

1. prednáška

CIA Triad - ciele počítačovej bezpečnosti:

- **dôvernosť** (confidentiality) - **informácie** sú prístupné iba **oprávneným** osobám a používateľ **kontroluje** aké informácie sú o ňom **zbierané**
- **integrita** (integrity):
 - integrita **dát** - dáta môžu byť **modifikované** iba **oprávneným** spôsobom
 - integrita **systému** - **systém** vykonáva funkcie **neprerušene** bez **neoprávnenej** manipulácie
- **dostupnosť** (availability) - **systémy** sú **dostupné** a služba je prístupná **autorizovaným** používateľom

Doplňujúce ciele bezpečnosti:

- **autentickosť** - overenie **používateľovej identity** a či vstup do **systému** pochádza z **overeného** zdroja
- **zodpovednosť** - umožňuje **vystopovať** entitu podľa jej **činnosti** v systéme

Pojmy:

- **adversary** (protivník) - **entita**, ktorá je **hrozbou** pre systém
- **útok** - zámerný pokus **obísť** bezpečnostné služby a **narušiť** bezpečnostnú **politiku** systému
- **protiopatrenia** - akcia/prostriedok, ktorá **redukuje** hrozbu, zraniteľnosť alebo útok **elimináciou** alebo **predchádzaním** poškodení. Môže vytvoriť **nové** zraniteľnosti.
- **hrozba** - **porušenie** bezpečnosti, ktoré môže prelomiť **bezpečnosť** a spôsobiť **škodu**
- **riziko** - **pravdepodobnosť**, že daná hrozba **nastane**
- **zraniteľnosť** - **chyba** v dizajne systému, ktorá môže byť **zneužitá** na narušenie **bezpečnosti** systému
- **bezpečnostná politika** - súbor **pravidiel** a **praktík** o tom ako systém **poskytuje** bezpečnostné služby
- **systémové zdroje** - **dáta** v systéme, **služba** poskytovaná systémom alebo **položka** systému (hardvér, softvér, firmware, dokumentácia)

Zraniteľnosti a útoky:

- **zraniteľnosti**:
 - strata **dôvernosti** - neoprávnený **používateľ** môže získať **prístup** do systému
 - strata **integrity** - **dáta** alebo **chovanie** systému boli **pozmenené**
 - strata **dostupnosti** - systém je **nedostupný** alebo **spomalený**
- **útoky**:
 - **aktívne** - pokus o **zmenu** systémových **zdrojov**
 - **pasívne** - pokus **naučiť** sa alebo **využiť** informácie zo **systému** bez **zmeny** systémových zdrojov
 - **vnútorné** - útočník má **prístup** do **systému**, kde používa prostriedky nepovoleným spôsobom
 - **vonkajšie** - iniciovaný **mimo** systému **neoprávnením** používateľom

Aktívne útoky:

- pozostávajú z **modifikácie dát**
- **kompletná** ochrana je **nemožná** - vyžadovala by **fyzickú** ochranu všetkých **zariadení** a **ciest**
- **typy**:
 - **masquerade**
 - **replay**
 - **DOS** útok (Denial of Service)
 - **modifikácia správ**

Pasívne útoky:

- **odpočúvajú** alebo **sledujú** prenosy **dát**
- **učia** sa a **využívajú** informácie zo **systému** bez ovplyvňovania systémových prostriedkov
- **ťažko** detekovateľné, keďže **nemenia** dáta
- obranou je **šifrovanie**
- **typy** - release of message contents & traffic analysis

Následky hrozieb:

- **unauthorized disclosure** (nepovolene prezradenie):
 - **udalosť**, kde entita **získa prístup** k dátam, ku ktorým **nemá** povolenie
 - **typy**:
 - **exposure** (odhalenie)
 - **interception** (zachytenie)
 - **inference** (záver)
 - **intrusion** (prenikanie)
- **deception** (podvod):
 - **udalosť**, kde **autorizovaná** entita dostane **falošné dáta**, ale bude **veriť**, že sú **pravé**
 - **typy**:
 - **masquerade** (maškaráda)
 - **falsification** (falzifikácia)
 - **repudiation** (odmietnutie)
- **disruption** (narušenie):
 - **udalosť**, ktorá **zabráni** korektnému **fungovaniu** systémových **služieb** a **funkcií**
 - **typy**:
 - **incapacitation** (zbavenie právnej spôsobilosti)
 - **corruption** (korupcia)
 - **obstruction** (obštrukcia)
- **usurpation** (osvojenie):
 - **udalosť**, ktorá môže viesť ku **kontrole** systémových **služieb** alebo **funkcií** **neautorizovanou** osobou
 - **typy**:
 - **misappropriation** (sprenevera)
 - **misuse** (zneužitie)

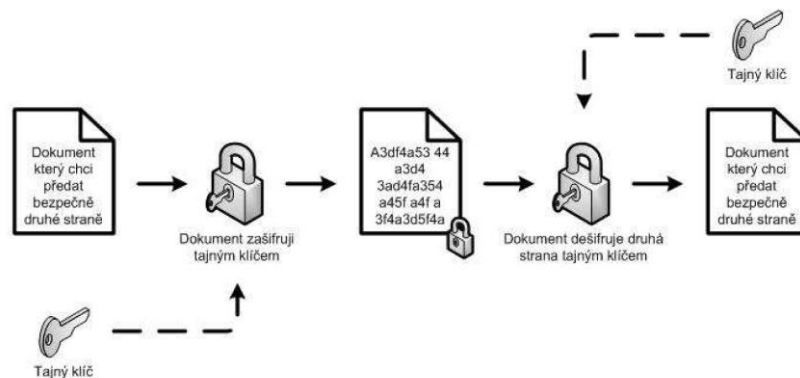
Implementácia bezpečnosti:

- **prevencia** - schéma bezpečnosti v ktorej **nie je možné** vykonať **útok**
- **detekcia** - **kompletná ochrana** je **nemožná**, ale je možné **detegovať** útoky
- **reakcia** - ak je útok **zdetegovaný**, je možné vykonať **protiopatrenia**
- **obnova** - ak **bezpečnosť** je **kompromitovaná**, tak **obnova** systému zo **zálohy** je formou ochrany

2. prednáška

Symetrické šifrovanie:

- **technika** pre utajenie **prenášaných** a **uchovávaných** dát
- **postup:**
 - **plain text** - **dáta**, ktoré budú **šifrované**
 - **šifrovací algoritmus** - vykonáva **substitúcie** a **transformácie**
 - **tajný kľúč** - **kľúč** podľa **ktorého** sa algoritmus **vykoná**
 - **šifrovaný text** - **zašifrované** správa alebo dáta
 - **algoritmus dešifrovania** - šifrovací **algoritmus**, ktorý pracuje v **opačnom smere**



- **odosielateľ** a **prijímateľ** musia mať **kópiu** kľúča, ak ho má niekto **iný** a vie **algoritmus**, tak vzniká **problém**
- **typy útokov:**
 - **dešifrovanie** (cryptanalytic attack):
 - využíva **vlastnosti** algoritmu, aby **vyvodil text** alebo **kľúč**
 - ak **odvodí kľúč**, tak **všetky** správy sú **ohrozené**
 - **útok hrubou silou** (**brute-force attack**):
 - skúša **možné kľúče** na **časti** zašifrovaného **textu**, kým nevznikne **zrozumiteľný preklad**
 - v **priemere** je nutné vyskúšať **polovicu možných kľúčov**

Populárne šifrovacie štandardy algoritmov:

- **DES (Data Encryption Standard):**
 - **náchylnejší** na **brute-force** útoky
 - najviac **študovaným** šifrovacím algoritmom
 - 64-bitov **textu** a 56-bitov **kľúč** => 64-bit **šifrovaného textu**
- **Triple DES (3DES):**
 - **opakuje** DES 3-krát s **využitím 2 alebo 3 kľúčov** (112 alebo 168-bitov)
 - 168-bit **kľúč** **vyriešil** problém s **brute-force** útokmi
 - **pomalý** v SW a používa **malý** 64-bit **block size**
- **AES (Advanced Encryption Standard):**
 - **šifruje** text 10/12/14-krát podľa **dĺžky kľúča**
 - **vlastnosti:**
 - **bezpečnosť** musí byť **rovnaká** alebo **lepšia** ako 3DES
 - **vyššiu** efektívnosť
 - **symetrickú** 128-bitov **block size**

Message Authentication Code (MAC):

- **postup**, ktorý umožňuje **overiť** či získané **dáta** sú **autentické**
- zdroj je **autentický:**
 - iba **odosielateľ** a **prijímateľ** majú **kľúč** - iba **originálny** odosielateľ bol schopný **zašifrovať** správu
 - ak **správa** obsahuje **poradové číslo** - útočník **nemôže zmeniť poradové číslo**

Secure Hash Function:

- transformuje dáta premenlivej dĺžky na **hash hodnotu** pevnej dĺžky
- slúži na **autentifikáciu** - ak dáta neboli **modifikované** tak **hash** je **rovnaký**
- **typy útokov**:
 - **dešifrovanie** (cryptanalytic attack) - využíva **logické nedostatky** algoritmu
 - útok hrubou silou (**brute-force attack**) - **sila hash** proti **brute-force** pre hash kód **dĺžky** n je:
 - **jednosmerná** rezistentná funkcia = $2n$
 - **druhá** rezistentná funkcia = $2n$
 - **rezistentná kolízia** = $2n/2$
- je lepšie ukladať **heslá** v **hash** podobe - ak **hacker** získa **prístup**, tak získa iba **zahashované** heslá

Asymetrické šifrovanie:

- využíva dva **public** a **private** key
- **požiadavky** - **jednoduchá tvorba** kľúčov, **šifrovanie** a **dešifrovanie** správy a **znemožnenie** útočníkovi **vydedukovať private key** alebo **dešifrovať** správu
- **prvky**:
 1. **plain text** - dáta, ktoré budú **šifrované**
 2. **šifrovací algoritmus** - vykonáva **transformácie**
 3. **public** a **private key** - jeden pre **šifrovanie** a druhý pre **dešifrovanie**
 4. **ciphertext** (šifrovaný text) - **zašifrované** dáta
 5. **dešifrovací algoritmus** - vytvorí **plain text** zo **ciphertext** použitím **private key**
- **postup**:
 1. Každý **používateľ** generuje **dvojicu kľúčov**
 2. Každý **používateľ** sprístupní jeho **public key**
 3. Bob **zašifruje** správu **Aliciným public key**
 4. Alice **dešifruje** správu svojim **private key**

Algoritmy asymetrického šifrovania:

- **RSA** - jedna z prvých **public key schém**
- **DIFFIE-HELLMAN KEY AGREEMENT** - prvý vydaný **public key algoritmus**
- **DSS** - **nemôže** byť použitý na **enkrypciu** a **výmenu kľúčov**
- **ECC** - **používa** sa vo väčšine **produktov** a **štandardov**

Certifikáty s využitím asymetrického šifrovania:

- **ktokoľvek** môže **vytvoriť** a rozoslať svoj **public key**, preto sa na **validitu kľúča** používa **certifikačná autorita**

Digitálne podpisy:

- **šifrovanie** s **private key** - umožňuje **autentifikáciu**
- **postup**:
 1. Bob chce **poslať správu** Alici
 2. Bob použije **hash funkciu**, aby vygeneroval **hash správy** ktorú **zašifruje** svojim **private key**, tj. vytvorí **digitálny podpis**
 3. Bob **pošle** správu s **pripojeným podpisom**
 4. Alice prijme **správu** a **podpis** - vypočíta **hash hodnotu správy** a **dešifruje podpis** Bobovým **public key**, ak sa **zhodujú** oba hashe, tak **správa** je **autentická**

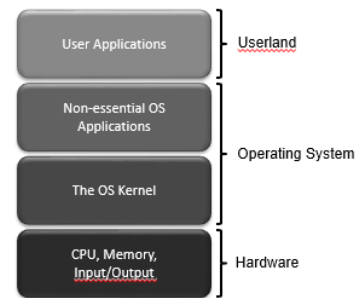
Digitálna obálka:

- chráni **symetrický kľúč** pomocou **public key**
- **postup**:
 1. Bob vytvorí **náhodný symetrický kľúč** na **jedno použitie**
 2. **Zašifruje** správu **symetrickým kľúčom**
 3. **Zašifruje** **jednorazový kľúč** Aliciným **public key**
 4. Pripojí **šifrovaný symetrický kľúč** k **šifrovanej správe** a odošle ju Alici
 5. Iba Alice je **schopná dešifrovať** **jednorazový kľúč** a **správu**

3. prednáška

Operačný systém:

- **interface** medzi **používateľom** a **hardvérom** (I/O, CPU, RAM, storage a pod.)
- **manažuje používateľov, procesy** a ako **aplikácie** prístupujú ku **zdrojom** počítača
- **multitasking** - každý **program** dostane **časť** CPU času na jeho **vykonanie**
- **kernel** - **jadro** OS, ktoré spravuje **low-level** časti **hardvéru**
- **I/O zariadenia** - reprezentované v OS **device drivermi**, **komunikácia** cez **API**
- aplikácie **nekomunikujú** priamo s **hardvérom** ale s **kernelom** cez **systémové volania**
- **proces** - **inštancia** programu, ktorej **dáta** sú načítané zo **storage** do **RAM**, kde mu je priradené **proces ID** (PID), user ID (UID), memory usage a pod., tj. **môžeme** mať viacero **inštancií** rovnakého **programu**



Bezpečnosť súborového systému:

- OS **organizuje** súbory **hierarchicky** do **adresárov**, kde **adresár** obsahuje ďalšie **súbory** alebo **adresáre**
- **I/O** so zariadeniami je cez **/dev/cdrom**
- dáta o **používateľoch** sú v **/etc/passwd** - user:password:UID:GID:FullName:/home/user:/bin/bash
- dáta o **skupinách** sú v **/etc/group** - group:password:GID:<list of users>
- **prístupové práva**: - read, write, execute
 - **práva** sú vo formáte **troch** setov **rwX** (r = 4, w = 2, x = 1), tj. pre **owner-a**, pre **group-u** a pre **other**
 - **sticky bit** - **znemožňuje** vymazanie adresára pokým to nevykoná **owner** alebo **root**
 - **setUID bit**:
 - program sa **spustí** ako keby ho spustil **owner**
 - zneužitelný **softvér** bugom, čo umožní **neautorizovaným** používateľom získať **root práva**
 - **setGID bit** - program sa **spustí** ako keby ho spustil **člen skupiny**

Kernel space:

- **pamäť**, ktorá je používaná **Linux kernelom** a jeho **modulmi**
- nikdy **neswappuje** na **disk** a iba **root** môže **load** a **unload** kernel **moduly**

User space - **pamäť**, ktorá je používaná **ostatnými procesmi**

Rootkits:

- umožňujú útočníkovi **zakryť** svoju **prítomnosť**
- zneužívajú **loadable** kernel **moduly** cez ktoré **interceptuje** systémové **volania** v **kernel space**
- ak je **nainštalovaný**, tak je ho takmer **nemožné** odhaliť a pomôže iba **premazanie systému**

Manažment pamäte:

- **RAM** je **adresný priestor**, ktorý drží **bežiacie** programy, ich **vstupné dáta**, **pracovnú pamäť** systému a pod.
- každý **proces** je **organizovaný** do **rozdielných segmentov**
- **typy segmentov**:
 - **text** - obsahuje **binárny kód** programu
 - **data** - obsahuje **statické** premenné programu **inicializované** programom
 - **BSS** - obsahuje **statické** premenné, ktoré **neboli inicializované**
 - **heap** - **dynamický** segment obsahujúci **dáta** generované **vykonávaním** procesu
 - **stack** - obsahuje **stack data štruktúru**, ktorá **rastie** nadol a **drží volania** subroutines a ich **argumenty**

Virtuálna pamäť:

- OS **dodá** každému **procesu** ilúziu **súvislého adresného priestoru**
- **pamäť** je **rozdelená** do „**pages**“ a OS sleduje, ktoré **pages** sú v **pamäti** a ktoré v **storage**

Page Faults:

- nastáva keď **proces** požiadava o virtuálnu **adresu**, ktorá **nie je** v **pamäti**
- **postup**:
 1. **proces** vyžiada **virtuálnu adresu**, ktorá **nie je** v **pamäti** - vznikne **page fault**
 2. blokový dozorca **vyhodí** z RAM **starý blok**
 3. blokový dozorca **vyhľadá** na disku požadovaný **blok** a **alokuje** ho do **pamäti**

Virtuálny stroj - pre daný **proces** sa OS **tvári** ako keby **bežal** na inej **architektúra** a **OS**

Hybernácia - umožňuje **útočníkovi** získať **hiberfil.sys** z ktoré môže získať **nezašifrované heslá** a pod.

Boot sekvencia:

- **proces** načítania **OS** do **pamäte** z **vypnutého** stavu
- najprv je vykonaní **kód** uložený vo **firmware** - **BIOS** (Basic I/O System), ktorý načíta „**second-stage boot loader**“, ktorý načíta **zvyšné** časti **OS** a presunie mu **kontrolu**
- na zabránenie **narušenia** chodu BIOSu sa používa **BIOS heslo**, ktoré **nedovolí** aby sa načítal „**second-stage boot loader**“ bez autentifikácie **heslo**

Hádanie hesiel:

- **dictionary attacks** - hesla zo **slovníka** sú **zahashované** a porovnávané so **zahashovaným heslom**
- **sťaženie** hádania hesiel je možné použitím „**salt**“

Salt value a hashed passwords:

- **RNG** hodnota (**salt**) je priradená ku **heslu** a uložená do **systému**
- pri prihlásení je **heslo zašifrované** so **salt value** a **hash** je porovnávaný s **hashom** uloženého hesla, tj. **nikdy** sa **pôvodne** heslo **nedešifruje**
- **salt value** slúži na tri účely:
 - **nevznikne duplicita** hesiel, aj keď **viacerí** používatelia majú **rovnaké** heslo (kvôli rôznemu **salt**)
 - **zvyšuje zložitosť** pri **hádaní** hesiel
 - **znemožňuje** zistiť či **jedna** osoba používa **rovnaké** heslo na **rôznych** systémoch

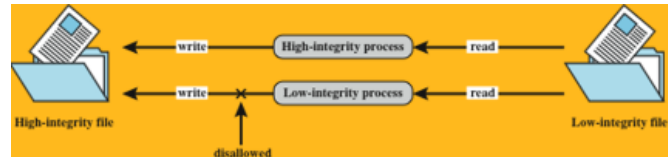
Sandbox:

- **bezpečnostný mechanizmus** slúžiaci na **spúšťanie** neotestovaného **kódu** a **programov**
- má prísne **riadený prístup** ku systémovým **zdrojom**, sieti a pod.
- ide o príklad **virtualizácie**

4. prednáška

Bell-LaPadula model:

- **formálny** model pre riadenie prístupu
- **subjekty** a **objekty** majú priradené **bezpečnostné triedy**, ktoré **riadia** spôsob prístupu **subjektu** ku **objektom**:
 - **top secret** > **secret** > **confidential** > **restricted** > **unclassified**
 - **subjekt** má **security clearance**
 - **objekt** má **security classification**
- **prístupové módy**:
 - **read** - **subjekt** môže iba **čítať** z **objektu**
 - **append** - **subjekt** môže iba **písať** do **objektu**
 - **write** - **subjekt** môže **čítať** aj **písať** do **objektu**
 - **execute** - **subjekt** **nemôže** čítať ani písať do **objektu**, ale môže ho **spustiť**
- ak definované **viaceré úrovne dát**, tak **podmienky** sú definované **multilevel security (MLS)**:
 - subjekt **vysokej** úrovne **nemôže** dodať **informácie** subjektu **nižšej** úrovne, ak prúd informácií **nie** je **deklasifikovaný** **autorizovaným** používateľom
 - **MLS systém** musí spĺňať:
 - **no read up**:
 - **subjekt** môže **čítať** objekt s **menšou** alebo **rovnakou** bezpečnostnou **triedou**
 - ide o **simple security property**, tj. **ss-property**
 - **no write down**:
 - **subjekt** môže **písať** iba do objektu s **väčšou** alebo **rovnakou** bezpečnostnou **triedou**
 - ide o **star property**, tj. ***-property**
- **Mandatory Access Control (MAC)** - prístup je **zamietnutý**, ak **ss-property** a ***-property** nie sú **splnené**
- **Discretionary Access Control (DAC)** - **ds-property**, ktorá umožňuje **používateľovi** alebo **roly** dať **prístup** do súboru inému **používateľovi**, ale pod **MAC** obmedzeniami.



BIBA Integrity model:

- rieši **prípady** keď **dáta** musia byť **dostupné** používateľom vo **viacerých** alebo **všetkých** bezpečnostných **triedach**, ale môžu byť **upravené** iba **autorizovanými** agentami
- základné **prvky** sú **podobné** ako v **BLP** (Bell-LaPadula), keďže obe pracujú so **subjektami** a **objektami**, kde **každému** je priradený **level integrity**

Clark-Wilson Integrity model:

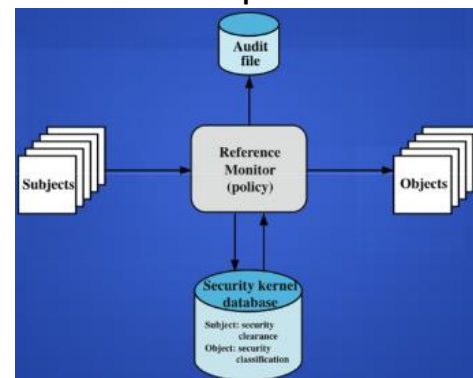
- zameraný na **komerčné použitie** - postavený na **dvoch** konceptoch **komerčných bezpečnostných postupov**:
 - **korektné transakcie** - používateľ môže **upravovať** dáta iba spôsobmi, ktoré **zachovávajú** ich **integritu**
 - **separácia povinností medzi používateľmi** - používateľ, ktorý má **právo** vytvoriť **korektnú transakciu** **nemúsí** mať právo ju **vykonať**

Chinese Wall model:

- zameraný na **komerčné použitie** kde môže vzniknúť **konflikt záujmov**
- využíva **DAC** (Discretionary Access Control) a **MAC** (Mandatory Access Control)
- príklad - market analyst **nemôže** ponúkať svoje **služby** spoločnosti, ak má **insider info** o ich **kompetitoroch**

Referenčné monitory:

- **kontrolné elementy** v **hardvéry** a **OS**, ktoré **regulujú** prístup **subjektov** ku **objektom** na báze **bezpečnostných parametrov** subjektov aj objektov
- má prístup ku **bezpečnostnej databáze kernelu** - **list** prístupový **práv** každého **subjektu** a **úroveň klasifikácia** každého **objektu**
- vyžaduje **bezpečnostné pravidlá** (no read up, no write down)
- **vlastnosti**:
 - **kompletné sprostredkovanie** - **bezpečnostné pravidlá** sú overované pri **každom** prístupe
 - **izolácia** - **referenčný monitor** a **databáza** sú chránené pred **neautorizovaným prístupom**



RBAC (Role-Based Access Control):

- **access control** na báze **rolí** a umožňuje **implementovať BLP multilevel bezpečnostné pravidlá**
- na implementáciu **MLS** (Multilevel security) musíme **špecifikovať**:
 - obmedzenia na **používateľov**
 - obmedzenia na **povolenia**
 - obmedzenia na **priradenie** používateľských **rolí**
 - **definície**
- parametre **role**:
 - **r-level** - indikuje **najvyššiu** bezpečnostnú **klasifikáciu** ku ktorým **má** prístup na čítanie
 - **w-level** - indikuje **najnižšiu** bezpečnostnú **klasifikáciu** ku ktorým **má** prístup na písanie

Databázy s MLS (Multilevel Security):

- s implementáciou **MLS** sa **zvyšuje** zložitosť **prístupových práv** a **dizajn** databázy
- problémom je **granularita klasifikácií**, ktorá je **riešiteľná** cez:
 - **celú databázu** - databáza je klasifikovaná ako **confidential** alebo **restricted** a spravovaná na **servery**
 - **tabuľky** - aplikácie majú priradené **klasifikáciu** na úrovni **tabuliek**
 - **stĺpcov** - aplikácie majú priradené **klasifikáciu** na úrovni **stĺpcov**
 - **riadky** - aplikácie majú priradené **klasifikáciu** na úrovni **riadkov**
 - **elementy** - najzložitejší spôsob, kde **jednotlivé elementy** môžu byť **klasifikované**
- vynucuje ***-security** pravidlo (no write down) pri **vpisovaní** do databázy:
 - ak **používateľ** s **nízkymi** právami pridá riadok s **primary key** ktorý už **existuje** v riadku **vyššej** úrovne:
 - databáza **notifikuje** používateľa, že **daný** primary key už **existuje** a **odmietne request**
 - databáza **nahradí** existujúci **riadok** s klasifikáciou **nižšej** úrovne

TPM (Trusted Platform Module):

- hardvérový **čip**, ktorý je v **matičnej doske**, **smart karte** alebo **CPU** a je využívaný na **Trusted Computing (TP)**
- **TPM** generuje **kľúče**, ktoré sú **zdieľané** medzi **zraniteľnými** komponentami a **slúžia** na:
 - **autentifikáciu** pri **boot-e**:
 - **digitálny podpis** je **verifikovaný** pre každú fázu **boot-u** a **TPM** udržiava **log** procesu
 - **certifikáciu**:
 - **TPM** certifikuje konfigurácie pomocou **TPM private key**, aby vytvoril **dôveru**, že konfigurácie **neboli** pozmenené, keďže iba **TPM** má **private key**
 - **enkrypciu**:
 - **TPM** má **master secret key**, ktorý je **unikátny** pre daný **stroj**
 - z **kľúča** sa generuje **secret encryption key** pre akúkoľvek **konfiguráciu** na danom **stroji**
 - ak dáta sú **šifrované** na stroji s **danou** konfiguráciou, tak sú **dešifrovateľné** iba ak **daný** stroj má tú **rovnakú konfiguráciu**
- **komponenty**:
 - **I/O** - všetky **príkazy** prechádzajú cez **I/O komponent** ktoré **komunikuje** s ostatnými **komponentami**
 - **kryptografický co-processor** - **procesor** špecializovaný na **enkrypciu/dekrypciu** (RSA, symmetric...)
 - **key generation** - vytvára **RSA public/private key pair** a **symetrické kľúče**
 - **HMAC engine** - **algoritmus** používaný pri **autentifikačných protokoloch**
 - ...

Protected Storage funkcia:

- **TPM** generuje a drží **dešifrovacie kľúče** v **hierarchie**
- na **vrchole** hierarchie je **storage root key** z ktorého sa **generujú** ďalšie **kľúče** a je **prístupný** iba pre **TPM**

Common Criteria – CC:

- **ISO štandard** definujúci **sadu bezpečnostných požiadaviek** na zväčšenie **dôvery** v **bezpečnosť** IT produktov:
 - **target of evaluation (TOE)** - časť systému, ktorá je **hodnotená** či **spĺňa** CC
 - **funkcionálne požiadavky** - definujú **žiadané bezpečnostné chovanie** systému
 - **požiadavky dôvery** - základ pre dosiahnutie **dôvery**, že bezpečnostné **opatrenia** sú **efektívne**

5. prednáška

Buffer Overflow/Overrun:

- útok využívajúci **pretečenie zásobníka** k vykonaniu **požadovaného príkazu**
- vzniká napr. kvôli **neoverovaniu** veľkosti dát **vkladaných** do **buffera** v **spúšťanom** programe
- môže **nastať** aj v:
 - **stack (Stack Buffer Overflow)** - nastáva keď **buffer** je v **stack-u** v podobe **lokálnej** premennej
 - **heap (Heap Overflow)** - nastáva keď **buffer** je v **heap-e**
 - **global data** - nastáva keď **buffer** je v **globálnych/statických** dátach programu
- môže **útočiť** na:
 - **systémové utility**
 - **network service daemon**
 - **knižnice kódov**
- ak proces je **spustený** ako **root**, tak **škodlivý kód** ktorý pretiekol bude **spustený** s **root** právami
- môže viesť ku **korupcii** dát, **pádu** systém, **prebraniu** kontroly a pod.
- častý v **C** programoch, kde sa **manuálne** alokuje **pamäť** a používajú **funkcie** ako **gets()**, **sprintf()** a pod.
- **ochrany**:
 - **compile-time** - potenciál na útok je **odhalený** už pri **kompilácii**:
 - **preferovať** nové jazyky s **manažovaním** pamäte a **kontrolami** pri kompilácii
 - aplikovať **techniky bezpečného** programovania (inšpekcia kódov a pod.)
 - používať bezpečné **rozšírenia** a **knižnice**
 - kontrolovať **stack** na znaky **korupcie**
 - **run-time** - **deteguje** a **abotuje** program v ktorom **nastáva** útok:
 - **blokovať** vykonanie kódu v **stack-u**
 - **randomizácia** pozície dát v **pamäti** s náhodným **posunom** pre každý **proces**
 - použitím „**guard pages**“ medzi **kritickými** regiónmi v **pamäti**

Shellcode:

- **strojový kód**, ktorý je **vložený** do **buffera** pri **Overflow** útokoch a **pokúša** sa **zväčša** otvoriť **shell**

Injekčné útoky:

- vznikajú keď **útočník** ovplyvní **vstupné** dáta **prenášané** ako **parametre** medzi **pomocnými** programami
- najčastejšie v **skriptovacích** jazykoch, ktoré využívajú **ďalšie** programy a utility (Perl, PHP, Python a pod.)

SQL Injection útok:

- **útočník** využije **vstup** od **používateľa** na vytvorenie **SQL query** aby získal **dáta** z **databázy**
- **obrana** - **validácia** vstupných dát

Code Injection útok:

- **útočník** vloží **kód**, ktorý je **spustený** cieľovým **systémom**, napr. prostredníctvom **PHP**

Cross Site Scripting (XSS) útoky:





- pozostávajú z **vloženia skriptu** (napr. JavaScript) do **HTML generovanej** stránky v **prehliadači** útočníkom, ktorý je **následne** spustený v **prehliadači** obeť
- hrozby sú **phishing**, **hijacking**, **krádež cookies/identity**, spustenie **škodlivého** kódu u **klienta** a pod.
- **ochrana** na strane **klienta**:
 - **proxy** - sleduje **prenos** dát medzi **prehliadačom** a **serverom** a hľadá **špeciálne HTML znaky**
 - aplikačný **firewall** - **analýza** stránok za účelom **nájst hyperlinky**, ktoré môžu viesť k **úniku** dát
 - **auditovací systém** - **monitoruje** spustenia **JavaScript kódu**

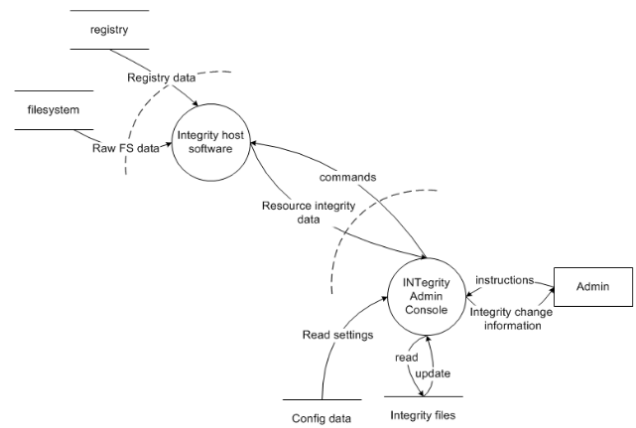
6. prednáška

Modelovanie hrozieb:

- **metodológia** na **identifikovanie**, **ohodnotenie** a **zdokumentovanie** hrozieb, útokov a zraniteľností
- cieľom je **minimalizovať riziká** počas **návrh**, **implementácie** a **údržby**
- aplikuje sa na **rôzne časti** systému
- **fázy**:
 - **identifikácia zdrojov**
 - **definícia cieľov bezpečnosti**
 - **návrh architektúry** aplikácie
 - **bezpečnostný profil** - dátové toky, vstupné a výstupné body
 - **identifikácia hrozieb a rizík** - **STRIDE**
 - **dokumentácia** hrozieb a rizík
 - **ohodnotenie** hrozieb
- **STRIDE** - systém **kategorizácie** hrozieb do **skupín**:
 - **spoofing**:
 - **maskovanie** totožnosti a **vydávanie** sa za inú **entitu**
 - **mitigácia**: **autentifikácia** - certifikáty, digitálne podpisy, cookies a pod.
 - **tampering**:
 - **úprava** dát
 - **mitigácia**: kontrola **integrity** - digitálne podpisy, ACLs (Access Control List) a pod.
 - **repudiation**:
 - **tvrdenie**, že daná akcia **nebola** vykonaná
 - **mitigácia**: logovanie, digitálne podpisy a pod.
 - **information disclosure**:
 - odhalenie **informácií** entite bez **dostatočnej** autorizácie
 - **mitigácia**: šifrovanie, ACLs a pod.
 - **Denial Of Service (DOS)**:
 - **znemožnenie** alebo **zhoršenie** prístupu **používateľov** ku službe **zaplavením** cieľa **packetmi**
 - **mitigácia**: ACLs, filtrovanie, kvóty a pod.
 - **elevation of privilege**:
 - **zneužitie** chyby na získanie **vyšších** práv **bez** oprávnenia
 - **mitigácia**: **autorizácia** - ACLs, groups/roles a pod.

Author's notes: „Neučte sa tabuľku ani diagram. Dávam ich sem len aby ste nezabudli z akých prvkov sa model hrozieb skladá.“

ELEMENT	S	T	R	I	D	E
						
External Entity	✓		✓			
						
Process	✓	✓	✓	✓	✓	✓
						
Data Store		✓	?	✓	✓	
						
Data Flow		✓		✓	✓	



7. prednáška

MAC spoofing:

- **útok**, pri ktorom **útočník** zistí **MAC adresu** cieľového stroja, ktorú **nastaví** na svojom **stroji** a **odpojí** cieľový **stroj** a zapojí **svoj** čo spôsobí, že **switch** bude **posielať** dáta **jeho** stroju, ktorý sa **tvári** ako **pôvodný** stroj
- **protiopatrenia**:
 - **zablokovanie portu** keď stroj sa **vypne** alebo **odpojí**
 - **znemožniť** aby v sieti boli **duplicitné MAC adresy**

ARP (Address Resolution Protocol):

- **ARP** slúži na **získanie MAC adresy** z **IP adresy** v **rovnamej LAN sieti**
- **ARP** sa používa v **IPv4** a **zistené adresy** sú **uložené** do **ARP cache**
- **ARP spoofing/poisoning**:
 - **vydáva** sa za **iný počítač** a neustále **posiela** svoju **MAC adresu** na **každý request IP adresy**
 - **ARP cache** je **aktualizovaná** pri **každej odpovedi** aj keď neposlala **žiaden request**
 - **ochranou** je použitie **statickej ARP tabuľky**

ICMP útok - pošle **packet**, ktorý presahuje **maximálnu veľkosť**, je poslaný **IP fragmentáciou** a poskladaný v **cieľovom systéme** čo spôsobí **pád** systému kvôli **pretečeniu zásobníka** (buffer overflow)

SYN Flood:

- zväčša ide o **DOS** (Denial of Service) útok, ale **môže byť kombinovaný** s iným útokom, napr. TCP hijacking
- zahlučuje cieľ s **TCP connection requests** rýchlejšie než ich **server** dokáže **spracovať**
- **server** odpovie s **SYN/ACK** na ktorý ale **nikdy** nedostane **odpovedať** na ktorú bude **čakať 3 minúty**
- nakoniec **server prestane** kompletne **odpovedať** a nastáva **DOS**
- **riešením** sú **SYN cookies** alebo **obmedzenie počtu nových** pripojení

Optimistic ACK útok:

- využíva **TCP congestion control**
- **útočník** pošle **viacero ACK** pre **segmenty**, ktoré **neprija**
- **server** si bude myslieť, že má **veľké množstvo** dostupného **bandwidthu**
- **server** bude **zvyšovať cwnd** (congestion window) až dokým **nepresiahne** svoj **limit**
- útok môže byť **vykonaný** medzi **viacerými** servermi čo spôsobí **výpadok** danej **časti siete**

TCP Session hijacking:

- pokus o **prevzatie** kontroly nad **network session**
- **IP spoofing**:
 - **útočník** pošle **packet** na **IP adresu** a **tvári** sa ako keby **pochádzal** z **inej**
 - **blind spoofing** - útok pochádza z **akéhokoľvek zdroja** - **musí poznať ACK number**
 - **non-blind spoofing** - útok pochádza z **rovnakého subnet-u** - s **packet sniffers** získa **ACK number**
- **packet sniffers**:
 - **čítajú informácie** tečúce sieťou **zachycovaním packetov**, používaním **ARP poisoning** a pod.
 - môžu slúžiť na **analýzu** siete alebo na **kradnutie** informácií
 - **obranou** je **šifrovanie packetov**

Port knocking:

- pokus o **vytvorenie** spojenia so **zablokovaním** portom pre jeho **otvorenie**
- **ochranou** je **časovo závislá knock sequence**

HTTPS:

- **zabezpečený** hypertextový prenosový **protokol**
- využíva **asymetrické šifrovanie** a **TCP/IP port 443**
- **postup**:
 - **obe strany** vygenerujú **public** a **private key**
 - **vymenia si public keys**, ktoré si **overia** digitálnym **podpisom** od **certifikačnej authority**
 - **šifruje** dáta cez **SSL** alebo **TLS protokol**, čím chráni pred **odpočúvaním** a poskytuje **overenie identity**

8. prednáška

Firewall:

- **súbor** opatrení, ktoré chránia **vnútornú** sieť pred **neoprávneným** prístupom z **vonka**
- **princíp** spočíva v **kontrole paketov** a povolenie/zamietnutia **komunikácie** podľa **pravidiel**
- **pakety** môžu byť:
 - **prijaté** (accepted) - **prešiel** cez firewall
 - **ukončené** (dropped) - **neprešiel** a **žiadna** akcia nebola **vykonaná**
 - **zamietnuté** (rejected) - **neprešiel** a bola vyslaná **správa** o **odmietnutom** pakete
- pri **kontrole paketov** sa preveruje **source** a **destination IP**, **porty**, prítomnosť **vírusov** a pod.
- **Blacklist** a **Whitelist** - *Author's notes: „Mali by ste poznať princíp blacklist a whitelist.“*
- **typy**:
 - **packet** (stateless) **filters** - pracuje na **nízkej** úrovni TCP/IP a **pustené** sú pakety, ktoré **spĺňajú** sadu pravidiel. Sú **rýchlejšie** ale **neloguje** si žiadne **dáta** o paketoch.
 - **stateful filters** - **zaznamenáva** každé spojenie a **rozhoduje** či ide o **počiatočný** paket nového spojenia, **existujúceho** spojenia, alebo sa jedná o **neplatný** paket
 - **application layer** - pracuje na **aplikačnej** úrovni TCP/IP, zachytáva **pakety aplikácie**
- **tunely** - slúžia na **šifrovanú** komunikáciu medzi **klientom** a **serverom** pomocou **tunneling protokolu**, keďže **obsah paketov** nie je bežne **šifrovaný**

Secure Shell (SSH): - sieťový **kryptografický** protokol pre **klient-server** TCP spojenie

- **postup**:
 - **klient** sa pripojí s **TCP session**
 - **klient** a **server** si vymenia **informácie** a vyberú **protokoly**, ktoré oba **podporujú**
 - **vymenia** si **public key** a vytvoria **shared secret session key** na ďalšiu komunikáciu
 - server pošle formy **autentifikácia** pre **klienta**, buď cez **login** alebo **private key authentication**
 - po **overení** je klient **pripustený** ku **CLI**

IPSec:

- **zoznam** protokolov pre **ochranu** sieťovej **komunikácie**
- každý **paket** je **šifrovaný** a každý **protokol** pracuje v **režime**:
 - **transport mode** - **dodatočná** IPSec **hlavička** je **vložená** do paketu
 - **tunnel mode** - **nový** paket s IPSec **hlavičkou** je vytvorený a **pôvodný** paket je v ňom **encapsulated**

Virtual Private Networking (VPN):

- umožňuje **počítaču** odosielať/prijímať **dáta** v rámci **verejných** sietí, akoby išlo o **privátnu** sieť
- **typy**:
 - **remote access VPN** - **klient** má prístup na **intranet** cez VPN endpoint **Network Access Server (NAS)**
 - **site-to-site VPN** - **bezpečný** most **medzi** viacerými **fyzickými** sieťami

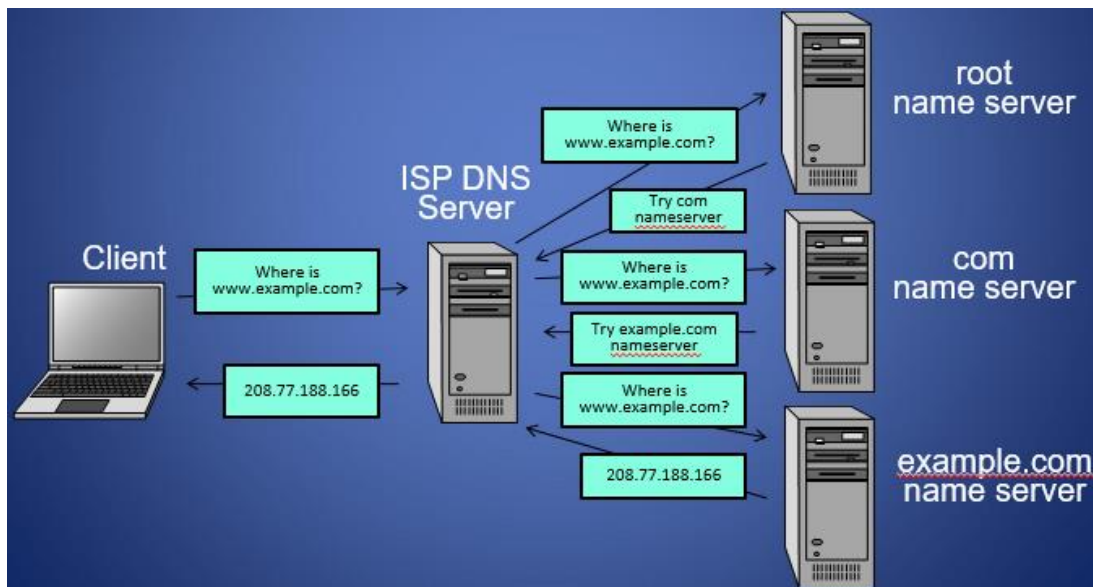
Intrusion Detection System (IDS):

- **správca IDS** spracováva **dáta** od **IDS senzorov**, ktoré **monitorujú** či **nedošlo** ku **narušeniu** podľa **súboru pravidiel**. Ak správca **zdeteguje** narušenie, **vyhlási** poplach.
- **IDS** je schopný **odhaliť**:
 - **masquarader** - **útočník** používajúci **falošnú** identitu
 - **misfeasor** - **oprávnený** používateľ, ktorý vykonáva **neoprávnené** akcie
 - **clandestine user** - **používateľ** snažiaci sa **zmazať** logy
- ako aj **útoky**:
 - **port scan** - zisťuje aké **porty** sú **otvorené**
 - **Denial of Service (DOS)**
 - **malware**
 - **ARP spoofing**
 - **DNS cache poisoning**
- **rule-based** - **rozpoznáva** útoky podľa **pravidiel** známych **profilov útoku**
- **statistical** - **štatisticky** rozoznáva kedy sa **user** alebo **host** sa chovajú **nezvyčajne**

Domain Name System (DNS):

- protokol **aplikačnej vrstvy**, ktorý prekladá **hostnames** do **IP adries**
- zabezpečuje **distribuovanú databázu** nad internetom, ktorá **ukladá** rôzne **záznamy**:
 - **address** - IP spojená s **názvom** rozhrania
 - **mail exchange** - **mail server** v doméne
 - **name server** - **autoritatívny** server pre **doménu**
- **DNS server referuje** na iný **menom** servera, **nie** pomocou **IP**
- **doménové mená** - dva alebo viac **mien** oddelených **bodkou** (google.com, kde „com“ je **top-level doména**)
- **DNS caching**:
 - **DNS** si **cachuje** záznamy a **maže** ich po určitej **dobe** (**time-to-live - TTL**)
 - **OS** a **prehliadače** si taktiež **udržia** **DNS cache** lokálne z ktorých sa **záznam** po istom čase **vymaže**
- **útoky**:
 - **DNS pharming/hijacking**:
 - **presmerovanie** obete na **falošnú** stránku so zámerom **phishingu** (získanie loginu obete...)
 - dosiahnutý zmenou **host súboru** v **počítači** obete alebo **zneužitím** chyby **zabezpečenia DNS**
 - **DNS cache poisoning**:
 - **nacachovanie zlých DNS záznamov** v **DNS servery** poskytovaním **falošných** záznamov
 - **možný** ak server **nerešpektuje** identifikátory alebo **akceptuje** **nevyžiadané DNS záznamy**
 - **obrana**:
 - používať **náhodné identifikátory** pre **dopyty**
 - **kontrolovanie** identifikátorov
 - **randomizácia portov** pre **DNS dopyty**
 - nasadenie **DNSec**
- **DNSec**:
 - **protiopatrenie** voči **DNS cache poisoning/spoofingu**
 - používa **asymetrického šifrovania**
 - **odpoveď** od DNS je **podpísaná** pomocou **public key** pri každom **kroku**

Preklad domény na IP cez DNS:



9. prednáška

IE image crash:

- chyba v prehliadači môže viesť ku **DOS** – napr. zmenou veľkosti obrázka na stránke (Internet Explorer)

Mobilný kód (Mobile Code):

- ide o **spustiteľný** program odoslaný cez **sieť** - **Javascript**, **JVM**, **Java Plugins**, **ActiveX**

Cookies:

- **informácie** uložené v počítači **uložené** špecifickým **serverom**
- keď **navštívime** danú stránku, **cookie** je **znova odoslaná** na server
- slúži na **udržanie** informácie nad **sessions**
- môže obsahovať **citlivé informácie** ako heslá, informácie o kreditnej karte a pod.

Elektronická pošta:

- **e-mail** - spôsob **posielania** a **prijímania správ** cez elektronické **komunikačné systémy** (Internet)
- **prvky**:
 - **MUA** - softvér na **čítanie** a **posielanie** elektronickej pošty - **e-mailový klient**
 - **MSA** - softvér, ktorý **akceptuje** správu od **MUA** a pracuje s **MTA**
 - **MTA** - softvér, ktorý **prenáša** správy medzi **používateľmi**
 - **MDA** - softvér, ktorý **zodpovedá** za **prenos správy** do schránky **prijemcu**
- **proces**:
 1. **odosielateľ** napíše **email** cez **klienta**
 2. **odošle email** použitím **MSA** a **MTA**
 3. **MTA** prečíta **adresu** a vyhľadá **doménu** v **DNS** aby zistil **mail exchange servery**
 4. **odošle email** použitím **SMTP**
 5. **prijímateľ** získa email v jeho **klientovi** pomocou **POP/IMAP**
- **SMTP** - **server**, ktorý sa snaží **doručiť** správu **adresátovi** podľa **druhu adresy** akú dostane
- **POP** - **protokol**, ktorý **preberie správu** a drží ju **dokým** si ju adresát **nevyzdvihne**

10. prednáška

Databáza:

- **štruktúrovaná** kolekcia **dát**
- **relačné databázy**: - „I am not going to explain this, sorry.“
 - **primary/foreign key**
 - **view**
 - **rows & attributes**
- využíva **database management system (DBMS)** na **spravovanie** databázy:
 - **Database Access Control**:
 - **centralized administration** - malí **počet** privilegovaných **userov** môže meniť **access rights**
 - **ownership-based administration** - iba **vlastník** môže **meniť práva** používateľov
 - **decentralized administration** - **vlastník** môže dať **práva** na **úpravu** access rights **userov**
 - **príkazy** na **manažment** prístupových práv používateľov:
 - **grant**
 - **revoke**
 - **prístupové práva** - select, insert, update, delete...
- **Role-based Access Control (RBAC)**:
 - uľahčuje **administráciu** a zvyšuje **bezpečnosť** kategorizáciou **používateľ** na:
 - **vlastník**
 - **administrátor**
 - **používateľ**

Inference:

- proces spájania viacerých povolených queries a dedukcie čím sa dostaneme ku dátam bez povolenia
- inference channel - cesta ktorou nepovolené dáta boli získané
- protiopatrenia:
 - detekcia databázovým návrhom - odstránenie inference channel-u zmenou štruktúry databázy
 - detekcia počas dopytu - deteguje útok počas dopytu a preruší ho

Štatistické databázy:

- poskytujú štatistické dáta
- typy:
 - čisto štatistické databázy - drží iba štatistické dáta a opravený user má prístup ku celej databáze
 - bežné databázy so štatistickým prístupom - obsahujú záznamy a users majú prístup ku daným častiam databázy podľa prístupových práv
- ochrana voči inference:
 - query restriction - odmietne query, ktoré môže viesť ku kompromitácií
 - perturbation - akceptuje query, ale odpoveď je iba približná
 - data perturbation:
 - data swapping - hodnoty sú vymenené aby sme nemohli získať záznamy, ale je zachovaná štatistika
 - generovanie upravenej databázy s približnými hodnotami
 - output perturbation - štatistiky, ktoré dostaneme sú upravené
 - random-sample query:
 - vhodná na veľké databázy
 - princíp:
 1. user pošle query $q(C)$, ktoré je nastavené na $X(C)$
 2. systém nahradí $X(C)$ vzorovou podsadou $X(C)$
 3. systém vypočíta query na vzorke a vráti hodnotu
- Tracker útok:
 - tracker obchádza obmedzenie veľkosti query jej rozdelením na viaceré menšie queries a ich výsledky následne spojí, čím sa dostane ku špecifickým dátam
 - iné restrictions:
 - query set overlap control - limituje overlap medzi novými a starými queries
 - partitioning - spája záznamy do viacerých vzájomne vylučujúcich sa skupín
 - query denial and information leakage

Šifrovanie databázy:

- nevýhody:
 - manažment kľúčov - user musí mať dešifrovací kľúč na prístup ku databáze
 - neflexibilita - keď časť databázy je šifrovaná, tak sa zvyšuje zložitosť hľadania záznamov
- aktéri schéma šifrovania:
 - data owner - organizácia, ktorá vytvára dáta
 - user - entita, ktorá zadáva queries
 - client - frontend, kde users zadávajú queries, ktoré sú spustené na servery so šifrovanými dátami
 - server - dostáva šifrované dáta od data ownera

Cloud risky v bezpečnosti:

- **zneužitie a zle využitie cloud vypočítavania** - útočník zneužíva free trials a **využíva resources**, čo môže viesť k DoS, code útokom a pod.
- **nezabezpečené rozhrania a APIs** - user ich používa na prácu s cloudom a sú ľahkým cieľom pre **útočníkov**, ak **nepožadujú** nejaké formy **autentifikácie** a pod.
- **malicious insiders** - spoločnosť si najíma **cloud**, ale nemôže **ovplyvňovať** samotného **provideru**, a nevie čo deje interne
- **zdieľanie technológie** - služby sú **prístupné** ako **škálovateľné**, avšak väčšina infraštruktúr **nemá** silnú **izoláciu** pri **zdieľaní** hardvéru. Problém je **riešený** izolovanými VMs pre každého klienta, ale ani to nie je 100%
- **strata dát**
- **hijacking účtu alebo služby** - krádež hesla a pod.

Obrana dát:

- **multi-instance model** - poskytuje unikátne **DBMS** na **VM** pre každého **subscribera**, čím získava **kontrolu** nad svojím **prostredím**
- **multi-tenant model** - poskytuje **predefinované** prostredie pre **subscribera**, ktoré je **zdieľané**, tj. **bezpečnosť** je na **providerovi**

11. prednáška

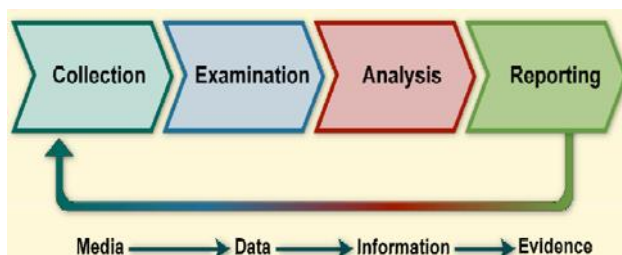
Dáta - špecifický kus digitálnej informácie, ktorá je **sformátovaná**

Forensic science:

- aplikácia vedy na **identifikáciu, zberanie, skúmanie a analýzu dát** so zachovaním **integrity** informácie
- použiteľná pre **monitorovanie logov, obnovu dát, získanie dát, pravidelných zálohách**, auditing, posielaní **logov**, udržiavanie **záznamov, databázy súborových hashov** a pod.

Forensic proces:

- **Collection -> Examination -> Analysis -> Reporting**
- **kroky:**
 - **Collection** - nájdenie **zdroja** a **získanie dát**. Pri dátach treba pozrieť na ich **volatilitu, námahu** pri zbere, **integritu** a pod.
 - **Examination** - **posúdenie** a **vytiahnutie** zmysluplných **informácií** zo zozbieraných **dát**
 - **Analysis** - **študovanie** a **analýza** dát na vytvorenie **záveru**
 - **Reporting** - **vytvorenie záveru** z daných **dát**



Na **forensic science** môžu byť použité **dáta** zo **súborov, siete, OS** a **aplikácií**:

- **súbory:**
 - **data file** - **súbor usporiadaných** informácií s **unikátnym** názvom
 - **zber a skúmanie súborov:**
 - čas **úpravy, prístupu, vytvorenia** a pod.
 - pomocou **forensic toolkit** - file viewer, **prístup** ku súborovým **metadátam** a pod.
- **OS:**
 - **volatilné dáta** - **OS** ich **vykonáva** v **RAM** - bežiacie **procesy**, otvorené **súbory** a pod.
 - **nevolatilné dáta** - data files, logy a pod.
- **sieť:**
 - **zdroje dát** - firewalls, router, proxy, DHCP a pod.
- **aplikácie:**
 - autentifikácia, logy a pod.
 - **typy:** email, chat, file sharing a pod.

12. prednáška

Cybercrime - kriminálna aktivita, v ktorej **počítač** alebo **sieť** je použitý ako **nástroj**, **cieľ** alebo **miesto** tejto **aktivity**

Intellectual Property:

- **nehmotný asset**, ktorý **obsahuje ľudské nápady** a **znalosť**
- **typy:**
 - **copyright:**
 - chráni **ucelenú myšlienku**, nie **myšlienku samotnú**
 - **owner** má **práva** na vydanie **kópií**, ich **úpravu** a **distribúciu**
 - **patent:**
 - **dáva** vynálezcovi **práva** na **vynález**
 - **typy:**
 - **utility** - nový spôsob, stroj a pod.
 - **design** - nový dizajn
 - **plant** - nová rastlina
 - **trademark:**
 - **slovo, názov, symbol** na označenie **produktu**
 - slúži na **zamedzenie** vzniku **značky**, ktorá by mohla **zmiestť** ľudí
 - **nezamedzuje** vzniku **rovnakých** alebo **podobných** produktov pod **inou značkou**

DMCA (Digital Millennium Copyright Act):

- **zákon** ochraňujúci **digitálne vlastníctvo** - **zákaz prístupu** a **kopírovania** práce **bez povolenia**
- **výnimky:**
 - **fair use** - môžeš použiť v **review**, **komente** alebo **diskusii**
 - **reverse engineering** - ak má **user vlastníctvo kópie**, tak môže **reverse enginernúť**, ale cieľom **nemôže byť duplikovania funkcionality**
 - **encryption research** - **dešifrovanie** je **povolené** pre prospech **výskumu** šifrovania
 - **security testing** - **hackovanie** je **povolené** pre prospech **výskumu** bezpečnosti
 - **personal privacy** - ak by došlo ku **narušeniu privacy** môžeš **ignorovať** DMCA

DRM (Digital Rights Management):

- **systémy** a **procedúry**, ktoré **zabezpečujú**, že **vlastníci** digitálnych práv dostanú **zaplatené** za svoju **prácu**
- **musí:**
 - poskytnúť **ochranu obsahu** voči **nepovolenému** použitiu
 - podporovať **rôzne typy** obsahu
 - podporovať **obsah** na **rôznych** platformách
 - podporovať **distribúciu** na viacerých **médiách**
- **DRM komponenty:**
 - **poskytovateľ obsahu** - má **práva** na **obsah**
 - **distribútor** - poskytuje spôsoby **predaja**
 - **consumer** - cez distribútorov **získa prístupu k obsahu**
 - **clearinghouse** - zodpovedá za **finančnú transakciu** pri **kúpe** licencie na obsah
- **DRM architektúra:**
 - **role:**
 - **vlastníci obsahu**
 - **poskytovatelia**
 - **consumers**
 - **služby:**
 - manažment **identity**
 - manažment **obsahu**
 - manažment **práv**
 - **funkcie:**
 - **security/encryption** - enkryptovanie **obsahu** a podpísanie **licencie**
 - **authentication/authorization** - **identifikovanie** skupín
 - **billing/payments** - zber **poplatkov** od **consumerov**
 - **delivery** - dodanie **obsahu** **consumerom**

Privacy kritéria:

- **anonymity** - je možné **použiť** systém/službu bez **poskytnutia** identity
- **pseudonymity** - **nemusí** odhaliť svoju **identitu**, ale je **zodpovedný** za **použitie** systému/služby
- **unlinkability** - je možné **využívať** systém/službu **bez toho**, aby sa **použitia** dali **prepojiť**
- **unobservability** - je možné **používať** systém/službu **bez toho** aby niekto **sledoval** tvoju **činnosť**

Privacy a Data Surveillance:

- **data transformation** - **enkrypcia** dát na zachovanie **privacy**, ale aby sa **dali** stále **analyzovať**
- **anonymization** - **odstránenie** osobných častí z **dát**
- **selective revelation** - **odhalenie** iba **non-private** dáta
- **immutable audit** - identifikácia **kde** a **ako** boli **dáta** **sprístupnené**
- **associative memory** - **softvér** na rozoznávanie **patternov** na tvorbu **prepojení** v **dátach**

Ethnical issues:

- **dáta** môžu byť ľahko **zneužívané**, preto vznikli **Codes of Conducts**
- **Codes of Conducts**:
 - ide o **guidelines**, nie **rules**
 - **summary**:
 - **confidentiality**
 - **zodpovednosť** za svoju prácu
 - **integrita** a **honesty**
 - **zlepšenie štandardov**
 - **dignity** a **worth** ostatných

Unique otázky – zoradenie podľa výskytu v daných rokoch, tj. na začiatku sú 2019 a nižšie sú staršie:

1. Popíšte 3 vlastnosti bezpečnosti – **1. prednáška**
2. Ransomware napadol počítač, ktoré základné vlastnosti bezpečnosti boli porušené a prečo?
 - typ **škodlivého softvéru**, ktorý **zablokuje OS alebo šifruje dáta** a **požaduje** od obete **výkupné** za obnovenie prístupu
 - **porušené boli**:
 - **dôvernosť** - k **dátam** sa dostala **neautorizovaná entita**
 - **integrita** - **dáta boli zašifrované**, teda **modifikované**
 - **dostupnosť** - používateľ **nedokáže pristupovať** k dátam
3. Používateľ chce poslať elektronicky podpísaný xml súbor na daňový úrad. Čo musí spraviť?
 - **2. prednáška – Digitálny podpis**
4. Bol zadaný e-mail (spam) so skrátenou url, napr. goo.gl. O aký typ emailu/útoku ide?
 - Ide o **phishing**, tj. **krádež údajov** kde si obeť bude myslieť, že je na genuine stránke.
5. Koľko kľúčov má byť distribuovaných pri symetrickom šifrovaní pre 4 ľudí?

Vzorec: $(n * (n - 1)) / 2$

A - BCD (3)

B - CD (2)

C - D (1)

$3 + 2 + 1 = 6$

6. V akej podobe sú ukladané heslá v OS a prečo je to bezpečné?

Ku **heslá** sú pridané **salt value** (random value) a **tie** sú ukladané v **zahashovanej podobe** v OS a sú **bezpečné** kvôli tomu, lebo ak sa k nim **dostane útočník** tak **nezíska** heslá v **použiteľnej podobe**.

7. Referenčný monitor, popis a využitie – **4. prednáška**

8. XSS - Cross-site scripting – **5. prednáška**

9. Janko si zmazal súbor dovolenka.jpg, aké základné vlastnosti bezpečnosti boli porušené?

Nedostatok informácií. Ak je Janko **vlastník**, tak asi **nejde** o **porušenie bezpečnosti**?

Inak ide asi iba o **porušenie dostupnosti** a **dôvernosti**.

10. Buffer overflow/Pretečenie zásobníka, popis, princíp, možné riešenie problému – **5. prednáška**
11. Symetrické a asymetrické šifrovanie + digitálne podpis + autentifikácia správy + hash funkcia – **2. prednáška**
12. Systémy IDS, popis a rozdelenie – **8. prednáška**
13. DNSSEC – **8. prednáška**
14. BLP model a jeho využitie – **4. prednáška**
15. SYN Flood – **7. prednáška**
16. Certifikáty s využitím asymetrického šifrovania – **2. prednáška**
17. ARP spoofing popis a riešenie – **7. prednáška**
18. Ako funguje elektronická pošta? – **9. prednáška**
19. Modelovanie hrozieb + popis + fázy + STRIDE – **6. prednáška**
20. Čo je forenzná analýza a jednotlivé fázy procesu analýzy – **11. prednáška**
21. Virtuálna pamäť v OS + page fault – **3. prednáška**

- 22. Autorizácia v databázach – **4. prednáška**
- 23. Riešenie bezpečnosti prostredníctvom sandboxu – **3. prednáška**
- 24. BIBA – **4. prednáška**
- 25. Princíp fungovania HTTPS – **7. prednáška**
- 26. Princíp fungovania DNS – **8. prednáška**
- 27. VPN princípy a využitie – **8. prednáška**
- 28. Čo je DRM a princíp fungovania – **12. prednáška**
- 29. Patent, Copyright, Trademark – **12. prednáška**