# Biztonság alapvető fogalmak

Programtervező informatikus BSc
Számítógépes hálózatok biztonságtechnikája
előadás

## Biztonság

- A biztonság fogalma
  - Adatbiztonság
  - Szolgáltatás biztonsága
  - Security (titkosítás)
  - Fizikai biztonság

## Fizikai biztonság

- Az üzemeltetés során felmerülő problémák azon együttese, melyek szándékos vagy tőlünk független történések, de az adatok és használt rendszerek biztonságával, ha úgy tetszik a hardver-eszközök biztonságával kapcsolatosak.
  - Fizikai hozzáférés
  - Lopás
  - Elemi károk (tűz, villám, stb.)

## Szerverszoba kialakítása

- Megfelelően kialakított, zárható ajtó
- Klimatizáció
- Automatikus tűzjelzés, tűzoltó készülék(ek)
- Riasztó
- Telefon
- Saját főkapcsoló, biztosítékok, vészvilágítás
- Antisztatikus burkolat
- Ablakok védelme

## Szerverszoba kialakítása

- Bőséges hely a szükséges eszközöknek
  - Kiszolgálók
  - Szünetmentes tápegység
  - Hálózati eszközök
- Kamerás megfigyelés
- Kerülendő
  - Munkahely kialakítása
  - Felesleges eszközök tárolása (raktár)
  - A szerverszoba helyének jelzése

## Szoftver biztonság

- Általános szoftveres biztonsági hibák toplistája (FBI / SANS / NIPS):
  - Alapértelmezetten installált operációsrendszerek és programok
  - Felhasználói nevek jelszó nélkül, vagy gyenge jelszavakkal
  - A mentések hiánya, vagy gyengeségei
  - A nyitott portok nagy száma
  - Az érvénytelen ki vagy bemenő című csomagok szűrésének hiánya
  - A naplózás hiánya, vagy gyengeségei
  - Törhető CGI programok

## Szoftver biztonság

- A leggyakoribb biztonsági hibák a UNIX-okon:
  - puffer túlcsordulások az RPC (Remote Procedure Call) szolgáltatásokban
  - Sendmail (levelezőrendszer) hibák
  - BIND (Berkeley Internet Name Daemon) gyengeségek
  - „R" parancsok (rlogin, rsh, rexec, stb.)
  - LPD (Line Printer Daemon)
  - sadmind (system admin daemon) és mountd (mount daemon)
  - alapértelmezett SNMP (Simple Network Management Protocol) sorok

## Felügyelet

- Jólképzett rendszer adminisztrátorok
- Vállalati hierarchia és szabályzatok
- Felhasználók tájékoztatása, oktatása
  - Rendszer használatáról
  - A rájuk leselkedő veszélyekről
  - A jelszavak fontosságáról
  - A hanyagság következményeiről
- Felhasználók szankcionálása

## Adatbiztonság

- Mentés
  - Rendszer mentése
  - Felhasználói állományok mentése
- Automatizált mentések
- Mentések megfelelő tárolása
- Feleslegessé vált archívum megsemmisítése
- Naplózás (journaling) alkalmazása az adatok fájlrendszerén (ResierFS, Ext3)

## Erőforrás-hozzáférési kontroll

- A minimális jogosultság elve
- /etc/security/limits.conf
  - `domain` - ki(k)re vonatkozik az adott szabály
  - `type` - korlátozás típusa (soft, hard)
  - `item` - korlátozandó erőforrás
  - `value` - a korlátozás (item) értéke
- Kvóta (quota)
- Szolgáltatásokhoz való hozzáférés

## Rendelkezésre állás

- Erőforrások mindenkori elérhetősége
- Uptime
- 24 órás szolgáltatások
- Üzemszerű leállások
- Hálózat rendelkezésre állása

- Optimális megoldás?

## Gazdaságos biztonság

- A biztonság plusz költséget jelent
- A gépidő mint költségtényező
  - Ingyenes szoftveres megoldások vs.
  - Hardveres megoldások
- Meddig éri meg növelni a biztonságot?

  → Amíg a rendszerben lévő komponensek (szolgáltatások, adatok, stb.) védelme még megéri.

## Biztonság és …

- Teljesítmény
  - Gyakran csak a teljesítmény rovására növelhető
  - Ezért sem érdemes túlbiztosítani a rendszert
- Kényelem
  - Nehezítheti a normál munkavégzést
  - Csökkentheti annak hatékonyságát
  - Felhasználó keresi a kényelmesebb kerülőket

## Mentés és archiválás

- Mentés
  - Rövid tárolási idő (néhány nap vagy hét)
  - Több példányban és generáció megőrzése
- Archiválás
  - Hosszú tárolási idő
  - Speciális eszközök és szoftverek
  - Több generáció és példány megőrzése
- Mentési és archiválási szabályzat!
- Dokumentálás, naplózás, tárolás.

## Biztonsági minősítések

- **ITSEC** (Information Technology Security Evaluation Criteria)
- **TCSEC** (Trusted Computer System Evaluation Criteria)
- **CCITSE** (Common Criteria for Information Technology Security Evaluation), vagy ismertebb nevén **CC** (Common Criteria)

## Ki vagy mi ellen védekezzünk?

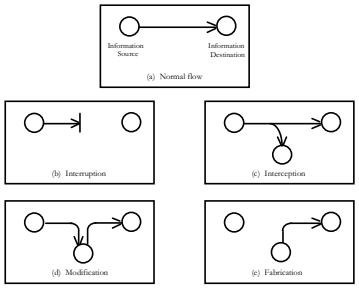
- **Véletlenek, (a)vagy hozzá nem értés**
- **Ártó szándékú programok**
  - **Vírus**
  - **Makró-vírus**
  - **Trójai programok**
  - **Férgek**
  - **Kiskapu**
- **Ártó szándékú emberek**

## Támadások fajtái

- Megszakítás (Interruption)
- Elfogás (Interception)
- Módosítás (Modification)
- Gyártás (Fabrication)

- Passzív: elfogás
- Aktív: megszakítás, módosítás, gyártás

## Támadások fajtái



# Titkosítás, kulcsok

Programtervező informatikus BSc
Számítógépes hálózatok biztonságtechnikája
előadás

## A titkosítás alapfogalmai

- Titkosítás, kriptográfia
- Rejtjelezés (encryption, decryption)
- Titokmegosztás (secret sharing)
- Hitelesítés (certification)
- Partnerazonosítás (identification)
- Hozzáférésvédelem (access control)
- jogosultság vizsgálat (authentication)
- Digitális aláírás (digital signature)

## Titkosítási modell

**Küldő** { Nyílt szöveg (plaintext)
⇩ kulcs (key)
Titkosított szöveg

⇩ nyílt csatorna

Titkosított szöveg
⇩ kulcs (key)
Nyílt szöveg (plaintext) } **Fogadó**

## Klasszikus rejtjelezések

- Caesar-féle rejtjelezés

  plaintext: `aábcdeéfghijklmnoöpqrstuüvwxyz`
  ciphertext: `deéfghijklmnoöpqrstuüvwxyzaábc`
- k-eltolás
- Általános egyábécés rejtjelezés
- Keverő kódok
- Egyszer használatos bitminta, a feltörhetetlen

## Számítógépes titkosítások

- Tendenciák napjainkban és régen
- Két alapvető kriptográfiai elv
  - Redundancia
    - Aktív támadások elleni védelem
    - Passzív támadások elleni védelem
  - Frissesség
    - Ismétléses támadások

## DES - Digital Encryption Standard

- 1977-ben az IBM fejlesztette ki.
- Szimmetrikus kulcsú algoritmus
  - 56 bites kulccsal kódol
- Blokk-kódolás
  - 64 bites blokkokat - 64 bites blokkokká
- 19 különálló lépés
- Ma már nem tekinthető biztonságosnak

## 3DES - Háromszoros DES

- Titkosítás
  - Nyílt szöveg kódolása K1 kulccsal
  - Az előző eredmény dekódolása K2 kulccsal
  - Az előző eredmény kódolása ismét K1 kulccsal
- Visszafejtés
  - Kódolt szöveg dekódolása K1 kulccsal
  - Az előző eredmény kódolása K2 kulccsal
  - Az előző eredmény dekódolása K1 kulccsal
- Két kulcs használata három helyett?
- EDE (kódol - dekódol - kódol) algoritmus vagy EEE?

## Blowfish

- Változó kulcshosszúság
  - maximum 448 bites kulcsokkal képes kódolni
- Blokk-kódolás
  - 64 bites blokkokat titkosít
- Működése
  - Kulcs kiterjesztési fázis (kulcstömb előállítása)
  - Titkosítás a kulcstömb alapján

## Nyilvános kulcsú titkosítás

- Szimmetrikus kulcsok szétosztási problémája
- Nyilvános kulcsú titkosítás
  - Nyilvános kulcs (N)
  - Titkos kulcs (T)
  - Kódolandó információ (x)
- Diffie és Hellmann követelményei:
  - **T(N(x)) = x** (N(T(x)) = x)
  - T előállítása N alapján rendkívül nehézkes legyen
  - T feltörthetetlen legyen választott nyílt szöveggel

## Diffie-Hellmann kulcscsere

- Kommunikációban Alíz és Bob vesz részt
- Alíz választ két nagy prímszámot n-t és g-t
- Mindketten előállítanak egy (n-1)-nél kisebb számot, hasonló nagyságrendben. Legyen Alíz száma x és Bob száma y
- Alíz elküldi ($g^x$ mod n)-t és (n,g)-t
- Bob ($g^y$ mod n)-t küld vissza
- Alíz kiszámolja K=($g^y$ mod n)$^x$ értékét
- Bob kiszámolja K=($g^x$ mod n)$^y$ értékét
- A közös titkos kulcs: K = $g^{xy}$ mod n

- Ezt a módszert használják például az **SSL (Secure Socket Layer**) esetében is

## RSA (Rivest, Shamir, Adleman)

- 28 éve túlél minden támadási kísérletet
- Legalább 1024 bites kulcsot igényel
- Az algoritmus lépései
  - Válasszunk két nagy prímszámot, *p*-t és *q*-t
  - Számoljuk ki az $n=p*q$ és $z=(p-1)*(q-1)$ számokat
  - Válasszunk egy *z*-hez relatív prímet, ez legyen *d*
  - Keressünk olyan *e* számot, melyre $e*d \equiv 1 \bmod z$

## RSA

- A nyílt szöveg bitsorozatát blokkokra osztjuk
  - a kódolandó szegmens (*P*) ahol $0 \le P < n$
- Kiszámítjuk $C = P^e \pmod n$ értéket
- C visszakódolása: $P = C^d \pmod n$ alapján
- Kódoláshoz: *e* és *n* számok (nyilvános kulcs)
- Dekódoláshoz: *d* és *n* számok (titkos kulcs)
- A módszer biztonsága a nagy számok faktorizálásának nehézségeiből adódik

## RSA példa

- $p$=3 és $q$=11
- $n$=$p$*$q$ azaz $n$=33; $z$=($p$-1)*($q$-1) azaz $z$=20
- $d$=7 (7-nek és 20-nak nincs közös osztója)
- $e$*$d$≡1 mod $z$ alapján $e$*7≡1 (mod 20) → $e$=3
- Kódolás: $C = P^3$ (mod 33)
- Dekódolás: $P = C^7$ (mod 33)
- A példában $P$ értékének alacsonynak kell maradni ($P$<33) így maximum egy karakteres blokkok kódolhatók

## RSA példa

| Szimbólum | Számérték | $P^3$ | $P^3$ (mod 33) | $C^7$ | $C^7$ (mod 33) | Szimbólum |
|---|---|---|---|---|---|---|
| S | 19 | 6859 | 28 | 13492928512 | 19 | S |
| U | 21 | 9261 | 21 | 1801088541 | 21 | U |
| Z | 26 | 17576 | 20 | 1280000000 | 26 | Z |
| A | 01 | 1 | 1 | 1 | 01 | A |
| N | 14 | 2744 | 5 | 78125 | 14 | N |
| N | 14 | 2744 | 5 | 78125 | 14 | N |
| E | 05 | 125 | 26 | 8031810176 | 05 | E |

Tannenbaum: Számítógép hálózatok

## Digitális aláírások

- Az aláírt üzenetnek a következő feltételeket kell teljesíteni:
  - A fogadó ellenőrizhesse a feladó valódiságát
  - A küldő később ne tagadhassa le az üzenet tartalmát
  - A fogadó saját maga ne rakhassa össze az üzenetet (ne változtathassa meg a tartalmát)

## Titkosítási algoritmusok összehasonlítása

- A szimmetrikus algoritmusok előnyei
  - gyors
  - viszonylag rövid (56-256 bit)
  - más kriptográfiai feladatokra is alkalmazhatók
  - produkciós kódolók
- A szimmetrikus algoritmusok hátrányai
  - a kulcsnak mindkét oldalon titokban kell maradni
  - nagy hálózatokban nehézkesen alkalmazható
  - kulcscsere szükségessége egy biztonságos csatornán
  - a rövid kulcsokat gyakran kell cserélni

## Titkosítási algoritmusok összehasonlítása

- Az aszimmetrikus algoritmusok előnyei
  - mindenkinek csak a saját titkos kulcsára kell vigyázni
  - nagy létszám esetén sem gond a kulcsok kezelése
  - a kulcsokat csak ritkán kell cserélni
  - a titkosító és megoldó folyamatok felcserélhetők
- Az aszimmetrikus algoritmusok hátrányai
  - általában lassú algoritmusok
  - lényegesen hosszabb kulcsok
  - szükséges egy megbízható harmadik fél
  - ha új matematikai áttörés születik, az sok kulcsot érinthet

## Szteganográfia

- Nem csak az üzenet tartalma hanem annak létezése is titkos
- Az üzenetet más (az üzenet szempontjából lényegtelen) adatok közé keverjük
- Üzenetek képekbe rejtése
- Üzenetek zenékbe rejtése
- Hátránya, hogy sok felesleges információt is át kell vinni

## Jelszavak

- Mikor „jó” egy jelszó?
  - A felhasználó képes megjegyezni
  - Felhasználónként különböző egy adott rendszerben
  - Megfelelő hosszúságú (min. 8 karakter)
  - Vegyesen tartalmaz kis- és nagybetűket, számokat
  - Önmagában ne legyen értelmes szó, dátum, stb.
  - Jelszavak cseréje meghatározott időközönként

## Felhasználók jelszavainak védelme

- Ismételt, folyamatos próbálkozások elleni védelem
  - Néhány hibás próbálkozás után a próbálkozó kizárása egy meghatározott időre
  - Helytelen próbálkozások közti várakozási idő folyamatos növelése
  - A rendszer nem árulja el, hogy a felhasználó név érvénytelen-e vagy a jelszó hibás
- Fontos a felhasználók képzése

# Rendszer szintű biztonság

Programtervező informatikus BSc
Számítógépes hálózatok biztonságtechnikája
előadás

## Állományrendszerek

- Az állományrendszerek és az operációs rendszerek szorosan egybeforrnak
  - Hiába erősek a fájlrendszer lehetőségei biztonság terén, ha azt az operációs rendszer nem használja ki.
  - Hiába használna az operációs rendszer fejlett biztonsági funkciókat, ha erre a fájlrendszer alkalmatlan
- Ideális esetben az állományrendszer pont annyit „tud” mint amit az operációs rendszer, és fordítva.
- Célszerű az operációs rendszert a hozzá javasolt fájlrendszerrel használni

## A FAT fájlrendszer

- Egyszerű, manapság elavult állományrendszer
- Több verziója létezik: FAT12, FAT16, FAT32, VFAT
  - FAT12: 12 bites FAT bejegyzések (max: 4.080 cluster)
  - FAT16: 16 bites FAT bejegyzések (max: 65.520 cluster)
  - FAT32: 28 bites FAT bejegyzések (max: 268.435.456 cluster elméletben, max: 4.177.920 cluster a gyakorlatban)
- A FAT a kötet elején meghatározott helyen található táblázat (minden clusterhez tartozik egy bejegyzés)
- Létezik belőle egy másolat is biztonsági okokból
- Fürtöket (cluster) foglal le, melyek mérete függ a kötet méretétől

## FAT12, FAT16, FAT32

| | FAT12 | FAT16 | FAT32 |
|---|---|---|---|
| Clusterek max. száma | 4080 | 65520 | 4177920<br>268435456 |
| Fájlok max. száma | 4096 | 65536 | 4194304 |
| Maximális kötetméret | 16 MB | 2 GB<br>4 GB (XP) | 32 GB<br>8 TB |
| Maximális fájlméret | 16 MB | 2 GB | 4 GB |

## Állományok a FAT kötetben

- Állomány létrehozása FAT kötetben
  - az adott könyvtárban keletkezik egy bejegyzés, melynek egy része az állomány tartalmazó első fürtre mutat
  - Ha az állomány elfér egy clusterben: az adott fürthöz tartozó FAT bejegyzés jelzi az állomány végét
  - Ha az állomány nem fér el egy clusterben: az adott fürthöz tartozó FAT bejegyzés az állomány következő darabját tartalmazó fürtre mutat,

## Gyökérkatalógus, katalógusbejegyzések

- A gyökérkatalógus a FAT tábla után található
- Mérete adott, a gyökérkönyvtárban található bejegyzések száma korlátozott
- Minden állományhoz egy 32 bájtos katalógusbejegyzés tartozik

| Cím (h) | Hossz | Jelentés | Cím (h) | Hossz | Jelentés |
|---|---|---|---|---|---|
| 0 | 8 | Fájlnév | 16 | 2 | Idő |
| 8 | 3 | Kiterjesztés | 18 | 2 | Dátum |
| 0B | 1 | Attribútumok | 1A | 2 | Kezdő cluster |
| 0C | 10 | Fenntartott | 1C | 4 | Méret |

## FAT névkonvenciók

- Tradicionális 8.3-as névkonvenció a könyvtár és állománynevekben
  - név: maximum nyolc betű, de minimum egy betű
    - névnek vagy számmal, vagy betűvel kell kezdődnie
  - elválasztó: egy „.” (separator)
  - kiterjesztés: maximum három betűs (extension)
- A nevek az ASCII karakterkészlet jeleit tartalmazhatják
- Nem engedélyezett karakterek: ␣ . " / \ [] ; : | = ,
- Foglalt nevek: CON, AUX, COM1, COM2, COM3, COM4, LPT1, LPT2, LPT3, PRN, NUL
- Minden karaktert nagybetűsre (uppercase) konvertál

## Alkönyvtárak

- Alkönyvtár attribútum=1 (fájloknál 0 értékű)
- Minden alkönyvtár első clusterén egy újabb katalógus kezdődik
  - mérete nem állandó (mint a gyökérkatalógusé)
  - tartozik hozzá egy FAT lánc (mint a fájlokhoz)
  - → katalógust tartalmazó fájlok
- Minden alkönyvtárban megtalálhatók a következő bejegyzések
  - . - aktuális könyvtár
  - .. - szülő könyvtár

## A FAT gyengeségei

- A FAT táblát folyamatosan frissíteni kell
  - a fejet mindig a logikai nullás track-re kell pozícionálni a FAT módosításához/olvasásához
  - ha ezt nem tennénk meg minden írási művelet után akkor adatvesztés történhetne
- az állományok mindig az első szabad helyre kerülnek (fragmentation)
- fájl attribútumok:
  - csak olvasási (read only)
  - rejtett (hidden)
  - archiválandó (archive)
  - rendszer (system)

## A FAT előnyei és korlátai

- Előnye az egyszerűség
  - kb. 100 MB-ig hatékonyabban tud működni mint más fájlrendszerek
  - nagyobb kötetek esetén a sebesség drasztikusan csökken
  - nagyobb cluster méret esetén nagyobb veszteség
- Korlátozások
  - partíció elméleti maximális mérete 4 GB
  - a legnagyobb fájl mérete 2 GB
  - gyökérkönyvtárban lévő bejegyzések száma maximum 512

## A HPFS

- High Performance File System (OS/2)
- Manapság már ritkán használatos
- Hosszú fájlnevek támogatása (254 karakter)
- A fájlok két részből állnak
  - adat
  - attribútumok
- Nem használ clustereket, fix foglalási egység a szektor, melynek mérete 512 bájt.

## HPFS könyvtárbejegyzések

- A bejegyzések nem az első clusterre mutatnak, hanem egy FNODE-ra
- Az FNODE tartalmazhatja
  - az állomány adatait
  - vagy más struktúrákat amik az állomány adataira mutatnak
- Könyvtárbejegyzések tartalmazzák
  - létrehozási dátum és idő
  - módosítási dátum és idő
  - hozzáférési dátum és idő

## HPFS band-ek

- Cél egy állomány számára az egymást követő szektorok lefoglalása
- 8 MB-os band-ek szervezése
  - lehetőség szerint egy fájl adatai egy ilyen egységbe kerülnek
  - az egységek között 2 kB-os foglaltsági térkép található, ami az adott band szektorainak foglaltságáról tájékoztat
  - az író/olvasó fejeknek nem minden esetben kell a lemez elejére visszapozícionálni

## Super Block, Spare Block

- Super Block
  - a 16-os logikai szektorban található
  - a gyökérkönyvtár FNODE-jára mutat
  - a gyökérkönyvtár bárhol lehet a partíción
  - egyetlen hibás szektor is végzetes lehet a Super Block-ban
- Spare Block
  - a 16-os logikai szektorban található
  - hot-fix tábla és Spare Directory Block
  - hibás szektorok átirányítása hibátlan szektorokra
  - transzparenssé teszi az I/O műveleteket

## HPFS előnyök és hátrányok

- Támogatottság
  - OS/2
  - MS Windows NT 3.1, 3.5, és 3.51
- A mai partíció méreteken kevésbé hatékony
- Pazarló a néhány 100 MB-os partíciókon
- Kis és nagybetűk megkülönböztetése
- A biztonsági kérdések (csak) OS/2 alatt megoldottak

## Az NTFS

- New Technology File System
- Leginkább a Microsoft Windows NT alapokra épülő operációs rendszerek használják
- Más operációs rendszerek maximum csak olvasni tudják
- Nincsenek speciális objektumok
- Nincs allokációs tábla vagy Super Block
- Változó méretű foglalási alapegységek
- Kis és nagybetűk megkülönböztetése
- Hard link támogatása
- Beépített file-műveletvégzési nyilvántartás, roll back
- Maximális elméleti kötetméret 16 exabájt ($2^{64}$ bájt)
- Maximális kötetméret a gyakorlatban 2 TB

## MFT - Master File Table

- Helye nem kötött, több példányban létezik
- Az MFT és a tükörállományok helye a partíció boot rekordjában kerül meghatározásra
- Támogatja az állományokhoz való hozzáférés szabályozást
- Az MFT is egy állomány, tehát probléma lehet a töredezettség
  - MFT zóna létrehozása (egymást követő clusterek)
  - Amíg van hely a lemezen, adat nem kerül az MFT zónába
  - az MFT zóna is töredezhet ha kicsi a lefoglalt hely például

## Az MFT felépítése

- A MFT rekordokból épül fel (egy rekord ált. 1 kB)
  - fejléc
    - sorszám
    - mutató (pointer) az első attribútumra
    - mutató az első szabad helyre a rekordban
    - az állomány kezdő MFT rekordjának száma
  - attribútumok: a rekordhoz tartozó állomány, könyvtár vagy adat sajátosságait adják meg. pl.: időbélyeg, biztonsági információk, könyvtár tartalom, stb.

## Az MFT attribútumok

- Az attribútumok két részre oszthatók
  - fejléc: ami az attribútum nevét, állapot bitjeit és helyét határozza meg
  - adat rész (pl.: név: FILE_NAME adat: akarmi.exe)
- Beágyazott (resident) attribútumok
  - az attribútum adat az MFT-ben tárolódik
  - az állomány név és a biztonsági információk mindig beágyazottak
- Külső (non resident) attribútumok
  - ha nem fér el minden attribútum adat az MFT-ben
  - az MFT-be csak a fejléc és egy cluster hivatkozás kerül

## Könyvtárbejegyzések

- Speciális, úgynevezett index attribútum ami állománynevek tárolására szolgál
- Tartalmazza az állományok nevét és a hozzá tartozó általános attribútumok másolatát
- Gyors listázás az állományok beolvasása nélkül
- INDEX_ROOT: a könyvtárbejegyzések elhelyezkedését leíró attribútum
- INDEX_ALLOCATION: ha a könyvtárbejegyzések már nem férnek el egyetlen MFT rekordba, akkor megmutatja a kiegészítő információk helyét a partíción

## NTFS rendszerállományok

| Metaadat állomány | MFT rekord | Leírás |
|---|---|---|
| $MFT | 0 | Master File Table |
| $MFTMIRR | 1 | Az MFT első 16 rekordjának másolata |
| $LOGFILE | 2 | Tranzakciós log állomány |
| $VOLUME | 3 | A kötettel kapcsolatos adatokat tartalmaz (kötetnév, időbélyeg, állapot flag) |
| $ATTRDEF | 4 | Attribútum definíciók |
| $ | 5 | A partíció főkönyvtára |
| $BITMAP | 6 | A partíció cluster-térképe (1 foglalt – 0 szabad) |
| $BOOT | 7 | A partíció boot rekordja |
| $BADCLUS | 8 | A partíció hibás cluster-einek listája (a HPFS hotfix táblázatához hasonló) |
| $QUOTA | 9 | A felhasználókhoz rendelt lemez-kvóta adatai (Valójában csak MS Windows 2000, XP esetén használatos NTFS 5.0) |
| $UPCASE | 10 | Kis-nagybetűs karakterkonverzió |
| $EXTEND | 11 | NTFS kiterjesztése |

## NTFS alapjogosultságok

| | ÉRTELME KÖNYVTÁRRA | ÉRTELME FÁJLRA |
|---|---|---|
| ***Read(R)*** | Megtekinthető a könyvtár tulajdonosa, attribútumai és jogosultságai, valamint kilistázható az alatta lévő fájlok neve | Megtekinthető a fájl tulajdonosa, attribútumai és jogosultságai, valamint tartalma |
| ***Write(W)*** | Megtekinthető a könyvtár tulajdonosa, és jogosultságai, megváltoztathatók a könyvtár attribútumai, valamint kilistázható az alatta lévő fájlok neve | Megtekinthető a fájl tulajdonosa, jogosultságai, megváltoztathatók a fájl attribútumai, valamint bővíthető a fájl tartalma. |
| ***Execute(X)*** | Megtekinthető a könyvtár tulajdonosa, attribútumai és jogosultságai, valamint beléphetünk az alkönyvtárakba. | Megtekinthető a fájl tulajdonosa, jogosultságai, attribútumai, valamint a fájl futtatható (ha bináris, vagy interpretált program) |
| ***Delete(D)*** | Törölhető a könyvtár. | Törölhető a fájl. |
| ***Change Permissions(P)*** | Megváltoztathatók a könyvtáron lévő jogok. | Megváltoztathatók a fájlon lévő jogok. |
| ***Take Ownership(O)*** | Saját tulajdonba vehető a könyvtár. | Saját tulajdonba vehető a fájl. |
| ***No Access*** | Minden jogosultság megtagadva. | Minden jogosultság megtagadva. |

## NTFS5

- EFS (Encrypting File System)
  - az állományok titkosíthatók
- Quota
  - felhasználók által használt hely szabályozása
- Sparse files
  - lefoglalt lemezterület üres fájlokkal
- Reparse points
  - Volume mount points
  - Hierarchical Storage Management
  - Single Instance Storage

## EXT2 (Extended File System 2)

- GNU/Linux alapú rendszerek tradicionális állományrendszere
- Az eredeti UNIX fájlrendszerekhez hasonlóan blokkokból, inódokból, és könyvtárakból áll
- Kiforrott és sokoldalú
- Megoldott a jogosultságok kezelése
- Gyengesége a naplózás hiánya, szabálytalan leállás után a teljes fájlrendszert ellenőrizni kell

## Blokkok, blokkcsoportok

- Blokkok
  - A tárterületet blokkokra osztva kezeli az ext2
  - Fix méretű blokkok (1024 bájt, 2048 bájt , 4096 bájt, (8192 bájt - Alpha))
- Blokkcsoportok
  - A blokkok csoportokba rendezettek (fragmentáció csökkentése, fejmozgás minimalizálása)
  - A blokkcsoport maximális mérete 8 blokk
  - A blokkcsoport első két blokkja fenntartott a csoporton belüli blokkok és inódok felhasználtságára vonatkozó információk tárolására

## A szuperblokk

- A partíció kezdetétől számítva 1024 bájtnál található
- Számos másolat létezik róla a blokkcsoportok elején
  - kezdetben minden blokkcsoport elején (pazarló)
  - manapság már csak meghatározott blokkok elején
- A szuperblokk tartalmaz minden információt arról, hogyan töltik meg az adatok a rendszert.
  - tartalmazza az inódok és a blokkok számát
  - az előbbiekből mennyi szabad és foglalt
  - fájlrendszer információk (mikor mountolták, mi a verziója mikor módosították utoljára, milyen op.rsz. hozta létre)

## Az inód (index-node)

- alapvető egység, minden objektumot egy inód reprezentál
  - tartalmaz egy mutatót, arra a blokkra, ami az objektum adatait tárolja
  - tartalmazza az objektum alap adatait:
    - hozzáférési jogok
    - tulajdonos, csoport
    - flagek
    - méret, blokkok száma
    - utolsó elérési idő, változási idő, módosítási idő, törlési idő
    - linkek száma, töredékek
    - verzió (az NFS-hez), kiterjesztett attribútumok

## Könyvtárak, szimbolikus linkek

- Könyvtárak
  - fájlrendszer objektum, épp olyan inódja van mint egy fájlnak
  - speciális formátumú fájl, ami egy listát tartalmaz, név - inódszám párosításokkal
- Szimbolikus linkek (symbolic links)
  - szintén inód reprezentálja
  - ha a link rövidebb mint 60 byte, az inód maga tartalmazza az információt

## Korlátok

| Fájlrendszer blokk méret: | 1024 B | 2048 B | 4096 B | 8192 B |
|---|---|---|---|---|
| Maximális fájlméret: | 16 GB | 256 GB | 2048 GB | |
| Maximális fájlrendszer méret: | 2048 GB | 8192 GB | 16384 GB | 32768 GB |

- Egy könyvtárban 32768 alkönyvtár helyezkedhet el
- 10-15 ezer állomány helyezkedhet el egy könyvtárban, ezután már sebességi problémák lépnek fel
- Fájlnév maximális hossza 255 karakter (1012 kar.)

## EXT3 (Extended File System 3)

- Könnyen frissíthető az ext2 fájlrendszerről
- Visszafelé kompatibilis az ext2-vel
- Tartalmaz egy naplózó (journaling) funkciót
- A metaadatok mellett az adatokat is képes naplózni megfelelő beállítás esetén
- Szabálytalan leállás esetén nem kell a teljes fájlrendszert ellenőrizni, a napló alapján felderíthetők a problémák

## Operációs rendszerek

- Alapvetően meghatározzák egy rendszer biztonságát
- Többfelhasználós (multiuser)
  - csak egy felhasználó használata (rendszergazdai jogokkal)
  - automatikus beléptetés az egyetlen felhasználónak
  - célszerű egy külön, korlátozott jogkörrel rendelkező felhasználói fiókot használni a mindennapi munkához
- Többfeladatos (multitask) rendszerek
- Naplók készítése és nyomonkövetése, karbantartása
- Alapértelmezetten telepített operációs rendszerek
- Rendszerindítás folyamata

## Adatbiztonság - RAID

- **R**edundant **A**rray of **I**nexpensive **D**isks
- **R**edundant **A**rray of **I**ndependent **D**isks

- Hozzáférési idők minimalizálása
- Adatvesztés kockázatának minimalizálása
- Szoftveres vagy hardveres megoldás
- hotplugable disks

## RAID szintek

- RAID 0 - striping (darabolás)
- RAID 1 - mirroring (tükrözés)
- RAID 0+1
- RAID 2
- RAID 3
- RAID 4
- RAID 5
- RAID 6

## Ami ellen a RAID sem véd …

- Áramszünet, tápegység hiba
  - Cache tartalma elveszik
  - Félbemaradt írási műveletek
- Lemez meghibásodása
- Egyidejűleg több lemez meghibásodása
- Vezérlők meghibásodása
- Elemi csapások
- Szoftverhibák

## A fájlrendszerekről bővebben

- http://www.reference.com/browse/wiki/Comparison_of_file_systems
- http://www.ntfs.com

# Külső és belső hálózatok

Programtervező informatikus BSc

Számítógépes hálózatok biztonságtechnikája előadás

## Növekvő biztonsági kihívások

- Régebben a fizikai biztonság volt az elsődleges
- Manapság a hálózatok és az Internet jelentik a legnagyobb biztonsági problémát
  - nyílt, szabad közösség, hatalmas populáció
  - szinte mindenki kötődik hozzá
    - potenciális piac
    - a felhasználókkal, vásárlókkal jönnek a támadók is
    - folyamatosan új vírusok, férgek, kémprogramok készülnek
  - Bárki szabadon rákapcsolódhat (vezeték nélküli hálózatok)
  - Letölthető hacker eszközök, útmutatók

## Támadások típusai – külső támadások

- Lopakodók – fizikai biztonság (gépek zárolása)
- DoS – Denial of Service
  - Nehezen kivédhető
  - Nem feltétlenül okoz kárt
- DDoS – DoS sok "zombi gép" felhasználásával
- Támadások az alkalmazási rétegben
  - Ismert és új biztonsági hibák kihasználás
  - A nem frissített operációs rendszerek hibáinak kihasználása
- A hálózat felderítése, figyelése
  - Védtelen vezeték nélküli hálózatok keresése
  - A hálózat, kiszolgálók és kliensek szkennelése
  - Csomagok figyelése, jelszavak, információk keresése

## Támadások típusai – belső támadások

- Fertőzött laptop – egyre gyakoribb probléma
- Nem engedélyezett eszközök
  - Alapértelmezéssel használt hálózati eszközök
    - switch
    - router
    - vezeték nélküli AP
- Nem engedélyezett szolgáltatások
  - Saját fájl és nyomtató megosztások (jelszó nélkül)
- Elbocsátott alkalmazott – Man in the middle
- Vírusok, trójai programok

## A várható támadások típusai

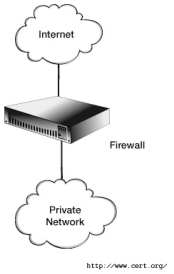
- Web szolgáltatások elleni támadások
- Komplex Web támadás
  - Apache biztonsági rés
  - IE biztonsági rés
- Spyware fenyegetés
- Mobil eszköz elleni támadások
  - Notebook
  - PDA, Telefon
- SPAM
- DoS, DDoS

## Lehetséges védekezési módok

- A megfelelően konfigurált és működő eszközök elégségesek lennének …
- De szükséges a védelem a következők miatt
  - a szoftverek hibái miatt
  - az emberi mulasztás, butaság vagy hozzá nem értés
- Lehetséges megoldások
  - Elosztott, jól koordinálható, több rétegű, független védelem
  - Integrált megoldás (szerverek, routerek, switchek)
  - Önmagát védő hálózatok
  - A hálózat felosztása zónákra
    - a zónák saját szabályrendszer szerint működnek
    - a zónák határán szükséges egy eszköz ami feloldja a konfliktusokat
    - általában tűzfalak látják el ezeket a funkciókat

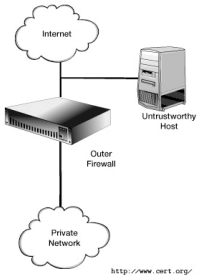
## Egyszerű (határ) tűzfal

- Egyrétegű megoldás
- Egy eszközre van telepítve minden tűzfal funkció
- Elválasztja egymástól a nyilvános és a védett hálózatot
- Egyszerű és olcsó
- A legkevésbé biztonságos



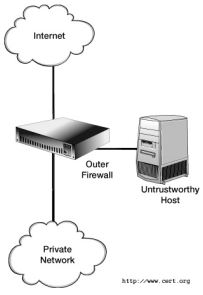

## A megbízhatatlan gép problémája

- A külvilág felé szolgáltatnak
  - WWW, POP3, SMTP, SSH
- A legveszélyeztetettebb elem
- Minimalizálni kell a nyújtott szolgáltatásokat
- A belső hálózat gépei nem tekinthetik megbízhatónak



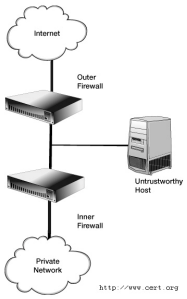

## A DMZ kialakítása

- Feladat a megbízhatatlan kiszolgálók védelme
- Külön hálózatot alakítunk ki ezeknek a gépeknek, szolgáltatásoknak
- Ezáltal növekszik
  - a biztonság
  - a megbízhatóság
  - a rendelkezésre állás



## Kettős tűzfal

- Célja
  - A belső hálózat és a DMZ védelme
  - A belső hálózat elkülönítése a DMZ-től
- Funkciók
  - Külső tűzfal
  - Belső tűzfal
- Védett hálózatok:
  - DMZ
  - Belső hálózat
- Lehetőség szerint eltérő architektúrájú tűzfalak használata javasolt




## Tűzfalak csoportosítása

- Tűzfalak osztályai:
  - Személyes tűzfalak (első osztály)
  - Forgalomirányítók tűzfalai (második osztály)
  - Alsó kategóriás hardver tűzfalak (harmadik osztály)
  - Felső kategóriás hardver tűzfalak (negyedik osztály)
  - Felső kategóriás szerver tűzfalak (ötödik osztály)
- Tűzfalak típusai
  - Csomagszűrő
  - Cím transzformáló
  - Állapottartó
  - Kapcsolat szintű átjáró
  - Proxy
  - Alkalmazásszűrés

## Első osztály - Személyes tűzfalak

- Egyre több otthoni kapcsolat
- Egyre több mobil számítógép, idegen környezet
- Általában PC-n futó szoftveres szolgáltatás
- Kis hálózat védelmére is alkalmas (otthoni hálózat)
- Manapság minden gépen erősen ajánlott a használata
- Minimális tudású megoldások, általában csak csomagszűrést végeznek

## Első osztály - Személyes tűzfalak

| | |
|---|---|
| Alapszolgáltatások | statikus csomagszűrés, NAT |
| Konfigurálás | Automatikus (manuális lehetséges) |
| IP címek engedélyezése vagy blokkolása | Igen |
| Portok vagy protokollok engedélyezése vagy blokkolása | Igen |
| ICMP üzenetek engedélyezése vagy blokkolása | Igen |
| Kimenő folyamatok szabályozása | Igen |
| Alkalmazások védelme | Esetleg |
| Látható vagy hallható riasztások | Esetleg |
| Napló állományok | Esetleg |
| Riasztások valós időben | Esetleg |
| VPN támogatás | Általában nem |
| Távoli felügyelet | Általában nem |
| Gyártói támogatás | Változó |
| Konkurens folyamatok száma | 1-10 |
| Moduláris bővíthetőség (hardver vagy szoftver) | Változó |
| Ár | Alacsony |

## Első osztály - Személyes tűzfalak

- Előnyök
  - Alacsony költség
  - Egyszerű konfigurálás
- Hátrányok
  - Központilag nehezen menedzselhető
    - Minden kliens külön konfigurálást igényel
  - Csak alapvető tűzfal funkciókat lát el
  - Korlátozott teljesítmény
    - Egy gép védelmére vannak tervezve

## Második osztály - Forgalomirányítók tűzfalai

- A routerekbe gyakran integrálnak tűzfal funkciókat is
- Az alsó kategóriás (low-end) routerek
  - forgalomszűrés IP cím alapján
  - forgalomszűrés port alapján
  - NAT szolgáltatás a címek elrejtésére
- A felső kategóriás (high-end) eszközök
  - programozhatóak
  - állapotkövetők
  - támogatják a magas rendelkezésre állást

## Második osztály - Forgalomirányítók tűzfalai

| | |
|---|---|
| Alapszolgáltatások | statikus csomagszűrés, NAT, (alkalmazás szűrés) |
| Konfigurálás | Automatikus (alsó kat.), manuális (felső kat.) |
| IP címek engedélyezése vagy blokkolása | Igen |
| Portok vagy protokollok engedélyezése vagy blokkolása | Igen |
| ICMP üzenetek engedélyezése vagy blokkolása | Igen |
| Kimenő folyamatok szabályozása | Igen |
| Alkalmazások védelme | Esetleg |
| Látható vagy hallható riasztások | Általában igen |
| Napló állományok | Legtöbbször igen |
| Riasztások valós időben | Legtöbbször igen |
| VPN támogatás | Közös (alsó kategória), különálló (felső kategória) |
| Távoli felügyelet | Igen |
| Gyártói támogatás | Korlátozott (alsó kategória), jó (felső kategória) |
| Konkurens folyamatok száma | 10-1000 |
| Moduláris bővíthetőség (hardver vagy szoftver) | Nincs (alsó kategória), korlátozott (felső kategória) |
| Ár | Változó |

## Második osztály - Forgalomirányítók tűzfalai

- Előnyök
  - Alacsony költség (minimális többletköltség)
  - Összevont konfiguráció, kevesebb hibalehetőség
  - Befektetések védelme (nem szükséges újrakábelezni, újabb képzéseket tartani)
- Hátrányok
  - Korlátozott funkciók
  - Csak alapvető tűzfal funkciókat lát el
  - Csökkenő forgalomirányítási teljesítmény
  - Csökkenő teljesítmény naplózáskor (támadás alatt)

## Harmadik és negyedik osztály - hardver tűzfalak

- Low-end hardver tűzfalak
  - Általában plug & play eszközök
  - Minimális konfigurálási igény
  - Integrálhat switch vagy VPN funkciókat is
  - Kis és középvállalatok számára lehet megoldás
- High-end hardver tűzfalak
  - Talán a lehető legjobb megoldás a hálózati teljesítmény csökkenése nélkül
  - Nagyvállalatok, központok védelmére

## Harmadik osztály - Alsó kategóriás hardver tűzfalak

| | |
|---|---|
| Alapszolgáltatások | statikus csomagszűrés, NAT, (alkalmazás szűrés) |
| Konfigurálás | Automatikus (manuális lehetséges) |
| IP címek engedélyezése vagy blokkolása | Igen |
| Portok vagy protokollok engedélyezése vagy blokkolása | Igen |
| ICMP üzenetek engedélyezése vagy blokkolása | Igen |
| Kimenő folyamatok szabályozása | Igen |
| Alkalmazások védelme | Általában nem |
| Látható vagy hallható riasztások | Általában igen |
| Napló állományok | Általában nem |
| Riasztások valós időben | Általában nem |
| VPN támogatás | Néha |
| Távoli felügyelet | Igen |
| Gyártói támogatás | Korlátozott |
| Konkurens folyamatok száma | 10-7.500 |
| Moduláris bővíthetőség (hardver vagy szoftver) | Korlátozott |
| Ár | Alacsony |

## Harmadik osztály - Alsó kategóriás hardver tűzfalak

- Előnyök
  - Alacsony költség (minimális többletköltség)
  - Egyszerű konfigurálás
- Hátrányok
  - Korlátozott funkciók
  - Csak alapvető tűzfal funkciókat lát el
  - Alacsony teljesítmény
  - Korlátozott gyártói támogatás (WEB, e-mail)
  - Korlátozott bővíthetőség (firmware upgrade)

## Negyedik osztály - Felső kategóriás hardver tűzfalak

| | |
|---|---|
| Alapszolgáltatások | statikus csomagszűrés, NAT, alkalmazásszűrés |
| Konfigurálás | Általában manuális |
| IP címek engedélyezése vagy blokkolása | Igen |
| Portok vagy protokollok engedélyezése vagy blokkolása | Igen |
| ICMP üzenetek engedélyezése vagy blokkolása | Igen |
| Kimenő folyamatok szabályozása | Igen |
| Alkalmazások védelme | Lehetőség szerint |
| Látható vagy hallható riasztások | Igen |
| Napló állományok | Igen |
| Riasztások valós időben | Igen |
| VPN támogatás | Lehetőség szerint |
| Távoli felügyelet | Igen |
| Gyártói támogatás | Jó |
| Konkurens folyamatok száma | 7.500-500.000 |
| Moduláris bővíthetőség (hardver vagy szoftver) | Igen |
| Ár | Magas |

## Negyedik osztály - Felső kategóriás hardver tűzfalak

- Előnyök
  - Magas teljesítmény
  - Jó használhatóság
    - összekapcsolható eszközök, terhelés elosztás
  - Moduláris felépítés (HW és SW bővítési lehetőségek)
  - Kifinomult távoli menedzsment
  - Rugalmasság, skálázhatóság
  - Alkalmazásszűrés (L4, L5, L6, L7)
- Hátrányok
  - Magas ár
  - Bonyolult konfiguráció, menedzsment

## Ötödik osztály - Felső kategóriás szerver tűzfalak

| | |
|---|---|
| Alapszolgáltatások | statikus csomagszűrés, NAT, alkalmazásszűrés |
| Konfigurálás | Általában manuális |
| IP címek engedélyezése vagy blokkolása | Igen |
| Portok vagy protokollok engedélyezése vagy blokkolása | Igen |
| ICMP üzenetek engedélyezése vagy blokkolása | Igen |
| Kimenő folyamatok szabályozása | Igen |
| Alkalmazások védelme | Lehetőség szerint |
| Látható vagy hallható riasztások | Igen |
| Napló állományok | Igen |
| Riasztások valós időben | Igen |
| VPN támogatás | Lehetőség szerint |
| Távoli felügyelet | Igen |
| Gyártói támogatás | Jó |
| Konkurens folyamatok száma | >50.000 |
| Moduláris bővíthetőség (hardver vagy szoftver) | Igen |
| Ár | Magas |

## Ötödik osztály - Felső kategóriás szerver tűzfalak

- Előnyök
  - Jól ismert környezet (Linux, FreeBSD, Windows)
  - Magas teljesítmény (megfelelően méretezett szerveren)
  - Nagyfokú integráltság az operációs rendszer szolgáltatásaival
  - Használhatóság, rugalmasság, skálázhatóság
- Hátrányok
  - Felső kategóriás hardver szükséges – magas ár
  - Sebezhető (egy ismert operációs rendszeren fut)

## Csomagszűrők

- Az egyes csomagok eldobása vagy továbbítása
- Szűrési feltételek:
  - Forrás / cél cím
  - Forrás / cél port
- Nem értik és nem vizsgálják az alkalmazásokról szóló információt (csak az IP fejlécet)
- Mivel a különböző hálózatokat leggyakrabban forgalomirányítók kötik össze ezért ezen funkció is leggyakrabban itt található
- Gyors és kis erőforrás igényű megoldás
- Önmagában általában nem elégséges megoldás

## Állapotkövető csomagszűrés

- TCP kapcsolatok nyomon követése
- A kimenő csomagok naplózása az állapot táblában
  - Forrás / cél cím
  - Forrás / cél port
- Bejövő (és kimenő) forgalomnál ellenőrizhető, hogy ki kezdeményezte
- Eldönthető, hogy a csomag része-e egy már létező kapcsolatnak
- Véd a portletapogatással szemben

## Socks Proxyk

- Kommunikáció három vagy több fél között
  - Kliens -> Proxy -> Szerver
- Titkosítatlan esetben a kliens nem látja közvetlenül azokat a csomagokat amelyeket a szerver küldött és fordítva
- Titkosított esetben a proxy ellenőrzi a fejléceket és ha mindent rendben talál akkor továbbítja a csomagot
- Gyorstár funkció
- Köztes megoldás, nem is csomagszűrés de még nem is alkalmazás szűrés
- Nem minden esetben biztosít transzparens átvitelt

## Alkalmazásrétegbeli proxyk

- Alkalmazásszintű intelligencia a közvetített szolgáltatásoknál
- Minden kapcsolatkérést megvizsgálnak
- Értelmezni tudják az adott alkalmazás adatait és ez alapján döntéseket hoznak
- HTTP, FTP, SMTP, DNS kérések, SPAM szűrés
- Pl.: FTP esetén kívülről lehetséges a USER, PASS, DIR, PORT és GET parancsok használat, de tiltott a PUT
- Protokoll validáció
- Bonyolult megoldás, minden protokollt ismernie kell
- Az alkalmazásokra nézve teljesen transzparensek

# Tűzfalak iptables és ipchains

Progamtervező informatikus BSc

Számítógépes hálózatok biztonságtechnikája előadás

## Tűzfalak különböző gépeken

- Egygépes rendszer
  - védi a kiszolgálót a a külvilággal szemben
- Többkapcsolatos kiszolgáló
  - több hálózati interfész
  - több egymástól független hálózat
- Útválasztó (router)
  - több hálózati interfész
  - IP továbbítás
  - keresztülhaladó forgalom az egyes hálózatok felé

## iptables vs. ipchains

- A linux kernelbe épített megoldás
- ipchains: nem állapotkövető csomagszűrés
- iptables: állapotkövető csomagszűrés
  - hatékonyabb
  - bonyolultabb szabályok egyszerűbben felírhatók
- Kernel támogatás
  - ipchains – 2.2-es és annál újabb kernelekben
  - iptables – 2.4-es és annál újabb kernelekben

## Láncok (chains)

- INPUT (bejövő forgalom)
- OUTPUT (kimenő forgalom)
- FORWARD (továbbított forgalom)

- Saját láncok

## Műveletek (targets)

- DROP (csomag eldobása)
- ACCEPT (csomag elfogadása)
- RETURN (csomag vizsgálatának visszaadása a hívó szűrőláncnak)
- QUEUE (csomag továbbküldése egy felhasználói program felé)

- REJECT (csomag eldobása, válasszal)

## Alapvető parancsok 1.

- Létező szabályok listázása:

```
# iptables -L [chain]
```

- Egy lánc összes szabályának törlése:

```
# iptables -F [chain]
```

- Létező szabályok törlése:

```
# iptables -D chain rule
# iptables -D chain rulenum
```

## Alapvető parancsok 2.

- Saját lánc létrehozása
  `# iptables -N chain`
- Saját lánc törlés
  `# iptables --delete-chain chain`
- Alapértelmezés (policy) beállítása
  `# iptables -P chain target`
  *Például:*
  `# iptables -P OUTPUT ACCEPT`

## Alapvető parancsok 3.

- Szabály hozzáfűzése
  `# iptables -A chain rule`
- Szabály beszúrása
  `# iptables -I chain [rulenum] rule`
- Szabály megváltoztatása
  `# iptables -R chain rulenum rule`

## Paraméterek 1.

- `-p, --protocol [!] protocol`
  protokoll megadása (tcp, udp, icmp, all, numerikus érték)
- `-s, --source [!] address[/mask]`
  forráscím megadása (hálózati név, IP cím, IP cím és netmask)
- `-d, --destination [!] address[/mask]`
  célcím megadása, source megadásához hasonlóan

## Paraméterek 2.

- `-j, --jump target`

  A szabályra illeszkedő csomagokkal kapcsolatos művelet adható meg.
- `-m, --match match`

  Kiterjesztett szűrést tesz lehetővé további feltételek alapján.
- `-i, --in-interface`

  A bejövő csomagok interfésze

## Szabályok sorrendje

- Hozzáfűzéssel

```
# iptables -A INPUT -s 192.168.1.0/24 -j DROP
# iptables -A INPUT -s 192.168.1.100 -j ACCEPT
DROP    all  --  192.168.1.0/24  anywhere
ACCEPT  all  --  192.168.1.100   anywhere
```

- Beszúrással

```
# iptables -I INPUT -s 192.168.1.0/24 -j DROP
# iptables -I INPUT -s 192.168.1.100 -j ACCEPT
ACCEPT  all  --  192.168.1.100   anywhere
DROP    all  --  192.168.1.0/24  anywhere
```

## Teljes hálózati forgalom tiltása

- Szeretnénk tűzfal segítségével elérni a teljes hálózati forgalom tiltását.

```
# iptables -F
# iptables -A INPUT -j REJECT
# iptables -A OUTPUT -j REJECT
# iptables -A FORWARD -j REJECT
```

## A beérkező forgalom tiltása

- A teljes beérkező forgalom tiltását szeretnénk megvalósítani úgy, hogy ez a saját rendszerünkből származó forgalmat ne érintse.

```
# iptables -F INPUT
# iptables -A INPUT -m state --state
  ESTABLISHED -j ACCEPT
# iptables -A INPUT -j REJECT
```

## A kimenő forgalom tiltása

- A teljes kimenő forgalom tiltását szeretnénk elérni úgy, hogy ez lehetőség szerint a bejövő forgalmat ne érintse.

```
# iptables -F OUTPUT
# iptables -A OUTPUT -m state --state
  ESTABLISHED -j ACCEPT
# iptables -A OUTPUT -j REJECT
```

## Hamis címek blokkolása

- Szeretnénk megakadályozni, hogy távoli gépek egy helyi gépének adják ki magukat.
  - Egy gép esetén

```
# iptables -A INPUT -i külső_interfész
  -s saját_IP -j DROP
```

  - Belső hálózat (pl.: 192.168.1.*) esetén

```
# iptables -A INPUT -i külső_interfész
  -s 192.168.1.0/24 -j DROP
```

## Beérkező szolgáltatás kérések tiltása

- Szeretnénk letiltani egy adott hálózati szolgáltatás (pl. HTTP) elérését.

```
# iptables -A INPUT -p tcp --dport 80
  -j REJECT
```

- Beérkező HTTP forgalom tiltása a helyi HTTP forgalom engedélyezése mellett.

```
# iptables -A INPUT -p tcp -i lo
  --dport http -j ACCEPTED
# iptables -A INPUT -p tcp --dport 80
  -j REJECT
```

## Távoli gép hozzáférésének tiltása

- Minden beérkező forgalom blokkolása az adott számítógépről

```
# iptables -A INPUT -s távoli_IP -j
  REJECT
```

- Csak bizonyos szolgáltatások blokkolása

```
# iptables -A INPUT -p tcp -s távoli_IP
  --dport www -j REJECT
```

## Távoli gép hozzáférésének tiltása

- Néhány gép hozzáférésének engedélyezése, a többi hozzáférésének tiltása mellett

```
# iptables -A INPUT -s távoli_IP_1 [-p
  protokoll --dport szolgáltatás] -j
  ACCEPT
# iptables -A INPUT -s távoli_IP_2 [-p
  protokoll --dport szolgáltatás] -j
  ACCEPT
# iptables -A INPUT [-p protokoll
  --dport szolgáltatás] -j REJECT
```

## Távoli számítógéphez történő hozzáférés tiltása

- Minden szolgáltatás tiltása (teljes tiltás)

```
# iptables –A OUTPUT –d távoli_IP -j
  REJECT
```

- Csak bizonyos szolgáltatások tiltása

```
# iptables –A OUTPUT -p tcp –d
  távoli_IP --dport 80 -j REJECT
```

- Az előbbi tiltás egy teljes alhálózaton

```
# iptables –A OUTPUT -p tcp –d
  192.168.1.0/24 --dport 80 -j REJECT
```

## Kizárólag SSH hozzáférés engedélyezése

- A külső zónából csak SSH kapcsolatok legyenek felépíthetők, de a belső hálózatból minden szolgáltatás maradjon elérhető.

```
# iptables –F INPUT
# iptables -A INPUT -p tcp --dport ssh
  -j ACCEPT
# iptables -A INPUT -i lo -j ACCEPT
# iptables -A INPUT -j REJECT
```

## Hozzáférés szabályozása MAC address szerint

- Saját gépünkről származó helyi kapcsolatok mellett egyetlen MAC cím alapján azonosított számítógép hozzáférését engedélyezzük

```
# iptables –F INPUT
# iptables -A INPUT -i lo -j ACCEPT
# iptables -A INPUT -m mac --mac-source
  00:11:22:33:44:55 -j ACCEPT
# iptables -A INPUT -j REJECT
```

## Kimenő Telnet kapcsolatok tiltása

- A nem biztonságos távoli bejelentkezések kezdeményezésének megakadályozása

```
# iptables –A OUTPUT -p tcp --dport
  telnet -j REJECT
```

- A Telnet lokális engedélyezésével

```
# iptables –A OUTPUT -p tcp -i lo
  --dport telnet -j ACCEPT
# iptables –A OUTPUT -p tcp --dport
  telnet -j REJECT
```

## Dedikált szerverek védelme

- A kiszolgálónk szolgáltatásai közül a helyi felhasználók mindent elérhetnek, a távoli kérések azonban csak bizonyos szolgáltatásokra engedélyezettek. Az egyéb kérések naplózásra majd elutasításra kerülnek

```
# iptables –F INPUT
# iptables -A INPUT -i lo -j ACCEPT
# iptables -A INPUT -m multiport -p tcp
  --dport smtp, www, ssh -j ACCEPT
# iptables -A INPUT -j LOG
# iptables -A INPUT -j REJECT
```

## A ping meghiúsítása

- Szeretnénk elkerülni, hogy távoli helyek választ kapjanak ha pingelnek minket.

```
# iptables -A INPUT -p icmp --icmp-type
  echo-request -j DROP
```

- Fontos, hogy ne tiltsunk minden ICMP típust! Az ICMP üzenetek listája:

```
# iptables -p icmp -h
```

## Összetett szabályrendszerek

- Az áttekinthetőség érdekében érdemes modulárisan felépíteni a tűzfalunkat saját láncok létrehozásával és összekapcsolásával

```
# iptables -N SAJAT1
# iptables -N SAJAT2

# iptables -A INPUT … -j SAJAT1
# iptables -A SAJAT1 … -j SAJAT2
```

## Tűzfal konfigurációk mentése és betöltése

- Létrehozott tűzfalszabályok mentése

```
# iptables-save >
  /etc/sysconfig/iptables
```

- Mentett szabályok betöltése

```
# iptables-restore <
  /etc/sysconfig/iptables
```

## Naplózás

- Több helyen szükség lehet az eldobott csomagok naplózására. Ezt érdemes egy új szabálylánccal megoldani

```
# iptables -N LOGDROP
# iptables -A LOGDROP -j LOG
  --log-level warning --log-prefix
  "eldobva" -m limit
# iptables -A LOGDROP -j DROP
```

## Kiszolgálónevek használata

- IP címek helyett kiszolgálónevek is megadhatók a tűzfalszabályokban.
- Ha a DNS több IP címet is megad az adott kiszolgálóhoz, akkor minden IP-hez külön szabály készül.

```
# iptables -N SAJAT
# iptables -A SAJAT -d www.aol.com -J
  REJECT
# iptables -n -L SAJAT
```

## Ajánlott irodalom

- Daniel J. Barrett, Richard E. Silverman, Robert G. Byrnes: Linux biztonsági eljárások. Budapest, Kossuth, 2004. p.42-69.

## Hálózati hozzáférés felügyelete Jogosultságok szabályozása

Progamtervező informatikus BSc

Számítógépes hálózatok biztonságtechnikája

előadás

## A felügyelet szintjei

- Hálózati interfész
- Tűzfal
- Szuperdémon
- Önálló hálózati szolgáltatások

## Hálózati interfészek kezelése

- Hálózati interfészek listázása

```
# ifconfig -a
```

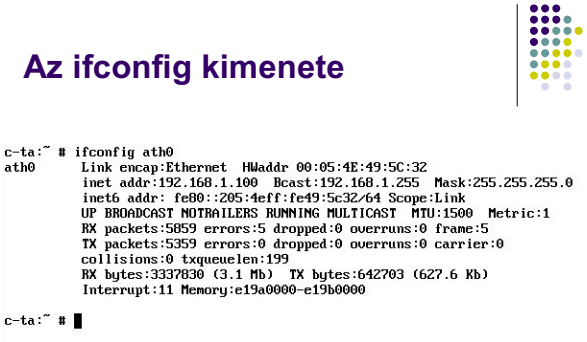
- Bekapcsolt hálózati interfészek listázása

```
# ifconfig
```

- Egyetlen interfész listázása (pl. eth0)

```
# ifconfig eth0
```

## Az ifconfig kimenete



```
c-ta:~ # ifconfig ath0
ath0      Link encap:Ethernet  HWaddr 00:05:4E:49:5C:32
          inet addr:192.168.1.100  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::205:4eff:fe49:5c32/64 Scope:Link
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5859 errors:5 dropped:0 overruns:0 frame:5
          TX packets:5359 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:199
          RX bytes:3337830 (3.1 Mb)  TX bytes:642703 (627.6 Kb)
          Interrupt:11 Memory:e19a0000-e19b0000

c-ta:~ #
```

## Hálózati interfészek indítása és leállítása

- Az eth0 interfész indítása
  `# ifconfig eth0 up`
- Az eth0 interfész leállítása
  `# ifconfig eth0 down`
- A teljes hálózat leállítása és indítása
  `# /etc/init.d/network stop`
  `# /etc/init.d/network start`
  `# service network stop | start`

## Hálózati interfészek indításának és leállításának szintjei

- Legalacsonyabb szint
  `# /sbin/ifconfig`
- Középső szint
  `# /sbin/ifup és /sbin/ifdown`
- Legmagasabb szint
  `# /etc/init.d/network`

## xinetd szolgáltatások ki- és bekapcsolása

- Konfigurációs állományok:
  ```
  # /etc/xinetd.conf
  # /etc/xinetd.d/
  ```
- Például tftp szolgáltatás tiltása
  `# /etc/xinetd.d/tftp` állományba
  `disable = yes` beszúrása
- Szolgáltatás engedélyezése:
  `disable = yes` törlése, kommentezése

## xinetd változások beolvasása

- A változások követése nem történik meg automatikusan, utasítani kell a xinetd-t, hogy olvassa újra a konfigurációs állományokat

```
# kill -USR2 `pidof xinetd`
```

## xinetd szolgáltatás hozzáadása

- Új konfigurációs állomány készítése

```
service SZOLGÁLTATÁS_NÉV
{
server = /UTVONAL/SERVER
server_args = A_SERVER_ARGUMENTUMAI
user = FELHASZNÁLÓ
socket_type = SZOLGÁLTATÁS_TÍPUSA
wait = YES |NO
}
```

## xinetd szolgáltatás példa

```
service ftp
  {
          socket_type         = stream
          protocol            = tcp
          wait                = no
          user                = root
          server              = /usr/sbin/vsftpd
  #        server_args        =
  #        log_on_success     += DURATION USERID
  #        log_on_failure     += USERID
  disable                     = no
  }
```

## A hozzáférés korlátozása távoli felhasználók szerint (xinetd)

- engedélyezés

```
service ftp
{
 only_from = 192.168.1.100
 …
}
```

- tiltás

```
service ftp
{
 no_access = ttk.pte.hu
 no access = 192.169.
 …
}
```

## A hozzáférés korlátozása távoli felhasználók szerint (xinetd)

- Hibalehetőség a konfigurációban

```
service ftp
{
 only_from = 192.160.1.1
 only_from = 192.168.1.    pte.hu
 no_access = 192.168.      ttk.pte.hu
 no_access = 192.160.1.1
 …
}
```

## A hozzáférés korlátozása az aktuális idő függvényében (xinetd)

- Az access_times attribútum

```
service ftp
{
 access_time = 18:00-7:00
 …
}
```

## Hozzáférés korlátozása SSH szerverhez kiszolgáló szerint

- TCP-burkolók (TCP wrappers) támogatása az sshd esetén csak opcionális. Ellenőrzés:

```
# strings /usr/sbin/sshd | egrep
  'hosts\.(allow|deny)'
```

- Ha a kimenet a következő, akkor támogatott

```
# /etc/hosts.allow
# /etc/hosts.deny
```

- Ha a kimenet üres, a TCP-burkolók támogatása nem elérhető

## /etc/hosts.allow /etc/hosts.deny

- SSH korlátozása egy számítógépre

```
sshd: 192.168.1.100
sshd: ALL: DENY
```

- Monitorozási funkció

```
lpd : foo.bar.com : spawn
/bin/echo "%h printer access" |
mail -s "tcp_wrappers on %H" root
```

## Hozzáférés korlátozása SSH szerverhez felhasználói név szerint

- Csak bizonyos felhasználóknak engedélyezzük SSH kapcsolatok fogadását
- Használjuk az `/etc/sshd/sshd_config` állományt
- Adjuk meg az SSH-t használható felhasználók listáját

```
AllowUsers user1 user2 … userN
```

## A root jogosultság

- Rendszergazdai jogosultságok
  - a 0-s UID-vel rendelkező felhasználó (ált. root)
  - ki rendelkezzen root jogosultsággal?
  - milyen módon osztjuk ki ezeket a jogokat?
- A root jelszó megosztása
  - több gép, több rendszeradminisztrátor
  - egyetlen jelszó
- Több root azonosító létrehozása
  - több felhasználó UID 0 és GID 0 kóddal (/etc/passwd)
  - több rendszeradminisztrátor különböző jelszavakkal

## X programok futtatása rootként

- Az X-et normál felhasználóként futtatva nem tudunk a root nevében grafikus programokat futtatni
  - `** WARNING ** cannot open display`
  - a root .Xauthority állománya nem rendelkezik megfelelő jogokkal a grafikus felület eléréséhez
- Megoldás: egy olyan script, ami a szükséges DISPLAY és XAUTHORITY környezeti változókat a hívó felhasználó értékeire állítja:
  - `su - -c "exec env DISPLAY='$DISPLAY' XAUTHORITY='${XAUTHORITY-$HOME/.Xauthority}' "'"$SHELL"'" -c '$*'"`

## sudo

- Teljes rendszerre érvényes konfigurációs állomány
  - `/etc/sudoers`
- Példa az `/etc/sudoers` bejegyzéseire
  - `bob host = (root) /usr/bin/command_1`
- Példa a `sudo` használatára
  - `bob$ sudo -u root / usr/bin/command_1`
- A root jelszó kiadása nélkül
  - jogosultságok gyors és egyszerű kiosztása, visszavonása
  - naplózás támogatása
  - root parancsok saját (felhasználói) jelszóval

## Hitelesítés a sudo használatakor

```
/etc/sudoers
Defaults timestamp_timeout = X
Defaults:bob timestamp_timeout = X
```

- X értéke határozza meg, hogy egy hitelesítés után meddig nem kell újra megadni a jelszót
- Ha X értéke 0, minden sudo parancshoz jelszó kell
- Felhasználói szabályozáskor
  - nem használható a `timestamp` opció, helyette `sudo -k`
  - `sudo` nevű script létrehozása (a PATH-ban a sudo előtt)

```
/usr/bin/sudo $@
/usr/bin/sudo -k
```

## A jelszavas hitelesítés átlépése a sudo-ban

- Hasznos lehet kötegelt feladatok végzésénél
- Vagy ha nem ember végzi a feladatot
- Biztonsági kockázatot az elhagyott terminál
- Például (`/etc/sudoers`):

```
bob ALL = (alice) NOPASSWD /usr/bin/com_1
```

## Kiszolgálónkénti azonosítás

- Csak bizonyos gépeken szeretnénk egy felhasználónak privilégiumokat adni
  - gépek listájának definiálása (`/etc/sudoers`)
    - `Host_alias SAFE_HOSTS = host1, host5`
  - engedélyezzük, hogy bob alice nevében futtasson egy programot a fenti gépeken
    - `bob SAFE_HOSTS = (alice) /usr/bin/com_1`
  - engedélyezzük, hogy bob alice nevében futtasson minden programot a fenti gépeken
    - `bob SAFE_HOSTS = (alice) ALL`

## Csoportok létrehozása

- Egy csoport számára szeretnénk engedélyezni a más felhasználó nevében történő program futtatásokat
- Használjuk a `/etc/group` csoportjait

```
sudogroup:x:1200:bob,alice,endre
```

- A /etc/sudoers állományban pedig a %csoportnév szintaxis használható

```
%sudogroup ALL = (root) /usr/bin/com_1
%sudogroup ALL = (ALL) ALL
```

## Jogok megadása könyvtárakra

- Egy adott könyvtárban bármely programot futtathassa a felhasználó más nevében
- Az adott könyvtárat perjellel lezárva jelöljük ki a `/etc/sudoers` állományban

```
bob ALL = (root) /usr/bin/
```

- Mire vonatkozik a felhatalmazás?

```
bob$ sudo -u root /usr/local/bin/com_1
bob$ sudo -u root /usr/bin/com_2
bob$ sudo -u root /usr/bin/gnu/com_3
```

## Argumentumok tiltása a sudoban

- Parancs futtatásának engedélyezése argumentumok megadásának lehetőségével
  - ha egy parancs után nem adunk meg argumentumot, minden argumentum engedélyezett

```
bob ALL = (root) /usr/bin/com_1
bob$ sudo -u root com_1 arg1 arg2 arg3
```

- Parancs futtatásának engedélyezése argumentumok megadásának tiltásával
  - a program neve után a "" argumentumot kell elhelyezni

```
bob ALL = (root) /usr/bin/com_1 ""
bob$ sudo -u root com_1 arg1 arg2 arg3
```

## Jogok jelszavak változtatásához

- Engedélyezzük bob számára, hogy megváltoztathassa néhány user jelszavát

```
bob ALL = NOPASSWD: /usr/bin/passwd
  bela, /usr/bin/passwd geza
```

- Engedélyezzük az ecdl rendszergzdának az ecdl userek jelszavának megváltoztatását
  - ecdl userek pl.: ecdl00 … ecdl29 között

```
ecdl_adm ALL = NOPASSWD:
  /usr/bin/passwd ecdl[0-2][0-9]
```

## Démonok kezelése sudo-val

- Egy csoport számára szeretnénk elérhetővé tenni bizonyos démonok kezelését
- Felhasználói csoport létrehozása

```
User_Alias = CSOP=bela, bob, ede
```

- Démon csoport létrehozása

```
Cmnd_Alias = DAEMON=/etc/init.d/apache start,
  /etc/init.d/apache stop, /etc/init.d/apache
  restart
```

- Jogosultság kiadása
  - `CSOP ALL = (ALL) DAEMON`

## Néhány rossz megoldás …

- Ne használjuk a veszélyes parancsok kizárását (minden más megengedését)

```
bob ALL (root) !/usr/bin/su
```

Például:

```
bob$ ln -s /usr/bin/su akarmi
bob$ sudo akarmi
```

- Ügyeljünk az argumentumokra!

```
sudo -u root cat ~root/.ssh/id_dsa
sudo -u root chmod 777 /etc/passwd
```

- A következő programokhoz soha ne engedjünk root jogosultsággal sudo hozzáférést

```
su, sudo, visudo
```

# Kimenő hálózati kapcsolatok védelme

Programtervező informatikus BSc

Számítógépes hálózatok biztonságtechnikája

előadás

## Védtelen és védett kimenő kapcsolatok

- Titkosítás nélküli megoldások
  - A hálózaton az adatforgalom elfogható és értelmezhető
    - telnet, ftp, rlogin, rcp stb.
- Az SSH protokoll
  - SSH: Secure Shell
  - Számos hálózati feladathoz nyújt megfelelő tehnológiát
  - OpenSSH: szabadon elérhető SSH
  - Szinte minden linux disztribúció tartalmazza

## OpenSSH programok és állományok

| | |
|---|---|
| ***Kliensprogramok*** | |
| ssh | távoli bejelentkezés, távoli parancsvégrehajtás |
| scp | állományok másolása két számítógép között |
| sftp | ftp-hez hasonlító interaktív megoldás a gépek között titkosított állománycserére |
| ***Szerverprogramok*** | |
| sshd | az SSH szerverdémon |
| ***Titkosító kulcsok előállítására és kezelésére szolgáló programok*** | |
| ssh-keygen | nyilvános és titkos kulcsok létrehozására, módosítására |
| ssh-agent | saját SSH kulcsok tárolására, a jelszavak begépelésének elkerülésére |
| ssh-add | az ssh-agent által kezelt kulcsok manipulálására |
| ***Állományok és könyvtárak*** | |
| ~/.ssh | a felhasználó kulcsait és SSH beállításait tartalmazó könyvtár |
| /etc/ssh | a teljes rendszerre érvényes kulcsokat és konfigurációkat tartalmazó könyvtár |
| ~/.ssh/config | a felhasználó által használt kliens konfigurációs állománya |
| /etc/ssh/ssh_config | a teljes rendszerre érvényes kliens konfigurációs állomány |

## Az ssh és az scp használata

- Távoli bejelentkezés
  `ssh -l távoli_felhasználó távoli_kiszolgáló`
  `ssh távoli_felhasználó@távoli_kiszolgáló`
  Például: `ssh c-ta@iatt.ttk.pte.hu`
- Távoli parancsvégrehajtás
  `ssh -l ruser rhost command`
  Például: `ssh -l c-ta iatt.ttk.pte.hu w`
- Állományok másolása
  `scp állomány távoli_kiszolgáló:távoli_állomány`
  `scp távoli_kiszolgáló:távoli_állomány állomány`

## Távoli programok futtatása

- Interaktivitást nem igénylő programok esetén
  `ssh -l ruser rhost command`
  Például: `ssh -l c-ta iatt.ttk.pte.hu w`
- Interaktivitást igénylő programok esetén
  `ssh -t -l ruser rhost command`
  Például: `ssh -t -l c-ta iatt.ttk.pte.hu vi`
- X Window programok esetén
  `ssh -X -f -l ruser rhost command`
  Például: `ssh -X -f c-ta@iatt.ttk.pte.hu xterm`
  (`/etc/ssh/sshd_config X11forwarding yes | no`)

## Távoli állományok másolása

- Egy állomány másolása
  `scp állomány távoli_kiszolgáló:`
  `scp távoli_kiszolgáló:állomány .`
- Egy állomány másolása más néven
  `scp állomány távoli_kiszolgáló:másolat`
  `scp távoli_kiszolgáló:állomány másolat`
- Több állomány másolása
  `scp állomány* távoli_kiszolgáló:`
  `scp távoli_kiszolgáló:állomány\* .`

## Távoli állományok másolása

- Másolás könyvtárba

```
scp állomány távoli_kiszolgáló:dir/subdir
scp távoli_kiszolgáló:dir/subdir/állomány .
```

- Rekurzív másolás más felhasználónéven

```
scp -r könyvtár user@távoli_kiszolgáló:
scp -r user@távoli_kiszolgáló:könyvtár .
```

- Attribútumok megőrzése

```
scp -p állomány* távoli_kiszolgáló:
scp -p távoli_kiszolgáló:állomány\* .
```

## Állományok tükrözése

- Biztonságos tükrözés az scp használatával
  - `scp -pr` parancs használatával
  - hátrányai:
    - az scp automatikusan követi a szimbolikus linkeket
    - minden állományt másol (akkor is ha már létezik)
- Az rsync és az scp kombinálása
  - optimalizált, szimbolikus linkek követése nélkül
  - `rsync -a -e ssh dir1 távoli:dir2`
  - `rsync -a -e ssh -v --progress dir1 távoli:dir2`

## Hitelesítés nyilvános kulccsal OpenSSH szerver és kliens között

- Kulcsok helyének előkészítése ha szükséges

```
mkdir -p ~/.ssh
chmod 700 ~/.ssh
```

- Szükséges kulcsok generálása

```
cd ~/.ssh
ssh-keygen -t dsa
```

- A nyilvános kulcs másolása a távoli kiszolgálóra

```
scp -p id_dsa.pub user@távoli_kiszolgáló:
```

## Hitelesítés nyilvános kulccsal OpenSSH szerver és kliens között

- A nyilvános kulcs telepítése a távoli kiszolgálón

```
mkdir -p ~/.ssh
chmod 700 ~/.ssh
cat id_dsa.pub >> ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
```

- Belépés a távoli kiszolgálóra
  - Hitelesítés a nyilvános és titkos kulcs alapján
  - Jelszó megadását nem igényli

```
ssh user@távoli_kiszolgáló
```

## Az ssh-agent

- Szeretnénk elkerülni minden egyes bejelentkezéskor a jelszó megadását
- Az ssh-agent közvetítő indítása

```
eval `ssh-agent`
```

- Kulcsok hozzáadása a közvetítőhöz

```
ssh-add
```

- Bejelentkezés jelszó nélkül a távoli gépekre

```
ssh user@távoli_kiszolgáló
```

## Az ssh-agent

- Kulcsok hozzáadása az ssh-add segítségével
  - `ssh-add` – alapértelezett kulcsok hozzáadása
  - `ssh-add ~/.ssh/kulcs1` – kulcs1 hozzáadása
- Kulcsok eltávolítása
  - `ssh-add -D` – minen kulcs eltávolítása
  - `ssh-add -d ~/.ssh/kulcs1` – kulcs1 törlése
- Az ssh-agent működése
  - csak addig használhatjuk amíg ki nem jelentkezünk
  - ismételt bejelentkezéskor újra meg kell adni a kulcsokat
  - keychain …

## Az SSH kulcsok előnyei

- Nagyobb biztonság a jelszavaknál
  - A titkos kulcsot soha nem továbbítjuk a hálózaton a (titkosított) jelszavakkal ellentétben
  - A kulcsok tárolása titkosított is lehet, így ellopásuk esetén sem használhatók fel, ellentétben egy jelszóval
  - Nem kell minden egyes bejelentkezésnél a hitelesítéssel bajlódnunk, ez automatikusan megtörténik a kulcsok segítségével

## Nyílt szöveges kulcsok

- Használatuk általában nem szerencsés
  - A kulcsokat tartalmazó állomány ellopása komoly problémát jelentene
- Az emberi beavatkozástól mentes folyamatoknál (kötegelt feladatok, cron feladatok) szükséges lehet
- Szűkíteni kell a kulcs felhasználásának lehetőségeit
  - Csak az általunk engedélyezett parancs(ok) legyen(ek) végrehajtható(k) a kulccsal

## Nyílt szöveges kulcsok a gyakorlatban

- Nyílt szöveges kulcsok készítése
  `ssh-keygen -t dsa -f key2 -N ""`
- Kulcs telepítése a kiszolgálón
- Kényszerparancs megadása a kiszolgálón
  - `command="/usr/bin/uptime" ssh-dss AAAAB3NzaC1kc …`
- Egyéb lehetőségek tiltása
  - `no-port-forwarding, no-X11-forwarding, no-agent-forwarding, no-pty, from="kliens.pte.hu", command="/usr/bin/uptime" ssh-dss AAAAB3NzaC1k …`
- A kulcs felhasználása
  - `ssh -i key2 távoli_kiszolgáló`

## OpenSSH és SSH Secure Shell

- SSH1 esetén kompatibilisek
- SSH2 esetén eltérő állományformátumok
- A nyilvános kulcsok formátuma
  - OpenSSH esetén:

```
ssh-dss AAAB3NzaC1kc3MAAAC …
ssh-rsa AAAB3NzaC1kc3MAAAC …
```

  - SSH Secure Shell esetében:

```
---- BEGIN SSH2 PUBLIC KEY ----
AAAAB3NzaC1kc3MAAACBAJNN2CbURaTm7oW5F2Z …
---- END SSH2 PUBLIC KEY ----
```

## OpenSSH és SSH Secure Shell

- Nyilvános kulcsok telepítése
  - OpenSSH
    - `~/.ssh/authorized_keys` állományban
  - SSH Secure Shell
    - `~/.ssh2` könyvtárba másolva
    - `~/.ssh2/authorization` állományban hivatkozva
- Titkos kulcsok
  - OpenSSH
    - Nincs megkötés
  - SSH Secure Shell
    - `~/.ssh2/identification` állományban hivatkozva

## OpenSSH és SSH Secure Shell

- OpenSSH kulcs exportálása SSH2 formátumú nyilvános kulccsá

```
ssh-keygen -e -f id_dsa > mykey-ssh2.pub
```

- SSH2 nyilvános kulcs élesítése

```
mv mykey-ssh2.pub ~/.ssh2/
echo "K1 mykey-ssh2.pub" >>authorization
```

- SSH2 titkos kulcs átalakítása OpenSSH formára

```
cp -p is_dsa_ssh2 kulcs_másolat
ssh-keygen2 -e kulcs_másolat
ssh-keygen -i -f kulcs_másolat > imp_ssh2_key
ssh-keygen -p imp_ssh2_key
```

## Az SSH bejelentkezés egyszerűsítése

- Kiszolgálói alias-ok készítése a ~/.ssh/config állomány használatával
- ~/.ssh/config

```
Host akarmi
 HostName akarmi.akarhol.hu
 User root
 IdentityFile ~/.ssh/akarmikey_dsa
 Port 33333
 Protocol 2
```

## Az SSH kliens alapértelmezett beállításai

- Kiszolgálói alias a ~/.ssh/config állományban "*" néven
- Ha ez az első bejegyzés, minden mást felülír
- Ha ez az utolsó bejegyzés, akkor csak tartalék szerepet tölt be
- Például

```
Host *
  User c-ta
Host iatt.ttk.pte.hu
  User root
  Port 3453
Host *
  Protocol 2
```

## SSH alagutak készítése

- Nem biztonságos TCP kapcsolatok alagúton történő továbbítása, SSH használatával
- Alagút létrehozása
  - `ssh -f -N -L helyi_port:localhost:távoli_port távoli_kiszolgáló`
- Az adatok áramlásának folyamata
  - az alkalmazás adatokat küld a helyi_port-ra
  - a helyi SSH kliens olvassa a portot, titkosítja az adatokat és az alagúton keresztül elküldi a távoli SSH szervehez
  - a távoli SSH szerver dekódolja az adatokat és helyben továbbítja a távoli_port értékével megadott portra

# Biztonságos levelezés

Programtervező informatikus BSc
Számítógépes hálózatok biztonságtechnikája
előadás

## Az e-mail szerepének változása

- Az e-mail használat változásaival kapcsolatban a **Symantec** végzett átfogó kutatást
- Kérdések:
  - A folyamatosan növekvő e-mail használat miértje és következménye
  - A növekvő e-mail használat hatásának vizsgálata az üzleti életre
  - A felhasználók e-mail függőségének elemzése
  - Az e-mailekben tárolt adatok fontossága, értéke
- A kutatásról:
  - Független kutatás az 500 főnél többet foglalkoztatók körében
  - 1700 alkalmazottal és IT menedzserrel folytatott interjú
  - 17 európai országban

## A kutatás eredményének gyors áttekintése

- Manapság az alkalmazottak munkaidejük egyre nagyobb részét töltik e-mailezéssel
- Egyre több embernél alakul ki függőség
- Az e-mailek száma jelentősen növekszik
  - az e-mailekre utaltság miatt
  - spam és egyéb levelek
- Terjed az e-mailezés mobil eszközökön is, ami oda vezet, hogy a felhasználók mindig és mindenhol ellenőrzik leveleiket

## A függőség

- Az alkalmazottak 75%-a szerint könnyű rászokni az e-mail használatára
- 21% már folyamatosan kényszert érez az e-mailek ellenőrzésére
- Az e-mailezők típusai
  - Túlterhelt (6%)
  - Függő (21%)
  - Elutasító (10%)
  - Tudatos (49%)

## Üzleti hatás

- Az IT menedzserek 91%-a szerint átlagosan 47%-al nőtt az e-mailek mennyisége az elmúlt években
- A felhasználók több mint fele (52%) napi 2 órát tölt levelezéssel (hetente 1 munkanap)
- 15% napi 4 órát fordít a levelezésre
- Legfontosabb felhasználási területek:
  - Tárgyalás paramétereinek ellenőrzése (74%)
  - Dokumentumok keresése (74%)
  - Kontakt adatok keresése (62%)
  - Személyes jellegű kommunikáció (60%)

## A mobil e-mailezők

- A megkérdezett személyek 31%-a vallotta magát mobil e-mailezőnek
- 27%-uk szerint ez rossz hatással van a munka/magánélet egyensúlyára, növeli a stresszt
- A mobil e-mailezők harmada ellenőrzi leveleit közvetlenül lefekvés előtt és ébredés után
- A mobil e-mailezők többsége a mobileszközt munkahelyi ügyek intézésére használja még baráti, családi társaságban is
  - szabadság alatt 40% lép be a levelező rendszerbe
  - betegség esetén 38% ellenőrzi a leveleit
  - a gyerek születésnapján is van 5% aki ezzel foglalkozik

## Az e-mailek biztonsága

- A vállalatoknak csak 23%-a rendelkezik valamiféle irányelvvel az e-mailek tárolásáról
- A dolgozók nem tudják, hogy melyik levelet kell tárolni és melyiket törölhetik
- Biztonsági másolat
  - az alkalmazottak 50%-a úgy gondolja, hogy az ő feladata a biztonsági másolat készítése
  - 47% úgy gondolja, hogy ez az IT részleg dolga
  - 64% abban a tudatban van, hogy az IT részleg minden fogadott és küldött e-mailről másolatot készít
  - 80% úgy tudja (tévesen), hogy a cég a törölt levelekről is őriz másolatot
  - A vállalatoknak csak 44%-a készít automatikus mentést a felhasználó merevlemezére mentett e-mailekről

## A tárolt információ értéke

- A mobil eszközön is e-mailezők 78%-a nyilatkozott úgy, hogy az e-mailekben (és a notebookon) tárolt adatok jelentő értéket képviselnek
  - Átlagos érték 200 millió forint …
  - Legnagyobb érték a kutatás során: 1,5 milliárd forint
- Megállapítható, hogy ezek az adatok jelentős értéket képviselnek
- A cégeknek így az adatok védelmére is célszerű lenne áldozni a hardware vásárlás mellett
- Jelenleg kevés vállalat van felkészülve az ellopott, megsemmisült adatok visszanyerésére

## Megoldás az e-mailek biztonságos tárolására

- Központilag minden beérkező és elküldött e-mailről egy másolat tárolása
  - Beérkezett üzeneteknél egy másolat marad a kiszolgálón
  - Elküldött leveleknél nehezebb a helyzet, más (külső) kiszolgálók is használhatók a küldéshez
- Központi szabályzatok kidolgozása és következetes alkalmazása
- Felhasználók képzése, oktatása

## E-mailek biztonsági problémái

- Minden e-mail a feladótól a címzettig számos gépen halad keresztül
- A célba ért levelek is könnyen elolvashatók (megfelelő jogosultságok birtokában)
- Sem a feladó, sem a címzett nem veszi észre ha más is olvasta az e-mailt
- A hagyományos POP vagy IMAP kapcsolatok esetén az üzenetet kívülállók is elfoghatják
- A különböző szegmensek biztonságosabbá tétele:
  - A küldőtől a címzettig: üzenetek titkosítása, aláírása
  - A levelezőszerver és a levelezőkliens között: biztonságos IMAP vagy biztonságos POP használatával, alagútazás segítségével

## Leggyakoribb visszaélések

- E-mailben küldött információk lehallgatása
- Levélbombák (postafiók megtöltése, levelezőrendszer összeomlása)
- Megszemélyesítés a levél feladójának megváltoztatásával
- Vírusok terjesztése
- Lánclevelek "hoax"-ok indítása
- A levelezőszerver kontrolljának megszerzése egyéb támadások indításához

## Levelek titkosítása - Pine

- Megoldás: a PinePGP használata
- A PinePGP telepítése:
  - Pine normál futtatása (~/.pinerc állomány létrhozása)
  - PinePGP letöltése és telepítése
    - pinegpg-install
      (csak a címzett tudja visszafejteni az üzenetet)
    - pinegpg-install c-ta@ttk.pte.hu
      (megadott e-mail cím (feladó) is vissza tudja fejteni)
- A pine a leveleket a ~/.pinerc állomány sending-filters és display-filters változóival szűri

## Üzenetek kezelése PinePGP-vel

- Üzenet küldése
  - Küldéskor (CTRL-x) a pine megkérdezi, hogy milyen szűrőt használjon
    - send message (filtered thru "gpg-sign")?
    - send message (filtered thru "gpg-encrypt")?
    - send message (filtered thru "gpg-sign+encrypt")?
  - Aláírásnál meg kell adni a használt kulcs jelszavát
- Üzenetek olvasása
  - Titkosított üzenetnél a pine bekéri jelszavunkat és ha ez helyes akkor megjeleníti az üzenetet
  - Az üzenet elején és végén a [PinePGP] jelölés található

## Levelek titkosítása - Mozilla

- Enigmail (enigmail.mozdev.org) használata
- Üzenet küldése
  - Levél írása hagyományos módon
  - Rendelkeznünk kell a címzett nyilvános kulcsával
  - Küldés (Send) helyett az Enigmail opciót használjuk
    - titkosítás, aláírás, titkosítás+aláírás
- Üzenetek olvasása
  - Titkosított levél megnyitásakor a Mozilla bekéri a kulcshoz tartozó jelszavunkat

## Biztonságos SSL kapcsolatok

- A legtöbb levelezőkliens támogatja
- A legtöbb kereskedelmi levelezőszerver viszont nem
- Az SSL támogatás módja a levelező szerverekben
  - Az SSL élesítése a kapcsolat felépítése után
    - IMAP: a szerver a normál 143-as portot figyeli, az SSL élesítése a STARTTLS paranccsal történik
    - POP: a szerver a normál 110-es portot figyeli, az SSL élesítése a STLS paranccsal történik
  - SSL port használata
    - IMAP: 993-as port, SSL egyeztetés a kapcsolat felvétele előtt
    - POP: 995-ös port, SSL egyeztetés a kapcsolat felvétele előtt

## POP/IMAP levelezőszerver + SSL

- Legfőbb cél a jelszavaink védelme
- Az adatfolyam (e-mail tartalmának) védelmével nem itt kell foglalkozni
  - Az üzenet tartalma korábbi fázisokban könnyebben megszerezhető
- Az imapd program használata SSL protokollal
- Kliens beállítása SSL protokoll használatára
  - Ha a kliens támogatja a STARTTLS megoldást, akkor nincs több tennivalónk, készen is vagyunk
  - Ha nem, a szerveren önálló portokat kell beállítani

## POP/IMAP levelezőszerver + SSL

- Önálló portok élesítése az SSL kapcsolathoz
  - IMAP használatánál
    - service imaps { … disabled = no } (xinetd)
    - imaps stream tcp nowait root /usr/sbin/tcpd imapd (inetd)
    - xinetd vagy inetd konfiguráció újraolvasása Önálló portok élesítése az SSL kapcsolathoz
  - POP használatánál
    - service pop3s { … disabled = no } (xinetd)
    - pop3s stream tcp nowait root /usr/sbin/tcpd ipop3d (inetd)
    - xinetd vagy inetd konfiguráció újraolvasása

## POP/IMAP levelezőszerver + SSL

- Tesztelés
  - `openssl s_client -quiet -connect localhost:993`
    - kilépés: `0 LOGOUT`
  - `openssl s_client -quiet -connect localhost:995`
    - kilépés: `QUIT`
- Egy lehetséges kimenet

```
depth=1
   /C=HU/ST=Baranya/L=Pecs/O=PROBA/CN=proba.dravane
   t.hu/emailAddress=akarki@akarhol.hu
verify error:num=19:self signed certificate in
   certificate chain
verify return:0
+OK Hello there.
```

## POP/IMAP SSH alagúttal

- A `mailhost` nevű szerverről szeretnénk lekérni üzeneteinket a `myclient` gépünkre
- Válasszunk egy tetszőleges (szabad) TCP portot a gépünkön. Legyen ez most : 12345
- Az alagút felépítése:

```
ssh -f -N -L 12345:localhost:110 mailhost
ssh -f -N -L 12345:localhost:143 mailhost
```

- Levelezőkliens beállítása az 12345 portra

## SMTP szerverek

- E-mailek fogadása és továbbítása az Interneten
  - Helyi levelek: a kiszolgáló egy helyi felhasználójához kell kézbesíteni az üzenetet
  - Nem helyi levelek: a kézbesítéshez másik kiszolgálóhoz kell továbbítani az üzenetet
- Open relay szerverek
  - bárki használhatja őket továbbítóként (célszerű kerülni)
  - Problémái:
    - spammerek előszeretettel használják
    - levelezőszerverünk ezáltal feketelistára kerül és használhatatlan lesz
- Szolgáltatók levelező szerverei
  - Általában csak a saját hálózatuk címeiről továbbítanak leveleket
  - A mobil használók szempontjából ez sok kellemetlenséggel jár

## SMTP hitelesítés

- Hitelesítés után bármely hálózatból küldhetünk levelet mindig azonos SMTP szervert használva
- Írjuk át a sendmail.mc állományban a következőt

```
DAEMON_OPTIONS(`Port=smtp, Addr=127.0.0.1, Name=MTA`)
```

erre:

```
DAEMON_OPTIONS(`Port=smtp, Name=MTA`)
```

- Engedélyezzük a következő sort a sendmail.mc-ben

```
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN`)
```

- A sendmail konfiguráció újratelepítése

```
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

- A sendmail újraindítása

```
/etc/init.d/sendmail restart
```

- Felhasználói fiókok létrehozása az SMTP hitelesítéshez

```
/usr/bin/saslpasswd -c c-ta
```

## Az SMTP szerver tesztelése

```
c-ta@proba:~$ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.
220 proba.dravanet.hu ESMTP Postfix (Debian/GNU)
helo akarmi.akarhol.hu
250 pvsk.dravanet.hu
mail from:<akarki@akarmi.akarhol.hu>
250 Ok
rcpt to:<c-ta@ttk.pte.hu>
250 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: Próba üzenet
Ez egy teszt üzenet
.
250 Ok: queued as 46A442800087
quit
221 Bye
Connection closed by foreign host.
```

# Webkiszolgálók biztonsága, Apache

Programtervező informatikus BSc
Számítógépes hálózatok biztonságtechnikája
előadás

## Hitelesítési protokoll

- Kliens elküldi a felhasználónevét és jelszavát
  - Az apache összehasonlítja ez a kiszolgálón tárolt felhasználó névvel és titkosított jelszóval
  - Azonosítás után a felhasználó jogosultságának vizsgálata
  - Csoportok kezelése (egyszerűbb)
- Biztonsági problémák
  - Jelszavak nyílt szövegként kerülnek továbbításra

## Hitelesítési direktívák

- Az `AuthType` direktíva
  - Leírás: az azonosítás típusa
  - Szintaxis: `AuthType Basic|Digest`
  - Használható: `<directory>, .htaccess`
- Az `AuthName` direktíva
  - Leírás: a tartomány megnevezése
  - Szintaxis: `AuthName auth-domain`
  - Használható: `<directory>, .htaccess`

## Hitelesítési direktívák

- A `Require` direktíva
  - Leírás: megadja, hogy melyik hitelesített felhasználó használhatja az erőforrást
  - Szintaxis:
    - `Require user userid1 userid2 …`
    - `Require group groupid1 groupid2 …`
    - `Require valid-user`
  - Használható: `<directory>, .htaccess`

## Hitelesítési direktívák

- Az `AuthUserFile` direktíva
  - Leírás: megadja a felhasználók nevét és jelszavát tartalmazó szöveges fájl elérési útvonalát
  - Szintaxis: `AuthUserFile file-path`
  - Használható: `<directory>, .htaccess`
- Az `AuthGroupFile` direktíva
  - Leírás: megadja a csoportok nevét és felhasználóit tartalmazó szöveges fájl elérési útvonalát
  - Szintaxis: `AuthGroupFile file-path`
  - Használható: `<directory>, .htaccess`

## Jelszavak kezelése

- Jelszavak tárolása egy szöveges fájlban
- Jelszó állomány helye
- Jelszavak kezelésére használható segédprogram
  - `htpasswd` vagy `htpasswd2`
- Szintaxis
  - `htpasswd [-c] passwordfile username`
    - `-c` új jelszófájt hoz létre, ha az még nem létezik
    - `passwordfile` a jelszófájl elérési útvonala
    - `username` a felhasználó neve

## Jelszavak kezelése - dbmanage

- Sok bejegyzés esetén a szöveges fájlokat használva a folyamat lelassulhat
- Használhatók adatbázis állományok is
- Jelszavak kezelésére használható segédprogram
  - `dbmanage` vagy `dbmanage2`
- Szintaxis
  - `dbmanage [enc] dbmfile command username`
    - `enc` - a titkosítás algoritmusa
    - `dbmfile` – a jelszófájl elérési útvonala
    - `command` – parancs
    - `username` – a felhasználó neve

## Jelszavak kezelése - dbmanage

- Titkosítási lehetőségek
  - `-d` crypt kódolás (alapértelmezett)
  - `-m` MD5 kódolás
  - `-s` SHA1 kódolás
  - `-p` nyílt szöveg használata
- Parancsok
  - `add` felhasználó hozzáadása csoporthoz
  - `adduser` felhasználó hozzáadása
  - `check` jelszó ellenőrzése
  - `delete` felhasználó törlése
  - `import` importálás (?)
  - `update` felhasználó jelszavának megváltoztatása
  - `view` felhasználó(k), csoport(ok) adatainak megtekintése

## Példák felhasználók kezelésére

- Példák a `htpasswd` használatára
  - `htpasswd -c http_jelszavak c-ta`
  - `htpasswd http_jelszavak viktor`
- Példák a `dbmanage` használatára
  - `dbmanage http_pw.db adduser c-ta`
  - `dbmanage http_group.db add c-ta oktatok`
  - `dbmanage http_group.db view`
    - `viktor:oktatok`
    - `c-ta:oktatok`
  - `dbmanage http_pw.db view`
    - `c-ta:VSVX7dCwOqBfs`
    - `viktor:RCdHZlcjKZO1k`

## .htaccess példa

- Minden hitelesített user számára elérhető hely

```
AuthType Basic
AuthName "Védett webhely"
AuthUserFile /home/c-ta/http_jelszavak
Require valid-user
```

- Csak "c-ta" és "viktor" számára elérhető hely

```
AuthType Basic
AuthName "Védett webhely"
AuthUserFile /home/c-ta/http_jelszavak
Require user c-ta viktor
```

## A .htaccess fájlok használata

- A direktívák elhelyezhetők:
  - `httpd.conf` konfigurációs állományban
  - `.htaccess` fájlokban
- A .htaccess használata esetén
  - A direktívák megváltoztatásakor nem kell minden alkalommal a szervert újraindítani
  - A felhasználók saját maguk tudják karbantartani az oldalaikat, képesek a jogosultságok szabályozására
  - A .htaccess fájlok minden egyes hozzáférésnél kiértékelésre kerülnek, ami jelentős teljesítménycsökkenést jelent

## További korlátozások

- A `<Limit>` direktíva
  - Leírás: bizonyos (felsorolt) metódusok hozzáférésének korlátozására. Általában használata csak ritkán indokolt.
  - Szintaxis: `<Limit method [method] …> … </Limit>`
  - Használható: `server config, virtual host, <directory>, .htaccess`
  - Lehetséges metódusok: `GET, POST, PUT, DELETE, CONNECT, OPTIONS, PATCH, PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK`
- Például:

```
<Limit POST PUT DELETE>
 Require valid-user
</Limit>
```

## További korlátozások

- A `<LimitExcept>` direktíva
  - Leírás: bizonyos (fel nem sorolt) metódusok hozzáférésének korlátozására. Általában használata ritkán indokolt.
  - Szintaxis: `<LimitExcept method [method] …> … </LimitExcept>`
  - Használható: `server config, virtual host, <directory>, .htaccess`
- Például:
  - ```
    <LimitExcept POST PUT DELETE>
     Require valid-user
    </LimitExcept>
    ```

## Kliensek hozzáférésének szabályozása - engedélyezés

- Az `Allow` direktíva
  - Leírás: hosztok hozzáférésének engedélyezése
  - Szintaxis: `Allow from all|host|env=env-variable [host|env=env-variable] ...`
  - Használható: `&lt;directory&gt;, .htaccess`
- Hosztok megadása:
  - (részleges) domain név: `Allow from pte.hu`)
  - IP cím: `Allow from 192.168.1.2 192.168.1.3`
  - Részleges IP cím: `Allow from 192 172.17.4`
  - IP+Netmask: `Allow from 10.1.0.0/255.255.0.0`
  - IP+Netmask: `Allow from 10.1.0.0/16`

## Kliensek hozzáférésének szabályozása - tiltás

- A `Deny` direktíva
  - Leírás: hosztok hozzáférésének tiltása
  - Szintaxis: `Deny from all|host|env=env-variable [host|env=env-variable] ...`
  - Használható: `&lt;directory&gt;, .htaccess`
- Környezeti változók használata

```
SetEnvIf User-Agent ^KnockKnock/2\.0 let_me_in
Directory /docroot>
 Order Deny,Allow
 Deny from all
 Allow from env=let_me_in
</Directory>
```
```

## Kliensek hozzáférésének szabályozása - sorrendek

- Az `Order` direktíva
  - Leírás: az `Allow` és `Deny` direktívák végrehajtásának sorrendjét szabályozza
  - Szintaxis: `Order ordering`
  - Használható: `<directory>, .htaccess`
- Az `ordering` lehetőségei:
  - `deny,allow` - a `deny` direktívák végrehajtása megelőzi az `allow` direktívák végrehajtását (alapértelmezés)
  - `allow,deny` - a `allow` direktívák végrehajtása megelőzi az `deny` direktívák végrehajtását
  - `mutual-failure` - az `allow` listában megjelenő és a `deny` listában meg nem jelenő hosztok számára biztosít hozzáférést

## Példák az `Order` használatára

- A 192.168.X.X IP címek engedélyezése, minden más cím tiltása

```
Order deny,allow
allow from 192.168
deny from all
```

- Mi történik a 192.168.X.X címekről érkező kérésekkel?

```
Order allow,deny
allow from 192.168
deny from all
```

## További lehetőségek

- A `Satisfy` direktíva
  - Leírás: hoszt- és felhasználói szintű azonosítás összehangolása
  - Szintaxis: `Satisfy` **`All`** `| Any`
  - Használható: `<directory>, .htaccess`
- Például:

```
require valid-user
Satisfy Any
Order Deny,Allow
Allow from 192.168.1.100
Deny from All
```

## Kivonatolt hitelesítés

- Egyirányú kódolás, a kivonatot amit az azonosításhoz használunk a jelszóból és más információkból számítja ki a program
- Kiegészítő információk
  - URL
  - Metódus
  - Egyéni érték (nonce), egy mindig változó szám
- A titkosítás nélküli és a teljes körűen titkosított megoldás között található
- Az MD5 kivonatolás élesítése:
  - AuthType Digest

## A hitelesítés folyamata

- A kliens igényel egy URL-t
- Ha az URL védett a szerver jelzi a hitelesítés szükségességét és a fejlécek között elküldi az egyéni értéket
- A kliens összekombinálja az URL-t, a metódust, az egyéni értéket és a felhasználó jelszavát. Az eredményt visszaküldi a szervernek.
- A szerver ugyanezt a műveletet végzi a felhasználó kódolt jelszavával és egyezteti az eredményeket.
- A kivonatoló függvény:
  - MD5(MD5(<password>)+":"+<nonce>+":"+MD5(<method>+":"+<URL>))

## A kivonatolt hitelesítés problémái

- A módszer használata csak teszt- vagy ellenőrzött környezetben ajánlott
- Problémát jelenthet az egyedi értékek visszajátszhatósága. Az Apache a visszakapott egyedi értéket nem ellenőrzi.
- Lehetséges megoldások
  - Az egyedi érték kombinálása egy időbélyeggel (jelenleg nem támogatott)
  - Az egyedi értékek időkorláttal történő ellátása az `AuthDigestNonceLifetime` direktívával

## Anonim hozzáférés

- Belépési lehetőséget biztosít vendégek számára
- Korlátozott jogokkal bíró "érdeklődők" számára
- Szükséges modul: `auth_anon_module`
- Az `Anonymous` direktíva
  - Leírás: Jelszó nélküli belépési lehetőséget biztosít
  - Szintaxis: `Anonymous user [user] ...`
  - Használható: `<directory>, .htaccess`

## További direktívák az anonim hozzáféréshez

- Az `Anonymous_Authoritative` direktíva
  - Leírás: csak anonymous bejelentkezés engedélyezése. Ha a login név nem szerepel az anonymous listában, akkor elutasítja a kérést
  - Szintaxis: `Anonymous_Authoritative On|Off`
  - Használható: `<directory>, .htaccess`
- Az `Anonymous_LogEmail` direktíva
  - Leírás: a jelszóként várt e-mail cím naplózása
  - Szintaxis: `Anonymous_LogEmail On|Off`
  - Használható: `<directory>, .htaccess`

## További direktívák az anonim hozzáféréshez

- Az `Anonymous_MustGiveEmail` direktíva
  - Leírás: az üres jelszavak megelőzésére. A kliens meg kell, hogy adja az e-mail címét jelszóként.
  - Szintaxis: `Anonymous_MustGiveEmail On|Off`
  - Használható: `<directory>, .htaccess`
- Az `Anonymous_VerifyEmail` direktíva
  - Leírás: a megadott e-mail cím "ellenőrzése". Minimum egy "@" és egy "." karaktert tartalmaznia kell.
  - Szintaxis: `Anonymous_VerifyEmail On|Off`
  - Használható: `<directory>, .htaccess`

## További direktívák az anonim hozzáféréshez

- Az `Anonymous_NoUserID` direktíva
  - Leírás: bekapcsolt állapotban lehetőséget ad a kliensnek arra, hogy ne adjon meg felhasználó nevet (esetleg jelszót sem).
  - Szintaxis: `Anonymous_NoUserID On|Off`
  - Használható: `<directory>, .htaccess`

## Biztonsági kérdések

- Felhasználók
  - Külső felhasználók
  - Belső felhasználók
- A külső felhasználók csak ahhoz férhessenek hozzá amit engedélyeztünk számukra
- A szerverünk se legyen sebezhető
  - Túlcsordulásokkal szembeni védelem
  - A címsorba írt speciális karakterekkel szembeni védelem
- Szerver indítása milyen jogokkal történjen?
- Szkriptek futtatása …

## Biztonsági óvintézkedések

- Indításkor több példány létrehozása
  - Csak az eredeti processz fut a superuser nevén
    - Nem szolgál ki hálózati kéréseket
    - Felügyeli a többi processzt, újakat indít, régieket leállít
  - A további példányok egy korlátozott jogokkal rendelkező identitásra váltanak (pl.: wwwrun)
    - A hálózati kérések kiszolgálására
- A shellprogramokhoz eljutó karakterek vizsgálata
  - A veszélyes karakterek védelme a "\" karakterrel
  - Értelmezéskor (kiértékeléskor) ezek a karakterek eltűnnek
    - Egy új shell-nek továbbadva már ismét veszélyt jelentenek

## Tanúsítványok

- Bizonyosak szeretnénk lenni abban, hogy a kommunikációs csatorna végein ténylegesen azok vannak akiket gondolunk
- Tanúsítványok alkalmazása
  - Egy köztiszteletben álló személy vagy hivatalos cégképviselő által aláírt (a kibocsátó titkos kulcsával titkosított) elektronikus dokumentum
  - Általában tartalmazza: a tulajdonos nyilvános kulcsát és adatait (név, e-mail cím, cégnév, stb.)
  - Manapság szinte minden igazoló hatóság saját maga által aláírt tanúsítványt használ. Nincs szabályozva, hogy mikor kell második szintű tanúsítványt beszerezni.
- Tanúsítványok használata
  - Egymás közti kölcsönös tanúsítványcsere (titkosított formában)
  - Egymás tanúsítványának ellenőrzése a kibocsátó nyilvános kulcsával

# Tesztelés és felügyelet

Programtervező informatikus BSc

Számítógépes hálózatok biztonságtechnikája

előadás

## Főbb területek

- A szerverek egyszeri tökéletes beállítása nem elég
- Folyamatos felügyelet szükséges
  - Biztonsági lyukak keresése
  - Szokatlan viselkedések figyelése
  - Gyanús felhasználói tevékenység észlelése
  - Támadási kísérletek naplózása
- Főbb témáik
  - Bejelentkezések és jelszavak
  - Fájlrendszerek
  - Hálózatkezelés
  - Naplózás

## Bejelentkezési jelszavak ellenőrzése John the Ripper

- SuSE disztribúcióknak általában része
- Egyszerű használat (minden fiókra)
  - `unshadow /etc/passwd /etc/shadow > jelszavak`
  - `john jelszavak`
- Feltört jelszavak megmutatása
  - `john -show jelszavak`
- Csak bizonyos jelszavak teszteléses
  - `john -users:bela,geza,rita jelszavak`
  - `john -groups:oktatok,hallgatok jelszavak`

## Bejelentkezési jelszavak ellenőrzése John the Ripper

- `unshadow` utasítás
  - egybegyűjti a `passwd` és `shadow` állományok információit
  - minden jelszó változtatás után le kell futtatni az utasítást
  - fontos az újraintegrált állomány biztonsága
  - éles környezetben az ilyen műveletet célszerű a hálózatról leválasztva végezni
- Szótárak
  - gyakori szavakat tartalmazó szótárak használata
  - szótári szavak és azok permutációi, kis és nagybetűk, stb.
  - érdemes a felhasználók nyelvéhez tartozó szótárt letölteni
    - `ftp://ftp.ox.ac.uk/pub/wordlists/`
    - `ftp://ftp.cerias.purdue.edu/pub/dict/wordlists/`
  - a környezetnek megfelelő szavak hozzáadása a szótárhoz

## Bejelentkezési jelszavak ellenőrzése a CrackLib használata

- A Crack jelszótörő program egy része
- Más programokba történő beágyazásra készült
- `FascistCheck` függvénnyel használható
  - első paramétere: jelszó
  - második paramétere: szótár
- Készíthető saját program is a tesztelésre
- PAM modulokba ágyazható
- SuSE esetén a Yast-ban is beállítható

## Bejelentkezési jelszavak ellenőrzése a CrackLib használata

```
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <crack.h>
#define DICTIONARY "/usr/lib/cracklib_dict"
int main(int argc, char *argv[]) {
 char *password;
 char *problem;
 int status = 0;
 printf("A kilépéshez adjon meg egy üres jelszót.\n");
 while ((password = getpass("\nJelszó: ")) != NULL && *password) {
  if ((problem = FascistCheck(password, DICTIONARY)) != NULL) {
   printf("Rossz jelszó: %s.\n", problem);
   status = 1;
  } else {
   printf("Jelszó rendben\n");
  }
 }
 exit(status);
}
```

## Jelszóval nem rendelkező fiókok szűrése

- A legveszélyesebb jelszó, ha nincs jelszó
- A jelszavak a /etc/shadow állományban találhatók
  - Csak a rendszergazda férhet hozzá az állományhoz
  - Kódolt jelszavak
  - A jelszavak a második mezőben találhatók
    - pl.: `c-ta:E5x9NtV/3uWxw:13269:0:99999:7:::`
- A jelszóval nem rendelkező fiókok listázása
  - `awk -F: '$2 == "" {print $1, "Üres jelszó" }' /etc/shadow`
- A jelszó nélküli és az "üres" jelszavas fiókok nem azonosak!

## Rendszeradminisztrátori fiókok szűrése

- A rendszergazda felhasználók azonosítója: 0
- Az azonosítók a /etc/passwd állományban találhatók
  - Csak a rendszergazda férhet hozzá az állományhoz
  - Az azonosítók a harmadik mezőben találhatók
    - pl.: c-ta:x:1000:100:Rébay Viktor:/home/c-ta:/bin/bash
- A rendszeradminisztrátori fiókok listázása
  - `awk -F: '$3 == 0 {print $1, "Rendszergazda" }' /etc/passwd`
- A nem 0-s azonosítóval rendelkező felhasználóknak is lehetnek kiemelt jogaik

## Gyanús fiókhasználat - az utolsó bejelentkezés adatai

- Veszélyt jelentő fiókok
  - Szunnyadó felhasználói fiókok
  - A számos sikertelen bejelentkezéssel rendelkező fiókok
  - Gyenge jelszóval rendelkező fiókok
- A felhasználó(k) utolsó bejelentkezése
  - `lastlog [-u felhasználónév]`
- A felhasználó(k) utolsó bejelentkezésének tárolása
  - /var/log/lastlog
    - adatbázis formátum (nem napló állomány)
    - látszólagos mérete nem változik
    - néhány disztribúció csak a rendszergazdáknak engedi olvasni
- A felhasználó(k) teljes login előzménye
  - `last [-u felhasználónév]`

## Gyanús fiókhasználat - sikertelen bejelentkezések

- Ki- és bejelentkezések, kikapcsolás, újraindítás, futási szint változás tárolása:
  - `/var/log/wtmp` állományban
- A sikertelen bejelentkezések naplózásának bekapcsolása
  - touch /var/log/btmp
  - chown --reference=/var/log/wtmp /var/log/btmp
  - chgrp --reference=/var/log/wtmp /var/log/btmp
- A btmp és wtmp állományok mérete folyamatosan növekszik, gondoskodni kell a forgatásukról

## Keresési útvonalak tesztelése

- A keresési útvonal megváltoztatásával elérhető hogy egy ismert program helyett valami más induljon el
- A keresési útvonalban ne szerepeljen relatív elérés
- `perl -e 'print "a PATH relatív könyvtárat tartalmaz\"$_\"\n" foreach grep ! m[^/], split /:/, $ENV{"PATH"}, -1;'`
- Például a `cat` parancs kiadása a /tmp könyvtárban
  - Normál esetben a `/bin/cat` indul
  - Ha a `/tmp` könyvtárban is van egy `cat` állomány
    - Ha a "." előrébb van a $PATH-ban mint "/bin", a /tmp-ben lévő indul
    - Ha a "." hátrébb van akkor a szokott `cat` parancs kerül végrehajtásra
- Veszélyek
  - Gyakran használt fájlnevek mögé bújtatott ártó programok
  - Hatásuk nem mindig vehető észre, mert elvégzik a várt funkciót is
  - Minél előbb szerepel a "." annál nagyobb a veszély
  - Az utolsó pozícióban sem veszélytelen, kihasználhatók az elírások

## A setuid vagy setgid programok

- A setuid bit segítségével megoldható, hogy a tulajdonos felhasználó nevében fusson a program
- Például a jelszó változtatásához a `passwd`
  - Szükséges, hogy minden user hozzáférjen a `/etc/shadow` állományhoz
  - Mindenki csak az engedélyezett műveletet hajthassa végre
  - `ls -l /usr/bin/passwd` kimenete
  - `-rwsr-xr-x  1 root shadow 75144 Sep  9 2005 /usr/bin/passwd`
- Az ilyen lehetőséggel megfontoltan kell bánni mert komoly biztonsági problémákat okozhatnak

## A setuid vagy setgid programok

- Érdemes tehát áttekinteni ezen programok listáját
  - `find / -xdev -type f -perm +ug=s -print`
  - Adjunk meg kezdőkönyvtárat, mert így minden csatolt fájlrendszerben keresni fog
    - NFS fájlrendszerekben ez elég lassú lehet
    - Bizonyos fájlrendszerekben pedig nincs is értelme
- Keresés a felhasználók könyvtáraiban
  - `find /home -xdev -type f -perm +ug=s -print`
- A setuid vagy setgid bitek eltávolítása
  - `chmod u-s állomány` *a setuid bit eltávolítása*
  - `chmod g-s állomány` *a setgid bit eltávolítása*

## Speciális eszközállományok

- Olyan állományok amelyek lehetővé teszik eszközök közvetlen elérését a fájlrendszeren keresztül
- A biztonság érdekében ezek hozzáférése gondos felügyeletet igényel
- Ezek másolatai kiindulópontok lehetnek a memória, a lemezmeghajtók és egyéb fontos eszközök olvasásához
- A speciális eszközállományok listázása
  - `find /dir -xdev \( -type b -o -type c \) -ls`
    - `-type b`: blokkos állományok
    - `-type c`: karakteres állományok
    - nem csak a `/dev` könyvtárban lehetnek ilyen állományok
- A /dev összes szabályos állományának listázása (kivéve MAKEDEV)
  - `find /dev -type f ! -name MAKEDEV -ls`
    - szabályos állományok a speciális állományok helyettesítésére
    - rejtőzködő setuid vagy setgid állományok
    - MAKEDEV kivétel, mivel ennek segítségével hozhatók létre új bejegyzések
- Fájlrendszerek csatlakoztatása a `nodev` opcióval
  - megakadályozza a eszközállományok felismerését és használatát

## Rootkitek keresése

- Rootkitek, férgek, trójai programok és egyéb támadásokra utaló jelek után kutat
- Általában az első lépés lehet ha támadásra gyanakszunk
- Érdemes a használt verziót folyamatosan frissíteni
- Telepítése
  - Része a disztribúciónak
  - Letölthető a http://www.chkrootkit.org címről
    - Ellenőrizzük a letöltött állomány ellenőrző összegét
- Futtatása
  - root jogosultságokkal
  - A chkrootkit számos linux parancsot használ, ha ezek már kompromittálódtak, akkor hamis lehet a kimenet
  - `chkrootkit -p /mnt/cdrecorder`
    - futtatás megbízható bináris állományok felhasználásával egy nem írható médiumon

## Nyitott portok ellenőrzése

- Általában a támadások első lépése, előzzük meg a támadókat!
- Kiszolgáltatottságunk függhet:
  - A támadás kiindulási pontjától
    - Külső és belső támadások
    - Forrás IP címe, portja (címhamisítás)
  - A kommunikáció során érintett tűzfalak
    - Saját tűzfal(ak)
    - Szolgáltató tűzfala
  - A védett rendszer hálózati konfigurációja
    - Beérkező kapcsolatok belépési pontja
    - Engedélyezett beérkező kapcsolattípusok
- A hálózat tesztelése kívülről
  - Saját távoli felhasználói fiók felhasználásával
  - Egy tesztkörnyezet kialakításával és használatával

## Az `nmap` parancs

- Hatékony eszköz a hálózati biztonság tesztelésére
- A tesztelést körültekintően kell végezni
  - A rendszergazdák támadásnak értékelhetik
  - célszerű mindenkit értesíteni aki az adott kiszolgálóval dolgozik
  - Az `nmap` megsérti a hálózati protokollokat ez megzavarhatja az éles rendszereket
- Az információgyűjtés fázisai
  - Kiszolgálók felkutatása
    - Egy vagy több kiszolgáló letapogatása a hálózatban
  - Portletapogatás
    - A kapcsolatot fogadó, nyitott portok feltérképezése
  - Az operációs rendszer ujjlenyomatának vizsgálata
    - A hálózati viselkedés egyedi jellemzői alapján

## `nmap` alapműveletek

- TCP portok scannelése
  - nmap -v kiszolgáló
- UDP portok scannelése
  - nmap -v -sU kiszolgáló
- Operációs rendszer ujjlenyomat azonosítás
  - nmap -v -O kiszolgáló
- Kiszolgáló felkutatása egy tartományban
  - nmap -v -sP 192.168.1.100-150
- Kiszolgálók felkutatása egy ("C") hálózati osztályban
  - nmap -v kiszolgáló /24
  - nmap -v 192.168.1.0/24
  - nmap -v 192.168.1.0-255
  - nmap -v "10.12.104.*"

## Kiszolgálók felkutatása

- Csak a bekapcsolt gépekhez tartozó portokat kell letapogatni
  - A kiszolgálók felkutatása során az aktív állapot lekérdezése történhet
    - ICMP ping üzenetek segítségével
    - TCP ping üzenetek segítségével
  - Ha tűzfalak tiltják a fenti ping-eket, kikapcsoltnak tűnhet a gép
  - Ha biztosak vagyunk a bekapcsolt állapotban
    - nmap -P0 opcióval tilthatjuk a kiszolgáló felkutatását

## A `telnet` és `nc` utasítások

- Portok ellenőrzése telnettel
  - telnet c-ta-linux.ttk.pte.hu ssh
    - nyitott port
    - zárt port
    - tűzfallal védett port
- nc (netcat)
  - nc -z -vv kiszolgáló portok
    - nyitott port
    - zárt port
    - tűzfallal védett port
- nc6
  - nc6 --recv-only -vv kiszolgáló port

## A `netstat` program

- Összefoglaló információ a hálózatkezelés állapotáról
- Aktív hálózati kapcsolatok megjelenítése:
  - `netstat --inet`
  - gyanús lehet a sok `SYN_RECV` (portletapogatás)
- Az aktív kapcsolatok fogadására kész szerver socket-ek megjelenítése
  - `netstat --inet --listening`
  - Igyekezni kell minden socket feladatát tisztázni
  - A felesleges socketeket kikapcsolni
- Minden listázása
  - `netstat --all`

## Az `lsof` parancs

- Processzekhez tartozó nyitott állományok listázása
- Kapcsoló nélkül mindent listáz
- Nyitott hálózati kapcsolatok listázása
  - `lsof -i [TCP|UDP][@server][:port]`
  - Például
    - `lsof -i :ssh`
    - `lsof -i TCP`
    - `lsof -i TCP@iatt.ttk.pte.hu:22`
- IP címek és portszámok automatikus konverziója
  - Sok nyitott hálózati kapcsolat esetén lassú lehet
    - `-n` IP címek használata a kiszolgálónevek helyett
    - `-P` portszámok használata a szolgáltatás neve helyett
    - `-l` user ID-k user felhasználói nevek helyett

## Rendszerhívások követése

- Egy processz rendszerhívásainak követése
  - `strace -p PID`
    - megjeleníti a rendszerhívások paramétereit, visszaadott értékei és az esetleges hibákat
    - a processz és a kernel között átadott információkat
  - minden rendszerhívás követése nagyon sok információt jelent
- Rendszerhívások egy csoportjának figyelése
  - hálózati tevékenység figyelése
    - `strace -e trace=network`
  - Mivel a hálózati socketek gyakran végeznek írási és olvasási műveleteket:
    - `strace -e trace=network,read,write`
- Például
  - `strace -e trace=network,read,write -p 12345`

## Hálózati forgalom figyelése

- A hálózati interfészen megjelenő csomagok
  - unicast: egycímes csomagok, az adott gépnek címezve
  - multicast: többcímes csomagok, például videó vagy hanganyagok esetén
  - broadcast: csoportos csomagok, a hálózat minden gépe számára fontos információ vagy ha ismeretlen a cél
- Az egyes interfészekre nem csak a nekik szóló csomagok érkeznek
  - Normál módba másnak szóló csomagokat figyelmen kívül hagyja
  - Promiscuous (lehallgató) mód: a hálózat összes csomagját fogadja
    - rendszergazdaként kapcsolható mód
    - az átkapcsolás naplózásra kerül
- Kapcsolók és jelosztók (hub) …
- Útválasztók és átjárók …

## A tcpdump program

- Promiscuous mód be- és kikapcsolása
  - `ifconfig interfész promisc`
  - `ifconfig interfész -promisc`
- Forgalomfigyelés a tcpdump programmal
  - `tcpdump -w dumpfile [-c csomagok száma] [-i interfész] [-s hossz] [kifejezés]`
    - az interfész lehet `all` is
    - alapesetben a csomagoknak csak az első 68 bájtját menti el, a -s opcióval ez növelhető, 0-val a teljes csomag rögzíthető
- Rögzítendő csomagok szűkítése
  - capture filter (rögzítési szűrő) alkalmazása
  - Például
    - `tcpdump -w proba.dump host freemail.hu`
    - `tcpdump -w proba.dump tcp port telnet and host freemail.hu`

## A tcpdump program

- Elmentett nyomkövetési adatok megjelenítése
  - nem igényel root jogosultságot
    - normál állomány
    - a tartalma miatt érdemes lehet egyéb módon védeni
  - `tcpdump -r dumpfile`
- Elfogott csomagok megjelenítése közvetlenül
  - `-w` és `-r` nélkül
- Az elfogott csomagokat javasolt állományba menteni:
  - az elemzés gyakran az adatok különböző formában történő többszöri megjelenítését igényli
  - Régebbi adatok elemzése és összehasonlítása is szükséges lehet
  - Nem mindig tudhatjuk előre, hogy mi érdekel (szűrési feltételek)
  - A megjelenítés további erőforrásokat használ fel
  - Elkerülhető a további felesleges forgalom generálása
    - ssh-n keresztüli futtatáskor a megjelenítés további ssh forgalmat generál
    - IP címek DNS nevekké konvertálása további "felesleges" DNS kéréseket generál

## Az ngrep program

- karakterláncok keresésére a hálózati forgalomban
- Lehetőségei:
  - `ngrep [grep opciók] reguláris-kifejezés [szűrő]`
  - `ngrep -X hexa számok [szűrő]`
  - `ngrep -O állomány [-n számláló] [-d interfész] [-s hossz] reguláris-kifejezés [szűrő]`
  - `ngrep -I állománynév reguláris-kifejezés [szűrő]`
- Csak önálló csomagokban figyeli a minták illeszkedését
  - FTP esetén például jól használható
  - Telnet esetén nem igazán használható
- Egy lehetséges megoldás
  - forgalom rögzítése a tcpdump segítségével
  - a rögzített forgalom szűrése az ngrep használatával
  - szükség esetén a teljes rögzített tartalom átnézése az ethereallal

# Vezeték nélküli hálózatok biztonsága

Programtervező informatikus BSc

Számítógépes hálózatok biztonságtechnikája előadás

## WLAN biztonsági problémák

- Forgalomfigyelés
  - Titkosítatlan forgalom könnyen figyelhető (AirMagnet, AiroPeek)
  - Titkosított forgalom is lehallgatható és rögzíthető
  - Nem szükséges a fizikai jelenlét
- Jogosulatlan hozzáférés
  - Alapértelmezésben használt eszközök veszélyei
  - Engedély nélküli hozzáférési pontok telepítése későbbi támadási pontként
- ARP támadások (nem csak WLAN esetén)
  - ARP (Address Resolution Protocoll): IP-hez MAC címet rendel
  - Érvényes IP címhez hamis MAC cím adható
    - Az adott IP-re küldött csomagok a hamis MAC felé továbbítódnak
  - Megoldás lehet a Secure ARP

## WLAN biztonsági problémák

- Szolgáltatásmegtagadás (DoS)
  - Legegyszerűbb támadási forma, számolni kell a lehetőséggel
    - Másodlagos csatornák üzemeltetése
  - Elárasztás (használhatatlan csomagokkal)
  - Elnyomás (nagyobb teljesítményű rádiójellel)
  - Véletlen interferenciák
  - Kívülről érkező rádiójelek elleni védelem
    - Adók teljesítményének hangolás, a jel épületen belül tartása
    - Az épület árnyékolása (fémfólia alapú ablakszigetelés, fémszórt ablaküveg, fém alapú festékek, stb.)

## WEP (Wired Equivalent Privacy)

- Alapvető problémák
  - Könnyen lehallgatható kommunikációs csatorna
  - A hálózati hozzáférés nem igényel fizikai kapcsolatot
- A WEP megoldásai
  - Üzenetek titkosítása
  - A csatlakozó eszköz hitelesítése
- A WEP jellemzői
  - MAC rétegben implementált opcionális szabvány
  - Szimmetrikus kulcsú eljárás
  - 40 (64) bites és 104 (128) bites kulcsok

## A WEP működése

- WEP hitelesítés
  - A kliens hitelesítés kérést küld az AP felé
  - Az AP egy véletlen számot küld a kliensnek
  - A kliens a közös kulccsal titkosítja az értéket és visszaküldi
  - Az AP dekódolja az üzenetet
    - Ha az elküldött számot kapja vissza: hitelesíti a klienst
    - Ha más számot dekódol: visszautasítja a csatlakozás kérést
  - Az AP tájékoztatja a klienst az eredményről
- Sikeres hitelesítés esetén a további kommunikáció a már használt közös kulccsal kerül titkosításra

## A WEP működése

- A keretek törzsének és CRC részének titkosítása
- RC4 kulcsfolyam kódolás
  - Kulcssorozat előállítása
    - felhasználó által megadott WEP kulcsból (titkos)
    - 24 bites, minden keret küldése előtt véletlenszerűen változó inicializáló vektorból (IV)
  - Kódolás
    - Az adatkeret és a ICV érték kódolása a kulcssorozattal (XOR)
- Az IV szerepe
  - Minden üzenet más álvéletlen bitsorozattal legyen titkosítva
  - Egyébként két rejtjelezett üzenet lehallgatása esetén **ismert**
    - *Üzenet1* XOR *Kulcssorozat* és *Üzenet2* XOR Kulcssorozat
    - (U1 XOR K) XOR (U2 XOR K) = **U1 XOR U2**

## A WEP problémái

- A kulcsok problémái
  - Közös, statikus jellegű, nehezen változtatható kulcsok
  - Ritkán kerül megváltoztatásra (akár évek is)
  - Csak a külső támadó ellen véd, egymás üzenetei megfejthetők
  - Csak elvben van lehetőség egyedi kulcspárok használatára
- A hitelesítési problémái
  - Egyirányú hitelesítés
  - Azonos kulcs a hitelesítéshez és a titkosításhoz
  - Hitelesítés csak a csatlakozáskor
  - A hitelesítés teljes folyamata lehallgatható

## A WEP problémái

- Integritás ellenőrzés
  - A titkosított adatok megváltoztatása után a titkosított ICV is kiszámolható a kulcs ismerete nélkül
  - Nincs módszer a visszajátszott üzenetek detektálására
- Titkosítás
  - Rövid (24 bites) inicializáló vektorok (kb.: 17 millió lehetőség)
  - Több adatcsomagnál előfordulhatnak azonos IV-k (IV ütközés)
  - Több eszköz egyidejű indulásakor azonnal lehet IV ütközés
  - Gyenge RC4 kulcsok → nem teljesen véletlen előállított bitsorozat
  - Mindezeket kihasználva a WEP kulcsok megfejthetők
- Csak minimális biztonság elérésének céljából alkalmazható
  - Jellemzően otthoni hálózatokban megfelelő lehet
  - Sokkal jobb megoldás a nyílt hálózatoknál …

## Támadás a WEP ellen

- Keressünk egy ARP kérésnek látszó csomagot
  - ARP: IP-hez tartozó MAC címet keres
  - A forrás és a cél MAC címe nincs titkosítva a WEP-nél
- Játszuk vissza az ARP kérést több alkalommal
- Az IP-hez tartozó hoszt minden kérdésre válaszol
  - Különböző IV-vel rendelkező üzenetekkel
  - Azonos tartalommal
- Szükséges:
  - Megfelelő WLAN interfész(ek)
  - Megfelelő illesztőprogramok

## WLAN titkosítás - TKIP

- TKIP (Temporal Key Integrity Protocol)
- Elégséges az eszközöz szoftverének frissítése
  - Megmarad a WEP kompatibilitás
- Cél a WEP hibáinak javítása
  - 48 bites IV használata
  - Védelem az üzenetek visszajátszása ellen
    - Az IV egyben sorszámot is jelent
    - Minden üzenetnél növelni kell az IV értékét (WEP-nél nem kötelező)
- A TKIP folyamat
  - 128 bites ideiglenes kulcs megosztás a kliens és az AP között
  - Az ideiglenes kulcs és a kliens MAC címének keverése
- Az ideiglenes kulcs 10.000 csomagonként változik
  - Minden üzenethez egyedi kulcs tartozik

## WLAN titkosítás - WPA

- WPA (Wi-Fi Protocol Access)
- A WEP továbbfejlesztése
  - Periodikusan változó titkosító kulcsok
    - Kulcsok változtatása a TKIP segítségével
  - Kölcsönös hitelesítési lehetőségek
  - Autentikációs szerverek (RADIUS) bevonásának lehetősége
  - 48 bites IV
- Átmeneti megoldás
  - A WPA a 802.11i egy részhalmaza
  - Végleges megoldás 802.11i szabványban
  - WPA2: a 802.11i szabvány véglegesítése után készült el
    - Komplexebb, robosztusabb megoldás
    - AES (Advanced Encryption Standard) használata a TKIP helyett

## WLAN hitelesítés - IEEE 802.1x

- EAP (Extensible Authentication Protocol) használata
- A kliens hitelesítés iránti kérelmet küld
- Az AP egy azonosító iránti kérést tartalmazó EAP üzenettel válaszol.
- A kliens az AP egyetlen nyitott portján továbbítja az azonosítást tartalmazó választ a hitelesítést végző szerverhez
  - Minden más portot lezárva tart a kliens előtt
- A hitelesítő szerver elutasító vagy elfogadó üzenetet küld a bázisállomásnak
- A bázisállomás sikeres hitelesítést jelző AEP csomagot küld a kliensnek
- Ha minden sikeres volt, a bázisállomás elérhetővé teszi a további engedélyezett portokat is a kliensnek

## WLAN beállítások - SSID

- SSID (Service Set IDentifier)
- Az AP olyan jelzőkereteket sugároznak amik tartalmazzák az SSID-t
- Csatlakozás akkor történhet ha az AP SSID-je egyezik a kliensen beállított SSID-vel
- SSID elrejtésének lehetősége
  - Az SSID azonosító a kommunikáció során titkosítás nélküli
  - Kliens eszközök csatlakozási kérelmének lehallgatása
- Nem eredményezi a biztonság jelentős növekedését

## WLAN beállítások - MAC filter

- A hozzáférési pontok általában támogatják a MAC szűrést
- Az AP megvizsgálja a beérkező kereteket
  - Ha a forrás MAC címe engedélyezve van továbbítja a keretet
  - Ha a forrás MAC címe nincs engedélyezve elutasítja a keretet
- A MAC címek az átvitel során nem titkosítottak
- Egy megszerzett MAC cím beállítható más eszközökön is, így álcázhatja magát a támadó
- Nehézkes a MAC címek adatbázisának kezelése is

## A WLAN biztonság fokozása

- A WLAN felhasználóinak elhelyezése a tűzfalon kívül, DMZ-ben
- Hatékony titkosítás alkalmazása
- A firmware-ek rendszeres frissítése
- A hozzáférési pontok fizikai rögzítése, elrejtése, reset gomb (!)
- Elérés tiltása üzemszünet esetén
- Bonyolult jelszavak megadása az AP-khez
- SSID broadcast tiltása
- Rádióhullámok terjedésének szabályozása
- Kliensek védelme személyes tűzfallal
- Bázisállomások forgalmának monitorozása, csaló AP-k kizárása
- Telepítések, fejlesztések felügyelete
- Hozzáférési pontok és kliensek MAC címének nyilvántartása