Ibiza vizsgajegyzet

Alapok:

Alice, üzenet küldő

Bob, üzenet fogadó

Charlie, a kommunikáció harmadik résztvevője

Darth, a kommunikáció negyedik résztvevője

Mallory, rosszindulatú támadó, tudja módosítani az üzenetet

Eve, hallgatózó, passzív támadó, nem tudja módosítani az üzenetet

1. Fólia

Informatikai biztoság alapfogalmai:

Biztonsági sérülés:

* Alice az alkalmazottak fizetési jegyzékét küldi el Bobnak. Eve, aki nem jogosult ezen adatok hozzáférésére, figyeli az adatátvitelt és készít magának egy másolatot.
* Alice szeretne könyvtárához teljes jogosultságot adni az új felhasználóknak. Ehhez "üzenetet küld" a gépnek (Bobnak), hogy módosítsa a jogosultság kezelő állományt. Mallory figyeli a kommunikációt és módosítja az üzenetet, jogosulatlanul hozzáadja saját adatait.
* Alice üzenetet küld brókerének (Bobnak), hogy vásároljon bizonyos értékpapírokat. Időközben, a megadott részvények veszítettek értékükből. Alice letagadja, hogy valaha is ilyen üzenetet küldött volna Bobnak.

Informatikai biztonság fogalma:

Az informatikai rendszer olyan kielégítő állapota, amely az informatikai rendszerben tárolt adatok bizalmassága, sértettsége és rendelkezésre állása, illetve rendszerelemek sértettsége és rendelkezésre állása szempontjából zárt, teljes körű, folytonos és a kockázatokkal arányos.

* zárt védelem: olyan védelem, amely az összes releváns fenyegetést figyelembe veszi
* teljes körű: a védelmi intézkedések a rendszer összes elemére kiterjednek
* folytonosság: a védelem az időben változó körülmények és viszonyok ellenére is folyamatosan megvalósul
* kockázatokkal arányosság: egy kellően nagy intervallumban a védelem költségei arányosak a potenciális kárértékkel

Információbiztonság (information security):

az informatikai biztonságnál szélesebb, szóban, írásban, az informatikai rendszerekben, vagy bármilyen más módon való információ közlés védelmére vonatkozik

* Információbiztonsági intézkedések:
  + Adatvédelem (data protection): a személyes adatok jogszerű kezelése, az érintett személyek védelmét biztosító alapelvek, szabályok, eljárások, adatkezelési eszközök és módszerek összessége. (NAIH - Nemzeti Adatvédelmi és Információszabadság Hatóság)
  + Adatbiztonság (data security): Az informatikai biztonság részhalmaza, az adatok jogosulatlan megszerzése, módosítása és megsemmisítése elleni műszaki és szervezési megoldások rendszere. (NAIH)

Biztonsági célok:

CIA hármas:

Bizalmasság (Confidentiality):

* Titkos vagy személyes információk (privacy) nem kerülhetnek jogosulatlanok kezébe. A bizalmasságot az adatok tárolásánál, feldolgozásánál és továbbításánál is garantálni kell.

Sértetlenség (Integrity):

* Adatintegritás (data integrity): Teljesülésekor az adat jogosulatlanul nem módosult tárolása, feldolgozása vagy küldése során.
* Rendszer sértetlensége (system integrity): A rendszer működése az elvártnak megfelelő, jogosulatlan módosításoktól mentes.

Rendelkezésre állás (Availability):

* Biztosítja, hogy a szolgáltatás az arra jogosultak számára a szükséges időben és időtartamra használható.

További célok:

Nyomonkövethetőség (Accountability):

* Az a tulajdonság, hogy egy entitás által végrehajtott tevékenység visszakövethető legyen az entitáshoz. A tevékenységek ellenőrzés céljára rögzítésre kerülnek azért, hogy visszakövethetőek legyenek, bizonyíték álljon rendelkezésre. Ez a tulajdonság lehetővé teszi a letagadhatatlanságot (non-repudiation), a behatolások (intrusion) detektálását, megelőzését.

Garancia, biztosíték (Assurance):

* A bizalom abban, hogy a négy másik biztonsági célt (bizalmasság, sértetlenség, rendelkezésre állás, nyomonkövethetőség) a biztonsági alrendszer megfelelően ellátja/eléri.

A képen szöveg, képernyőkép, diagram, sor látható

Automatikusan generált leírás

Fogalmak közötti kapcsolatok:

Bizalmasság függ a sértetlenségtől:

* Ha a rendszer sértetlensége sérült, akkor nem várható el, hogy a bizalmas adatok titokban maradtak.

Sértetlenség függ a bizalmasságtól:

* Ha bizonyos információ (pl. jelszó) bizalmassága sérül, a sértetlenséget biztosító mechanizmusok kikerülhetőek.

Rendelkezésre állás és számonkérhetőség függ a bizalmasságtól:

* Ha bizonyos információ (pl. jelszó) bizalmassága sérül, a rendelkezésre állást és számonkérhetőséget biztosító mechanizmusok kikerülhetőek.

Rendelkezésre állás és számonkérhetőség függ a sértetlenségtől:

* Ha a rendszer sértetlensége sérült, akkor a rendelkezésre állást és számonkérhetőséget biztosító mechanizmusok szabályos működése is sérülhetett.

Hitelesség:

Valaminek a forrása az, amit megjelöltek, és a tartalma az eredeti.

* Felhasználó hitelesítése (Entity Authentication):
  + Az a folyamat, amikor egy entitás meggyőződik egy másik entitás identitásáról.
* Üzenet hitelesítési kód (Message Authentication Code):
  + Egy rövid, fix hosszúságú érték, mely lehetővé teszi az üzenet sértetlenségének és forrásának ellenőrzését, de nem biztosítja a letagadhatatlanságot.

Sérülési szintek:

Három szintet különböztetünk meg attól függően, hogy milyen fokú sérülés történik szervezeti, illetve személyi szinten.

* Alacsony:
  + A sérülésnek csak korlátozott hatása van a szervezet, illetve a személy tulajdonára. A szervezet képes legfőbb feladatait végrehajtani, csak minimális kár keletkezik vagyontárgyaiban, bevétele csak minimális csökken, minimális személyi sérülés történik.
  + Példa: Egyetemi kurzusokra regisztrált hallgatók névsora
* Közepes:
  + A sérülésnek komoly hatása van a szervezet, személy tevékenységére, tulajdonára. A sérülés jelentős kárt, bevétel csökkenést okoz, a szervezet még képes ellátni legfőbb feladatát, de hatásfoka jelentősen csökken.
  + Példa: Hallgatók beiratkozási információi.
* Magas:
  + A sérülésnek végzetes hatása van a szervezet, személy tevékenységére, tulajdonára. Katasztrofális kár történik a vagyontárgyakban, tragikus költséget, személyi sérülést okoz, a szervezet nem képes ellátni feladait.
  + Példa: Hallgatók érdemjegyeinek listája.

Sértettség példa:

* Alacsony: Sok honlap lehetővé teszi, hogy ügyfelei leadhassák szavazatukat, véleményüket. Általában az így kapott eredmények csak irányt mutatóak. A rendszer implementálásánál nem alkalmaznak komoly védelmi mechanizmusokat.
* Közepes: Egy honlap lehetővé teszi, hogy a regisztrált felhasználók fórumon megvitassanak különböző témákat. Akár egy regisztrált felhasználó, akár egy külső támadó módosítja a bejegyzéseket. Ha a fórum szórakoztató jellegű, akkor pénzügyi kiesést (pl. reklám bevétel), adatvesztést jelenthet.
* Magas: Orvosi adatbázis bejegyzéseit egy arra nem jogosult alkalmazott (pl. ápoló) módosítja. A valótlan adatok akár tragikus, életveszélyes helyzetet is eredményezhet.

Rendelkezésre állás:

* Alacsony: Webes telefonkönyv elérhetetlensége idegesítő lehet, de van más út egy telefonszám megszerzésére.
* Közepes: Egy alapítványi iskola honlapja információkat ad a támogatás módjáról. Ha a honlap nem elérhető, akkor a bevételük csökkenhet.
* Magas: Tekintsünk egy egészségügyi adatokhoz való hozzáférést vezérlő rendszert. Amennyiben ez a rendszer nem elérhető, akkor az orvosok nem jutnak hozzá az adatokhoz, mely akár a betegek életét is veszélyeztetheti.

Hálózati kommunikáció védelme:

OSI (Open Systems Interconnections) biztonsági struktúra (X.800 szabvány):

* Biztonsági támadások: Bármilyen tett, mely az információ biztonságát veszélyezteti.
* Biztonsági mechanizmusok: Eljárások, melyeket arra terveztek, hogy detektálja, megelőzze a támadást, vagy kijavítsa a támadás okozta kárt.
* Biztonsági szolgáltatások: Szolgáltatások, melyek védik az adatátviteli és feldolgozó rendszerek biztonságát. Egy biztonsági szolgáltatás egy vagy több biztonsági mechanizmust használ.

Fő kategóriák:

Passzív támadás:

* Az adatátvitel lehallgatása, monitorozása.
* Cél: A továbbított adat megszerzése. Nehéz észrevenni, hiszen az átküldött adat nem módosul. (Üzenet tartalmának megszerzése, Forgalom elemzése)

Aktív támadás:

* Az adatfolyam módosítása, vagy egy hamis generálása. Nehéz teljes mértékben megelőzni.
* Cél: A támadás detektálása és az okozott kár korrigálása. (Megszemélyesítés, Üzenet visszajátszása, Adat módosítása, Terheléses támadás)

Támadások:

* Üzenet tartalmának megszerzése: A támadó lehallgatja a csatornát és szeretné tudni az elküldött üzenet (pl. e-mail, átküldött állomány, telefonbeszélgetés) tartalmát.
* Forgalom elemzése: Tegyük fel, hogy a támadó a lehallgatott üzenetekből nem tudta megszerezni az információt. A támadó megfigyeli ezen üzenetek formátumát, meghatározhatja a kommunikáló felek helyét, kilétét, az üzenetek hosszát, gyakoriságát. Ezen információk alapján kitalálhatja egy üzenet tartalmát.

A képen vázlat, diagram, tervezés látható

Automatikusan generált leírás

* Megszemélyesítés: A támadó eljátsza a legális fél szerepét. (2) Általában ehhez a támadáshoz szükséges valamely másik aktív támadás (pl. üzenet visszajátszása).
* Üzenet visszajátszása: A támadó lehallgatja az üzenetet, majd újra elküldi (1, 2, 3).
* Adat módosítása: A támadó az üzenetek valamely részét vagy sorrendjét megváltoztatja (1, 2).
* Túlterheléses támadás: A támadás eredményeképpen a rendszer nagyon lelassul, elérhetetlenné válik, esetleg össze is omolhat (3).

A képen vázlat, tervezés, illusztráció, képregény látható

Automatikusan generált leírás

Kapcsolatok:

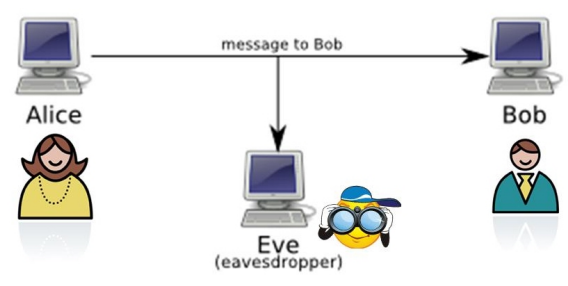
A képen szöveg, képernyőkép, nyugta, szám látható

Automatikusan generált leírás

2. Fólia

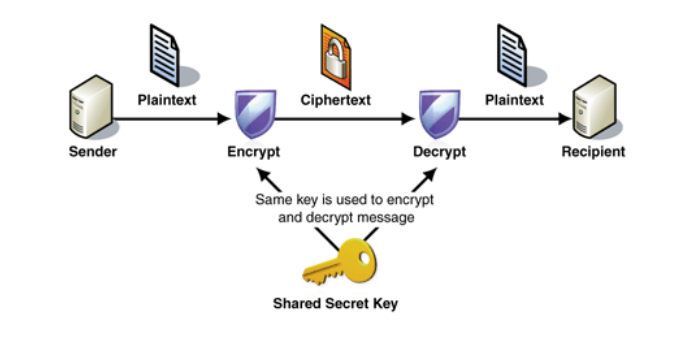
Titkosítási sémák:

Alice szeretne egy bizalmas üzenetet küldeni Bobnak a nyílt csatornán.

* P (plaintext space): nyílt üzenetek véges halmaza
* C (ciphertext space): titkosított üzenetek véges halmaza
* K (key space): a lehetséges kulcsok véges halmaza
* m ∈ P (plaintext): nyílt üzenet
* c ∈ C (ciphertext): titkosított üzenet

Szimmetrikus titkosítási séma:

A titkosító és visszafejtő kulcs megegyezik, vagy a visszafejtő a titkosító kulcsból könnyen (polinomiális időn belül) kiszámítható.



Definíció:

A SE = (Key, Enc, Dec) hármas egy szimmetrikus titoksítási séma, ha

* Key: kulcsgeneráló algoritmus, mely egy k biztonsági paraméterhez (kulcs méretére utal) megad egy K ∈ K titkos kulcsot.
* Enc: tiktosító algoritmus, mely ∀m ∈ P nyílt üzenethez és ∀K ∈ K titkos kulcshoz generál egy c ∈ C titkosított üzenetet.   
  c = EncK (m)
* Dec: visszafejtő algoritmus, mely egy c ∈ C titkosított üzenethez és egy adott K ∈ K kulcshoz megad egy m ∈ P nyílt üzenetet.   
  m = DecK (c)

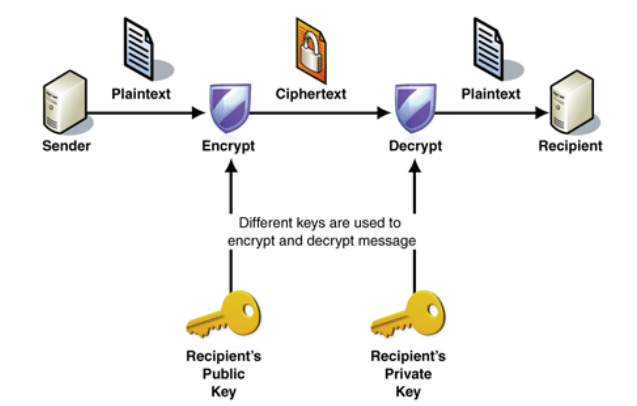
Sok esetben a tikosítási algoritmus inputja egy r véletlen is. Így ez randomizálható.

A visszafejtő algoritmus determinisztikus.

Korrekt visszafejtést biztosít, ha ha ∀m ∈ P és ∀K ∈ K esetén DecK (EncK (m)) = m.

Aszimmetrikus titkosítási séma:

A titkosító és visszafejtő kulcs különbözik olyannyira, hogy a visszafejtő a titkosító kulcsból csak nehezen (nem ismerünk rá polinomiális idejű algoritmust) számítható ki.



Definíció:

A AE = (Key, Enc, Dec) hármas egy aszimmetrikus titkosítási séma, ha

* Key: kulcsgeneráló algoritmusok, mely egy k biztonsági paraméterhez (kulcs méretére utal) megad egy (PK, SK) ∈ K nyilvános és titok kulcsból álló párt.
* Enc: titkosító algoritmus, mely ∀m ∈ P nyílt üzenethez és PK nyilvános kulcshoz generál egy c ∈ C titkosított üzenetet.   
  c = EncPK (m)
* Dec: visszafejtő algoritmus, mely egy c ∈ C titkosított üzenethez és egy adott SK kulcshoz megad egy m ∈ P nyílt üzenetet.   
  m = DecSK (c)

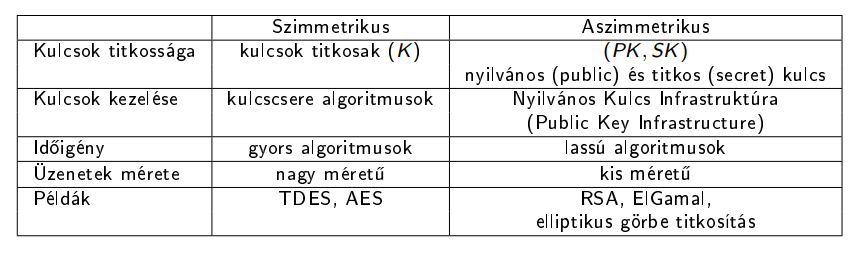
Sok esetben a tikosítási algoritmus inputja egy r véletlen is. Így ez randomizálható.

A visszafejtő algoritmus determinisztikus.

A kulcsgeneráló algoritmus outputja meghatározza a P, C, K halmazokat.

Korrekt visszafejtést biztosít, ha ∀m ∈ P és ∀(PK, SK) ∈ K esetén DecSK (EncPK (m)) = m.

Összehasonlítás:



Passzív támadások:

Cél:

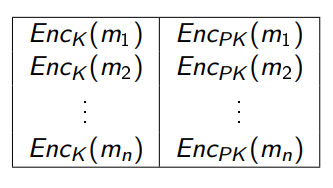
* A titkos visszafejtő kulcs megszerzése.
* Egy adott titkosított üzenethez tartozó nyílt üzenet megszerzése.

Módok:

* Csak a titkosított üzenet ismert (Ciphertext Only Attack)
* Ismert nyílt üzenet alapú támadás (Known Plaintext Attack)
* Választott nyílt üzenet alapú (Chosen Plaintext Attack)
  + Nem alkalmazkodó (Non-adaptive)
  + Alkalmazkodó (Adaptive) (aktív támadás)
* Választott titkosított üzenet alapú (Chosen Ciphertext Attack)
  + Nem alkalmazkodó (Non-adaptive)
  + Alkalmazkodó (Adaptive) (aktív támadás)

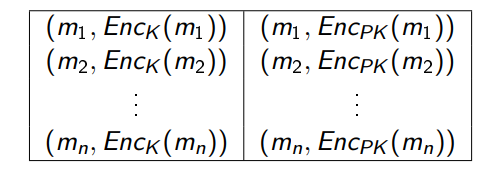
COA: Csak a tikosított üzenet ismert (Ciphertext Only Attack):

A támadó rendelkezésére áll egy ugyanazon kulccsal titkosított üzenetekből álló lista:



KPA: Ismert nyílt üzenet alapú támadás (Known Plaintext Attack):

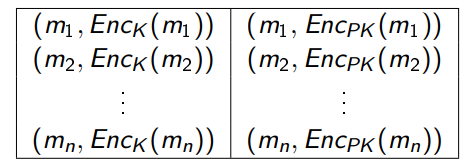
A támadó rendelkezésére áll egy ugyanazon kulccsal titkosított nyílt és titkosított üzenetekből álló lista:



A cél vagy a titkos kulcs, vagy a listán nem szereplő titkosított üzenethez tartozó nyílt üzenet megszerzése.

CPA: Választott nyílt üzenet alapú (Chosen Plaintext Attack):

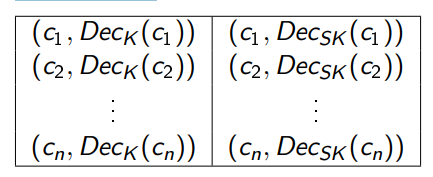
A támadó rendelkezésére áll egy ugyanazon kulccsal titkosított nyílt és titkosított üzenetekből álló lista, ahol a nyílt üzenetek a támadó által választottak:



Nem alkalmazkodó esetben a támadó előre kiválasztja a nyílt üzeneteket, míg alkalmazkodó esetben a kapott titkosított üzenetek alapján választja ki a következő nyílt üzenetet. A cél vagy a titkos kulcs, vagy a listán nem szereplő titkosított üzenethez tartozó nyílt üzenet megszerzése.

CCA: Választott titkosított üzenet alapú (Chosen Ciphertext Attack):

A támadó rendelkezésére áll egy ugyanazon kulccsal titkosított nyílt és titkosított üzenetekből álló lista, ahol a titkosított üzenetek a támadó által választottak:

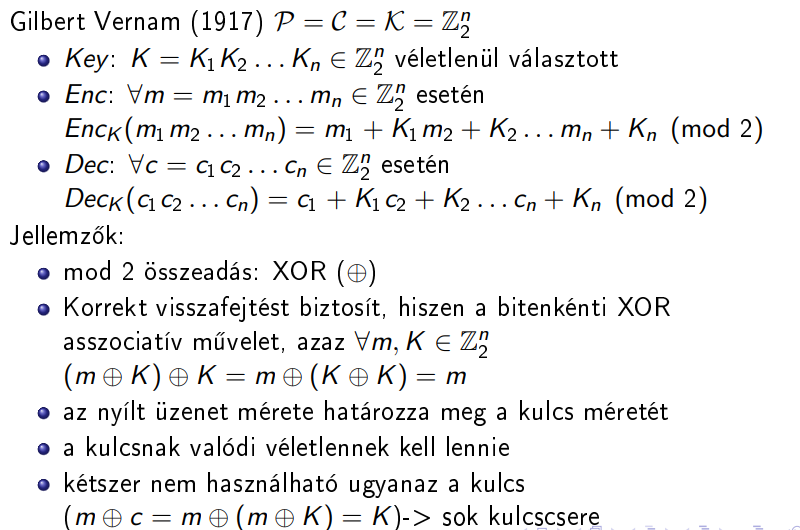


Nem alkalmazkodó esetben a támadó előre kiválasztja a titkosított üzeneteket, míg alkalmazkodó esetben a kapott nyílt üzenetek alapján választja ki a következő titkosított üzenetet. A cél vagy a titkos kulcs, vagy a listán nem szereplő titkosított üzenethez tartozó nyílt üzenet megszerzése.

Biztonsági kérdések:

* Feltétel nélküli biztonság: A támadó korlátlan számítási kapacitással rendelkezik. Nehéz a gyakorlatban megvalósítani.
* Feltételes biztonság: A támadó korlátos számítási kapacitással rendelkezik, polinom idejű algoritmusokat használ.
* Kerckhoff-elv: Azaz a biztonság egyedül a kulcsnak, és nem magának az algoritmusnak a titkosságán alapuljon. Feltesszük, hogy a támadó a rendszert ismeri. Mert:
  + tömeges méretű alkalmazásoknál úgy sem lehetne az algoritmust titokban tartani
  + az algoritmus az implementációkból visszafejthető
  + egy nyilvános, tesztelt módszer nagyobb bizalmat érdemel, mint egy soha nem látott "szupertitkos"

Vernam One Time Pad:



3. Fólia

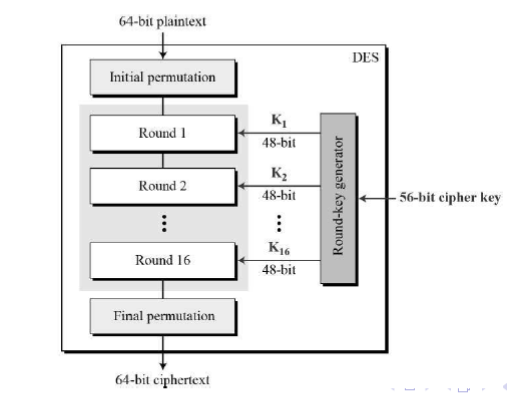
Data Encrytion Satandard (DES):

SE = (Key; Enc; Dec) szimmetrikus titkosítási séma

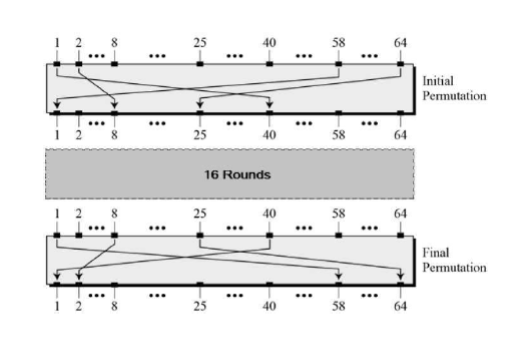
P = C = {0,1}64, K ={0,1}56+8, minden K ∈ K 56 véletlen bitből és 8 paritás bitből áll

Key: K ∈ K véletlen választása

Enc:

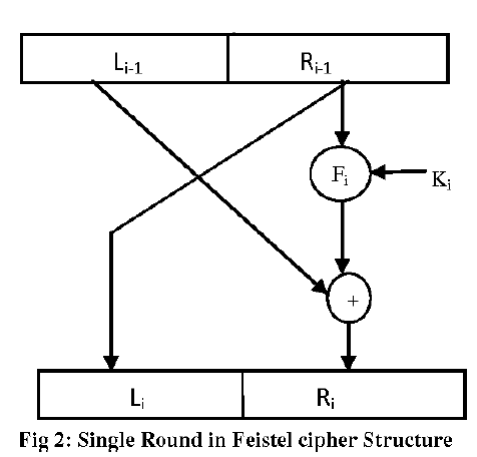


Az első lépés egy kulcsfüggetlen permutáció (IP) a 64 bites bementen. Az utolsó lépés ennek pontosan az inverze (IP-1).



Fiestel struktúra:

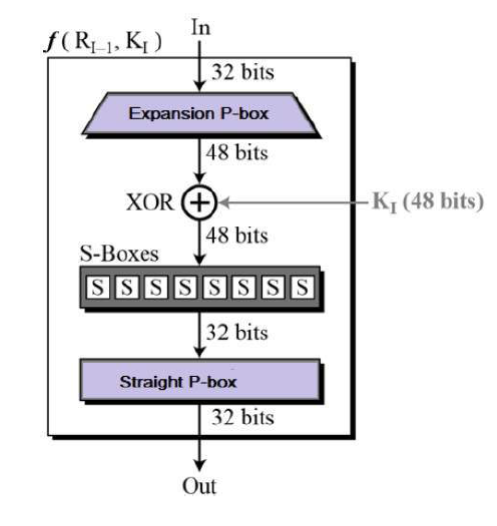
* A két permutáció (IP, IP-1) között 16 kör (Round) hajtódik végre, minden körben a Fiestel struktúra fut le.



* Minden kör két 32 bites bemenetből ugyancsak kettő 32 bites kimenetet produkál.
* A bal oldali kimenet egyszerűen a jobb oldali bemenet másolata (Li = Ri-1)
* A jobb oldali kimenetet kizáró vagy (xor) művelettel kapjuk, amit egyrészt a bal oldali bemenet, másrészt a jobb oldali bemenet, valamint az adott körhöz tartozó kulcsérték alapján egy adott f függvénnyel képzett érték között végzünk el. (Ri = Li-1 ⊕ f (Ri-1; Ki))

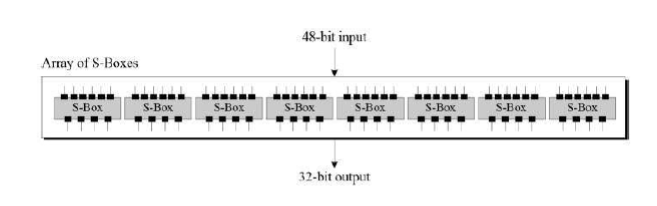
Belső függvény:

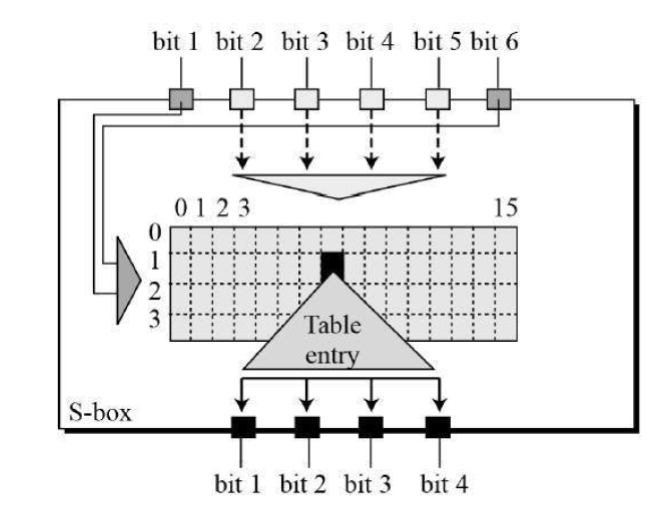
* Első lépésben egy 48 bites számot képzünk a 32 bites jobb oldal kiterjesztésével.
* A második lépésben az E kimenete és a Ki bitjei között kizáró vagy műveletet hajtunk végre.
* Az így kapott eredményt 8 db 6 bites csoportra osztjuk, amiket aztán különböző S-dobozokba pumpálunk.
* Az egyes S-dobozok 4 bites kimenetet generálnak.
* Végül az így nyert 32 bitet egy P-dobozon engedjük keresztül.



S-boxok alkalmazása:

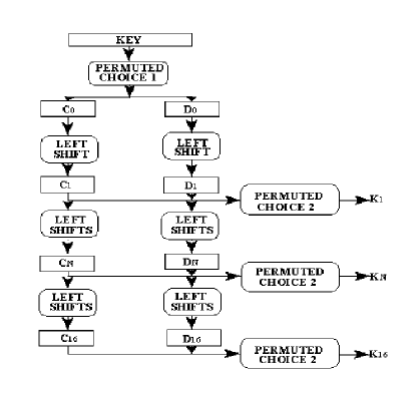
Minden S-doboz 4 sorból és 16 oszlopból álló táblázat. A bemenet két szélső bitje (1. és 6.) címzi meg a sort, és a középső 4 bitje (2-5. bit) címzi meg az oszlopot. Az így meghatározott cella tartalma az S-doboz kimenete, ami 4 biten ábrázolható.





Körkulcsok generálása:

* Mind a 16 iterációs lépésben különböző körkulcsokat használunk. Az algoritmus kezdetekor egy 56 bites permutációt végzünk a kulcson.
* A kulcsot két 28 bites részre particionáljuk, mindkettőt az iteráció sorszámának megfelelő számú bittel balra forgatjuk.
* A 56 bites kulcs egy 48 bites részét minden fokozatban még külön permutáljuk.



Visszafejtés:

* A visszafejtést és a titkosítást ugyanazzal a kulccsal és körkulcsokkal végezzük.
* A visszafejtő algoritmus megegyezik a titkosító algoritmussal, csak a körkulcsok alkalmazásának sorrendje más.
* Visszafejtésnél K16, K15, …, K1 sorrendet alkalmazunk.

3DES:

* Az IBM már 1979-ben felismerte, hogy a DES-kulcs túlságosan rövid (sikeres brute force támadás 1999-ben), és a biztonság növelésének érdekében kidolgoztak egy háromszoros titkosítást használó eljárást.
* A 3DES algoritmus a DES titkosító vagy visszafejtő algoritmusát hajtja végre háromszor. Az EDE (Encryption-Decryption-Encryption) algoritmus az elterjedt.
* EDE: Három kulcsot és három fokozatot használnak. Az első lépcsőben a nyílt üzenetet a szokott módon a K1 kulccsal titkosítjuk. Második lépésben visszafejtést végzünk, melyhez a K2-t használjuk kulcsként. Végül az így kapott eredményt K3 kulccsal titkosítjuk.

Advanced Encryption Standard (AES):

Rijndael: Joan Daemen, Vincent Rijmen

SE = (Key, Enc, Dec) szimmetrikus titkosítási séma.

P = {0,1}128

C = {0,1}128

K = {0,1}128, k ∈ {128,192, 256}

Key: véletlenül választunk egy K ∈ K

Egy kör: SubByte, ShiftRow, MinColumn, AddRoundkey

Körök száma: k=128,10;

k=196, 12;

k=256, 14.

!ANIMÁCIÓ!

4.Fólia

RSA:

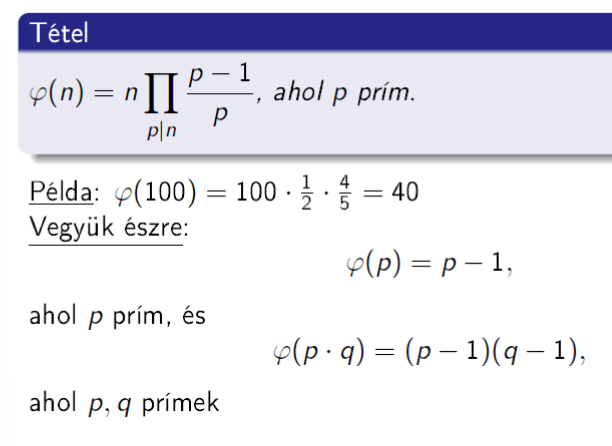
1977-ben jelent meg. Ron Rivest, Adi Shamir, és Leonard Adlema tervezték. legtöbb Nyilvános Kulcs Infrastuktúra (PKI) termékben megtalálható, SSL/TLS tanusítvánayok. Biztonságos email: PGP, Outlook.

Kongurancia:

* Legyenek a és b egész számok és m pozitív egész. Azt mondjuk, hogy a kongruens b-vel modulo m, ha m|a - b.
* Jelölés: a ≡ b (mod m)
* Az m számot modulusnak nevezzük.
* Két szám pontosan akkor kongruens modulo m, ha m-mel osztva ugyanazt a maradékot adják.
* Amennyiben a és b nem kongruensek modulo m, akkor a és b inkongruensek modulo m.

Euler-féle φ függvény:

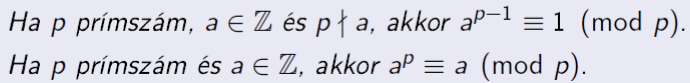
* Az a1, a2, …, an számok relatív prímek, ha nincs egységtől különböző közös osztójuk, azaz (a1, a2, …, an) = 1.
* Tetszőleges n pozitív egész esetén φ(n) jelöli az 1, 2, …, n számok közül az n-hez relatív prímek számát.



Euler-Fermat tétel:

* Ha (a, m) = 1, akkor aφ(m) ≡ 1 (mod m).

kis Fermat tétel:



Lineáris kongurencia:

* Ha a, b ∈ Z és m pozitív egész, akkor az ax ≡ b (mod m) kongurenciát lineáris kongurenciának nevezzük.
* A kongruencia megoldásai olyan egész számok, melyeket x helyébe írva a kongruencia teljesül. Megoldások számán a különböző maradékosztályok számát értjük, melyekből vett egészek a kongruenciát kielégítik.
* Ha a, b ∈ Z és m pozitív egész, az ax ≡ b (mod m) lineáris kongurencia akkor és csak akkor oldható meg ha (a, m) | b.
* Ha az ax ≡ b (mod m) lineáris kongurencia megoldható, akkor a megoldások száma (a,m).
* Ha (a,m) = 1, akkor az ax ≡ b (mod m) lineáris kongruencia egyetlen megoldása x ≡ aφ(m) \*b (mod m).

Szimultán kongurenciarendszer:

* Ha ugyanazon ismeretlenre több különböző modulusú kongruenciafeltételt adunk, akkor szimultán kongruenciarendszert kapunk.
* Az x ≡ y1 (mod m), x ≡ y2 (mod n) szimultán kongruenciarendszer megoldhatóságának szükséges és elégséges feltétele, hogy (m, n) j y1 - y2. Az összes megoldás egy maradékosztályt alkot modulo [m, n] (a modulusok legkisebb közös többszöröse).

Kínai maradéktétel:

* Legyenek m1, m2, …, mk páronként relatív prímek. Ekkor x ≡ y1 (mod m1), x ≡ y2 (mod m2), …, x ≡ yk (mod mk) szimultán kongruenciarendszer bármilyen y1, y2, …, yk egészek esetén megoldható, és a megoldások egyetlen maradékosztályt alkotnak modulo m1, m2, …, mk.

Titkosítási séma:

Aszimmetrikus titkosítási séma: AE = (Key; Enc; Dec)

Key:

* Véletlenül választunk két nagy prímet: p; q. (Miller-Rabin primteszt)
* Kiszámítjuk az RSA modulust: n = p \* q.
* Kiszámoljuk az Euler-féle φ függvényt: φ(n) = (p-1)(q-1).
* Választunk egy véletlen e egészt: 1< e < φ(n) és (e, φ(n)) = 1. e a titkos kitevő. (Euklédeszi algoritmus)
* Kiszámítjuk d-t: 1<d< φ(n) és ed ≡ 1 (mod φ(n)). d a visszafejtő kitevő. (Kibővített Euklédeszi algoritmus)

PK = (n, e); SK = d and φ(n),p,q titkos paraméterek

P = C = Zn

EncPK(m) = me (mod n) ∀m ∈ P és PK = (n,e) mellett. (Gyorshatványozás)

DecSK(c) = cd (mod n) ∀m ∈ C és SK = d mellett. (Kínai maradéktétel)

Biztonsági elemzés:

SK kiszámítása PK ismeretében nehéz, 🡪 prímfaktorizáció:

* A támadó célja a titkos kulcs megszerzése.
* A d exponens kiszámítása az (n, e) paraméterek ismeretében ugyanolyan nehéz, mint az n modulus p és q prímfaktorainak meghatározása, ahol |p| = |q|.
* Ha meg tudjuk határozni n faktorait, akkor d kiszámítható a ed ≡ 1 (mod φ(n)) kongurenciából, φ(n) kiszámításához p és q ismerete szükséges.
* Ha van egy hatékony algoritmus d kiszámítására (n,e) ismeretében, akkor az algoritmus alkalmas n faktorizálására.
* Ha az n modulus p és q faktorai elég nagyok, akkor nem ismerünk hatékony (polinomiális idejű) algoritmust n faktorizálására.

A nyílt üzenet m kiszámítása a c ismeretében nehéz 🡪 RSA probléma:

* Adott (n, e) RSA nyilvános kulcs és c ≡ me (mod n) titkosított üzenet mellett m nyílt üzenet kiszámítása.
* A támadó nem feltétlenül ismeri a titkos kulcsot. Feladat: c^1/e (mod n) kiszámítása.
* Az RSA Probléma nehéz, ha az n modulus elég nagy és a prímek véletlenül generáltak, valamint az m nyílt üzenet (emiatt a c titkosított üzenet is) egy a 0 és n - 1 közé eső véletlen egész.
* Az m nyílt üzenet véletlensége a [0, n - 1] intervallum felett fontos feltétel. Ha m egy kis halmazból vett, akkor a támadó könnyen meghatározhatja m-et úgy, hogy egyenként
* próbálgatja az összes lehetséges m-et (brute force).

Prímfaktorizáció vs. RSA probléma:

* Prímfaktorizáció -> RSA probléma:
  + Az RSA Probléma nem nehezebb, mint a prímfaktorizáció hiszen, ha a támadó képes az n modulus faktorizálására, akkor ki tudja számolni a d titkos kulcsot az (n; e) nyilvános kulcsból.
* RSA probléma -> Prímfaktorizáció:
  + Nem tudjuk, hogy ha az RSA probléma megoldható, akkor tudunk -e hatékony algoritmust adni a prímfaktorizációra.

Támadások az RSA-val szemben:

Speciális nyílt üzenetek halmaza:

* Támadás: Csak a titkosított üzenet ismert (COA)
* Input: c, (n,e) Output: m. ahol me ≡ c (mod n)
* Algoritmus: Minden lehetséges nyílt üzenetet kipróbálunk.

Titkosított üzenetek közötti kapcsolat:

* Támadás: Választott üzenet alapú támadás (CCA)
* Input: c, (n,e) és m’ egy választott c’-re Output: m
* Algoritmus:
* Választunk egy véletlen r ∈ P.
* Kiszámoljuk r’ ≡ re (mod n), és kérjük c’ ≡ r’\*c (mod n) visszafejtését.
* Megkapjuk m’-t, ahol m’ ≡ (c’)d ≡ (r’\*c)d ≡ r\*m (mod n)
* r ismeretében m kiszámítható m’-ből.

RSA-OAEP:

A tankönyvi RSA egy nem biztonságos titkosítási séma.

A képen szöveg, képernyőkép, Betűtípus, algebra látható

Automatikusan generált leírás

Egyirányú függvény:

* Kiszámítása könnyű: Adott x, és könnyű kiszámítani f(x)-et.
* Nehéz invertálni: Adott f(x), nehéz kiszámítani x-et.
* Nem tudjuk, hogy létezik -e egyirányú függvény.
* Egyirányúnak bizonyul: p, q prímek szorzata, ahol |p| = |q|.
  + Könnyű kiszámítani: Adott p, q, könnyű f (p, q) = p\*q kiszámítása.
  + Nehéz invertálni: Adott f (p, q) = p \* q, nehéz kiszámítani p vagy/és q-t. (prímfaktorizáció).

Egyirányú csapóajtó függvény:

* Egyirányú függvény + csapóajtó információ:
  + Bizonyos plusz információval viszont könnyű invertálni: x kiszámítása f(x)-ből.
* Nem tudjuk, hogy létezik -e egyirányú csapóajtó függvény.
* Egyirányú csapóajtó függvénynek bizonyul: moduláris hatványozás RSA modulussal
* Egyirányú függvény:
  + Kiszámítani könnyű: Adott m, könnyű kiszámítani f (m) = me (mod n) (gyors hatványozás).
  + Nehéz invertálni: Adott f (m) = me (mod n) és (n; e), nehéz kiszámítani m-et. (RSA probléma)
* Csapóajtó információ:d; p, q, φ (n).

5.Fólia

Hash függvény:

Kriptográfiai hash függvény:

* A H: {0,1}\* 🡪 {0,1}n n ∈ N függvényt hash függvénynek nevezzük.
* Tetszőleges véges hosszú üzenethez n hosszú üzenetet rendelünk.
* Példa: MD5, SHA-1, SHA-256, SHA-512, SHA-3
* Adatintegritás ellenőrzése: Hash függvénnyel ellenőrizhetjük, hogy egy állomány változott -e vagy sem. Az állomány hash értéke szeparáltan tárolt. Kiszámítjuk az állomány hash értékét és összevetjük a tárolt hash értékkel. Ha különböznek, akkor az állomány módosult.
* A hash értéket lenyomatnak is hívjuk.
* Lavinahatás: Egy bit változása az inputban, jelentős változást eredményez az outputban. (pl. az output fele)

Elvárások:

* A hash függvények nem injektívak.
* Az (x,x’) ∈ {0,1}\*x{0,1}\* a H hash függvény egy ütközése, ha x nem egyenlő x’ és h(x) egyenlő h(x’).
* Őskép ellenálló: Adott y ∈ Y értékhez, nehéz olyan x ∈ X értéket megadni, hogy H(x) = y.
* Második őskép ellenálló (gyengén ütközésmentes): Adott x értékhez nehéz olyan x’ értéket találni, hogy x nem egyenlő x’ és H(x) = H(x’).
* Ütközésmentes (erősen ütközésmentes): Nehéz olyan x; x’ ∈ X értékeket találni, hogy H(x) = H(x’)

Üzenethitelesítés (Message Authentication Codes-MAC):

* Hitelesség: forrása az, amit megjelöltek, adatintegritás

A képen szöveg, diagram, Betűtípus, sor látható

Automatikusan generált leírás

A képen szöveg, Betűtípus, fehér, algebra látható

Automatikusan generált leírás

Digitális aláírási sémák:

* Hitelesség (forrása az, amit megjelöltek, adatintegritás)
* Letagadhatatlanság

DS = (Key; Sign;Ver ) hármas, ahol

* Key: A Key kulcsgeneráló algoritmus a k biztonsági paraméterre kiszámítja a (PK; SK) kulcspárt, ahol PK nyilvános és SK titkos.
* Sign: A Sign aláíró algoritmus az SK titkos kulcshoz és az m ∈ {0,1} \* üzenetre generál egy s = SignSK (m) aláírást.
* Ver: A Ver ellenőrző algoritmus a PK nyilvános kulcsra, az m üzenetre, és az s aláírásra IGAZ vagy HAMIS értéket ad vissza. IGAZ esetén az aláírás érvényes, HAMIS esetén érvénytelen.
* M: üzenetek halmaza, S: aláírások halmazaA képen szöveg, képernyőkép, Betűtípus, sor látható

  Automatikusan generált leírás

Támadó célja:

* Teljes feltörés: A támadó ki tudja számolni az aláíró fél titkos kulcsát.
* Univerzális hamisítás: A támadó bármilyen üzenethez képes érvényes aláírást generálni.
* Szelektív hamisítás: A támadó képes egy általa választott üzenethez aláírást generálni.
* Egzisztenciális hamisítás: A támadó képes egy aláírt üzenetet generálni.

Támadási módok:

* Csak a nyilvános kulcs ismert (Key-only attack): A támadó csak a nyilvános kulcsot ismeri.
* Ismert üzenet alapú támadás (Known message attack): A támadó ismer egy ugyanazon kulccsal aláírt üzenetlistát.
* Választott üzenet alapú támadás (Chosen message attack): A támadó rendelkezésére áll egy általa választott üzenetek és a hozzájuk tartozó aláírások listája.
* Adaptívan választott üzenet alapú támadás (Adaptive chosen message attack): A támadó rendelkezésére áll egy általa választott üzenetek és a hozzájuk tartozó aláírások listája, ahol az üzenetet a korábban megkapott aláírások alapjám választja ki.

RSA aláírás:

* Véletlenül választunk két nagy prímet: p; q. (Miller-Rabin primteszt)
* Kiszámítjuk az RSA modulust: n = p \* q.
* Kiszámoljuk az Euler-féle φ függvényt: φ(n) = (p-1)(q-1).
* Választunk egy véletlen e egészt: 1< e < φ(n) és (e, φ(n)) = 1. e a titkos kitevő. (Euklédeszi algoritmus)
* Kiszámítjuk d-t: 1<d< φ(n) és ed ≡ 1 (mod φ(n)). d a visszafejtő kitevő. (Kibővített Euklédeszi algoritmus)

PK = (n,e), SK = d és φ(n), p,q titkos paraméterek M = S = Zn.

SignSK(m) = md (mod m) ∀m ∈ M, ahol SK = d.

VerSK (m,s) = TRUE, se ≡ m (mod n), False, egyébként.

∀m ∈ M x S, ahol PK = (n,e).

RSA-val szembeni támadások:

* Az RSA univerzálisan hamisítható a választott üzenet alapú támadás mellett.
* Input: tetszőleges m,PK = (n, e), s’ egy adott m’ üzenetre Output: s
* Algoritmus:
  + Véletlenül választunk egy r ∈ M.
  + Kiszámítjuk: r’ ≡ re (mod n)
  + Kérjük az m’ ≡ m\*r’ (mod n) aláríását, megkapjuk s’-t.
  + Kiszámítjuk s ≡ s’\*r-1 (mod n)
* Az RSA egzisztenciálisan hamisítható a csak nyilvános kulcs ismert támadás mellett.
* Input: PK = (n; e) Output: (m, s) ∈ M x S
* Algoritmus:
  + Véletlenül választjuk: s ∈ S
  + Kiszámítjuk: m ≡ se (mod n)

RSA-FDH (Full Domain Hash) aláírási séma:

DS = (Key, Sign, Ver )

Key:

* Véletlenül választunk két nagy prímet: p; q. (Miller-Rabin primteszt)
* Kiszámítjuk az RSA modulust: n = p \* q.
* Kiszámoljuk az Euler-féle φ függvényt: φ(n) = (p-1)(q-1).
* Választunk egy véletlen e egészt: 1< e < φ(n) és (e, φ(n)) = 1. e a titkos kitevő. (Euklédeszi algoritmus)
* Kiszámítjuk d-t: 1<d< φ(n) és ed ≡ 1 (mod φ(n)). d a visszafejtő kitevő. (Kibővített Euklédeszi algoritmus)

PK = (n,e), SK = d és φ(n), p,q titkos paraméterek M ={0,1}\*, S = Zn.

SignSK (m) = H(m)d (mod n) ∀m ∈ M, ahol SK = d, H: {0,1} \* 🡪 Zn hash függvény.

VerPK (m,s) = True, se  ≡ H(m) (mod n), False, egyébként.

∀m ∈ M x S, ahol PK = (n,e).

6.Fólia

Nyilvános Kulcs Infrastruktúra Public Key Infrastructure(PKI):

Azon hardver, szoftver, emberierőforrások, szabályzatok, eljárásokösszessége, melyek szükségesek tanúsítványok létrehozására, kezelésére, terjesztésére, használatára, tárolására és visszavonására.

A képen rajzfilm, diagram látható

Automatikusan generált leírás

Szolgáltatók felépítése:

* Regisztrációs hivatal (RA – registration authority):
  + Az ügyfelek megbízható hitelesítése.
  + A tanusítványkérés összeállítása és továbbítása.
* Hitelesítő hivatal (CA – certification authority) feladatai:
  + Tanúsítványkérések fogadása.
  + Kulcspárok generálása a különböző implementációk esetén.
  + A nyilvános kulcsú tanúsítványok kiállítása.
  + A kiadott tanúsítványok közzététele a nyilvános tanúsítványtárban.
  + Korábban kiadott tanúsítványok és szükség esetén kulcspárok megújítása.
  + Tanúsítványok visszavonása.
  + A visszavont tanúsítványok listájának közzététele (publikálása) a tanúsítványtárban.
  + Jellemzői:
    - A CA védelme (fizikai, logikai) alapvetőfontosságú.
    - Egy CA kiadhat tanúsítványt felhasználók vagy más tanúsítványkiadók részére (akár mindkettő).
    - Felhasználó tanúsítványok esetén szavatolja, hogy a tanúsítványban szereplő publikus kulcshoz tartozó privát kulcsa tanúsítványban szereplő entitás birtokában van. Egyéb információk (pl elérhetőség, eljárásrend, felhasználhatósággal kapcsolatos infók) feltüntetésével, azok helyességét is szavatolja.
    - Minden tanúsítványban elhelyezi saját nevét és aláírását ezáltal, ha a CA megbízható, akkor a tanúsítvány is.
    - A CA titkos kulcsa az alapján az összes általa aláírt tanúsítványban vetett bizalomnak, ezért a CA legfontosabb feladata a saját titkos kulcsának védelme.
* Tanúsítványtár:
  + Speciális adatbázis, amely tartalmazza
    - a CA által kibocsájtott tanúsítványokat,
    - a visszavont tanúsítványokat
    - egyéb, a tanúsítványra vonatkozó adatokat.
  + Feladata:
    - Bármely tanúsítvány állapotáról valós időben információt szolgáltasson.
  + Szolgáltatásai:
    - Biztosítja az ügyfeleket egy adott tanúsítvány hitelességéről, érvényességéről.

Tanúsítványok életciklusa:

Tanúsítvány kiadása, használata, visszavonása.

Tanúsítvány használata:

* A tanúsítvány szerepe egy adott személy, szervezet vagy szerver egy publikus kulcs közötti kapcsolat igazolása.

1. Az alkalmazás működése közben olyan pontra ér, ahol valamelyik fél publikus kulcsára van szükség. (pl. digitális aláírás, aszimmetrikus titkosítás, felhasználó hitelesítés)
2. Az alkalmazás ekkor valamely szabvány segítségével elkéri a szolgáltató tanúsítványtárából a tanúsítványt. Ellenőrzi a rajta lévő aláírást. Ha az nem hiteles, vagy a tanúsítvány érvénytelen akkor az alkalmazás megakad, biztonsági intézkedés következik.
3. Amennyiben a tanúsítvány hitelesésérvényes, az alkalmazás eldönti, hogy a kiállító hitelesítő szervezet megbízható-e vagy sem. A legtöbb PKI-t használó szoftver rendelkezik egy beépített listával a megbízhatónak megítélt hitelesítő szervezetekről. Ha a szolgáltató nincs benne ebben a listában, akkor az alkalmazása hitelesség szolgáltató aláíró tanúsítványának kiállítójához fordul.
4. Az alkalmazás mindaddig ismétli a 2. és 3. lépést, amíg megbírható hitelességszolgáltatóhoz nem jut, vagy pedig óvintézkedésekre nem kerül sor.

Tanúsítvány felépítése:

Verzió, Sorozatszám, Aláírási algoritmus, Aláírás-kivonatoló algoritmus, Kiállító, Érvényesség kezdete, Érvényesség vége, Tulajdonos, Nyilvános kulcs, Ujjlenyomat-algoritmus, Ujjlenyomat.

A képen szöveg, diagram, képernyőkép látható

Automatikusan generált leírás

Tanúsítvány visszavonása, Visszavonási lista (Certificate Revocation List):

* A hitelesítés szolgáltató által kiadott tanúsítványok közül az adott időpillanatban visszavont tanúsítványokat tartalmazó, aláírás lista.
* A szolgáltató előre meghatározott időközönként hozza nyilvánosságra.
* A szolgáltató eseményvezérleten bocsát ki, azaz minden esetben, amikor változik egy általa kibocsátott tanúsítvány állapota új listát bocsát ki.

7.Fólia

RFC (Request For Comments) 4949:

Az azonosítási folyamat két lépése: azonosítás + hitelesítés

* Azonosítás (identification): Azonosító megadása.
* Hitelesítés (authentication): Ellenőrző adat megadása és annak ellenőrzése. Ezzel alátámasztjuk, megerősítjük az állított azonosságot. (pl.: Azonosítás: felhasználói név megadása (bárki ismerheti), Hitelesítés: jelszó megadása (titokban kell tartani))
* Feljogosítás (authorization): Az erőforrásokhoz való hozzáférési jogok megadása.
* A rendszer a hitelesített felhasználónak engedélyezi az erőforrások igénybevételét, ha arra jogosult.

Hitelesítés módjai:

* Tudás alapú: jelszó, PIN (personal identification number), előre meghatározott kérdésekre vonatkozó válaszok
* Birtoklás alapú: kulcs, elektronikus kártyák, smart kártya. Ezeket az eszközöket tokeneknek nevezzük.
* Biometrikus:
  + Statikus: ujjlenyomat, írisz, retina, DNS
  + Dinamikus: aláírás, kézírás, beszédhang, gépelési ritmus, járási mód, szóhasználat, testbeszéd, arcmimika

Jelszavas hitelesítés:

Jelszavak segítségével hitelesítjük a felhasználót, aki a rendszerben azonosítójával van jelen. Azonosítók (ID) szerepe:

* Az azonosító alapján dönt a rendszer arról, hogy egy felhasználó jogosult -e hozzáférésre.
* Az azonosító alapján rendel a rendszer privilégiumokat egy felhasználóhoz. (superuser, guest or anonymous accounts etc.)
* Az azonosítóknak nagy szerepe van a diszkrecionális hozzáférés-vezérlésben (DAC) is. (Pl. más felhasználók azonosítóinak megadásával jogosultságot adhatunk olvasásra.)

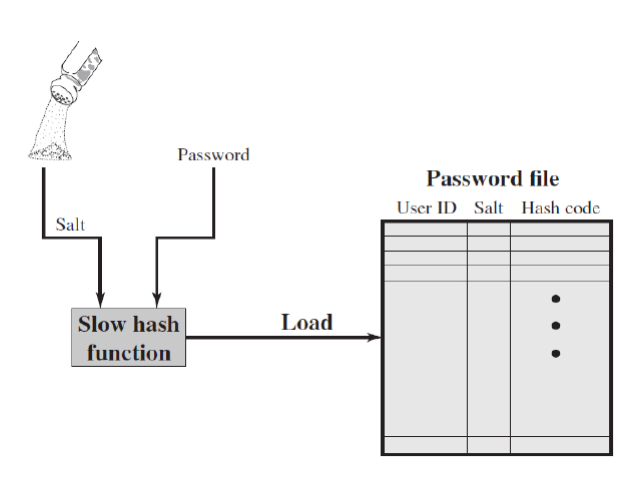
Jelszavak népszerűsége:

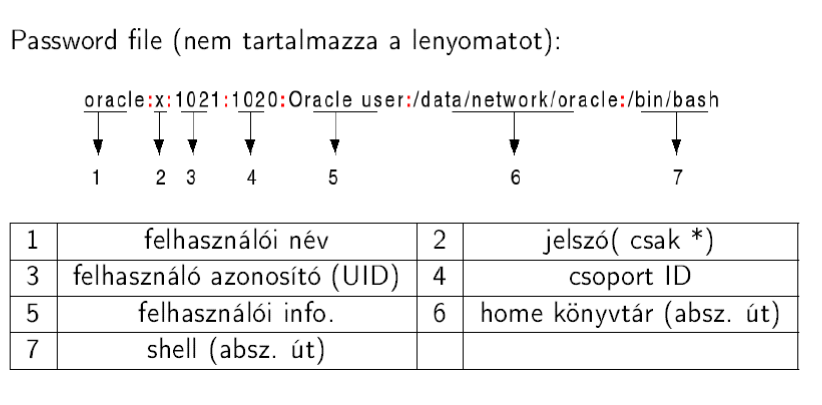
* Könnyen, bárki számára elérhető az azonosítási folyamat.
* Nincs szükség semmilyen klisen oldali szoftver telepítésére, hogy a kliens oldali hardver (pl.: kártyaolvasó) használni tudjuk.
* A tokenek drágák és sokszor kényelmetlen hordozásuk, főleg, ha több tokent is használunk.

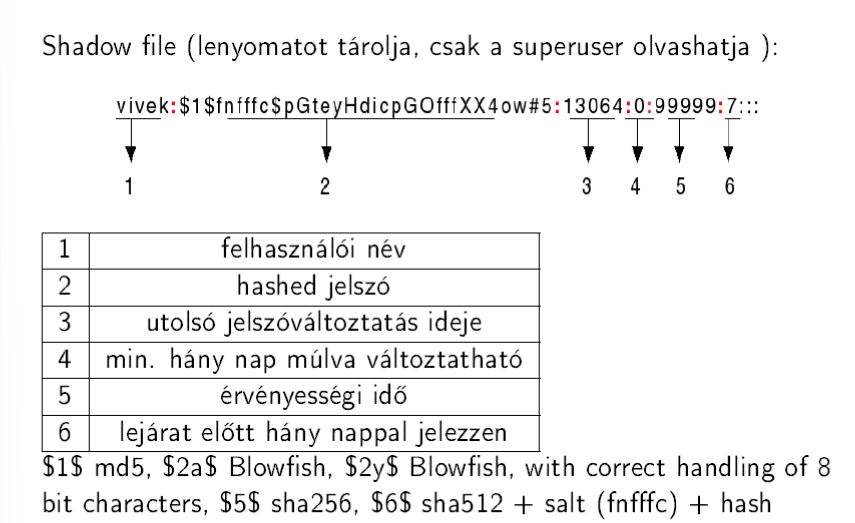
Helyi hitelesítés:

Unix állományok: A jelszó és egy só hash értékét adjuk meg.

* A felhasználó választ vagy kap egy jelszót.
* Ehhez a jelszóhoz tartozik egy fix hosszúságú só érték. A só idő-alapú (régebbi implementációk), álvéletlen vagy véletlen (új implementációk) generátorok által megadott véletlen szám.
* Kiszámoljuk a jelszó és a só konkatenációjának lenyomatát.
* A jelszófájlban tároljuk a sót (nyíltan) és a lenyomatot a megadott azonosítóval.



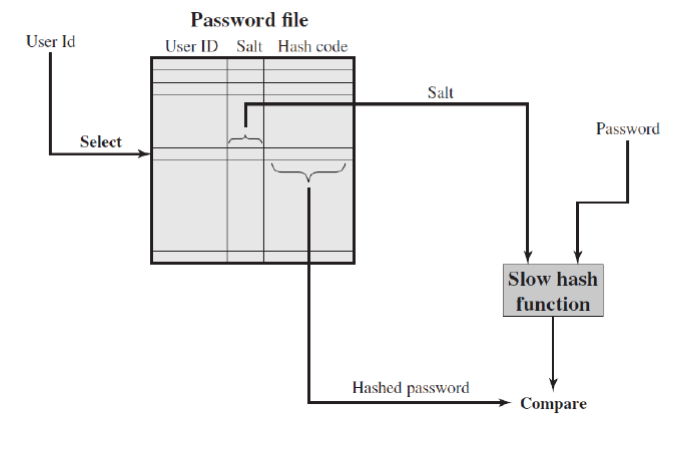




UNIX azonosítási folyamat:

* A felhasználó megadja azonosítóját (ID) és jelszavát.
* Az operációs rendszer az ID segítségével kikeresi a nyílt sót és a lenyomatot.
* Kiszámítja a lenyomatot és összehasonlítja a tárolttal.
* A só jelentősége:
  + Ugyanazon jelszó mellett is különböző lenyomatok tárolódnak.
  + Nehezíti az offl­ine szótártámadást. A támadónak minden egyes jelszó sejtését minden egyes az állományban szereplő sóval ki kell próbálnia (többszörözi a próbálgatások számát).
  + Megnehezíti annak ellenőrzését, hogy egy felhasználó ugyanazt a jelszót használja -e a különböző rendszereknél.

Jelszó ellenőrzés:



OpesnBSD:

* Az OpenBSD operációs rendszer ismert arról, hogy a biztonságra fókuszál.
* OpenBSD a Blowfish szimmetrikus titkosításon alapuló hash függvényt (Bcrypt) használ. Elég lassú.
* Bcrypt legfeljebb 55 karakter hosszú jelszavakat és 128 bites sóhoz 192 bites hash értéket ad meg.
* Bcrypt egy költség változót is kezel, melynek növelésével a Bcyrpt hash kiszámítása több időbe kerül.
* Új jelszó esetében a költség változó konfigurálható, magasabb privilégiumú felhasználóhoz nagyobb költség rendelhető.
* SUSE Linux is Bcrypt-et használ.

Offline szótár támadás:

A támadók megszerzik a lenyomatokat tartalmazó állományt. A támadók a lehetséges jelszavakból egy nagy szótárat készítenek. A szótár elemeit (minden egyes sóval) hash-elik és hasonlítják az állományban levőkkel. Ha sikertelen, akkor a szótár elemein transzformációkat hajtanak végre, és úgy keresik az egyezést. Transzformációk: fordított sorrend, tükrözés, speciális karakterek/sztringek beszúrása stb.

Szivárvány táblák:

A támadó előre kiszámolja a lehetséges hash értékeket a szótárban levő elemekre, a lehetséges sókkal véve. Az eredmény egy hatalmas tábla.

Támadási példa:

A szerző (1990 Klein, D.) UNIX jelszófájlokat gyűjtött össze, közel 14,000 hash értéket. A jelszók negyedét sikerült kitalálnia kevesebb, mint egy óra alatt.

Algoritmus:

* A felhasználó neve, annak kezdőbetűi, felhasználói név más személyes információ felhasználása.
* Különböző szótárak szavainak kipróbálása.
* A kapott szavakra különböző transzformációk végrehajtása: (permutációk, első betű vagy a teljes szó nagybetűssé konvertálása, vagy az "o" betű "zero"-val való helyettesítése, stb)

Védelmi intézkedések:

* Elég hosszú só és elég hosszú hash érték használata. (OpenBSD biztonságosan működik belátható ideig.)
* Jelszófájlok illetéktelen hozzáféréstől való védelme (shadow file).
* Behatolásérzékelő rendszerek jelzik az incidenst, új jelszó gyors újrabeállítása.

Modern megvalósítások:

* Számítási kapacitás jelentősen megnőtt. 2012: A AMD Radeon HD7970 GPU-vel rendelkező PC átlagosan 8:2 \* 109 jelszót tud kipróbálni másodpercenként. A 2000-es évek elején csak drága szuperszámítógépek voltak képesek erre.
* Modellek alkalmazása, melyek a természetes nyelv betűinek előfordulási valószínűségét vizsgálják (pl. standard Markov modell, valószínűségi környezetfüggetlen nyelvtanok). Cél: A szótár/tábla méretének csökkentése. 2009-ben egy SQL befecskendezéses támadással (RockYou.com online játék) 32 millió nyílt jelszóra tettek szert. Azóta egyre több minta áll rendelkezésre analízis szempontjából.

Általános védelmi intézkedések:

* Felhasználók oktatása: A felhasználókban tudatosítjuk a jelszóválasztás fontosságát és módját (hossz, nagybetűk, számok, speciális karakterek, „My dog's first name is Rex” 🡪 MdfniR). Sajnos nem elegendő ez az intézkedés a sikerhez.
* Számítógép-generált jelszavak: Ha a jelszó elég random, a felhasználó nem fogja megjegyezi. FIPS 181 megoldást ad megfelelő számítógépes generálásra. Az algoritmus olyan szótagokat generál, melyek konkatenációja könnyen kiejthető, megjegyezhető.
* Reaktív jelszó ellenőrzés: Felhasználói jelszavak folyamatos ellenőrzése valamely jelszó feltörő programmal. A kitalált jelszavakat jelentik a felhasználóknak. Hátrány: erőforrás igényes. Egy password cracker: http://www.openwall.com/john/doc/.
* Összetett eljárásrend: A felhasználó maga választja jelszavát, bár a rendszer ellenőrzi annak megfelelőségét. Cél: egyensúly a felhasználói elfogadottság és a megfelelő jelszóerősség között.

Birtoklási alapú hitelesítés:

Kártyák:

* Dombornyomott kártyák: Vésett karakterek az elején.
* Mágnescsíkos kártyák: Mágnescsík hátul, karakterek az elején.
* Memória kártyák: A chipkártya legegyszerűbb formája, tárolni tud adatokat, de számításokat nem végez.
* Smart kártyák: A kártya egy mikrokontrollert és memóriát tartalmaz, valamint hatékony kriptográfiai és hitelesítési algoritmusok bevezetésére nyújt lehetőséget.
  + Kontaktusos: Az olvasóba kell helyezni. A kártya és az olvasó fizikai kapcsolaton keresztül kommunikálnak.
  + Kontaktusmentes: A kontaktusmentes kártyát elég néhány centiméternyire közelíteni a leolvasó berendezéshez és a tranzakció máris megvalósítható. Az olvasó és a kártya rádió frekvencián keresztül kommunikál.
  + Hitelesítés:
  + Statikus: A felhasználó hitelesíti magát a tokennek, és a token hitelesíti a felhasználót a rendszernek.
  + Dinamikus jelszó generátor: A token időközönként (pl. percenként) új jelszót generál, melyet kell majd a felhasználónak megadnia. (időszinkronizációs egyszer használatos jelszó)
  + Kihívás-és-válasz: A rendszer generál egy véletlent (kihívást), melyre a token választ (pl. digitális aláírás) ad.

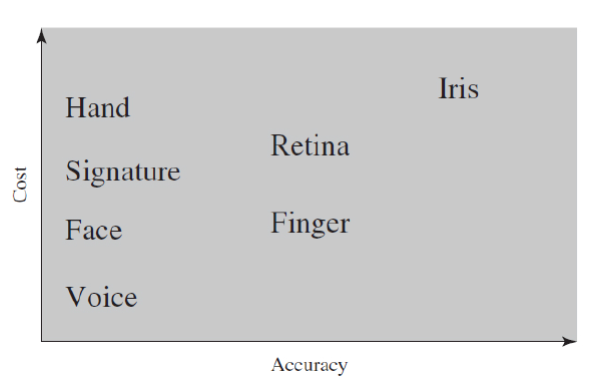
Biometrikus hitelesítés:

* A felhasználót egyedi, fizikai tulajdonságai alapján hitelesítjük.
* Technológiai szempontból összetettebb és drágább is, mint a jelszavas vagy birtoklás alapú.
* Fizikai tulajdonságok:
  + Arcfelismerés: Az arc elemeinek (szem, szemöldök, orr stb.) alakja, elhelyezkedése alapján hitelesítünk. Alternatív megoldás infravörös hőkamera használata.
  + Ujjlenyomat(ujjnyom): A bőrlécrendszer által hátrahagyott nyom a felületeken. Egyediek.
  + Kéz-geometria: A kéz alakja, az ujjak szélessége és hossza alapján hitelesítünk.

Fizikai tulajdonságok:

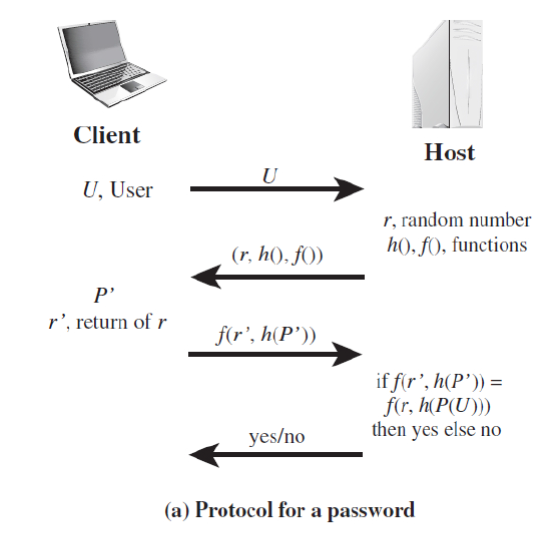
* Retinal pattern: A szem hátsó falán található vérerek mintázata egyedi. Alacsony intenzitású infravörös sugarakkal világítja át a leolvasó a szemfenéket. A fejet rögzíteni kell.
* Írisz: Az írisz a szem szivárványhártyája. A szivárványhártya képét a szemet egyedivé tevő összes jellegzetességgel (gödröcskék, körök, árkok, korona, szövetszálak) háromdimenziós kontúr-térképpé alakítják.
* Aláírás: Mindenkinek egyedi az aláírása, bár egy tulajdonostól többféle minta származhat.
* Hang: A hang jobban köthető tulajdonosához, mint az aláírás. Idővel változik.

Költség vs. Pontosság:

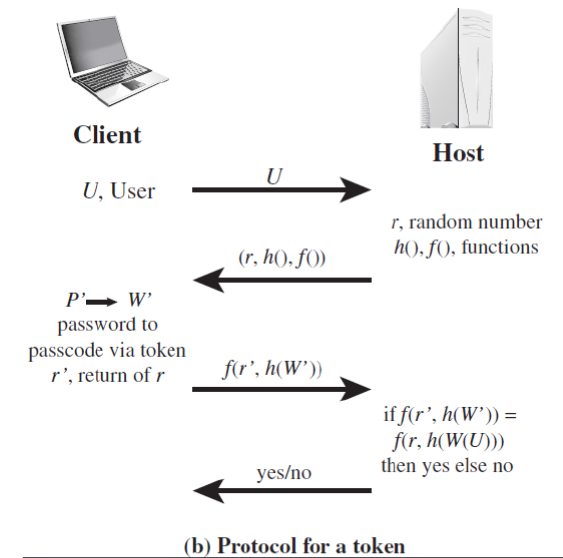


Távoli hitelesítés:

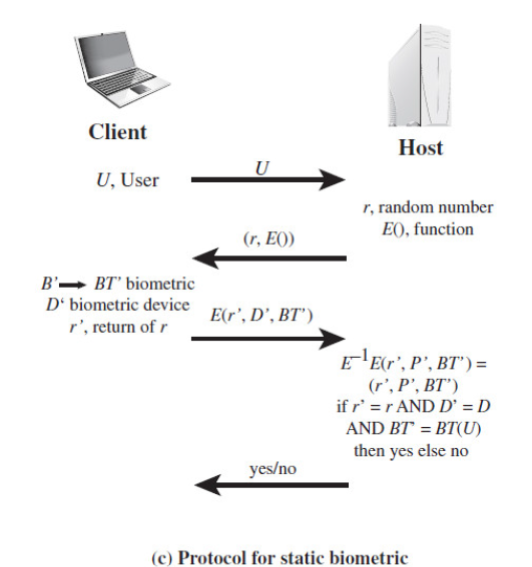
* Távoli hitelesítés az Interneten, hálózaton keresztül történik.
* További fenyegetés: lehallgatás, üzenet-visszajátszás.
* A támadások kivédésére kihívás-válasz protokollokat használunk.
* Jelszó protokol:
  + A felhasználó elküldi azonosítóját.
  + Az ellenőrző fél generál egy r véletlent, melyet noncenak hívunk, és a noncet két függvény nevével visszaküldi.
  + Felhasználó válasza: f(r’h(P’)), ahol r’ = r és P’ a jelszó, h egy hash függvény.
  + Az ellenőrző fél tárolja a hash értékeket: h(P(U)) adott U felhasználónál.
  + Ha f(r’,h(P’)) = f(r, h(P(U))), a felhasználó sikeresen hitesítettük.
* Biztonsági kérdések:
  + A jelszó hash-e van letárolva.
  + Az f védi a lenyomatot a lehallgatástól.
  + A nonce a visszajátszásos támadással szemben véd.



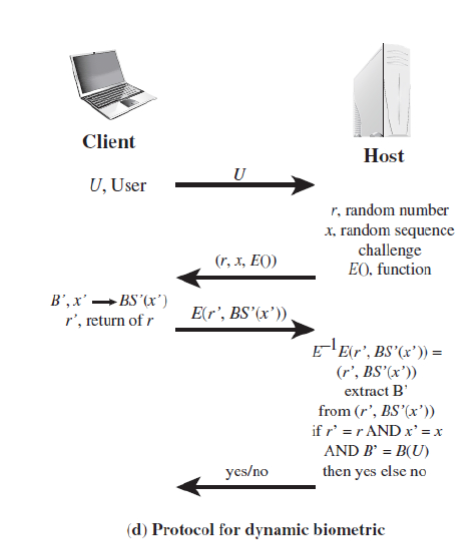
* Token protokoll:
  + A felhasználó elküldi azonosítóját.
  + Az ellenőrző fél generál egy r noncet, és a noncet két függvény nevével visszaküldi.
  + A token megad egy W’ jelszókódot. Ez a jelszókód vagy statikus, tárolt, vagy egyszer használatos, generált.
  + A felhasználó P’ jelszavának megadásával aktiválja a jelszókódot.
  + A token elküldi: f (r’ h(W’)), ahol r’ = r és P’ a jelszó, h egy hash függvény.
  + Statikus jelszókód esetén az ellenőrző fél tárolja a h(W(U)) értéket az U felhasználónál. Dinamikus esetben az egyszer használatos jelszó legenerálása után kiszámítja a lenyomatot.
  + Ha f (r’, h(W’)) = f (r, h(W(U))), a felhasználót sikeresen hitelesítettük.



* Statikus biometrikus protokoll:
  + A felhasználó elküldi azonosítóját.
  + Az ellenőrző fél generál egy r noncet, és a noncet egy titkosító függvény nevével visszaküldi.
  + Kliens oldalon legenerálódik egy BT’ biometrikus minta a felhasználó B’ biometrikus adatából, majd visszaküldi a
  + E(r’, D’, BT’) titkosított üzenetet, ahol D’ a biometrikus eszköz azonosítója.
  + Az ellenőrző fél visszafejtés után a kapott értékeket összehasonlítja a tároltakkal.

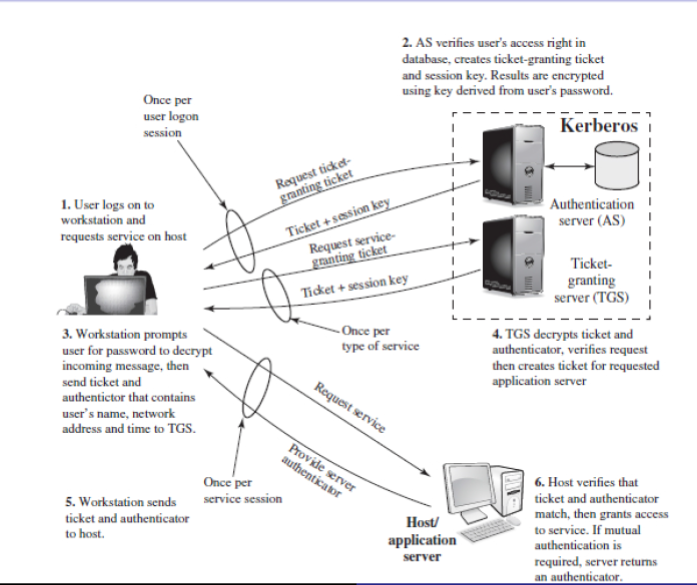


* Dinamikus biometrikus protokoll:
  + A felhasználó elküldi azonosítóját.
  + Az ellenőrző fél generál egy r noncet, egy x véletlen kihívást és egy titkosító függvény nevével visszaküldi. A kihívás egy karaktersorozat. A felhasználónak fel kell olvasnia (hang alapú hitelesítés), be kell gépelnie (billentyű lenyomási technika), vagy le kell írnia (kézírás ellenőrzés) hogy legenerálja a BS’(x’) biometrikus értéket.
  + A kliens titkosítja a véletlent és a biometrikus értéket.
  + Az ellenőrző fél visszafejti az üzenetet és összehasonlítja a tárolt biometrikus értékkel.



Internetes alapú hitelesítés (Kerberos):

* Az autentikációt nem több szerver, hanem egy autentikációs szerver végzi.
* Az MIT fejlesztette ki, Internet szabvány.
* Kerberos elvárja, hogy a felhasználó minden szolgáltatásnak hitelesítse magát és minden szerver hitelesítse magát a kliensnek.
* A protokoll résztvevői: kliensek, alkalmazás szerverek és egy Kerberos szerver.
* Az autentikációs szerver (AS) tárolja valamennyi kliens jelszavát egy központi adatbázisban.
* Ha AS hitelesíti a klienst és erről értesíti az alkalmazás szervert. AS minden alkalmazás szerverrel kicserél egy-egy titkos kulcsot.



* Azonosítási folyamat:
  + A felhasználó belép egy munkaállomásra és kéri valamely szerver szolgáltatását.
  + Kliens processz elkezdődik: A kliens elküldi az ID-t és kéri a TGT(ticket-granting ticket)-et.
  + AS visszaküldi: TGT, session key (egyszer használatos titkos kulcs) titkosítását, ahol a kulcs a felhasználó jelszavából származtatott.
  + A kliens kéri a felhasználó jelszavát, mellyel visszafejti az üzenetet. Ha a jelszó helyes, akkor a TGT-t és a session key-t megkapja.
  + Sem a jelszó, sem a lenyomata nem lett átküldve.
* Ticket:
  + A ticket jelzi, hogy AS hitelesítette a klienst és a felhasználót. A ticket tartalmazza: felhasználó ID, TGS ID, időbélyeg, érvényességi idő, és ugyanaz a session key. A ticket titkosítva van a szimmetrikus kulccsal, melyet AS és a szerver cserélt ki.
  + Ez a ticket hozzáférést biztosíthatna közvetlenül az adott alkalmazás szerverhez, ekkor viszont minden egyes szerver hozzáféréssel az AS-hez kellene fordulni. Ehelyett a ticket TGS (ticket-granting server) számára generált.
  + A ticket-tel még több ticket-et lehet igényelni. TGT többször felhasználható.
  + Támadó célja: TGT megszerzése (AS és TGS közös titkos kulcsával titkosított TGT titkosítva van egy a jelszóból származtatott kulccsal rövid érvényességi idő (8 óra))
* Szolgáltatás igénylése:
  + A felhasználó szeretne egy szolgáltatást igénybe venni.
  + A kliens processz elküldi TGS-nek: szolgáltató szerver ID, TGT, hitelesítő kód, ami session key-jel titkosított felhasználói ID+hálózati cím+időbélyeg.
  + TGS visszafejti a TGT-tét, ellenőrzi (érvényességi idő, ID-k stb.) és megkapja a session key-t.
  + A session key-jel visszafejti a hitelesítő kódot és ellenőrzi.
  + Ticket: session key szétosztása, hitelesítő kód: bizonyítja a kliens identitását
  + A hitelesítő kód rövid érvényességi idejű és egyszer felhasználható. Következő szolgáltatás igénylésénél új hitelesítő kód szükséges.
* Szolgáltatói ticket:
  + A következő 2 lépés ugyanaz, mint az előző 2.
  + TGS egy service-granting ticket-et, új session key-t küld a kliensnek. Az egész üzenet a régi session key-jel van titkosítva.
  + A service-granting ticket TGS és a szolgáltatói szerver közös kulcsával van titkosítva.
  + Minden egyes alkalommal, amikor a kliens a szolgáltatást szeretné használni elküldi a service-granting ticket-et és egy új hitelesítő kódot.
  + Ha a kliens hitelesítést kér, a szolgáltatói szerver a kapott időbélyeghez hozzáad 1-et, és titkosítja a session key-jel.
  + A session key alkalmas a későbbi üzenetek titkosítására.

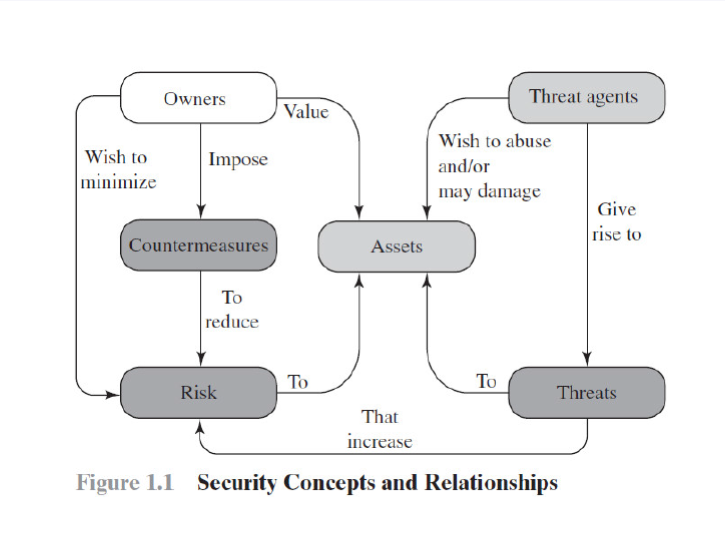
8.Fólia

Informatikai biztonság modellje, tervezési alapelvek:

Alapfogalmak:

* Biztonság alanya (asset): Az informatikai rendszer erőforrásai: hardver, hálózat és adathordozók, szoftver, adatok.
* Fenyegetés (threat): Olyan lehetséges művelet vagy esemény, amely sértheti az informatikai rendszer vagy az informatikai rendszer elemei védettségét, biztonságát.
* Sérülékenység (vulnerability): Az informatikai rendszer olyan gyengesége, amelyen keresztül valamely fenyegetés megvalósulhat. Kategóriák:
  + rendszerelem módosulhat: nem megfelelően működik, rossz válaszokat ad, pl. a tárolt adatok jogosulatlanul megváltoznak
  + rendszerelem szivárogtathat, pl. valaki jogosulatlan hozzáféréssel információkhoz jut
  + rendszerelem nem elérhető vagy nagyon lassú, a rendszer vagy hálózat használata lehetetlen
* Támadás(attack): Fenyegetést előidéző cselekmény, mely valamilyen védett érték megszerzésére vagy megsemmisítésre, károkozásra irányul.
  + Módja szerint:
  + aktív: Rendszerelemeket vagy azok működését módosítja
  + passzív: Rendszerinformációk megszerzése és felhasználása
  + Végrehajtója szerint:
  + belső: Jogosult a rendszerelem hozzáférésére, de arra nem jogosult módon használja fel.
  + külső: Egyáltalán nincs feljogosítva a rendszer használatára
  + A gyakorlatban a támadások nagy hányada belső támadás. A külső támadás célja sokszor a rendszerhez való hozzáférés, és azon belső támadás végrehajtása.
* Kockázat (risk): A fenyegetettség mértéke, amely valamely fenyegető tényezőből ered. A kockázat egy fenyegetés bekövetkezése gyakoriságának (bekövetkezési valószínűségének) és az ez által okozott kár nagyságának a függvénye. Cél: a kockázat minimalizálása
* Védelmi intézkedés (countermeasure): Olyan eljárás, eszköz, technológia, mely csökkenti a rendszer sérülésének kockázatát.
  + Alkalmazás területe szerint:
  + Fizikai: kábelezés, védelmi eszközök, ajtók, tűzoltó készülékek, légkondicionálók stb.
  + Ügyviteli: szabályozások, eljárások, oktatás stb.
  + Technikai/algoritmikus: tűzfalak, autentikációs rendszerek, titkosítások stb.
  + Funkcionalitás szerint:
  + Preventív intézkedések: Megelőzik a támadás bekövetkeztét (lehetnek fizikai, adminisztratív vagy technikai) pl. biztonsági frissítések, titkosítás
  + Detektív intézkedések: Ha a preventív intézkedések meghiúsulnak vagy nem lehetségesek, akkor észleljük a támadást. pl. ellenőrző összeg, naplófájlok
  + Korrektív intézkedések: Próbálják kijavítani a sérülést. (lehetnek technikai, adminisztratív) pl. backup/visszaállás.
* A védelmi intézkedések új sérülékenységeket eredményezhetnek.

Modell:



Tervezési alapelvek:

Folyamat:

* Mik azok a vagyontárgyak, erőforrások, amiket meg akarunk védeni? Teljeskörűen kell felmérni a rendszerelemeket.
* Milyen veszélyek fenyegetik az adott erőforrásokat? Kik/mik ellen védjük a rendszerelemeket, milyen lehetőségekkel, erőforrásokkal rendelkeznek.
* Mekkorák a kockázatok? Milyen védelmi intézkedéseket vezessünk be?
* Milyen hatásfokkal kezeli ezeket a kockázatokat a választott biztonsági megoldás? Sikeresség vizsgálata, kudarc gyakorisága.
* A választott megoldás milyen új biztonsági réseket okoz? Működésbeli módosítások dominószerűen hullámzanak végig az adott rendszeren.
* Megéri-e alkalmazni a megoldást? Pénz, idő, alkalmazás kényelmetlensége, csökkenő teljesítmény.

Jogosultságok minimalizálása:

* Ne adjunk több jogosultságot, csak annyit, amennyi feltétlen szükséges a feladat végrehajtásához.
* Preventív intézkedés, hiszen a jogosultságok korlátozásával csökkentjük a véletlen vagy direkt károkozást.
* Bármely informatikai rendszernél alkalmazható.
* Példák:
  + A megosztott állományokhoz csak olvasási jogot adunk a felhasználóknak, ha csak erre van szükségük.
  + A help desk kollégáknak nem adunk jogosultságot a felhasználói fiókok létrehozására, törlésére, ha csak a jelszó módosítását vezényelheti le.
  + Szoftverfejlesztőknek nem adunk jogosultságot a szoftverek fejlesztői szerverekről az éles szerverekre történő átmásolására.

Minimalizálás:

* A jogosultságok minimalizálása alapelv testvére, csak a rendszerkonfigurációra vonatkozik.
* Olyan szoftvereket, alkalmazásokat, szolgáltatásokat ne futtassunk, melyek nem feltétlenül szükséges a biztonságos működéshez.
* Növeli a teljesítményt, tárhelyet takarít meg.
* Példa:
  + Ha egy számítógép csak az elektronikus levelezés szolgáltatást biztosítja, akkor egyéb szolgáltatásokat lehetőleg ne installáljunk.

Több szintű védelem:

* Több szintű és többféle védelmet biztosítsunk
* Egy szintű vagy egyféle védelmet könnyebb támadni (bármilyen erősnek is hisszük), mint többet.
* Valamennyi védelmi mechanizmusnak szerepelni kell:
* preventív, detektív, korrektív
* Példa:
  + Tűzfal használata az Internet és a LAN között és IP Security Architecture (IPSEC) segítségével titkosítják a bizalmas adatokat. Ha a tűzfalat feltörik, a támadóknak még mindig fel kell törniük a titkosítást.

Open design:

* Egy biztonsági mechanizmus elemeinek, működési módjának nyilvánosnak kell lenniük.
* Szakértők elemezhetik az algoritmusokat, így a felhasználók jobban bíznak bennük.
* 1883 Auguste Kerckhoffs alapeve: "az ellenség ismeri a rendszert", azaz azzal a feltétellel tervezzünk rendszereket, hogy az ellenség kezdettől fogva a teljes felépítését ismeri.
* Példa:
  + Csak a tikos kulcsokat tartsuk titokban, a titkosító algoritmusok nyilvánosak.

Felosztás:

* Parcellák, zónák, virtuális terek kialakítása
* Limitálja a kárt, ha egyik megsérül, más terek még védve vannak.
* Különböző zónákban futó alkalmazások egymástól elszigetelődnek.
* Példa:
  + A webszerver szoftver kompromitálódása, nem befolyásolja a levelező szerver működését, ha külön szerverekre telepítjük, vagy virtuális szervereket hozunk létre.
  + Solaris 10 operációs rendszer: zónák. A zóna egy virtuális operációs rendszer környezet, CPU idő, virtuális memória, hálózati sávszélesség, I/O teljesítmény stb. is szabályozható, virtuális szerverekként viselkednek.

Egyszerű megoldást válasszuk:

* Az összetett, komplex rendszerek a legnagyobb ellenségünk.
* Nehéz tervezni, implementálni, tesztelni.
* Ha választani kell egy komplex, sokoldalú rendszer és egy egyszerű, mely kicsivel kevesebbet nyújt, válasszuk az egyszerűbbet.

Pszichológiai elfogadhatóság:

* Ha egy biztonsági mechanizmus gátolja az erőforrások könnyű hozzáférhetőséget, használatát, akkor a felhasználók kikapcsolhatják azt.
* A biztonsági mechanizmusnak a felhasználók számára transzparensnek kell lennie.
* A biztonsági algoritmusoknak követniük kell az emberi gondolkodást.

Erőforrások és vagyontárgyak fenyegetései:

* Hardver:
  + Rendelkezésre állására irányuló fenyegetések a leggyakoribbak.
  + Véletlen vagy szándékos fizikai rongálás, lopás. Pendrive-ok, tabletek, DVD-k stb. eltulajdonítása során az adatok bizalmassága is sérül.
  + Fizikai és adminisztratív intézkedések adhatnak védelmet.
* Szoftver:
  + Rendelkezésre állásra irányuló fenyegetés: alkalmazás törlése, módosítással haszontalanná válhat.
  + Technikai védelem: Backup.
  + Sértetlenségre irányuló fenyegetés: szoftver módosítása, vírusok.
  + Szoftverkalózkodás: másolatok készítése jogosulatlanul (nehéz a probléma megoldása).
* Adat:
  + Védeni kell, mert értékes, egyedi. Mind továbbítás, tárolás, feldolgozás során sérülhet.
  + Bizalmasságra irányuló fenyegetés: pl. személyes adatok, tervek, gazdálkodási adatok jogosulatlan olvasása, megszerzése adatbázisokból.
  + Sértetlenségre irányuló fenyegetés: adatok módosítása.
  + Rendelkezésre állásra irányuló fenyegetés: adat törlése véletlenül vagy szándékosan.

Támadási fa:

A rendszer sérülékenységeit kiaknázó lehetséges támadásokat, fenyegetéseket tartalmazza.

* A fa gyökere a támadás célja.
* A levélelemek a támadások különböző módjait adják meg.
* A gyökérből kiinduló utak nem levélelem csúcsai a cél eléréséhez szükséges részcélok.

Bankszámla feltörése, Internet bank felhasználó hitelesítés alkalmazás:

A képen szöveg, képernyőkép, Párhuzamos, diagram látható

Automatikusan generált leírás

* Felhasználói terminál, felhasználó (UT/U):
  + Ezek a támadások a felhasználói eszközöket célozzák meg, pl. token, smartcard, jelszó generátorok vagy felhasználói tevékenységek.
  + Példa:
  + Féreg (worm): Olyan program, amely a számítógép hálózaton keresztül terjed és károkozó hatását önmaga reprodukálásával, továbbításával éri el.
* Kommunikációs csatorna (CC): Kommunikáció során felmerülő támadások.
  + Példa:
  + Hálózati forgalom lehallgatása (sniffing): csatorna figyelésével bizalmas adatok megszerzése
* Internet bank szerver (IBS): Off-line támadások az Interent bank alkalmzást hosztoló szerverrel szemben.
* Utasítások befecskendezése (Injection of commands):
  + A támadó megfigyeli az UT és IBS közötti kommunikációt. A támadások lényege egy legális résztvevő megszemélyesítése.
* Felhasználó személyazonossági adatainak kitalálása:
  + Nyers erő támadás, teljes kimerítő kipróbálások a felhasználó hitelesítési sémával szemben véletlen felhasználói nevek és jelszavak küldésével. A támadás módja osztott zombi számítógépeken automatizált felhasználói név és jelszó generálása.
  + Zombi számítógép: az Internetre kapcsolódó számítógép, melyeket a támadó irányítása alá vesz és erőforrásait saját célra használja.
* Adathalászat - Pharming:
  + A támadó valamilyen rosszindulatú szoftver vagy kémszoftver segítségével az eredeti lapról egy másik, hamisított weblapra téríti el a felhasználót.
* Munkamenet-eltérítés (Session hijacking):
  + Érvényes, már belépett felhasználó sessionId-jét (vagy session key-jét) próbálja meg megszerezni, hogy jogosulatlanul információhoz vagy szolgáltatáshoz jusson a támadó.
* Pszichológiai manipuláció (Social engineering):
  + Az emberi természetet igyekeznek kihasználni, személyes információk megszerzése céljából.

9.Fólai

Kártékony programok:

* Malicious software = malware.
* Olyan program, mely valamely rendszerbe beépülve az áldozat adatainak, szoftvereinek bizalmasságát, integritását, rendelkezésre állását veszélyezteti, vagy erkölcsi károkat okoznak, mert pénz, időt, embert kötnek le.
* Főbb kategóriák:
  + Terjedés módja szerint: vírusok(virus), férgek(worm) és trójaiak (Trojan).
  + Büntető rutinuk szerint: botnet, adathalászat(phishing), kémprogram(spyware), gyökércsomagok (rootkit)-ek.

Csoportosítás:

* Terjedés módja:
  + fertőzött fájl futtatásával (vírus)
  + szoftver sérülékenységeit aknázza ki (féreg)
  + pszichológiai támadás (trójaiak installálása, adathalász emailek megválaszolása, spamek)
* Gazdaprogram szükségessége:
  + szükséges (vírus)
  + nem szükséges (féreg)
* Sokszorosítás:
  + nem képes önmaga megsokszorozására (trójaiak, spam email)
  + megsokszorozódik (vírus, féreg)
* Büntető rutin alapján:
  + rendszer- és adatfájlok megrongálása szolgáltatás eltulajdonítása (botnet támadás esetén zombie gépek szerzése)
  + információ ellopása (jelszavak, személyes információk)

Támadói eszközrendszer (Attack Toolkit):

* Régen: egy terjedési mód egy büntető rutinnal
* Most: terjedési módok és büntető rutinok széles skálája
* Kevert támadások (blended attack) a fertőzés sebességének maximalizálása céljából többféle terjedési módot is használnak. Többségük még update mechanizmusokat is támogatnak, mely lehetővé teszi aktiválásuk után, hogy megváltoztassák terjedési módukat és büntető rutinukat.
* Az eszközrendszer, vagy crimeware(bűncselekmény+szoftver), többféle terjedési módot és büntető rutint is tartalmaz, melyet akár egy nem szakember is képes telepíteni és használni.

Zeus Crimeware Toolkit (2007):

* Személyes információk (banki információk) megszerzése a felhasználók számítógépeinek megfertőzésével.
* A legfelkapottabb aktív botnet-ek egyike. A botnet olyan hálózatra kapcsolt gépek összessége, amelyek felett átvették az irányítást.
* Többféle módon lopják el a felhasználó identitását, kontrollálják számítógépét.
* HTML kód-injektálás, minden egyes egérkattintásnál lementi a képernyőképet (virtuális billentyűzet ellen) stb.

Fejlett Perzisztens Fenyegetés (Advanced Persistent Threat - APT):

Kiberbűnözés, mely üzleti és politikai célpontokat célozzák meg, többféle behatoló technológia és kártékony program alkalmazásával.

* Fejlett: A támadók számos eszközt használnak fel céljuk eléréséhez. Céljuk, hogy elkerüljék a célkeresztbe állított rendszert körülölelő vagy az abban működő védelmi eszközöket. A NIST (National Institute of Standards and Technology) mindezt így fogalmazta meg: a támadók alkalmazkodnak a védők erőfeszítéseihez annak érdekében, hogy azokkal szembeszállhassanak.
* Perzisztens: A támadók egy jól meghatározott céllal tevékenykednek, és többnyire nem véletlenszerűen, találgatás útján próbálnak rájönni, hogy milyen sebezhetőségeket tudnak kihasználni, hanem már felkészülten, alapos felderítőmunka után lépnek akcióba. Ameddig el nem érik a céljukat (például meg nem találják az általuk keresett bizalmas adatokat), addig nem hagyják el a rendszert. Gondoskodnak arról, hogy a hozzáférésük fenntartható legyen.
* Fenyegetés: Ez esetben célzott, irányított és komplex akciókról beszélünk, amelyek komoly fenyegetést jelenthetnek az informatikai rendszerekre és az adatokra.
* Célok:
  + APT-k különböznek a többi támadástól a célpont körültekintő kiválasztása miatt. Információt gyűjt, hogy pl. megállapítsa az alkalmazott szerepét és a hozzáférés szintjét. Amennyiben ez nem vezet célra, akkor más felhasználók után kutathat, akik jobb pozícióban, több privilégiummal rendelkeznek. Nem egyszerűen feltöri a szervezet infrastruktúráját, sokkal inkább az alkalmazottakra koncentrálnak.
  + Például: Aurora, RSA, APT1, Stuxnet
  + Céljuk: szellemi termék, szerzői tulajdon ellopása, biztonsági és infrastruktúrális adatok megszerzése, infrastruktúra fizikai lekapcsolása. A speciális célpont és a perzisztens jellegük miatt a védekezés nem kivitelezhető hatékonyan csupán hagyományos biztonsági eszközök bevetésével.
  + Védelmi intézkedés: többféle technikai védintézkedés kombinációja, oktatás.
* Technikák:
  + Pszichológiai támadás
  + Célzott adathalász e-mail (spear-phishing e-mail): E-mailen keresztül egy meghatározott személyt, szervezetet vagy vállalatot céloz meg azzal, hogy megszerezze bizalmas adatait vagy kártékony programot installáljon számítógépére.
  + Drive-by-downloads: Támadás, mely egy fertőzött website-on levő kóddal kiaknázza a böngésző sérülékenységét, hogy megtámadja a felhasználó rendszerét böngészés közben.
  + Nulladik napi támadás (zero-day exploits): Valamely számítógépes alkalmazás olyan sebezhetőséget használja ki, ami még nem került publikálásra, a szoftver fejlesztője nem tud róla, vagy nem érhető még el azt foltozó biztonsági javítás. A „nulladik napi” támadás az első vagy „nulladik” napon történik, amikor a fejlesztő még nem juttatta el a biztonsági javítást a szoftver felhasználóihoz.
* Stuxnet:
  + Stuxnet az első katonai féreg, mely 2010-ben bukott le egy iráni nukleáris erőmű számítógépén.
  + A felfedezés óta bebizonyosodott, hogy a Stuxnet vírus az erőműre közvetlen fenyegetést nem jelentett, mert annak az urániumdúsító berendezések voltak a célpontjai. Közvetett értelemben azonban az a tény mindenképpen fenyegető, hogy egy vírus bejutott a létesítménybe.
  + 2010. november 16-án Irán leállította az urándúsítóit, miután a centrifugák több mint 20%-a megsemmisült a Stuxnet tevékenysége nyomán.
  + A férget úgy tervezték meg, hogy törölje magát 2012-ben.
  + Négy nulladik napi támadást hajtott végre, Microsoft Windows operációs rendszerű gépeken jutott be és a Siemens Step7 mérnöki szoftvert kereste.
  + Három modulja:
    - féreg, mely a támadás büntető rutinjait futtatja.
    - kapcsolat fájl, mely automatikusan futtatja a féreg másolatokat.
    - gyökércsomag, mely felelős a kártékony fájlok, processzek elrejtéséért megelőzve az esetleges lebukást.
  + Stuxnet egy fertőzött USB pendrive segítségével jutott be.
  + A féreg elterjedt a rendszerben.
  + Stuxnet gyökércsomaggal módosította a kódot és nem várt parancsokat hajtott végre.

Vírusok:

Program, amely saját másolatait helyezi el (megfertőzi) más, végrehajtható programokban vagy dokumentumokban.

A vírus kód megfertőzhet alkalmazásokat, vagy rendszerfájlokat, vagy programkódokat, melyeket rendszerbetöltésnél futtat a rendszer.

Mostanában script kód formában gyakoriak, melyek a Microsoft Word dokumentumok, Excel táblák vagy Adobe PDF dokumentumok aktív tartalmát adják.

A vírus kód lefutásával valamilyen kárt okoz, pl. törli az adatokat, programokat, melyet az aktuális felhasználó jogosultsága enged.

* Részei:
  + Fertőző mechanizmus (infection vector): Mechanizmus, mellyel a vírus terjed, sokszorozódik.
  + Indíték: Esemény vagy feltétel, mely meghatározza mikor aktiválódik a büntető rutin. Az ilyen a programokat logikai bombának is hívják.
  + Büntető rutin: Kárt okozó tevékenységek. (pl. törlés)
* Célpontok:
  + Boot vírusok: A merevlemez boot szektorába ágyazódik be, így még az operációs rendszer betöltése előtt aktiválódik. Ennek hatására a fertőzött merevlemez az összes meghajtóba helyezett lemezt megfertőzi.
  + Alkalmazásvírusok: A megfertőzőt állományokba beírják a saját kódjukat. Két fajtáját különböztetjük meg: hozzáfűződő (append) és felülíró (replace) vírusokat. Amennyiben egy fertőzött fájlt elindítunk, a vírus betöltődik a memóriába és megfertőzi az összes többi elindított programot.
    - A hozzáfűződő vírusok az alkalmazások végéhez fűződnek, elhelyeznek azonban a program elején egy kódot, hogy az alkalmazás indulásakor előbb ők töltődjenek be.
    - A felülíró vírusok az alkalmazások elejét írják felül saját kódjukkal, így a fertőzött állomány adatot veszít, az eredeti állapot nem állítható helyre.
  + Macrovírusok: Elsősorban makrókat támogató dokumentumszerkesztő programokat támadnak meg (Word, Excel), általuk készített dokumentumokkal terjednek. Terjedésükhöz elegendő egy fertőzött állomány megnyitása, és a vírus már be is töltődik a memóriába, mely a későbbiekben megnyitott dokumentumokat megfertőzi. Ide sorolhatók a levelező vírusok is. A csatolt fertőzött fájlok megnyitásakor aktivizálják magukat és általában a levelezési listában szereplő partnereknek írnak levelet, melyhez saját maguk másolatát is hozzáfűzik. Ha a levelezési címlistában nagyon sok partner van, akkor olyan mennyiségű levéláradat indulhat, amely megbénítja egy nagyvállalat levelezőrendszerét is.
  + Összetett vírus: Többféle módon, többféle állományt képesek megfertőzni.
* Rejtettségi stratégiák:
  + Titkosított vírus: Titkosítással rejti el tartalmát. Az egyik vírusrész egy véletlen titkosító kulcsot generál és titkosítja a vírust. Mivel a vírus titkosítva van különböző kulcs esetén, nem lesznek megegyező bitminták.
  + Lopakodó vírus: Az antivírus programot és az operációs rendszert megkerülve a gép memóriájában maradva, a fájl méreteit, az operációs rendszer jellemzőit és a könyvtár struktúráját megváltoztatják. Látszólag tiszta állapotról szóló jelentést képesek mutatni, például a fertőzött állomány hossza az eredeti, tiszta állapotnak megfelelő értéket jelenít meg. Használhat gyökércsomag technikákat.
  + Gyökércsomag (rootkit): Eszközök, melyeket a támadó használhat, miután megszerezte a legmagasabb felhasználói jogosultságokat.
  + Polimorf vírus: Bizonyos fertőzési ciklusonként képesek megjelenési formájukat változtatni. Ezzel a módszerrel a bitminta alapú felismerést lehet nehezebbé vagy lehetetlenné tenni. (pl. titkosítás, más instrukciókat szúrnak be). Azon vírusrész, mely felelős a kulcs generálásért és a titkosításért/visszafejtésért mutációs motornak hívjuk. A mutációs motor is módosul minden használat után.
  + Metamorfózisra képes vírus: Minden egyes alkalommal teljesen felülírják magukat. Megváltoztatják viselkedésüket és "kinézetüket" is.

Férgek:

* Terjedési technikák:
  + Kliens és szerver oldali szoftver sebezhetőségének kiaknázásával férnek hozzá más számítógépekhez.
  + Használhatnak hálózati kapcsolatokat, hogy rendszerről rendszerre terjedjenek.
  + Terjedhetnek megosztott médián keresztül, pl. USB drive, CD, DVD lemezek.
  + E-mail vagy üzenőfal férgek makrók és script kódok segítségével terjednek e-mail csatolmány vagy Instant Messenger csatolmányain keresztül.
  + Önsokszorosítóak, nincs szükségük gazdaprogramra.
* Távoli rendszer elérés:
  + Terjedés során általában a következőket hajtja végre:
    - Megfelelő hozzáféréseket keres host táblákban, címtárakban vagy cserélhető médiákat keres, melyekkel más gépekhez hozzáférhetnek.
    - Ezekkel a hozzáférési mechanizmusokkal lemásolják magukat távoli rendszerre és végrehajtódnak.
* Hálózati címek keresési stratégiák:
  + Véletlen: A fertőzött gazdagépek (hostok) véletlen IP címet használnak. Hatalmas Internet forgalmat eredményez, mely észrevehető, így lebukhat.
  + Hit-lista: A támadó összegyűjti a lehetséges sebezhető gépek címének egy hosszú listáját, és el is kezdi a lista elején levő címeket felhasználni. Minden fertőzött gép kap egy darabot a listából. Rövid ideig történik a próbálgatás, nehéz észrevenni.
  + Topológiai: A támadó a már fertőzött gép információit felhasználva talál további hostokat.
  + Helyi alhálózat: Tűzfal mögötti host próbál megfertőzni további hostokat a helyi hálózatban. Az alhálózat címstruktúráját használja.
* Modern férgek:
  + Multiplatform: Többféle platformot is támadnak, főleg a népszerű UNIX variánsokat, illetve többféle népszerű dokumentumok makró és script kódjait.
  + Multi-kihasználás: Többféle módú behatolási rendszer (Web szerverek, böngészők, e-mail, fájl megosztás, egyéb hálózati alkalmazások stb.)
  + Ultragyors terjedés: Optimalizálják terjedésüket, hogy rövid idő alatt minél nagyobb valószínűséggel minél több sebezhető gépet próbáljanak.
  + Polimorf: Vírusok polimorf technikáit használja.
  + Metamorf: Megváltoztatja viselkedését és "kinézetét".
  + Szállítási eszközök: Ideális többféle büntetési rutin terjesztésére, pl DDOS botok, gyökércsomagok, spam e-mail generátorok, és kémprogram.
  + Nulladik napi támadás: Azért, hogy maximális meglepetést és terjedést érjen el, a féregnek olyan ismeretlen sebezhetőséget kell kiaknázni, melyet a többség csak végrehajtás után ismer meg.
* Mobiltelefonos férgek:
  + Első féreg: Cabir worm (2004), Lasco and CommWarrior (2005)
  + Bluetooth vezeték nélküli hálózaton vagy MMS-en keresztül kommunikálnak.
  + Célpontok az okostelefonok (applikációk intallálása), Android és iPhone rendszerek.
  + Használhatatlanná teszik a telefont, adatok törlése, költséges SMS-ek küldése. Bár mobiltelefon férgek léteznek, de nagyobb mennyiségben fordulnak elő trójai alkalmazások, melyek magukat installálják.
* Mobil kód:
* Mobil kód egy program (pl. script, makró vagy más mozdítható kód), mely egy távoli rendszerről a helyire kerül, majd ott végrehajtódik a felhasználó utasítása nélkül.
* Vírusok, férgek, trójaiak számítógépre való továbbításának mechanizmusa.
* Más esetek: Kihasználva a sebezhetőségeket a mobil kód támad, pl. illetéktelen hozzáféréssel vagy megszerzi a legnagyobb jogosultságokat.
* Népszerű szállító eszközök: Java applets, ActiveX, JavaScript és VBScript.
* Drive-by-Downloads:
  + Cél: Felhasználói alkalmazások hibáinak (bug) kihasználása.
  + Böngésző sebezhetőségét használja ki. A felhasználó egy a támadó által kontrollált weboldalt megnéz, a felhasználó tudta nélkül a böngésző bug-ját kihasználva letölt és installál egy kártékony programot.
  + A támadó feltérképezi áldozatát, összegyűjti azon weboldalakat, melyeket gyakran látogat. Megvizsgálja ezen oldalakat, a sebezhetőekkel szemben drive-by-download támadást indít és vár míg az áldozat meglátogatja az oldalt. Más felhasználókra nem biztos, hogy hat.

Kliens oldali sebezhetőség:

* Malvertising: A támadó fizet, hogy hirdetések jelenjenek meg a célpont weboldalakon. A hirdetések kártékony programot tartalmaznak. A weboldalak látogatóit fertőzik meg.
* E-mail kliensek bug-jainak kihasználása: pl. Klez mass-mailing worm (2001) a Microsoft's Outlook és Outlook Express programokat támadta, hogy automatikusan végrehajtódjon.
* PDF-olvasó bug-jainak kihasználása: PDF-olvasó (sokszor kiegészítő modulok révén, multimédiás tartalmak megjelenítése) kártékony programot installál, amikor PDF dokumentumot nézünk. Ezek a dokumentumok e-mail spammel terjednek, vagy célzott adathalász támadással.

Kattintáseltérítéses támadás (Clickjacking):

* A támadó célja összegyűjteni a fertőzött felhasználói klikkeket.
* Például: "Akaratlan lájkolás": Funkciógombnak (jóváhagyás, eltávolítás, bezárás stb.) álcázott like gombok formájában a felhasználók üzenőfalára kikerülnek az üzenetek.
* Ráveszik a felhasználót, hogy tegyen bizonyos dolgokat, pl. kártékony tartalmat lájkolhatunk, és az így becsempészett rosszindulatú kódok villámgyorsan elterjednek az ismerőseink között, vagy egy weboldalt kell megnéznie stb., így installálnak kártékony programokat.
* Például: Adobe Flash vagy JavaScript segítségével lehet nyomógombokat beszúrni.
* Tipikus támadás az átlátszó rétegek alkalmazása, ráveszik a felhasználót, hogy kattintson egy linkre vagy gombra.
* Az iFrame technológiával megoldható, hogy egy láthatatlan, az egész ablakot beborító Like gombot húzzunk az oldalunk tartalma fölé, ahol bárhová klikkel a látogató, ha éppen be van jelentkezve Facebookra, azonnal megjelenik az üzenőfalán, hogy neki tetszett az oldal.
* Billentyű lenyomásokat is megszerezhetik. Jól megtervezett stíluslapok, iframe-ek, szöveges beviteli mezők félrevezethetik a felhasználót, aki az átlásztó rétegen adja meg jelszavát stb. a támadónak.

Pszichológiai támadás:

Spam e-mail:

* Ötlet: "átverni" és rávenni a felhasználókat, hogy saját rendszerüket veszélyeztessék vagy kiadjanak személyes információkat.
* A felhasználó megnéz vagy válaszol egy SPAM e-mailre, vagy megengedi, hogy felinstallálódjon vagy lefusson egy trójai program vagy script kód.
* A felhasználó aktív részvétele szükséges.
* Mai becslések szerint az e-mail forgalom 90% vagy annál több spam e-mail.
* Ez a jelentős növekedés fellendítette az anti-spam ipart, mely igyekszik detektálni és kiszűrni a spam e-maileket.
* Kis része a spam-eknek legális mail szerverekről küldődnek ki, nagy részük botnet-eken keresztül mennek, felhasználva a fertőzött (zombie) gépeket.
* A spamek nagy része csak reklám.
* Másrészük valamilyen kártékony programot hordoz: Csatolmány tartalmaz egy trójait vagy script kódot, melynek futtatásával megfertőzzük a rendszert.
* Spamek használhatók még adathalászatra is (spear phishing emails), tipikusan átirányítják a felhasználót egy a támadó által kontrollált weblapra, mely valamely valós szolgáltatás tükrözése.

Trójai lovak:

* A trójai ló egy olyan kártékony program, mely magukat hasznos programnak álcázzák, de képesek a háttérben segíteni egyéb ártó szándékú programok bejutását és működését a számítógépen.
* Nem sokszorozódnak.
* Mivel hasznos programnak mutatják magukat, ezért leggyakrabban a felhasználó tölti azt le fertőzött honlapokról a számítógépére. Emellett terjedhetnek e-mailben, vagy adathordozókon is.
* Többségük tartalmazza a hátsó kapu telepítését, ami a fertőzés után biztosítja a hozzáférést a célszámítógéphez.
* Büntető rutinok: valamely más kártékony program telepítése, kémprogramok, botnet.
* Három modell:
  + A program továbbra is folytatja eredeti funkcióját, de rajta kívül még valamilyen büntető rutint is futtat.
  + A program továbbra is folytatja eredeti funkcióját, de módosítja annak működését, hogy végrehajtsa a büntető rutint (pl. login program gyűjti a jelszavakat)
  + Teljes mértékben lecseréli az eredeti program funkcióját.
* Telefonra:
  + 2004-ben jelent meg elsőnek: Skuller
  + A trójaiak általában az alkalmazás áruházakból tölthetőek le. (pl. Good Weather időjárás előrejelzés alkalmazás Google Play áruházban(2017), már visszavonták, büntető rutin: banki adatok ellopása, képernyő lezárása, feloldása)
  + Több olyan mobil trójai (Ztorg, Gorpo, Leech) is van, mely root jogosultságokat használ. Az ilyen trójaikkal megfertőzött eszközök általában hálózatba szervezik magukat egyfajta reklám botnetet alkotva, amelyet a hackerek különféle reklámprogram (adware-ek) telepítésére használnak. Röviddel azután, hogy rootolják az eszközt a trójaik letöltenek és telepítenek egy hátsó ajtót. Majd letölt és aktivál két modult, amely képes letölteni, telepíteni és elindítani alkalmazásokat.

Büntető rutinok:

Tevékenységek, melyek végrehajtódnak a fertőzött rendszeren.

* Vannak olyan kártékony programok, melyeknek nincs büntető rutinjuk, céljuk a terjedés.
* Általában van büntető rutinjuk.
* Kategóriák:
  + Rendszer károsítása: adat megsemmisítése, Fizikai károkozás, logikai bomba
  + Támadó ügynökök: Botok, zombik
  + Információ lopás: billentyűzetfigyelők (keyloggers), adathalászat (phishing), kémprogramok (spyware)
  + Lopakodás: hátsóajtó (backdoor), gyökércsomag (rootkit)

Rendszer károsítása:

Adatok megsemmisítése:

* 1998, Csernobíl vírus: Windows-95 és 98-as vírus. Ahogy az indítási feltétel teljesül, törli az adatokat a merevlemez első megabájtjának bitjeit 0-ra állítva, komoly kárt okozva a teljes fájlrendszerben.
* 2001, Klez mass-levelező féreg: Windows-95-től XP-ig terjedő rendszerek. E-mailen keresztül terjed, másolatokat készít és küld önmagáról a címtárban levő email címekre. A merevlemezen levő antivírus programot és fájlokat törli.
* Ransomware: Lezárja a felhasználó fájljait, blokkolja az áldozat hozzáférését a számítógéphez, az okozott károk visszafordításáért minden esetben váltságdíjat követel, képesek lehetnek az áldozat érzékeny személyes adatainak megszerzésére (jelszavak), a védelmi szoftverek (antivírus, Anti-Spyware stb.) leállítására, megtévesztő figyelmeztetések megjelenítésére és más kéretlen tevékenységekre is. Gyakran "drive-by-downloads" útján terjednek.
  + Fajtái:
    - Fájltitkosító ransomware: Ez a változat főként trójai vírusok, email mellékletek, népszerű programok (Adobe Flash, Microsoft Office...stb) sebezhetőségeit kihasználva terjed. A rendszerbe jutva gyors sebességgel megkeresi és titkosítja a személyes fájlokat.
    - Nem titkosító ransomware: Ez a típus a teljes rendszert lezárja. Valamilyen bűnüldöző hatóság üzenetének adja ki magát (FBI, CIA, Nemzeti Nyomozó Iroda, Rendőrség) megkeresi az illegális fájlokat (torrentről vagy egyéb fájlmegosztó oldalról származó fájlok). Megtalált fájlokat mind bizonyítékként sorolja fel. Továbbá tájékoztatja az áldozatot arról, hogy ha nem fizeti ki a követelt büntetést, akár börtönbe is kerülhet.
    - Böngészőlezáró ransomware: Egy Javascript segítségével egyszerűen blokkolja a böngészőket, és figyelmeztető üzenetet jelenít meg. Többnyire illegális tevékenységeket sorol fel, és a börtönbüntetés elkerülése érdekében büntetés megfizetését kéri.
  + Példa:
    - WannaCry féreg - 2017. május:
    - Windows operációs rendszer EternalBlue nevű biztonsági résén keresztül (is) támad.
    - Titkosítja az adatokat, Bitcoin-ban kéri a váltságdíjat.
    - A sérülékenységeket kihasználó kód (EternalBlue kódnéven) a ShadowBrokers néven ismertté vált csoport által került ki az Internetre 2017. április 14-én, és a Microsoft március 14-ével javította az MS17-010 javítócsomag részeként.
    - Az első jelek leginkább egy spear-phishing kampányra utalnak, amelyben JavaScript és PowerShell kódot alkalmazó makrókkal ellátott MS O‑ce dokumentumot terjesztenek csatolmányként.

Fizikai sérülés:

* A büntető rutin fizikai eszközöket károsít meg.
* Csernobil vírus nemcsak az adatokat károsítja, de megpróbálja újraírni a BIOS kódot. A rendszer addig használhatatlan, míg a BIOS chip nincs újraprogramozva vagy kicserélve.
* Stuxnet féreg ipari működtető rendszert céloz meg. A féreg az eredeti rendszer kódját módosítja, mely a berendezés meghibásodását eredményezi.

Logikai bomba:

* Logikai bombák, mint büntető rutinok egy adott feltétel teljesülésével vagy valamely esemény bekövetkeztével "robbannak" (hajtják végre rutinjukat).
* Indíték: Bizonyos fájlok megléte vagy hiánya, valamely dátum vagy a hét valamely napjának bekövetkezte, vagy egy szoftver verzió konfigurálása, vagy valamely speciális felhasználó belépése stb.
* "Robbanás": adatok vagy teljes fájlok módosítása vagy törlése, rendszer leállása, illetve más károk okozása

Támadó ügynökök:

Büntető rutin: A kártékony kód (vírusok, trójaiak) módosítja a rendszert úgy, hogy a támadó irányítása alá vehesse, használhassa annak erőforrásait.

A számítógépet zombie számítógépeknek hívjuk.

A zombi számítógépek hálózatba kötve, koordináltan működhetnek. Ezt a hálózatot botnetnek nevezzük.

Ezek a büntető rutinok a fertőzött rendszer integritását és rendelkezésre állását támadják.

Zombik használata:

* Elosztott szolgáltatásmegtagadással járó támadás (Distributed denial-of-service (DDoS)): A DDoS támadás számítógépes rendszerekre vagy azok hálózatára irányul, azzal a céllal, hogy azok szolgáltatását teljesen vagy részlegesen megbénítsa, eltérítése a helyes működési módjától.
* Spam-ek küldése: Botnet - zombik ezrei - segítségével komoly mennyiségű kéretlen e-mailt lehet kiküldeni.
* Forgalomfigyelés: Zombikat a hálózati forgalom megfigyelésére is használhatjuk. Figyelik a rajtuk átmenő érdekes nyílt üzeneteket. Érzékeny információk megszerzése a cél (pl. felhasználói nevek és jelszavak).
* Billentyűzetfigyelés: A fertőzött gépek figyelik a billentyűzet használatát és kinyerik az érzékeny adatokat.
* Kártékony programok terjesztése: Botnetek alkalmasak új botok terjesztésére, melyek képesek valamely új fájl HTTP-n, FTP-n keresztül történő letöltésére, majd futtatására.
* Reklámok elhelyezése: Botnetek alkalmasak pénzszerzésre is. Vegyünk egy hamis Web site-ot, mely rengeteg reklámot tartalmaz. Vannak cégek, melyek fizetnek a reklámok megjelenítése, illetve a rájuk történő kattintások után. A támadók zombi gépekkel oldalakat látogattatnak és klikkeltetnek. Fejlesztett változat: a bot eltéríti a böngésző kezdőoldalát, így minden egyes alkalommal, ha a böngészőt megnyitják, gyűjtik a látogatottságot, klikkeket. Az adware-ek általában kapcsolódó eszköztárakat, más adware-eket és egyéb nemkívánatos, harmadik felektől származó programokat is hirdetnek. A felhasználók internetes tevékenységeinek megfigyelése, az érdeklődési körök, a leggyakrabban látogatott weboldalak, a beírt adatok stb. begyűjtése. A megfigyelt adatokat távoli szerverre továbbítják.
* Online szavazások/játékok manipulálása: Mivel minden botnak saját IP címe van, minden egyes szavazat számít

Információszerzés:

* A kártékony program a fertőzött gépeken tárolt adatokat gyűjti össze, melyeket a támadó felhasznál.
* Az adatok bizalmasságát támadják.
* Kategóriák:
  + Hitelesítő adatok (credential) ellopása, billentyűzetfigyelés, kémprogramok:
    - Billentyűzetfigyelők összegyűjtik a billentyűlenyomásokat, hogy a támadó érzékeny adatokat szerezzen be.
    - Mivel az összes szöveget begyűjtik, szükséges egy "beépített" szűrő, mely csak a megadott kulcsszóhoz kapcsolódó adatokat továbbítja.
    - Néhány bank alkalmazás átállt a grafikus appletek alkalmazására.
    - Válaszul a támadók általánosabb kémprogram büntető rutint használnak: böngésző előzmények figyelése, weboldalak éltérítése a támadó által kontrolláltakra, és a böngésző és bizonyos weboldalak között kicserélt adatok dinamikus változtatása.
    - Valamennyi személyes adatok megszerzésére irányulnak.
  + Adathalászat és személyazonosság ellopása
    - A felhasználó belépési adatainak megszerzésére irányuló másik megoldás, egy a támadó által kontrollált weboldalra mutató link elhelyezése egy kéretlen e-mailben.
    - Ez általában valamilyen sürgős üzenetben jelenik meg, mely a felhasználói fiók aktiválására kér.
    - Adathalászat: pszichológiai támadással kihasználják a felhasználók bizalmát úgy, hogy valamely megbízható felet személyesítenek meg.
  + Felderítés, kémkedés
    - A támadó bizonyos típusú információt szeretne megtudni.
    - Operation Aurora 2009-ben egy trójait használt, hogy hozzáférjen (esetlegesen módosítsa is) high tech és biztonsági cégek forráskódjaihoz.
    - A Stuxnet féreg 2010-ben összegyűtött számos hardver és szoftver konfigurációk részleteit, hogy meghatározza, hogy tudná-e a célpontokat támadni.
    - Célzott adathalászattal ipari kémkedéseket vagy más felderítéseket végeznek.
* Célzott adathalászat (spear-phishing attack):
  + Leveleket küldenek, egy olyan cég/személy nevében, akiben a címzettek megbíznak.
  + Talán 5-10 jól kiválasztott embernek küldik csak ki, a támadók tanulmányozzák a kiszemelt célpontokat, mint pl. azok LinkedIn vagy Facebook fiókjainak átolvasásával, vagy az általuk nyilvános blog-okon, fórumokon közölt üzeneteik vizsgálatával.
  + Egy nagyon testre szabott levelet készítenek, amely az érintett célpontok számára relevánsnak tűnhet. Így az egyének nagyobb valószínűséggel válnak a támadás áldozataivá.
  + A támadó kifejezetten a felhasználót, vagy a cégét akarja támadni, bizalmas információkhoz igyekeznek hozzáférni, mint pl. a cég üzleti titkai, az érzékeny technológiai tervek, vagy bizalmas kormányzati kommunikáció. Lehet, hogy egy másik embert/céget akarnak elérni rajtunk keresztül.

Lopkodás:

A kártékony programok azon rutinjai, melyekkel elrejtik jelenlétüket és hozzáférésüket a fertőzött gépeken.

Ez a rutin a rendszer integritását célozza.

Hátsóajtó (Backdoor):

* A program egy titkos belépési pontja, mely lehetővé teszi, hogy a belépési pont ismerője a biztonságos hitelesítés nélkül kapjon hozzáférést.
* Programozók legálisan használták a hátsóajtókat program tesztelésre. A hátsóajtók veszélyessé válnak, ha a tisztességtelen programozók illetéktelen hozzáférésre használják.
* A védelmi intézkedések szoftverfejlesztésekre és szoftver updatekre irányulnak.

Gyökércsomag:

* Egy programcsomag, mely installálása után fedett hozzáférést biztosít, tart fent a már fertőzött géphez adminisztrátori (root) jogosultságokkal. Hátsóajtó hozzáférést biztosítanak trójaiak számára úgy, hogy fontos rendszer fájlokat módosítanak.
* Az operációs rendszer valamennyi funkciójához és szolgáltatásához hozzáférést ad.
* Adminisztrátori jogosultsággal, a támadónak teljes kontrollja van a rendszer felett, felrakhat, módosíthat programokat, fájlokat, processzeket monitorozhat, küldhet, fogadhat hálózati forgalmat és hátsóajtó hozzáférést kaphat.
* A gyökércsomag elrejti ezeket a mechanizmusokat.
* Jellemzői:
  + Perzisztens: A gyökércsomag perzisztens helyen tárol kódot, pl. registry-ben és a kódot a felhasználó részvétele nélkül futtatja le.
  + Memória alapú: Nincs perzisztens kód, tehát újraindítást nem éli túl. Mivel csak a memóriában van, nehezebb detektálni.
  + Felhasználói (user) módú: Felhasználói szinten működik az operációs rendszerben. Lehallgatja az API hívásokat és módosítja kapott választ. (Példa: ha egy alkalmazás egy könyvtár listázását kéri, a lista, amit megkap nem tartalmaz olyan bejegyzéseket, melyek a rootkittal kapcsolatos fájlok beazonosítását segítik.)
  + Kernel módú: A gyökércsomag úgy rejti el jelenlétét, hogy módosítja a kernelt.

Kernel módú rootkitek:

* A rendszermag (kernel) az operációs rendszer alapja (magja), amely felelős a hardver erőforrásainak kezeléséért (beleértve a memóriát és a processzort is).
* A korai rootkitek felhasználói módúak voltak. A változtatásokat detektálni lehetett kernel kódokkal.
* A következő generációs rootkitek egy szinttel lejjebb kerültek, kernel szintű változtatásokat hajtottak végre.
* A felhasználói szintű programok a kernellel rendszerhívásokon keresztül teremt kapcsolatot. A kernel szintű rootkitek rejtőzködéséhez az elsődleges célpontok a rendszerhívások.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

* Techinkák:
  + Linux esetében minden egyes rendszerhíváshoz egy azonosító van rendelve.
  + A kernel egy rendszerhívási táblázatot kezel, minden egyes rendszerhíváshoz tartozik egy rekord. Minden rekord tartalmaz egy a megfelelő rutinra mutató mutatót.
  + Három technika:
    - Rendszerhívási tábla módosítása: A támadó a tárolt rendszerhívások címeit módosítja, átirányítva a rendszerhívás mutatóját a legális rutinról a rootkit rutinjára.

A képen szöveg, diagram, sor, képernyőkép látható

Automatikusan generált leírás

* + - Rendszerhívási tábla célpontjainak módosítása: A támadó a legális rendszerhívási rutint írja felül a kártékony kóddal. A tábla nem változik.
    - Rendszerhívási tábla átirányítása: A támadó a rendszerhívási táblára vonatkozó hivatkozásokat átirányítja egy másik kernel memóriabeli táblára.

Biztonsági intézkedések:

Megelőzés:

* Valamennyi biztonsági intézkedést "antivírus" mechanizmusnak hívunk, mivel elsőnek tipikusan a vírusok írtására jöttek létre.
* Ideális megoldás: megelőzés. Elsőnek ne engedjük, hogy a kártékony program a rendszerünkbe bejusson. Ha bejutott, blokkoljuk, ne módosíthassa a rendszert.
* Férgek, vírusok, trójaiak terjedésének megelőzése:
  + Sérülékenység csökkentése: Összes javítócsomag (patch) alkalmazása, a felszínre került programhibák orvosolására.
  + Fenyegetés csökkentése: A rendszer adataihoz, fájljaihoz megfelelő hozzáférés-vezérlés beállítása.
* Pszichológiai támadás megakadályozható megfelelő felhasználói felvilágosítással és oktatással.
* Külső adatmentés, szalagos mentés: offl­ine, szalagos mentési rendszer esetén a lementett adatokat a szalagos technológiából fakadóan a zsaroló vírus nem képes titkosítani, TÁROLJUK a havi mentéseket fizikailag is biztonságos külső helyen.
* Folyamatos frissítések, többrétegű vírusvédelem: többrétegű védelemmel ellátott vírusírtó, frissítése, tűzfal.
* Szabályozás kialakítása és betarttatása.
* Oktatás, tudatosítás: Ismeretlen feladótól érkezett e-mailekben ne nyissuk meg a mellékletet, főleg, ha ez tömörített állomány. Munkahelyünkön figyelmeztessük azon kollégáinkat a veszélyekre, akik főleg külső irányból kapják leveleik többségét (pl. pénzügyi vagy HR osztály). A felhasználóknak ismerniük kell a vírus működését és terjedését, valamint azokat a technikákat, amelyeket a kártékony kód alkalmaz.

Technikai mechanizmusok, ha a megelőzés nem sikerül:

* Detektálás: Ha a kártékony kód már megfertőzte a rendszert, határozzuk meg a helyét.
* Azonosítás: Ha detektáltuk, azonosítsuk be a kártékony kódot, mely megfertőzte a rendszerünket.
* Eltávolítás: Ahogy azonosítottuk, távolítsuk el a kártékony kód nyomait valamennyi fertőzött állományból, hogy ne terjedjen tovább.

Ha sikerült detektálni, de se azonosítani, se eltávolítani nem tudjuk, akkor töltsünk be a tiszta backup verziót.

Antivírus szoftver:

* Első generációs: egyszerű szkennerek: a kártékony program valamennyi másolatában ugyanaz a struktúra és bitminta kell, hogy legyen. Másik típus a program állományok méretét tárolja és figyeli azok változását.
* Második generációs: heurisztikus szkennerek: A szkenner a kártékony kódhoz kapcsolható programkód részleteket keresi. (pl. polimorf vírusok esetén a titkosítási részből próbálják kinyerni a kulcsot.) Másik megoldás az adatintágritás vizsgálata. Egy ellenőrzőösszeg (hash vérték, MAC) adható meg minden egyes programhoz. Ha a kártékony kód módosítja a programot az ellenőrzőösszeg változtatása nélkül, akkor a változás detektálható. A kulcs külön tárolandó.
* Harmadik generációs: aktivitást figyelők: olyan programok, melyek a kártékony kód tevékenységét figyelik, nem a fertőzött programbeli struktúráját. Elegendő néhány tipikus kártékony kód tevékenységet felismerni.
* Negyedik generációs: teljes védelem: Ezek a programcsomagok többféle antivírus megoldást alkalmaznak.
* Fejlett antivírus szoftverek: általános visszafejtés (generic decryption (GD)): Ha egy polimorf vírust tartalmazó fájlt futtatunk, a vírusnak vissza kell fejtenie magát. Az általános visszafejtő szkenner detektálja ezt.
  + Viselkedés-blokkoló szoftver valós időben figyeli a kártékony programok működését és megpróbálja blokkolni mielőtt komoly károkat okoz. A következő eseményeket figyeli:
    - Fájlok módosítására, törlésére, megnyitására irányuló próbálkozások;
    - Merevlemez formatálása vagy más javíthatatlan művelet;
    - Indítási beállítások módosítása;
    - Futtatható tartalmak küldése e-mailen vagy messenger-en keresztül;
    - Hálózati kommunikáció kezdeményezése;
  + A kártékony kód már elkezdi működését mielőtt a tevekénységét detektáljuk.
* Rootkit védelmi intézkedések:
  + A gyökércsomagokat nehéz detektálni és megsemmisíteni főleg, ha kernel módúról van szó.
  + Javítócsomagok (patch), update-k használata.
  + Admin jelszavak megváltoztatása. Ha egy kernel szintű rootkitet detektálunk az egyetlen és megbízható megoldás, ha teljes operációs rendszert újrarakjuk.

10.Fólia

Szolgáltatásmegtagadással járó támadás (Denial-of-service attack - DoS):

A támadás valamely szolgáltatás rendelkezésre állását veszélyezteti.

Kategóriák:

* Sávszélesség elleni támadás: Az Internet és a szerver közötti hálózati kapcsolat szűk kapacitását használja ki. A támadónak nagyobb sávszélességgel kell rendelkeznie, mint amilyen az áldozaté.
* Rendszer erőforrások elleni támadás: A rendszer erőforrásait célozza, úgy túlterhelik, hogy a hálózat kezelő rendszer összeomlik. pl. SYN flood támadás a memóriát terheli
* Alkalmazási rétegbeli támadás: Speciális alkalmazásokat célozza meg. Pl.: SMTP (email) szerver nagy mennyiségű e-mailt kap, megtöltve a háttértárat, amely a további levelek fogadását lehetetlené teszi (email bomb). Másik lehetőség a Web szerver terhelése jónéhány valós kéréssel (pl. adatbázis lekérdezések).
* Ping of death támadás: Pingen alapul, melynél 65353 bájt csomag lehet a legnagyobb méretű csomag. A támadó ettől nagyobb csomagokkal bombázza az áldozatot, megtelik a memóriaterület (Buffer), és ez összeomláshoz vezet.
* Elárasztásos (flooding) támadások:
  + ICMP Flood (Internet Control Message Protocol): Echo (ping) parancson alapul. Csomagokat küldünk hamis forráscímekkel addig, míg az áldozat válaszolni tud. Egy idő után nem tud válaszolni.
  + SYN Flood (synchronization): A TCP protokoll három-utas kézfogásos üzenet visszaigazolási rendszerét használja ki. A célpont kapcsolatok állapotát tároló memóriája megtelik, így nem lesz képes új TCP kapcsolatot létrehozni.
* IP forrás cím spoofing: Hamis forrás cím megadásával a válasz csomagok nem térnek vissza a támadóhoz. Vannak hamis forrás címek, melyek válaszolnak error csomaggal, illetve a célt nem érő csomagok visszaküldődnek, növelve a forgalmat.

Három utas kézfogás:

* A kliens TCP kapcsolatot kezdeményez a szerverrel a SYN csomag elküldésével.
* Ez a csomag tartalmazza a kliens címét, portját, valamint egy kezdősorszámot. A szerver kezdeményezés valamennyi adatát tárolja egy TCP kapcsolat táblában. SYN-ACK csomag tartalmazza a szerver új sorszámát és növeli a kliens sorszámát.
* Ahogy a kilens megkapja a csomagot, küld egy ACK csomagot a szervernek a megnövelt szerver sorszámmal és a kapcsolatot létrejöttnek állítja be.
* Amikor a szerver megkapja az ACK csomagot, a kapcsolatot létrejöttnek állítja be.
* Mind a kliens, mind a szerver nyomon követi mely csomagokat küldte el, ha nem kap választ, újraküldi a csomagokat.

A képen szöveg, képernyőkép, diagram, tervezés látható

Automatikusan generált leírás

SYN flood támadás:

A támadó számos SYN kapcsolat kezdeményező csomagot generál, melyben hamis forrás IP-címek vannak. (spoofing)

Minden egyes kérésre a szerver eltárolja az adatokat és küld egy SYN-ACK csomagot a megadott forrás címre.

Ha forrás című rendszer elfoglalt, vagy ha nincs, akkor a szerver nem kap válasz csomagot és újra elküldi a SYN-ACK csomagot, jó néhányszor. Végezetül, feltételezve, hogy a kezdeményezés sikertelenné vált törli a kezdeményezés adatait.

A támadó nagy mennyiségű hamis kapcsolat kezdeményezést küld a szerver felé. Ahogy a tábla telítődik, a valós kezdeményezéseket visszautasítja.

A képen szöveg, képernyőkép, sor, diagram látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, képernyő, tervezés látható

Automatikusan generált leírás

Elosztott DoS (Distributed Denial-of-Service) támadás:

* A támadó egyetlen rendszer helyett, több rendszert is használ a támadáshoz.
* Ezek a rendszerek tipikusan fertőzött rendszerek: zombigépek, melyek botnetet alkotnak.
* A támadók egy kontrollált hierarchiát alkotnak. Kevés számú kezelő zombi irányítja a nagyobb számú ügynök zombikat.
* Ahogy egy ügynök szoftver feltöltődött egy fertőzött gépre, egy-két kezelő géppel egyből közli, hogy rendelkezésre áll.
* A legjobb védekezés a megelőzés.

A képen diagram, vázlat, sor, tervezés látható

Automatikusan generált leírás

Védelmi intézkedések:

* Tűzfalak használata: Amennyiben a tűzfalat lehet úgy konfigurálni, hogy a lehető legtöbb rosszindulatú forgalmat eldobja, akkor legalább maga a szerver visszakaphatja kiszolgáló és feldolgozó teljesítményének egy részét. A tűzfal alkalmasságát a rosszindulatú forgalom a többitől való megkülönböztetésének képessége adja. Lehetnek bizonyos körülmények a támadás szerkezetében, amik aktív szerepet játszhatnak a rosszindulatú és szabályszerű forgalom szétválasztásában.
* Célszerű létrehozni tükör másolatot. Igen hatékonyan lehet egy szerver tartalmának formáját visszaállítani és azt tetszőleges sebességgel elérhetővé tenni.
* Megosztott tartalom: Az adatok terjesztés (elosztás) szempontjából mindig rendelkezésre állnak megosztott módon. Nem egy szerver, hanem több szerver is válaszol a kérésekre

Tűzfal:

Tűzfal alatt olyan biztonsági rendszert értünk, amely a számítógépes hálózatok kapcsolódási pontján helyezkedik el, és minden átmenő (kimenő és bejövő) hálózati forgalom figyelésével, szűrésével nyújt védelmet.

A teljes hálózati forgalmat képes figyelni, akár csomagonként átvizsgálva azt, hogy átengedje-e azokat, vagy veszélyesnek minősítse, és megakadályozza a továbbításukat.

A kimenő csomagok ellenőrzésére is sor kerülhet, így egyszerűen lehet megakadályozni a belső hálózatunkból induló támadásokat, vagy a nem kívánt hálózati helyek elérését. (pl. nem lehet a közösségi oldalakat elérni)

A tűzfalak védik a helyi hálózatunkat hálózat-alapú támadásoktól.

* Elhelyezkedésük alapján:
  + Személyes (szoftveres) tűzfal: Egyedi számítógépekre telepített tűzfal alkalmazás, amely szoftveresen fog védeni, de csak azt az egy gépet.
    - Jellemzői: operációs rendszer része, alapbeállítással is hatékonyak, minden számítógépen külön el kell végezni a beállításokat
  + Központi (hardveres) tűzfal: Nem egy számítógépen fut a szolgáltatás, hanem egy speciális hardveren(router), ezért ezt a módszert hívjuk hardveres tűzfal szolgáltatásnak is. A teljes belső hálózatunkat védhetjük.
    - Jellemzői: olcsóbb, egyszerűbb hálózatmenedzsment, központi kezelhetőség

Mit várunk el?:

* Jogosulatlan felhasználókat a védett hálózaton kívül tarthatjuk, kockázatos szolgáltatások be- és kikerülését megakadályozza, védelmet nyújt IP spoofing és routing támadások ellen.
* Biztonsággal kapcsolatos események monitorozása (audits, figyelmeztetések).
* Nem biztonsággal kapcsolatos funkcionalitás biztosítása: NAT (network address translator) hálózati cím átalakítása, hálózatmenedzsment funkciók: Internet használat figyelése audit vagy log fájlok alapján.

Korlátai:

A központi tűzfal nem véd a belső, tűzfal mögötti támadástól, ha a hálózatunkon belül található már fertőzött, feltört számítógép, esetleg rosszindulatú felhasználó, aki belülről támad. (Megoldás: személyes tűzfal)

Kintről behozott fertőzött laptop, PDA, vagy hordozható tároló eszköz használata esetén szintén nem véd a központi tűzfal.

Típusai:

* Csomagszűrő tűzfalak (Packet Filtering Firewall):
  + A csomagszűrő tűzfalak szabályok halmazát(házirendet) alkalmazzák minden egyes bejövő és kimenő IP csomagra, mely alapján vagy továbbítják, vagy eldobják a csomagot. (pl. forrás és cél IP cím, port száma)

A képen képernyőkép, szöveg, szám, Betűtípus látható

Automatikusan generált leírás

* Állapot szerinti szűrés:
  + Ez a módszer alapvetően a kapcsolatorientált protokollok esetén nyújt újabb lehetőségeket. A tűzfal azonosítani tudja egy kapcsolat kezdetét és végét, valamint a kettő között zajló adatforgalmat. Csomagokat tárolja átmenetileg, amíg a szolgáltatás a döntéshez szükséges adatokat össze nem gyűjti. Néhányuk tárolja a TCP sorszámot, hogy megelőzze a session hijacking támadást.

A képen szöveg, képernyőkép, szám, nyugta látható

Automatikusan generált leírás

* Alkalmazásszintű tűzfal (proxy):
  + A tisztán csak a forgalomhoz tartozó, mint a forrás, cél és szolgáltatás adatokon kívül a hálózati csomagok tartalmát is figyeli.
  + A proxy megoldások azzal a céllal készültek, hogy kiküszöböljék a felhasználók kiszolgálókra való bejelentkezésből adódó kényelmetlenségeket, illetve az ebből fakadó veszélyeket.
  + A kliensek és a kiszolgálók között nem épül fel közvetlen kapcsolat, hanem mindketten a tűzfalon futó proxy alkalmazással kommunikálnak.

11.Fólia

Fizikai védelem:

A fizikai védelem feladata azon fizikai erőforrások védelme, melyek az adatok tárolását, feldolgozását, továbbítását biztosítják. A védelmi intézkedések többsége preventív vagy detektív.

* Fizikai infrastruktúra:
  + Informatikai rendszer hardver elemei: Adatfeldolgozó és tároló eszközök, adatátviteli és hálózati elemek és offl­ine tároló eszközök. Ide soroljuk az informatikai rendszer dokumentációit is.
  + Épületek: Épületek, ahol az informatikai rendszer fizikai elemei megtalálhatóak.
  + Kiszolgáló rendszerek: Elektromos vezetékek, kommunikációs hálózatok, víz- és gázvezetékek.
  + Személyzet: Azon személyek, melyek az informatikai rendszer használói, fenntartói vagy működtetői.

Fizikai fenyegetések kategóriái:

Környezeti fenyegetések, természeti csapások:

* Tornádó/forgószél:
  + szerkezeti kár, épületek tetejét veszélyezteti, kültéri berendezések sérülése, elvesztése
  + a szél és repülő tárgyak okozhatnak kárt
  + helyi közmű, és kommunikáció ideiglenes elvesztése
  + a közmű szolgáltatások gyors helyreállítása követi
* Trópusi/ciklonok:
  + hurrikánok, trópusi viharok, tájfunok
  + jelentős szerkezeti károk és kültéri berendezések sérülése
  + közmű szolgáltatások, kommunikáció sérülése
  + személyzet vészhelyzeti ellátása szükséges, generátor szükséges
* Földrengés:
  + teljes rombolás, jelentős, hosszú ideig fennálló kár informatikai rendszer épületeiben
  + hardver, közmű, infrastruktúra megsemmisülése, megemelt padlók összeomlása
  + személyzetet veszélyeztetik a törött üvegek, repülő tárgyak
* Jégvihar:
  + informatikai rendszer épületeinek megrongálódása
  + közmű és kommunikáció megsérülése
* Villám:
  + egyszerű becsapódástól a katasztrófáig terjedhet elektromos vezetékek megsérülése, tűz keletkezhet
* Árvíz:
  + árterületen, illetve alacsony szinten levő berendezések veszélyeztetettek hosszú ideig tartó sérülés, komoly takarítás szükséges
* Nem megfelelő hőmérséklet:
  + A legtöbb számítógépes rendszert 10 és 32 fok közötti hőmérsékleten kell tárolni.
  + Ezen az intervallumon kívül az erőforrás továbbra működőképes, de lehet, hogy nem megfelelő eredményeket ad.
  + Ha a környezet hőmérséklete nagyon magas lesz, a számítógép nem lesz képes megfelelően hűteni magát és a belső komponensek sérülhetnek.
  + Ha a hőmérséklet túl alacsony, bekapcsolásnál a rendszer hőtani sokkon esik át, mely integrált áramkörök sérüléséhez vezethet.
  + Okostelefonok, digitális kamerák, táblagépek és laptopok stb. akkumulátorainak kapacitása is csökken, ha túl meleg vagy túl hideg van.
  + Az eszköz belső hőmérséklet jelentősen nagyobb, mint a szoba hőmérséklete. Saját hűtésük külső feltételektől is függ.
* Magas páratartalom:
  + Magas pára korróziót okozhat.
  + A vízcsepp a mágneses és optikai tárolókat is veszélyezteti.
  + A vízcseppek zárlatot okozhatnak az alkatrészekben.
* Sztatikus elektromosság:
  + A sztatikusan feltöltött személyek, tárgyak kárt okozhatnak az elektromos eszközökben.
  + Már a 10 voltos kisülések is kárt okozhatnak az áramkörökben.
  + Több száz voltos kisülések jelentős kárt okozhatnak.
  + Az emberi test jóval több elektromos ellenállás tárolására alkalmas, mint egy átlagos IC. Az emberi sztatikus kisülések több ezer voltot is elérhetik.
* Tűz, füst:
  + emberi életre és a berendezésekre is vonatkozó fenyegetés
  + a közvetlen láng és a hő is veszélyforrás mérgező gázok felszabadulása veszélyes az emberekre nézve
  + tűzoltósból keletkező vízkár
  + füstkár
* Víz:
  + a számítógépes eszközöket, papírokat, elektromos tároló eszközöket veszélyezteti
  + elektromos rövidzárlat keletkezhet
  + mozgó vizek: vízvezetékek, eső, hó, jég okozta víz
  + ha vízhálózat két szinttel van feljebb, akkor már nem annyira kockázatos
* Por:
  + általában ezzel a fenyegetéssel kevésbé foglalkozunk
  + a por és kosz mindenütt megtelepszik.
  + szellőzőréseken át bejutó por eltömíti a levegő szabad áramlásának útját, ezért a belső ventilátor nem tudja kellő hatékonysággal hűteni a működése során forróvá váló processzort.
* Rovarfertőzés:
  + élő organizmusok is fenyegetések: rovarok, penész, rágcsálók
  + penész mind a személyzetre, mind a berendezésekre is vonatkozó veszélyforrás

Technikai fenyegetések:

* Feszültséghiány:
  + a berendezés kevesebb feszültséget kap, mint amennyire szüksége van a normál működéshez.
  + a legtöbb számítógép ellenáll a kb. 20%-os feszültséghiánynak, még nem áll le, nem történik működésbeli hiba.
  + nagyobb feszültséghiány leállítja a rendszert.
  + komolyabb kár sérülés nem keletkezik, csak a szolgáltatás szakad meg.
* Túlfeszültség:
  + áramszolgáltatási anomáliák, villámcsapás okozhatja
  + processzorokban, memóriákban okozhat kárt
* Elektromágneses Interferencia:
  + elektromos eszközök, más számítógépek elektromos zajt generálnak, mely kárt okozhat a saját számítógépünkben.
  + ez a zaj a térben és elektromos vezetékeken is továbbítódik.
  + zaj eredhet a közeli mikrohullámú antenna, vagy akár mobiltelefon révén is.

Emberi fenyegetések:

* Jogosulatlan fizikai hozzáférés:
  + szerverek általában lezárt területen vannak, ahova való bejutás korlátozott. Néhány alkalmazottnak van jogosultsága.
  + jogosulatlan fizikai hozzáférés lopáshoz, vandalizmushoz és visszaéléshez vezethet.
* Lopás:
  + berendezések eltulajdonítása, adatok megszerzése
  + csatorna lehallgatása is ide tartozik
* Vandalizmus:
  + berendezések tönkre tévese
* Visszaélés:
  + az erőforrások jogosulatlan használata

Fizikai preventív kontroll:

Általános preventív védekezés: felhők használata

* lokálisan lényegesen kevesebb erőforrásra van szükség
* a nagy mennyiségű adatok lokálisan nincsennek fizikai fenyegetéseknek kitéve

A képen szöveg, képernyőkép, diagram, tervezés látható

Automatikusan generált leírás

Környezeti fenyegetések:

* Nem megfelelő hőmérséklet és páratartalom:
  + Mérőeszközök segítségével a megfelelő környezetet el lehet érni.
  + Ha az érték túllép a megengedett határon, akkor jelez is.
* Vízkár:
  + Vízérzékelők elhelyezése a padlón és az emelt padlók alatt.
  + Víz esetén automatikusan le kell, hogy kapcsolódjon az áram.
* Por:
  + Ventilátor szűrő karbantartása és a helyiség tisztán tartása.
* Tűz, füst:
  + Tűzjelzők, megelőző intézkedések, tűz oltása
  + Ritkán keletkezik katasztrofális tűz egy jól védett számítógépes helyiségben.
  + Úgy kell a helyiséget kiválasztani, hogy minimális legyen a környezetében keletkező tűz, víz, füst kockázata.
  + Védelmi intézkedések: Közös falak legalább egy óra hosszat tűzállók legyenek.
  + Légkondicionálók úgy legyenek megtervezve, hogy a tüzet ne terjesszék.
  + Gyúlékony anyagokat ne tároljunk a helyiségben.
  + Kézi tűzoltókészülék legyen elérhető, egyértelműen jelezve, és rendszeresen tesztelt.
  + Automata tűzoltó rendszer is legyen telepítve.
  + Tűzjelzők vészjelet adjanak le a helyiségben és külső felügyeletnek is.
  + Főkapcsoló szükséges és egyértelműen jelezve legyen.
  + Menekülési útvonalak ki legyen függesztve.
  + Fontos adatok, dokumentumok tűzálló kabinetben legyenek.
  + Az adatok, programok up-to-date másolata más helyiségben legyen.
  + Biztosítási cégek, tűzoltóság vizsgálja át az épületet.

Technikai fenyegetések:

* Elektromos teljesítmény, Elektromágneses interferencia:
  + Szünetmentes tápegység kapcsolása minden egyes kritikus berendezéshez.
  + Szünetmentes tápegység elektromos áramot biztosít, ha megszűnik a hálózati áramforrás, áramingadozás van, vagy áramszünet lép föl.
  + A szünetmentes tápegység áthidalási ideje néhány perctől pár óráig terjed. Hosszabb kimaradások esetén generátor szükséges.

Emberi fenyegetések:

* Csak az arra jogosult léphet be az épületbe. Nem vonatkozik az alkalmazottakra, jogosulatlan belső támadókra.
* Erőforrásokat tegyük zárható tárolókba, széfekbe, szobákba.
* Berendezéseket rögzítsük olyan tárgyakhoz, melyeket nem lehet elmozdítani.
* Mozdítható berendezéseket nyomkövetővel láthatunk el, mely jelzi, ha elhagyja a területet és állandó monitorozást tesz lehetővé.
* A megfigyelő rendszer része az épületnek. Ezek a rendszerek valós idejű távoli megfigyelést és rögzítést jelent.

Fizikai helyreállító kontroll:

Hasonlít a korrektív kontrollhoz, csak komolyabb helyzetekben alkalmazzuk.

* A legfontosabb helyreállító intézkedés a másolatok készítése: Backups.
* A másolatok nem védenek az esetleges bizalmassági sérülésekkel szemben, de az adatok visszaállíthatóak.
* Hot site: Közel valós idejű szinkronizálással készített másolat, mely képes egyből átvenni a szolgáltatást.
* A víz, a füst, a tűz maga után hagy maradványokat, melyeket el kell takarítani. Speciális tisztítókat kell hívni.

12.Fólia

Emberi beavatkozás elleni védelem:

* Felhasználó/ügyintéző:
  + szűk jogkör
  + adatbevitelt végez
  + a hozzárendelt, korlátozott erőforrásokat használhatja
  + ha gyakorlatlan, akkor például nem érzi a hiteles azonosítás fontosságát (egyszerű jelszó, stb)
* Üzemeltető:
  + széleskörű ismerete és jogosítványa van a rendszer felhasználásával kapcsolatban
  + titoktartási kötelezettsége van
  + beállítások végrehajtása, módosítása
  + felhasználók nyilvántartása, jogosultságok beállítása, jogosultságok életciklusának nyomkövetése
  + utasítások alapján dolgozik
* Mérnök:
  + magas szintű informatikai végzettsége van
  + bizalmas információk – titoktartási kötelezettsége van
  + rendszer beállítása, módosítása, javítása
  + széles hatáskör
* Programozó:
  + rendszer készítése teljes hatáskör
  + komoly minőségbiztosítási rendszerrel felügyelik a programot

Emberi tényezőre visszavezethető veszélyek:

* Szándékos károkozás:
  + behatolás az informatikai rendszerek környezetébe,
  + illetéktelen hozzáférés (adat, eszköz),
  + adatok- eszközök eltulajdonítása,
  + rongálás (gép, adathordozó),
  + megtévesztő adatok bevitele és képzése,
* Nem szándékos, illetve gondatlan károkozás:
  + figyelmetlenség (ellenőrzés hiánya),
  + szakmai hozzá nem értés,
  + a megváltozott körülmények figyelmen kívül hagyása,
  + vírusfertőzött adathordozó behozatala,
  + pszichológiai támadásokban való részvétel,
  + biztonsági követelmények és gyári előírások be nem tartása,
  + adathordozók megrongálása (rossz tárolás, kezelés),
  + a karbantartási műveletek elmulasztása.

Ügyviteli védelem:

* Az informatikai rendszert üzemeltető szervezet ügymenetébe épített
  + védelmi intézkedések,
  + biztonsági szabályok és
  + tevékenységi formák együttese.
  + más néven ez az adminisztratív védelem
  + a szabályozás alapját a törvények és jogszabályok jelentik, ezeket pontosítják.
* Két szintje van:
  + stratégiai, tervezési szint:
    - Informatikai Biztonsági Koncepció (IBK)
  + mindennapi gyakorlatot érintő és szabályozó szint:
    - Informatikai Biztonsági Szabályzat (IBSZ)
* ez a két szint szorosan összefügg egymással és a szervezet más szabályaival
* a szabályozás uniformizál, csökkenti a kreativitást, kényelmetlenségeket okoz
* például előírhatjuk, hogy ki jogosult a hardver- vagy szoftverhibák elhárítására, s az ügyintézőnek meg kell várnia az illetékest, hiába tudná esetleg ő is elvégezni a javítást
* biztonságot is nyújt a szabályozás a dolgozóknak: a szabályok betartása esetén nem lehet őket felelősségre vonni
* ha a szabályozás nem megfelelő, akkor a rendszergazdákra hárul a felelősség
* a szabályozás eredményességéhez hozzájárul a dolgozók együttműködése
* nemcsak betanulási időszakban, hanem rendszeresen képezni kell a munkatársakat

Informatikai Biztonsági Koncepció (IBK):

A szervezet felső vezetésének informatikai biztonsággal kapcsolatos stratégiai elképzeléseit foglalja össze.

A koncepció tartalmazza a szervezet informatikai biztonságának követelményeit, az informatikai biztonság megteremtése érdekében szükséges hosszú távú intézkedéseket, ezek kölcsönhatásait és következményeit.

Részei:

* a védelmi igény leírása: jelenlegi állapot, fenyegetettségek, fennálló kockázatok, fenyegetések
* az intézkedések fő irányai: a kockázatok menedzselése,
* a feladatok és felelősségek meghatározása és felosztása a védelmi intézkedésekben,
* idő- és költségterv a megvalósításra és időterv az IBK felülvizsgálatára.

Koncepció elkészítésének fő szakaszai:

* Védelmi igény feltárása:
  + lényeges informatikai rendszerek és alkalmazások kiválasztása (amiket védenénk)
  + vállalat szerverei, tárolóegységei, lokális hálózata
  + a vállalat tágabb környezete: világhálón való megjelenés, beszállítókkal, alvállalkozókkal szembeni követelmények
  + az aktuális, a közép, esetleg a hosszú távú technológiai és szervezeti fejlesztéseket és változtatásokat is át kell gondolni
* Fenyegetettség elemzés:
  + veszélyforrások feltárása
  + a rendszerek gyenge pontjai
  + belső munkatársak
  + az eszközökhöz és az adatokhoz való hozzáférés irányelveit is meg kell határozni
  + hozzáférések naplózásának módja
* Kockázatelemzés:
  + a fenyegető tényezők, károk hatása az informatikai rendszerekre és a szervezetre
  + lehetséges károk várható bekövetkezési gyakorisága
  + kárérték
  + ezeknek függvényében a szükséges védelem technológiáját és mértékét
* Kockázat menedzselés:
  + veszélyforrások elleni védekezés módjainak kiválasztása (preventív védelem)
  + intézkedési tervek, illetve ezek hasznossága, költsége
  + váratlan helyzetekre adott lépések meghatározása
  + felelősök kijelölése, felelősségi körük definiálása
  + időterv az intézkedések bevezetésére
  + intézkedések hatása felülvizsgálatának ütemezése
  + IBK felülvizsgálatának ütemezése

Informatikai Biztonsági Szabályzat (IBSZ):

Célja:

* az informatikai rendszer alkalmazása során biztosítsa az adatvédelem elveinek, az adatbiztonság követelményeinek érvényesülését, s megakadályozza a jogosulatlan hozzáférést, az adatok megváltoztatását és jogosulatlan nyilvánosságra hozatalát.
* Az informatikai vezető és az informatikai biztonsági ellenőr készíti el, a vállalat vezetője adja ki.
* Az általános élethelyzetekre vonatkozik.
* A szervezet többi szabályzatát is figyelembe kell venni.
* Rendszeresen felül kell vizsgálni.

Személyi hatálya:

* kiterjed a vállalat informatikai szolgáltatásaiban részt vevő munkatársaira (szolgáltató, felhasználó).
* Különösen fontosak a rendszer-, valamint az adatgazdák jogait és kötelességeit meghatározó fejezetei

Tárgyi hatálya:

* vállalat tulajdonában lévő, illetve általa használt számítástechnikai berendezések, szoftverek, adatok, adathordozók, dokumentációk.
* passzív adatátviteli vonalak (Ethernet, Token Ring, FDDI, ATM szegmensek, optikai és hagyományos összeköttetések), csatlakozók
* hálózati aktív elemek (repeaterek, bridge-k, switchek, routerek, transceiverek, modemek, terminálszerverek)
* minden hálózatra kötött számítógépes munkahely (PC, workstation, terminál, hálózati nyomtató) és szerver

Biztonsági fokozata:

* Alapbiztonság: általános informatikai feldolgozás (nyilvános és személyes adatok)
* Fokozott biztonság: szolgálati titok, átlagos mennyiségű különleges adat (bizalmas adatok)
* Kiemelt biztonság: államtitok, nagy mennyiségű különleges adat. (titkos adatok)

Felelősségi rendszer megadása:

* Adatvédelmi felelősök megadása: pl. informatikai vezető, rendszergazda
* Adatvédelmi felelősök feladatainak megadása
  + Példák:
    - IBSZ alkalmazásának módja: oktatás, munkaköri leírások stb.
    - IBSZ karbantartása
    - Védelmi rendszer ellenőrzése
* Rendszergazda feladatainak megadása
  + Példák:
    - Gondoskodik a folyamatos vírusvédelemről
    - Felelős a vállalkozás informatikai rendszer hardver eszközeinek karbantartásáért
* Felhasználók feladatai
  + Példák:
    - Ismerniük és be kell tartaniuk a szabályzatot, részt kell venniük adatvédelmi oktatáson
* A felelős joga és kötelessége:
  + Az előírások betartásának ellenőrzése
  + Figyelmeztessen a betartásra
  + Fegyelmi felelősségre vonást kezdeményezhet
  + Az általa észlelt, tudomására jutott veszélyhelyzetnek megfelelően védekező lépéseket meghozza
    - Az adott gépek külső elérhetőségének letiltása
    - Az adott gépek hálózatról való eltávolítása
    - Egyebek
    - Az incidensekről beszámolót készíteni, azt az illetékes vezetőknek eljuttatni
* Minden felhasználónak kötelessége
  + Betörésgyanús esetek jelentése a biztonsági csoport felé
  + Együttműködés a károk elhárításában

Védelmi intézkedések:

* Infrastruktúra védelme:
  + Eszközök megközelítése az épület belseje felől
  + Kulcs, naplózott belépés
  + Rács, biztonsági üveg a földszinti ablakokon
  + Ki- és beviteli engedélyek az eszközökre
  + Hibaelhárítás felelősei, módjai
  + Selejtezés módja (adatok végleges törlése)
  + Azokat az adathordozókat, amelyeken érzékeny adatokat tároltak nem ajánlatos törlés után tovább adni, hanem ellenőrzött módon meg kell semmisíteni!
* Hozzáférési jogosultságok meghatározása/Felhasználói jogok kezelése:
  + Jogok kiosztásáért felelős ≠ Végrehajtásért felelős
  + Naplózni és visszakereshetővé tenni minden akciót
  + Felhasználói életciklus
  + Intézkedési terv az illetéktelen hozzáférés, illetve a jogosultságokkal való visszaélés eseteire
  + Biztonsági eseménynapló
  + Automatikus naplózás
* Hozzáférési jogosultságok meghatározása/Felhasználói jogok kezelése:
  + A rendszert csak illetékes vezető engedélyével szabad megváltoztatni
  + Külső személy a kezelt adatokhoz nem férhet hozzá
  + Jelszómenedzsment
  + Felhasználók listájának rendszeres aktualizálása
  + Ideiglenesen v. tartósan távol levő munkatárs helyettesítése
  + Külső partnerek hozzáférési jogosultsága (federation)
* Szoftver
  + Informatikáért felelős egység szerzi be, telepíti
  + Az egység rendszergazdája végzi, vezetői utasításra
  + Szoftverek jogvédelmét figyelembe kell venni
  + Munkamásolatokat kell készíteni (elkülönített helyen kell tárolni)
  + Jogtiszta szoftver
  + A vállalat tulajdonát tilos eltulajdonítani
  + Szabad szoftver használata esetén kártékony programokat kerülni kell
* Adathordozó:
  + Nem használható személyes célokra a vállalati adathordozó
  + Sajátot sem ajánlatos munkacélra használni
  + Fizikai védelemre ügyelni kell
  + Minősített adatot csak nyilvántartott adathordozóra szabad felvinni
    - melyeknek külön azonosítója van
    - tilos felügyelet nélkül nyílt helyen tárolni
    - másolat készítése engedéllyel
* Adathordozó:
  + Sérült, hibás – tilos használni
  + Ha csak olvassuk, akkor legyen írásvédett
  + Használaton kívül elzárni, esetleg raktárban tárolni, archiválni
  + Selejtezéskor megsemmisíteni
    - Minősített adatot tartalmazókat zúzással.
* Dokumentumok, adatok:
  + Védelmére a hardver és szoftver védelmi eljárások vonatkoznak
  + Papíron -> levéltárban (biztonsági rendszerrel őrzött szobában)
  + A visszakövethetőség miatt: folyamatosan dokumentálni kell, naplóállományokat megfelelően kezelni kell
  + Minden olyan eseményt, amely eltér a megszokott üzemviteltől naplózni kell (tartalmaznia kell az esemény pontos leírását)
* Adatok:
  + hozzáférés védelem,
  + rendszeres mentés,
    - tükrözés,
    - biztonsági mentés,
  + adatállományok védelme
  + Az input adatok helyességének biztosítása
  + A feldolgozás helyességének védelme
  + Archiválás
* Hálózati védelem:
  + A szervezet belső hálózatának és központi levelező szerverének üzemeltetése, valamint a világhálóra való csatlakozás a központi informatikai egység feladata
    - Ők szűrik a kártékony programokat, leveleket
  + Hálózati rendszeradminisztrátor: az informatikai egység vezetője által megbízott személy
    - A hálózat mindenkori kiépítettségét
    - A hálózatra kötött minden eszköz naprakész nyilvántartását vezeti
    - Csak a hozzájárulásával lehet a hálózatra eszközt kötni, továbbá több egységet érintő szerverszolgáltatást indítani
    - Jogkörét részben átadhatja valamely egység rendszeradminisztrátorának (az átadott jogok és kötelezettségek pontos leírásával)
* A belső elektronikus levelezés:
  + Leghasznosabb és leggyakrabban használt szolgáltatás
  + Növeli a vállalat belső és külső információ forgalmának sebességét és hatékonyságát
  + Dolgozók együttműködését is egyszerűbbé teszi
  + Negatív hatások is vannak (pl. kéretlen levelek)
  + Tilos olyan adatok továbbítása (küldése, letöltése), amely alkalmas kártékony kódnak a vállalat informatikai rendszerébe juttatására
* A belső elektronikus levelezés:
  + A dolgozók milyen célra használhatják
  + Az elektronikus levelezési cím és a hozzátartozó elektronikus láda a vállalat tulajdona
    - Iktatni kell és ellenőrizhetik
    - Ha a dolgozó kiesik, beteg, más is hozzáférhessen
    - Projekttel kapcsolatos leveleket külön mappában tárolni, lezárás után archiválni
    - Az informatikáért felelős egységnek rendszeres mentést kell végeznie a dolgozók e-postaládájáról.