos és illetéktelen behatolások (támadások) jelentik. Ezt információs támadásnak nevezzük. Látnunk kell azonban, hogy a veszélyek nagy része nem ebbe a kategóriába tartozik. Olyan veszélyekre is fel kell készülnünk, mint a természeti katasztrófák, civilizációs és ipari katasztrófák, terrorizmus és az infrastruktúrák teljesítőképességének kimerülése.

Az alábbi ábra a teljesség igénye nélkül mutatja be az informatikai rendszert veszélyeztető tényezőket.



6. ábra Az informatikai rendszerek biztonságát befolyásoló tényezők[3]

2.3.8. Dimenziók

A veszélyek csoportosításakor jó kiindulási pont a veszély dimenziójának meghatározása. Az alábbi dimenziókból származnak a veszélyek:

- *Fizikai dimenzió (hard)*

  Az informatikai rendszer fizikai elemeit veszélyeztető tényezők tartoznak ide.

- *Információs dimenzió (soft)*

  Az információs folyamatokat érintő, elektronikus úton fenyegető veszélyek.

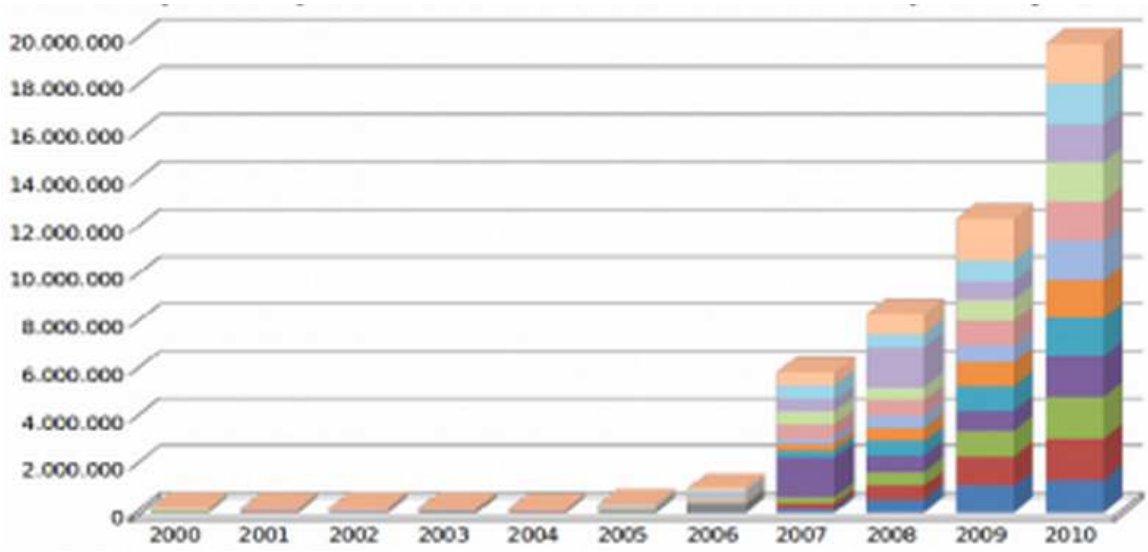
[3] Forrás: Bodlaki Ákos, Csernay Andor, Mátyás Péter, Muha Lajos, Papp György, Vadász Dezső: Informatikai Rendszerek Biztonsági Követelményei: MeH ITB 12. sz. ajánlás

- *Kognitív (tudati) dimenzió*

  Az ember gyengeségeiből és hiányosságaiból eredő veszélyek.

2.3.9. Malware

Az előző fejezetben áttekintettük a veszélyeztető tényezők körét. Most vizsgáljuk meg közelebbről a szándékos támadások egyik típusát, amelynek száma az elmúlt években töretlenül növekszik (ezt tanúsítja az alábbi ábra is). Ezek a rosszindulatú szoftverek, vagy közismertebb nevükön, a malware-ek. A malware szó a malicious software (rosszindulatú szoftver) szavak összefésüléséből jött létre. Ez nem egy konkrét szoftvert jelent, hanem különböző típusú kártevők gyűjtőneve.



7. ábra Kártevők számának alakulása (forrás: Kaspersky Lab)

Definíció szerint azt mondhatjuk, hogy a malware olyan szoftverek gyűjtőneve, amelyek kijátszva a védelmi rendszert, képesek illetéktelen hozzáférést biztosítani az informatikai rendszer objektumaihoz. A kártevők ellen védekezni kell, és a legjobb megoldás, ha a rétegzett védelmi modellünkbe elhelyezünk egy vírusvédelmi réteget. Ennek feladata nem más, mint a malware-ek keresése és kiszűrése a rendszerből. A mai vírusirtók a mintakeresésen alapulnak, azaz olyan jellegzetességek után kutatnak a rendszerben, amelyek megtalálhatók a vírus-adatbázisokban.

A malware-ek elsősorban a hálózaton keresztül érkeznek az informatikai rendszerbe (például egy e-mail csatolmányaként), de gyakran előfordul az is, hogy cserélhető lemezen keresztül jutnak be. Az algoritmikus védelmi pillér mellett fontos, hogy ügyviteli eszközökkel is védekezzünk a malware-ek ellen, azaz olyan rendszabályokat kell hozni, amelyek például megtiltják ismeretlen pendrive-ok használatát az alkalmazottak számára.

Az alábbiakban a leggyakoribb malware-ek jellegzetes tulajdonságait tekintjük át.

Vírusok

A vírusok a legrégebbi és leggyakoribb malware-ek. A köznyelv gyakorta a vírus kifejezést használja az összes kártevőre. Ezt példázza a vírusirtó kifejezés is, amely természetesen nemcsak vírusok irtására szolgál, hanem minden malware típus ellen használható.

A vírus egy egyszerű program, amely más programokat, úgynevezett gazdaprogramokat (virus host) keres annak érdekében, hogy megfertőzze őket. Ez a folyamat pontosabban azt jelenti, hogy a gazdaprogramba beágyazza a vírus a saját kódját. Amikor a gazdaprogramot futtatják, akkor a vírus kódja is lefut és végrehajtja azokat a műveleteket, amelyre beprogramozták. A különböző típusú vírusok különböző műveleteket hajtanak végre (adatokat törölnek, módosítanak, terjesztik magukat…), egyetlen közös tulajdonságuk, hogy gazdaprogramra van szükségük, a nélkül nem életképesek és nem tudnak terjedni.

Az első vírusokat még az 1970-es években készítették, a világ legelső ismert vírusa a Creeper volt, amely a Tenex operációs rendszert használó számítógépek hálózatán terjedt. A Creeper kiirtására hozták létre a Reeper nevű programot, ez az első ismert vírusirtó. A „számítógépes vírus” kifejezést először a neves elméleti víruskutató, Fred Cohen használta 1983-ban, egy tudományos munkában.

Férgek

A férgek (worms) abban különböznek a vírusoktól, hogy ezek önálló programok, nincs szükségük gazdaprogramra. Miután egy féreg futtatása elkezdődik, már úgy viselkedik, mint egy vírus: az a célja, hogy jogosulatlanul hozzáférjen adatokhoz és jogosulatlanul tevékenykedjen.

Logikai bombák

A logikai bombák műveletek sorozatát hajtják végre, amikor egy speciális esemény bekövetkezik. Ilyen események lehetnek bármely rendszeresemények. Például egy adott dátum elérése vagy a számítógép leállítása a 666. alkalommal, kiválthatja a logikai bombák „robbanását”.

Trójai falovak

A trójai faló (torjan horse) olyan malware típus, amely leginkább a férgekhez hasonlítható. Ez is egy önálló program, de első ránézésre veszélytelen, sőt úgy tűnik, mintha a segítségünkre is lenne. Amikor azonban elindítjuk, akkor nemkívánatos tevékenységeket végez a rendszerben. Gyakori eset, hogy a trójaiak vírusirtónak álcázzák magukat.

2.3.10. Egy malware bemutatása

Napjaink legveszélyesebb vírusai a zsaroló vírusok (ransomware) családjába tartoznak. Ilyen vírus a Cryptolocker is.

A ransomware egy olyan típusú malware, amely megfertőzi és korlátozza a hozzáférést a számítógépes rendszerhez és/vagy annak fájljaihoz. Pénzt követel a felhasználótól annak érdekében, hogy a korlátozásokat feloldja. Néhány formája a zsaroló vírusoknak titkosítja a

fájlokat a merevlemezen, míg néhány egyszerűen zárolja a rendszert, majd üzeneteket jelenít meg az eszközön, melyeknek célja rávenni a felhasználót a „váltságdíj” kifizetésére.

A Cryptolockert 2013 szeptemberében detektálták először. Csak Microsoft Windows (Windows 8 – 8.1, Windows 7, Vista és XP) operációs rendszerek vannak kitéve ennek a támadásnak. A Cryptolocker legtöbbször e-maileken keresztül terjed. Ezek a levelek általában befizetésekkel, vásárlásokkal, adózással kapcsolatos üzeneteknek és más hivatalosnak tűnő értesítéseknek mutatják magukat. Amint a csatolt fájlt megnyitja a felhasználó, a kártevő a rendszerbe férkőzik. Terjed még botnet hálózatokon, torrent és warez oldalakról letöltött fájlok futtatásakor vagy (feltört) weboldalakon lefutó támadó kódokkal, amik a fertőzéshez kihasználják az operációs rendszer vagy a böngésző biztonsági réseit. Speciális esetben USB és más adathordozó eszközökkel is terjedhet.

8. ábra Cryptolocker vírusüzenet (forrás: computerworld.com)

Amikor a Cryptolocker a rendszerbe férkőzik, akkor a háttérben futva RSA titkosítási algoritmussal, a támadó szerverén található adatbázisból kiválasztott publikus kulccsal titkosítja a számítógépen található személyes fájlokat, majd törli az eredetieket. Minden felcsatlakoztatott adathordozón és hálózaton megosztott területen is lefut a titkosítási procedúra. Miután végzett, egy ablakban tájékoztatja a felhasználót a támadásról, és pénzt követel a feloldáshoz nélkülözhetetlen privát kulcsért.

Mivel 2048 bites egyedi RSA-kulccsal dolgozik, a dekódolás a privát kulcs nélkül (szinte) lehetetlen. Ha ez még nem volna elég, a megjelenő ablak bal oldalán megtalálható egy számláló is, ami 72 órától számol visszafelé. A program azt a tájékoztatást közli, ha a számláló lejár, a rendszerükből törlik a privát kulcsot, így az érintett adataink végleg titkosítva maradnak, és soha többé nem tudjuk majd feloldani azokat.

A „váltságdíj” összege verziófüggően lehet 100–600 USD/EUR, ami a határidő lejártának közeledtével nőhet is.

A zsaroló vírusok terjesztése jövedelmezőnek tűnik (már amennyire a bűnözés az lehet). Mi sem bizonyítja jobban, minthogy a Cryptolocker megjelenése óta eltelt egy év alatt

világszerte több mint egymillió dollárt sikerült kicsalni a szerencsétlenül járt felhasználóktól. Ez az összeg soknak tűnik, főleg ha azt is figyelembe vesszük, hogy a megtámadott felhasználók elenyésző hányada fizeti ki a követelt összeget.

2.3.11. Egyéb támadások

Az alábbiakban néhány tipikus támadást tekintünk át, amelyeket gyakorta kell felismernünk és kezelnünk a védelmi oldalon.

Hátsó ajtó (back door)

A hátsó ajtó egy olyan speciális hozzáférési pont, amelyet a programozók gyakran elhelyeznek programjaikban. Alapvető célja megakadályozni, hogy a fejlesztők kizárják magukat saját programjaikból. Az egyik probléma ezekkel a „kiskapukkal”, hogy a fejlesztő a program eladása után is képes láthatatlanul hozzáférni adatokhoz, azaz a fejlesztőnek megmarad a képessége jogosulatlanul hozzáférni a rendszerhez.

A másik probléma az, hogy ezeket a hátsó ajtókat nemcsak a fejlesztők, hanem egyéb támadók is ki tudják használni. Az ilyen jellegű kártevőkkel szemben a legjobban úgy tudunk védekezni, hogy megbízható fejlesztők szoftvereit használjuk, illetve teszteljük a programokat, mielőtt élesben használnánk őket.

Puffer túlcsordulás

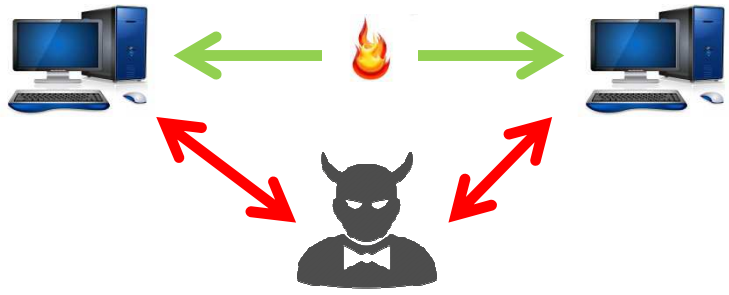
A puffer túlcsordulásos támadásokat a figyelmetlen szoftverfejlesztések teszik lehetővé. Alapvetően akkor történik puffer túlcsordulás, ha kódunkból olyan memóriaterületre címzünk, amelyre nem lenne szabad. Például 110 bájt adatot másolunk egy 100 bájtos pufferbe.

Egy puffer túlcsordulás általában kivételt okoz és leáll a program működése. Ha a támadó körültekintően helyezi el káros kódját a túlcímzett memóriaterületre, akkor akár képes lesz malware bejuttatására is, illetve képes lehet átvenni a célpont feletti vezérlést.

A puffer túlcsordulásos támadások nagyon népszerűek, mert rengeteg program sérülékeny ilyen módon.

Közbeékelődéses támadás (man-in-the-middle attack)

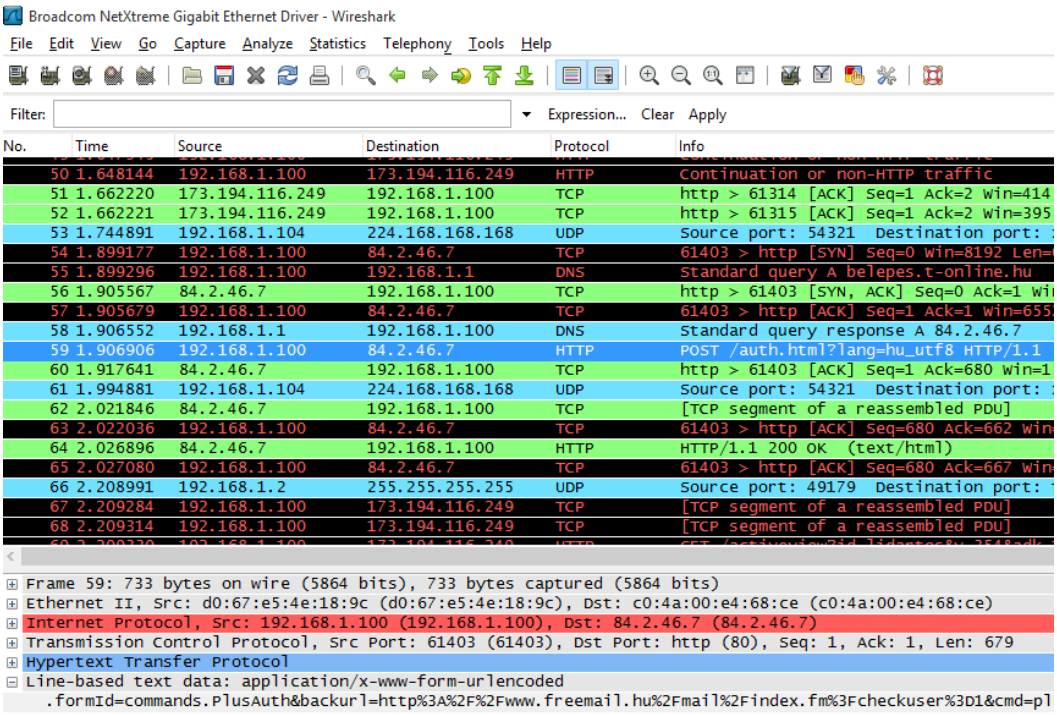
A támadás során a támadó lehallgatja a kommunikációs partnerek közötti üzenetcserét, majd elfogja az üzeneteket. Az elkapott üzeneteket módosítva vagy egyszerűen csak hamisított üzenetek összeállításával más nevében kommunikál.



9. ábra Közbeékelődéses támadás

Lehallgatás

A támadók előszeretettel monitorozzák a számítógép-hálózatok forgalmát. Erre különösen nagy esélyük van vezeték nélküli hálózatok esetében. Amennyiben a hálózaton titkosítatlan formában forgalmazunk adatokat, azok illetéktelenek kezébe juthatnak. Az alábbi ábrán egy szabadon elérhető[4] protokoll analizátor képernyőképe látható, miközben egy levelező rendszerbe történő belépés szenzitív adatait monitorozza.



10. ábra Lehallgatás Wiresharkkal

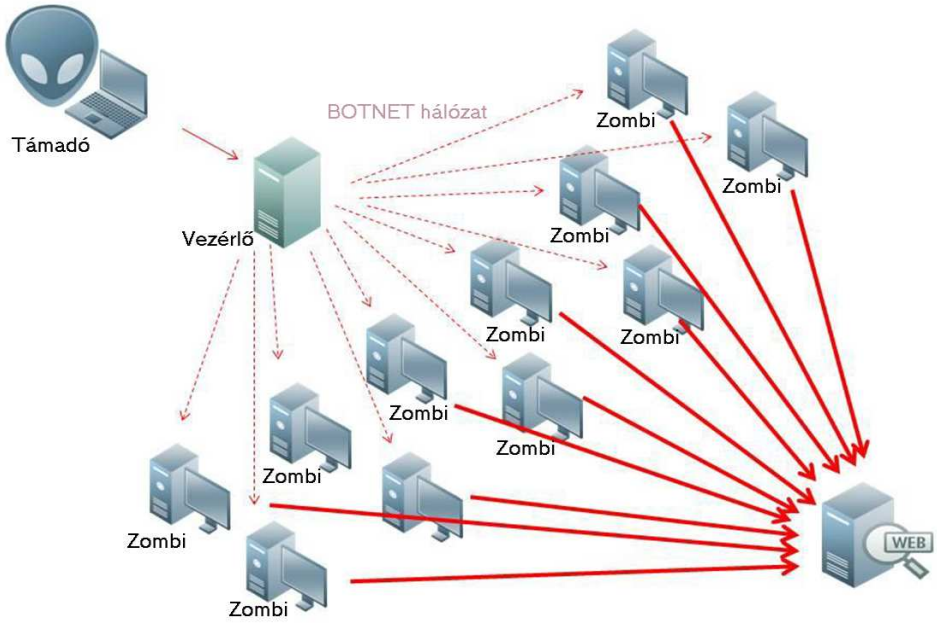
DoS és DDoS támadás

A DoS (Denial of Service) támadás lényege, hogy a támadó(k) a célpontot túlterhelik és megakadályozzák annak normális működését. Ezt úgy érik el, hogy a célpontot – például egy

[4] http://www.wireshark.org

WEB szervert – folyamatosan kérésekkel bombázzák. Ezt addig teszik, amíg végül a szerver képtelen lesz kiszolgálni más (jogosult) felhasználók kéréseit.

A DoS támadás továbbfejlesztett változata az elosztott szolgáltatás megtagadásos támadás, az úgynevezett DDoS (Distributed Denial of Service) támadás. A DoS-hoz hasonlóan itt is a célrendszer túlterhelése a cél, de ebben az esetben nem egy, hanem több támadó terheli kérésekkel a szervert. A támadók csoportja rendszerint egy botnet hálózat fertőzött gépeit jelenti, amelyeket zombiknak nevezünk. A támadást az alábbi ábra illusztrálja.



11. ábra DDoS támadás

Social engineering[5]

A social engineering kifejezésnek nincsen megfelelő magyar fordítása. A fogalom annyit takar, hogy az informatikai rendszer jogosult felhasználóinak gyengeségeit kihasználva lehet jogosulatlan hozzáférést szerezni a rendszerhez. A támadás az embert, mint a biztonsági rendszer leggyengébb láncszemét, használja ki. A támadás többnyire megtévesztéssel és az emberi hiszékenység kihasználásával éri el célját.

Tipikus példája a támadásnak, amikor a támadó telefonon felhívja az egyik alkalmazottat és úgy mutatkozik be, mint a cég új rendszergazdája. Megpróbál az áldozat bizalmába férkőzni, majd kicsalja tőle a jelszavát.

A teljesség igénye nélkül ide tartoznak az alábbi technikák:

- megszemélyesítés, segítség kérése,
- shoulder surfing, piggybacking,
- dumpster diving.

[5] A témáról bővebben Kevin Mitnick könyveiben olvashatunk.

3. Logikai védelem

Az előző fejezetben láthattuk, hogy az informatikai rendszer és az abban tárolt adatok védelme különösen fontos feladat. Függetlenül attól, hogy a felhasználók közvetlenül vagy közvetett módon kerülnek kapcsolatba a rendszerrel, az alábbi három kérdést feltétlenül tisztázni kell:

- Ki akar hozzáférni a rendszerhez?
- Mit tehet a felhasználó a rendszerben?
- Milyen műveleteket hajtott végre a felhasználó a rendszerben?

E három kérdéssel foglalkozik az AAA keretrendszer (ejtsd: „három A” keretrendszer). A kifejezés az angol nyelvű hitelesítés (authentication), feljogosítás vagy authorizáció (authorization) és naplózás (audit vagy accounting) szavak kezdőbetűiről kapta a nevét. A fentieket úgy is megfogalmazhatjuk, hogy az AAA keretrendszer olyan folyamatok összessége, amely megoldja a felhasználó hitelesítését, ellenőrzi és rögzíti annak tevékenységét.

A folyamat könnyebb megértéséhez tekintsük azt a hétköznapi példát, amikor egy banki ügyfél készpénzt szeretne felvenni személyesen a bankfiókban.

Első lépésben az ügyintéző megbizonyosodik arról, hogy ki az, aki ügyet szeretne intézni. Elkér valamilyen személyazonosításra szolgáló okmányt és különböző adatok (név, születési hely és idő…) segítségével igyekszik beazonosítani az ügyfelet (ellenőrzi, hogy a bank ügyféladatbázisában van-e ilyen ügyfél). Az ügyfél beazonosításának része az is, hogy a fényképes igazolvány segítségével ellenőrzik, az ügyfél valóban az a személy-e, akinek mondja magát. Ez a hitelesítés folyamata.

Második lépésben az ügyfél elmondja, pénzt szeretne felvenni a számlájáról. Ekkor az ügyintéző ellenőrzi, hogy rendelkezésre áll-e az összeg a számlán és jogosult-e felvenni azt az összeget (mert például egy bizonyos limitösszeg felett nem vehet fel pénzt). Ez az authorizáció folyamata.

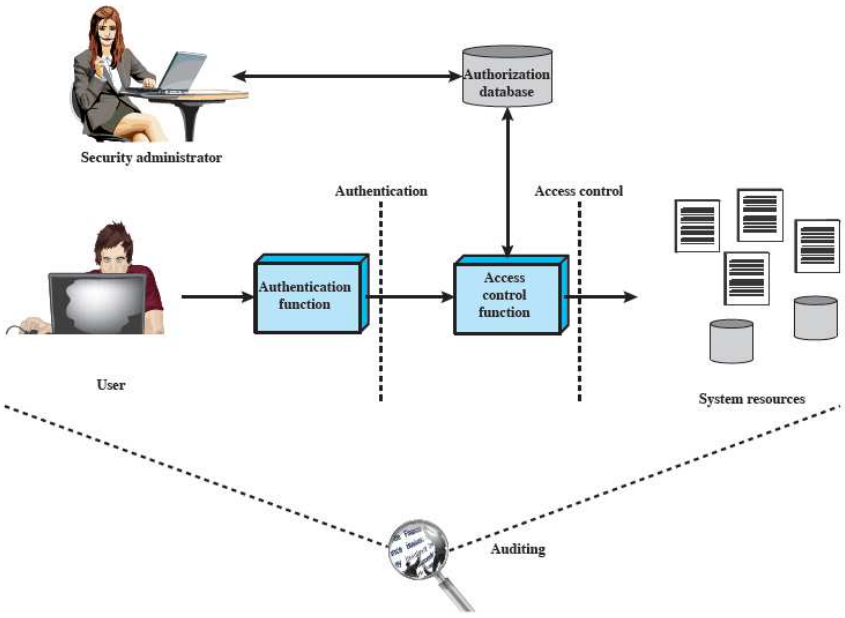
A készpénzfelvétel során a bank rögzíti a tranzakció időpontját, a felvett összeget, az ügyintéző nevét… és minden körülményt, ami fontos lehet egy későbbi ellenőrzéshez. Ez az audit folyamatának felel meg.

Ebben a fejezetben áttekintjük az informatikai rendszerekben alkalmazott hitelesítés, feljogosítás és naplózás elméleti hátterét, valamint bepillantunk a gyakorlati megoldásokba is.

3.1. AAA rendszer

Már a betárcsázós internetelérés idején is nagy szükség volt arra, hogy például a szolgáltató képes legyen egyértelműen meghatározni, ki vett igénybe egy adott szolgáltatást, mennyi ideig tartott az. Az is fontos, vajon a megfelelő szintű szolgáltatást kapta-e az ügyfél. Mostanra már az egyre szegmentálódó hálózati topológiák, a vezeték nélküli hozzáférési pontok és az ezekhez csatlakozó rengeteg mobileszköz egyre nagyobb elvárásokat támaszt az AAA-val szemben.

Az AAA keretrendszertől elvárt funkciók azonosításához tekintsük meg az alábbi ábrát. Első lépésben a felhasználó kapcsolatba lép az informatikai rendszerrel, amely elvégzi a felhasználó hitelesítését. Ezután a hitelesített felhasználó különböző tevékenységeket szeretne végrehajtani a rendszerben (fájlokhoz szeretne hozzáférni, nyomtatni szeretne stb.). Természetesen a különböző felhasználóknak más és más jogosultságaik lesznek. Egyes felhasználók jogosultak hozzáférni egy fájlhoz, míg másoknak erre nem szabad lehetőséget biztosítani. Ezen a ponton azonosíthatjuk a második fontos funkciót, amely a hitelesített felhasználók tevékenységét kontrollálja, azaz a rendszer objektumaihoz való hozzáférést vezérli. Végül, de nem utolsósorban a felhasználók által végrehajtott tevékenységek rögzítését is el kell látni.



12. ábra AAA keretrendszer (forrás: [4])

A fenti funkciók ellátásáért felelős az AAA keretrendszer, amely az alábbi építőelemekből tevődik össze:

- *Kliens:* Az a fél, legyen természetes személy vagy gép, aki/ami csatlakozni szeretne a rendszerhez.

- *Hitelesítő:* Más néven PEP (Policy Enforcement Point). Olyan eszköz, amely kezeli a kliens hozzáférési kérését. Lehet útválasztó, tűzfal, VPN (Virtual Private Network) eszköz, Ethernet switch vagy Wifi hozzáférési pont.

- *PIP (Policy Information Point):* Olyan rendszer, illetve adatbázis, amely a felhasználók vagy eszközök azonosítóit, és egyéb hozzáférési kérelmek számára releváns információkat tárol. Ez lehet LDAP vagy OTP Token Server is.

- *PDP (Policy Decision Point):* Ez maga az AAA szerver, amely a kapcsolódási kérelmeket elbírálja és dönt, hogy melyeket engedélyezi. A kapcsolatfelvételekről információt gyűjt a PEP-en keresztül, ezenkívül egyéb adatokat is figyelembe vesz a PIP-től, amelyek segíthetik egy-egy döntésben. Miután sikerül döntenie, visszaküldi végrehajtásra a megfelelő információkat, beállításokat a PEP-nek.

- *Monitorozó és naplózó rendszer:* A fenti funkciók fontos kiegészítője. Feladata, hogy a most már kontrollált hozzáférésekről összegyűjtsön minden lehetséges információt. Így már követhető, hogy ki, mikor és hol csatlakozott a hálózathoz. Megvalósulhat külön rendszerként is vagy akár a PDP-vel egybeépítve. Gyakran a különböző komponensek is keverednek egymással: PEP–PDP, PDP–PIP.

3.2. Authentikáció

Az RFC2828 alapján „az authentikáció az a folyamat, amelynek során ellenőrizzük a felhasználó identitását és azt, hogy hozzáférhet-e a rendszerhez”.

A korábbiakban bemutatott banki példát felhasználva, hasonló helyzetet próbáljunk elképzelni az informatika oldaláról megközelítve. A banki adatbázisban a felhasználókat egyedi azonosítóval (ID, identifier) különböztetik meg egymástól. Az azonosítási (identification) folyamat során meg kell határozni, hogy a felhasználóhoz milyen egyedi azonosító tartozik a számítógépes rendszerben, majd a hitelesítési folyamat során a felhasználó – hitelt érdemlően – bizonyítja, hogy az azonosító hozzá tartozik.

A következőkben arra az esetre koncentrálunk, amikor egy ember próbálja magát hitelesíteni egy számítógépes rendszer előtt. Meg kell jegyezni azonban, a gyakorlatban szükség van arra is, hogy a számítógépes rendszer hitelesítse magát az ember számára, illetve a számítógépek egymás közötti hitelesítéséről sem feledkezhetünk meg. A számítógép–ember közötti hitelesítésre jó példa lehet egy online bank weboldala, ahol a felhasználó szeretne megbizonyosodni arról, hogy a weboldal valóban a saját bankjához tartozik és nem egy hamisított adathalász weboldal. Az ilyen jellegű azonosításról a kriptográfiáról szóló fejezetben lesz szó.

A felhasználók azonosításakor az alábbi négy lehetőség közül választhatunk:

- Tudás (valami, amit csak a felhasználó tud).

- Tulajdon vagy birtok (valami, ami csak a felhasználónál van).

- Tulajdonság (a felhasználóra jellemző egyedi biológiai tulajdonság).

- Képesség (valami, amit csak a felhasználó képes elvégezni).

13. ábra Felhasználó-azonosítás eszközei

Mind a négy módszer arra a feltételezésre épít, hogy csak a jogosult személy van birtokában annak a tudásnak, tulajdonnak, tulajdonságnak vagy képességnek, amely bizonyítja az ő személyének valódiságát.

A fenti megoldások önmagukban is képesek betölteni a felhasználó-azonosítás funkcióját, de kiemelten fontos rendszerek esetében javasolt ezek kombinációját alkalmazni. A kombinált megoldást használó rendszerekben ún. többlépcsős hitelesítés (multifactor authentication) történik. A mindennapi életben is találkozunk ilyen rendszerekkel. Gondoljunk csak a bankjegykiadó ATM automatákra, amelyek csak akkor adnak készpénzt, ha behelyezzük a bankkártyánkat (tulajdonalapú azonosítás) és megadjuk a hozzá tartozó PIN kódot (tudásalapú azonosítás). A korábbi gondolatmenetet folytatva azt mondhatjuk, hogy a bankjegykiadó esetében kétlépcsős hitelesítésről (two-factor authentication) van szó.

3.2.1. Tudás

A legáltalánosabban alkalmazott azonosítási módszer, amely a felhasználó tudatában lévő információra épít. Széles körű alkalmazásának oka, hogy az ilyen azonosító rendszerek nem igényelnek semmilyen különleges hardvert, illetve kiépítésük sem bonyolult.

Ebbe a kategóriába tartoznak a PIN (Personal Identifier Number) kódok is, ahol az azonosító egy legalább 4 jegyű kód. Leginkább a bankjegykiadó automatáknál, beléptető rendszerekben és a mobiltelefon-hálózatok SIM kártyáinál használatosak. Mivel a leggyakrabban alkalmazott négy számjegyes esetben tízezer különböző lehetőség adódik, ezért ez a rendszer nagyon érzékeny a találgatásos támadásokkal szemben.

A tudásalapú felhasználó-azonosítási metódusok körét bővítik a jelszavak is. A PIN kódokhoz képest a jelszavak már sokkal nagyobb bonyolultságot tesznek lehetővé, hiszen tetszőleges számú tetszőleges karakterből állhatnak.

A jelszavas felhasználó-azonosítási rendszerekkel kapcsolatban számtalan probléma merülhet fel, amelyekkel jó, ha tisztában vagyunk. Ezek a következők:

- *Illetéktelenek kezébe kerül a jelszó*

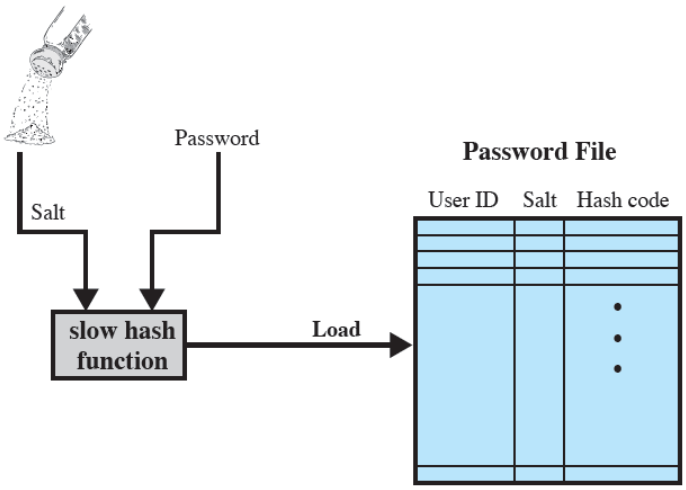
  Talán a legnagyobb probléma a jelszavakkal kapcsolatban az, hogy ellophatók anélkül, hogy a tulajdonos tudna róla. Ha valaki meglesi a billentyűzetet a jelszó begépelése közben, a tulajdonosnak nem fog feltűnni, hogy kiszivárgott a titok. Ez azért veszélyes, mert a felhasználó továbbra is ugyanazt a jelszót fogja használni a hitelesítéshez, és közben a támadónak is lehetősége lesz ilyen módon hozzáférni a jelszóval védett rendszerekhez. Azt, hogy a jelszó illetéktelenek kezébe került, csak akkor lehet felismerni, ha a tolvaj felhasználja, és szokatlan tevékenységeket végez a rendszerben. Fontos észrevenni, hogy ezt a problémát kizárólag az anomália-detektáló rendszerekkel lehet kiszűrni, amelyek felismerik, ha egy felhasználó nevében idegenek lépnek kapcsolatba a rendszerrel. Anomália-detektáló rendszerekről a behatolásjelző rendszerek című fejezetben olvashatunk részletesebben.

- *Jelszótárolási problémák*

  A tudásalapú technikák lényege, hogy a hitelesítési információnak csak a jogosult felhasználó tudatában szabad léteznie. Az ember csak korlátozottan képes jelszavak megjegyzésére, ezért (az egyszerű jelszavak használatán túl) a jelszót feljegyzi valahova, például egy papírra. Ettől a ponttól kezdve már nem csak a felhasználó tudatában van a titok, így a rendszer kevésbé biztonságos, hiszen a támadónak elég megszerezni a papírt.

  Másrészről azonban feltétlenül szükség van a jelszó tárolására az informatikai rendszerben is, hiszen a felhasználó által begépelt karaktersorozatot össze kell hasonlítani a hitelesítési adatbázisban tárolt párjával. Kezdő informatikusok gyakorta elkövetik azt a hibát, hogy a hitelesítési jelszót nem titkosított formában tárolják. Ez azért probléma, mert a támadók a rendszer feltörése után könnyedén hozzáférnek a felhasználók jelszavaihoz, másrészt a magas jogosultsági szinttel rendelkező felhasználók, például a rendszergazdák is megismerhetik a felhasználók jelszavait.

  Jól bevált gyakorlat a jelszavak tárolásánál az a megoldás, amikor nem magát a jelszót, hanem annak hash lenyomatát (pl.: MD5 lenyomat) tároljuk.



14. ábra Jelszó hash lenyomata (forrás:[4])

A megoldás tovább finomítható az ábrán látható *salt* technikával, amely kiküszöböli azt is, hogy az azonos jelszavak megkülönböztethetők legyenek egymástól. Ezt úgy érjük el, hogy a hash függvény alkalmazása előtt a jelszót kiegészítjük egy rögzített karaktersorozattal. Így biztosítható, hogy a jelszavak lenyomataiból nem tehet következtetni a jelszavakra még szivárványtáblás támadással sem (rainbow-table attack).

- *Több helyen használt jelszavak*

  Az emberek tucatnyi számítógépes rendszerrel állnak kapcsolatban, amelyekhez általában jelszóval kell hitelesíteni magukat. Az egyszerűség kedvéért a felhasználók gyakorta ugyanazt a jelszót több rendszerben is használják. Ilyenkor viszont fennáll annak a veszélye, hogy ha valamelyik rendszerből kitudódik a jelszó, akkor egyetlen incidens kihatással van a felhasználó többi fiókjára is.

- *Gyenge jelszavak*

  A leggyengébb láncszem általában az ember. A jelszavas hitelesítő rendszerekkel kapcsolatos támadások többsége ezt igyekszik kihasználni. Ha a felhasználóra bízzuk a jelszó megválasztását, akkor jó eséllyel egyszerű (rövid és/vagy könnyen megjegyezhető) jelszavakat fognak választani. Az ilyen jelszavak használata megkönnyíti a támadók dolgát.

  - *Kimerítő kereséses támadás*

    Az angolul brute-force (nyers erő) technikának nevezett megoldás lényege, hogy az összes lehetséges jelszó-kombinációt kipróbáljuk. Könnyen belátható, hogy egyszerű jelszavak esetén nagyon kevés, míg komplex, hosszú jelszavak esetén nagyon hosszú idő kell ahhoz, hogy kitaláljuk a jelszót. E támadási módszer előnye, hogy mindenképpen sikeres lesz.

- *Szótáralapú támadás*

  A felhasználók számára könnyebben megjegyezhetők azok a jelszavak, amelyek értelmes szavakra épülnek.

1	123456
2	password
3	12345
4	12345678
5	qwery
6	123456789
7	1234
8	baseball
9	dragon
10	football

15. ábra A leggyakoribb jelszavak a világon

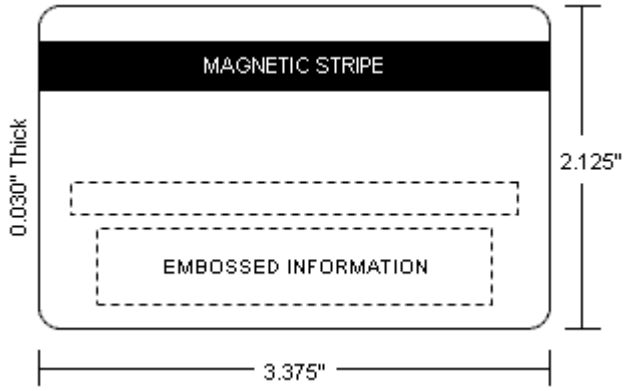
Az értelmes szavak megtalálhatók a szótárakban, és innen kapta a nevét a szótáralapú támadási technika (dictionary attack), amely a leggyakrabban használt jelszavakat tartalmazó listából próbálja végig a lehetséges kombinációkat.

3.2.2. Birtok

Mivel nem kell hozzá az embernek megjegyezni semmit, ezért egy fokkal kényelmesebb és nagyobb biztonsági fokozatot jelentő megoldás, ha az ún. token játssza a kulcsszerepet az azonosításban. Ezt a felhasználónak magánál kell tartania. Általában valamilyen kártyáról van szó, amibe egy memóriaegységet vagy mikroprocesszort integráltak, hogy tároljon, illetve továbbítson adatokat. Használatukhoz minden esetben ki kell alakítani a megfelelő infrastruktúrát az olvasó egységekkel.

Mágneskártya

A mágneskártya adattárolásra alkalmas kártya, amely mágnesezhető felületet tartalmaz. A rajta lévő adatokat egy leolvasó készülék segítségével lehet kinyerni.



16. ábra Mágneskártya (forrás: Wikipédia)

A mágnescsíkot előszeretettel alkalmazzák bankkártyákon, beléptető kártyákon és személyazonosító kártyákon.

Smart-card/Smart-token

Ebbe a kategóriába sorolandó a korábban említett SIM kártya is, de ezenkívül alkalmazzák a közlekedésben, egészségügyben, szórakoztatóiparban és bankszektorban egyaránt. Leginkább elterjedt változata az ISO/IEC 7816-ban rögzített „Contact Card". Itt az olvasó egységbe helyezést követően az érintkező chipen keresztül valósul meg az adatátvitel. Lehetnek egyszerű memóriakártyák, melyek 1 Kbit–1 Mbit méretű EEPROM-mal (Electrically Eraseable Programmable Read-Only Memory) rendelkeznek. Használatukkor az alkalmazó rendszernek kell tudnia, hogy milyen kártyát fognak használni egy adott helyen, mivel ezek nem képesek önmaguk azonosítására.

17. ábra Smart card (forrás: SecurId)

A következő csoport a védett vagy szegmentált memóriájú kártyák. Esetükben beállítható, hogy képesek legyenek kontrollálni a kártyához való hozzáférést, akár írásról, akár olvasásról van szó, illetve feloszthatóa tárhelyük több logikai egységre, így érve el multi-funkcionalitást. Nehezebben másolhatóak, mint az egyszerű memóriakártyák. [8]

CPU/MPU multifunkciós kártyák

Ezek a típusú kártyák saját adatfeldolgozó egységgel rendelkeznek, amelynek feladata a memória allokációja egymástól független egységekbe rendezve a kártya különböző alkalmazásaihoz mérten. Ezeket a memóriaműveleteket a kártyán lévő COS[6] végzi, melyhez az eszközön 8 Kbit–1 Mbit közötti tárhely áll rendelkezésre. Emiatt a képesség miatt könnyen frissíthetőek rajta az információk a kártya cseréje nélkül. Sokféle típusú chip létezik, amelyek közt találunk Java Card VM egységekkel vagy PKI[7] integrált matematikai processzorral ellátott chipeket is.

[6] Card Operating System – a Smart Card operációs rendszere.

[7] Public Key Infrastructure – Nyilvános kulcsú infrastruktúra.

RFID

A legnagyobb különbség az előzőekhez képest, hogy az ilyen típusú kártyák érintkezésmentesek. Létezik belőlük csak olvasható, ezek tipikusan 125 MHz-en kommunikálnak, illetve kis memóriájúak. A ténylegesen írható, olvasható verzióra a 13,56 MHz a jellemző, melyet az ISO/IEC 14443 ír le. Megkülönböztetünk aktív, illetve passzív RFID-t, melyek a tápellátásukban térnek el egymástól. Aktív esetben valamilyen áramforrás szükséges a működéshez, ami több hátulütőt is jelent. Például nagyobb a méretük, és drágábbak. A passzív megoldásnál, csak az RFID taget építik be a kártyába, és csak az olvasóegység az, ami aktiválja a kommunikációt.

A technológia fő előnyei:

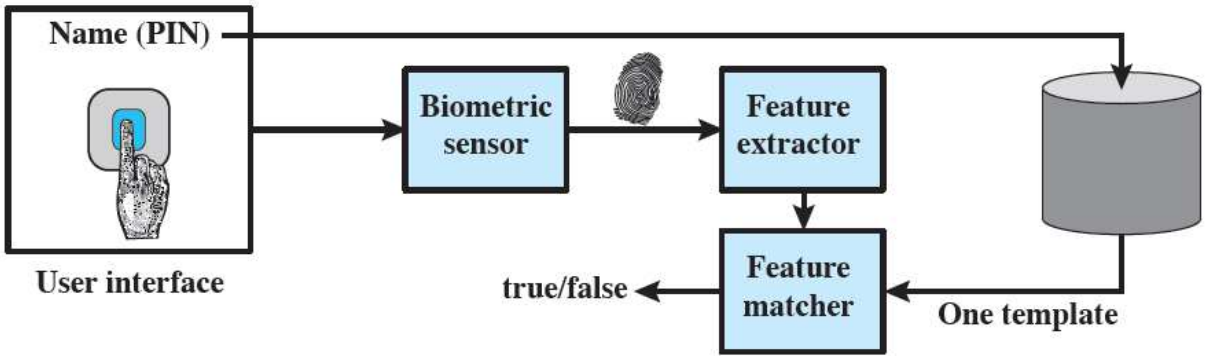
- érintkezésmentes,
- elegendő memóriakapacitás: 16–64 Kbyte,
- néhány milliszekundumos írási/olvasási sebesség.

Habár az RFID használatakor az olvasó berendezés és a leolvasandó tag között kicsi a távolság, ettől még potenciális veszélyt jelent a hallgatózás, hiszen még ha a leolvasandó tag hatósugarán kívül is esik a hallgatózó, attól még az olvasóból érkező jeleket észlelheti. Igaz, ez passzív esetben kevésbé jellemző, mint aktív RFID-nál. Elképzelhetőek még a hálózatokon ismert DoS támadások is, viszont a megvalósítás itt különböző. Alapvetően a kommunikációs csatorna zajosítását jelenti a rendszer hullámhosszán olyan mértékben, hogy meghiúsuljon a két eszköz közti adatcsere.

3.2.3. Biometria

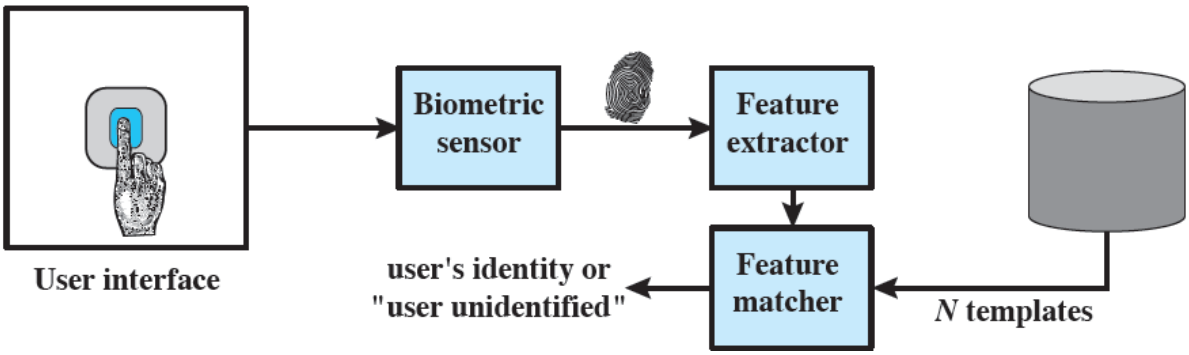
A hitelesítésnek ez a típusa az azonosítandó személy különféle biológiai jellemzőinek vizsgálatára épül, így értelemszerűen az eddig említett megoldásokhoz képest több pozitívumot és előnyt rejt magában, tekintve, hogy az efféle azonosító jegyeket sem elfelejteni, sem elhagyni nem lehetséges. Alapvetően minden biometriai rendszernél az azonosítási folyamatnak két lehetséges módja van.

Az egyik eset az ellenőrzés, melynek során az alany a meglévő azonosítójával együtt adja meg a szükséges mintázatot az ellenőrzéshez, ami lehet ujjlenyomat-olvasásból, retina, arc vagy egyéb jellemző olvasásából származó adathalmaz. Ebben az esetben csak az adott felhasználó korábbi mintázatai között keres egyezést a rendszer.



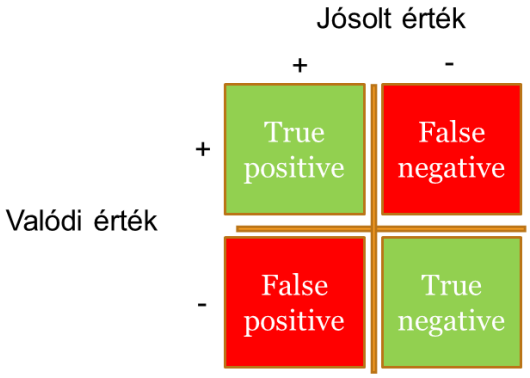
18. ábra Biometriai ellenőrzés (forrás: [4])

A másik lehetőség, amikor nem ad meg azonosítót az alany. Ilyenkor az összes előzetesen elmentett mintázat között kell keresni azt az egyet, amire megfelelően illeszthető a megadott minta, és ez alapján beazonosítani, hogy ki is akar belépni.



19. ábra Biometriai azonosítás (forrás: [4])

Ennek eldöntésekor természetesen adódhatnak hibák az azonosítási módszer pontosságából és az aktuális minta milyenségéből következően is. Döntési hibákat tekintve megkülönböztetünk hamis pozitívokat (FP), amikor egyezést mutat a módszer a nem megfelelő mintákra is, illetve hamis negatívokat (FN), amikor pedig az egyező minták illesztésekor sem talál egyezést az adott rendszer.



20. ábra Megbízhatóság

Ujjlenyomat

A biológiai jellemzők alapján való azonosítás legelterjedtebb módszere, hiszen a szükséges ujjlenyomat-olvasók napjainkban már igen olcsók, átlagosan 80–90 dollárba kerülnek.[8] Emiatt találkozhatunk vele mind egyszerű laptopokban, mind nagyvállalati környezetben. A másik fő ok, ami miatt széles körben használatos, hogy az ujjlenyomat ténylegesen különböző minden embernél, beleértve az ikreket is, tehát kézenfekvő megoldás a minták különbözősége miatt.

21. ábra Notebookba épített ujjlenyomat-olvasó (forrás: mobilport.hu)

A módszer lényege, hogy a bőrön lévő redőzetből kialakuló formákat, mintázatokat vizsgálja, melyeknél szükséges megkülönböztetni a redők tipikus építőelemeit: ívek, hurkok, elágazások, örvények. A legfontosabb összetevő viszont az ún. mag, ami köré épül a mintázat, és a delta pont, ami mutatóként szolgál a minták közti elágazásokhoz. Ezeket az elemeket felhasználva vizsgálja a redők hosszúságát, vagy egy egységnyi területre eső számukat. A feldolgozáshoz szükséges olvasókészülék a felület alatt található prizmából, fényforrásból és lencséből épül fel. Olvasáskor a fényforrás megvilágítja az leolvasandó ujj felületét, majd a lencse továbbítja a képet vagy CCD[9] vagy CMOS[10] képfeldolgozó szenzorhoz. [9]

A módszer hátulütője, hogy elég nagy számítási kapacitást igényel, tekintettel a rengeteg lehetséges mintára és az ezek közti keresésre. Ezenkívül előfordulhatnak olyan esetek, amikor az azonosítás nem megfelelő, pl.: rendszeres fizikai munkát végzők körében, ahol sérülések roncsolhatják az ujjak felszínét, így befolyásolva az azonosítás eredményességét.

Retina

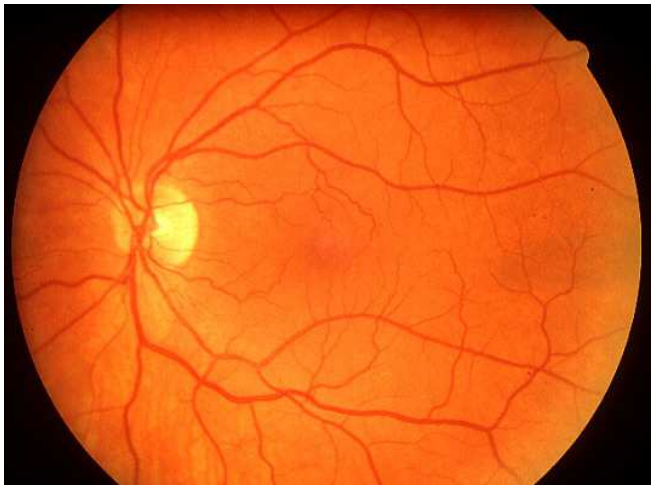
A retináról való képalkotás alapján azonosítani felhasználókat hasonlóan hatékony megoldás, mint az ujjlenyomatok vizsgálata. A folyamat lényege, hogy az emberi szem

[8] http://www.cio.com/article/2441829/servers/biometric-technologies--fingerprint-recognition--retina-scans-and-facial-recognition.html

[9] Charge Coupled Device.

[10] Complementary Metal Oxid Semiconductor.

hátulsó részén elhelyezkedő retinát és az érhártyát tapogatják le egy alacsony intenzitású fényforrás segítségével, melynek sugarait az erek elnyelik, így létrehoznak egy világos és sötét részekből álló képet. A sötét részek mutatják az ereket, melyek helyzetét alapul véve jön létre az azonosításra használható mintázat.



22. ábra Retina (forrás: Wikipédia)

Ez a minta is teljesen eltérő emberenként. A képalkotáshoz szükséges fény, mire a retinához ér, áthalad a pupillán. A pupilla meghatározza a bejutó fény mennyiségét, ezért ez is növeli a képek egyediségét. A módszer szinte soha nem szolgáltat hamis eredményeket, és befolyásolni vagy megmásítani sem lehet az azonosító jegyeket, hiszen azok az emberek szemében vannak, ahol nem érheti a közeget semmilyen környezeti hatás közvetlenül. Másfelől viszont, akinél valamilyen szembetegség alakul ki (pl.: szürke hályog), csökkenhet a pontossága.

Az ujjlenyomat-olvasáshoz képest kevésbé elterjedt és kedvelt módszer, mert a felhasználók jellemzően fenntartásokkal kezelik, hogy valamilyen eszközbe kell belenézniük és félnek az egészségügyi kockázatoktól. További hátrány, hogy a módszer igen költséges.

3.2.4. Képesség

A dinamikus, vagy képességen alapuló biometria, csakúgy, mint a korábban említett statikus megoldások, nagyfokú egyediséget biztosítanak az emberek azonosítása szempontjából, viszont működési elveik alapjaiban térnek el egymástól. Ebben az esetben nem az emberek fizikai vagy biológiai jellemzőit veszik alapul, hanem olyasféle kihívás elé állítják őket az azonosításukhoz, amelyhez egyes képességeiket kell felhasználniuk, mint például a kézírásuk, hangjuk, rajztudásuk vagy gépelésük ritmusa. Mindegyik esetben egy kép, hang vagy egyéb feldolgozó eljárást követően nyerik ki az adathalmazból a felismeréshez használható mintát.

Kézírás és rajz

Mivel minden embernek teljesen más az írásképe, egy-egy szó leírása vagy egy alakzat lerajzolása is alkalmas a megkülönböztetésükre. Egyre szélesebb körben elterjedt módszer. A bankszektorban például már régóta bevett gyakorlat az aláírás ellenőrzése, de

napjainkban egyre több alkalmazás készül okostelefonra vagy táblagépre, mely a kézírásból betűket, de akár matematikai összefüggéseket is képes felismerni.

A módszer folyamatában megkülönböztetünk offline és online felismerést. Offline esetben tipikusan egy korábbról meglévő dokumentum vagy annak egy részlete az, amit vizsgálni kell. Online esetben pedig valós időben egy érintésérzékeny felületen egy tollal (stylus) kell bevinnie a felhasználónak a kívánt írási mintát vagy rajzot. Ahhoz azonban, hogy a betűket vagy alakzatokat fel lehessen ismerni, szükséges meghatározni, hogy milyen paramétereket kell vizsgálni. Online esetben például a toll nyomásának értéke, az érintőképernyőn és a fölötte töltött idők, másodpercenkénti karakterszám stb. A felismeréshez vezető folyamat alapvetően három nagy részből áll: az írt vagy rajzolt minta előfeldolgozása, a meghatározott tulajdonságok kinyerése, majd végül annak meghatározása, hogy ahhoz az emberhez tartozik-e a minta, mint vártuk. Az ilyen mintázatok felismerésére javarészt neurális hálózatokat alkalmaznak, melyek bizonyos fokú tanítás után, képesek osztályozni és felismerni különböző mintákat. Ahhoz, hogy az ilyen típusú hálók használhatóak legyenek az azonosítási folyamatban, a különböző betűket először szét kell választani egymástól, majd a betűk képeit a felbontásuk csökkentése után mátrixos alakban reprezentálva már felhasználhatóak bemenetként. Az ilyen mátrixok úgy alakulnak ki, hogy nullának veszik a fehéren maradt területeket, és egyesnek azt, ahova írt a felhasználó. Az ilyen számértékek már betáplálhatóak a hálók bemeneteire. A neurális hálók működéséről még olvashatunk a behatolásjelző fejezetnél.

Hangfelismerés

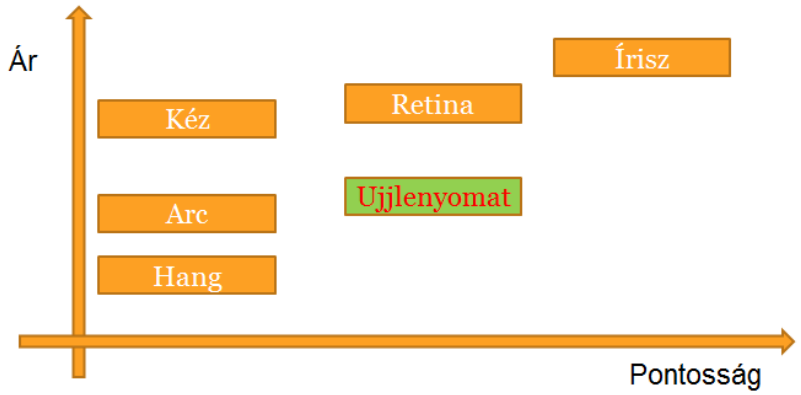
Bár nem tartozik a legmegbízhatóbb módszerek közé, a felhasználók körében mégis elfogadottnak mondható, hiszen egyszerűen csak beszélniük kell az authentikációhoz, és egészségügyi szempontból nem kell tartani semmitől, mint például a retinaolvasó esetében. A hang azonosításra való felhasználásánál a beszédhangnak olyan egyedi jellemzőit figyeli az adott rendszer, mint a hangmagasság, hangszín, hangerő. Ezeknek a minőségét befolyásolja minden egyes szerv, ami a beszéd kialakításában játszik szerepet, tehát szinte lehetetlen reprodukálni, hiszen minden ember máshogy beszél, másképp fejlődnek az artikulációhoz szükséges izmok és különböznek a hangképző szervek is.

Alapvetően ennek is kétféle megvalósítása létezik, de mindkét esetben a beszédhang spektruma figyelendő. Az egyik, amikor a hangot kell azonosítani, hogy egyezik-e a korábbi rögzített hanggal. Ilyenkor minden alanyhoz több rögzített hangkép tartozik, melyekkel a rendszer összeveti az aktuális hangot. A másik lehetőség, amikor a beszélő személyét kell felismerni a hang alapján. Ez a beszédhang tulajdonságainak kinyerésével zajlik, melyekből különböző matematikai eljárásokkal kapják meg az azonosításra szolgáló értékeket. Ez a fajta eljárás sokkal érzéketlenebb magas háttérzajú közegekben.

3.2.5. Összehasonlítás

A különböző felhasználó-azonosítási módszerek közül egyértelműen a tudásalapú azonosítási rendszerek a legelterjedtebbek, azonban a biometriai rendszerek is egyre nagyobb népszerűségnek örvendenek. A biometriai megoldások között azonban nagy

szórást tapasztalunk mind költség, mind pontosság szempontjából. Ez látható az alábbi ábrán.



23. ábra Biometriai felhasználó-azonosítási módszerek összehasonlítása

3.3. Authorizáció

Definíció szerint az authorizáció olyan biztonsági mechanizmusok gyűjteménye, mely meghatározza, hogy a felhasználók mit tehetnek a rendszerben, azaz milyen erőforrásokhoz férhetnek hozzá és milyen műveleteket hajthatnak végre.

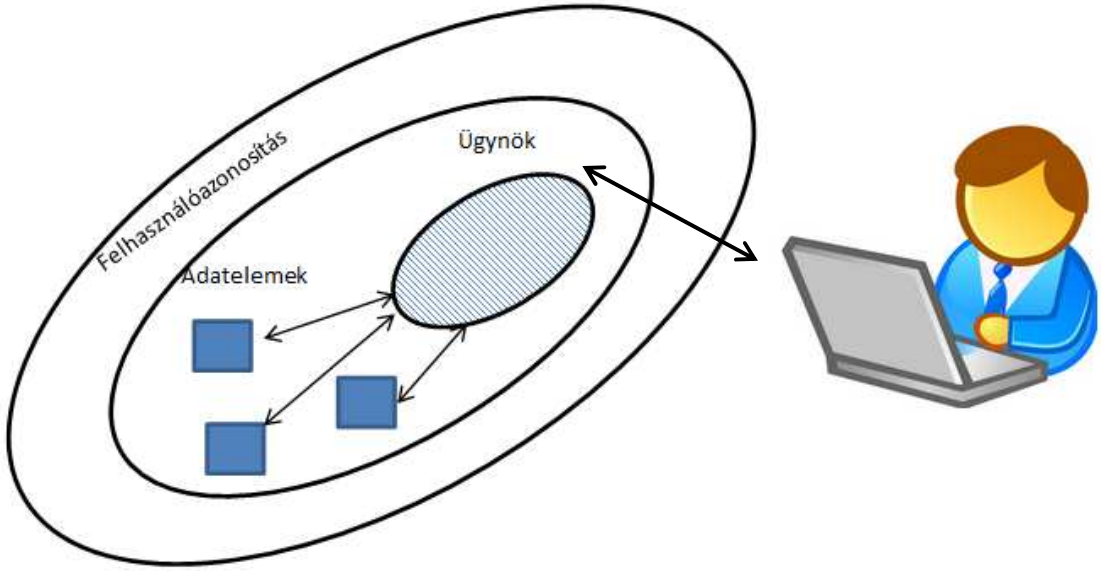
Azt követően, hogy egy klienst sikerült azonosítani, még mindig sok lehetséges kérdés marad fenn. Tisztázni kell, hogy az adott, és már bizonyított identitással mire jogosult a rendszerben, milyen erőforrásokhoz, hálózatokhoz férhet hozzá. Vagy ha internet-előfizetők szemszögéből nézzük: milyen minőségű szolgáltatásra jogosultak. [10]

3.3.6. RBAC

RBAC (Role Based Access Control). A jogosultságok kezelése szerepkörök alapján egy adott szervezeten belül. Például bármely iskolai nyilvántartó vagy tanulmányi rendszerben más és más jogokkal kell rendelkeznie egy tanárnak és egy diáknak, hogy mindenki csak a saját szerepéhez mérten láthasson vagy tehessen különböző dolgokat. Ebben az esetben a jogosultságok kezelését központilag végzi egy arra kinevezett személy. A szervezeti felépítésből adódóan ő lehet akár egy rendszergazda vagy akár egy csoportvezető.

3.3.7. DAC

DAC (Discretionary Access Control). Az előzőhöz képest ebben az esetben nincs központi jogosultságkezelés. Ez is a legnagyobb hátránya ennek a módozatnak. Mindent az adat tulajdonosa szab meg, beleértve, hogy egyes felhasználók mit tehetnek az információkkal. Ezen felül a tulajdonosok átruházhatják a tulajdonjogukat, így a bizalmasság kialakítása nem garantálható.



24. ábra Általános DAC modell

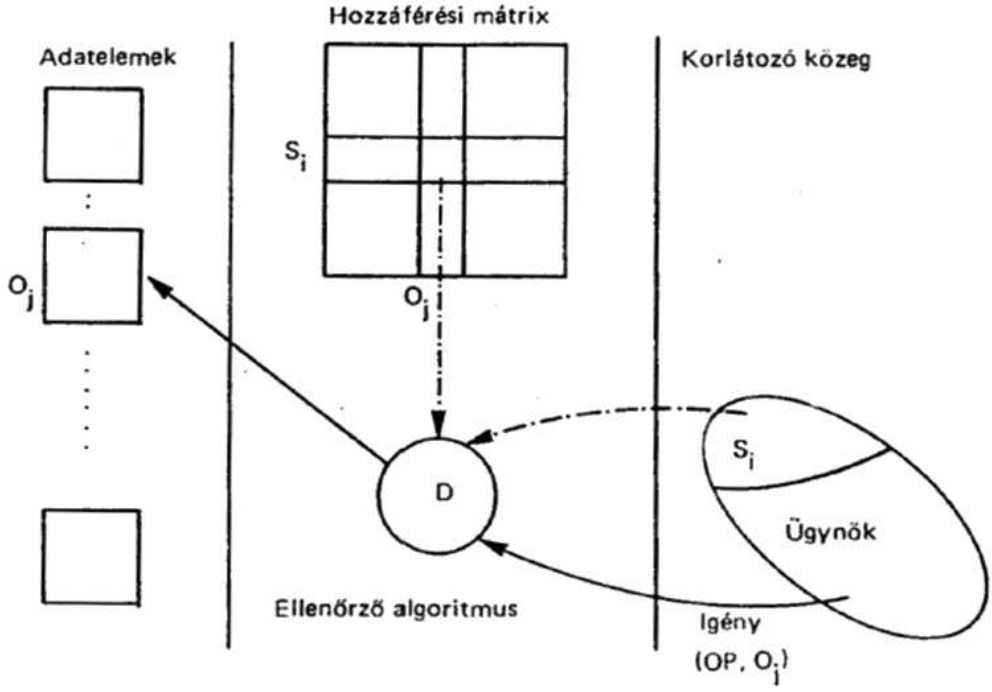
A DAC modell szerint az informatikai rendszerben az adatok úgynevezett adatelemek formájában vannak jelen. Az adatelemek hozzáférése csak ügynökök (agent) segítségével történhet meg. A felhasználó a felhasználó-azonosítási procedúra után válik jogosulttá belépni a rendszerbe, és ezt követően rendelhető hozzá az ügynök.

Az ügynökök egy korlátozó közegben, többnyire az operációs rendszerben dolgoznak. Az adatelemeket az ügynökök csak egy ellenőrző algoritmuson keresztül érhetik el.

Az adatelemek méretének meghatározásakor ellentétes követelményeket kell kielégítenünk. Az árnyalt védelmi stratégia finom szemcsézettségű adatbázist kívánna meg, ami kis adatelemméretet jelentene. Ebben az esetben viszont az adatelemek nyilvántartása nem lenne költséghatékony. Így kompromisszumot kell kötnünk.

Ügynökök[5]

Az algoritmikus védelmi rendszerben a külső felhasználó igényeit az általa működésbe állított folyamatok közvetítik, amelyeket a védelemhez való kapcsolatukban ügynököknek nevezünk. Egyetlen felhasználó feladatainak végrehajtásához számos ügynököt vehet igénybe, esetenként még párhuzamosan tevékenykedőket is. Alapvető szerepet játszik azonban az az ügynök, amelyet egy felhasználóhoz a rendszerbe lépésekor rendel a felhasználó-azonosítás. Ennek az ügynöknek elsődleges feladata a képviselt felhasználó igényeinek közvetítése a rendszerhez. Mivel ez az ügynök a felhasználót mindenben helyettesíti, a védelmi rendszer további tárgyalása során csak ügynököket veszünk tekintetbe.



25. ábra Lampson modell (forrás: [5])

Az ügynökök a számítógépes rendszer „teremtményei”, tevékenységük ezért szigorú korlátok közé szorítható. Minden adatelem-hozzáférési igényüket a védelmi rendszeren keresztül kell benyújtaniuk, és a döntéseit velük szemben a rendszer kötelező erővel érvényesíti. Az ügynökök a védelmi rendszert nem kerülhetik meg (még akkor sem, ha összejátszanak). Egy ügynök tevékenységének korlátait egyrészt annak a felhasználónak a lehetőségei határozzák meg, akinek a megbízásából tevékenykedik, másrészt maga a tevékenység, amelynek elvégzésére az ügynöknek „képesítése" van. Ez utóbbit az ügynökprogram része testesíti meg. A védelmi rendszer szempontjából az ügynököt tehát ez a két információ határozza meg:

- a megbízó személye és
- a végrehajtott program funkciója.

Az ügynök lehetőségeit ezekből származtathatjuk.

Ellenőrző algoritmus

Minden, a rendszer védelmét potenciálisan érintő művelet csak az ellenőrző algoritmusnak a közbeiktatásával fejtheti ki hatását. A célba vett adatelemet és a hozzáférési igényt (a kezdeményezett művelet megjelölésével együtt) az ügynök nyújtja be.

Az ügynökazonosítót viszont a végrehajtó környezet szolgáltatja. Ezeknek a jellemzőknek az ismeretében az ellenőrző algoritmus döntését a hozzáférést szabályozó adatbázis alapján hozza, és a végrehajtást szabályozó közeggel tudatja is. Ez utóbbi kötelező érvényre emeli azáltal, hogy a döntésnek megfelelően engedélyezi vagy megakadályozza a

kezdeményezett művelet végrehajtását. Magától értetődőnek tűnhet, de érdemes külön is hangsúlyozni: az ellenőrző algoritmus csak döntéseihez használhatja fel a hozzáférést szabályozó adatbázist, de nem módosíthatja, és tartalmát a külvilággal nem ismertetheti meg.

Hozzáférési adatbázis

A hozzáférési adatok struktúrájából úgy tűnik, hogy a rendszer védelmét meghatározó adatokat mátrix formájában, az ún. hozzáférési mátrixban célszerű nyilvántartani. A hozzáférési mátrixban minden egyes felhasználóhoz (subject) egy sor, minden egyes adatelemhez (object) egy oszlop tartozik, az A(ij) elemben pedig az S(i) ügynöknek az O(j) adatelemmel kapcsolatos hozzáférési jogai (Aij) találhatók meg.

		OBJEKTUMOK						
		Alanyok			Fájlok		Eszközök	
		S_1	S_2	S_3	F_1	F_2	D_1	D_2
ALANYOK	S_1		block wakeup		Read write		seek	
	S_2			Stop		update		Seek
	S_3				Delete	execute		

26. ábra Hozzáférési mátrix (forrás:[25])

A hozzáférési jogokat az esetek nagy részében azoknak a műveleteknek a felsorolásával adjuk meg, amelyeket az ügynök a szóban forgó adatelemre alkalmazhat. Egy adatokat és programokat tároló adatbázisban például adatelemeket olvashatunk (r), módosíthatunk (w) és törölhetünk (d), programok végrehajtását kezdeményezhetjük (x), és új adatelemeket hozhatunk létre (a). Az egyes hozzáférési jogokat így egyszerűen az *r w d x a* műveletek valamilyen kombinációja határozza meg. Amelyik ezek közül a hozzáférési jogban nem szerepel, annak végrehajtására az ügynök nem jogosult. A mátrix egy-egy sorát hozzáférési tartománynak nevezzük, mert a sorhoz tartozó ügynök által elérhető adatelemeket (engedélyezett felhasználási módjukkal együtt) tartalmazza. A mátrix mindegyik oszlopa egy-egy hozzáférési lista, amelyen az adatelemhez hozzáférő ügynökök (hozzáférési jogaikkal együtt) vannak felsorolva.

A hozzáférési mátrix az első pillanatban megtévesztően egyszerűnek tűnik. Egy működő rendszerben azonban dinamikus adatstruktúra. Amikor ügynököt vagy adatelemet hozunk létre vagy meglévőket törlünk, a hozzáférési mátrix a formájában is változik. Szövevényes problémahalmazra bukkanunk továbbá, ha a hozzáférési mátrix tartalmát vizsgáljuk.

3.3.8. MAC

MAC (Mandatory Access Control). Ennek a típusnak a lényege, hogy minden információhoz egy mutatót rendelnek, ami arra utal, hogy mennyire érzékeny az adat. Ez a mutató határozza meg a biztonsági szintet, amit az egyes felhasználók profiljaihoz is hozzáfűznek.

Egy felhasználó a saját biztonsági szintjénél alacsonyabb szintű információkhoz is hozzáférhet. Új adatot írni viszont a sajátjánál magasabb szintre is lehetséges.

3.4. Audit

Egy informatikai rendszert akkor tekinthetünk sikeresnek, ha működése hatékony és minden körülmények között biztosított. Az, hogy valami hatékonyan működik, csak úgy derülhet ki, ha vizsgálatokat végzünk a rendszeren és a vizsgálatok eredményeiből levonjuk a tapasztalatokat. Kicsi, jól átlátható rendszerek esetekben nincs szükség bonyolult vizsgálatokra, ezeket a vizsgálatokat nem is szokás auditnak nevezni. [6] A témáról részletesebben a [26] irodalomban lehet olvasni.

Ha rendszerünk nagyobb, akkor működésének átvilágítása már annyira komplex feladat, hogy vizsgálatához audit rendszerre van szükség. Az auditálás többek között a rendszer hatékonyságát, rutinszerű működését, irányítását és nem utolsósorban a biztonságát felügyeli.

Ha egy szervezet életében felmerül az audit rendszer szükségességének kérdése, akkor az általában az alábbi négy fogalom valamelyikének kapcsán történik:

- rendelkezésre állás (avaliability)
- hatékonyság (efficiency)
- biztonság (security)
- integritás (integrity).

Az audit rendszerek elsődleges feladata, hogy meghatározzák és minimalizálják az informatikai rendszer rendelkezésre állásának, hatékonyságának, biztonságának és integritásának kockázatait. Vannak olyan auditok, amelyek a fentiek közül csak az egyikre fókuszálnak, de vannak olyanok is, amelyek akár mindegyikre. [14]

Az, hogy egy szervezetnek milyen típusú auditra van szüksége, csak hosszas vizsgálat után dönthető el. Ehhez a típusok bemutatásával nyújt segítséget a következő fejezet.

3.4.9. Audittípusok

Az előző fejezetben megállapítottuk, hogy az auditálást a rendszer különféle jellemzőinek szem előtt tartásával hozzák létre. Nyilvánvaló tehát az, hogy a különböző jellemzők biztosítására különböző típusú auditokat használunk. Az auditok csoportosításának egy lehetséges módja az irodalom által alkalmazott egyik szemléletmód, amely az alábbi csoportokat emeli ki:

- rendszer- és alkalmazásaudit: azt vizsgálja, hogy a rendszer megfelelően működik e, a bemenetek és a kimenetek megfelelnek-e a követelményekben leírtaknak;
- információs folyamatok auditja: azt vizsgálja, hogy a folyamat ütemezése és készültségi foka megfelel-e az elvártaknak;

- rendszerfejlesztés-audit: azt vizsgálja, hogy a fejlesztés alatt álló rendszer megfelel-e a nemzetközileg elfogadott szabványoknak;
- biztonsági audit: ellenőrzi, hogy az adatokkal kapcsolatos tevékenységek és e tevékenységben részt vevő folyamatok, eszközök megfelelnek-e a különböző biztonsági előírásoknak. Az előírások lehetnek szervezeten belüli belső szabályok, de lehetnek külső szabályok is (törvények, szabványok stb.).

Egy másik megközelítés szerint auditálással bármit lehet ellenőrizni, ezek az úgynevezett ellenőrzési auditok. Ilyenek lehetnek például az „alkalmazás-ellenőrzési audit", „rendszer-ellenőrzési audit" vagy a „biztonság-ellenőrzési audit".

Az ellenőrzési auditok három alapvető csoportba sorolhatók:

- preventív (megelőző) ellenőrzés: előre lefektetett szabályokat érvényesít, ilyen szabályokat gyakran szabványok vagy törvények formájában is előírnak;
- detektív (nyomozó) ellenőrzés: az ilyen típusú auditok monitorozzák a rendszer működését, de korlátozásokat nem alkalmazhatnak;
- reaktív (reagáló) ellenőrzés: a detektív ellenőrzéssel szorosan együttműködik, munkája során figyelmeztetéseket, riasztásokat eredményez bizonyos események bekövetkezésekor.

Természetesen itt is a konkrét rendszertől függ, hogy melyik módszert alkalmazzuk, de általánosságban elmondható, hogy e három típus egyidejű alkalmazása, és körültekintően megtervezett együttműködése révén érhetjük el a legjobb eredményeket.

3.4.10. Audit és biztonság

A defense-in-depth stratégia rétegeinek megvalósításával a védendő rendszer megközelítése lényegesen nehezebb feladatot jelent a támadóknak. Minden réteg készít bizonyos szintű naplókat a bekövetkezett eseményekről, sőt jobb esetben még riasztás is történik. Ez azonban nem garancia arra, hogy egy nagy informatikai rendszerben a biztonsági figyelmeztetések eljutnak az illetékes személyekhez.

A másik szembetűnő hiányosság, hogy a külsőbb biztonsági rétegek védelmezik ugyan a belsőbb rétegeket, de azt egymástól függetlenül teszik.

Ezen problémák eltörpülnek amellett, hogy a fenti stratégia csak a kívülről jövő fenyegetések ellen hivatott védelmet nyújtani, a belső támadások ellen semmilyen módon nem véd. Egy informatikai rendszer jogosult felhasználója belső támadást hajthat végre a rendszer ellen, akár úgy, hogy veszélyezteti a rendszer működését, de akár úgy is, hogy bizalmas adatokat lop el. Az ilyen típusú támadások ellen a fenti módszerek nem védenek, ezért egy olyan módszert kell alkalmazni, ami kiküszöböli a védelmi rendszer hiányosságait. E célnak tökéletesen megfelel az auditálás.

Az alábbi ábrán a biztonsági piramis látható, amely vertikálisan mutatja a biztonsági rétegek egymásra épülését.

27. ábra Biztonsági piramis (forrás: audits.uillinois.edu)

A piramis csúcsán az audit helyezkedik el, amely az alatta lévő rétegek által biztosított információk (naplók, riasztások stb.) összegyűjtésével és feldolgozásával szavatolja, hogy a rendszer biztonságáról egységes képet alkothassunk. Ezzel megvalósítja a rétegek együttműködését és megkönnyíti a biztonság menedzselését.

Másrészt az audit segítségével lehetőségünk van a belső felhasználók tevékenységeinek figyelésére, amivel visszakereshetővé, jelezhetővé és nem utolsósorban megelőzhetővé válnak a belső támadások.

Az auditálás lehetőségei és felhasználási formái nagyon szerteágazóak, de mindig szorosan kapcsolódnak a biztonság kérdésköréhez. Kijelenthető, hogy audit nélkül nem létezhet valódi biztonság.

3.4.11. Audit alapelemek

Az informatikai rendszerek üzemeltetésében már évtizedekkel ezelőtt felismerték az auditálás jelentőségét, ezért már régen is auditáltak. Természetesen ezek a rendszerek egészen más lehetőségek mellett látták el feladatukat és önmagukban állva a mai értelemben nem is nevezzük őket audit rendszereknek. Lefektették azonban a modern auditálás alapjait, ezért megismerésük elengedhetetlen.

Egyszerű naplózás (Logging)

Az informatikai rendszerek kezdetleges auditálási lehetősége az egyszerű naplózás volt, amelynek során különböző – a szoftver kódjában rögzített – tevékenységek esetén rögzítették az adatokat. Itt a felhasználónak nem volt lehetősége a naplózandó tevékenységek sorát bővíteni vagy módosítani. Fontos tulajdonsága és természetesen hátránya a módszernek, hogy csak passzív rögzítés történik, az információ belekerül egy állományba vagy egy adatbázis-táblába, de semmilyen aktív művelet nem történik. Nincs lehetőség figyelmeztetések vagy riasztások elküldésére, és megakadályozni sem lehet a nemkívánatos tevékenységet.

A legtöbb szoftver, köztük az adatbázis-kezelők is rendelkeztek ilyen szolgáltatással. Amellett, hogy kezdetleges technikáról van szó, minimális biztonságot mégis nyújtott a

módszer, mert hiba esetén utólag meg lehetett vizsgálni a naplók tartalmát (feltéve persze, ha valaki ki nem törölte addig).

Alkalmazásvezérelt naplózás (Application Controlled Logging)

Ez a naplózás lényegében azt jelenti, hogy az adatbázissal kapcsolatban álló alkalmazás kódjában valósul meg az auditálás.

Tipikus megoldás, hogy az adatbázisban létrehoznak egy audit táblát, amelynek feltöltéséért az alkalmazás felel. A naplózandó események sorának csak a programozó, illetve az alkalmazás tervezőjének fantáziája szabhat határt. Amikor egy „figyelt" esemény bekövetkezik, akkor a végrehajtással párhuzamosan megtörténik az esemény rögzítése az adatbázisba. Egyszerű példaként, egy felhasználó bejelentkezési folyamatában az alkalmazás kéréssel fordul az adatbázishoz, amelynek célja a felhasználó hitelesítése (például felhasználónevének és jelszavának ellenőrzése). Ekkor az audit táblában rögzítik a felhasználó bejelentkezési kísérletét. A hitelesítés sikerességétől függően egy újabb bejegyzés készül az audit táblába, ami alapján később megállapítható, hogy sikeresen belépett-e avagy sem.

Audit
audit_azonosíto
felhasznalo_azonosito
IP_cim
hoszt_cim
idobelyeg
tevekenyseg

28. ábra Egy tipikus audit tábla

Sajnos az auditálás ebben az esetben is kimerül a tények puszta rögzítésében, figyelmeztetések küldésére csak az alkalmazásból van lehetőség. Bár ez a módszer már sokkal rugalmasabban képes naplózni, használata esetén alapvető hiányosságokkal kell számolnunk. A leginkább szembetűnő probléma, hogy csak azokat a tevékenységeket naplózzák, amelyeket az alkalmazásból indítottak. Az adatbázishoz történő alkalmazáson kívüli hozzáférések, a rendszergazdai tevékenységek és egyéb fontos momentumok auditálására nincs lehetőség. Ezenkívül alapvető szemléletbeli különbség, hogy ez a módszer adatbázis-tranzakciók helyett alkalmazás-tranzakciókon keresztül valósítja meg az auditálást.

Naplóelemzés (Log Mining)

A naplóelemzés során rendelkeznünk kell olyan naplóállományokkal, amelyek például a fenti módszerek valamelyikével állíthatók elő. A naplók nem feltétlenül származnak csak egy adatbázisból, elképzelhető, hogy egy informatikai rendszer több komponensének naplóit vizsgáljuk egyszerre (például operációs rendszerek, tűzfalak, vírusirtók naplói).

A naplók rengeteg adatot tartalmaznak, amelyek az ember számára átláthatatlanok, ezért az érdemi összefüggések felismerése is nehézkes. Vannak azonban olyan módszerek, amelyek nagy mennyiségű adatok vizsgálatával adnak érdekes következtetéseket. Érdekesnek tekinthetők azok az összefüggések, amelyek az elemző számára korábban nem ismert, nem triviális és vélhetően hasznos információt nyújtanak. Az ilyen módszereket adatbányászati módszereknek nevezzük.

Naplóelemzés alatt tehát a különböző forrásból származó naplók adatbányászati elemzését értjük.

Az elemzés során olyan események keresése a cél, amelyek befolyásolják a rendszer biztonságos működését. A vizsgálat középpontjában állnak:

- a szokatlan események;
- a megszokott események elmaradása;
- a normális események láncolata.

Az adatbányászat eszközei között különféle módszerek találhatók, közülük a legfontosabbak:

- *Filterezés*

  Annak vizsgálata, hogy egy egyszerű vagy akár egy összetett kifejezés megtalálható a naplóban. Segítségével reguláris kifejezések is kereshetők.

- *Asszociációs szabályok*

  Az asszociációs szabályok egy bizonyos következményt kötnek össze feltételek egy sorával. Segítségükkel azt lehet behatárolni, hogy több feltétel teljesülése esetén milyen egyéb feltétel teljesülése várható. A módszer iskolapéldája a sör és a virsli kapcsolatának felismerése. Ha egy bevásárlásnál valaki virslit vásárol, akkor nagy valószínűséggel sört is fog vásárolni, ugyanis e két terméket előszeretettel fogyasztják együtt. Hasonló jellegű összefüggéseket a naplókból is ki lehet következtetni, és fel lehet használni. Az asszociációs szabályalgoritmusok automatikusan végzik az ilyen szabályok felderítését.

- *Osztályozás*

  Az osztályozás az adatbányászat egyik leggyakrabban alkalmazott eszköze. Osztályozásnak nevezzük azt a folyamatot, amely során egy adathalmaz elemeit ismert tulajdonságai alapján osztályozzuk előre meghatározott osztályokba. Biztonság szempontjából ezt úgy lehet hasznosítani, hogy meghatározzuk azokat a tulajdonságokat és feltételeket, amelyek befolyásolják, hogy egy esemény kockázatot jelent-e. Egy algoritmus (például logisztikus regresszió) pedig megállapítja, hogy a vizsgált tulajdonságok aktuális értékei alapján az esemény milyen osztályba tartozik, például kockázatos avagy sem. Ezt a módszert credit scoring néven előszeretettel használják a hitelintézetek is ügyfeleik minősítésére (az ügyfeleket jellemzőik

vizsgálatával osztályokba sorolják, ami alapján hiteligénylésüket elfogadják vagy elutasítják).

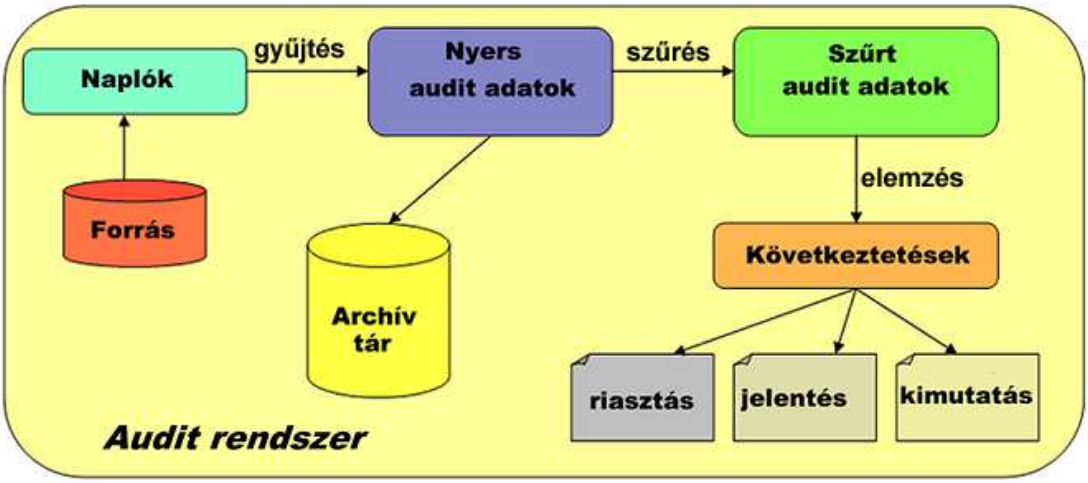
- *Neurális hálózatok*

  A mesterséges neurális hálózatok a mesterséges intelligencia világában született eszközök, amelyek tanulást követően automatikus következtetésre képesek. A tanulási folyamat alatt a háló egy algoritmus (például back propagation algoritmus) segítségével „megtanulja”, hogy bizonyos bemenetek esetén milyen kimenetet kell produkálnia, majd a tanulás után képes lesz különféle bemenetek esetén következtetni a kimenetre. Esetünkben képes lehet arra, hogy az audit információk egy részét bemeneti adatként felhasználva következtetéseket vonjon le a rendszer biztonsági állapotáról. A neurális hálók hatékonyságának kulcsa a megfelelő tanításuk, amely sok esetben idő- és erőforrás-igényes feladat.

3.4.12. Komplex audit rendszerek

Az auditálás eddig bemutatott elemei önmagukban nem, vagy csak korlátozott mértékben képesek megvalósítani egy informatikai rendszer auditálását. Láthattuk, hogy a naplók önmagukban még nem jelentenek biztonságot és a naplóelemző módszereknek szükségük van a forrásadatokra, amelyekből algoritmusaik segítségével következtetéseket vonhatnak le. Korántsem olyan triviális, hogy az elemző algoritmusok honnan és hogyan szerzik be az adatokat, és miként alakítják a nyers adatokat feldolgozásra alkalmas adatokká. A megoldás kulcsa a fenti audit alapelemek összekötésében rejlik, azaz szükség van egy naplózó funkcióra, ami előállítja az audit információkat. Szükség van egy alkalmazásra, amely elvégzi az audit információk összegyűjtését, és feldolgozásra alkalmas állapotban az elemző algoritmusok rendelkezésére bocsátja. Az algoritmusok által előállított következtetéseket végül egy prezentációért felelős funkció fogja az auditorok[11] számára elérhetővé tenni a legkülönfélébb formátumokban. Ezeken kívül egyéb biztonsági ellenőrzési feladatok is integrálhatók a funkciók közé. Az ilyen rendszereket komplex audit rendszereknek nevezzük. A következő fejezetekben az adatbázis audit rendszereinek példáján keresztül mutatjuk be az auditot.

[11] Auditornak nevezzük az auditálásért felelős személyeket, akik legtöbbször nem azonosak a rendszer adminisztrátoraival.



29. ábra Komplex audit rendszer

A komplex értelemben vett audit rendszerek nemcsak az auditadatokra koncentrálnak, hanem az információt létrehozó informatikai rendszerre is. Például egy adatbázis audit rendszer fő feladata az adatbázisban tárolt adatokkal kapcsolatos tevékenységek auditálása (hozzáférések, módosítások stb.). Egy komolyabb adatbázis audit rendszer ezenkívül az adatbázis szervert, a futtató operációs rendszert, tehát az auditadatok teljes környezetét is auditálja. Ez esetben az audit kiterjed a rendszer ismert gyengeségeinek ellenőrzésére, és az adatbázis konfigurációjának vizsgálatára is. Lényegében egy állapotfelmérés történik. Például ellenőrzik, hogy az adatbázis-kezelő rendszerhez telepítve vannak-e a legújabb javítások, de akár az is ellenőrizhető, hogy az adatbázis alapéltermezett adminisztrátori jelszavát megváltoztatták-e. Természetesen ezen vizsgálatok eredményeiről is értesítést kapnak a megfelelő személyek, annak érdekében, hogy a szükséges módosításokat elvégezzék.

3.4.13. Audit rendszerek tervezési kérdései

A korábbi fejezetekben áttekintettük az auditálás szükségességét meghatározó tényezőket, valamint azt is láthattuk, hogy milyen alapelemek állnak rendelkezésre az implementáláshoz. Most azt tekintsük át, hogy egy audit rendszer elkészítésekor milyen szabályokat kell betartani annak érdekében, hogy hatékony, jól működő rendszert kapjunk.

Célhoz kötöttség

Az auditálást mindig valamilyen céllal végezzük. Alapvető szabály, hogy audit nélkül nem létezik valódi biztonság, tehát ebben az esetben az auditálás célja a biztonság megteremtésének támogatása. Nem szabad azonban mindent feláldozni az audit érdekében, mert akkor könnyen előfordulhat, hogy az auditálás nem eszköz, hanem cél lesz. Lényeges, hogy csak a rendszer szempontjából valóban fontos eseményeket auditáljuk, azokat, amelyek a cél eléréséhez szükségesek. Ennek két szempontból is jelentősége van:

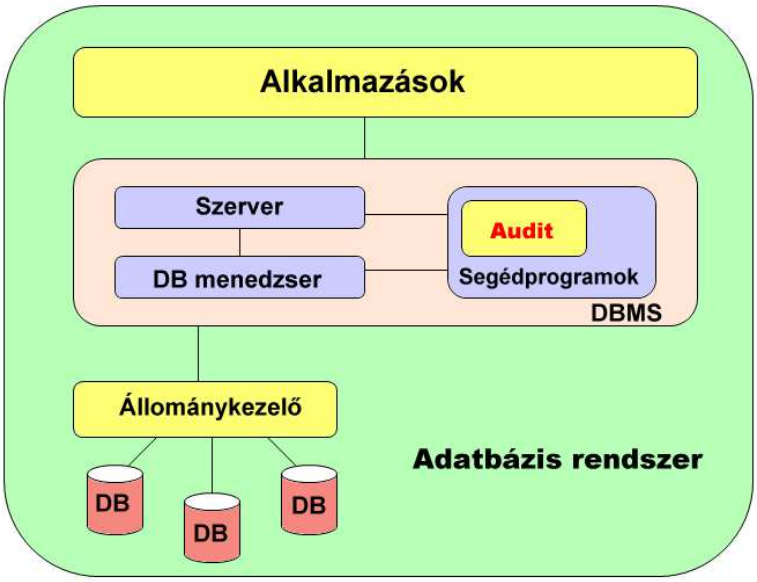
- Az egyik a törvényi szabályozás oldala, ami egyrészről nagyon szigorúan előírja, hogy egy informatikai rendszer működésével kapcsolatban mely területeken kell kötelezően ellenőrzéseket végezni, másrészről arról is rendelkezik, hogy audit címszó alatt nem szabad felesleges adatgyűjtéseket végezni. A törvényi szabályozások szinte mindegyikének alapelve a célhoz kötöttség, ami azt jelenti, hogy adatokat gyűjteni, továbbítani, elemezni csak az előre deklarált cél megvalósulásához szükséges mértékben szabad.
- A másik szempont a teljesítmény**Hiba! A könyvjelző nem létezik.** kérdése. Egy audit rendszer a működése során erőforrásokat használ. Processzoridőre van szüksége az adatok gyűjtéséhez, sávszélesség-igénye van az adatok hálózaton keresztül történő továbbításának, és tárterületre van szükség az adatok tárolásához. Arról nem is beszélve, hogy nagy mennyiségű auditinformáció feldolgozása komplexebb, időigényesebb feladat elé állítja az elemző algoritmusokat is. Az erőforrások (számítási kapacitás, sávszélesség, tároló kapacitás, idő) korlátozott mennyiségben állnak rendelkezésre, ezért költséghatékonyság szempontjából is fontos, hogy felesleges adatok auditálásával ne terheljük a rendszert. Azt is fontosnak tartom megemlíteni, hogy az auditálás mértékét úgy kell megválasztani, hogy az ne zavarja az auditált rendszer működését, hiszen ebben az esetben a „22-es csapdájába" esnénk, és a biztonság erősítése helyett gyengítenénk azt.

Miután meghatároztuk, hogy a cél elérése érdekében milyen tevékenységeket auditálunk, nem szabad megfeledkezni arról sem, hogy az összegyűjtött adatok önmagukban nem jelentenek semmilyen biztonságot. Az adatok úgy válnak hasznossá, ha belőlük információkat nyerünk. Ez az elv további segítséget nyújt a tervezőknek a feleslegesen gyűjtött adatok kiszűrésében, hiszen ha egy adatot semmilyen formában nem használunk fel információk előállításához, akkor arra az adatra nincs szükség.

Belső audit rendszerek

Egy komplex informatikai rendszer különböző komponensekből épül fel és szolgáltatásai között különféle funkciókat biztosít. Sok esetben integrált audit rendszert is tartalmaz. Ha az adatbázisokat tekintjük, akkor ott is sok esetben előfordul, hogy az adatbázis-kezelő rendszerek[12] saját audit funkciókat látnak el, amelyek a korábbi fejezetekben említett egyszerű naplózástól egészen az automatikus riasztásokat biztosító komplex auditálásig bezárólag nyújtanak szolgáltatásokat.

[12] DBMS: Database Management System.



30. ábra Belső audit rendszer

A fenti ábrán látható, hogy az audit rendszer integráns része az adatbázis-kezelő rendszernek. Ezek a belső audit rendszerek. Segítségükkel az érintett adatbázis ellenőrzése megvalósítható, az újabb és újabb fejlesztéseknek köszönhetően pedig egyre hatékonyabban végzik feladatukat. Például az FGA**Hiba! A könyvjelző nem létezik.** (Fine-Grained Auditing**Hiba! A könyvjelző nem létezik.**[13]) segítségével –ahogy a neve is mutatja – nagyon finom audit-beállítások is alkalmazhatók. [17]

A belső audit rendszereknek azonban vannak olyan tulajdonságaik, amelyek kétséget kizáróan szükségessé teszik más megoldások alkalmazását is. Az alábbi felsorolásban ezekről a problémákról lesz szó:

- *Függőség a DBMS-től (Database Management System)**Hiba! A könyvjelző nem létezik.***

  Mivel az auditálást maga az adatbázis-kezelő végzi, ezért az auditálás nem függetleníthető magától az adatbázis-kezelő rendszertől. Ha az adatbázis-kezelő összeomlik, éppen akkor szűnik meg az auditálás is, amikor a leginkább szükség lenne rá (rögzíteni kellene, hogy milyen események bekövetkezése vezetett a rendszer összeomlásához). Ez a függőség lehetővé teszi a támadók számára, hogy az adatbázis-rendszer feltörése esetén az audit rendszert is ellehetetlenítsék. Megengedhetetlen, hogy az auditált rendszer (adatbázis) hiányosságait, biztonsági réseit kihasználva kiiktatható az a rendszer (audit rendszer), aminek feladata éppen ezeknek a kiskapuknak a felismerése lenne. [15]

- *Jogosultsági problémák*

[13] Szó szerinti fordításban „finomszemcsés auditálást” jelent.

Az adatbázis-kezelő rendszerek felügyeletéért, karbantartásáért az adatbázis-adminisztrátorok (DBA**Hiba! A könyvjelző nem létezik.**, DataBase Administrator) felelősek. Ők az adatbázisok „teljhatalmú urai”, akik rendszerint maximális jogosultsággal rendelkeznek rendszerükhöz. Az auditálás egyik fő feladata annak biztosítása, hogy a jogosult felhasználók tevékenységei is követhetők és visszakövethetők legyenek. Természetesen egy olyan audit rendszer létjogosultsága, amelyben az ellenőrzött személyek befolyásolhatják a rendszer működését, megkérdőjelezhető.

- *Korlátozott együttműködés nagy rendszerek esetében*

  Nagy informatikai rendszerekben több kisebb önálló rendszer (adatbázis-rendszerek, alkalmazásszerverek stb.) működik, egymástól akár nagy földrajzi távolságokban. Biztonság szempontjából az önálló rendszerek hatással vannak egymásra, ezért az egész rendszer biztonságáról csak egy átfogó audit segítségével bizonyosodhatunk meg. A helyben meghozott következtetések nem relevánsak a rendszer egészének szempontjából, ezért központosítani kell az auditálást.

- *Nehezebb menedzselhetőség*

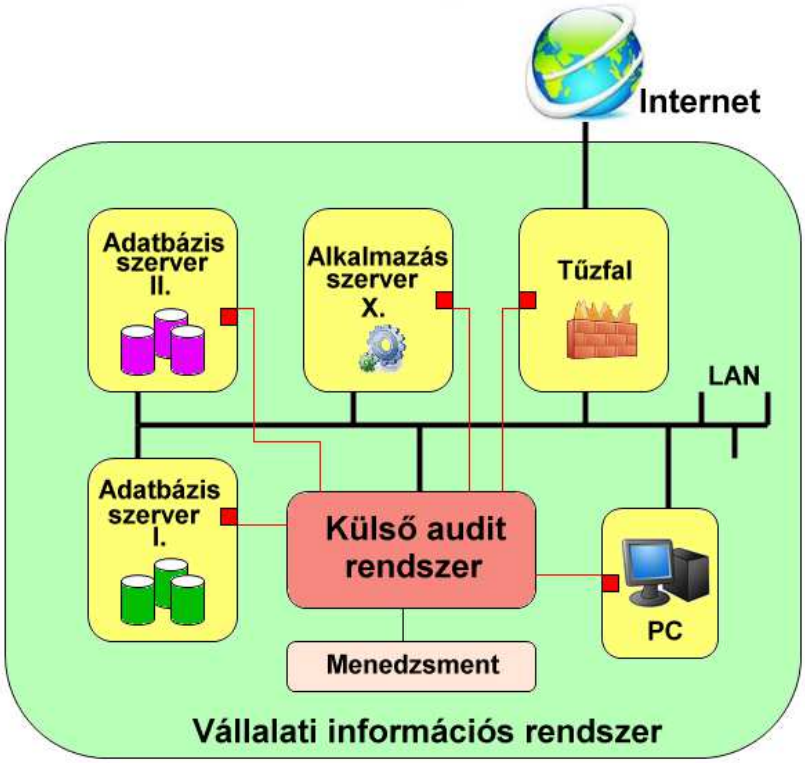
  A belső audit rendszerek helyi konfigurálása szükséges, ami jelentősen megnehezíti az adminisztrációt és lehetetlenné teszi a központi menedzselést. Gondoljunk csak bele, hogy ha a szervezet egy új ellenőrzési stratégiát készít, akkor azt csak a különálló audit rendszerek módosításával lehet megtenni. A cél az lenne, hogy a különböző rendszerek által szolgáltatott auditadatokat egy központi helyen lehessen tárolni, az elemzéseket ezen a széles körből származó adathalmazon lehessen elvégezni és végül, de nem utolsósorban a módosításokat is központilag lehessen érvényesíteni.

- *Terhelés*

  Egy audit rendszer működése során bizonyos idő elteltével már olyan hatalmas mennyiségű adattömeg keletkezik, amit helyben már nem lehet hatékonyan tárolni, illetve ehhez kapcsolódik, hogy az adatbázis-kezelő rendszerre előbb vagy utóbb olyan mértékű terhet ró az adatok elemzése, ami már hátráltatja „normál” teendőinek ellátásában. A következtetés az, hogy az auditadatok tárolásával és elemzésével kapcsolatos terheket le kell venni az auditált rendszerek „válláról”, és egy kifejezetten erre a célra fejlesztett rendszerre kell bízni.

Külső audit rendszerek

Az előző fejezetben bemutatott belső audit rendszerek hiányosságait egy független, külső audit rendszer alkalmazásával lehet kiküszöbölni. Egy független rendszer hatékonyabb védelmet nyújt a külső támadásokkal szemben, hiszen külön védelmi réteget jelent a korábban bemutatott mélységi védelmi stratégiában, és nagyobb eséllyel veszi fel a küzdelmet a belső támadások ellen is, mert a felügyelt rendszer adminisztrátoraitól független szakembergárda felelős a működtetéséért.



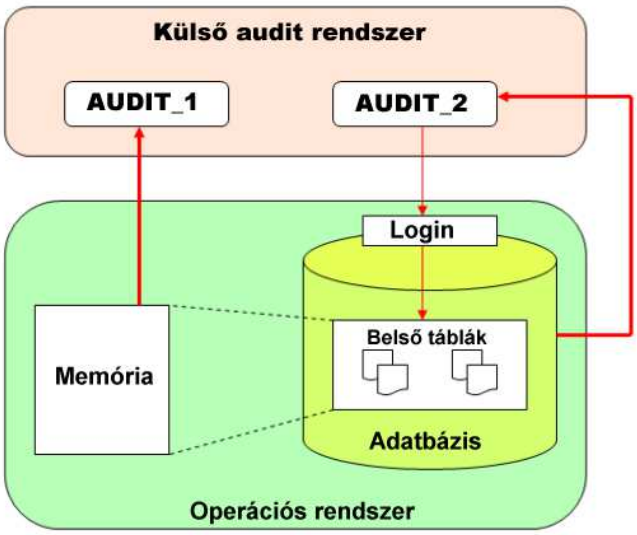
31. ábra Külső audit rendszer

A biztonság fokozása érdekében elképzelhető az is, hogy a külső és belső audit rendszerek egymással párhuzamosan, illetve egymást kiegészítve végezzék feladatukat, ezáltal az egyik rendszer esetleges meghibásodása vagy megtámadása esetén még mindig működik a másik rendszer, mintegy biztonsági tartalékként.

Adatbázisok auditálásakor a külső audit rendszerek három alapvető módszert alkalmazhatnak auditadatok összegyűjtésére.

- *Auditálás az adatbázis belső táblái alapján*

Az adatbázisok belső táblákat használnak az utasítások feldolgozására és az eredmények tárolására. Ezen táblák a feldolgozás során a memóriában helyezkednek el, különböző adatstruktúrák formájában.



32. ábra Belső táblák auditálása

Az adatbázisok működésének auditálására az egyik lehetőség, hogy ezeket a memóriában lévő adatstruktúrákat vizsgáljuk és próbálunk belőlük auditadatokat előállítani. Természetesen ebben az esetben az audit rendszernek pontosan ismernie kell a memóriában tárolt adatok struktúráját, és hozzáféréssel kell rendelkeznie a megfelelő memóriaterülethez. Előnye a módszernek, hogy nagyon alacsony szinten gyűjt adatokat, ezért kivételesen pontos ellenőrzést tesz lehetővé. Hátránya viszont az, hogy működtetéséhez tökéletesen ismerni kell a megfigyelt rendszer felépítését, továbbá időigényes feldolgozást kell végrehajtani a megfigyelt adatokon, amíg azokból hasznos auditadatok lesznek. Az ábrán az AUDIT_1 rendszer így gyűjti az adatokat.

Ugyanezeket az információkat el lehet érni úgy is, hogy az audit rendszer adatbázis-adminisztrátori jogosultságokkal bejelentkezik az adatbázis-szerverre, és egy lekérdezés segítségével jut hozzá a belső táblák tartalmához (AUDIT_2). Ez a módszer az előzőnél egyszerűbben teszi lehetővé az auditadatok megszerzését.

Mindkét esetben a belső táblák tartalma alapján következtetünk az adatbázis működésére, ezért fontos megjegyezni, hogy a tranzakciókkal kapcsolatos adatok csak korlátozott ideig állnak rendelkezésre a belső táblákban, így az audit rendszernek megfelelő gyorsasággal kell megszereznie az adatokat. Az AUDIT_2 által megvalósított módszer esetén ügyelni kell arra is, hogy az audit rendszer által kezdeményezett lekérdezések ne terheljék túlságosan az adatbázist.

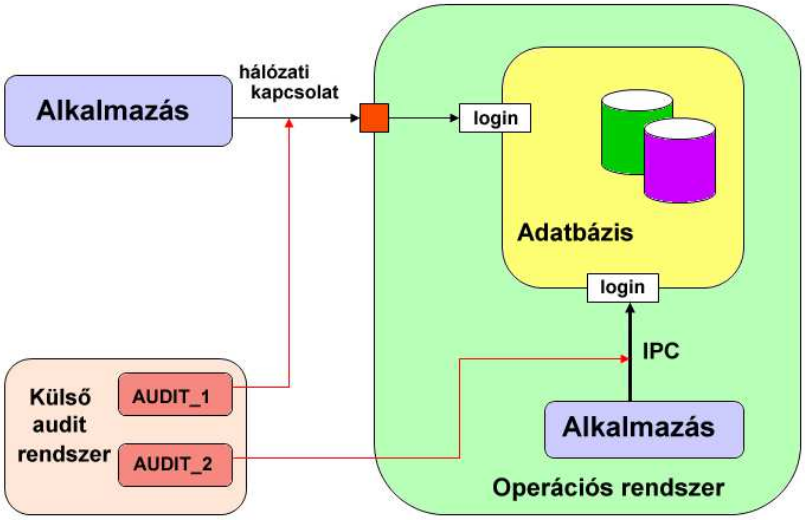
- *Auditálás az adatbázis kommunikációja alapján*

Napjaink adatbázisszerverei többnyire kommunikációs csatornákon keresztül szolgálják ki a klienseket. Az adatbázisok kizárólag ezeken a csatornákon keresztül kommunikálnak a külvilággal, innen érkeznek a kérések, és ezeken a csatornákon keresztül történik a kérések kiszolgálása is. Kézenfekvő lehetőség, hogy az adatbázissal kapcsolatos tevékenységek auditálását a kommunikációs csatornák megfigyelésével valósítsuk meg.

Attól függően, hogy helyi vagy hálózati kommunikációról van szó, kétféle auditálási lehetőség adódik.

Abban az esetben, ha a kapcsolódó alkalmazás az adatbázissal azonos operációs rendszerben található, akkor folyamatok közötti kommunikáció (IPC**Hiba! A könyvjelző nem létezik.**, Inter Process Communication) történik. Ez esetben a két folyamat közötti adatcserét kell vizsgálni és a megfigyelt adatokat eljuttatni az audit rendszerhez (az ábrán az AUDIT_2 végzi ezt a feladatot).

A másik esetben az alkalmazás egy távoli helyről, hálózati kommunikációs csatornán keresztül kapcsolódik az adatbázishoz. Ekkor a hálózati médián folyó adatokat kell vizsgálni, például úgy, hogy a hálózati réteg csomagjait vetjük alá vizsgálatnak[14] vagy a szállítási réteg – legtöbbször TCP vagy UDP – szegmenseiből nyerjük ki az audit szempontjából lényeges adatokat. Az ábrán az AUDIT_1 folyamat végzi a hálózati kapcsolat monitorozását.



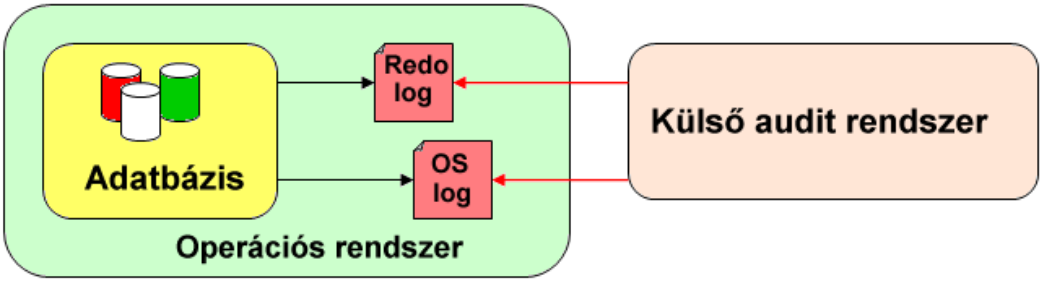
33. ábra Kommunikációs csatornák auditálása

- *Auditálás külső állományok alapján*

Szinte kivétel nélkül minden rendszer, köztük az adatbázis-kezelő rendszerek is, működésük során különböző állományokat használnak. Egyes fájlokat (például konfigurációs fájlokat) bemenetként használnak, míg másokat (például naplófájlokat) maguk állítanak elő.

Adatbázisok esetén tipikus állomány a tranzakciós napló, ami az esetek többségében tartalmazza az adatbázis struktúráját, és a benne tárolt adatokat érintő utasításokat. A másik széles körben használt fájl a helyreállítási napló (redo log**Hiba! A könyvjelző nem létezik.**), ami szintén információkat tartalmaz az adatbázissal kapcsolatban. Az auditálás folyamán ezekből az állományokból fontos auditinformációk állíthatók elő.

[14] Ilyen vizsgálati lehetőség a Deep Packet Inspection.



34. ábra Auditálás külső állományok alapján

Az ábrán az is látható, hogy az adatbázis környezetét befolyásoló különféle fájlok vizsgálata is érdekes lehet, ilyen például az adatbázis-szervert futtató operációs rendszer naplója (OS log).

Adatok kezelése

Az auditált rendszer sajátosságai döntik el, hogy az imént megismert három auditálási architektúrából melyiket érdemes választani. Mindhárom megoldás közös tulajdonsága, hogy nagy mennyiségű adatot gyűjt. Ezek az adatok képezik az audit rendszerek „lelkét", ezért körültekintően kell bánni velük.

- *Adatok tárolása*

Fontos észrevenni, hogy hatalmas mennyiségű adatot kell összegyűjteni és feldolgozni. Gondoljunk csak bele, hogy egyetlen adatbázis-rendszernek is milyen sok SQL**Hiba! A könyvjelző nem létezik.** utasítást kell végrehajtania. Ha az utasításoknak csak egy részét kell auditálni, akkor is hatalmas számokról van szó, nem beszélve arról, hogy egy komplex audit rendszernek több adatbázis- és egyéb rendszerek auditálásáról kell gondoskodnia.

Az adatok mennyiségének szemléltetésére példaként tekintsük a hazánkban működő egyik legnagyobb közösségi oldalt. A rendszert kb. 3 millió felhasználó használja. Feltételezzük, hogy egy felhasználó naponta átlagosan 10 műveletet végez (bejelentkezik, megnézi ismerősei friss képeit stb.). Ez a műveletsorozat kb. 50 adatbázis-tranzakciót eredményez. Ez napi 150 millió, éves szinten közel 55 milliárd auditeseményt jelent, és ezeket az adatokat rendszerint hosszú évekig megőrzik.

Ezt a hatalmas mennyiségű adathalmazt tárolni kell, és nemcsak egyszerűen tárolni kell, hanem lényeges, hogy feldolgozható állapotban kell tárolni. Az audit rendszerek többnyire adatbázisban tárolják az adatokat. A fenti számok egyértelművé teszik, hogy a hatékony tárolásra és feldolgozásra egy közönséges, hétköznapi adatbázis nem képes. Audit rendszerekben érdemes adattárházakat alkalmazni, mert ezek a rendszerek éppen az ilyen nagy mennyiségű adatokra vannak optimalizálva.

- *Adatok archiválása*

Az adatok archiválását három tényező indokolja.

Az egyik ok az, hogy az adatokról biztonsági mentést kell készíteni annak érdekében, hogy egy esetleges adatvesztés esetén a fontos adatok később visszaállíthatók legyenek.

Másrészről a törvényi és egyéb szabályok is megkövetelik az archiválást. Előírják, hogy bizonyos adatokat egy ideig kötelező megőrizni. Például évekig kell adatokat őrizni arról, hogy egy atomerőműbe kik léptek be. A szabályok sokszor években határozzák meg az adatok kötelező tárolási idejét.

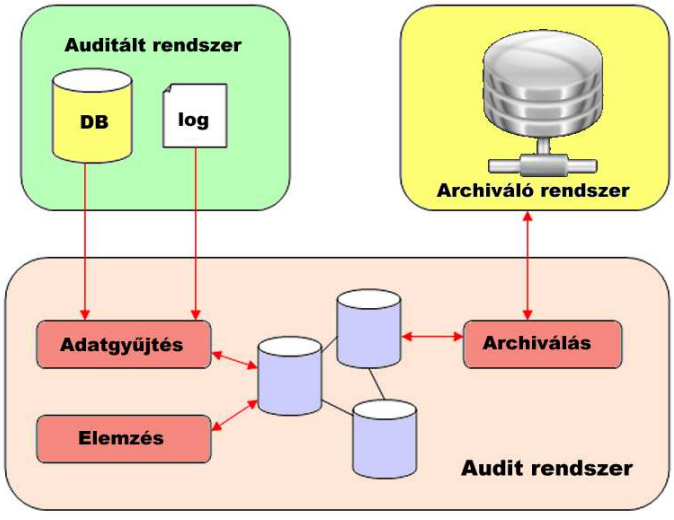
A harmadik ok a már említett hatalmas mennyiségű adatból következik. Egy idő után egyszerűen megtelnek a véges tárolási kapacitások, ezért az adatokat egy archiváló rendszerre kell bízni.

Függetlenül attól, hogy a felsorolt okok közül melyik vezet az archiválásig, olyan megoldást kell választani, amely hatékonyan és biztonságosan képes ellátni ezt a feladatot. Érdemes a feladatot a jól bevált tároló hálózatokra (SAN, Storage Area Network) bízni, amelyek optimális megoldást biztosítanak a rájuk bízott adatok redundáns archiválására.

- *Adatok védelmének biztosítása*

Miután megfelelő megoldást találtunk az auditinformációk tárolására és archiválására, gondoskodnunk kell azok biztonságáról. Ez egy többlépcsős feladatnak ígérkezik, hiszen a védendő adatok hosszú utat tesznek meg a forrásrendszerektől az archiválást végző rendszerekig.

Az adatok útja az alábbi ábrán követhető.



35. ábra Auditadatok útja

Az adatgyűjtés során rendszerint távoli rendszerekből történik az adatok kinyerése. Ezek hálózati kommunikációs csatornán keresztül jutnak el az audit rendszerhez. Az adatok védelme érdekében a hálózati csatorna védelmét kell ellátni, amire a korábbi fejezetekben ismertetett algoritmikus módszerek, közöttük is leginkább a titkosítási algoritmusok alkalmazhatók.

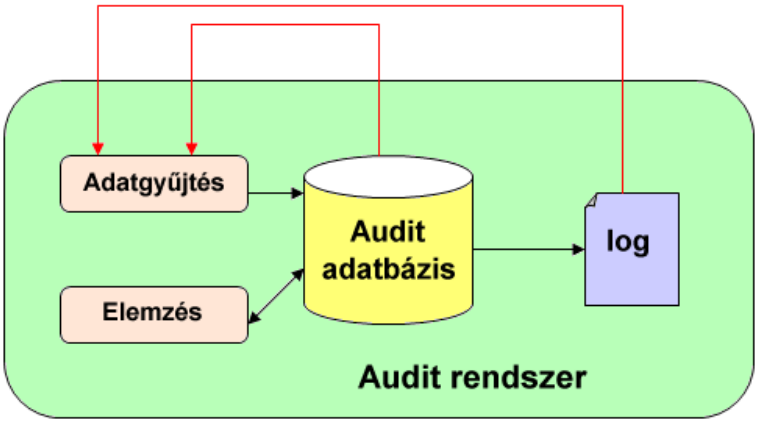
Az audit rendszer szempontjából érdekesebb feladat az adattárolásra használt adatbázis és a benne tárolt adatok védelme. Ezt a rendszert külső felhasználóknak nem szabad elérni, hozzáférést csak az audit rendszerből szabad engedélyezni. Különleges figyelmet kell szentelni a hozzáférési szabályok kidolgozására. Lévén, hogy ez is egy adatbázis, feltehetően ennek üzemeltetését is adatbázis-adminisztrátorok végzik majd. Biztosítani kell, hogy az audit adatbázis is éppen olyan szigorú szabályok szerint legyen ellenőrizve, mint az auditált rendszerek. Az adminisztrátorok teljhatalmának csökkentésére tevékenységeik naplózása mellett egy új módszert is kifejlesztettek. Ennek a módszernek a lényege, hogy a rendszergazdáknak csak az adatbázis struktúrájához van hozzáférésük, a tárolt adatokat nem tudják megnézni.[15] Ezzel minden korábbinál hatékonyabban megoldható az érzékeny adatok védelme úgy, hogy az adminisztrátorok nincsenek akadályozva munkájuk ellátásában.

Végül az adatok biztonságáról akkor sem szabad megfeledkeznünk, amikor azokat archiválják. Azért sem szabad alulbecsülni az archív adatok védelmét, mert itt már nemcsak pusztán az auditált rendszerből kinyert nyers adatokat helyezik el, hanem sokszor az elemző algoritmusok által előállított következtetéseket is mentik. Ha egy támadó hozzájutna ezekhez az információkhoz, akkor talán még nagyobb kárunk lenne. A fenti ábrán látható, hogy archiváláskor az adatokat egy külső rendszerre bízzuk, hiszen az audit rendszert nem ilyen feladatokra készítették, nála sokkal hatékonyabban képes ellátni ezt a funkciót egy olyan rendszer, amit erre a feladatra optimalizáltak. Az előnyök mellett hátrány viszont, hogy elveszítjük az ellenőrzést az adatok felett. Ezért ebben az esetben is érdemes bevetni az algoritmikus eszközöket, például úgy, hogy egy erős algoritmussal titkosítjuk az adatot, majd egy digitális aláírással látjuk el. Így biztosak lehetünk abban, hogy az adatokhoz illetéktelenek nem férhetnek hozzá, és egy későbbi visszaolvasáskor arról is meggyőződhetünk, hogy az archív tárból visszaolvasott adatot valóban mi helyeztük el.

Az audit rendszer biztonsága

Az auditadatok az előző fejezetben bemutatott intézkedések után biztonságban tudhatók, azonban védekezni kell magát az audit rendszert érő támadásokkal szemben is. Ha egy jogosulatlan vagy akár egy rosszindulatú, jogosult felhasználó módosítja az audit rendszer beállításait, például szűkíti a gyűjtendő adatok körét, akkor hiába vannak biztonságban adataink, a rendszer nem fog céljainknak megfelelően működni.

[15] Egy ilyen megoldást nyújt az Oracle Database Vault rendszere.



36. ábra Audit rendszer auditálása

Bármilyen furcsán hangzik, szükség van az audit rendszer auditálására. Ugyanúgy adatokat kell gyűjteni a fontos eseményekről, elemezni kell őket és szükség esetén figyelmeztetéseket kell generálni. Az audit rendszer ellenőrzését bízhatjuk egy független rendszerre, de legtöbbször önmagának auditálása is megoldható.

Adaptálhatóság

Egy audit rendszer adaptálhatósága azt jelenti, hogy a rendszer nagyobb átalakítások nélkül képes alkalmazkodni a változó követelményekhez.

Az auditálásnak követnie kell a vállalati rendszerek változásait, és igazodnia kell a folyamatosan változó törvényi szabályozáshoz is. Ebből következik, hogy az auditkövetelményeket is folyamatosan újra kell értékelni. Feltétlenül olyan rendszert kell készíteni, amely rugalmasan tud alkalmazkodni a változásokhoz.

A független, külső auditrendszerek esetén az auditálás három fő fázisából az adatok feldolgozását és az eredmények elkészítését központosítva oldották meg, azok módosítása és követelményekhez illesztése nem okoz különösebb problémát. A kritikus fázis az auditadatok összegyűjtése. Ha megváltoznak a követelmények, akkor előfordulhat, hogy más adatokra van szükség, mint amikkel eddig dolgoztunk. Alapvetően két szemléletmódot lehet megkülönböztetni:

- *Mennyiségi adatgyűjtés**Hiba! A könyvjelző nem létezik.***

Ez azt jelenti, hogy a rendszer a lehető legtöbb információt igyekszik összegyűjteni, és csak a feldolgozás során dől el, hogy ezek közül melyeket használják fel. Ez a szemléletmód nagyon megkönnyíti a változások követését, mert csak az adatok felhasználásának módját kell megváltoztatni, hiszen minden adat rendelkezésre áll. Természetesen hátránya a módszernek a felesleges adatok gyűjtésével kapcsolatos erőforrások pazarlása.

- *Minőségi adatgyűjtés**Hiba! A könyvjelző nem létezik.***

Ebben az esetben csak azoknak az adatoknak a gyűjtése történik, amelyek feltétlenül szükségesek a követelményekben leírtak teljesítéséhez. A változások kezelése itt

nehezebben kivitelezhető, mert az adatgyűjtés szabályait is módosítani kell az elvárt működés biztosításához. Előnye, hogy nem használja feleslegesen az erőforrásokat.

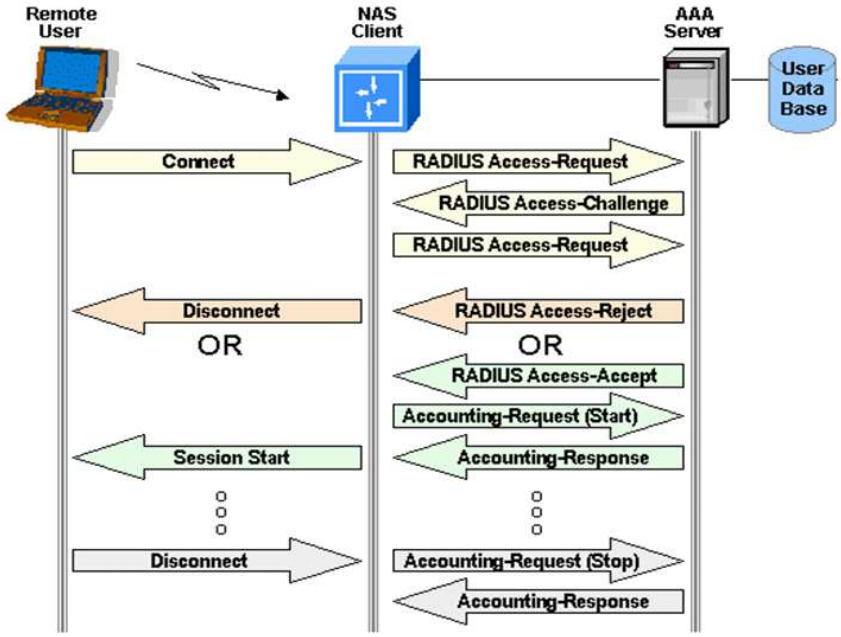
Mint mindig, most is egy egészséges kompromisszummal érhető el a maximális hatékonyság. Bizonyos területekről érdemes több adatot szerezni, máshonnan pedig elég a minimálisan szükséges adatok gyűjtése.

3.5. Protokollok

3.5.14. RADIUS (Remote Access Dial-In User Service)

A RADIUS-t, a legelterjedtebb AAA protokollt az IETF által a 2865-ös RFC (Request for Comments) dokumentumban specifikálták. Eredetileg egy kliens szerver felépítésében szereplő felhasználó szerverre való authentikációját írta le PAP vagy CHAP segítségével. A dokumentumban nem esett szó az audit folyamatáról. Utóbbit egy külön dokumentumban rögzítették (RADACC2866). [12]

Egy RADIUS architektúrában meg kell különböztetnünk a végponton lévő klienst és a RADIUS klienst, ez utóbbi ugyanis jellemzően a NAS (Network Access Server). Azonosításkor a RADIUS kliens azért felel, hogy megfelelő formában továbbítson egy kérést a RADIUS szerver felé a végponton lévő kliens adataival, majd ezt követően várja a szerver válaszát.



37. ábra RADIUS protokoll létradiagram (forrás: Wikipédia)

RADIUS-üzenetek

- Access Request (NAS→AS): A RADIUS kliens generálja a szerver felé, hogy továbbítsa a kérelmező adatait.

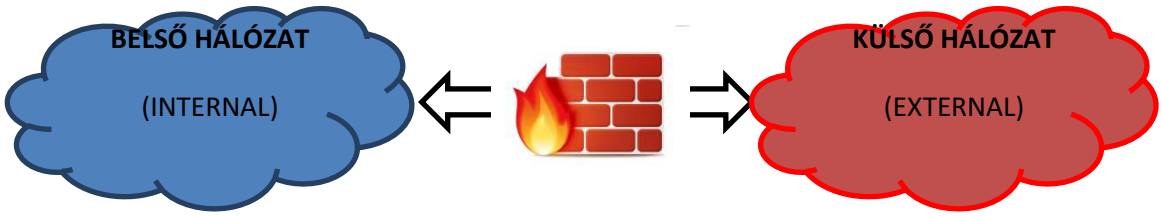
- Access Challenge (AS→NAS): A szerver egy kihívást küld a kliens felé.
- Access Accept (AS→NAS): A sikeres kérés jelzésére küldi a szerver a kliensnek.
- Access Reject (AS→NAS): Abban az esetben küldi a szerver a NAS-nak, ha a felhasználó adataival nem sikerült a kérést teljesíteni.
- Accounting Request (NAS→AS): Ezzel szállítja a végponton lévő klienstől a szerver felé a naplózási információkat a szolgáltatás alapján.
- Accounting Response (AS→NAS): A szerver ezzel jelzést ad a RADIUS kliens irányába, hogy a kliens információi sikeresen célba értek.

4. Tűzfalak

4.1. Fogalmak

A biztonsági szakemberek – legyenek azok informatikabiztonsággal vagy fizikai biztonsággal foglalkozók – jól ismerik a határvédelem (perimeter security) fogalmát. Egy épület fizikai védelme esetében első védelmi vonalként egy kerítést építünk a védendő telekre, majd az épület külső falait és nyílászáróit védjük. A határvédelmi eszközök célja, hogy a támadók kívül maradjanak a határvonalon, hiszen így könnyebb megvédeni az értékeinket. Ezek után természetes dolognak tűnik, hogy a belső támadókkal szemben nem jelent védelmet a határvédelem.

Az informatikabiztonságban a hálózat határán elhelyezett védelmi eszközöket *tűzfal*aknak (firewall) nevezzük.



38. ábra Tűzfal

A hálózat határán különböző eszközök helyezkedhetnek el. Itt találhatók a routerek, proxy szerverek és átjárók, amelyek mindegyike alkalmas lehet a tűzfal funkcióinak ellátására.

Routerek

A routerek hálózatok összekapcsolására szolgáló eszközök, amelyek az egyik hálózatból a másikba áramló csomagok irányítását végzik. Szerepük meghatározó a hálózatok világában, nélkülük elképzelhetetlen a hálózatok működése. [12]

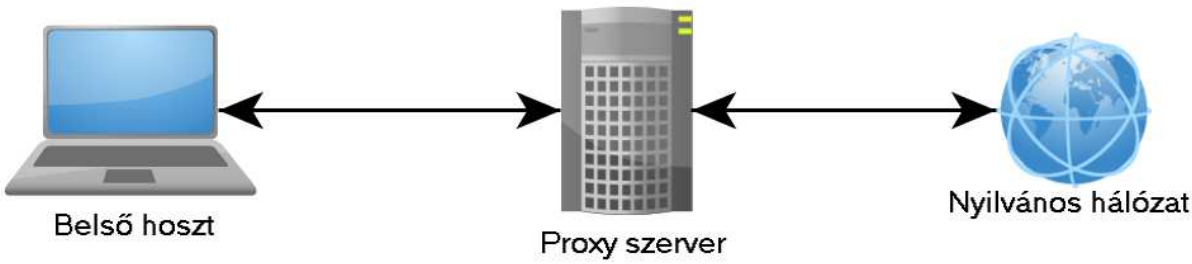
Azáltal, hogy a routereken „keresztülfolyik” a hálózati forgalom, tökéletesen alkalmasak a forgalom szűrésére. Megfelelően beállított szabályrendszerrel a segítségükkel kiszűrhető a nemkívánatos forgalom, hiszen eldönthetjük, hogy milyen üzeneteket engedünk be a hálózatunkba. Fontos lesz az is, hogy a kifele menő forgalmat is szűrjük.

Legyen szó egyszerű SOHO (Small Office Home Office) routerekről vagy nagy teljesítményű vállalati routerekről, általánosságban elmondható, hogy ezek az eszközök tartalmaznak biztonsági funkciókat. Például a Cisco routerek hozzáférés-vezérlési listái (ACL, Access Control List) segítségével csomagszűrést végezhetünk. Ezenkívül a routerek segítséget nyújtanak a szolgáltatásmegtagadásos DoS támadások kivédésében is azáltal, hogy csökkentik a megtámadott szerver felé küldött adatok mennyiségét.

A rengeteg biztonsági segédeszköz mellett egyet mindenképpen érdemes még megemlíteni. Ez a hálózati címfordítás NAT (Network Address Translation) és a portfordítás PAT (Port Address Translation) szolgáltatás. Ezeket a szolgáltatásokat szintén routerek végzik és segítségükkel a belső (privát) hálózat „elrejthető” a külső (publikus) hálózat elől. Így a támadók nehezebben találják meg a célpontokat, és ha meg is találják, a célpontok megtámadása is komplikáltabb lesz.

Proxy

A számítógép-hálózatokban a proxy szerverek közvetítő szerepet játszanak a belső hálózat végberendezései (kliensei) és a külső hálózat szerverei között. Ahelyett, hogy a kliens közvetlenül a külső hálózat szerveréhez fordulna a kérésével, először a saját proxy szerveréhez fordul, amelyik megvizsgálja a kérést, és ha minden rendben van, akkor a proxy a kliens nevében fog a külső hálózat szerveréhez fordulni. A proxy beállításainak megfelelően megváltoztathatja, vagy akár el is utasíthatja a kliens kérését. A külvilág felé tehát minden kérés a proxy szerveren keresztül valósul meg. Ez azért hasznos, mert a védett hálózat gépeit kontrollálhatjuk. Például egy potenciálisan veszélyes weboldal meglátogatásakor a proxy közbe tud avatkozni és le tudja tiltani a kapcsolatot.



39. ábra Proxy szerver elhelyezkedése

Emellett a proxynak fontos szerepe lesz a korábbi fejezetekben bemutatott audit logok rögzítésében is. A belső hálózat minden, külvilág felé irányuló kérése segítségével rögzíthető.

Érdemes megjegyezni azt is, hogy a proxy szerver fontos szerepet játszik a hálózati forgalom optimalizálásában is. Ezt úgy teszi, hogy megjegyzi a korábbi kéréseket és rögzíti (cache-eli) a rajta keresztül letöltött objektumokat. Ha később egy hoszt olyan kérést intéz hozzá, amelyet korábban már kiszolgált, akkor a hálózat tehermentesítése érdekében saját tárolójából szolgálja ki a kérést.

Tűzfal

A tűzfal funkcióját elláthatja bármely eszköz, amely a hálózat határán helyezkedik el és megfelelő „intelligenciával” rendelkezik. Ezen a ponton azonban el kell oszlatni egy tévhitet, amely nagyon sok felhasználó és köztük sok informatikus fejében él. A tűzfalak nem csodaszerek. Nagyon hasznos és fontos eszközei a védelemnek, de korántsem alkalmasak mindenre.

4.2. Típusok

A különböző védelmi feladatokra különböző tűzfalak állnak rendelkezésre. E fejezet célja áttekinteni ezek típusait.

4.2.1. Hardveres és szoftveres tűzfalak

A tűzfalak megválasztásakor az első kérdés, amit meg kell válaszolnunk, hogy hardveres vagy szoftveres tűzfalra van-e szükségünk. A hardveralapú tűzfalak (tűzfal készülékek) olyan speciális eszközök, amelyek tartalmazzák mind a hardveres és szoftveres részeit egy tűzfalnak, ráadásul a készüléket kizárólag erre a célra tervezték. Az adminisztrátorok számára gyakran grafikus felületet biztosítanak, ahol minden szükséges beállítás elvégezhető. Legfontosabb előnyük, hogy szoftveres társaikhoz képest gyorsabban végzik feladatukat, így nagy sebességű hálózatokban (akár gerinchálózatokban) is alkalmazhatók. A sebességnek persze ára van, ezek az eszközök viszonylag drágák.

A szoftveres tűzfalak relatíve olcsók, viszont a szoftver licencének megvásárlása mellett saját magunknak kell gondoskodni a megfelelő hardverről. Persze léteznek nyílt forráskódú, ingyenes tűzfalak is. Ilyen tűzfalakat a korlátozott feldolgozási sebességük miatt inkább kisebb hálózatokban érdemes használni.

4.2.2. Csomagszűrő tűzfalak (packet filter firewall)

A csomagszűrő tűzfal a legegyszerűbb és legrégebbi megoldás. A tűzfalon átmenő csomagok sorsáról egy előre definiált szabálygyűjtemény alapján dönt. Vagy eldobja azt (deny) vagy továbbengedi (permit). A csomagokat független objektumként kezeli (stateless, azaz állapotmentes), a szűrési feltételek a protokoll fejrészekre korlátozódnak, nem vonatkoznak a csomag törzsére. Sokféle fejlécparaméter vizsgálatára van lehetőség, de a leggyakrabban vizsgált paraméterek az alábbiak:

- forrás IP-cím,
- cél IP-cím,
- célport, szolgáltatás,
- forrásport,
- szállítási protokoll (pl.: TCP vagy UDP).

A fejrészparamétereken kívül egyéb tényezőket is számításba vehetünk a szűréskor. Például időfüggő szűrés alkalmazásával biztosítható, hogy egy cégnél munkaidőben más szabályok legyenek érvényben, mint munkaidőn kívül.

A csomagszűrő tűzfalak erőssége gyorsaságukban rejlik, illetve abban, hogy a felhasználók számára transzparensen működnek. Hátrányuk viszont az, hogy a csomagok tartalmának szűrésére alkalmatlanok, így azokban akár malware adatok is átjuthatnak a védelmen.

4.2.3. Állapotgép-alapú tűzfalak (stateful packet inspection)

A csomagszűrő tűzfalak utáni következő generációt az állapotgép-alapú tűzfalak képviselik. Nevezik őket kapcsolatalapú tűzfalnak (circuit-level firewall) is.

Elődeikhez hasonlóan ezek is előre definiált szabályrendszert használnak, viszont képesek a csomagok többszintű vizsgálatára, hiszen nem egymástól függetlenül vizsgálják a csomagokat, hanem képesek az összetartozókat együtt kezelni. Fogalmazhatunk úgy is, hogy képesek a kapcsolatok kezelésére.

Tekintsük az alábbi példát. Egy webböngésző kérést küld a webszervernek:

- a kliens az 5678-as forrásportról küldi a csomagot a szerver 80-as portjára,
- a szerver elfogadja a kérést és válaszol az 5678-as portra.

Ennek hatására létrejön egy socket kapcsolat, amelyben csomagok közlekednek a kliens és a szerver között. Míg a csomagszűrő tűzfalak ezeket a csomagokat különálló egységként kezelnék, addig az állapotgép-alapú tűzfalak felismerik, hogy itt egy kapcsolat (connection) alakult ki, és képesek ezt egységként kezelni. Itt már lehetőség van a kapcsolatokra vonatkozólag dönteni azok biztonságáról, és miután egy kapcsolatot biztonságosnak ítélt, tovább már nem kell elemezni az ahhoz tartozó csomagokat. Ezzel a technikával akár a csomagszűrő tűzfalaknál is gyorsabb lehet.

4.2.4. Alkalmazási szintű tűzfalak (application gateway firewall)

Ezen a szinten az alkalmazási tűzfal minden kapcsolatot szétvág, majd közvetítőként működve biztosítja a kapcsolatot. Minden kommunikációs kapcsolatot elfog és maga építi ki a kapcsolatot a címzettel. Így lehetőség nyílik az OSI modell minden rétegében szűrést végezni.

Ezzel a típussal lehet a legnagyobb bizonyossággal kivédeni a nemkívánatos tartalmakat, viszont ez erősen a sebesség rovására megy.

4.2.5. Hibrid tűzfalak

A hibrid tűzfalakat a fenti három típus előnyös tulajdonságainak ötvözésével alakítják ki. Manapság leginkább ilyen tűzfalakkal találkozhatunk.

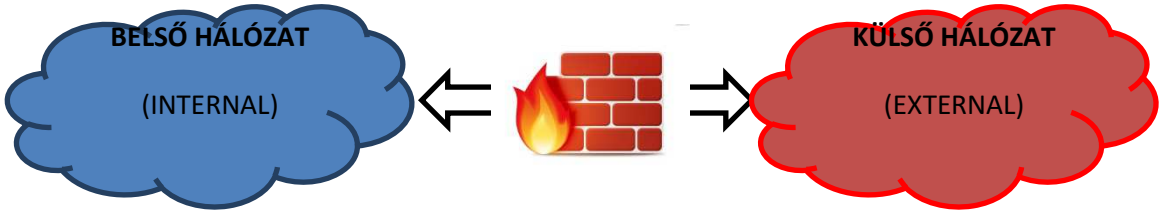
4.3. Tűzfal-topológiák

Annak érdekében, hogy a tűzfalunk hatékonyan lássa el feladatát, megfelelő helyre kell tenni a hálózatban. Alapvetően mindig a belső és a külső hálózat között működnek, de így is többféle topológia lehetséges. Ebben a fejezetben a három legelterjedtebb topológia bemutatása következik.

4.3.6. Bástya topológia

A bástya topológiában (bastion host topology) a tűzfal a külső és a belső hálózat között helyezkedik el. Minden hálózati forgalom keresztülmegy a tűzfalon. Ennek a topológiának az

implementálása a legegyszerűbb, azonban komoly biztonsági kockázatot is jelenthet abban az esetben, ha a belső hálózatban olyan szervereket szeretnénk üzemeltetni, amelyeket elérhetővé kívánunk tenni a külvilág számára is.

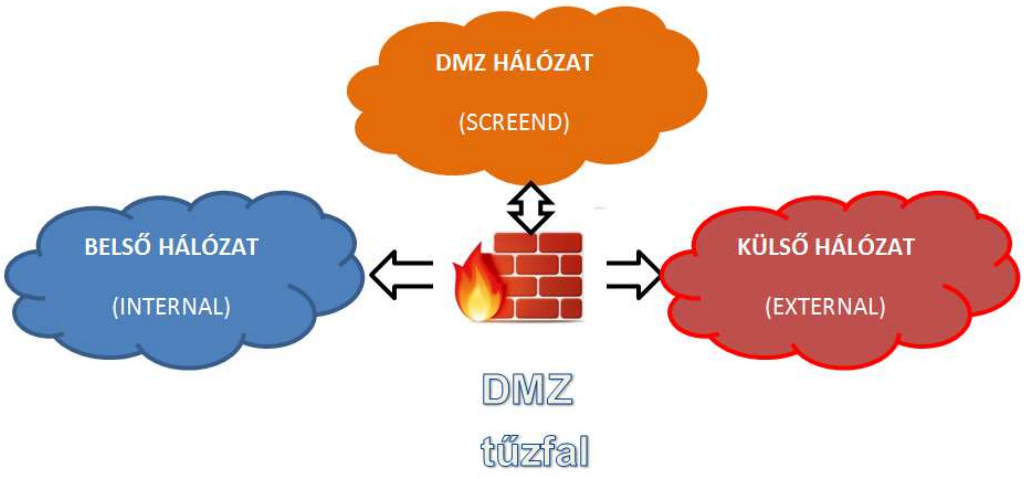


40. ábra Bástya topológia

A kockázatok megértéséhez tegyük fel, hogy a belső hálózatban egy webszervert üzemeltetünk. Természetesen annak érdekében, hogy a kliensek elérjék a szervert, meg kell nyitni a 80-as portot a tűzfalon. Miután a szerver hozzáférhető a külvilágból, előfordulhat, hogy egy támadó átveszi az irányítást a szerver felett. Így már könnyen megtámadhat minden gépet a belső hálózatban, hiszen bejutott a tűzfal mögé, a többi gépet már nem védi más tűzfal. [18]

4.3.7. Demilitarizált zóna (DMZ)

A DMZ (DeMilitarized Zone) vagy más néven védett alhálózati architektúra (screend subnet architecture) is egyetlen tűzfalat használ, de kettő helyett most már három hálózat összekötésére. Az egyik hálózat a külső (nyilvános) hálózat, ahonnan a támadások érkezhetnek. A második hálózat a belső hálózat, amely a tűzfal teljes védelmét élvezi. A harmadik hálózat – amelyről a topológia a nevét kapta – a DMZ hálózat. Utóbbi szintén a tűzfalon keresztül érhető el, és ebbe a hálózatba helyezzük el azokat a szervereket, amelyek szolgáltatásokat nyújtanak a külső hálózat számára.

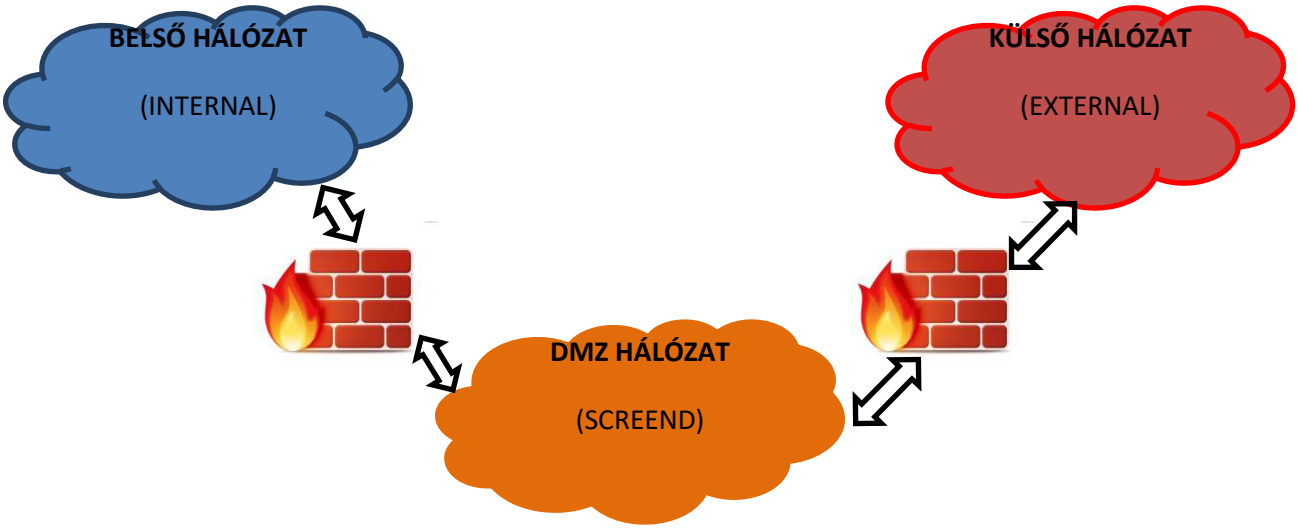


41. ábra DMZ topológia

A DMZ hálózat célja egy olyan hálózat létrehozása, amely a külső és a belső hálózatok között átmenetet képez. Ha egy támadó az előző példához hasonlóan hozzáférést szerez valamely szerverhez, akkor továbbra sem férhet hozzá a belső hálózat védett elemeihez.

4.3.8. Kettős tűzfal topológia

Az alábbi ábrán látható a kettős tűzfal topológia (dual firewall topology) lényege.



42. ábra Kettős tűzfal topológia

Alapvetően ugyanaz az elv érvényesül ebben a topológiában is, mint az egyszerű DMZ topológiában, viszont itt egy helyett kettő tűzfal látja el a védelmi funkciókat. Ennek előnye, hogy a DMZ hálózatban lévő – publikusan elérhető – objektum megtámadása esetén csökkenti a belső hálózatnak a fenyegetettségét. A szerver megtámadása esetén a támadó próbálkozhat a tűzfal támadásával. Ha ez is sikerül neki, akkor még mindig van egy védelmi réteg a belső hálózat előtt (a másik tűzfal). Éppen ezért fontos, hogy a gyakorlatban törekedjünk különböző tűzfalak implementálására, ezáltal is nehezítve a támadók dolgát.

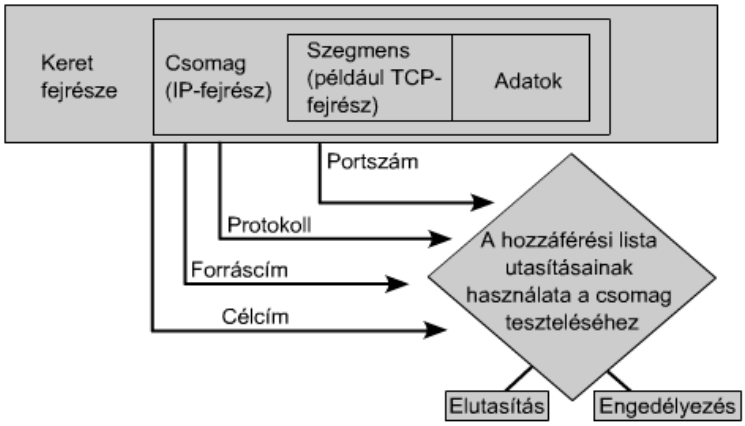
A két tűzfalnak az alábbiakban kell különböznie:

- az egyik legyen hardveres, a másik szoftveres tűzfal,
- különböző gyártótól szerezzük be őket.

4.4. Hozzáférés-vezérlési listák (ACL)

A hozzáférés-vezérlési listák feltétellisták, amelyek a router interfészein keresztülmenő forgalomra fejtik ki hatásukat. A tűzfalak típusai közül a csomagszűrő tűzfalak családjába soroljuk őket. Ebben a fejezetben bemutatjuk a Cisco routereken alkalmazott ACL-ek működési elvét, amely a korábban használt ACL megoldások egy leképzése (ld. authorizációról szóló fejezet).

A forgalom szűrése a hálózati csomagok és a szállítási szegmensek fejrészei alapján történik:

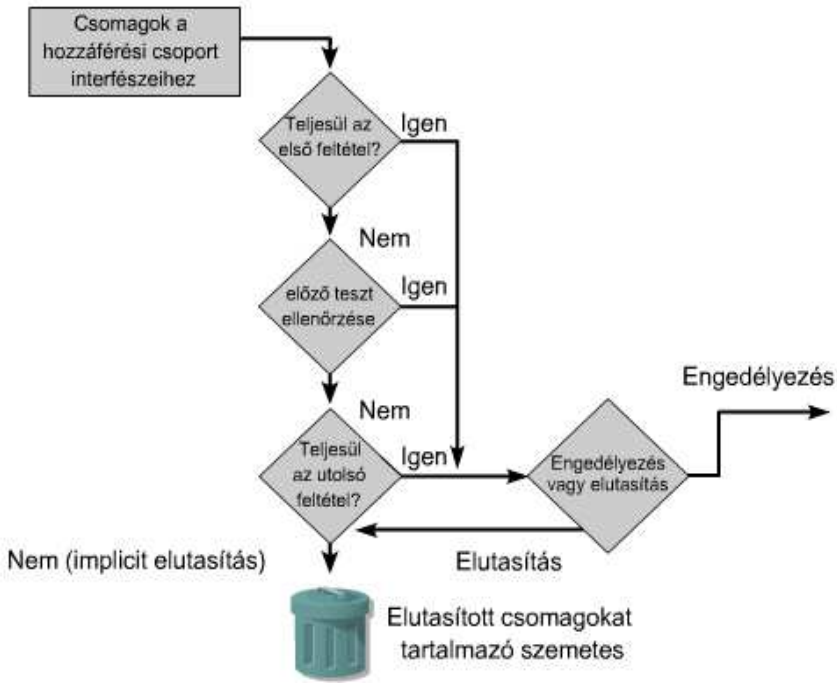


43. ábra ACL szűrési paraméterek (forrás: www.netacad.com)

Amikor egy csomag megérkezik a routerre, az lefejti az adatkapcsolati fejrészt és megvizsgálja az IP-csomag, valamint a transzport szegmens alábbi paramétereit:

- forrás és cél IP-cím,
- forrás és cél portszám,
- protokollok.

Ezután az interfészre beállított feltétellista feltételeit értékeli ki. Ha egy feltételre illeszkedik a csomag, akkor az aktuális feltételsorban meghatározott műveletet végrehajtja.



44. ábra ACL működése (forrás: www.netacad.com)

A művelet kétféle lehet:

- permit: a csomagot engedélyezték, átadható a router csomagtovábbító alrendszerének;
- deny: a csomag áthaladása megtagadva, a csomagot eldobja.

Amennyiben a feltétellista egyetlen szabályára sem illeszkedik a csomag, úgy az alapértelmezett (implicit) szabály lép életbe. A Cisco rendszerekben a szigorú szabályozás az alapértelmezett (deny any), ami azt jelenti, hogy illeszkedés hiányában a csomagot eldobja.

Az alábbi példákban egy normál és egy kiterjesztett ACL létrehozását láthatjuk.

```
Router(config)#ip access-list standard Proba
Router(config-std-nacl)#deny host 192.168.1.5
Router(config-std-nacl)#deny host 192.168.1.6
Router(config-std-nacl)#permit any
```

45. ábra Normál ACL létrehozása

```
Router(config)#ip access-list extended Proba_kiterjesztett
Router(config-ext-nacl)#deny tcp host 10.0.0.1 any eq www
Router(config-ext-nacl)#deny tcp host 10.0.0.1 any eq 21
Router(config-ext-nacl)#permit ip any any
Router(config-ext-nacl)#
Router(config-ext-nacl)#
```

46. ábra Kiterjesztett ACL

```
Router#show access-lists
Standard IP access list Proba
    10 deny host 192.168.1.5
    20 deny host 192.168.1.6
    30 permit any
Extended IP access list Proba_kiterjesztett
    10 deny tcp host 10.0.0.1 any eq www
    20 deny tcp host 10.0.0.1 any eq ftp
    30 permit ip any any
Standard IP access list 1
    deny host 192.168.1.1
    permit any
Extended IP access list 155
    permit ip host 10.0.0.1 any
    permit ip host 10.0.0.2 any
```

47. ábra Beállított ACL-ek ellenőrzése

5. IRODALOMJEGYZÉK

[1] Muha Lajos, Krasznay Csaba – Az elektronikus információs rendszerek biztonságának menedzselése, NKE, Budapest, 2014.

[2] Munk Sándor – Információbiztonság Vs. Informatikai biztonság, Hadmérnök folyóirat, Budapest, 2007.

[3] Muha Lajos: A Magyar Köztársaság kritikus információs infrastruktúráinak védelme, PhD értekezés, ZMNE, Budapest, 2007.

[4] William Stallings, Lawrie Brown – Computer Security, Pearson, 2012.

[5] Pandur Béla – Informatikabiztonság, egyetemi jegyzet, Pécs, 2011.

[6] Gyurák Gábor – Adatbázis audit rendszerek, Budapest, 2010.

[7] Chuck Easttom – Computer Security Fundamentals, Pearson, 2006

[8] Wm. Arthur Conklin, Gregory White – CompTIA Security+, McGraw-Hill, 2011

[9] Eric Maiwald - Fundamentals of Network Security, McGraw-Hill, 2004.

[10] Randall J. Boyle – Corporate Computer Security, Pearson, 2013.

[11] Muha Lajos – Az informatikai biztonság egy lehetséges rendszertana: Az információbiztonság egy lehetséges taxonómiája. Bolyai Szemle XVII: (4), 2008.

[12] James F. Kurose – Számítógép-Hálózatok Működése, Panem, 2008.

[13] Buttyán Levente, Vajda István – Kriptográfia és alkalmazásai, Typotex, Budapest, 2004.

[14] Fleiner Rita – Az adatbázis biztonság alapjai, Hadmérnök V/2., 2010.

[15] Gajdos Sándor – Adatbázisok, Műegyetem kiadó, Budapest, 2004.

[16] Stuart J. Russell, Peter Norvig – Mesterséges Intelligencia, Panem, Budapest, 2000.

[17] Ron Ben Natan – Implementing Database Security and Audit, Elsevier, 2005.

[18] RandyWeaver, Dawn Weaver - Guide to Tactical Perimeter Defense: Becoming a Security Network Specialist, 2008.

[19] Berta István – Nagy e-szignó könyv, Microsec Kft., Budapest, 2011.

[20] Haig Zsolt, Kovács László – Kritikus infrastruktúrák és kritikus információs infrastruktúrák, Tanulmány, Nemzeti Közszolgálati Egyetem, 2012.

[21] Gonda János – A rejtjelezés néhány kérdése, ELTE, Budapest, 2010.

[22] Dénes Tamás – TitokTan Trilógia, Bagolyvár Kiadó, Budapest, 2004.

[23] Simon Sight – Kódkönyv, Park Könyvkiadó, 2007.

[24] ZHOU, M., GENG, G., WU, Z. – Digital Preservation Technology for Cultural Heritage, Springer, Berlin, 2012.

[25] Muha Lajos – Formális biztonsági modellek I. , A diszkrecionális hozzáférés-védelem, Hadmérnök VII. évfolyam 1.szám, Budapest, 2012.

[26] Déri Zoltán, Lobogós Katalin, Muha Lajos, Sneé Péter, Váncsa Julianna (szerk.: Muha Lajos): Informatikai Biztonság Irányítási Követelmények (IBIK), Budapest: Miniszterelnöki Hivatal, 2008. 275 p., (Közigazgatási Informatikai Bizottság ajánlásai; 25.), 1-2., Magyar Informatikai Biztonsági Ajánlások (15.3 pont)