Kódolás és IT Biztonság

diák ellenőrző kérdései és annak válaszai

(hibák, illetve hiányok előfordulhatnak!)

1)

Mit értünk rendszer-kompromittálódás alatt?

- A rendszer szolgáltatásaihoz vagy erőforrásaihoz történő illetéktelen hozzáférés, azok illegitim módon történő használata, módosítása, vagy elérhetetlenné tétele

Definiálja az alábbi fontosabb fogalmakat: CIA, AAA, Security engineering, security operations

- C = Confidentiality (bizalmasság) az információhoz történő jogosulatlan hozzáférés megakadályozása
  I = Integrity (integritás) az információ jogosulatlan módosításának megakadályozása
  A = Availability (rendelkezésre állás) az információ elérhetőségének biztosítása az arra jogosultak számára

- Authentication (hitelesítés)
  Authorization & accesscontrol (engedélyezés & hozzáférés-vezérlés)
  Accounting (felelősségre vonhatóság)

Hogyan definiáljuk a biztonsági kockázatot?

- biztonság = kockázat menedzsment: a kockázat a sikeres támadásból eredő veszteség várható értéke

Mik a kockázatot befolyásoló tényezők?

- Risk = Likelihood * Impact
- Impact: a sikeres támadásból bekövetkező potenciális veszteség
- Likelihood: a sikeres támadás valószínűsége, mely függ a támadótól, a rendszer sérülékenységeitől, az alkalmazott ellenintézkedésektől

Mi az a maradvány-kockázat? Miért nem csökkenthető nullára?

- a kockázat általában nem eliminálható teljesen, mert minden elképzelhető támadás megakadályozása túl drága (még ha egyáltalán lehetséges is volna)

Milyen tényezők alapján jellemezhetők a támadók?

- Motiváció
- Szándék
- Lehetőség
- Képességek: információszerző képesség; műszaki tudás, szakértelem; megtévesztési képességek
- Erőforrások

Milyen információkat érdemes megszerezni támadás előtt?

- a rendszer általános felépítése, elérhető szolgáltatásai, hardver és szoftver komponensei és azok konfigurációja, a hálózati topológia, az alkalmazott protokollok, ..
- az alkalmazott biztonsági mechanizmusok (pl. tűzfalak, behatolás detektáló rendszerek, anti-vírus szoftverek, ...)
- a rendszer és az alkalmazott biztonsági megoldások ismert sérülékenységei
- kik a rendszer felhasználói, milyen jogosultságokkal rendelkeznek?

A támadó műszaki képességeinek, szakértelmének milyen szintjei lehetnek?

- alapvető műszaki és informatikai ismeretek
- ismert sérülékenységek és támadási módszerek, eszközök ismerete; IT biztonsági szakértelem
- új sérülékenységek felfedezésének, új támadási módszerek és eszközök kifejlesztésének képessége

Miért fontos az anyagi erőforrás a támadó számára?

- szabadon konvertálható információvá, tudássá, szakértelemmé, emberi erőforrássá
- Információszerző képesség növelése
- Műszaki tudás, szakértelem növelése
- Speciális támadói képességek növelése

Milyen tipikus támadó profilok vannak? Minden profil esetén foglalja össze az adott támadó típus motivációját, képességeit, és erőforrásait!

- script kiddie
- hacktivist group
- cybercriminal group
- government sponsored attackers

Milyen szolgáltatások érhetők el a kiberbűnözőkszámára az underground piacon?

- információk és eszközök (pl. exploit-ok, malware) vásárlása
- térben és időben kiterjedt támadó kampányok futtatása
- „szakértők, specialisták” alkalmazása
- műszaki megoldások (spyware telepítése; szerverek, felhasználói fiókok feltörése)

Mik a célzott támadás jellemzői?

- Testreszabott támadó eszközök és behatolási technikák
- Időben kiterjedt (persistent), mégis rejtve maradó (stealthy) működés
- Erőforrásokban gazdag háttér

Mi az a Stuxnet, és miért fontos?

- kifinomult malware, több 0-day exploit, hamis digitális aláírás

Mi az a „killchain” modell?

- A támadás felépítésének, az alkalmazott technikáknak az ismerete elengedhetetlen a támadás megakadályozását célzó biztonsági mechanizmusok tervezéséhez

Milyen típusú sérülékenységek lehetségesek IT rendszerekben?

- tervezési és implementációs hibák a:
  - rendszert alkotó hardverben és szoftverben
  - komponensek közötti interfészekben
  - komponensek között használt protokollokban

- fizikai gyengeségek, melyek kihasználása fizikai hozzáférést eredményezhet a rendszerhez (pl. folyosón elhelyezett router)

- Üzemeltetési gyengeségek (pl. alapértelmezett jelszó megváltoztatása nincs kikényszerítve)

- Személyi: alkalmazottat vagy szerződéses partnereket érintő gyengeségek (pl. biztonságtudatosság hiánya, szakértelem hiánya, lefizethetőség, megzsarolhatóság)

Mik a műszaki jellegű sérülékenységek okai?

- A rendszerek nagy komplexitása
- tervezési, implementációs vagy ellenőrzési tesztek/módszerek hiánya
- Korlátozott erőforrások
- Rossz feltételezések használata a tervezés vagy az üzemeltetés során
- Gyenge minőségű specifikáció az implementáció számára

Hogyan kezeljük a publikusan ismert sérülékenységeket?

- Sérülékenységek rutinszerűen kerülnek így nyilvánosságra, általában egy kontrollált folyamaton (responsible disclosure procedure) keresztül
- A publikus sérülékenységek egyedi azonosítót kapnak, CVE ID-t
- A publikus sérülékenységekkel kapcsolatos információkat publikus sérülékenység-adatbázisokban tárolják-strukturált, kereshető
- A nyilvánosságra hozatal lehetővé teszi az ismert sérülékenységekből származó problémák eliminálását, mely önmagában drasztikusan csökkenti a biztonsági kockázatot

Mik azok a 0-day sérülékenységek? Miért veszélyesek?

- Olyan sérülékenységek, melyek csak a támadó számára ismertek nagy problémát jelentenek, mert nem lehet rájuk hatékonyan felkészülni, nem lehet ellenük védelmet kialakítani
- ritkák és drágák, ezért főként célzott támadásokban használják őket, ahol a támadás sikere nagyon fontos es a támadás észlelésének

Milyen kockázat csökkentő ellenintézkedések lehetségesek IT rendszerekben?

- Műszaki (logikai) -a hosztokonés a hálózatokban használt biztonsági mechanizmusok, megoldások; pl. tűzfalak, anti-vírus szoftverek, jelszavas hitelesítés, kritpográfiaialgoritmusok és protokollok
- Fizikai -fizikai védelmet biztosító mechanizmusok; pl. zárak, kerítések, biztonságiőrök, bontásellenálló (tamperresistant) hardver elemek
- Üzemeltetési -a rendszer üzemeltetésével és a személyzet menedzsmentjével kapcsolatos szabályzatok és eljárásrendek; pl. jelszócserére és -erősségre vonatkozó szabályozás, rendszeres biztonsági tesztelés, felvétel és elbocsátás során alkalmazott eljárások, kötelező szabadság
- Személyi -a személyzet biztonságtudatosságának, megbízhatóságának, lojalitásának növelését célzó mechanizmusok; pl. tréning, gyakorlat, jó fizetés, vonzó munkakörülmények

Soroljon fel néhány megelőző jellegű és néhány észlelést célzó biztonsági mechanizmust!

- Megelőzést célzó mechanizmusok:
  - Rejtjelezés, Hitelesítés, Hozzáférés-vezérlés, Address Space Layout Randomization (ASLR), Control Flow Integrity (CFI), Víruskeresés (scanning), Tűzfal, Network Intrusion Prevention Systems (IPS), Bontásellenállóborítás, Biztonságiképzés
- Észlelést célzó mechanizmusok:
  - Üzenet integritásvédő ellenőrzőösszeg, Rootkit detekció, Network Intrusion Detection Systems (IDS,) log analízis, Security Information and Event Management (SIEM), “tamper evident” borítás

Mi az a biztonsági incidens?

- Biztonsági incidens alatt egy sikerestámadásból származó, észlelt rendszer-kompromittálódást értünk

Miért fontos az incidensek hatásos és hatékony kezelése? Mik az incidenskezelés céljai?

- A biztonsági incidenseknek komoly negatív hatása lehet az üzemeltetőre (cég, szervezet, egyén) nézve
- Üzletmenet-folytonosság megszakadásának minimalizálása, Hasonló incidensek jövőbeli bekövetkezésének megelőzése, Igazságügyi eljárás, nyomozás támogatása

Mik az incidenskezelés főbb lépései?

- Incidens előtti előkészületek, észlelés, Incidens kezelési stratégia alkotása, Adatgyűjtés-elemzés, a rendszer helyreállítása, Incidens riport készítése, Sérülékenységek javítása, továbbképzés

2)

Mi a CVE?

- nyilvános adatbázis a megtalált sérülékenységekről

Miért nehéz biztonságos szoftvert fejleszteni?

- Fejlesztői megkötések: idő, erőforrások, funkcionalitásTámadói megkötések: motiváció és felkészültség
- Funkcionális tesztelés: hogyan kéne működnie a rendszernekBiztonsági tesztelés: hogyan NEM kéne működnie a rendszernek

Milyen fázisai vannak a Microsoft SDL-nek?

- before SDL, SDL, after SDL

Milyen feladatokat és szerepeket kell a személyzetnek ellátnia?

- development team: develop
- development team security contact: security contact informations
- security advisor: security
- security team: product security
- security leadership team: security management

Mit nevezünk bugkövető rendszernektracking system? Hogyan kell létrehozni ahhoz, hogy sikeres legyen a bugkövetés?

- Bugokat tartalmazó adatbázis
- Biztonsággal/privacy-velkapcsolatos infókat is csatoljunk!
- Szükséges mezők: Ok, Hatás
- redmine, github, jira, bugzilla

Mit mond a "economy of mechanism" alapelv?

- security mechanisms should be as simple as possible

Mit mond a "fail-safe defaults" alapelv?

- whitelist/blacklist of privileges

Mit mond a "complete mediation" alapelv?

- Ellenőrizzünk minden elérést minden objektumhoz

Mit mond a "separation of privilege" alapelv?

- Többféle feltételnek is tegyenek eleget, mielőtt megadjuk az engedélyt

Mit mond a "least privilege" alapelv?

- Csak annyi jogosultságot használjunk, ami feltétlenül szükséges a feladat elvégzéséhez

Mit mond a "open design" alapelv?

- A terveket elérhetővé kell tenni, hogy minél szélesebb körben elemezhessék

Mit mond a "least common mechanism" alapelv?

- Minimalizáljuk azon mechanizmusok számát, amik Több felhasználó számára is elérhetőek,és Minden felhasználó számára szükségesek

Mit mond a "psychological acceptability" alapelv?

- Ha a felhasználók nem fogadják el, meg fogják próbálni kikerülni
- A rendszer használata legyen egyszerű és intuitív

Mit értünk egy szoftver támadási felülete alatt?

- Minden végrehajtási út, min keresztül adat/parancs érkezik/távozik
- Kód, ami ezeket a végrehajtási utakat véd
- Minden értékes adat, amit a szoftver használ
- Kód, ami ezeket védi

A szoftvere hibás bemenetet észlel. Hogyan reagáljon?

- Normalizáció/ kanonikalizáció: átalakítás a legegyszerűbb (és elvárt) formára
- bizonyos kritériumok alapján egyes elemek eltávolítása a bementből
- bemenet ésszerűségének ellenőrzése
- dispose and log

Milyen információkra van szükség naplózáskor?

- mikor, hol, ki, mit

Miért fontosak a naplóbejegyzések?

- Biztonsági incidensek azonosítása
- Szabályszegések monitorozása
- Letagadhatatlansági mechanizmusok támogatása
- Incidensek kivizsgálása

Milyen típusú adatot nem szabad sose naplózni?

- Key
- password
- source code
- token
- sensitive informations

Milyen előnyei és hátrányai vannak a statikus elemzésnek?

- Az utasításokat csak értelmezzük, nem hajtjuk végre
- nagy kódbázist is képes kezelni
- fejlesztés során korán el lehet kezdeni alkalmazni
- Nincs futásidejű információ fals pozitív eredmény: olyan utasításokat is sérülékenynek jelezhet, amit a valóságban sose lehet lefuttatni

Milyen előnyei és hátrányai vannak a dinamikus elemzésnek?

- A szoftver futtatás közben elemezzük
- Elérhetők futási idejű információk pontosabb eredmények
- Jelzett sérülékenységek biztosan benne vannak a szoftverben
- Instrumentációralehet szükség fordított kódra nem triviális
- Alacsony kód-lefedettség: Nem tud semmit mondani olyan viselkedésről, amit nem figyelt meg

Mi a fuzzingalapötlete? Mi az egyes komponensek feladata?

- Automatikus generált véletlenszerű bemenetek: Szoftvert futás közben monitorozzuk hibákat keresve

Mi a célja a penetrációs tesztelésnek? Hogyan végzünk pentesztelést?

- Futó rendszeren (teszt vagy production) végzett elemzés Cél: demonstrálni, hogy mit tehet egy támadó

Mi a különbség a security response és az incidens kezelésközött?

- security response: Mit tegyünk, ha érintettek vagyunk?
- incident response team: Mi legyen, ha a sérülékenységet nem felelősen hozták nyilvánosságra?

Mit jelent a sérülékenységek felelős nyilvánosságra hozatala?

- incidens kezelési plan

Hogyan néz ki egy sérülékenység életciklusa?

- tamado elemzi > uzemelteto fejleszto javitasba kezd > patch> kapcsoolodo kodreszlet elemzese

3)

Melyik programozási nyelvek vannak leginkább kitéve a memória korrupciós hibáknak?

- c/c++

Hol találhatóak a memóriában a függvény paraméterek és lokális változók?

- stack

Hogyan működik egy stack overflow támadás?

- modositja a stack elemet

Mi lehet a célja egy támadónak a visszatérési cím módosításán kívül?

- rendszer feletti iranyitas

A stack overflown kívül milyen más memória korrupciós hibákat ismer?

- buffer, integer overflow, null pointer, dangling pointer

Milyen védekezési módszereket ismer memória korrupciós támadásokkal szemben? Hogyan működnek ezek?

- DEP, canaries, aslr

4)

Hogyan épül fel egy URL?

- scheme://authority/path/?query=#fragment
- PL.: https://www.crysys.hu/education/?q=bprof#top

Mik a leggyakoribb web biztonsági problémák?

- Biztonságos kommunikáció a böngésző és a szerver között
- Kliens oldali támadások
- Szerver oldali támadások

Mi a célja az OWASP projektnek?

- Céljuk, hogy felhívják a figyelmet a webes biztonsági kérdésekre, hogy a cégek és a fejlesztők jó döntéseket tudjanak hozni

Általában mi az első lépése egy támadónak egy szerver ellen?

- A támadó a szerverhez csak hálózati kommunikáción keresztül fér hozzá

Mi az általános probléma injection típusú támadásoknál?

- nagyon gyakori probléma: egy SQL lekérdezése előállítása stringműveletekkel külsőadatok alapján

Mutasson egy példát egy SQL injection támadásra!

- username=a’ union select 1,2,3,* from users #

Hogyan működik egy blind SQL injection?

- ha a lekérdezésnek semmilyen olvasható eredménye nem látszik (hibaüzenet sem)

Hogyan működik egy időzítés alapú SQL injection?

- sleep utasítás

Hogyan kell védekezni egy SQL injection ellen?

- Kerülni kell a dinamikus SQL lekérdezéseket
- A bemeneteket szűrni és ellenőrizni kell, Parameterized statement-ek használata
- megfelelő jogosultság az adatbázison

A kliens oldalon hogy definiáljuk a biztonsági határt?

- Origin = scheme + domain + port

Mi az a SameOrigin Policy?

- Egy weboldal csak a saját originjéheztartozó erőforrásokhoz férhet hozzá korlátozás nélkül. Más originhez tartozó erőforrásokra a korlátozások vannak.

Mit limitál a SameOrigin Policy egy XMLHttpRequest elküldésekor?

- HTTP kérések küldése és a válasz olvasása csak a saját domainre lehetséges

Hol jelenhet meg JavaScript kód?

- HTML-fileban
- .JS fileban
- CSS -ben is, de trükkösen

Mit tud megtenni egy kártékony JavaScript kód?

- Minden script, ami egy adott originből származik, hozzáférhet minden más script változóihoz, függvényeihez, objektumaihoz

Mi az a XSS támadás?

- a támadó képes JavaScript kódot futtatni egy másik originen belül
- A kártékony kód lényegében bármit megtehet a támadás során

Milyen típusai vannak az XSS támadásnak?

- Persistent/StoredXSS → szerver oldalon
- ReflectedXSS → JS egy GET/POST paraméteren át fertőz
- DOM alapú XSS → támadás tisztán a kliens oldalon történik

Hogyan lehet védekezni XSS támadással szemben?

- kívülről származó adatot ellenőrizni kell mielőtt az a HTML-be kerül!
- A kontextus fontos, különböző ellenőrzéseket kell végezni! Tiltó lista alapú megközelítés vagy egyszerű törlés nem elég
- HTTP-only sütik

Mi az a Content Security Policy?

- HTTP fejléc megadja, hogy honnan tölthet be tartalmat a böngésző egy megadott URL-re jelzi, ha szabálysértés történik

5)

Micsoda és mire való az authentikáció (hitelesítés)?

- Ki vagy te?

Micsoda és mire való az authorizáció (engedélyezés)?

- Kinek mihez van engedélye

Micsoda és mire való az accesscontrol(hozzáférés-szabályzás)?

- Van engedélyed elvégezni a(z) X műveletet a(z) Y objektumon?

Micsoda és mire való az accounting (naplózás)?

- Célja, hogy naplózzuk a felhasználók által végzett műveleteket
- Érdemes nem csak a sikertelen, hanem a sikeres műveleteket is naplózni

Mit jelent az AAA rövidítés?

- authentication, authorization, accounting

Milyen formái vannak az authentikációnak? Mindegyikre adj példát!

- Tudásalapú (knowledge-based) –valami, amit tudsz–Jelszavak–PIN-kódok
- Birtokalapú (possession-based) –valami, ami nálad van–Mobil eszközök (SMS-ben küldött kódok, tokengenerátoralkalmazások)–Offline tokengenerátorok
- Tulajdonságalapú (inherence-based) –valami, ami vagy–Ujjlenyomat–Arc, fül geometriája–Retina és írisz

Mi az, hogy 2FA, MFA? Mi a célja, alapötlete?

- Sikeres hitelesítéshez kettő (vagy több) hitelesítési módszer egyidejű használatát követeljük meg

Hogyan lehetséges jelszóalapú rendszereket megtámadni?

- Hálózati forgalom lehallgatásával, eyloggerek, malware-eksegítségével, social engineering, shoulder surfing
- Brute force ("nyers erő") támadások, Szótáras támadások, Hibrid támadások

Hogyan érdemes fejlesztőként jelszavakat kezelni, tárolni?

- Hashing: A plain text jelszó helyett egy egyirányú függvényen átfuttatott (hashelt) változatot tárolunk az adatbázisban
- Salting: Hashelésnél nem csak a jelszót használjuk fel, hanem egy, felhasználónként különböző random értéket is
- Stretching: mesterségesen megnöveljük a hashkiszámításának idejét
- Kulcsderiváló függvényekkel (Key Derivation Function, KDF): egyszerre használnünk erős hashfüggényeket, sózást és nyújtást
- Használjunk létező függvénykönyvtárakat, ne írjunk saját megoldást

Hogyan érdemes felhasználóként jelszavakat választani?

- Lehet jelmondat
- Hosszú, komplex nem szótári szó mutációja
- Rendszeresen változtatni

Ismertesd a birtokalapú hitelesítési módszereket! Ezeknek mik lehetnek a hátrányai?

- Egyszer használatos jelszavak segítségével
- smartkártyákkal
- Lemerül az elem - Kiesik a szinkronból
- Tönkre megy a telefon / Ellopják a telefont

Ismertesd az engedélyezés alapfogalmait és modelljét!

- Művelet (Operation)
- Alany (Subject)
- Objektum (Object)
- Biztonsági házirend (Policy)

Tipikusan kik és hogyan definiálhatnak engedélyeket?

- A rendszer üzemeltetői által
- Az objektumok tulajdonosai által
- Engedélyezési (authorization) megoldások segítségével

Mi az, hogy Discretionary Access Control?

- Az objektumoknak tulajdonosaik vannak
- A tulajdonosok dönthetnek úgy, hogy más alanyoknak engedélyt adnak a saját objektumaikhoz
- Tipikusan hozzáférés-vezérlési listák (ACL) segítségével valósul meg

Mi az, hogy Mandatory Access Control?

- az alanyok és az objektumok címkézettek
- a címkék között összerendeléseket hozunk létre
- SELinux - AppArmor

Hogyan néz ki Linux és Windows esetében ... ?

A felhasználók kezelése

- W: SID (Security Identifier) Security Accounts Manager (SAM) adatbázis tárolja
- L: minden felhasználónak van egy egyedi azonosítója (UID) és egy egyedi felhasználóneve

Az engedélyeik kezelése

- W: Összesen 14 különböző engedély adható, ezek az engedélyek megjelölhetők, örökölhetők (vagy megszakíthatók)
- L: /etc/shadow fájl alapján

A hozzáférés-szabályzás

- W: Egy objektum tulajdonosa mindig módosíthat az objektum jogosultsági listáján, még akkor is, ha ez valamiért explicit tiltva lenne neki
- L: A hozzáférést engedélyek szabályozzák, a tulajdonosnak (u), a tulajdonos csoportnak (g), és mindenki másnak (o)

Mit jelent, ha egy fájl jogosultságáról azt látom, hogy 755?

- read and execute access for everyone and also write access for the owner of the file

Mi a setuid, setgidés a stickybit szerepe a hozzáférés-szabályzásban?

- setuid: ha egy futtatható állományra be van állítva, az mindig a tulajdonos nevében fut, függetlenül attól, hogy ki indította el
- setgid: setuidhoz hasonló, de a tulajdonos csoportot állítja át
- sticky bit: ha be van állítva egy könyvtárra, az itt lévő fájlokat csak a tulajdonosuk törölheti

Mi az engedélyek precedenciasorrendje?

- Explicit Deny> Explicit Allow> Implicit Deny> Implicit Allow

6)

Miért van szükség tűzfalakra?

- Szabályok alapján hálózati hozzáférést szabályozó rendszer vagy rendszerek csoportja
- Minden forgalom kintről befelé vagy fordítva menjen keresztül a tűzfalon
- Csak az engedélyezett forgalmak mehetnek át
- a tűzfal maga is védett legyen a támadások ellen

Mi a különbség egy csomagszűrő és egy állapot alapú tűzfal között?

- Csomagszűrő tűzfal: Tipikusan útvonalválasztóval együtt kerül kialakításra, a csomagokat a 3-as illetve 4-es rétegbelifejlécek alapján szűri
- Állapot alapú tűzfal: Kapcsolatok állapotát (kezdeményezés, élő kapcsolat, lezárás) is figyelembe veszi a döntéseknél

Hogy működik egy proxitűzfal?

- magasabb rétegbeli információkat is figyelembe tud venni a döntéseknél (bonyolult szoftveres megoldás)

Mire valók a láncok és táblák iptables esetén?

- Tábla = láncok halmaza (PREROUTING, INPUT, FORWARD, OUTPUT és POSTROUTING.)
- Lánc = szabályok rendezett listája

Mire jó egy IDS?

- Behatolás detektálás: biztonsági szolgáltatás amely figyeli a rendszer eseményeit, és megpróbálja a behatolásokat (közel) valós időben érzékelni, és riasztásokat generálni

Hogy tud detektálni egy IDS?

- Mintázat alapú detekció
- Anomália alapú detekció
- Állapot alapú protokoll ellenőrzés

Mi a különbség egy IDS és egy IPS között?

- Intrusion Detection System = Betaloást detektáló rendszer
- Intrusion Prevention System = Behatolást Megelőző Rendszer
  - kb. ids + blokkolási képesség
  - Forgalom átmegy rajta

Mire való egy VPN?

- VPN = Virtual Private Network: egy hálózatba hoz össze távoli rendszereket

Milyen problémát old meg egy SIEM?

(secure information and event managger system)

- Big data problémák és megoldások
- Események közötti összefüggéseket figyelni kell
- fals riasztások nagy részét el lehet kerülni

7)

Milyen feladatokra jó a kriptográfia?

- rejtjelezés, üzenet hitelesítő kód (MAC) , digitális aláírás

Mi a rejtjelezés alapmodellje

- kuldo es fogado birtokol egy kulcsot, az uzenet kodolva van, amit csak ezzel tud elolvasni

Mi a különbség a kulcsfolyam rejtjelezők és a blokkrejtjelezők között?

- A kulcsfolyam rejtjelezők egyáltalán nem biztosítanak integritás-védelmet, szimuláljuk a valódi véletlen kulcsfolyamot egy álvéletlen byte sorozattal, amit egy algoritmussal generálunk egy véletlen bemenetből
- A blokkrejtjelező több byte-bólálló blokkokat kódol/dekódol, Ha valaki nem ismeri a kulcsot, akkor számára a blokkrejtjelező olyan, mint egy véletlen függvény

Hogyan működik a one-timepad?

- kodolas: Az üzenetet egy byte sorozatként reprezentáljuk: m1, m2... mL,
  kulcsként egy valódi véletlen byte sorozatot használunk: k1k2... kL, a kettőt egymáshoz XOR-oljuk, ez adja a rejtett üzenetet
- dekodolas: A kulcsként használt byte sorozatot a rejtett üzenethez XOR-oljuk

Mi a Kerckhoff-elv?

- A támadóról mindig azt érdemes feltételezni, hogy ismeri a kódoló (és dekódoló) algoritmust, és csak a kulcs ismeretlen számára

Milyen támadó modellek léteznek rejtjelezés esetén?

- ciphertext-only attack
- known plaintext attack
- chosen plaintext/ciphertext attacks → orákulum támadások

Mi a kimerítő kulcskeresés támadás átlagos komplexitása?

- Átlagos komplexitása 2^k-1, ha a kulcs k bites

Mi a publikus kulcsú rejtjelezés alapötlete?

- Ezeknek a rejtjelezőknek a biztonsága általában valamilyen nehéznek vélt matematikai problémára vezethető vissza (pl: faktorizáció, diszkrét logaritmus számítás)

Mi a hibrid rejtjelezés motivációja, és hogyan működik?

- Az aszimmetrikus kulcsú rejtjelezők általában sokkal lassabbak mint a szimmetrikus kulcsú rejtjelezők és nagyobb a kulcsméretük (pl. 2048-4096 bit)
- A sebesség probléma hibrid rejtjelezéssel oldható meg

Mik azok a kriptográfiai könyvtárak? Miért hasznosak?

- Kriptográfiai algoritmusok implementációját tartalmazzák, és ezek használatát API-kon keresztül teszik lehetővé
- Általában egyéb hasznos, kriptográfiai alkalmazásokban szükséges funkciókat is tartalmaznak

8)

Mi a kriptográfiai hashfüggvény?

- Egy hashfüggvény tetszőlegesen hosszú bemenetből egy adott fix hosszúságú (n bites) kimenetet állít elő

Mi a hashfüggvények 3 fő biztonsági tulajdonsága?

- Könnyű (általában gyors) számíthatóság
- Gyenge ütközés ellenállóság, Erős ütközés ellenállóság
- Őskép ellenállóság

Hogyan használható egy MAC függvény üzenet-hitelesítésre?

- Tetszőlegesen hosszú bemenetből és egy fix (k bit) hosszúságú kulcsból, egy fix (n bit) hosszúságú kimenetet állítanak elő
- MAC sikeres ellenőrzése után, a fogadó tudja, hogy az üzenetet olyan valaki küldte, aki ismeri a kulcsot (azaz a küldő), és az nem változott meg az átvitel során

Milyen támadói modellek léteznek MAC függvények esetén?

- MAC hamisítás
- MAC függvény feltörése
- orákulum támadás

Mik a MAC függvények biztonsággal kapcsolatos tulajdonságai?

- Key non-recovery
- computation resistance

Hogyan működnek a nyers erő támadások a MAC függvények ellen?

- Egy adott üzenethez véletlen választott MAC érték 2-n valószínűséggel helyes--» a kimerítő próbálkozás átlagos komplexitása 2n-1
- Minden lehetséges kulccsal kiszámoljuk az miMAC értékét és összehasonlítjuk Mi-vel–Ha nincs egyezés, akkor új kulccsal próbálkozunk
- Egyezés esetén a kulcsot teszteljük az összes rendelkezésre álló (mi, Mi) páron

Mi a digitális aláírás? Mik a fő különbségek a digitális aláírás és a MAC függvények között?

- Hasonló a MAC függvényhez, de a digitális aláírás a fogadó által hamisíthatatlan, bármely harmadik fél által ellenőrizhető
- Üzenet-hitelesítés és integritás-védelem
- Eredet letagadhatatlanság

Milyen támadói modellek léteznek digitális aláírás sémák ellen?

- key-only attack, known-message attack , chosen-message attack

Mi a hash-and-sign modell? Miért használjuk a gyakorlatban? Milyen követelményt támaszt az alkalmazott hashfüggvénnyel szemben?

- Nagy inputon végzett publikus/privát kulcsú műveletek lassúak
- A hatékonyság növelhető, ha nem magát az üzenet írjuk alá, hanem annak a hashértékét
- Ekkor nagyon fontos, hogy a hashfüggvény ütközés ellenálló legyen

Milyen problémák fordulnak elő a gyakorlatban kriptográfiát használó rendszerekben?

- Kulcs menedzsment problémák
- Protokoll gyengeségek
- implementációs problémák,
- emberi butaság

Mit jelent az, hogya valódi véletlen számok megjósolhatatlanok?

- Egy valódi véletlenszám generátor kimenete megjósolhatatlan, még akkor is, ha korábbi kimeneteit ismerjük

Miben különbözik egy álvéletlen generátor egy valódi véletlen szám generátortól?

- Az álvéletlen generátorok (pseudo-random numbergenerator-PRNG) nem valódi véletlen kimenetet állítanak elő

Hogyan épül fel egy PRNG? Honnan kaphat valamennyire meg jósolhatatlan bemenetet?

- aktuális idő (óra)–billentyű leütések közti idő, egér mozgás/diskhozzáférési idő, kamerakép, mikrofon–hardver mellékhatások

Hogyan tárolhatunk kulcsokat biztonságosan?

- Bontás ellenálló hardver modul (pl: chip-kártya)
- Rejtjelezett fájl

Mire kell oda figyelni, ha jelszóból generálunk kriptográfiai kulcsot?

- Figyelni kell arra, hogy a jelszó potenciálisan lehet gyenge, ami off-line szótáras támadásokhoz vezethet
- Speciális módon kell a fájlrejtjelező kulcsot generálni a jelszóból!

Mi aza salting és mi az a stretching?

- Stretching: a jelszó találgatás költségének növelése azzal, hogy a kulcsot a jelszóból egy sok iterációt igénylő számítással állítjuk elő
- Salting: salt = véletlen érték, amivel a jelszót módosítjuk a kulcsgenerálás előtt–Biztosítja, hogy ugyanabból a jelszóból több különböző kulcs is generálható

9)

Mi nevezünk biztonságos csatornának?

- Egy biztonságos kommunikációs kapcsolatot értünk

Mi az a hitelesített rejtjelezés? Mik a hitelesített rejtjelezés megvalósításainak módszerei?

- Egyszerre biztosítanak titkosítást valamint üzenet-hitelesítést és integritásvédelmet
- Egy rejtjelező és egy MAC függvény valamilyen kombinációja
  dedikált hitelesített rejtjelezés módok (pl. CCM, GCM, OCB)

Mutassa be a rejtjelezés és a MAC kombinálásának generikus módszereit! Melyik módszer a legbiztonságosabb?

- ENCKe(m| MACKm(m))
- ENCKe(m)|MACKm(m)
- ENCKe(m)|MACKm(ENCKe(m))

Hogyan használjuk az üzenet sorszámozást visszajátszások detektálására?

- A visszajátszás elleni védelmet az üzenetek sorszámozásával lehet megoldani, ahol megköveteljük, hogy minden vett üzenet sorszáma nagyobb legyen, mint a korábban vett üzenetek sorszáma

Hogyan lehet megoldani a visszajátszás elleni védelmet nem megbízható (unreliable) hálózatokban?

- A vevő ilyenkor egyetlen „vételi” sorszám helyett egy „vételi ablakban” tartja nyilván az eddig vett sorszámokat

Milyen biztonsági szolgáltatásokat nyújt a TLS protokoll?

- A TLS egy olyan protokoll, ami egy biztonságos csatornát implementál két távoli fél között(tipikusan egy webszerver és egy böngésző között https)

Mi az a CA? Mik a feladatai? Miért bízunk meg a CA-kban?

- Certificate-Authority
- A publikus kulcs és a kulcs gazdája nevének (azonosítójának) elválaszthatatlan összekötése egy bizalmi szolgáltató digitális aláírása által

Mik a tanúsítvány életciklusának fázisai?

- A felhasználó regisztrál a CA-nál és igényel egy tanúsítványt
- A CA ellenőrzi a felhasználó személyazonosságát
- Kulcspár generálás
- Tanúsítvány kiadása
- A tanúsítvány esetleges visszavonása
- A tanúsítvány lejár

Hogyan tároljuk és védjük a privát kulcsokat?

- A CA a privát kulcsát fizikailag is védett, bontás-ellenálló hardver modulban tárolja.

Hogyan épül fel a hierarchikus PKI?

- ahol egy gyerek CA kulcsát a hierarchiában a szülő CA hitelesítt
- ez egy fával modellezhető; a gyakorlatban, több fa van külön gyökerekkel

Mi az a tanúsítvány lánc? Hogyan történik az ellenőrzése?

- a root CA önaláírt tanúsítványával kezdődik, a felhasználó tanúsítványával végződik
- további CA-k tanúsítványait is tartalmazza, melyek a fában a root CA-tól a felhasználóig futó úton találhatók

Miért és hogyan történik a tanúsítványok visszavonása?

- egy tanúsítványt esetleg a lejárata előtt vissza kell vonni, ha a tanúsítványban található publikus kulcshoz tartozó privát kulcs kompromittálódott
- ha a tanúsítványban található adatok (pl. név, e-mail cím) megváltozott
- A visszavont tanúsítványok visszavonási listára (CertificateRevocationList) kerülnek, amit a CA kezel, és rendszeres időközönként nyilvánosságra hoz

10)

Nevezd meg a hibakorlátozó kódolás főbb alkalmazási területeit!

- internet, adattarolok, hibajavito memoria, satellite broadcasting (tv), deep space communication

Mi a hibadetekciós feladat?

- célja átviteli hiba detektálása

Hogy mérjük a hibajavító kód hatékonyságát?

- C(n, k, dmin)

Mi a CRC? Nevezzen meg néhány alkalmazását!

- ciklikus redundancia ellenőrző kód. A bináris szavakat bináris polinomként ábrázoljuk

11)

Mi a forráskódolás célja?

- Cél az információ hatékony továbbítása vagy tárolása:
  több információ átvitele adott csatornán, több információ tárolása adott tárhelyen, sávszélesség, tárkapacitás igény csökkentés

Miért hívunk veszteségmentesnek egy kódolást?

- teljesen visszaallithato az eredeti allapot

Nevezze meg a Huffmann-kódolás néhány alkalmazását!

- GZIP, PKZIP, BZIP2, JPEG rész-algoritmusa

Nevezze meg az LZW-kódolás néhány alkalmazását!

- unix, gif, pdf

Hogyan generáljuk a Huffmankódolás frekvencia tábláját?

- binaris fa a karakterek gyakorisaga alapjan

Mi a Huffman-kódolás tömörítés ötlete?

- rövidebb kódszavak tendenciájukban a nagyobb gyakoriságú karakterekhez tartoznak

Mi az alaplépés LZW könytár bővítésekor?

- szoveg/index

Mi az LZW-kódolás tömörítés ötlete?

- Ahogy egyre beljebb haladunk az inputban, egyre hosszabb stringekkerülnek a könyvtárba, így tendenciájában az input egyre hosszabb szeleteittömöríthetjük indexbe.

Milyen inputokra a leghatékonyabb az LZW tömörítés?

- ismetlodo mintázatokra

12)

Miért hívjuk a veszteséges kódolást veszteségesnek?

- A visszaállított adat csak az eredeti egy approximációja.

Nevezzen meg veszteséges audiokódolás alkalmazásokat!

- Mp3, aac

Nevezzen meg veszteséges kép kódolás alkalmazásokat!

- jpeg, jpg , mpeg

Nevezzen meg veszteséges video kódolás alkalmazásokat!

- MPEG-2, H.264/AVC

Hogyan definiáljuk a torzítást veszteséges audiokódolásnál?

- hallgatási teszteket használnak a kódolási eredmények szubjektív minőségének meghatározására

Mi a JPEG konstrukció tapasztalati háttere?

- A tömörítési arány a veszteségmentes módszerek esetén (pl. Huffman, LZW) nem elegendő kép és videó tömörítéshez. JPEG transzformációs kódolást alkalmaz, amely lényegében a következő empirikus észrevételen alapul

Mi a DCT transzformáció célja JPEG kódolásnál?

- DCT transzformáció a kép átalakítja a pixelek egy 8x8 méretű blokkjában található információt a térbeli tartományból a frekvenciatartományba

Mi a veszteség fő forrása JPEG kódolásnál?

- A kvantálási hiba

Milyen tapasztalatot használ a nem-egyenletes JPEG kvantálás?

- kvantálási táblázattal definiáljuk

Mi a futamhossz kódoló tömörítés ötlete JPEG-nél?

- A magasfrekvenciás komponensek durvábban kvantáltak, így a sorvektor végefelé túlnyomóan 0-k állnak. A 0-futamokat “skip-value” párokba kódoljuk, ahol “skip” a futambeli 0-k száma, “value” pedig a futamot követő nemzérus komponens.

Mik a redundancia fő forrásai videoesetén? Hogyan használja ezt ki az MPEG kódolás?

- A képcsoporton belüli tömörítés kezdete egy predikcióskódolás, ahol az egymás utáni képek hasonlóságát használjuk ki. A tömörítés makroblokkszinten történik.