1 Introduction

1.1 Pocitacova bezpecnost

- zaobera sa zabezpecenim informacii v pocitacoch.

Vlastnosti definujuce bezpecnost:

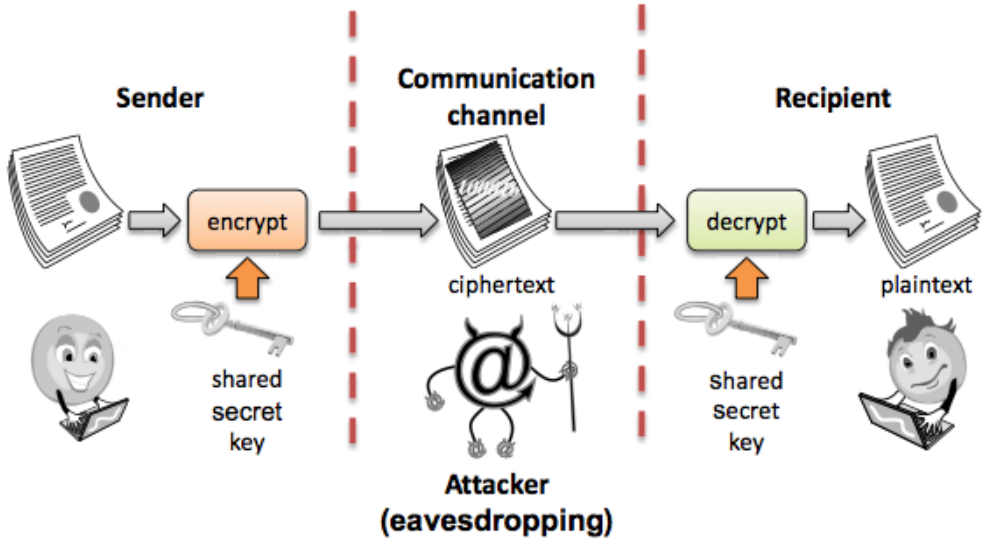
1.1.1 Utajenost

Mechanizmus na zabranenie pristupu k informaciam neautorizovanym osobam ktori nemaju na to dostatocne prava.

Nastroje pre zabezpecenie utajenia:

1.1.1.1 Sifrovanie (Encryption)

Je to transformacia informacie s pouzitim privatneho sifrovacieho kluca do podoby, ktora nema ziadnu informacnu hodnotu (obsah). Takato zasifrovana informacia moze byt precitana (transformovana spat na informacny obsah) len prijemcom, ktory vlastni (pozna) desifrovaci kluc, ktory je v niektorych pripadoch rovnaky ako sifrovaci kluc. Ak prijemnca (utocnik), desifrovaci kluc nema k dispozicii, tak zo zasifrovanych dat nezisti ziaden informacny obsah, jedine to, ze existuje. Velmi casto sa vyuziva ak chceme docielit utajenost.



1.1.1.2 Kontrola pristupu (Access Control)

Mechanizmus ako rozlisovat medzi pouzivatelmi ze kto ma alebo nema pristup k urcitym informaciam (napr. k suborom v operacnom systeme, alebo k jednotlivym castiam weboveho portalu).

Nastroje pre kontrolu pristupu

Autentifikacia

- overenie pravosti identity alebo roly pouzivatela v systeme. Overenie identity prebieha tak, ze pouzivatel spoji svoju identitu s tajnou informaciou, napriklad heslom, otlackom prsta, alebo mobilnym telefonom.

Autorizacia

- stanovenie ci ma osoba alebo system pristup k urcitej operacii. Tato kontrola pristupu je vacsinou zadefinovana v tzv. bielom zozname, pricom vsetky operacie su implicitne zakazane a v dannom zozname sa definuje kto aku operaciu smie vykonavat.

Fyzicka kontrola pristupu

Zavedenie fyzickych mantinelov (barier) k obmedzeniu pristupu ku chranenym vypoctovym zdrojom. Napriklad: zamky na dverach, pocitace v miestosti bez okien, alebo specificky konstrukcne postavene budovy, kde elektromagneticke signaly, nemozu vniknut ani uniknut do/z budovy.

1.1.1.3 Integrita

Ak komunikuju 2 alebo viac strany navzajom pomocu nejakoho komunikacneho media, tak cheme mat istotu ze tie prenasane informacie su v konzistentnom tvare, tzn. ze ak niekto nieco poslal, tak prijamca dostal tuto spravu v nezmenenom tvare. Cize aby v komunikacii nedoslo k nejakej modifikacii tejto spravy, ci uz nejakym utocnikom alebo nejakym inym sposobom (nezavinena modifikacia, napr. zly prenosovy kanal, rusenie wifi siete a pod). Pokial dojde k nezavinenemu poruseniu integrity, tak na to existuju korekcne kody, ktore vyziadaju nanovo komunikaciu alebo komunikaciu zrusia.

Nastroje:

- **Zalohovanie**: periodicka zaloha dat. Ak sa nieco stane s povodnymi datami, mame k dispoziciu zalohovane data
- **Kontrolne sum**y: funkcia ktora vypocita ciselnu hodnotu (kontrolnu sumu) zo suboru, tzn. ak bol subor co i len trochu upraveny (staci aby mal otoceny len 1 bit) bude kontrolna suma odlisna.
- **Datova korekcia kodov**: mechanizmus na ukladanie dat v takej podobe, ze male zmeny mozu byt jednoducho detekovane a automaticky opravene

1.1.1.4 Dostupnost

Pokial ma pouzivatel pristup k urcitej sluzbe alebo zdrojom, tak mal by mat k tomu pristup vzdy ked o to poziada.

Nastroje na zabezpecenie dostupnosti:

- **Fyzicka ochrana**: zabezpecenie zariadeni, budovy, atd. tak aby nemohli byt narusene tretou osobou.
- **Vypoctove redudancie**: automaticke nahradzanie zdrojov, pokial jeden z nich vypadne. (napr. pri diskoch ak su zapojene v RAID0, RAID5)

1.1.2 Dalsie koncepty bezpecnosti

1.1.2.1 Istota (Dovernost)

Manazment doverihodnosti (dovernost medzi systemom a pouzivatelom) je na takom stupni ze system a pouzivatel si navzajom doveruju.

Zavisy na:

- **Bezpecnostnej politike**: specifikuje spravanie ludi alebo systemov vramci seba a vramci ostatnych (povinna alebo diskretna kontrola pristupu)
- **Bezpecnostne prava**: opisuju spravanie ze co pouzivatel moze vykonat a co nemoze
- **Bezpecnostna ochrana**: opisuje mechanizmy na zaistenie prav a politiky na zaklade identity pouzivatela

1.1.2.2 Autenticnost

Stanovenie, ze udaje, postupy a prava vydane osobami alebo systemom su prave.

Hlavnym nastrojom na zaistenie autenticnosti je **digitalny podpis (DP)**.

DP: - je analogicky rucnemu podpisu, ktory sluzi ako dokaz autorstva, resp. suhlasu s obsahom dokumentu.

- je to urcita datova struktura, ktora je zavisla na dokumente, vznika hasovanim tohto dokumentu a tento kod je zasifrovany sukromnym klucom, ktory je jednoznacnym vlastnictvom vlastnika dokumentu.

Nie je mozne dosiahnut 100% autenticnosti udajov.

1.1.2.3 Anonymita
Vlastnost ze urcite zaznamy alebo transakcie nepripadnu ku ziadnemu jednotlivcovy.

Nastroje na zabezpecenie anonymity:

- **Agregacia**: kombinacia dat od viacerych pouzivatelov, takze zverejnene ciastky alebo priemery nemozu byt spojene so ziadnym konkretnym pouzivatelom
- **Mixacia**: agregovanie informacii z viacerych stran a spajanie ich do zloziek, ktore sa nedaju rozlozit.
- **Proxy**: doverihodni agenti, ktori nahradzaju skutocnu identitu pouzivatela, vacsinou vo vacsich organizaciach sa jedna o nejake systemy ktore nahradzaju identitu skutocneho pouzivatela
- **Pseudonym**: fiktivna identita pouzivatela, ktory predstiera identitu

1.1.3 Hrozby a utoky

1.1.3.1 Odpocuvanie
Odposluch informácií určených pre niekoho iného v priebehu ich prenosu cez komunikačný kanál. Najvacsie riziko odpocuvania je v nezabezpecenych lokalnych pocitacovych sietach.

1.1.3.2 Modifikacia informacii
Neautorizovana modifikacia informacie, tzn. utocnik odchiti informaciu odosielatela, upravi ju a posle ju dalej prijimatelovi.

1.1.3.3 Denial-of-service (DOS) – Odmientnutie sluzby
Zahltenie cieľa nepotrebnými požiadavkami natoľko, že nestíha obsluhovať bežných používateľov (služby). Prikladom je napr. Spam, ktory zahlti emailovy server natolko, ze nakoniec padne, tzn. nebude stihat spracovavat tolko poziadaviek naraz.

1.1.3.4 Maskarada
Pouzivanie identity cudzej osoby ziskanych osobnych udajov.

1.1.3.5 Odmietnutie (Repudiation)
Tento utok vznika v tedy ak system nezaznamenava co pouzivatelia v dannom systeme vykonavaju alebo ak vznikne modifikacia napr. logovacich suborov s tym ze utocnik napr. urobi v systeme nejake zmeny a nasledne upravi logovacie subory tak, ze danne zmeny urobil iny pouzivatel.

1.1.3.6 Correlation a traceback
Identifikacia identity napr. pouzivatela na zaklade pospajanych ciastkovych informacii z viacerych zdrojov.

1.1.4 Desat principov bezpecnosti

1.1.4.1 Ekonomicky mechanizmus
Cim je system jednoduchsi tym jednoduchsie sa docieluju vlastnosti bezbecnosti.

1.1.4.2 Vychodzie bezpecne hodnoty (Fail-safe defaults)
Prednastavene hodnoty aplikacie by mali byt relativne bezpecne.

1.1.4.3 Kompletna mediacia
Zmyslom tohto princípu je, že každý prístup k prostriedkom, musí byť kontrolovany pre splnenie systémovej ochrany a taktiez by v systeme nemal existovat mechanismus, ktory vie bezpecnostne prvky systemu obist.

1.1.4.4 Otvoreny design
Podľa tohto principu, bezpečnostna architektúra a návrh systému by mali byt verejne dostupné.

- bezpečnosť by sa mala spoliehať iba na udržanie kryptografickych kľúčov v tajnosti.
- umožňuje systému byť posudzovanym viacerimi stranami, čo vedie k skorému odhaleniu a náprave bezpečnostných zraniteľností spôsobených chybami v návrhu.
- je opakom prístupu známeho ako cierna skrinka, ktora sa snaží dosiahnuť bezpečnost tým, že drží kryptografické algoritmy tajné, pricom tento pristup nepriniesol v histrorii ziadny uspech.

1.1.4.5 Separacia privilegii
Princip hovori o tom ze pri pristupe k zabezpecenym informaciam alebo pri volani nejakej operacie by malo byt vyzadovane viacnasobne potvrdenie o pristupe k dannemu zdroju.

1.1.4.6 Minimalne prava (Least privilege)
Princip spociva v udelovani prav pre system alebo pre pouzivatela len v tom rozshahu kolko prav potrebuje.

Každý program alebo používateľ počítačového systému by mali fungovať s holým minimom prav nevyhnutnych pre správne fungovanie. Ak je tento princip dodrzany, tak zneuzitie prav je obmedzene a potencionalne skody su minimalizovane.

1.1.4.7 Minimalny spolocny mechanizmus (Least common mechanism)
V systémoch s viacerými užívateľmi, by mechanizmy umožňujúce pristup k spolocnym zdrojom, mali byt minimalizovane. Napríklad, ak súbor alebo aplikácia musia byť prístupné viac ako pre jedneho používateľa, potom títo používatelia by mali mať oddelené kanály, pomocou ktorych pristupuju k týmto prostriedkom, aby sa zabránilo nepredvidatelnym následkom, ktoré by mohli spôsobiť bezpečnostné problémy.

1.1.4.8 Psychologická prijateľnosť
Tento princip hovorí, že používateľské rozhranie by malo byť dobre navrhnuté a intuitívne, a všetky prvky súvisiace so zabezpečením by mali byt pre pouzivatela intuitivne a jednoducho pouzitelne.

1.1.4.9 Pracovny faktor
Princip hovori o tom, ze kolko usilia a nakladov by sa malo vynalozit na implementaciu bezpecnostnych prvkov systemu, vzhladok k tomu, kde danny system bude pouzivany.

1.1.4.10 Zaznamenavanie kompromitacie
Zaznamenavanie informacii ze co sa v systeme deje. Odradenie utocnika utocit na system, z dovodu ze ho mozme neskor vypatrat.

1.1.5 Sposoby definovania kontroly pristupu

1.1.5.1 Kontrola pristupu na zaklade matice
Matica kontroly pristupu je tabulka definujuca opravnenia.

- Kazdy riadok v tabulke je definovany subjektom, ktorym moze byt pouzivatel, skupina, alebo system, ktory uskutocnuje operacie.
- Kazdy stlpec v tabulke je definovany objektom, ktorym moze byt subor, priecinok, dokument, zariadenie, zdroj alebo nejaka ina entita pre ktoru chceme definovat prava.
- Kazda bunka tejto tabulky obsahuje pridelene prava medzi dannym subjektom a dannym objektom
- Pristupove prava mozu obsahovat akcie ako su: citanie, zapisovanie, kopirovanie, spustanie, mazanie a komentovanie
- Prazdna bunka v tabulke znamena ze zdiadne opravnenia neboli pridelene

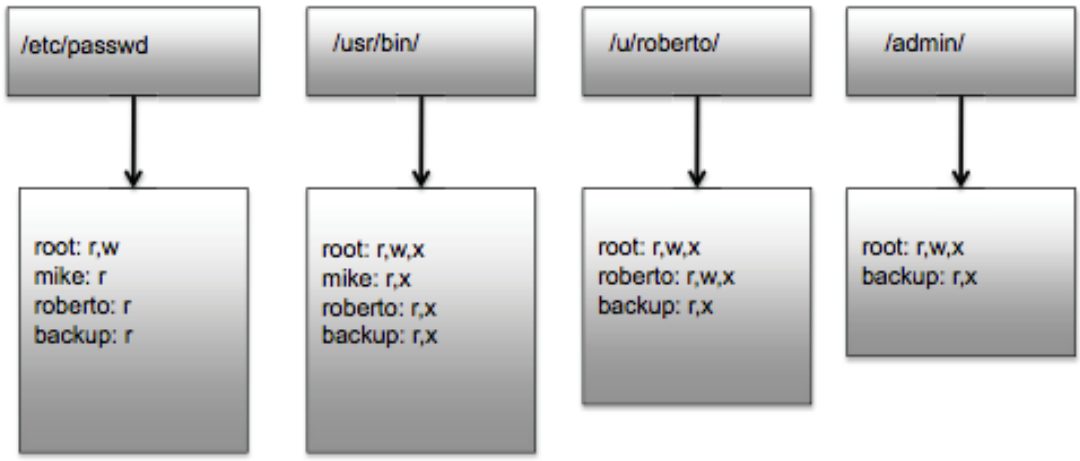
Vyhody: jednoduchost
Nevyhody: realizacia implementacie v pripade ze mame vela subjektov a objektov

	/etc/passwd	/usr/bin/	/u/roberto/	/admin/
root	read, write	read, write, exec	read, write, exec	read, write, exec
mike	read	read, exec		
roberto	read	read, exec	read, write, exec	
backup	read	read, exec	read, exec	read, exec
...	...	...	...	...

1.1.5.2 Kontrola pristupu na zaklade zoznamu

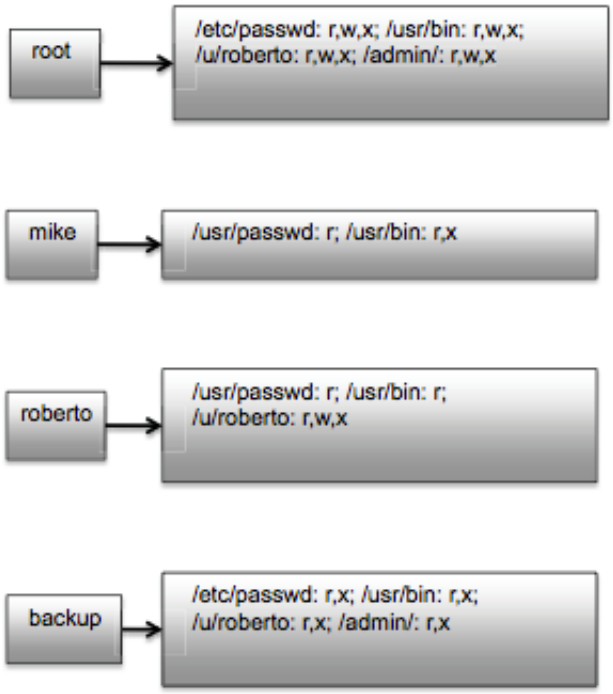
Definuje pre kazdy objekt zoznam nazyvany aj zoznam kotroly pristupu, ktory pozostava zo vsetkych subjektov ktori maju pristup k dannemu objektu a kazdy z tychto subjektov ma definovane pristupove prava (citanie, zapisovanie, ...) k danemu objektu.

Vyhody: jednoducha implementacia
Nevyhody: pomerna zlozitost vyhladavania vsetkych objektov pre danny subjekt



<u>Druha moznost implementacie zoznamu kontroly pristupu</u>

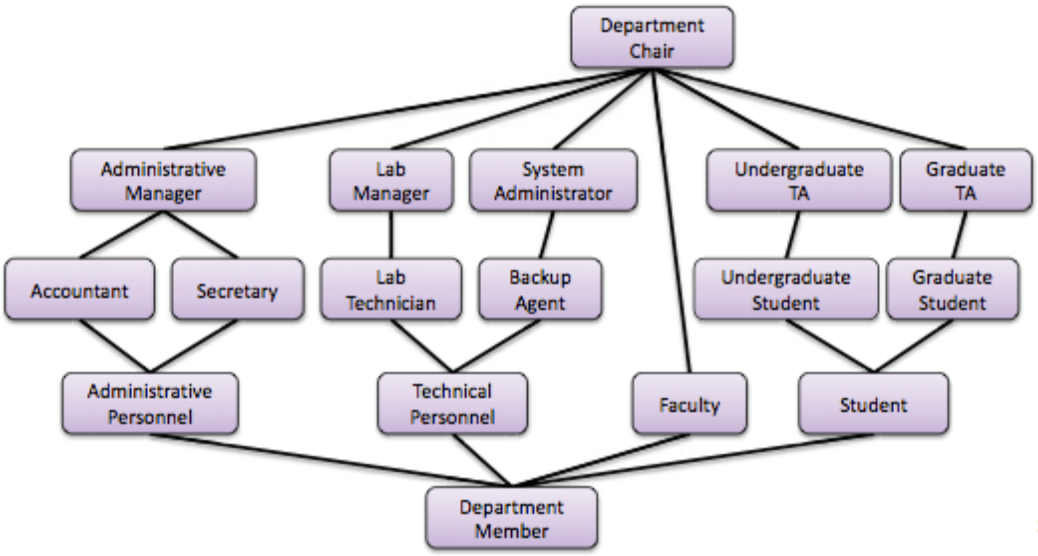
Definuje pre kazdy subjekt zoznam objektov pre ktore ma subjekt nejake specificke prava.



Vyhody: vyhladavanie
Nevyhody: podobne ako pri matici, zlozita implementacia

1.1.5.3 Kontrola pristupu na zaklade roli

Definovanie roli a specifikovanie opravneni pre tieto role, ktore sa potom priradzuju subjektom. Rola vie dedit prava od inej roli a tym sa nam vytvara struktura (strom) roli.



1.1.6 Kryptografické pojmy

1.1.6.1 Sifrovanie a desifrovanie

Dovoluje relativne bezpecnu komunikaciu dvoch entit A a B v nezabezpecenom kanali, ktory je predmetom odpocuvania. Tzn. Entita A zasifruje spravu tajnym klucom, odosle ju entite B a entita B tuto spravu desifruje svojim tajnym klucom. Ak je sprava odchytena, tak utocnik nevie spravu desifrovat ak nema k dispozicii tajny kluc.

Sprava (M) je nazyvana ako holy text. Odosielatel prekonvertuje holy text M do zasifrovanej podoby s pouzitim sifrovacieho algoritmu (E), ktoreho vystupom je kryptovany text (C). Prijimatel prijme kryptovany text C a ten desifruje pomocou desifrovacieho algoritmu (D) na holy text M.

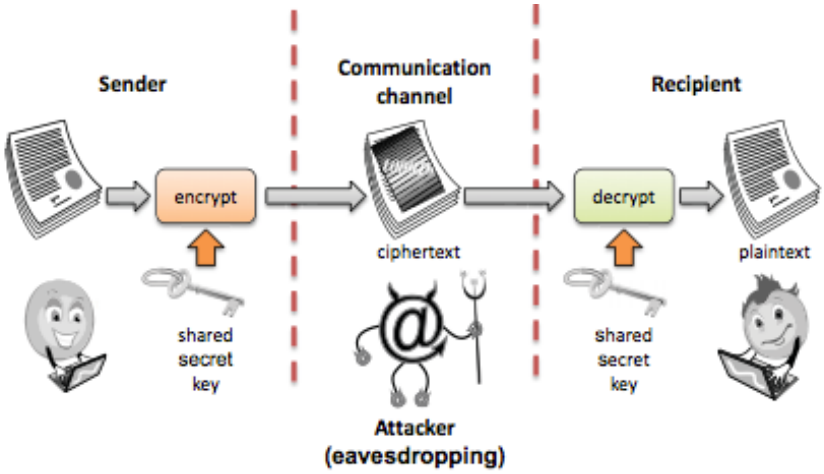
C = E(M)
M = D(C)

1.1.6.2 Cezarova sifra

Je druh šifry, pri ktorej je každé písmeno správy posunuté o n pozícií ďalej v abecede, pričom n môže byť 1 až m − 1, kde m je počet znakov príslušnej abecedy. Je to bez klucovy sifrovaci system.

1.1.6.3 Symetricke sifrovanie

Na sifrovanie aj desifrovanie sa pouziva len jeden privatny kluc. Je pomerne rychle. Kolko komunikujucich stran mame, tolko klucov musime rozdistribuovat (n(n - 1)/2 klucov).

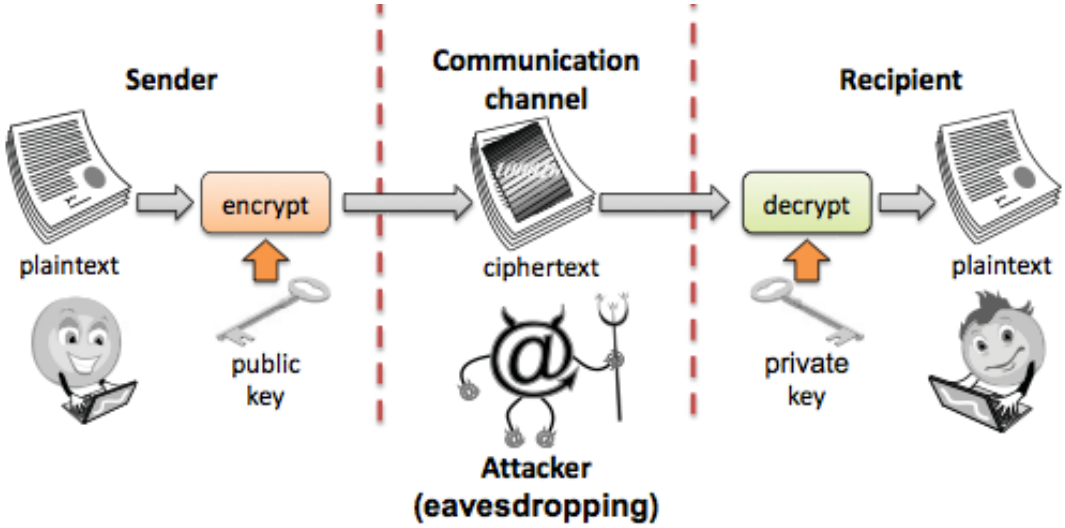


1.1.6.4 Symetricka distribucia klucov

- Entita A vyberie kluc a ten sa fyzicky doruci z entity A do entity B.
- Entita C vyberie a doruci kluc do entity A aj B
- Pouziju predoslu utajenu komunikaciu na dohodnutie kluca
- Ak maju spolocneho znameho entitu C, tak utajenou komunikaciou cez neho vymenia kluc

1.1.6.5 Asymetricke sifrovanie

Dva kluce, privatny a verejny. Odosielatej zasifruje spravu verejnym klucom prijimatela. Prijimatel desifruje spravu svojim privatnym klucom. Velkou vyhodou AS je, ze vieme v nom vyuzit digitalny podpis na zabezpecenie autencity spravy.



1.1.6.6 Digitalny podpis

- je analogicky rucnemu podpisu, ktory sluzi ako dokaz autorstva, resp. suhlasu s obsahom dokumentu.
- je to urcita datova struktura, ktora je zavisla na dokumente, vznika hasovanim tohto dokumentu a tento kod je zasifrovany sukromnym klucom, ktory je jednoznacnym vlastnictvom vlastnika dokumentu.

verifikacia DP:

- sprava sa desifruje verejnum klucom (to znamena, ze sprava bola zasifrovana sukromnym klucom a jej jednoznacnym vlastnikom je odosielatel - tato vlastnost sa nazyva autentizacia), dostaneme hashovaci kod. Ak ma odosielal a prijemca rovnake hasovace kody, tak mame istotu ze sprava nebola zmenena (nebola narusena utocnikom). Ak sprava bola zmenena tak na vystupe nedostaneme rovnaky hashovaci kod (tato vlastnost sa nazyva integrita).

vlastnosti DP:

- forma skupiny bitov, ktorych hodnoty su zavisle na podpisanej sprave
- vyuziva urcitu jedinecnu informaciu (sukromny kluc), krora je vlastictvom drzitela podpisu a zabezpecuje ochranu pre falsovanim a odmietnutim
- realizacia a implementacia digitalneho podpisu by mala byt relativne lahka
- falsovanie DP by malo byt vypoctovo obtiazne
- ulozenie kopie DP v pamati by malo byt jednoduche

1.1.6.7 Kryptograficke Hashovacie funkcie

- pozname: MD5 (128 bitov dlhy otlacok), SHA-1 (160 bitov dlhy otlacok)

Hashovacia funkcia

- je funkcia, ktora transformuje spravu s lubovolnou dlzkou na vystupnu hodnotu vyjadrenu fixnym poctom bitov bez pouzitia kluca. Vystupna hodnota oznacovana ako hasovaci kod sluzi

ako identifikator. Vystupny hasovaci kod sa oznacuje: h = H(M), kde h ma pevnu dlzku desiatky az stovky bitov.

Hasovacie funkcie musia spnat 2 zakladne poziadavky:

1. jednocestnost

- hasovacia hodnota sa pocita lahko
- na zaklade hasovacej hodnoty je tazke najst dokument s tou hasovacou hodnotou

2. odolnost voci kolizii.

- je velmi tazke najst 2 rozlicne dokumenty s rovnakou hasovacou hodnotou.

1.1.6.8 Certifikacna autorita

Zabezpecuje, ze verejny kluc zo strany B, ktory prijme strana A skutocne pochadza od B. Neexistencia CA by mohla sposobit, ze A posiela verejny kluc B, avsak C odchyti tento verejny kluc, zmeni ho za svoj a posle dalej B. Ak B bude chciet komunikovat s A bude sifrovat spravu verejnym klucom C a tak nepovolana strana C bude schopna odpocuvat komunikaciu bez vedomosti B o tejto situacii, kedze verejny kluc nenesie v sebe informaciu o majitelovi.

1.1.6.9 Hesla

Sekvencia roznych znakov ulozenych v zasifrovanej podobe.

Utoky:

Slovnikovy utok – skusanie roznych znamych slov a beznych hesiel
Utok hrubou silou – skusanie vsetkych kombinacii znakov

1.1.6.10 Socialne inzinierstvo

• pretexting: vytvorenie príbeh, ktorý presvedčí správcu alebo operátora aby mu poskytol napr. heslo danneho pouzivatela.
• baiting: ponúka akysi "dar", napr. v podobe CD alebo USB, za ucelom nalakat pouzivatela aby spustil obsah na dannom mediu a tym si infokovat svoj system
• Quid pro quo: nieco za nieco. Utocnik moze predstierat identinu niekoho ineho a moze vytvorit socialny tlak na svoju obet a tak ziskat nejake cenne informacie.

1. Physical Security

1.1.Zámky (Locks)

Fyzická bezpečnosť – každý predmet, ktorý tvorí prekážku voči neoprávnenému prístupu (napr. zámky, trezory, zavory, strážne psy, okná, steny, dvere ...). Naskytá sa tu otázka,či je fyzické zabezpečenie IT koncernom, nakoľko sa kladie veľký dôraz na zabezpečenie siete pred počítačovými útokom (použitím antivírusových programov, firewall, systémy detekcie prieniku ...), ale čo ak sa útočník dostane priamo do serverovej miestosti, alebo do skrine sieťového zapojenia?

Deštruktívny vs nedeštruktívny vstup

Deštruktívny vstup (násilné vniknutie)– vyznačuje sa použitím hrubej sily na prekonanie fyzického zabezpečenia, čo ma za následok poškodenie alebo zničenie zámku alebo okolitých objektov ako sú devre, steny ... za použitia páčidiel, závitorezov (vrtákov), kladív...

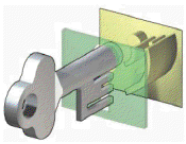
Nedeštruktívny vstup – porušenie ochrany bez zanechania stôp porušenia. Sem môžeme zaradiť napr. otvorenie zámku pomocou šperhákov.

Zámky – už po celé stáročia predstavujú zámky základ fyzického zabezpečenia. Avšak väčšina zámkov môže byť ľahko prekonaná použitím nedeštruktívnych metód (niekedy za pár sekúnd s použitím ľahko dostupných nástrojov).

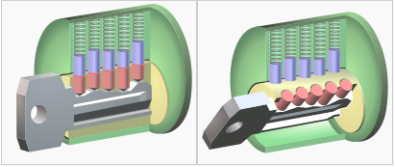
V minulosti bolo otváranie zámkou doslova umením, avšak v dnešnej dobe internetu sú metódy a nástroje pre otvorenie zámku ľahko dostupné.

Typy zámkov

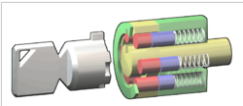
- **TSA** (Transportation Security Administration) – americká agentúra, ktorá bola založená po 11. Septembri, a jej hlavnou úlohou je monitorovanie bezpečnosti leteckej dopravy. Vytvorily zoznam pravidiel pre kontrolu batožín aj bez prítomnosti pasažierov. Špeciálne TSA chválené zámky umožňujú ochranu ale aj kontrolu pasažierov.
- **WARDEN** – zámky používané vstredoveku, avšak ich dizajn sa používa aj v dnešnej dobe. Jedná sa o zámku, ktorá je tvorená sadou prekázok, oddielov, čím sa má zabezpečiť to, že sa zámka nedá otvoriť pokiaľ sa nevloží správny kľúč. Tento kľúč musí mať zárezy odpovedajúce prekážkam v zámku, čím sa umožní voľné otáčanie kľúča. V dnešnej dobe sa používajú len pri objektoch s veľmi nízkym zabezpečením.



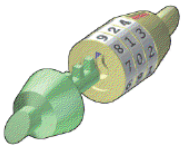
- **Yale Pin Tumbler** – využíva kolíky rôznych dĺžok, aby sa zabránilo otvoreniu bez správneho kľúča. Pre otočenie kľúčom je dôležité, aby všetky kolíky boli v jednej rovine.



- **Tubular** - axiálny, alebo radiálny je typom Tumbler zámku s 6-8 pinmi usporiadanými do kruhu a odpovedajúci kľúč ma dutý, alebo valcový tvar.

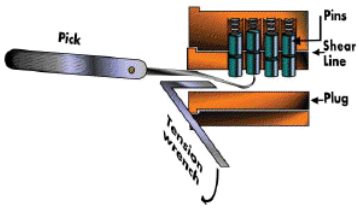


- **Combination** – Zámky, ktoré nevyžadujú kľúč ale číselnú kombináciu pre otvorenie zámkzy. Počet kombinácii = počet číslic x dĺžka kombinácie

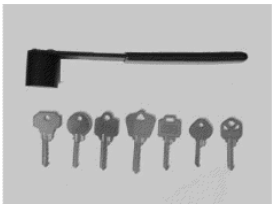


Nástroje pre otváranie zámkov

- **Feelers** – Vždy pracujeme s jedným pinom, kým nenájdeme správnu hladinu. Pin zafixujeme pomalým pootočením zámku.Nájdeme ďalší pin a proces opakujeme.



- **Scrubbers –** Vhodné pre začiatočníkov. Aplikujeme jemný tlak a pracujeme s pinmi od zadu, pričom vykonávame krúživé pohyby a snažíme sa aby zapadli do jednej hladiny.
- **Bumping –** prakticky vsetky Yale a podobné zámky môžu byť otvorené nárazom. Je to spoľahlivý, opakovateľný a jednoducho naučiteľný spôsob otvárania zámkov.



Side channel attack - predtým než sa pokúsime obýsť zámok, hľadáme iné slabé stránky zabezpečenia ako sú napríklad pánty na dverách.

1.2. Autentifikácia (Authentication)

Určenie identity na základe kombinácie

- Osoba niečo vlastní (čipová karta ...)
- Osoba niečo vie (napr. heslo)
- Osoba niečim je (človek s odtlačkami prstvov)

Čiarový kód (Barcode) – strojovo čítateľná reprezentácia dát vzťahujúca sa k objektu na ktorý je pripojený. Prvá generácia reprezentovala údaje ako zvislé čiary atramentu variabilnej šírky, čo je v podstate 1D (jednorozmerná) schéma kódovania. Neskôr sa vyvinuli do obdĺžnikov, bodiek, 6-uholníkov a iných geometrických vzorov v 2 rozmeroch 2D, ktoré môžu byť prečítané špecialnimy optickými skenermi.

Čiarové kódy sú v podstate jednoduché obrázky, ktoré sa dajú ľahko zduplikovať.

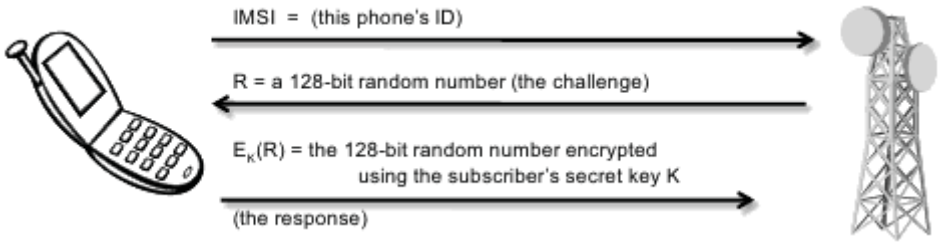
Karty s magnetickým prúžkom (Magnetic Stripe Card) – plastová karta s magnetickým prúžkom obsahuje osobné údaje o držiteľovi karty. Prvá stopa magnetického prúžku obsahuje plné meno držiteľa karty a ďalšie údaje. Druhá stopa môže obsahovať číslo účtu, plastnosť karty, údaje o banke a atď. **Ochrana** – Magnetický prúžok sa dá ľahko čítať a reprodukovať. Magentické pruhové čítačky su pomerne lacné. S použitím zapisovača magnetických prúžkov, ktorý je len o trochu drahší sa dajú jednoducho karty klonovať. Preto je potrebná ďalšia ochrana ako napr. PIN kód.

Čipové karty (Smart Cards) – obsahujú integrovaný obvod, voliteľne so zabudovaným mikroprocesorom, ktorý umožňuje čítatnie a zápis, čím môžeme pristupovať k údajom na karte a meniť ich. Môžu obsahovať rôzne autentifikačné mechanizmy pre ocharnu údajov vlastníka a su ťažko napodobiteľné. Môžu byť použité ako „elektronická peňaženka“

Sim karta (subscriber identity module) – použitie v telefónoch. Kertu vydáva prevádzkovateľ siete. **Ocharana** – Karta obsahuje niekoľko informácii pre identifikáciu majiteľa a overenie v rámci siete. Karta zodpovedá záznamu v DB poskztovateľa. Obsahuje integrovaný obvod karty ID (ICCID), čo je unikátne 18-miestne číslo pre identifikáciu hardvéru. Karta ďalej obsahuje unikátne IMSI (international mobile subscriber identity), ktoré identifikuje vlastníkovú krajinu, sieť, identitu. Ďalej obsahuje 128-bit tajný kľúč, ktorým sa telefón identifikuje v mobilnej sieti. Karta vyžaduje PIN.

GSM Challeng-Response Protocol

1. Keď sa chce mobilné zariadenie pripojiť k mobilnej sieti, najprv sa pripojí k miestnej základňovej stanici vo vlastníctve poskytovateľa a prenáša svoj IMSI
2. Ak IMSI odpovedá záznamom v DB, stanica pošle náhodné 128- bit číslo
3. Toto číslo je dekódované použitím tajného kľúča uloženého na SIM karte a algoritmu nazývaného A3, a pošle to späť stanici.
4. Stanica vzkoná tú istú operáciu a ak je zhoda, tak používateľ je overený na sieti a môže volať a prijímať hovory



RFID (Radio Frequency Identifikation) – rádio frekvenčná identifikácia, informácie sa odovzdávajú pomocou rádiových vĺn. RFID čipy majú integrovaný obvod pre ukladanie informácii a stočenú anténu pre vysielanie a prijímanie rádiového signálu.
Používa sa v spojení so samostatnou čítačkou a zapisovačom. Niektoré RFID vyžadujú batérie, ostatné sú pasívne. Účinný dosah je od niekoľko centimetrov do niekoľko metrov, ale nakoľko sa dáta prenášajú rádiovými vlnami, nie je potrebná veľká vzdialenosť a preferuje sa priama viditeľnosť čítacieho zariadenia. Ďalšie faktory sú prenosová rýchlosť, priepustnosť materiálov

Použitie RFID – kľuče od auta, elektronické mýto, cestovné pasy. Niektoré cestovné pasy majú vstavaný RFID čip, ktorý obsahuje inf. o majiteľovi vrátane digitálnej fotografie tváre. Pre ochranu údajov, je komunikácia šifrovaná pomocou tajného kľúča, ktorý sa však dá ľahko zistiť kedže sa skladá z čísla pasu, dátumu narodenia držiteľa pasu a dátumu expirácie, presne v tomto poradí. Pričom väčšina údajov je priamo na pase.

Biometria – slúži pre jendoznačnú identifikáciu používateľa na základe biologických alebo fyziologickych zvláštnosti. Biometrické systémy pozostávajú z nejakého senzoru alebo skenera, ktorý vie tieto zvláštnosti prečítať a porovnať so záznamom.
Požoadavky biometrie:

- Univerzálnosť – každá osoba musí mať danú charakteristiku
- Rozlíšiteľnosť – každý by mal mať zrejme rozdiely v danej charakteristike
- Trvácnosť – charakteristika by sa nemala výrazne meniť v priebehu času
- Vymožiteľnosť – charakteristika musí byť efektívne stanoviteľná a vycísliteľna

1.3.Priamy útok na výpočtové zariadenia (Direct Attacks on Computational Devices)

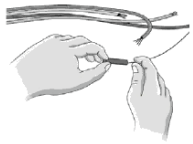
Prírodné útoky (Environmental Attacks)

- Elektrina – výpočtová technika potrebuje pre fungovanie elektrinu, a preto je nevyhnutné, aby takéto zariadenie malo stabilné nepretržité napájanie
- Teplota – počítačové čipy majú nejakú bežnú prevádzkovú teplotu, ale vyššia teplota ich môže poškodiť
- Obmedzená vodivosť – nakoľko sú výpočtové zariadenia elektronické, je potrebná v ich prostredí obmedzená vodivosť.

Eavesdropping– tajné počúvanie súkromného rozhovoru. Ochrana informácii msí byť nad rámec počítačového zariadenia. A bezpečnosť musí byť rozšírená aj na prostredie, kde sa informácie zadávajú a čítajú.
Jenoduché odpočúvacie techniky zahŕňajú

- Použitie techniky sociálneho inžinierstva, ktoré umožňujú čítať informácie ponad plece obete
- Inštalácia malej kamery
- Použitie ďalekohľadu, pre zobrazenie monitora obete

Wiretapping – mnoho komunikačných sieti používa lacné koaxiálne káble, kde sa údaje prenášajú pomocou elektrických impulzov. Existuje lacné zariadenie, ktoré dokáze impulzi merať a zrekonštruovať dáta. Táto technika je pasívna, nakoľko pri takomto odpočúvaní nenastane zmena signálu, a je veľmi ťažké takúto formu odposluchu zistiť.



Signal emination

- Počítačová obrazovka vysiela rádiové frekvencie, ktoré môžu byť použité na detekciu toho, čo je zobrazené
- Viditeľný odraz svetla od stien, okuliarov môže byť použitý pre rekonštrukciu obrazu
- Oba spôsoby vyžadujú aby bol prijímač dosť blízko, aby mohol zachitiť sygnál.

Acoustic emmissions - Dmitri Asonov and Rakesh Agrawal vydali v roku 2004 publikáciu o tom, ako by útočnik mohol zrekonštruovať to čo bolo napísané na klávesnici použitím zvukového záznamu. Nakoľko každá klávesa vydáva rozdielny zvuk, a niektoré klávesy sú využívané častejšie.

Hardware keyloggers – malé konektory inštalované medzi klávesnicu a počítač, napr. USB keylogger

TEMPEST – kódove označenie pre súbor noriem pre obmedzenie informácii vyžarovaných z výpočtovej techniky. Stanovuje 3 úrovne ochrany

- Útočnik má priamy prístup k zariadeniu, ako napr. v priľahlej miestnosti alebo meter od zariadenia v tej istej miestnosti
- Útočnik sa nedostane bližšie ako na 20m k zariadeniu, alebo je blokovaný budovou
- Útočnik sa nedostane bližšie ako na 100m k zariadeniu

Computer Forensics – techniky pre získanie údajov obsiahnutých v počítačových systémoch, HDD, optických diskoch. Tieto techniky sa väčšinou používajú v súvislosti so súdnymy procesmi, ale môžu byť zneužité.

ATM – bankomaty umožňujúce finančné operácie bez ľudskej asistencie. Vloží sa karta, zadá PIN a potom sa vykoná transakcia. Má zabudovaný kriptografický procesor, ktorý porovná zadaný PIN so zašifrovaným PIN na karte (u starších zariadení, ktoré neboli pripojené k sieti) alebo v databáze.

<u>Útoky na ATM</u>

- Libanónska slučka – útočník vloži do slotu ATM púzdro, obeť vloží kartu a mysliac si, že automat má poruchu odíde a utočník si potom vezme púzdro aj s kartou
- Skimmer – zariadenie, ktoré číta magnetický prúžok. Dá sa nainštalovať na slot ATM. Pričom si potom vie vyrobiť duplikát karty
- Fake bankomat – zachytí kredit/debit kartu a PIN v rovnakom čase

Ch02-RFIDSecurity.pdf a Ch02-ComputerForensics.pdf som nespracuvaval, si to len prečitaj, lebo v podtste su tam hovadiny

3 Prednaska

3.1 Buffer Overflow Attack

3.1.1 Exploit?

Exploit je specialny program, data alebo sekvencia prikazov, ktore vyuzivaju programatorsku chybu, ktora sposobi povodne neocakavanu cinnost softveru, hardveru alebo nejakeho elektronickeho zariadenia ktora prinesie nejaky uzitok utocnikovi.

3.1.2 Buffer Overflow Attack (Pretecenie zasobnika)

Najbeznejsia chyba v operacnom systeme je pretecenie zasobnika (buffer overflow).

Vznik pretecenia zasobnika:

- developer nespravi kontrolu ci vstup sa zmesti do zasobnika
- vstup v spustenom procese presahuje dlzku zasobnika
- vstup prepise casr pamate procesu
- sposobi ze sa aplikacia zacne spravat nespravne a neocakavane

Efekt pretecenia zasobnika:

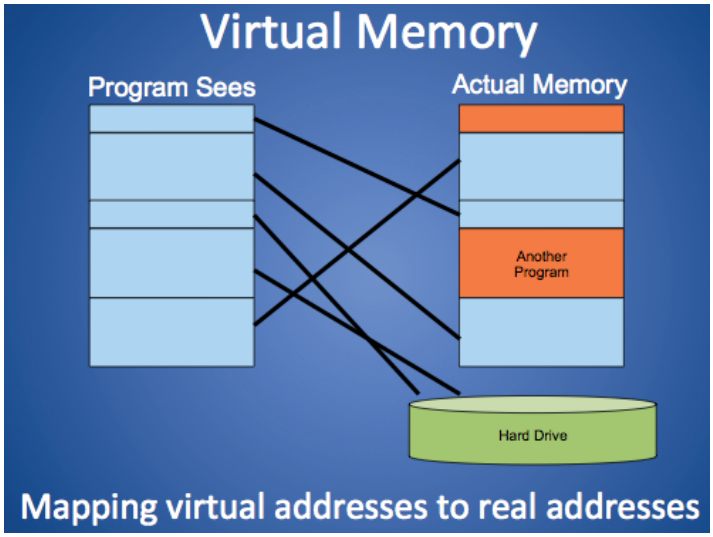
- proces moze pracovat so skodlivymi datami alebo moze spustat skodlivy kod vlozeny do vstupu utocnikom
- ak proces je spusteny ako root, skodlivy kod bude spusteny s root pravami

3.1.3 Adress Space (adresny priestor)

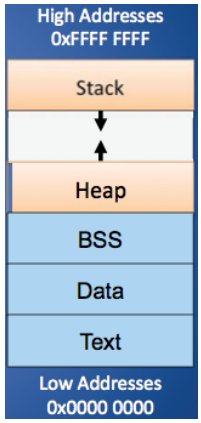
- Kazdy program podrebuje pre svoj bez pristup do pamate
- Kvoli zjednoduseniu, bolo by dobre povolit kazdemu procesu (napriklad kazdemu spustenemu programu) konat tak akoby vlastnil celu pamat
- Adresny priestor sluzi na to aby to bolo dosiahnute
- Kazdy proces moze alokovat priestor v pamati kde potrebuje
- Mnoho jadier spravuje kazdu procesnu alokaciu pamate cez virtualny model pamate
- Pre proces je nepodstatne ako je pamat spravovana

3.1.3.1 Virtualna pamat

- v pocitaci nie je dostatok pamate pre adresny priestor kazdeho spusteneho procesu
- OS vytvara iluziu, ze spusteny proces ma pristup k konpletnemu adresnemu priestoru
- V realite tento pohlad je virtualny a nie je to najlepsia organizacia pamate
- Namiesto toho, pamat je rozdelena na bloky a OS rozhoduje, ktory z nich je v pamati a ktory je ulozeny na disku



3.1.4 Adresny priestor v UNIXe (Organizacia pamate)



- **Text**: strojovy kod programu skompilovany zo zdrojoveho kodu
- **Data**: staticke premenne programu inicializovane v zdrojovom kode pred spustenim
- **BSS (block started by symbol)**: neinicializovane staticke premenne
- **Heap**: dynamicke data vygenerovane pocas behu procesu
- **Stack**: struktura ktora rastie smerom dole a udrzuje aktivovane volania metod, ich argumenty a lokalne premenne

3.1.5 Zranitelnosti a sposob utoku

Zranitelnostne scenare:

- program ma root prava (setuid) a je spusteny zo shellu
- program je castou web aplikacie

Typicky sposob utoku:

- najdenie zranitelnosti
- spatna analyza programu (reverse engineer)
- vytvorenie exploitu

3.1.6 Utok pretecenim zasobnika v Nutshell

- utocnik vyuzije nekontrolovany zasobnik k utoku pretecenim zasobnika
- najvyssi ciel pre utocnika je ziskat schell ktory mu dovoli spustat lubovolne prikazy s vysokymi pravami
- Druhy utokov pretecenia zasobnika: **Heap / Stack smashing**

3.1.7 strcpy() vs. strncpy()

funkcia strcpy()

- kopiruje retazec v druhom argumente do prveho argumentu
- napr. strcpy(dest, scr)
- ak zrojovy retazec je vacsi ako cielovy retazec, presahujuce znaky mozu obsadit pamatove miesto pouzite inymi premennymi
- znak null je automaticky pridany na koniec retazca

funkcia strncpy()

- kopiruje retazec specifikovanim poctu znakov pre skopirovanie
- napr. strncpy(dest, src, n) dest[n] = '\0'
- ak zdrojovy retazec je vacsi ako cielovy retazec, presahujuce znaky su zlikvidovane automaticky
- znak null musime pridat manualne

3.1.8 Return Address Smashing

- Unixove **fingerd()** systemove volanie, ktore je spustane pod rootom, obsahuje zranitelnost pretecenia zasobnika.

- Napise sa skodlivy kod, ktory prepise navratovu adresu tak ze bude smerovat ku skodlivemu kodu
- Ked sa zavola navratova adresa, tak sa spusti skodlivy kod s plnymi pravami roota

3.1.9 Shellcode Injection

Schellcode je:

- kod zostrojeny v processorovych nativnych instrukciach
- je vlozeny ako cast zasobnika ktory pretecie

Utocnik vlozi kod priado do zasobnika ktory posle pre utok. Zasobnik obsahujuci shellcode je vlastne naloz.

3.1.10 Preco pretecenie zasobnika vznika

- Pretecenie zasobnika je cisto problem jazyka C.
- Zabranit vieme tomu tak, ze zasobnik budeme alokovat dynamicky, budeme pouzivate bezpecne funkcie, nas program nebude bezat pod root pravami

3.2 Ochrana suboroveho systemu (Filesystem Security)

Principy:

- subory a priecinky su spravovcane operacnym systemom
- aplikacie pristupuju k suborovemu systemu cez API
- **Access Control Entry (ACE):** povolenie/zakazanie kontretneho typu pristupu k suboru/zlozke prostrednictvom pouzivatela/skupiny
- **Access Control List (ACL):** zoznam **ACEs** pre subor/priecinok
- **File handle:** identifikator suboru/priecinku
- **Operacie so subormi:** otvorenie suboru (vrati file handle), citanie zo suboru, zapisovanie do suboru spustanie suboru, zatvaranie suboru
- **Hierarchycka suborova organizacia:** Strom (Tree) – Windows, DAG – Linux

3.2.1 Discretionary Access Control (DAC) – Diskretna pristupova kontrola

- pouzivatel moze chranit co vlastni
    - vlastnik moze udelovat pristup ostatnym
    - vlastnik moze definovat typ pristupu (citanie, zapisovanie, spustanie) ostatnym
- je standartny model pouzivany v operacnych systemoch
- Mandatory Access Control (MAC)
    - alternativny model
    - viacere levely pre bezpecnost pouzivatelov a dokumentov

3.2.1.1 Closed vs. Open Policy

Closed Policy: defaultne pouzivatelia nemaju ziadne prava a prava im udelujeme
Open Policy: defaultne pouzivatelia maju vsetky prava a prava im odoberame

3.2.1.2 Closed Policy so zapornou autorizaciou a odopieracou politikou

- pouziva sa vo Windowse
- napr. udelime prava na citanie a zapis a potom tomu istemu pouzivatelovi odoberieme prava na zapis

3.2.2 Access Control Entries (ACEs) and List (ACL)

- **ACL:** pre zdroj (napr. subor alebo priecinok) je usporiadany zoznam 0 alebo viac **ACEs**
    - **ACL prikazy:**
        - **getfacl:** precitanie ACLs
        - **setfacl:** nastavenie ACLs
- **ACE**: - urcuje ktory konkretny zoznam pristupov (citanie, spustanie, zapisovanie) k zdrojom je povoleny alebo zakazany pre pouzivatela alebo skupinu

- **Napriklad**:
  - Bob; Read;Allow
  - tAs; Read; Allow
  - TWD; Read, Write; Allow
  - Bob; Write; Deny

3.2.3 Linuxovy suborovy system

- strom zloziek
- kazda zlozka ma odkazy na 0 alebo viac suborov alebo zloziek
- **Hard Link – tvrda linka**
  - Zo zlozky na subor
  - rovnaky subor moze mat hard linky z viacerych zloziek, kazka z nich ma svoje vlastne meno ale vsetky zdielaju vlastnika, skupinu a prava
  - Subor je vymazany ked na neho nie su ziadne hard linky
- **Symbolic link (symlink) – Symbolicka linka**
  - zo zlozky na cielovy subor alebo zlozku
  - uklada cestu k cielu, ktora je maprec vsetkymi pristupmi
  - rovnaky subor alebo zlozka mozu mat viacero symlinks
  - vymazanie symlinky neovplivni cielovy subor/zlozku
  - vymazanie clela zneplatni ale nevymaze symlink
  - analogia k windows shorcut alebo k Mac OS aliac

3.2.4 Unix permissions

- standartne pre vsetky UNIXy
- kazdy subor je vlastneny pouzivatelom a ma priradenu skupinu
- prava su casto zobrazene v kompaktnej 10-znakovej notacii
- ak chceme zobrazit prava v terminali, je potrebne zadat prikaz ls –l
- r – citanie, w – zapis, x - spustanie
- Priklad:
  - subory: -rw-r--r-- (citanie/zapisovanie pre vlastnika, citanie pre vsetkych ostatnych)
  - priecinky: drwxr-xr-x (vsetci mozu vidiet zoznam suborov v zlozke ale len vlastnik vie pridavat/vymazavat subory v tejto zlozke)

3.2.4.1 Specialne bity pre prava

- **Set-user-ID („suid“ alebo „setuid“) bit**
  - na spustitelnych suboroch sposobuju ze sa program spusta pod dannym vlastnikok bezohladu na to kto ho spustil
  - ignorovany pre vsetko ostatne
  - nahradzuje 4 znak (**x**) za (**s**) alebo (**S**) - ak nie je spustitelny
    - -rwsr-xr-x: setuid, spusttelny vsetkymi
    - -rwxr-xr-x: spustitelny vsetkymi ale nie je setuid
    - -rwSr--r--: setuid ale nie je spustitelny (neuzitocne)
- **Set-groupID („sgid“ alebo „setgid“) bit**
  - na spustitelnych suboroch sposobuje ze program sa spusta s pravami skupiny, bez ohladu na to aky pouzivatel ho spustil
  - na priecinkoch sposobuje to ze sa vytvori subor v ramci tejto zlozky tak ma rovnaku skupinu ako zlozka
  - ignorovany pre vsetko ostane
  - nahradzuje 7 znak (**x**) za (**s**) alebo (**S**) - ak nie je spustitelny
    - -rwxr-sr-x: setgid subor spustitelny vsetkymi
    - drwxrwsr-x: setgit precinok, subory budu mat skupinu ktora je na zlozke
    - -rw-r-Sr--: setgit subor ale nie spustitelny
- **Sticky bit**

- na priecinkoch sposobuje ze pouzivatelia z neho nemozu vymazat alebo zmenit meno na suboroch ktore nevlastnia
- ignorovany pre vsetko ostatne
- nahradzuje 10 znak (**x**) za (**t**) alebo (**T**) - ak nie je spustitelny
  - drwxrwxrwt: sticky bit nastaveny, plny pristup pre vsetkych
  - drwxrwx--T: sticky bit nastaveny, plny pristup pre pouzivatela/skupinu
  - drwxr--r-T: sticky bit nastaveny, vlastnik ma plny pristup, ostatni moze len citat

3.2.5 NTFS prava

- **zakladne NTFS prava**
  - citanie – otvaranie priecinkov/podpriecinkov/suborov
  - zoznam obsahu proecinka
  - citanie a spustanie – otvaranie, spustanie programov
  - zapisovanie – vytvaranie podpriecinkov a pridavanie suborov do priecinka/upravovanie suborov
  - upravovanie – vsetko nahore + vymazavanie
  - plna kontrola – vsetko nahore + zmena prav a preberanie vlastnictva
- **viacnasobne NTFS prava**
  - prava su suhrne
  - prava na subore su nad pravami ktore su na priecinku
  - zakaz je nad povolenim
- Explicitne: je nastaveny vlastnik pre kazdeho pouzivala/skupinu
- Dedene: dynamicky dedene z explicitnych prav
- Ucinne: nadobudnute kombinaciou explicitnych a dedenych prav

3.3 Koncepty operacnych systemov

3.3.1 Pocitacovy model

- CPU
- Random access memory (RAM)
- Vstupne/vystupne zariadenia
- Ulozny priestor (disk drive)

3.3.2 Koncepty OS

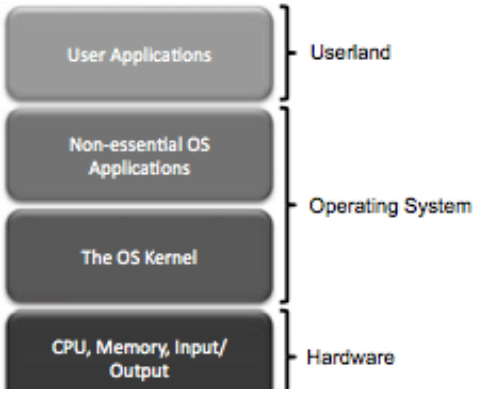
- Operacny system poskytuje rozhranie medzi pouzivatelmi pocitaca a pocitacovym hardverom.
- Operacny sytem spravuje sposob ako aplikacie pristupuju k pocitacovym zdrojom ako ulozny priestor, CPU, hlavna pamat, vstupne/vystupne zariadenia a sietove rozhrania
- Operacny system spravuje viacero pouzivatelov
- Operacny system spravuje viacero programov

3.3.3 Multitasking

- Dava kazdemu programu podiel CPU casu.
- CPU pracuje tak rychlo ze pre pouzivatela to vyzera tak, akoby vsetky programy v pocitaci bezali suvisle

3.3.4 Kernel

- jadro operacneho systemu
- spravuje nizkourovnove hardverove zdroje ako pamat, procesor, vstupne/vystupne zariadenia
- vela operacnych systemov definuje ulohy spojene s jadrom na zaklade vrstvy s hardverovymi komponentami (CPU, pamat, I/O zariadenia)



zacinajuce na dne a pouzivatelov a aplikacie zacinajuce na vrchole

3.3.5 Input/Output

- Vstupne/vystupne zariadenia pocitaca obsahuju veci ako klavesnica, mys, monitor, sietovy kartu a mnoho dalsich periferii,
- Kazde zo zariadeni je reprezentovane v operacnom systeme s pouzitim ovladaca zariadenia ktore zapuzdruje detaily ako pracovat s tymto zariadenim
- Kazde zariadenie poskytuje API (aplikacne rozhranie), ktore poskytuje danne zariadenie programom, ktory moze s dannym zariadenim pracovat

3.3.6 Systemove volania

- Aplikacie nepracuju priamo s nizkourovnovymi hardverovymi komponentami ale namiesto toho zadeluju jadru ulohy prostrednictvom systemovych volani
- Systemove volania su zvycajne obsiahnute v zozname programov, ktore su kniznicami napriklad jazyka C a tieto programy poskytuju rozhranie, ktore dovoluje aplikaciam pouzivat preddefinovane serie APIciek, ktore definuju funkcie pre komunikovanie s jadrom.

3.3.7 Procesy

- proces je instanciou programu ktore je aktualne spusteny
- Aktualny obsah programu je ulozenyna disku
- Pri spusteni programu musi byt program nahraty do RAM a musi byt identifikovany ako unikatny proces
- Viacere instancie toho isteho programu mozu byt spustene ako rozne procesy.

3.3.8 Identifikatory procesu

- kazdy proces v pocitaci je identifikovany unikatnym kladnym integerom nazyvanym identifikator procesu (PID)
- kazdemu PIDu procesu mozeme priradit jeho vytazenie CPU, vyuzitie pamate, pouzivatelov ID, meno programu atd.

3.3.9 Page Faults

- proces si vyziada virtualnu adresu ktora nie je v pamati a vznikne Page Fault
- Blokovy dozorca vyhodi z RAMky stary blok
- Blokovy dozorca vyhlada na disku pozadovany blok a alokuje ho do pamati

3.3.10 Virtualne stroje

- je to Operacny system ktory reprezentuje proces spusteny na specifickej architekture pod nejakym operacnym systemom
- Vyhody:
    - Portatibilita
    - bezpecnost
    - sprava
    - hardverova efektivita

3.4 Bezpecnost v operacnych systemoch

3.4.1 Sekvencia zavadzania (boot sequence)

- akcia nacitania OS do pamate z vypnuteho stavu je znama ako booting (zavadzanie) alebo bootstrapping
- ked sa pocitac spusti, najprv sa spusti kod ulozeni vo firmweri znameho ako BIOS
- V modernych OS BIOS nacita do pamate druhostavovy boot loader, ktory sa postara o nacitanie celeho OS do pamate a potom odovzda kontrolu spustania operacnemu systemu

3.4.2 Hesla v BIOSe

- zabranuju utocnikovi spustit operacny system bez autentifikacie

3.4.3 Hibernacia

- moderne operacne systemy mozu byt hibernovane namiesto vypnutia
- ked je pocitac hibernovany, OS ulozi obsah pamate do suboru, ktore neskor pri spusteni pocitaca moze byt rychlo nacitany
- ale bez dodatocnej bezpecnosti hibernacia nemusi byt bezpecna, nakolko utocnik sa moze dostat k suboru hibernacie a vytiahnut z neho napr. nesifrovane hesla

3.4.4 Event logging

- logovanie toho co proces spusta, ako ine stroje komunikuju so systemom cez internet a ak operacny system sa zacne spravat neocakavane, mozme zistit co sa s nim deje prostredictvom logov a pripadne mozme odhality bezpecnostne narusenie.

3.4.5 Bezpecnost pamate a suboroveho systemu

- obsah pocitaca je zabaleny v pamati a v suborovom systeme
- zabezpecenie pocitacoveho obsahu musi zacat so zabezpeceni jeho pamate a suboroveho systemu

1. Malware: Škodlivý softvér

Malware – môžeme zaradiť do niekoľkých kategórii v závislosti na šírení a maskovaní:

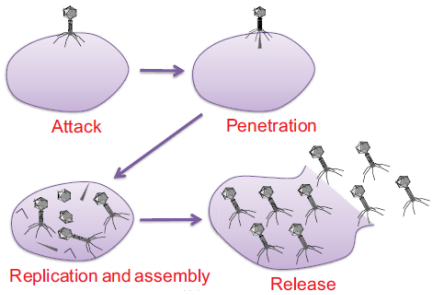
- Šírenie
  - Vírus – za šírenie je zodpovedný človek (napr. email príloha)
  - Worm (červ) – šírenie bez ľudskej asistencie
- Maskovanie
  - Rootkit – modifikuje operačný systém pre utajenie svojej existencie
  - Trojan – vykonáva požadované operácie, ale utajuje nebezpečnú prevádzku

Vnútorný útok (Insider attack) – jedná sa o porušenie bezpečnosti niekim z organizácie, ktorá sa stará o zabezpečenie produktu. V prípade Malware sa jedná o bezpečnostnú dieru, ktorá bola v softvéry vytvorená jedným z developerov.

Zadné dvierka (Backdoor) – jedná sa o skrytú funkcionalitu alebo príkaz v programe, ktorá používateľovi umožní vykonávať akcie, ku ktorým by ináč nemal prístup. Ak sa program používa normálne, vykonáva všetko podľa očakávania, ale keď sa spustí skrytá funkcionalita, tak program vykonáva niečo, čo je v rozpore s bezpečnostnou politikou (napr. obídeme overovanie, získame nelegálny vzdialený prístup atď.).

Logická bomba (Logic bomb) – kus kódu zámerne vložený do systému, ktorý sa aktivuje po splnení nejakých podmienok. Napr. programátor môže ukryť kus kódu, ktorý bude mazať súbory (napr. triger v databáze pri platbách). V podtate aj samotné výrusy a červi obsahuju logické bomby, ktoré sa aktivujú pri splnení nejakej podmienky.

- **Počítačový výrus (Computer virus)** – jedná sa o počítačový kód, ktorý sa dokáže replikovať tým, že modifikuje ostatné programy, súbory – vloží do nich kód, ktorý sa vie ďalej replikovať. Tento výrus je zvláštny tým, že na to aby sa mohol replikovať potrebuje asistenciu zo strany používateľa (napr. kliknutie na prílohu v emaile, zdielanie USB)



Fázy počítačového výrusu

- Spánok – vírus jednoducho existuje, vyhýba sa detekcii
- Šírenie – replikuje sa, napáda nové systémy
- Spúšťanie – nejaký podnet aktivuje vírus, ktorý začína vykonávať svoju akciu
- Akcia – vykonáva škodlivé akcie, ku ktorým bol vytvorený

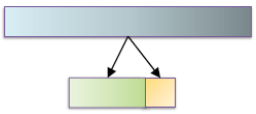
Typy infekcií

- Prepísanie (Overwriting) – zničí originálny kód

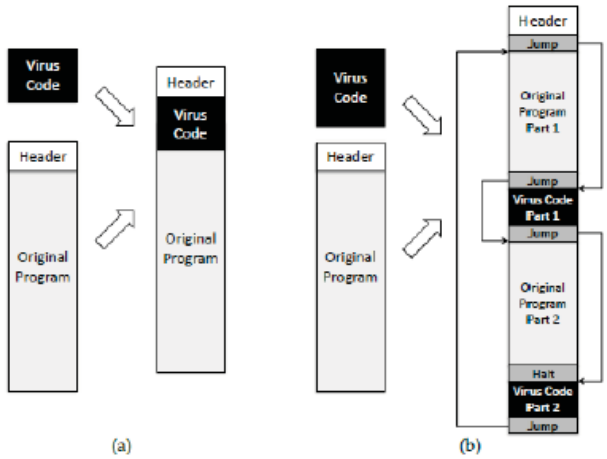
- Pre-pending – ponechá pôvodný kód, ak je to možné, tak skomprimovaný

- Infikovanie knižníc – umožňuje výrusu byť pamäťovo odolným (napr kernel32.dll)

- Makro vírus – infikujú MS Office dokumenty, najčastejšie inštalované ako templaty



Úrovne kompilácie vírusov – vírusy majú rozne stupne kompilácie v tom, ako sa vkladajú do počítačového kódu



- **Počítačový červ (Computer worm)** – šíri svoje kópie bez toho, aby musel infikovať iné súbory a bez zásahu človeka.Technicky sa nejedná o počítačový výrus (neinfikuje iné programy), no napriek tomu si ľudia tieto pojmy zamieňajú. Na svoje šírenie často využíva počítačovú sieť a spolieha sa na bezpečnostné chyby na cieľovom počítači. Pred samotným infikovaním musí zistiť či už daný systém nie je infikovaný. Vo väčšine prípadov je červ zodpovedný za mazanie súborov a inštalovanie zadných dvier.

- **Trójsky kôň (Trojan horse)** – škodlivá aplikácia, ktorá sa navonok tvári ako užitočný súbor, program ale skutočnou úlohou je napr. poskytnúť neoprávnený prístup hackerovi k systému. Môžu kradnú informácie, alebo poškodiť systém

- **Rootkit** – je navrhnutý tak, aby utajil svoju existenciu pred bežnými metódami detekcie a umožňuje privilegovaný prístup k počítaču. Detekovať ho je niekedy ťažké, keďže dokáže rozvrátiť aj softvér, ktorý ho má vyhľadať. Jeho odstránenie je tiež náročné, niekedy nemôžné, hlavne keď sa usídli v kerneli, tu pomôže už len preinštalovanie systému.

- **Adware** – softvér, ktorý automaticky zobrazí reklamu s cieľom vytvárať zisk pre autora. Funkcie môžu byť navrnuté tak, že skúmajú stránky, ktoré používateľ navštevuje a na základe toho mu poskytne nejakú raklamu.

- **Spyware** - program nainštalovaný na počítači, ktorý zhromažďuje informácie o používateľovi bez jeho vedomia. Je ťažké ich nájsť. Niekedy môžu byť inštalované zámerne napr. vo firmách na sledovanie aktivít zamestnancov.

Antivírusy

Podpis - kaďy vírus sa dá identifikovať na základe svojho odtlačku. Odtlačok je reprezentovaný ako postupnosť reťazcov. Vyhľadávanie je na základe Pattern Matching. Každý odtlačok sa ukladá v malware databáze.

Heuristic Analysis – metóda využívaná mnohými antivírusovými programami pre odhaľovanie nových a poymenených výrusov. Analýza kódu je založená na inštrukciách, o ktorých antivírus rozhoduje či sú škodlivé (napr. mazanie súborov).

Shield vs On-demand

- Shield – beží na pozadí. Kontroluje súbor zakaždým keď je pozmenený (otvorený, skopírovaný, premiestnený, vykonávaný...)
- On-demand – skenovanie na vyžiadanie používateľa.

Online vs Offline Anti Virus Softvare

- Online – pluginy do browserov, certifikáty, skanovanie vyžaduje internetová pripojenie.
- Offline – inštalované na OS, ľahko konfigurovateľné, skenovanie bez internetového pripojenia.

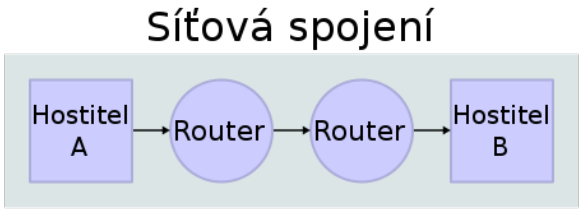
Karanténa – súbor môže byť izolovaný v priečinku s názvom karanténa. Súbor nie je zmazaný, ale len zneškodnený, teda používateľ sa môže rozhodnúť, či ho zmaže ale znovu obnoví ako falošný poplach. Preacovať so súborov v karanténe sa dá iba pomocou antivírusového programu.

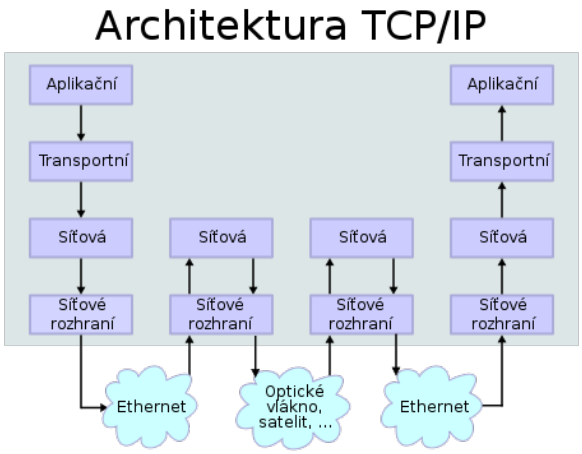
Static vs Dynamic Analysis

- Static – kontrola kódu bez jeho spúšťania. Kontroluje sa byte kód na nezrovnalosti
- Dynamic – vykonanie kódu vo virtual sandbox. Kontrola zemny súborov, zmeny registrov, procesy a vlákna, sieťové porty

Internetovy Protokol

- zakladny protokol pracujuci na sietovej vrstve
- je zodpovedny za smerovanie packetov od zdroja k cielu
- sklada sa z riadiacich (metadat) a uzivatelskych udajov
- udaje sa posielaju po blokoch nezavislymi cestami
- poskytuje nespolahlivu sluzbu - tzv. best effort - vsetky prvky sa snazia dorucit packety najkratsou moznou cestou ale nezarucuju ze budu dorucene – na to treba implementovat dalsie vrstvy architektury
- navrh predpoklada neodmyslitelne nespolahlivu infrastrukturu
- riesenim je vypocet kontrolneho suctu pri prechode kazdym uzlom
- v maji 1981 bol implementovany IPv4 – 32bitove adresy, celkom 4 mld. adries
- vdaka NAT prekladu adries bolo mozne oddialit vycerpanie adries dodnes (bez NAT uz v 90.rokoch)
- nova verzia IPv6 – 128 bitove adresy, celkom 4.3e9 adries
- Connection-oriented – TCP – data su odosielane organizovane s potvrdzovanim o prijati (vacssina trafficu)
- connectionless – UDP – data su odosielane ako je to len mozne bez potvrdzovania o prijati (vacssinou len streamovacie sluzby – TV, video, radio, Apple keynotes :-D)
- packet sa sklada z obsahu, TCP hlavicky, IP hlavicky a hlavicky spojovej vrstvy





-
- TCP architektura obsahuje narozdiel od OSI modelu len 4 vrstvy:
  - aplikacna
  - transportna
  - sietova
  - vrstva sietoveho rozhrania (ekvivalentne fyzicka alebo spojova)

IP adresy

- sklada sa z adresy siete, adresy podsiete, adresy pocitaca
- urcovanie poctu pocitacov v sieti zavisi od druhu pouzitia masky podsiete
- existuju aj rezervovane adresy ako napr 127.0.0.1 (localhost) alebo xxx.xxx.xxx.254, kde xxx je adresa siete v pripade ze maska je 255.255.255.0
- jednotlive siete sa delia do tried:
    - A – 1.bajt v rozmedzi 0-127, maska je 255.0.0.0
    - B – 1.bajt v rozmedzi 128-191, maska je 255.255.0.0
    - C – 1.bajt v rozmedzi 192-223, maska je 255.255.255.0
    - D – 1.bajt v rozmedzi 224-239, maska je multicast
    - A – 1.bajt v rozmedzi 240-255, je vyuzite ako rezerva

IP routovanie

- router je zariadenie, ktore spojuje viacere siete dokopy
- pouziva sa na to tzv. NAT (network address translatiton),
- sluzi na prekladanie IP adries, t.j. kazdy packet s danou hlavickou zdrojovej adresy ziska novu adresu
- pre adresovanie packetu do siete za NAT pre konkretny pocitac, ktory je navonok neviditelny je vhodne pouzit uzivatelsky port (ktory nie je rezervovany pre nejaku konkretnu sluzbu) a tento port nasledne presmerovany na vnutornu IP adresu za NAT)
- na testovanie routingu sa pouziva ICMP protokol a obsahuje 2 najcastejsie sluzby
    - ping – sluzi na ziskanie odozvy od pocitaca, odosielatel odosle ping packet (32 bytov) adresatovi a ten ich vrati naspat, ziska sa prehlad o tom ci adresat existuje a ci je online. Ping packet ma istu zivotnost pocas ktorej sa musi stihnut vratit
    - traceroute – sluzi na ziskanie celkovej trasy od odosielatela k prijemcovi. tento packet ma tiez svoju zivotnost avsak kazdym prechodom uzla sa zvysuje jeho zivotnost
    - Ping of death – specialne modifikovana verzia ICMP packetu
        - klasicky ping avsak vacssi
        - sposobuje pad systemu vdaka buffer overflow

MAC adresa

- Media Access Control – jedinecny identifikator sietoveho rozhrania, ktory pouzivaju rozne protokoly spojovej vrstvy OSI modelu.
- je to 48bitove cislo, ktore sa vo vacssine reprezentuje ako 6 dvojic Hexadecimalnych cisel
- existuje vsesmerova adresa (ff:ff:ff:ff:ff:ff)

Zabezpecenie pouzitim MAC adresy

- najcastejsie pouzitie MAC adresy pre bezpecnost je pouzitie filtrovania pristupu, whitelist alebo blacklist IP adries, ktore maju povolenie sa pripojit
- vacssinou sa pouziva ako doplnkova ochrana v pripade bezdrotovych sieti (v kombinaciis WPA2 napr)
- s tym suvisi utok MAC spoofing, ktoreho principom je odpocuvat komunikaciu (bezdrotovu, pri kabli sa to neda) a prevziat existujucu MAC adresu, za ktoru sa vydava utocnik

- tento typ utoku sa tazko odhaluje pri bezdrotovej komunikacii a prakticky jedina moznost je detekovat pri kablovej komunikacii pritomnost rovnakych IP adries na roznych portoch routera
- zmena MAC adresy
    - v Unix based systemoch zistime existujucu pomocou ifconfig
    - vo Windowse pomocou ipconfig /all (bez parametra /all len IP adresa)
    - samotna zmena
        - Unix: ifconfig eth0 hw ether <mac adresa> (kde eth0 je ID sietovky)
        - Windows: neexistuje prikaz, vykonava sa cez GUI, nepodporuje vsak vsetky sietove karty
        - v oboch pripadoch pozaduje admin prava

ARP protokol

- vyuziva sa na ziskavanie MAC adresy pocitaca z jeho IP adresy
- zjednodusene – odosielatel sa pyta: „kto ma tuto IP, nech mi posle svoju MAC“
- dovodom je poznat fyzicku (MAC) adresu pocitaca v rovnakej LANke
- pri posielani cez Ethernet je nutne poznat prave MAC adresu
- taketo zistovanie adries sa zapisuje do ARP cache a urychluje komunikaciu v lokalnej podsieti
- alternativou ARP protokolu je pouzivanie tabulky MAC adries sluziacej k priradeniu IP adries pri pripajani jednotlivych PC k routeru (tato tabulka je ulozena iba v routeri, zabezpecuje, ze IP adresy sa nemenia s roznym poradim prihlasovania pocitacov do siete)
- ARP pouziva iba IPv4, novsie IPv6 pouziva NDP (Neighbor discovery protocol)
- s tym suvisi RARP (reverse ARP)
    - sluzi na ziskavanie IP na zaklade MAC adresy
    - je zakladom pre DHCP server
- ARP spoofing
    - vydavanie sa za iny pocitac
    - utocnik neustale preposiela svoju MAC adresu na vsetky requesty IP adries
    - vyuziva sa hlavne pri ziskavani komunikacie pocitacov napojenych na switche, kedze switch si robi vlastnu ARP cache
    - ochranou je pouzitie statickej ARP tabulky (vopred danej)

Telnet

- oznacenie protokolu k pripojeniu na vzdialeny pocitac
- pracuje na aplikacnej vrstve pouzivajuc TCP/IP
- standardne pouziva port 23
- pouzival sa ako emulacia terminalu k umozneniu prace s prikazovym riadkom
- nevyhodou je vsak absencia sifrovania (vratane hesiel) preto ho dnes nahradilo SSH

SSH

- secure shell
- nastupca telnetu, bezi na aplikacnej vrstve pouziva TCP/IP
- poskytuje volitelne bezstratovu kompresiu

- v 1996 bola vydana verzia SSH-2, ktora implementovala Diffie-Hellman algoritmus pre vymenu klucov
- dalsou vyhodou je moznost riadit lubovolny pocet shellov pomocou jedneho SSH spojenia
- verzia SSH-2 vsak bola pouzita ako proprietarna technologia s uzavretou licenciou
- v 1999 vzniklo OpenSSH (vid iPhone) a prvykrat sa objavilo v OpenBSD 2.6
- v 2006 bol navrhnuty SSH-2 ako internetovy standard
- poskytuje bezpecny prenos suborov pomocou SFTP alebo SCP
- je nutne pri vsetkych verziach SSH aby neznamy verejny kluc bol riadne overeny pred ostrou komunikaciou k zamedzeniu Man-in-the-middle utoku
- overovanie ucastnikov v SSH
    - vyuziva asymetricke sifrovanie
    - klient vygeneruje dvojicu klucov (verejny, sukromny, klasika)
    - verejny ulozi na server
    - server vykona challenge-respons, t.j. posle klientovi nahodne data zasifrovane klientovym verejnym klucom
    - ulohou klienta je desifrovat pomocou svojho privatneho kluca a odoslat nesifrovane naspat na server
    - server ma istotu ze klient je ten za koho sa povazuje

Denial of Service Attack

- tzv. DOS utok
- technika utoku, pri ktorej dochadza k zahlteniu servera neustalymi poziadavkami, ktorych traffic je vacssi ako mozny traffic zvladnutelny serverom.
- ciel utoku je bud vynutit reset cieloveho pocitaca alebo narusenie komunikacie servera s obetou (spomalenie ci uplne znemoznenie)
- najcastejsie je vykonany posielanim nahodnymi datami, ktore zabranuju pretekaniu skutocnych dat
- moze to byt extremnym zatazenim CPU
- Smurf attack – vyuziva chybnu konfiguraciu systemu, ktora dovoli rozoslaniu packetov vsetkym pocitacom v sieti pomocou Broadcast adresy.
- Ping flood – zahltenie cieloveho PC pomocou ping ziadosti. predpokladom je, ze utocnik ma rychlejsiu konektivitu ako obet (note: takto sme DoSovali 99% pred volbami)
- SYN flood – utocnik posle zaplavu TCP packetov s falosnou hlavickou. kazdy packe je serverom prijaty ako normalny a odosle TCP/SYN-ACK potvrdenie a caka na TCP/ACK potvrdenie – to vsak nikdy nedorazi, a tym je na isty cas blokovana komunikacia (efektivnejsie ako Ping flood)
- Obrana
    - SYN flood – vnutorne sa modifikuje chovanie TCP protokolu tak, ze vstup k zdrojom servera sa spristupni az po overeni platnosti adresy
    - Firewall – filtruje IP adresy alebo protokoly pouzivane k utokom, z principu fungovanie nie je mozne odfiltrovat celu utocnu komunikaciu pretoze by nebolo mozne pouzit tu realnu
    - Intrusion Prevention System – ucinny iba ak utok ma zhodny podpis (napr. cast hlavicky) ako uz skor vykonany utok, tym sa moze filtrovat od normalneho provozu v sieti

1. Network Security 2

DNS (Domain Name System) – jedná sa o protokol aplikačnej vrstvy, ktorý slúži ako"telefónna kniha" pre internet, t.j. prekladá hostname do IP adries. DNS zabezpečuje distribuovanú databázu nad internetom, ktorá ukladá rôzne záznamy vrátane:

- Address (A) – IP adresy spojené s názvom hostiteľského zariadenia
- Mail Exchange (MX) – mail server v doméne
- Name server (NS) – autoritatívny server pre doménu

Menný server (Name server)

- Doménové mená
  - 2 alebo viac častí oddelených bodkou (napr. cs166.net)
  - Úplne vprvo je top-level domain (TLD)
- Hierarchia autoritatívnych menných serverov
  - Informácia o koreňovej doméne
  - Informácie o jej subdoméne (A záznam) alebo odkaz na iné menné servry (NS záznamy)
- Koreňové servery poukazujú na DNS servry pre TLD
- Koreňové servry, a servy pre TLD sa niekedy menia
- DNS servry sa odkazujú na ine DNS servry cez meno, nie cez IP

Namespace Management

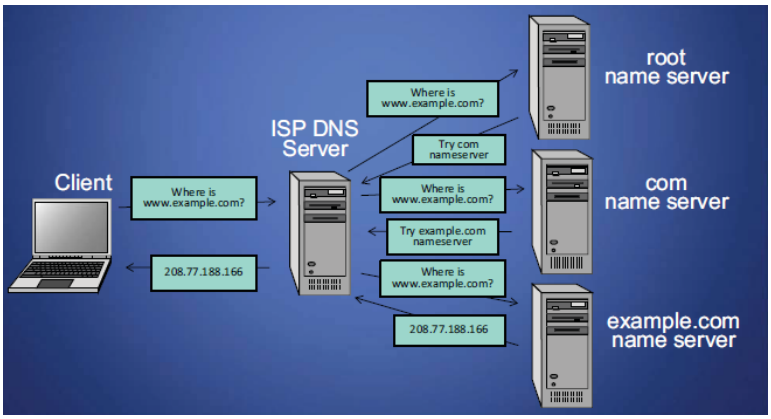
ICANN (Internet Corporation for Assigned Names and Numbers) – spoločnosť založená v roku 1998, ktorá je zodpovedná za riadenie celosvetového internetu čo sa týka jedinečných identifikátorov a zabezpečenie jeho stabilného a bezpečného fungovania. Kontroluje root domény, TLD (každá krajina má vlastné TLD kontrolované vládou).

TLD (Top-level domain) – ide o doménu na najvyššej úrovni DNS. TLD sa nachádza v koreňovej zóne menného priestoru. Pre všetky domény na nižších úrovniach je reprezentovaná ako posledná časť názvu domény (napr. www.example.com, kde TLD je .com, resp. .COM, lebo nie je Case Sensitive). Za riadenie väčšiny TLD je zodpovedná ICANN.

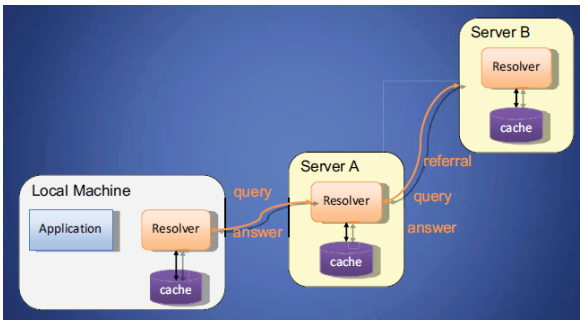
- TLD sa začalo používať v roku 1984. Pôvodne mal byť názov odvodený iba od funkcie (ciže .com pre komerčné webové stránky a .mil pre armádu). V roku 1994 sapovolilo TLD pre krajiny ako .it .us. V roku 2000 bol pokus k návratu vytvárania mien podľa účelu (.aero, .museum)

Name Resolution

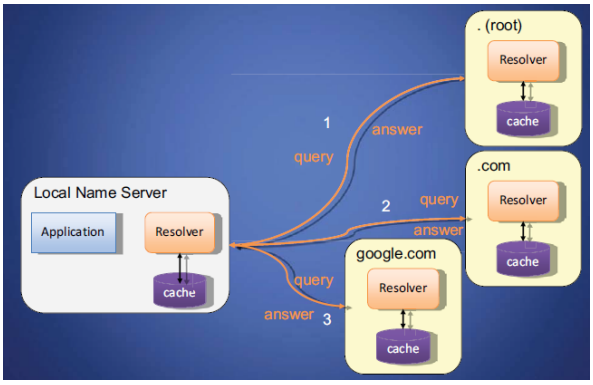
- Odpoveď sa nekešuje



- Recursive name resolution



- Iterative name resolution



DNS caching – DNS si odpamätáva výsledky za určitú dobu (time-to-live => TTL). Operačné systémy a prehliadače udržiavajú DNS cash.

1) Dotaz na zourdomain.com
2) Získanie odpovede a cache na localnom NS a hoste
3) Použi radšej cash než sa opäť dotazovať
4) Vylúčenie položky z cache po uplinutí TTL

Pharming: DNS Hijacking – presmerovanie na falošnú stránku za účelom získania používateľských prihlasovacích údajov. Môže byť vykonaný zmenou host súboru na počítači obete, alebo zneužitím chyby zabezpečenia na DNS servri.

DNS cache poisoning – poskytnutie falošného záznamu DNS servru, ktorý sa zakešuje. Tento tzp útoku môže byť uskutočnení, keď DNS server nerešpektuje alebo má predvídateľné identifikátory, alebo akceptuje nevižiadané DNS záznamy.

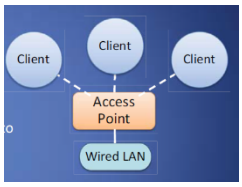
Prevencia – používať náhodné identifikátory pre dotazy a vždy ich skontrolovať, nasadenie DNSSEC.

2. Wireles Networks (Bezdrôtové siete)

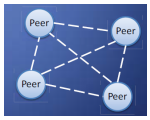
Bezdrôtová komunikácia – komunikácia medzi dvoma, alebo viacerými zariadeniami, ktoré nie su navzájom prepojené. Najčastejšie sa používa elektromagnetická bezdrôtová telekomunikácia (rádio, od malého po veľký dosah).

Typy bedrôtových sieti:

- Infraštruktúra – Klientský stroj vytvorí rádiové pripojenie na špeciálne sieťové zariadenie nazývane acces point. Toto zariadenie je pripojené ku káblovej sieti, ktorá predstavuje bránu k internetu



- Per-to-peer – viacero strojov navzájom prepojených. Typicky sa využíva v ad-hoc sieťach a pri zdieľai dát cez inetrnet.



SSID (Service Set ID) – identifikátor o veľkosti 32-znakov, ktorý slúži pre identifikovanie siete. Defaultné SSID je reprezentovane názvom výrobcu. SSID je často vysielané, aby potencionálny klienti mohli danú sieť objaviť.

Wardriving – jazda vozidlom za účelom hľadania Wi-Fi s použitím antém pre zvýšenie dosahu.

Warchalking – kreslenie symbolov na verejných mietach za účelom informovania o free Wi-Fi sieťach.

WEP (Wired Equvalent Privacy) – cieľom bolo zabezpečiť dôvernosť (zabránenie odpočúvaniu), integritu dát (nemožno manipulovať s paketmy), riadenie prístupu (len správne šifrované pakety sú smerované). Prístupový bod a klient zdieľajú 40-bit kľúč, ktorý sa počas WEP session nemení. Authetifikácia zo strany klienta prebieha tak, že prístupový bod pošle klientovi náhodnú nešifrovanú výzvu, a klient ju spätne odošle v šifrovanej podobe.

IP redirect attack – útok spočíva v tom, že útočník presvedčí prístupový bod, aby dešifroval packet. Postup: odchytíme IP adresu pricházajúceho paketu. Prepošleme paket na externe zariadenie kontrolované útočníkom. Príjmeme dešifrovaný paket od prístupového bodu.

Opakujeme s odchádzajúcimi paketmi. Snažime sa nájsť cieľovú adresu v rámci podsiete LAN. Zmeníme cieľovú adresu na adresu externého zariadenia útočníka.

Authentication attack – útočník nepozná tajný kľúč K, ale môže odpočúvať authentifikačnú správu.

Slow attack: WEP Sniffing – pre rozlúsknutie 64-bit WEP kľúča 50tis. – 200tis. paketov, ktoré obsahujú Initializačný vektor (IV), ale len ¼ zo všetkých prijatých paketov ho obsahuje, preto musíme prijať 200-800tis paketov, čo môže trvať dosť dlho.

Fast attack: Packet injection – posielanie paketov v sieti do už vytvoreného spojenia. Vyžaduje si to špeciálny driver.

WPA a WPA2

Jená sa o bezpečnostné protokoly a certifikačné programy, ktoré boli vyvinuté Wi-Fi Alianciou pre zabezpečenie bezdrôtových sieti. Boli vytvorené v reakcii na nedostatky zistené v WEP

WPA – dostupným sa stal v roku 2003. Predstavoval medziprodukt v očakávaní dostupnosti bezpečnejšieho a komplexnejšieho WPA2. Oproti WEP má nasledujúce zlepšenia:

- Väčší tajný kľúč (128-bit) a inicializačné dáta (48-bit)
- Okrem zdieľaného tajomstva podporuje aj iné formi overovania ako username/password
- Kryptovacia metóda pre kontrolu integrity
- Počítadlo framov, ktoré má zabrániť opakovaným útokom

WPA2 – dostupným sa stal v roku 2004. Oproti WPA obsahuje zmeny skôr vo filozófii ako napr používanie AES namiesto RC4, MAC obsahuje counter mode.

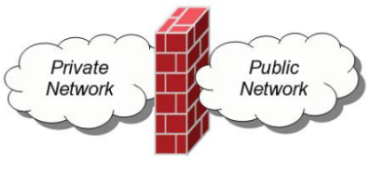
Alternatívy a doplnky:

WEP,WPA a WPA2 poskytujú ochranu len po prístupový bod. Za prístupovým bodom nie je žiadne zabezpečenie.

Iné metódy možno šifrovať end-to-end (SSH,SSL,VPN atď.). End-to-end poskytuje nepretržitú ochranu a integritu prenášaných dát tým, že ich na vstupnom bide zakóduje a v cieľovom dekóduje.

3. Firewalls, Tunnels, and Network Intrusion Detection

Firewalls – súbor bezpečnostných opatrení, ktoré majú chrániť vnútornú sieť (bezpečná, dôverihodná) pred neoprávneným elektronickým prístupom z vonkajšej nedôverihodnej siete ako napr. Internet. Jej hlavná úloha spočíva v kontrole paketov pre prichádzajúcu a odchádzajúcu sieťovu prevázdku a rozhodovanie na základe preddefinovaných pravidiel (**firewall policies**), či má byť daná komunikácia povolená.



Policie Action – pakety prúdiace cey firewall môžu mať jeden z 3 výsledkov:

- Prijaj (Accepted)– prešiel cez firewall
- Ukončená (Dropped) – nepovolený prechod, žiadna indikácia poruchy
- Zamietnuté (Rejected) – nepovolený prechod, snaha informovať zdroj paketu

Pri kontrole paketov sa preverujú ich vlastnosti (ako zdrojová a cieľová IP adresy a porty, či neobsahuje výrusy) ale aj typ protokolu (TCP, UDP)

Blacklist and Whitelist

- Dve základné pravidlá pri vytvárani firewall pravidiel, aby sa minimalizovala zraniteľnosť voči okolitému svetu a zachovala funkčnošt vnútorného.

Blacklist – sú povolené všetky pakety okrem tých, ktoré vyhovujú pravidlám v blackliste. Táto konfigurácia je pružnejšia z toho pohľadu, že vnútorná sieť nie je narúšaná firewallom. Ale na druhej strane je naivná v tom, že správca siete dokáže vymenovať všetky škodlivé vlastnosti.
Whitelist – bezpečnejší prístup, v ktorom sú pakety drooped alebo rejected ak nie su firewallom výslovne povolené.

Firewall types

- Packet filters (stateless) – pracuje na nízkej úrovni TCP/IP. Cez firevall sú pustené iba pakety, ktoré odpovedajú sade pravidiel, ktoré definuje správca firewallu.
- „Statefull“ filters – vedie si evidenciu všetkých spojení, ktoré cez neho prešli a na základe toho rozhoduje, či sa jedná o počiatočný paket nového spojenia, časť existujúceho spojenia, alebo sa jedná o neplatný paket.
- Application layer – pracuje na aplikačnej úrovni TCP/IP. Royhoduje o tom, či proces môže prijať poskytované spojenie. To znamená, že môže zachytiť všetky pakety smerujúce do/z aplikácie. Takýmto spôsombom sa dá obmedziť/zabrániť šíreniu počítačových cervov alebo trójskych koňov.

Stateless firewall – sú efektívnejšie, lebo kontrolujú len hlavičkovú časť každého paketu. Nevýhodou je, že nemajú žiadnu pamäť o predchádzajúcich paketoch. Týmpádom nie je schopný zistiť, či daný paket je súčasťou nejakého existujúceho spojenia, či sa snaží vytvoriť nové spojenie. Čo ho robí náchylným na Spoofing attack.

Statefull firewall – udržuje stav sieťového pripojenia v pamäti. Uchováva informácie ako IP adresy a porty, poradové čísla paketov. Tieto informácie si ukladá do dynamických tabuliek a na základe týchto údajov vyhodnocuje nové pakety a rozhoduje o tom, či patria do danej komunikácie.

Tunnels – obsah TCP paketov nie je zvyčajne šifrovaný, takže ak niekto odpočúva na TCP spojení, môže vidieť celý obsah.Jeden zo spôsobov, ako zabrániť takémuto odpočúvaniu bez zmeny fotvéru, ktorý vykonáva danú komunikáciu je použitie **tunelling protocol**- komunikácia medzi klientom a serverom je automaticky šifrovaná, takže odpočúvanie je neuskutočniteľné.

Secure Shell (SSH) – sieťový kryptografický protokol pre bezpečnú dátovú komunikáciu.

1) Klient sa pripojí pomocou TCP session
2) Klient a server si vymenia informácie ako sú : podporované metódy šifrovania a verzie protokolov. Vyberú si sadu protokolov, ktoré ten druhý podporuje.
3) Vymenia si tajne kľúče, aby si mohli vytvoriť tajný zdielaný kľúč, ktorý bude používaný pri ich ďalšej komunikácii
4) Server pošle klientovi zoznam možných foriem autentifikácie, ktoré klient postupne vyskúša. Najčastejšie pomocou hesla alebo **public-key authentication method** :
    a. Klient odošle na server svoj verejný kľúč
    b. Server skontroluje, či tento kľúč je v zozname autorizovaných kľúčov. Ak áno, tak serevr týmto kľúčom zašifruje výzvu a odošle ju klientovi
    c. Klient výzvu dešifruje svojim privátnym kľúčom a pošle odpoveď na server, čo dokazuje jeho identitu.
5) Po úspešnom overení server sprístupní klientovi svoje zdroje ako je napr. Príkazový riadok

IPSec – definuje zoznam protokolov pre ochranu v sieťovej komunikácii. Každý IP paket je šifrovaný. Každý protokol môže pracovať v jednom z 2 režimov:

- Transport mode – pred dátovú časť paketu sa vloží do hlavičky ďalšia dodatočná IPSec informácia.
- Tunnel mode – paket sa zašifruje a potom sa zapuzdrí do ďalšieho paketu s novou IP hlavičkou.

Virtual Private Networking (VPN) – Umožňuje hostiteľskému počítaču odosielať/prijímať dáta v rámci zdieľaných/verejých sieti, akoby sa jednalo o privátnu sieť so všetkými funkciami, bezpečnosťou a politikou riadenia privátnej siete.

Types of VPNs

- Remote access VPN – umožňuje klientovi prístup na súkromnú sieť označovanú ako intranet. Napr. firma chce umožniť zamestnancom vzdialený prístup k firemnej sieti, ale aby to vyzeralo ze sú pripojený lokálne.
- Site-to-site VPN – bezpečný most medzi 2> fyzickými sieťami.

Intrusion Detection System (IDS)

- Narušenie (Intrusion) - akcia zameraná na narušenie bezpečnosti
- Detekcia narušenia – identifikácia narušení pomocou ich podpisov a následne publikovanie správy o narušení.
- Prevencia narušení – detekcia narušení a riadenie automatických reakcií na narušenie

IDS Components – Správca IDS spracuváva dáta z IDS senzorov s cieľom určiť, či nedošlo k narušeniu. Toto rozhodovanie je založené na súbore pravidiel a podmienok, ktoré definujú pravdepodobné prieniky.Ak správca detekuje narušenie zaznie ALARM.

Intrusions

- IDS ja navrhnutý tak aby odhalil rôzne hrozby vrátane:
  - Masquerader – používa falošnú identitu legitívneho používateľa k získaniu prístupu
  - Misfeasor – opávnený používateľ, ktorý vykonáva činnosť, na ktorú nie je oprávnený
  - Clandestine user – používateľ sa znaží utajiť svoje akcie mazaním log súborov
- IDS je určený na detekciu automatizovaných útokov a hrozieb vrátane
  - Port scan – zhromažďovanie informácii za účelom určenia ktoré porty sú otvorené pre TCP spojenie
  - Denial-of-service attack – sieťový útok chce premôcť hostiteľa a vypnúť legitívny prístup
  - Malware attack - replicating malicious software attacks
  - ARP spoofing – pokus o presmerovanie IP v lokalnej sieti
  - DNS cache poisoning

Base-Rate Fallacy – táta chyba nastane keď sa pravdepodobnosť niektorých posudzovaných udalosti posudzujú bez ohľadu na predchádzajúcu pravdepodobnosť. Tým môže byť účinnosť niektorých IDS nesprávna kvôli štatistickej chybe.

IDS data - Dorothy Denning identifikovala niekoľko údajov, ktoré je potrebné zahrnúť do záznamov IDS udalosti:

- Subject – iniciátor útoku
- Object – čo je terčom útoku (súbor, príkaz, zariadenie)
- Action – operácia vykonaná smerom k objektu
- Exception-condition – chybové hlásenie, ktoré vzniklo pri akcii
- Resource-usage – prostriedky vynaložené ako reakcia na túto akciu
- Time-stamp – jedinečný identifikátor pre začiatok akcie

Types of IDS

- Rule-Based - pravidlá identifikujúce typy akcii odpovedajúce známim profilom pre vznik útoku. To znamená že sa rozpozná rukopis daného útoku. Keď je rukopis rozoznaný, okamžite zaznie alarm aj s popisom útoku.
- Statistical – štatistická reprezentácia typického spôsobu používania hostiteľom. Uloží sa nejaký profil o tom, ako bežne používa zariadenie. Tým môžeme určiť nezvičajné spôsoby správania.

7 Web Security

7.1 HTML (Hypertext markup language)

- Znackovaci jazyk urceny na vytvaranie webovych stranok a inych informacii zobrazitelnych vo webovom prehliadaci
- Opisuje obsah a formatiovanie webovych stranok.
- Renderuje sa prostrednictvom prehliadaca.

7.1.1 Funkcie

- Staticky popisovaci jazyk
- Podporuje linkovanie na ine stranky a vkladanie obrazkov na zaklade odkazu.
- Vstupy pouzivatela sa posielaju na server pomocou pormularov

7.1.2 Rozsirenia

- Dodatocny medialny obsah (napr. PDF, video) podporovany prostrednictvom pluginov prehliadaca
- Vkladanie programov v podporovanych jazykov (napr. JavaScript, Java), ktore poskytuju dynamicky obsah, ktory interaguje s pouzivatelom, vie modifikovat pouzivatelske prostredie prehliadaca a vykonava sa na klientskom pocitaci

7.2 Phishing

je činnosť, pri ktorej sa podvodník snaží vylákať od používateľov rôzne heslá, napr. k bankovému účtu. Väčšinou prebieha tak, že sa založí webstránka, ktorá vyzerá ako presná kópia už existujúcej dôveryhodnej stránky, alebo ponúka nejaké výhody po prihlásení cez ich webstránku. Meno a heslo zadané do phishingovej stránky, sa odošlú podvodníkovi, ktorý ich môže zneužiť. Phishing môže prebiehať aj tak, že sa rozposielajú e-maily, ktoré oznamujú používateľom zmenu účtu alebo jeho obnovenie a tak utocnik vie ziskat heslá.

7.2.1 Techniky pishingu

- **Podobna Url adresa**: url pishingovej web stranky sa nezhoduje s url adresou originalnou web stranky (napr. www.paypall.com != www.paypal.com)
- **Zkreslene odkazy**: odkaz nemusi viest na danu adresu ktora je v odkaze napisana. Kazdy prehliadac zobrazuje realnu adresu odkazu povacsinou v pravo alebo v lavo dole.
- **Vyuzitie poddomen**: napr. odkaz www.slovenskasporitelna.novasluzba.sk vyzera tak ze vedie na web slovenskej sporitelne, ale v skutocnosti odkazuje na web pishingovej webstranky nova sluzba.

7.3 IE Image Crach

Chyba v implementacii weboveho prehliadaca moze viest k DOS utoku. Napriklad klasicky priklad s velkym obrazkom v internet exploreri je dobrym prikladom:

- Vytvori sa jednoducha webstranka s extremne velkymi rozmermi obrazka (`<img src="./imagecrash.jpg" width="9999999" height="9999999">`) a po otvoreni tejto web stranky, IE padne. Dokonca sa moze niekedy stat ze windows zamrzne.

Tato chyba sa stale vyskytuje v poslednej verzii IE.

7.4 Mobilny kod (Mobile Code)

- Spustitelny program
- Odoslany cez pocitacovu siet
- Vykonavany u klienta

Priklady: JavaScript, ActiveX, Java Plugins, Integrovane Java virtualne masiny

7.4.1 JavaScript

- Je to skriptovaci jazyk intrepretovany webovym prehliadacom
- Kod je uzavrety medzi HTML znazkami <script></script>
- Umoznuje definovanie funkcii: `<script>function hello() { alert(„Hello“) }</script>`
- Odchytavanie eventov vlozenych v HTML: `<img src="picture.gif" onMouseOver="javascript:hello()">`
- Vstavane funkcie pre pracu s oknom prehliadaca: `window.open(„http://www.google.com“)`
- Klik utok: `<a onClick="window.open(http://www.evilsite.com)" href="http://www.trustedsite.com/">Trust me!</a>`

7.4.2 ActiveX vs. Java

7.4.2.1 ActiveX

- Technologia podporujuca iba internet explorer beziaci na windowse
- Vykonavanie binarneho kodu v zaujme prehliadaca
- Moze pristupovat k suborom na disku
- Podporuje podpisany kod
- Kod moze spustat hocijaka stranka (do verzii IE7)
- Konfigurovatelne moznosti: povolit, zakazat, vynutit, vyzadovat potvrdenie administratorom

7.4.2.2 Java Applet

- Platformovo nezavysla technologia prostrednictvom pluginu vo webovom prehliadaci
- Java kod je spustany v ramci prehliadaca
- Sandboxovane vykonavanie programu
- Podporuje podpisany kod
- Applet moze bezat len na stranke v ktorej je vlozeny.
- Doveryhodne aplety mozu byt vykonavane mimo sandboxu

7.4.3 Autentifikacia v ActiveX

- Podpisany ActiveX program poziazada pouzivatela o udelenie prav pre spustanie. Ak pouzivatel tieto prava udeli, tak tento ActiveX program sa spusti s rovnakymi pravami ako ma pouzivatel
- Zaskrtnutim checkboxu „Vzdy doveruj obsahu od...“ automaticky bude povolovat spustanie tohto ActiveX programu (Pravdepodobne zly napad ☺).

7.4.4 Doveryhodne a nedoveryhodne ActiveX programy

Doveryhodny vydavatelia

- Zoznam ulozeny vo Windows registroch
- Utocnikov ActiveX program vie zmodifikovat tuto tabulku registrov a urobit zo seba doveryhodneho vydavatela
- Vsetky programy ktore su od doveryhodneho vydavatela sa spustaju bez vyzvy pouzivatela na udelenie prav

Nepodpisane programy

- Pouzivatel je upozorneny ze program je nepodpisany a dava pouzivatelovi moznost tento program povolit alebo zamietnut
- Aj ked je program zamietnuty, tak aj tak bol stiahnuty
- Program po zamietnuti sa nespusti ale sa ani nezmaze

7.4.5 Klasicke ActiveX Exploity

Explorer exploit:

- ActiveX program, pre ktory kupil digitalny podpis
- Program znizoval vykon pocitaca

Runner exploit:

- ActiveX program, ktory otvaral DOS a spustal prikazy ako format C a podobne nebezpecne prikazy.

Quicken exploit:

- Quicken je osobny financny nastroj na spravu financii. Moze byt nakonfigurovany pre prihlasovanie do bankovych a kreditnych stranok
- Exploit vyhladaval v pocitaci Quicken a vykonaval transakcie z uctu obete na ucet utocnika.

7.5 Cookies

Su informacie ulozene v pocitaci spojene s niektorym specifickym serverom.

- Ak navstivime specificku web stranku, tak server moze ulozit nejaku informaciu v pocitaci ako cookie
- Vzdy ked opat navstivime danu webstranku, cookie je znovu odoslana na danny server.
- Efektivne sa pouziva na udrzanie stavovej informacie nad sessions.
- Mozu obsahovat hocijaku informaciu
- Moze obsahovat citlive udaje ako hesla, informacie o kreditnej karte, a pod.
- Sessions cookies, trvale/netrvale cookies
- Skoro kazda web stranka pouziva cookies
- Vela stranok vyzaduje povolenie cookies aby mohli stranku pouzivat
- Ich ulozenie v pocitaci sa prirodzene hodi exploitom (napr. ActiveX mozu zneuzit cookies)
- Pouzivatel moze respektive by mal vymazat cookies z pocitaca
- Webove prehliadace podporuju vypnutie cookies a ich povolenie len pre konkretne vymenovane webove stranky

Expiracia cookies:

- Expiracia je defaultne nastavena prostrednictvom session zo stranok servera
- To znamena ze cookie budu aktivne nejaku dobu

Manazovanie cookies:

- Vymazanie konkretnej cookie
- Vymazanie vsetkych cookies
- Zobrazenie informacii o konkretnych cookies

7.6 Cross site scripting (XSS)

Utocnik vlozi kod do stranky vygenerovanej web aplikaciou.

- Skript moze by skodlivy kod
- Javascript (Ajax), VBScript, ActiveX, HTML alebo Flash

Hrozby:

- Phishing, hijacking, zmena pouzivatelskych nastaveni, odcudzenie cookies, klamliva reklama, vykonavanie kodu u klienta

7.6.1 Klientska obrana proti XSS

Proxy

- Sledovanie trafficu HTTP medzi prehladacom a web serverom
- Hladanie specialnych HTML znakov
- Enkodovat vsetky znaky predtym ako je stranka vyrenderovana (napr. firefox plugin **NoScript**)

Aplikacny firewall

- Analyza HTML stranok za ucelom najst hyperlinky ktore moze viest k uniku citlivych informacii

- Zastavenie zlych requestov s pouzitim sady pravidiel

Auditovaci system

- Sledovanie spustania javascriptoveho kodu a porovnavanie operacii oproti vysoko prioritnym podmienkam pre detekovanie skodliveho kodu

7.7 SQL Injection Attack

- Vela web aplikacii ziskava pouzivatelov vstup z formulara
- Casto tento vstup je okamzite pouzity pre skontruovanie SQL query, ktora je odoslana do databazy. Napriklad: SELECT user FROM table WHERE name = ‚user_input';
- SQL injection utok pozostava vo vlozeni SQL prikazov do vstupu.

SQL injection utok pri logine:

- Standartne query pre autentifikovanie pouzivatelov vyzera takto:
  `select * from users where user='$user' AND pwd='$password'`
- Server nastavi hodnoty premennych $username a $password z pouzivatelovho vstupu
  Tieto premenne su potom pouzite pri konstruovani SQL query:
  `select * from users where user='$username' AND pwd='$password'`
- Utocnik moze do vstupu zadat specialne znaky (SQL syntax);
  `select * from users where user='M` OR `1=1' AND pwd='M` OR `1=1'`
- Utocnik po zadani takehoto vstupu ziska pristup bez poznania spravneho puzivatelskeho mena ci hesla
- Takymto sposobom vie utocnik ziskat napriklad celu tabulku pouzivatelov:
  `select user,pwd from users where user='M` OR `1=1`

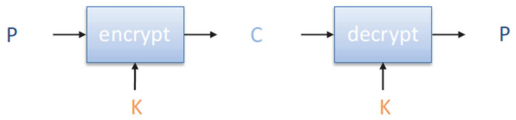
Spravne riesenie loginu:

- Mozeme pouzit Escape metodu, ktora vsetky skodlive znaky zmeni:
  `Escape("t' c")` – nam vrati vysledok v takejto podobe: **`"t \' c"`**
  `$usern = escape("M' ;drop table user;")`
  `select user,pwd from users where user='$usern'`
- Vysledkom je bezpecna query:
  `select user,pwd from users where user='M\' drop table user;\''`

1. CryptoConcepts

Symetric Cryptosystem

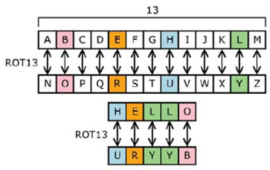
- Scenár – Alice chce poslať správu Bobovi, ale komunikačný kanál môže byť odpočúvaný. Ak sa už vopred dohodli na nejakej cymetrickej šifrovacej schéme a tajnom kľúči K, tak správa môže byť poslaná šifrovane (ciphertext C).
- Problém
  - Čo je dobrý symetrický šifrovací systém?
  - Aká je zložitosť šifrovania/dešifrovania?
  - Aká je veľkosť ciphertext vzhľadom k otvoreným textom?



Základ

- Značenia
  - K – tajný kľúč
  - EK (P) – šifrovanie
  - DK (C) – dešifrovanie
  - šifrovanie a dešifrovanie sú permutačné funkcie
- Účinnosť
  - Funckie EK a DK by mali mať efektívne algoritmy
- Konzistencia
  - Dešifrovaním ciphertext dostaneme pôvodný čistý text
  - DK(EK(P)) = P

Brute-Force attack – systematicky skúša všetky možné kľúče K kým nenájde ten správny. Z toho dôvodu musí byť kľúč dostatčne dlhý, aby sme takýto útok znemožnili.



Substitution Cipher (šifra) – každý znak je unikátne nahradený iným. Existuje 26! možných kombinácii rôznych šifier. Obľúbenou šifrou je ROT13

Frequency analysis – táto metóda sa používa ako pomôčka pri klasickom rozbíjaní šifier. Je založená na tom, že písmená a skupiny písmen sa v ciphertexte vyskytujú s určitou frekvenciou.

Substitution Boxes – substitúcia môže byť vykonávaná aj nad binárnimi číslami. Tieto substitúcie sú obvikle označované ako substitution boxes alebo S-boxes

	00	01	10	11
00	0011	0100	1111	0001
01	1010	0110	0101	1011
10	1110	1101	0100	0010
11	0111	0000	1001	1100

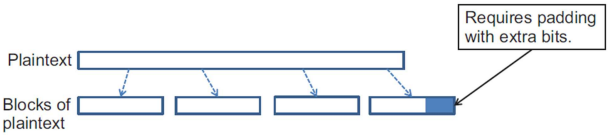
(a)

	0	1	2	3
0	3	8	15	1
1	10	6	5	11
2	14	13	4	2
3	7	0	9	12

(b)

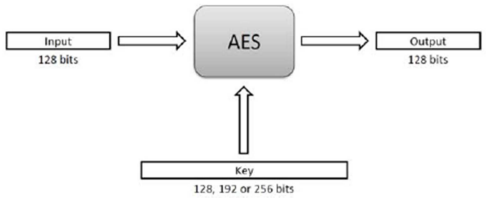
One-time pad – jedná sa o úplne neprekonateľnú šifru. Bola vytvorená v roku 1917 (Joseph Mauborgne and Gilbert Vernam). Každý bit/znak plaintextu sa zašifruje tak, že sa pridá ďalší bit alebo znak z tajného náhodného kľúča o rovnakej dĺžke ako plaintext. Ak je kľúč skutočne náhodný, tak bez toho, aby sme ho vedeli, nebudeme schopný dešifrovať. Nevíhoda – kľúč musí byť aspoň taký dlhý ako text, a nesmie sa opakovane používať.

Block-ciphers – Paintext a ciphertext majú pevnú dĺžku. Plaintext o dĺžke n je rozdelený do sekvencie m blokov. Každá správa sa rozdelí do sekvecnie blokov a následne šifruje/dešifruje.



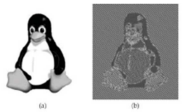
Block Cipher in Praxis

- DES (Data Encryption Standard) – vyvinutý spoločnosťou IBM a NIST prijatý v 1977. Pozostáva zo 64-bit blokov a 56-bit kľúčov. Search attack - uskutočniteľný
- 3DES (Triple DES) – Vnorené aplikácie DES s 3 rôynzmi kľúčmi KA,KB,KC. Efektívna dĺžka kľúča je 168-bit. Search attack – neuskutočniteľný. C = EKC(DKB(EKA(P))); P = DKA(EKB(DKC(C))). Odpovedá DES ak KA=KB=KC (backward compatible)
- AES (Advanced Encryption Standard) – 2001 vybraný NISTprostredníctvom medzinárodnej súťaže a verejnej diskusie. 128-bit bloky a kľúče niekoľkých dĺžok (128, 192, 256-bit). Search attack – nemožný. Algoritmus sa vykoná v 10 kolách, pričom v 1. kole XOR plaintextu a kľúča. Každé kolo poyostáva yzo 4 časti (S-box substitúcia, permutácia, násobenie matíc, XOR s kľúčom).



Block Cipher Modes

ECB (Electronic Code Book) – najjednoduchšie šifrovanie. Správa sa roydelí do blokov a každý blok je šifrovaný samostatne. Funguje dobre s náhodnými reťazcami (kľúče, inicializačné vektory) a reťazcami odpovedajúce jednému bloku. Dokumenty a obrázky nie sú vhodné.



CBC (Cipher Block Chaining) – predchádzajúci ciphertext je kombinovaný s aktuálnym paintextom cez XOR. Aby bola každá správa unikátna, v prvom bloku sa musí použiť inicializačný vektor (náhodný samostatne prenášaný šifrovaný blok). Metóda je rýchla a relatívne

jednoduchá. Slabou stránkou je že vyžaduje spoľahlivý prenos všetkých blokov postupne a nie je vhodný pre aplikácie, ktoré umožňujú stratu paketov (streamovanie hudby, videa)

Stream cipher

- Key stream – pseudo-náhodná postupnost bitov. Môže byť generovaný za behu na jednom bit/byte v čase.
- Stream cipher – XOR palintextu s key streamom. Vhodný pre obyčajný text ľubovoľnej dĺžky generovaný za behu (napr. media stream)
- Synchronous stream – key stream zísakný len z tajného kľúča. Funguje pre nespoľahlivé kanály, kde plaintext obsahuje pakety s poradovými číslami.
- Self-synchronizing stream – Key steram získaný z tajného kľúča a q predchádzajúcich ciphertext. Stratené paket spôsobujú čakanie q krokov pred dešifrovaním.

Key stream generation

- RC4 – Navrhnutý v roku 1987. Bol obchodným tajomstvom až do roku 1994. Používa kľuče až o veľkosti 2048-bit . Jednoduchý algoritmus.
- Block cipher in counter mode(CTR) – používa blokovú šifru o veľkosti bloku b. Tajný kľúč je dvojica (K,t) kde K=kľúč a t (counter) je b-bitová hodnota. Key steram je konkatenáciou ciphertext. Možno používať kratšie zreťazenia s náhodnou hodnotou.

Attack on Stream cipher

- Repetition attack – Ak je key stream znovu použitý, útočník ho získa XOR dvoch plaintextov
- Insertion attack – prenos plaintextu (preposlanie mailovej správy s novým číslom)

Public Key Encription

Greatest Common Divisor (Najväčší spoločný deliteľ) – gcd(...), najväčšie číslo, ktorým sú deliteľné čísla A a B.
Modular arithmetic – celá časť po delení dvoch čísel, r = a mod b
Euclids GCD Algorithm – opakovane sa použije vzorec gcd(a, b) = gcd(b, a mod b)
– gcd(412, 260) = 4

a	412	260	152	108	44	20	4
b	260	152	108	44	20	4	0

Multiplicative Inverses – Zvyšok modula a kladného čísla n je množina Zn {0, 1, 2, …, (n-1)}. Majme dve prvky x a y , ktore patria množine a platí xy mod n = 1. Potom hovoríme že y je multiplikatívne inverzné x => $y=x^{-1}$.

- Example:
  - Multiplicative inverses of the residues modulo 11

x	0	1	2	3	4	5	6	7	8	9	10
x^{-1}		1	6	4	3	9	2	8	7	5	10

Fermat's Little Theorem – Nech p je prvočíslo, potom pre každé nenulové x zo Z_p, platí x^{p-1} mod p = 1

Euler's Theorem – ak n a a su kladné čísla, potom $a^{\varphi(n)}$ (mod n) = 1, kde φ(n) - Euler's totient function je aritmetická funkcia, ktorá spočíta počet nesúdeliteľných čisel k n.

Example ($n = 10$)

$3^{\phi(10)} \bmod 10 = 3^4 \bmod 10 = 81 \bmod 10 = 1$

$7^{\phi(10)} \bmod 10 = 7^4 \bmod 10 = 2401 \bmod 10 = 1$

$9^{\phi(10)} \bmod 10 = 9^4 \bmod 10 = 6561 \bmod 10 = 1$

RSA Cryptosystem – musí existovať základný utajený faktor (kľúč). Hocikto môže použiť verejný kľúč na zašifrovanie správy, ale len ten, kto pozná tajný faktor, môže dekódovať správu. RSA zahrňuje 3 kroky:

Generovanie kľúča

1) Zvolíme si 2 prvočísla p q – zvolené náhodne s podobnou bit-dĺžkou
2) Vypočítame n = pq. Používa sa ako modul pre súkromné i verejné kľúče. Jeho dĺžka v bit vyjadruje dĺžku kľúča
3) Vypočítame φ(n) = (p – 1)(q – 1), kde φ je Euler's totient function
4) Zvolíme si celé čislo e take, že 1 < e < φ(n) a GDC(e, φ(n)) = 1, t.j. e a φ(n) sú nesúdeliteľné.
5) Určíme d, d ≡ e−1 (mod φ(n)). d je vedený ako exponent súkromného kľúča

Verejný kľúč – skladá sa z modula n a verejného exponentu e

Súkromný kľúč – skladá sa z modula n a súkromného expomentu d, ktorý musí byť utajený takisto ako p, q, and φ(n), lebo sa z nich dá vypočítať d.

Šifrovanie

- Alice pošle Bobovi verejný kľúč (n,e). Bob zoberie svoju správu M, zmení správu na celé číslo m, $0 \leq m < n$ použitím padding schémy a vzráta svoj ciphertext $c = m^e \bmod n$

Dešifrovanie

- Alice môže obnoviť m z c pomocou exponenta súkromného kľúča d : $m = c^d \bmod n$. Pomocou reverznej padding schémy môže z m obnoviť pôvodnú správu M

Cryptographic Hash Function

Hash function - označíme h, mapuje plaintext x na hodnotu fixnej dĺžky x = h(P). Kolízia – dvojice paintextov P a Q, ktoré majú rovnakú hash value h(P) = h(Q). Kolízie sú nevyhnutné. Hash table – hash funkcia sa tu využíva pre rýchle nájdenie záznamov. Funkcia sa tu používa pre mapovanie hľadaného kľúča a indexu. Index určuje miesto v hash tabuľke, kde by sa mal záznam nachádzať.

Cryptographic Hash Function – zoberie ľubovoľný blok dát a vráti jeho hash hodnotu. Takže v prípade ak sa zmenia údaje (náhodne/úmyselne) tak sa zmení aj hash hodnota. Vlastnosti: ľahký výpočet hash hodnoty, nemožne zmeniť správu bez zmeny hash, nemožné nájsť 2 rozne správy s rovnakou hash.

Birthday attack – snaží sa nájsť kolízie v hash funkciách. Náhodne sa vygenerujú plaintexty X1..Xn. Pre každý palintext Xi sa vyráta jeho hash hodnota yi= h(Xi) a testuje sa či yi=yj kde j<i. Končime hneď, keď sa nájde kolízia.

Message-Digest Algorithm 5 (MD5) – vytvorený v roku 1991 Ron Rivest. Používa 128-bit hash hodnotu. MD5 je zvyčajne vyjadrená ako hexadecimálne číslo o dĺžke 32 znakov. Stále je široko používaný v aplikáciach, aj napriek tomu že sa v ňom našli závažné chyby.

- Choosen-prefix colision attack – objavený Marc Stevens, Arjen Lenstra and Benne de Weger. Podarilo sa im vytvoriť dva plaintexty s rovnakým kontrolným súčtom MD5.

Secure Hash Algorithm (SHA) – vytvorený spoločnosťou NSA a spoločnosťou NIST schválený ako federálny štandard.

- SHA0, SHA1 – 160-bit hash hodnota. Menej závažné chyby zabezpečenia než v MD5
- SHA2 - 256 bitov (SHA-256) alebo 512 bitov (SHA-512). Stále považované za bezpečné aj napriek tomu, že boli zverejnené útočné techniky
- SHA3 (Keccak) – zvolená v roku 2012 po verejnej súťaži. Podporuje hash dĺžky ako SHA2 a jeho vnútorná štruktúra sa výrazne líši od predchádzajúcich SHA

Data Integrity: Application of Cryptographic Hash function

Message Authentication Code (MAC) – krátky kus informácie určený pre zabezpečenie integrity (či nebola správa náhodne alebo úmyselne pozmenená) a autentickosti správy (pravosť správy, či sa jedná o pôvodnú správu). Príjemca a odosielateľ zdieľajú tajný kľúč K pomocou

ktorého sa vypočita hash hodnota správy M - MAC c = h(K,M) . Kľúč sa môže odoslať v samostatnej správe

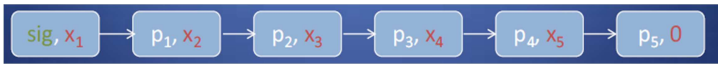
HMAC (Hashed MAC) – kombinuje tajný kľúč s hash funkciou - $h(K \oplus A \| h(K \oplus B \| M))$

Zabezpečenie komunikačných kanálov

- Sign and encrypt – prenáša sa šifrovaná dvojica (správa, podpis)
- MAC and encript - prenáša sa šifrovaná dvojica (správa, MAC). Efektívnejšie a kratší výpočet ako Sign and encrypt.

Hash Chain – aplikovanie kryptografickej hash funkcie na časti dát. Produkuje veľa jednorazových kľúčov z jedného kľúča alebo hesla.

Validation Chain – pre každý paintext p...pn sa vypočita jeho hash : xi = h(pi || xi1). Pakety sa posilajú v tvare (pi,xi+1) co znamená že každy paket obsahuje hash nasledujúceho paketu. Prvý paket obsahuje podpis



Hash tree – vyvážený binárny strom. Môže byť použitý pre overenie akýchkoľvek údajov. V súčasnosti sa najviac využíva prei peer-to-peer aby sme sa uistili, že dátivé bloky ktoré prijímame sú nepoškodené a bez zmeny.

9 Prednaska

9.1 Kerberos

- Sietovy autentifikacny protokol, ktory autentifikuje klientov na sluzby a naopak
- po nezabezpecenej sieti je mozne bezpecne overit identitu medzi dvomi ucastnikmi
- postaveny na symetrickom sifrovani
- pouziva sa port 88
- vydavany pod licenciou podobnou BSD
- vyuziva koncept listka ako tokenu, ktory reprezentuje identitu pouzivatela
- listky su digitalne dokumenty, ktore drzia session kluce. Tieto kluce su typicky vydane pocas login session a potom mozu byt pouzite namiesto hesiel pre rozne Kerberizovane sluzby.

9.1.1 Kerberos Servers

K dosiahnutiu bezpecnej autentifikacie, Kerberos pouziva doverihodnu tretiu stranu znamu ako disctribucne centruk klucov (**key distribution center - KDC**), ktore je tvorene 2 komponentami typicky integrovanimi do jedneho servera:

- autentifikacny server (**authentication server - AS**), ktory vykonava autentifikaciu
- udelovaci server pre listky (**ticket-granting server - TGS**), ktory udeluje listky pouzivatelom

Autentifikacny server udrzuje databazu ulozenych skrytych klucov od pouzivatelov a sluzieb. Skryty kluc pouzivatela je typicky vygenerovany pomocou funkcie ktora vygeneruje jednocestny hash z pouzivatelovho hesla.

9.1.2 Princip Kerberos autentifikacie

- pri sifrovani vystupuju tieto entity:
    - AS – autorizacny server
    - SS – servisne stredisko
    - TGS – ticket-granting server – riadiaci server
    - TGT – ticket granting ticket – ticket opravnujuci komunikaciu s TGS

1. Uzivatel zada login a heslo, vygeneruje na zaklade neho hash (**kluc1**), ten nikam neposiela
2. Poziada AS o pristup k sluzbe a odosle mu svoje IDcko (nezasifrovane)
3. AS skontroluje ci taky user existuje, ak ano, posle mu 2 spravy:

- sprava A - TGS kluc (**kluc2**) zasifrovany klucom **kluc1**
- sprava B - TGT obsahujuci ID, sietovu adresu, zivotnost ticketu a **kluc2**, toto vsetko je zasifrovane klucom TGS (**kluc3**)

4. klient obdrzi tieto 2 spravy, spravu A desifruje hashom, ktory ziska zo svojho loginu a hesla a pouzije na desifrovanie spravy s klucom 1
5. spravu B nie je schopny desifrovat pretoze user nema **kluc3**
6. ziadanim o pristup posiela user 2 spravy serveru

- sprava C - obsahuje spravu B a ID sluzby, ktorej pristup ziada
- sprava D - autentifikator (ID klienta a casova znacka) sifrovany klucom **kluc2**

7. TGS rozkoduje C a z nej ziska B, ktoru desifruje pomocou **kluc3**, z toho ziska **kluc2**
8. pomocou kluc2 desifruje D a odosle klientovi dalsie 2 spravy:

- sprava E - klient/server ticket obsahujuci ID klienta, jeho sietovu adresu, dobu platnosti a klient/server kluc (**kluc4**), to vsetko zasifrovane pomocou kluc5
- sprava F - klient/server kluc (**kluc4**) sifrovany klient/TGS klucom (**kluc2**)

9. klient ma teraz dost informacii k autentizacii voci SS. Klient sa k nemu pripoji a posle 2 spravy:

- sprava E - len preposle existujucu spravu E, ktoru uz dostal v kroku 8
- sprava G - novy autentifikatror, obsahuje ID klienta, casovu znacku, zasifrovane pomocou kluc4

10. SS desifruje E pomocou **kluc5** a ziska **kluc4**

11. SS desifruje G, z nej ziska autentifikator a posle klientovi spravu aby potvrdil svoju identitu a ochotu posluzit
- sprava H - inkrementovana casova znacka klientovho autintifikatora, zasifrovane **kluc4**

12. klient desifruje H pomocou **kluc4** a skontroluje ci je casova znacka spravne inkrementovana, pokial ano, moze doverovat serveru a moze zacat posielat ziadosti o sluzby
13. server poskytuje sluzby

9.1.3 Vyhody

- Kerberos protokol je navrhnuty byt bezpecny aj v nezabezpecenej sieti
- Pretoze je prenos sifrovany s pouzitim tajneho kluca, utocnik nemoze sfalsovat platny listok k ziskaniu neautorizovanemu pristupu k sluzbam
- Vyuziva symetricke sifrovanie ktore je vypoctovo efektivnejsie ako asymetricke

9.1.4 Nevyhody

- musi byt nepretrzity beh centralneho servera - pokial nebezi, nikto sa neprihlasi
- musi byt prisna synchronizacia casu - nesmie sa lisit o viac ako 5 minut, pretoze sa pouzivaju casove znamky
- ak sa utocnik dohodne s KDC, vsetky autentifikacne informacie o pouzivateloch a serveru na sieti budu odhalene

9.2 Bezpecnostna politika, Bezpecnostne modely, Bezpecnostna doverihodnost

9.2.1 Bezpecnostna politika

Definovany zoznam pravidiel obsahujuce:

- **Osoby (Subjekty)**: agenti ktori posobia v ramci systemu, ktori mozu byt definovani ako jednotlivci (prezident, CEO, CFO), role alebo skupiny jednotlivcov (pouzivatelia, administratori, generali, majori, dekani, manazeri, asistenti, utocnici, hostia) v ramci organizacie.
- **Objekty:** informacne (dokumenty, subory, databazy) a vypoctove zdroje (servery, softver) pre ktore je bezpecnostna politika navrhnuta tak aby ich chranila a spravovala
- **Akcie:** veci, ktore subjekty mozu alebo nemozu robit vzhladom k objektom. Napiklad: citanie alebo pisanie dokumentov, updatovanie softveru, pristupovat k obsahu databazy
- **Prava:** mapovanie medzi subjektmi, akciami a objektmi ktore jasne hovoria o tom kto aky druh akcie moze alebo nemoze vykonavat
- **Ochrany:** specificke bezpecnostne vlastnosti alebo pravidla ktore su obsiahnute v politike, ktore pomahaju dosiahnut bezpecnostne ciele ako dovernost, integritu, dostupnost alebo anonymitu

9.2.2 Bezpecnostne modely

9.2.2.1 Bezpecnostny model

- je abstrakcia ktora poskytuje koncepcny jazyk pre administratorov k specifikovaniu bezpecnostnej politiky.
- Definuje hierarchiu pristupovych alebo modifikacnych prav ktore clenovia organizacie mozu mat , takze subjekty v organizacii mozu lahko priradit specificke prava zalozene na pozicii tychto prav v hierarchii.
- Napriklad armadne prava k dokumentom: „unclassified (nezaradene)“, „confidential (doverne)“, „secret (tajne)“ a „top secret (super tajne)“

9.2.3 Diskretna kontrola pristupu - DAC (Discretionary Access Control)

- schema kde pouzivatelia maju moznost rozhodnut o pravach pristupu k ich vlastnym suborom
- DAC typicky uvadza koncept pouzivatelov a skupin a povoluje pouzivatelom nastavovat kontrolu pristupu na zaklade tychto kategorii
- Navyse DAC schemy povoluju pouzivatelom udelovat privilegia na zdroje ktore vlastnia ini pouzivatelia v ramci toho isteho systemu

9.2.4 Povinna kontrola pristupu – MAC (Mandatory Access Control)

- je viacej obmedzujuca schema ktora povoluje pouzivatelom definovat prava na subory, vzhladom na vlastnictvo. Namiesto nich bezpecnostne rozhodnutia robia administratori.
- Kazda bezpecnostna rola pozostava zo subjektu, ktory reprezentuje ucastnika pokusajuceho sa pridelit pristup objektu
- Security-Enhanced Linux (SELinux)
    - Obsahuje MAC

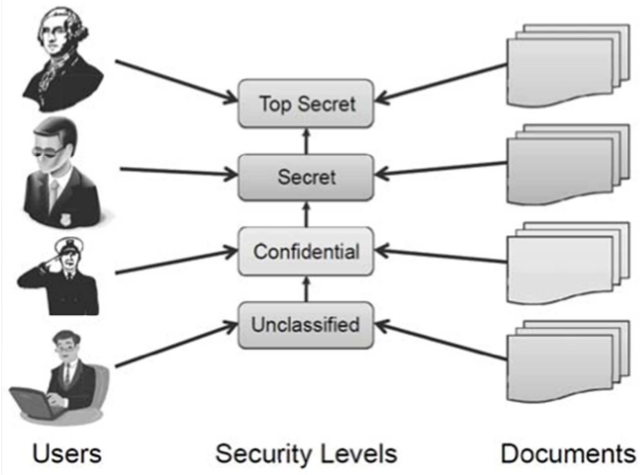
9.3 Bezpecnostna doverihodnost (Trust Management)

- je formalny ramec pre specifikovanie bezpecnostnej politiky v dannom jazyku, ktory je zvycajne typom logickeho alebo programovacieho jazyka, spolocne s mechanizmom pre zabezpecenie presadenia specifikovanej politiky
- pozostava z 2 komponentov:
    - jazyk politiky
    - kontrolor dodrziavania
- pravidla su specifikovane v jazyku politiky a su zabezbecenovane kontrolorom ktory kontroluje ci sa dodrziavaju
- ma pravidla ktore opisuju:
    - **Akcie** – operacie s bezpecnostnymi dosledkami na system
    - **Principy** – pouzivatelia, procesy, alebo ine entity ktore mozu vykonavat akcie v systeme
    - **Politiky** – precizne napisane pravidla ktore riadia ktore osoby su autorizovane k vykonaniu konkretnych akcii
    - **Osobne udaje** – digitalne podpisane dokumenty ktore viazu osobne identity k povolenym akciam, vratane prava k povolovaniu osob na delegovanie prav pre ine osoby

9.4 Modely kontroly pristupu

9.4.1 The Bell-La Padula (BLP) model

- je klasicky model povinnej kontroly pristupu (MAC) pre zabezpecenie dovernosti
- je odvodeny z vojenskehej viacvrstvovej bezpecnostnej paradigmy, ktora bola tradicne pouzivana vo vojenskych organizaciach pre klasifikaciu dokumentov
- ma striktne linearne poradie bezpecnostnych levelov tak, ze kazdy dokument ma specificky bezpecnostny level a kazdy pouzivatel je priradeny k striktnemu levelu pristupu, takze mozu pristupovat k vsetkym dokumentom s korespondujucim levelom a vsetkymi levelmi nizsie.



9.4.1.1 Uplne poradie a Ciastocne poradie

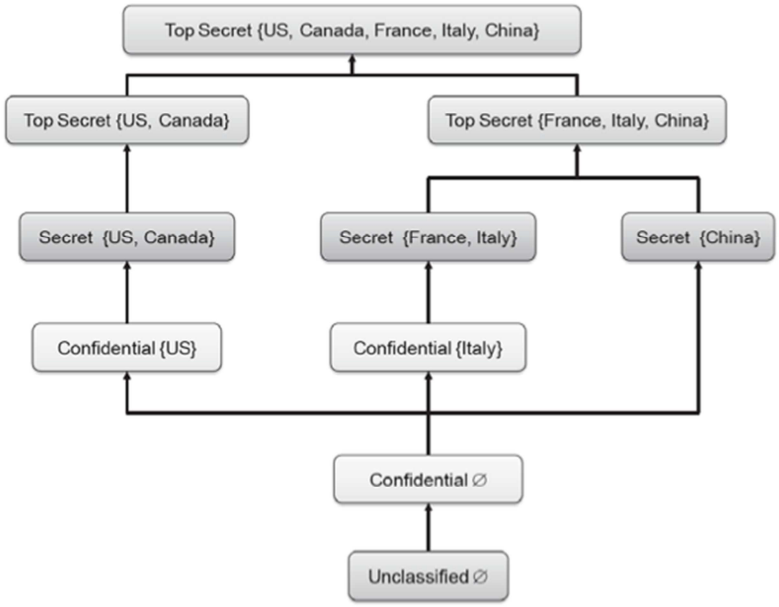
- linearne zoradzovanie pre dokumentov moze by definovane na zaklade porovnavacieho pravidla. Mozme povedat ze pravidlo definuje uplne poradie na univerzalnom zozname U ak splna tieto vlastnosti:

- **Reflexia**: ak x patri U potom x <= x
- **Antisymetria**: ak x <= y a y <= x potom x = y
- **Tranzitivnost**: ak x <= y a y <= z potom x <=z
- **Uplnost**: ak x a y su v U potom x <= y alebo y <= x

- vsetky bezne definicie „menej nez alebo rovne k“ pre cisla, ako integre a realne cisla su uplne poradie
- ak vynechame poziadavku uplnosti potom dostavame Ciastocne poradie
  - priklad ciastocneho poradia je zoznam kurzov na univerzite kde mozme povedat ze 2 kurzy A a B, A<=B ak A je prerekvizita pre B

9.4.1.2 Ako BLB model pracuje

- kazdy level bezpecnosti v BLP formulari je ciastocnym poradim, <=
- kazdy objekt, x, je priradeny do levela bezpecnosti, L(x), Podobne kazdy pouzivatel, u, je priradeny do levela bezpecnosti, L(u). Pristup k objektom cez pouzivatelov je kontrolovany nasledujucimi pravidlami:
  - **Jednoducha bezpecnostna vlastnost**: pouzivatel u moze citat objekt x jedine ak L(x) <= L(u)
  - ***-vlastnost**: pouzivatel u moze zapisovat (vytvarat, editovat, pridavat) objekt jedine ak L(u) <= L(x)
- **jednoducha bezpecnostna vlastnost** taktiez nazyvana „no read up“ pravidlo, zabranuje pouzivatelom citat objekty ktore maju bezpecnostny level vysssi ako pouzivatelia
- ***-vlastnost** je tiez nazyvana „no write down“ pravidlo. To znamena ze zabranuje propagacii informacii pouzivatelovi s nizsiim bezpecnostnym levelom



9.4.2 The Biba model

- ma podobnu strukturu ako BLP ale adresuje radsej integritu nez dovernost
- objektom a pouzivatelom su priradene integracne levely ktore formuju ciastocne poradie, podobne k BLP modeli.
- Integracne levely indikuju stupen spolahlivosti alebo preciznosti pre objekty a pouzivatelov radsej ako levely pre rozhodujuce o dvovernosti
  - napriklad, subor ulozeny na stroji, ktory je vo vysoko monitorovanom datacentre, mal by mat vyssi integritny level ako subor ulozeny v notebooku
  - vo vseobecnosti datecentrum je se menej kompromitovany nez nahodny notebook. Podobne u pouzivatelov, senior developer s rokmi skusenosti by mal mat vyssi integracny level ako novacik

9.4.2.1 Pravidla Biba modelu

- kontrola pristupu je presne opacna ako pri BLP. Biba nepovoluje citanie z nizsich levelov a zapisovanie do vyssich modelov
- ak oznacime integracny level pouzivatela u a I(x) oznacuje integracny level objektu, potom existuju taketo pravidla:
    - pouzivatel u vie citat objekt x jedine ak l(u) <= l(x)
    - pouzivatel u vie citat (vytvarat, editovat a pridavat do) objekt x jedine ak l(x) <= l(u)
- Preto Biba pravidla vyjadruju princip ze informacia moze prudit len dole z vyssieho integracneho levelu do nizsieho

9.4.3 The Low-Watermark model

- je rozsirenim Biba modelu ktory povoluje „no read down“ obmedzenie ale inac je presne rovnaky ako Biba model
- inac povedane, pouzivatelia s vyssimi integracnymi levelmi mozu citat objekty s nizsimi integracnymi levelmi
- pri citani objektu s nizsim integracnym levelom je pouzivatel degradovany taktiez na ten insty level

9.4.4 The Clark-Wilson model

- Preferuje radsej vykonavanie transakcii nez riesenie dovernosti a integrity
- Popisuje mechanizmy pre zaistenie toho ze integrita systemu je chranena napriec realizacie transakcie. Klucovymi komponentami CW modelu su:
    - **integracne obmedzenia**: vyjadruju vztahy uprostred objektov ktore musia byt pre system validne.
    - **Certifikacne metody**: overuju transakcie oproti integracnym obmedzeniam. Ked uz bol raz program pre transakciu je certifikovany, nemusi byt overovany pri kazdej dalsej transakcii
    - **Separacia povinnych pravidiel**: vedie pouzivatela ktory vykonava transakcie k ich certifikacii. Vo vseobecnosti kazdej transakcii je priradeny disjunkcny zoznam pouzivatelov ktory ich certifikuju a vykonavaju.

9.4.5 The Chinese Wall model (The Brewer and Nash model)

- je navrhnuty pre pouzitie v obchodnom sektore tak aby eliminoval moznosti konfliktu zaujmov
- k dosiahnutiu toho, model zoskupuje zdroje do tried konfliktov zaujmu
- model presadzuje obmedzenie ze kazdy pouzivatel moze jedine pristupovat k jednemu zdroju z kazdej triedy konfliktov zaujmu
- takato politika moze byt implementovana na pocitacovych systemoch k regulovaniu pouzivatelovho pristupu k citlivym datam

9.4.6 Riadenie pristupu na zaklade roli – Role-Based Access Control (RBAC)

- moze byt povazovany za evoluciu skupinovo zalozenych prav v suborovom systeme
- je definovany s respektom k organizacii, napriklad pre spolocnost: zoznam zdrojov, nejake dokumenty, tlacove sluzby, sietove sluzby a zoznam pouzivatelov ako su zamestnanci, zakaznici

9.4.6.1 Komponenty

- pouzivatel je entita ktora chce pristupovat k zdrojom v organizacii za ucelom vykonania ulohy. Zvycajne pouzivatelia su ludia, ale pouzivatel moze byt aj stroj alebo aplikacia.
- Rola je definovana ako zoznam pouzivatelov s podobnymi funkciami a zodpovednostou v ramci organizacii. Napriklad pre univerzitu mozu byt taketo roly: student, fakulta, dekan, zamestnanec. Vo vseobecnosti pouzivatelia mozu mat viacero roli.
- Prava popisuju povolene pristupove metody ku zdrioju

- Session pozostava z aktivacie pod zoznamu roli pouzivatela pre ucel vykonat urcitu ulohu

9.5 Penetration testing

Penetračný test (niekedy nazývaný ako ethical hacking) je test, ktorý odhalí formou pokusu o neoprávnený prienik do systémov slabiny a mieru zraniteľnosti organizácie.

Externý penetračný test preverí ochranu pred pokusom o prienik z Internetu. Interný penetračný test odkryje slabiny umožňujúce odcudzenie či poškodenie citlivých firemných dát pracovníkom spoločnosti.

White box

Black box

9.6 Secure storage:

- podla statistiky sa kazdych 53 sekund odcudzi notebook
- intel anti-theft
  - sledovanie podozrivych cinnosti ako napr
  - pocet nespravnych prihlaseni
  - uspesnost prihlasenia k serveru pre sledovanie kradeze v pravidelnych intervaloch
  - rezim kradeze je mozne spustit aj cez internet
- po spusteni rezimu kradeze dojde k vymazaniu sifrovacich klucov pre znemoznenie desifrovania dat na pevnom disku
- vsetky data na disku su sifrovane klucom ulozenym v pamati (napr. v BIOSe)
- pre rychlejsie odhalenie zlodeja je mozne mu ponechat pristup do operacneho systemu bez vstupu do citlivych udajov pre ucel ziskania jeho identity
  - ziskanim fotky cez webkameru
  - nahratim jeho hlasu pomocou mikrofonu
  - ziskanie polohy pomocou vstavaneho GPS ci podla znamej polohy SSID wifi routera
  - zaznamenavanie cinnosti zlodeja a nasledne posielanie majitelovi (nieco na sposob keyloggera)
- najprepracovanejsiu ochranu implementuje firma Lenovo do svojich notebookov

pouzitie TPM - trusted platform module

- hardwarovy kryptoprocesor

Šifrování disku

Aplikace pro úplné šifrování disku (jako například BitLocker Drive Encryption obsažená v operačních systémech Windows Vista Ultimate, Windows Vista Enterprise, Windows Server 2008, Windows 7 Enterprise a Windows 7 Ultimate od Microsoftu) používají tuto technologii ke chránění klíčů používaných k zašifrování pevných disků v počítači a poskytují ověření integrity pro důvěryhodnou zaváděcí cestu (například BIOS, boot sektor, atd.). Mnoho produktů třetích stran pro plné šifrování disku také podporují TPM čip.

Ochrana hesla

Přístup ke klíčům, datům nebo systému je často chráněn pomocí hesla. Pokud je ověřovací mechanismus implementován pouze v softwaru, přístup je náchylný ke "slovníkovým útokům". Protože je TPM implementován v jednoúčelovém hardwarovém modulu, tak byl vytvořen mechanismus zabraňující slovníkovým útokům, který efektivně zabraňuje hádání hesla a automatizovaným slovníkovým útokům, zatímco umožňuje uživateli dostatečně vysoký počet pokusů. S touto hardwarově založenou ochranou před slovníkovým útokem může uživatel volit kratší nebo slabší hesla, která se lépe pamatují. Bez této úrovně ochrany nabízí dostatečnou ochranu pouze hesla s velkou složitostí.

Distributed-Application Security

1. Database

Database Security

Zabezpecenie databaz je potrebne z toho dovodu, ze uchovavaju velke mnozstvo potencionale cennych informacii, ktore su casto tercom utokov. Utocnik sa snazi ziskat pristup k tymto udajom, a preto je dolezitenavrhnut dobre sposoby, ako ich zabezpecit.

Relacna databaza – velmi casty sposob ukladania informacii. Informacie su ukladane do sady tabuliek, medzi ktorymi mozu a nemusia byt vazby. Kazdy riadok tabulky reprezentuje zaznam (suvisiace informacie o entite (subjekte)). Kazdy stlpec predstavuje atribut, ktory dana entita moze vlastnit.

SQL query – vacsina DB pouziva jazyk SQL (Structured Query Language) pre realizaciu dotazov nad DB, s pouzitim prikazov (SELECT, INSERT, UPDATE, DELETE, UNION-kombinovanie zaznamov z viacerych dotazov) . Vyuziva podmienene prikazy pouzitim WHERE a zakladnych Boolean operacii ako AND, OR.

Two-Phase Commit – tento protokol vyuziva vacsina DB pri urcovani doverihodnosti otazky a pomaha DB dosiahnut integritu a dostupnost. Sklada sa z dvoch faz:

1. request phase (poziadavka) – identifikuju sa a oznacia vsetky casti DB, ktore maju byt zmene. Tato faza skonci uspesne, ak su oznacene vsetky casti, ktore maju byt zmene a protokol pokracuje 2. fazou. Alebo sa prerusi ak nedokazala oznacit vsetky casti (oznacil ich uz niekto iny, zlyhanie siete alebo systemu) a resetne vsetky pozadovane zmeny.
2. commit phase (potvrdenie) – DB sa zablokuje pred vykonavanim dalsich zmien a vykona zmeny identifikovane v prvej faze. Ak zbehne uspesne, tak odstrani vsetky vlajky identifikujuce pozadovane zmeny a zrusi zamok nad DB.
   Ak operacia zlyha, tak vrati vsetky vykonane zmeny a to tak, ze vrati DB do stavu tesne pred dokoncenim prvej fazy.

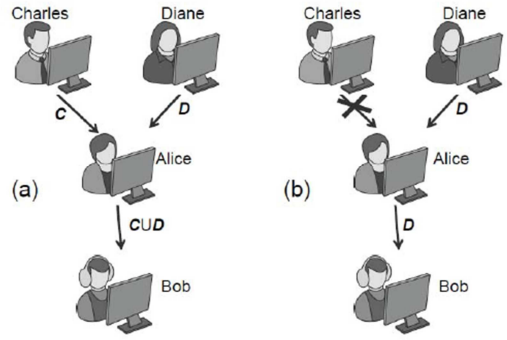
Database Access Control – sada ovladacich prvkov, ktore riadia pristup k databaze. Pojednavaju o :

- Pouzivatel by mal mat zakladne prava na vykonavanie jeho poziadaviek, a nic nad ich rames
- Pouzivatel moze mat rozne prava na zaklade uloh, ktore potrebuje vykonavat

Access Control Using SQL – SQL definuje framework riadenia pristupu, ktory sa bezne pouziva pri riadeni opravneni nad DB. Majitel tabulky ma vyhradne prava. Potom moze pridelit prava inym pouzivatelom a to : GRANT {action} ON {object} TO {person}, kde:

- {action} – typ prava – SELECT, INSERT, UPDATE … alebo ALL
- {object} – nazov tabulky, trigra, procedury…
- {person} – meno pouzivatela, ktoremu chce pridelit prava

Privilege Delegation - taktiez moze danemu pouzivatelovi umoznit , aby aj on mohol nad danym objektom pridelovat prava inym pouzivatelom. Staci ak na koniec prikazu napise WITH GRANT OPTION.
Privilege Revocation – prava moze aj zrusit pomocou prikazu REVOKE {action} ON {object}FROM {person}. V pripade ze dana osoba poskytla prava niekomu dalsiemu, tak sa automaticky odeberu aj tymto pouzivatelom.



Dovernost udajov
Okrem toho ze DB maju vhodne opatrenia pre riadenie pristupu k DB, musia byt prijate aj opatrenia, ktore ochrania citlive udaje pouzivatelov.
Using Cryptography – udaje sa v DB ulozia v zasifrovanej podobe pomocou nejakej kryptografickej funkcie. Desifrovaci kluc by mali poznat len opravnene osoby a nemal by byt ulozeny v DB.
Privaci Protection – aj vlastnik DB musi brat ohlad na ochranu citlivych udajov a dopad publikovania a poskytovania pristupu k takymto udajom, napr. pre ucely vyskumu musia byt taketo udaje ako mena, adresy atd. bud skrite, alebo nejak zamaskovane.

Inference attack – spociva v tom, ze viacery pouzivatelia mozu mat pristup len k casti tabulky. Napr A vidi len mena a ich identifikacne cisla, B vidi identifikacne cisla a platy. Ak sa A a B spoja tak si vedia lahko odvodit data a ku kazdemu menu priradit plat.
Ochrana:

- Cell suppression – vo zverejnenej verzii su niektore bunky odstranene
- Generalization – niektore hodnoty su nahradene vseobecnejsimi (napr vek 42 nahradi 40-50)
- Noise addition – prida nahodne hodnoty, alebo vo vsetkych zaznamoch nastavi na rovnaku

2. SpamCybercrime

SMTP (Simple Mail Transfer Protocol) – internetovy standard pre prenos internetovej posty (email) cez IP siet. Klient sa pripaja k servru naTCP porte 25. Klient posiela prikazy na server, ktory ich akceptuje, alebo zamietne. Mozu nastat bezpecnostne problemy:

- Odosielatel nie je doverihodny
- Sprava a hlavicka su odoslane ako plaintext
- Sprava nie je chranena
- Lahke dosaihnutie spoofingu

Email Spam – nevyziadana posta alebo hromadny email (UBE - unsolicited bulk email) je podmnozinou elektronickeho spamu obsahujuci takmer identicke spravy odoslane mnohym prijemcom emailu. Kliknutim na odkaz v spame moze byt pouzivatel presmerovany na povodne webove stranky, alebo na stranky so skodlivym obsahom. Spam tiez moze obsahovat malware.

Blacklisting

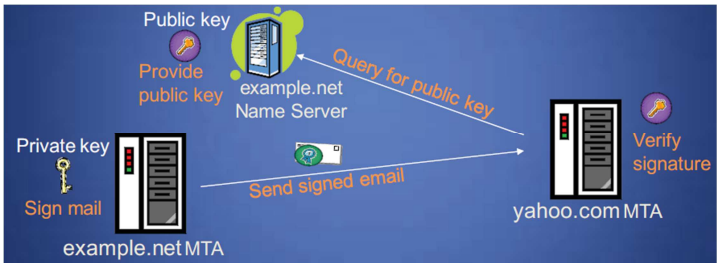
Spamhaus Black List (SBL) – real-time databaza v ktorej su zaznamenane IP adresy potvrdenych spam zdrojov.
Pred ukoncenim odovzdavania eliminuje okolo 10% spamu. Zapis a vyradenie z DB je formalny.

Graylisting – metoda brani pouzivatela e-mailu voci spamu. MTA (Mail Transfer Agent) automaticky zamietne kazdy email od zdroja, ktory neuznava. Pri doruceni noveho emailu si vytvori trojicu :

- IP adresa odosielatela
- Odosielatel podla SMTP obalky
- Prijemca podla SMTP obalky

Ked je trojica hotova, pozrie sa do DB, ci sa jedna o znami zdroj. Ak ano email sa doruci a prijme. Ak ide o neznamu trojicu, email odmietne a ohlasi odosielatelovi docasnu nedostupnost sluzby. Zaroven si trojicu zapise do DB a nastavi k nej casovac., ktory bude po danu dobu odmietat vsetky spravy s rovnakou trojicou.
Podla SMTP ma odosielatel po danu dobu opakovat odosielanie s urcitymi prestavkami. Po uplinuti casu sa prijimajuca MTA z databazi dozvie ze blokovanie skoncilo a ze sa vazne niekto pokusa dorucit spravu. Preto zmeni zaznam v DB, trojicu povoli a prijme spravu. Zaroven pre trojicu nastavi velku zivostnost (>mesiac), ktora sa pri kazdej uspesne dorucenej sprave predlzi.
To znamena ze email od daneho zdroja sa oneskori len raz – pri prvom doruceni.

DomainKeys Identified Mail (DKIM) – mail server odosielatela podpise spravu, aby sme mohli identifikovat jeho domenu. Verejny kluc je dostupny v DNS zaznamoch. Pouziva sa s inymi spam filtering methods.



Cybecrime (pocitacova kriminalita) – akykolvek trestny cin, ktory bol spachany s poucitim pocitaca, siete, alebo hardveroveho zariadenia. Pocitac alebo zariadenie moze byt ako zastupcom, sprostredkovatelom alebo cielom trestneho cinu.

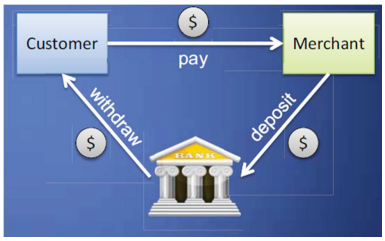
Payment system

Electronic Payment scheme (EPS)– platobnym nastrojom je elektronicka minca, ktora ma fixnu hodnotu a moze byt vymenena za tradicky penazny nastroj. Ucastnikmi su :

- Payer (customer) – platitel
- Payee (merchant) – prijemca
- Bank

Transakcia – transakcia v EPS tradicne zahrna:

- Withdrawal – vyber zakaznikom z banky
- Payment – platba prijemcovi
- Deposit – ulozenie do banky



- Online schema – banka sa zucastnuje na transakcii
- Offline schema – banka sa nezucastnuje na transakcii

Payment with digital signature – v case vyberu su mince digitalne oznacene bankou. Tym padom si prijemca vie overit podpis banky a samozrejme aj banka prijima taketo mince pri vklade. No naskyta sa tu otazka bezpecnosti a sukromia – zakaznik moze mincu skopirovat a tak dvakrat utracat, banka sa dozvie o kazdej transakcii (riesenie – **blind signature** (podpis bez toho aby som o tom vedel), banka sa pri vybere podpise, prijemca si to vie overit a urobit vklad, banka nevie spojit peniaze s platitelom).

DRM

- Subor technyk, ktore obmedzuju pouzivanie digitalnych medii (napr. nemozes kopirovat CD, stiahnuty film si mozes prehrat len raz, …)
- Je pouzivany firmami ako AT&T, apple, amazon, Microsoft, EA, sony
- v r.1998 vstupil do platnosti zakon umoznujuci pouzitie DRM pre pravne ucely
- Druhy implementacie
    - limitovanie poctu aktivacii - pri kazdej instalacii napr. hry sa pocas toho na serveri overi pocet instalacii a na zaklade toho sa instalacia povoli alebo zakaze akciu
    - persistentna autentifikacia - overovanie pocas behu programu
    - tampering - neinvazivny sposob kontroly kedy pri zisteni neoriginality zacne program (najcastejsie hra) vykazovat nespravne fungovanie, napr. zbrat v hre zacne nepresne mierit, pomaly dobijat atd...
    - limitovanie funkcionality - tento sposob vyuziva kooperaciu poskytovatelov obsahu s vyrobcom hardwaru. Typickym prikladom je pouzivanie MP3 suborov produktami Microsoftu. Zakupena skladba moze obsahovat informacie o mnozstve prehratia, casovej platnosti prehravania, kopirovanie na dalsie zariadenia podporujuce DRM
    - watermark - osobitym druhom implementacie je tzv. watermark, ktorehou ulohou nie je obmedzovat uzivatela ale mat moznost identifikovat uzivatela, ktory sposobuje nelegalne sirenie obsahu
- k moznosti prehravat DRM chraneny obsah musi byt splnena podmienka celeho retazca, tzn. pri prehravani hudby musia implementovat DRM ochranu softwarovy prehravac, hardwarove zariadenie, aj konektor k prenosu - priklad: DVD prehravac, HDMI rozhranie aj TV musi implementovat ochranu HDCP. Napr. prehravanie hudby cez SPDIF alebo filmu cez DVI nie je mozne nakolko prenasa len surovy zvuk a obraz bez podpory DRM
- v sucasnosti DRM sposobuje v pripade filmov a pesniciek problemy, kedy ochrana pred kopirovanim je kompromis medzi bezpecnostou pred piratstvom a moznostou rozumne nakladat so zakupenym dielom, z toho dovodu mnohe firmy odstupuju z DRM ochrany a prechadzaju na Watermark (vid. slovenske knihkupectva)

Dmitry Skylarov – pracoval pre Elcomsoft v Rusku. Vytvoril produkt, ktory vie previest chraneny Adobe eBook na nechraneny PDF subor.
Profesor Edward Felten of Princeton – dokazal odstranit vodotlac a vidal o tom knihu

Analog hole – kopirovanie media pocas jeho prehravania, napr v radiu ide nejaka pesnicka, tak si ju nahram.

CD/DVD protection – vacsina je ich chranena tak, aby sa nedali kopirovat. Avsak CD nie su neznicitelne a su potrebne zalohy. Preto je vo vascine krajin legalne robit si zalohy CD, ale predavat taketo medium je uz nelegalne. Takmer kayda sifrovacia technika bola prelomena.

SafeDisc V1 a V2 – ochrana proti kopirovaniu vytvorena MacroVision. Pouzivane pre hry zacinajuce v roku 1999. Na originalnom disku boli zamerne chyby. Lahke obist pouzitim specialnych nastrojov pre kopirovanie CD v pomere 1:1.

CSS: Content Scrambling System – sluzi pre ochranu DVD. Obsah videa je zasifrovany pomocou disc key. Najprv sa disc key desifruje asi 400x vzdy s pouzitim ineho player key. Video sa desifruje za behu, pocas prehravania.

DeCSS – zdrojovy kod vytvoreny v roku 1999 Jon Johansen. Desifroval CSS a umoznil kopirovanie na pevny disk.

Bezpecnost, bezpecnostna politika, meranie bezpecnosti

Pocitacova bezpecnost

- zaobera sa zabezpecenim informacii v pocitacoch.

Bezpecnostna politika

Definovany zoznam pravidiel obsahujuce:

- **Osoby (Subjekty)**: agenti ktori posobia v ramci systemu, ktori mozu byt definovani ako jednotlivci (prezident, CEO, CFO), role alebo skupiny jednotlivcov (pouzivatelia, administratori, generali, majori, dekani, manazeri, asistenti, utocnici, hostia) v ramci organizacie.
- **Objekty:** informacne (dokumenty, subory, databazy) a vypoctove zdroje (servery, softver) pre ktore je bezpecnostna politika navrhnuta tak aby ich chranila a spravovala
- **Akcie:** veci, ktore subjekty mozu alebo nemozu robit vzhladom k objektom. Napiklad: citanie alebo pisanie dokumentov, updatovanie softveru, pristupovat k obsahu databazy
- **Prava:** mapovanie medzi subjektmi, akciami a objektmi ktore jasne hovoria o tom kto aky druh akcie moze alebo nemoze vykonavat
- **Ochrany:** specificke bezpecnostne vlastnosti alebo pravidla ktore su obsiahnute v politike, ktore pomahaju dosiahnut bezpecnostne ciele ako dovernost, integritu, dostupnost alebo anonymitu

???

Identifikacia a autentifikacia

Sluzia ako nastroje pri kontrole pristupu (rozlisovanie medzi pousivatelmi ktori maju/nemaju pristup k informaciam).
Napr. pri pristupe k osobnemu pocitacu je potrebne zadat username a password.

- Username (meno pouzivatela) – identifikacia pristupovanej entity (pouzivatela), ale to je len trvenie pouzivatela ze je kto je, musi to nejako dokazat – autentifikovat sa napr heslom
- Password (heslo pouzivatela) – autentifikacia = overenie/potvrdenie pravosti identity alebo roly pouzivatela v systeme.

Ked pouzivatel spojil svoju identitu (username) s tajnou informaciou (heslom) overi sa jeho identita a to tak ze sa zadane udaje porovnaju s udajmi ulozenymi na pristupovanom zariadeni (napr. password file).

Identifikacia a autentifikacia moze byt neuspesna:

- Obmedzeny pocet prihlaseni
- Zablokovanie uctu
- Upozornenie na nelegalnost neopravneneho pristupu a pravne postihy

Problemy s heslom:

- Zabudnute hesla (ukladanie do zapecatenych obalok do trezoru)
- Hadanie hesla (volba silneho hesla)
- Odchytenie, odpozorovanie hesla (technicke, technologicke opatrenia)
- Umyselne/neumyselne vyzradenie hesla

Integrita OS, prerusenia

- Aké bezpečnostné mechanizmy by mali byť obsiahnuté v bezpečnostnom kerneli? Predpokladajme operačný systém, ktorý by mohol presadiť všetky politiky prístupu používateľa. Neautorizovaný prístup k zdrojom je nemožný, pokiaľ operačný systém funguje tak, ako bolo zamýšľané, aby fungoval. Toto je však „pokyn“ pre útočníka, ktorý sa snaží odstaviť ochranné mechanizmy operačného systému tým, že operačný systém modifikuje. Teraz je to primárne ochrana integrity operačného systému, aj keď iniciálne išlo o ochranu dôvernosti. Operačný systém je nielen arbiter žiadostí o prístup, ale sám je i objekt riadenia prístupu. Novou bezpečnostnou politikou je:
  - **Používateľ nesmie byť schopný modifikovať operačný systém.**.
- Vyššie uvedené je generická bezpečnostná politika, požadujúca silnú a efektívnu podporu. Aby sme si skomplikovali život, musíme uviesť dve protirečivé požiadavky na operačný systém vo vzťahu k používateľovi:
  - **Používateľ by mal byť schopný používať (zavolať) operačný systém.**
  - **Používateľ by nemal byť schopný zneužiť operačný systém.**
- Na dosiahnutie týchto cieľov sa spoločne používajú dva koncepty: **stavové informácie a riadené zavolanie** (obmedzené privilégiá). Tieto dva koncepty je možné použiť v ľubovoľnej vrstve počítačového systému, či už to je aplikačný softvér, operačný systém alebo hardvér. Mechanizmy týchto konceptov však môže útočník zablokovať, pokiaľ sa dostane do nižších vrstiev operačného systému.
- **Prvý koncept** – predpoklad, že operačný systém je schopný sa sám chrániť pred používateľmi je, že má schopnosť rozlíšiť medzi výpočtami „v zastúpení“ operačného systému a „v zastúpení“ používateľa.
- Na rozlíšenie týchto stavov sa používa stavový príznak, ktorý umožňuje operačnému systému pracovať v rôznych režimoch. Napríklad procesor 80x86 má dva stavové bity, ktoré indikujú štyri režimy. Operačný systém Unix rozlišuje medzi režimom supervízor (root) a používateľ.
- K čomu je rozlíšenie režimov dobré? Napríklad na zastavenie používateľa pred priamym zápisom do pamäti a korupciou logickej štruktúry súboru. Operačný systém môže povoliť prístup zápisu iba ak procesor je v režime supervízor.
- **Druhý koncept** – pokračujeme v predchádzajúcom príklade. Používateľ chce vykonať operáciu vyžadujúcu režim supervízora, napríklad zapísať do pamäti. Aby sa procesor vysporiadal s touto žiadosťou, musí prepnúť medzi režimami. Ale ako toto prepnutie urobí? Jednoducho zmenou stavového bitu do režimu supervízora. Toto prepnutie mu umožní získať všetky privilégiá režimu. To znamená, že je žiaduce, aby iba systém vykonával preddefinovanú množinu operácií v režime supervízor a potom sa navrátil späť do používateľského režimu predtým než odovzdá riadenie používateľovi späť. Tomuto procesu sa hovorí **riadené vyvolanie**.

Bezpecny program

Poziadavky na bezpecnost DB systemu

Database Security

Zabezpecenie databaz je potrebne z toho dovodu, ze uchovavaju velke mnozstvo potencionale cennych informacii, ktore su casto tercom utokov. Utocnik sa snazi ziskat pristup k tymto udajom, a preto je dolezitenavrhnut dobre sposoby, ako ich zabezpecit.

Relacna databaza – velmi casty sposob ukladania informacii. Informacie su ukladane do sady tabuliek, medzi ktorymi mozu a nemusia byt vazby. Kazdy riadok tabulky reprezentuje zaznam (suvisiace informacie o entite (subjekte)). Kazdy stlpec predstavuje atribut, ktory dana entita moze vlastnit.

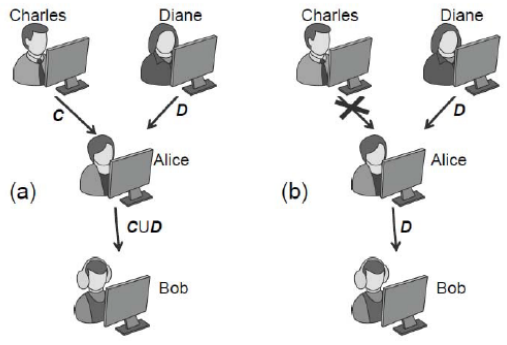
SQL query – vacsina DB pouziva jazyk SQL (Structured Query Language) pre realizaciu dotazov nad DB, s pouzitim prikazov (SELECT, INSERT, UPDATE, DELETE, UNION- kombinovanie zaznamov z viacerych dotazov) . Vyuziva podmienene prikazy pouzitim WHERE a zakladnych Boolean operacii ako AND, OR.
Database Access Control – sada ovladacich prvkov, ktore riadia pristup k databaze. Pojednavaju o :

- Pouzivatel by mal mat zakladne prava na vykonavanie jeho poziadaviek, a nic nad ich rames
- Pouzivatel moze mat rozne prava na zaklade uloh, ktore potrebuje vykonavat

Access Control Using SQL – SQL definuje framework riadenia pristupu, ktory sa bezne pouziva pri riadeni opravneni nad DB. Majitel tabulky ma vyhradne prava. Potom moze pridelit prava inym pouzivatelom a to : GRANT {action} ON {object} TO {person}, kde:

- {action} – typ prava – SELECT, INSERT, UPDATE … alebo ALL
- {object} – nazov tabulky, trigra, procedury…
- {person} – meno pouzivatela, ktoremu chce pridelit prava

Privilege Delegation - taktiez moze danemu pouzivatelovi umoznit , aby aj on mohol nad danym objektom pridelovat prava inym pouzivatelom. Staci ak na koniec prikazu napise WITH GRANT OPTION.
Privilege Revocation – prava moze aj zrusit pomocou prikazu REVOKE {action} ON {object}FROM {person}. V pripade ze dana osoba poskytla prava niekomu dalsiemu, tak sa automaticky odeberu aj tymto pouzivatelom.



Dovernost udajov
Okrem toho ze DB maju vhodne opatrenia pre riadenie pristupu k DB, musia byt prijate aj opatrenia, ktore ochrania citlive udaje pouzivatelov.
Using Cryptography – udaje sa v DB ulozia v zasifrovanej podobe pomocou nejakej kryptografickej funkcie. Desifrovaci kluc by mali poznat len opravnene osoby a nemal by byt ulozeny v DB.
Privaci Protection – aj vlastnik DB musi brat ohlad na ochranu citlivych udajov a dopad publikovania a poskytovania pristupu k takymto udajom, napr. pre ucely vyskumu musia byt taketo udaje ako mena, adresy atd. bud skrite, alebo nejak zamaskovane.

Inference attack – spociva v tom, ze viacery pouzivatelia mozu mat pristup len k casti tabulky. Napr A vidi len mena a ich identifikacne cisla, B vidi identifikacne cisla a platy. Ak sa A a B spoja tak si vedia lahko odvodit data a ku kazdemu menu priradit plat.

Ochrana:

- Cell suppression – vo zverejnenej verzii su niektore bunky odstranene
- Generalization – niektore hodnoty su nahradene vseobecnejsimi (napr vek 42 nahradi 40-50)
- Noise addition – prida nahodne hodnoty, alebo vo vsetkych zaznamoch nastavi na rovnaku

Sandbox

Výsledkom prvotnej snahy zabezpečenia, ktoré by nepripustilo poškodenie klientského počítača spustením kódu z Internetu je model zabezpečenia nazývaný SANDBOX (krabica z piesku).

Model pieskoviska je bezpečnostné prostredie okolo systému alebo aplikácie, ktoré je postavené na zákaze potenciálne nebezpečných činností (prístup k súborom, sieťovým prostriedkom a pod.).

Je pouzivany antivirusovymi programami pri dynamickej analyze podozriveho kodu. Pri podozrivych suboroch sa antivirusovy program opyta, ci ma byt subor spusteny v tomto sandboxe. Jedna sa o akusi "bublinu" mimo ukladania na disk. Pri vykonavani kodu sa kontoluje zmena suborov, zmeny v registroch, procesy, vlakna a sietove porty.

Rizika bezpecnosti, hrozby, zranitelnost

Hrozby a utoky

Odpocuvanie

Odposluch informácií určených pre niekoho iného v priebehu ich prenosu cez komunikačný kanál. Najvacsie riziko odpocuvania je v nezabezpecenych lokalnych pocitacovych sietach.

Modifikacia informacii

Neautorizovana modifikacia informacie, tzn. utocnik odchiti informaciu odosielatela, upravi ju a posle ju dalej prijimatelovi.

Denial-of-service (DOS) – Odmientnutie sluzby

Zahltenie cieľa nepotrebnými požiadavkami natoľko, že nestíha obsluhovať bežných používateľov (služby). Prikladom je napr. Spam, ktory zahlti emailovy server natolko, ze nakoniec padne, tzn. nebude stihat spracovavat tolko poziadaviek naraz.

Maskarada

Pouzivanie identity cudzej osoby ziskanych osobnych udajov.

Odmietnutie (Repudiation)

Tento utok vznika v tedy ak system nezaznamenava co pouzivatelia v dannom systeme vykonavaju alebo ak vznikne modifikacia napr. logovacich suborov s tym ze utocnik napr. urobi v systeme nejake zmeny a nasledne upravi logovacie subory tak, ze danne zmeny urobil iny pouzivatel.

Correlation a traceback

Identifikacia identity napr. pouzivatela na zaklade pospajanych ciastkovych informacii z viacerych zdrojov.

Pouzivatel nieco vie, nieoc ma, niekto je, nieco robi, niekde je

1.1.Autentifikácia (Authentication)

Určenie identity na základe kombinácie

- Osoba niečo vlastní (čipová karta ...)
- Osoba niečo vie (napr. heslo)
- Osoba niečim je (človek s odtlačkami prstvov)

Mandatory ACL (MAC), Discretionary ACL (DAC)

- **ACL (Access Control List):** pre zdroj (napr. subor alebo priecinok) je usporiadany zoznam 0 alebo viac **ACEs**
    - **ACL prikazy:**
        - **getfacl:** precitanie ACLs
        - **setfacl:** nastavenie ACLs
- **ACE**: - urcuje ktory konkretny zoznam pristupov (citanie, spustanie, zapisovanie) k zdrojom je povoleny alebo zakazany pre pouzivatela alebo skupinu
- **Napriklad**:
    - Bob; Read;Allow
    - tAs; Read; Allow
    - TWD; Read, Write; Allow
    - Bob; Write; Deny

Discretionary Access Control (DAC)

- pouzivatel moze chranit co vlastni
    - vlastnik moze udelovat pristup ostatnym
    - vlastnik moze definovat typ pristupu (citanie, zapisovanie, spustanie) ostatnym
- je standartny model pouzivany v operacnych systemoch

Mandatory Access Control (MAC)

- alternativny model
- viacurovnove levely pre bezpecnost pouzivatelov a dokumentov

Chyba Time-of-check , Time-of-use

In software development, time of check to time of use (TOCTTOU or TOCTOU, pronounced "TOCK too") is a class of software bug caused by changes in a system between the checking of a condition (such as a security credential) and the use of the results of that check. This is one example of a race condition.

A simple example is as follows: Consider a Web application that allows a user to edit pages, and also allows administrators to lock pages to prevent editing. A user requests to edit a page, getting a form by which he can alter its content. Before the user submits the form, an administrator locks the page, which should prevent editing. However, since the user has already begun editing, when he submits the form, his edits are accepted. When the user began editing, his authorization was checked, and he was indeed allowed to edit. However, the authorization was used later, after he should no longer have been allowed.

TOCTTOU race conditions are most common in Unix between operations on the file system, but can occur in other contexts, including local sockets and improper use of database transactions. In the early 90's, the mail utility of BSD 4.3 UNIX had an exploitable race condition for temporary file because it used mktemp() C library function.[1] Early versions of OpenSSH had an exploitable race condition for Unix domain sockets.[2]

Redundancia, detekcia chyb

Integrita obsahu, kontrola parity

Integrita

Ak komunikuju 2 alebo viac strany navzajom pomocu nejakoho komunikacneho media, tak cheme mat istotu ze tie prenasane informacie su v konzistentnom tvare, tzn. ze ak niekto nieco poslal, tak prijamca dostal tuto spravu v nezmenenom tvare. Cize aby v komunikacii nedoslo k nejakej modifikacii tejto spravy, ci uz nejakym utocnikom alebo nejakym inym sposobom (nezavinena modifikacia, napr. zly prenosovy kanal, rusenie wifi siete a pod). Pokial dojde k nezavinenemu poruseniu integrity, tak na to existuju korekcne kody, ktore vyziadaju nanovo komunikaciu alebo komunikaciu zrusia.

Nastroje:

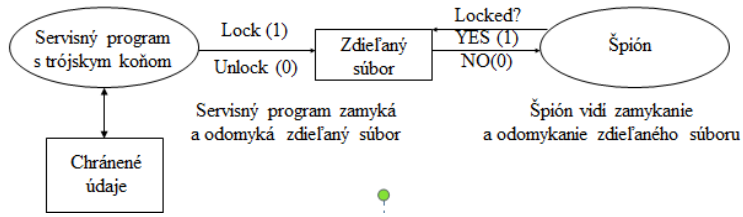
- **Zalohovanie**: periodicka zaloha dat. Ak sa nieco stane s povodnymi datami, mame k dispoziciu zalohovane data
- **Kontrolne sum**y: funkcia ktora vypocita ciselnu hodnotu (kontrolnu sumu) zo suboru, tzn. ak bol subor co i len trochu upraveny (staci aby mal otoceny len 1 bit) bude kontrolna suma odlisna.
- **Datova korekcia kodov**: mechanizmus na ukladanie dat v takej podobe, ze male zmeny mozu byt jednoducho detekovane a automaticky opravene

Viacvrstvovy navrh jadra OS

Komunikativny filter vo viacurovnovych DB

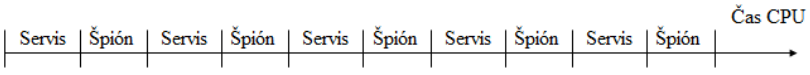
Hrozby – skryte prenosove kanaly

- Programy sú vybavené nástrojmi na **posielanie dôverných informácií** osobám, ktoré by ich nemali dostávať. Informácie sú posielané nebadane (steganografia) s ostatnými informáciami, perfektne legálnymi:
- Aj keď programátor spravidla po odovzdaní systému do prevádzky nemá dôvod, aby on sám pristupoval k ostrým údajom, má vždy možnosť naprogramovať a do programu skryť kanály, cez ktoré sa môže dostať k ostrým údajom v prevádzke, napríklad pomocou skrytých významov vo výpise.
- Ak napr. má číslo viacej významných miest ako je potrebné , môže sa používať viac typov hlášok (TOTAL/TOTALS), absencia alebo existencia prázdnych riadkov po istej hláške, výpis môže mať rôzny počet riadkov na stránke, použitie . namiesto :, použitie číslic na nevýznamnom mieste.
- Pamäťové kanály sú skryté kanály, ktoré indikujú existenciu/neexistenciu objektov v pamäti (v hlavnej alebo na disku). Jednoduchým príkladom je kanál uzamykania súborov vo viacpoužívateľskom prostredí.
- V takýchto prípadoch je prístup k súboru výlučný (objekt vlastní jeden používateľ, iba jeden používateľský proces môže k objektu pristúpiť). Operačný systém zabezpečuje jeho výlučné používanie.
- V prostredí sa vykonáva aplikácia s chránenými údajmi. K údajom pristupuje servisný program. V viacpoužívateľskom prostredí sa vykonáva tiež **program špión**.
- Predpokladáme, že **servisný program obsahuje trójskeho koňa**, ktorý môže uzamykať a odomykať dohodnutý zdieľaný súbor a tým indikovať obsah chránených údajov.
- Špión si uchová odpoveď a prenesie ju útočníkovi.

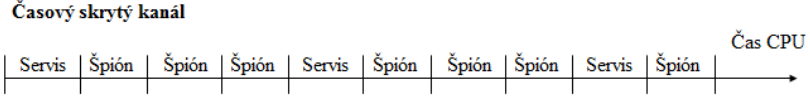


- ❑ Podobnú schému skrytého kanálu je možné realizovať prostredníctvom pamäťového priestoru na disku.
  - o Servisný program s trójskym koňom pravidelne vytvára a ruší veľký súbor na disku (po jeho vytvorení je diskový priestor vyčerpaný)
  - o Špión sa tiež pokúša vytvoriť súbor na disku. Ak sa mu to podarí, servisný program indikuje hodnotu 0, ak sa mu to nepodarí, servisný program indikuje 1.
- ❑ Ďalším príkladom pamäťového skrytého kanála je zisťovanie stavu servera na generovanie jedinečných identifikátorov (mená dočasných súborov, príznaky na označenie správ).
  - o Rôzne procesy môžu žiadať nasledujúci identifikátor zo servera.
  - o Podľa následnosti identifikačných čísel možno usudzovať na postupnosť vytvárania súborov, správy, resp. či aj servisný proces si vyžiadal identifikátor.
  - o Tento fakt môže byť využitý na vytvorenie skrytého kanála.
- ❑ Časové kanály – prenášajú informácie rýchlosťou, pri akých sa udalosti odohrávajú.
  - o V skutočnosti je čas spoločným prostriedkom, ktorý zdieľajú časové skryté kanály.
  - o V multiprogramovom prostredí dva procesy (servisný s trójskym koňom a špión) zdieľajú čas procesora.
  - o Ak je procesu pridelený čas CPU a proces nie je pripravený, potom tento čas odmietne.
  - o Servisný proces odmietne čas CPU – signalizácia 1, servisný proces neodmietne čas CPU – signalizácia 0

Normálne plánovanie



Časový skrytý kanál



Servisný proces s trójskym koňom signalizuje špiónovi postupnosť 01010

Utoky DOS

Denial-of-service (DOS) – Odmientnutie sluzby

Zahltenie cieľa nepotrebnými požiadavkami natoľko, že nestíha obsluhovať bežných používateľov (služby). Prikladom je napr. Spam, ktory zahlti emailovy server natolko, ze nakoniec padne, tzn. nebude stihat spracovavat tolko poziadaviek naraz.

Typy:

ICMP Floods:

- **Smurf attack:** spociva v chybnej konfiguracii systemu, ktory dovoli rozosielanie packetov vsetkym pocitacom zapojenych v sieti cez Broadcast adresu. Potom staci aby odoslany paket mal dostatocnu velkost, aby nebol odfiltrovany a vsetky pocitace v sieti ho musia prijat a spracovat (zahodit)
- **Ping-flood:** zahlcuje cielovy ocitac ziadostami o ping odozvu
- **SYN flood:** Útočník pošle cieľovému počítaču postupnosť paketov s príznakom SYN ale potom už ďalej neodpovedá. Týmto útokom môže útočník zaplniť frontu obete, ktorá je určená na začaté spojenia. V prípade zaplnenia fronty, obete už nemôžu prijímať nové spojenia.

Teardrop utok:

- tento typ utoku zahrnuje zasielanie IP fragmentu s prekryvajucim sa prilis velkym mnozstvom dat na cielovy pocitac. Chyba v TCP/IP pri preskladavani takehoto paketu moze na starsich OS sposobit ich pad.

Peer-to-peer utok:

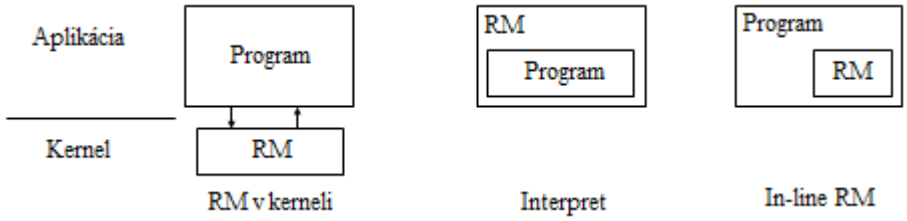
- na P2P klienta sa pripoji v jednom case masivne mnozstvo ludi a to moze sposobit az pad systemu

Nukes:

- su specialne pakety k zniceniu cieloveho pocitaca. Totiz nie vzdy je potrebne sserver zahltit, obcas sa v nom vyskytne chyba ktora zapricini pad uz pri spracovani jedneho paketu

Referencny monitor (rm) a jeho umiestnenie

- V počítačovej bezpečnosti sú tri fundamentálne koncepty, ktorú sú dostatočne blízke, aby spôsobovali pomýlenia, ale zaslúžia si, aby boli samostatne vyčleňované.
  - **Referenčný monitor** – koncept riadenia prístupu, ktorý odpovedá abstraktnému stroju sprostredkujúcemu všetky prístupy subjektov k objektom
  - **Bezpečnostný kernel** – hardvérové, firmvérové alebo softvérové elementy dôveryhodnej výpočtovej báze (Trusted Computing Base – TCB), ktoré implementujú koncept referenčného monitora. Musí sprostredkovať všetky prístupy, musí byť chránený pred modifikáciou a musí byť verifikovaný ako korektný
  - **Dôveryhodná výpočtová báza** – všetky ochranné mechanizmy v počítačovom systéme, vrátane hardvéru, firmvéru a softvéru. Ich kombinácia je zodpovedná za presadzovanie bezpečnostnej politiky. TCB pozostáva z jedného alebo viacerých komponentov, ktoré spolu presadzujú jednotnú bezpečnostnú politiku v celom systéme alebo produkte. Schopnosť TCB korektne presadzovať bezpečnostnú politiku závisí výlučne na mechanizmoch v rámci TCB a na správnych vstupoch parametrov od systémových administrátorov (napríklad oprávnenia používateľa) majúcich vzťah k bezpečnosti.
- Referenčný monitor je abstraktný koncept, bezpečnostný kernel je jeho implementácia a TCB obsahuje bezpečnostný kernel medzi ostatnými ochrannými mechanizmami. Kritické požiadavky na implementáciu referenčného monitoru sú:
  - Mechanizmus validácie referencie (prístupe) musí byť odolný proti útokom
  - Mechanizmus validácie referencie (prístupe) musí byť vždy zavolaný (úplné sprostredkovanie)
  - Mechanizmus validácie referencie (prístupe) musí byť dostatočne malý, aby mohol byť analyzovaný a testovaný, s cieľom preverenia jeho korektnosti.
- Principiálne môže byť referenčný monitor umiestnený kdekoľvek v architektúre počítačového systému. V príkladoch možných rozhodnutí návrhárov možno nájsť:
  - **V hardvéri** – riadenie prístupu do pamäti, privilégiá procesov
  - **V jadre operačného systému** – hypervízor je virtuálny stroj, ktorý presne emuluje hostový počítač, na ktorom sa vykonáva. Môže byť použitý pre oddelených používateľov alebo aplikácie. Každému poskytuje separátny virtuálny stroj.
  - **V operačnom systéme** – príkladom sú operačné systému Unix a Windows.
  - **Vo vrstvách služieb** – príkladom môže byť riadenie prístupu v DBMS, Java Virtual Machine, CORBA middleware architektúra.
  - **V aplikácii** – vývojári aplikácií s veľmi špecifickými bezpečnostnými požiadavkami sa môžu rozhodnúť uprednostniť zahrnúť bezpečnostné kontroly do aplikačného kódu pred zavolaním bezpečnostných služieb z nižšej systémovej vrstvy.
- V aplikácii môže byť referenčný monitor zabezpečený **nižšou systémovou vrstvou**, čo je typický vzor riadenia prístupu v operačnom systéme. Aplikačný program žiada prístup pre ochranu zdrojov. Referenčný monitor je časťou kernelu operačného systému a sprostredkováva všetky žiadosti o prístup. Riadenie prístupu v CORBE sleduje tento istý vzor.
- Program môže byť vykonávaný interpretom. Interpret sprostredkuje všetky žiadosti programu o prístup. Java zjednodušuje tento prístup a program je umiestnený v referenčnom monitore.
- V treťom prípade je program prepísaný tak, aby obsahoval kontroly riadenia prístupu. Príkladom sú in-line referenčné monitory.



Pocitacova bezpecnost, dovernost, integrita, dostupnost, (audit, nepopieratelnost)

Pocitacova bezpecnost

- zaobera sa zabezpecenim informacii v pocitacoch.

Integrita

Ak komunikuju 2 alebo viac strany navzajom pomocu nejakoho komunikacneho media, tak cheme mat istotu ze tie prenasane informacie su v konzistentnom tvare, tzn. ze ak niekto nieco poslal, tak prijamca dostal tuto spravu v nezmenenom tvare. Cize aby v komunikacii nedoslo k nejakej modifikacii tejto spravy, ci uz nejakym utocnikom alebo nejakym inym sposobom (nezavinena modifikacia, napr. zly prenosovy kanal, rusenie wifi siete a pod). Pokial dojde k nezavinenemu poruseniu integrity, tak na to existuju korekcne kody, ktore vyziadaju nanovo komunikaciu alebo komunikaciu zrusia.

Nastroje:

- **Zalohovanie**: periodicka zaloha dat. Ak sa nieco stane s povodnymi datami, mame k dispoziciu zalohovane data
- **Kontrolne sum**y: funkcia ktora vypocita ciselnu hodnotu (kontrolnu sumu) zo suboru, tzn. ak bol subor co i len trochu upraveny (staci aby mal otoceny len 1 bit) bude kontrolna suma odlisna.
- **Datova korekcia kodov**: mechanizmus na ukladanie dat v takej podobe, ze male zmeny mozu byt jednoducho detekovane a automaticky opravene

Dostupnost

Pokial ma pouzivatel pristup k urcitej sluzbe alebo zdrojom, tak mal by mat k tomu pristup vzdy ked o to poziada.

Nastroje na zabezpecenie dostupnosti:

- **Fyzicka ochrana**: zabezpecenie zariadeni, budovy, atd. tak aby nemohli byt narusene tretou osobou.
- **Vypoctove redudancie**: automaticke nahradzanie zdrojov, pokial jeden z nich vypadne. (napr. pri diskoch ak su zapojene v RAID0, RAID5)

Dalsie koncepty bezpecnosti

Istota (Dovernost)

Manazment doverihodnosti (dovernost medzi systemom a pouzivatelom) je na takom stupni ze system a pouzivatel si navzajom doveruju.

Zavisy na:

- **Bezpecnostnej politike**: specifikuje spravanie ludi alebo systemov vramci seba a vramci ostatnych (povinna alebo diskretna kontrola pristupu)
- **Bezpecnostne prava**: opisuju spravanie ze co pouzivatel moze vykonat a co nemoze
- **Bezpecnostna ochrana**: opisuje mechanizmy na zaistenie prav a politiky na zaklade identity pouzivatela

Autenticnost

Stanovenie, ze udaje, postupy a prava vydane osobami alebo systemom su prave.

Hlavnym nastrojom na zaistenie autenticnosti je **digitalny podpis (DP)**.

DP: - je analogicky rucnemu podpisu, ktory sluzi ako dokaz autorstva, resp. suhlasu s obsahom dokumentu.

- je to urcita datova struktura, ktora je zavisla na dokumente, vznika hasovanim tohto dokumentu a tento kod je zasifrovany sukromnym klucom, ktory je jednoznacnym vlastnictvom vlastnika dokumentu.

Nie je mozne dosiahnut 100% autenticnosti udajov.

Anonymita

Vlastnost ze urcite zaznamy alebo transakcie nepripadnu ku ziadnemu jednotlivcovy.

Nastroje na zabezpecenie anonymity:

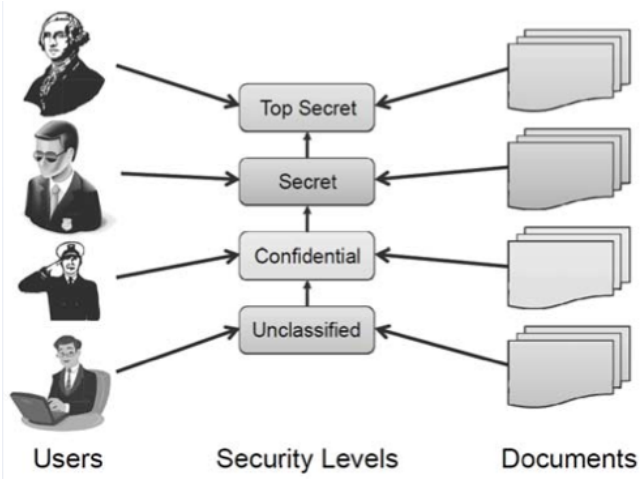
- **Agregacia**: kombinacia dat od viacerych pouzivatelov, takze zverejnene ciastky alebo priemery nemozu byt spojene so ziadnym konkretnym pouzivatelom
- **Mixacia**: agregovanie informacii z viacerych stran a spajanie ich do zloziek, ktore sa nedaju rozlozit.
- **Proxy**: doverihodni agenti, ktori nahradzaju skutocnu identitu pouzivatela, vacsinou vo vacsich organizaciach sa jedna o nejake systemy ktore nahradzaju identitu skutocneho pouzivatela
- **Pseudonym**: fiktivna identita pouzivatela, ktory predstiera identitu

Operacie pristupu (Bell-LaPadula model)

Modely kontroly pristupu

The Bell-La Padula (BLP) model

- je klasicky model povinnej kontroly pristupu (MAC) pre zabezpecenie dovernosti
- je odvodeny z vojenskehej viacvrstvovej bezpecnostnej paradigmy, ktora bola tradicne pouzivana vo vojenskych organizaciach pre klasifikaciu dokumentov
- ma striktne linearne poradie bezpecnostnych levelov tak, ze kazdy dokument ma specificky bezpecnostny level a kazdy pouzivatel je priradeny k striktnemu levelu pristupu, takze mozu pristupovat k vsetkym dokumentom s korespondujucim levelom a vsetkymi levelmi nizsie.

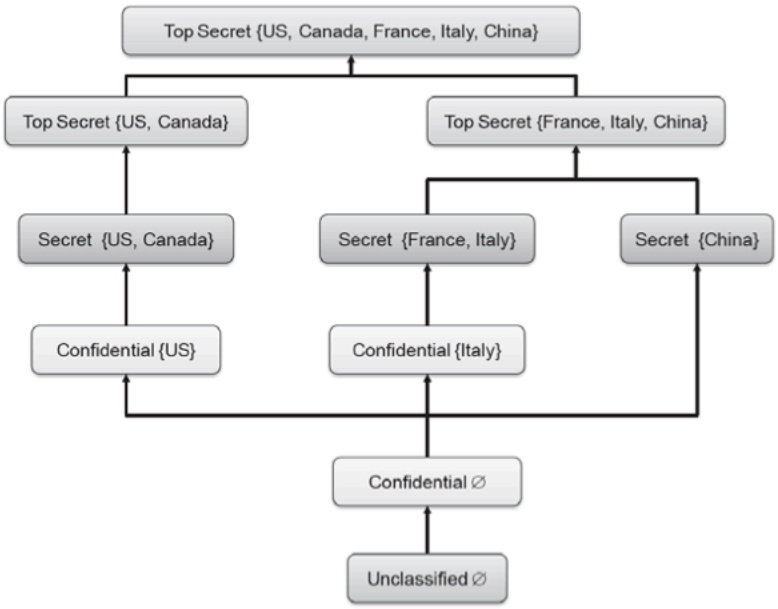


Uplne poradie a Ciastocne poradie

- linearne zoradzovanie pre dokumentov moze by definovane na zaklade porovnavacieho pravidla. Mozme povedat ze pravidlo definuje uplne poradie na univerzalnom zozname U ak splna tieto vlastnosti:
    - **Reflexia**: ak x patri U potom x <= x
    - **Antisymetria**: ak x <= y a y <= x potom x = y
    - **Tranzitivnost**: ak x <= y a y <= z potom x <=z
    - **Uplnost**: ak x a y su v U potom x <= y alebo y <= x
- vsetky bezne definicie „menej nez alebo rovne k“ pre cisla, ako integre a realne cisla su uplne poradie
- ak vynechame poziadavku uplnosti potom dostavame Ciastocne poradie
    - priklad ciastocneho poradia je zoznam kurzov na univerzite kde mozme povedat ze 2 kurzy A a B, A<=B ak A je prerekvizita pre B

Ako BLB model pracuje

- kazdy level bezpecnosti v BLP formulari je ciastocnym poradim, <=
- kazdy objekt, x, je priradeny do levela bezpecnosti, L(x), Podobne kazdy pouzivatel, u, je priradeny do levela bezpecnosti, L(u). Pristup k objektom cez pouzivatelov je kontrolovany nasledujucimi pravidlami:
    - **Jednoducha bezpecnostna vlastnost**: pouzivatel u moze citat objekt x jedine ak L(x) <= L(u)
    - ***-vlastnost**: pouzivatel u moze zapisovat (vytvarat, editovat, pridavat) objekt jedine ak L(u) <= L(x)
- **jednoducha bezpecnostna vlastnost** taktiez nazyvana „no read up“ pravidlo, zabranuje pouzivatelom citat objekty ktore maju bezpecnostny level vysssi ako pouzivatelia
- ***-vlastnost** je tiez nazyvana „no write down“ pravidlo. To znamena ze zabranuje propagacii informacii pouzivatelovi s nizsiim bezpecnostnym levelom


Top Secret {US, Canada, France, Italy, China}
Top Secret {US, Canada}
Top Secret {France, Italy, China}
Secret {US, Canada}
Secret {France, Italy}
Secret {China}
Confidential {US}
Confidential {Italy}
Confidential ∅
Unclassified ∅

Skupiny a kontrola pristupu, role, ochranne urovne

Sposoby definovania kontroly pristupu

Kontrola pristupu na zaklade matice

Matica kontroly pristupu je tabulka definujuca opravnenia.

- Kazdy riadok v tabulke je definovany subjektom, ktorym moze byt pouzivatel, skupina, alebo system, ktory uskutocnuje operacie.
- Kazdy stlpec v tabulke je definovany objektom, ktorym moze byt subor, priecinok, dokument, zariadenie, zdroj alebo nejaka ina entita pre ktoru chceme definovat prava.
- Kazda bunka tejto tabulky obsahuje pridelene prava medzi dannym subjektom a dannym objektom
- Pristupove prava mozu obsahovat akcie ako su: citanie, zapisovanie, kopirovanie, spustanie, mazanie a komentovanie
- Prazdna bunka v tabulke znamena ze zdiadne opravnenia neboli pridelene

Vyhody: jednoduchost

Nevyhody: realizacia implementacie v pripade ze mame vela subjektov a objektov

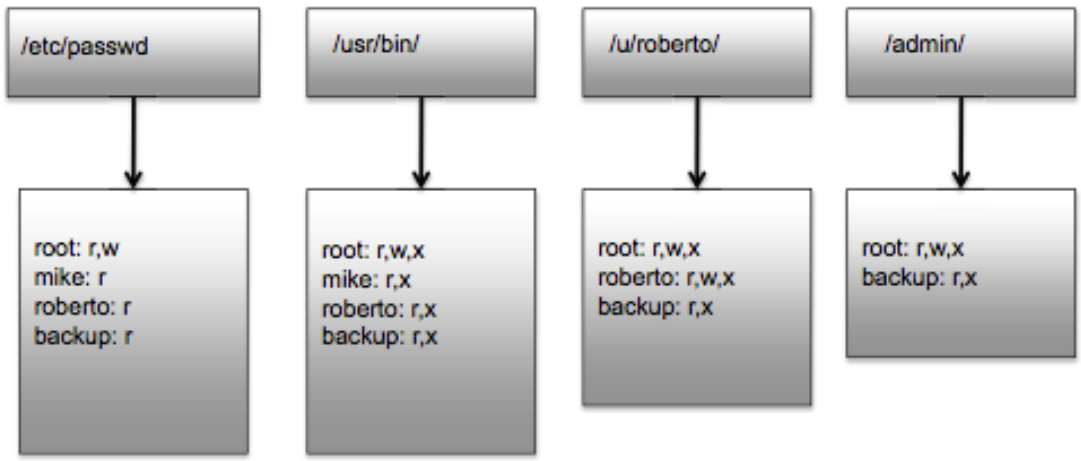
	/etc/passwd	/usr/bin/	/u/roberto/	/admin/
root	read, write	read, write, exec	read, write, exec	read, write, exec
mike	read	read, exec		
roberto	read	read, exec	read, write, exec	
backup	read	read, exec	read, exec	read, exec
. . .	. . .	. . .	. . .	. . .

Kontrola pristupu na zaklade zoznamu

Definuje pre kazdy objekt zoznam nazyvany aj zoznam kotroly pristupu, ktory pozostava zo vsetkych subjektov ktori maju pristup k dannemu objektu a kazdy z tychto subjektov ma definovane pristupove prava (citanie, zapisovanie, ...) k danemu objektu.

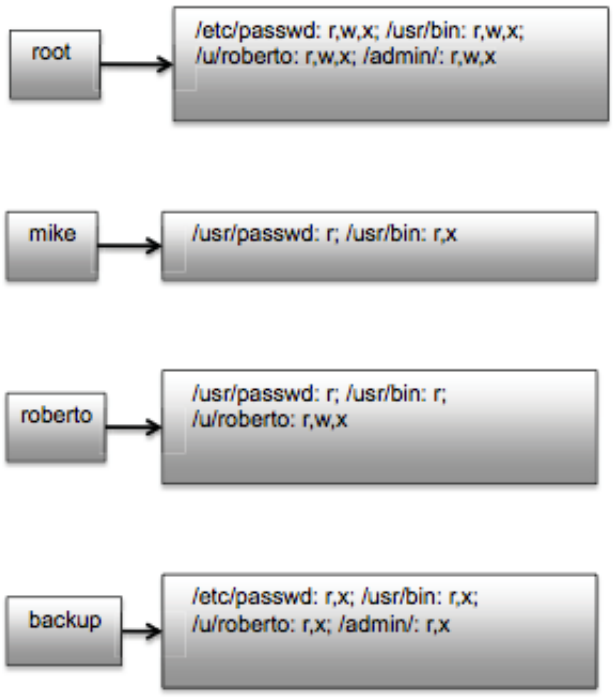
Vyhody: jednoducha implementacia

Nevyhody: pomerna zlozitost vyhladavania vsetkych objektov pre danny subjekt



<u>Druha moznost implementacie zoznamu kontroly pristupu</u>

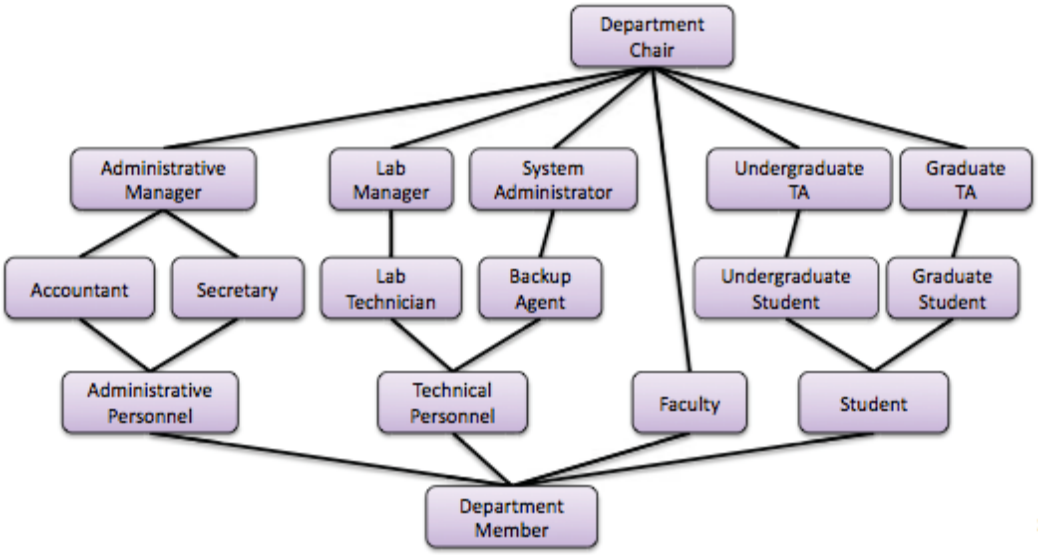
Definuje pre kazdy subjekt zoznam objektov pre ktore ma subjekt nejake specificke prava.



Vyhody: vyhladavanie
Nevyhody: podobne ako pri matici, zlozita implementacia

Kontrola pristupu na zaklade roli

Definovanie roli a specifikovanie opravneni pre tieto role, ktore sa potom priradzuju subjektom. Rola vie dedit prava od inej roli a tym sa nam vytvara struktura (strom) roli.



Doveryhodna vypoctova baza TCB

Polymorficka pocitacova hrozba

Two-phase update

Two-Phase Commit – tento protokol vyuziva vacsina DB pri urcovani doverihodnosti otazky a pomaha DB dosiahnut integritu a dostupnost. Sklada sa z dvoch faz:

1. request phase (poziadavka) – identifikuju sa a oznacia vsetky casti DB, ktore maju byt zmene. Tato faza skonci uspesne, ak su oznacene vsetky casti, ktore maju byt zmene a protokol pokracuje 2. fazou. Alebo sa prerusi ak nedokazala oznacit vsetky casti (oznacil ich uz niekto iny, zlyhanie siete alebo systemu) a resetne vsetky pozadovane zmeny.
2. commit phase (potvrdenie) – DB sa zablokuje pred vykonavanim dalsich zmien a vykona zmeny identifikovane v prvej faze. Ak zbehne uspesne, tak odstrani vsetky vlajky identifikujuce pozadovane zmeny a zrusi zamok nad DB.
   Ak operacia zlyha, tak vrati vsetky vykonane zmeny a to tak, ze vrati DB do stavu tesne pred dokoncenim prvej fazy.

Kerberos

- Sietovy autentifikacny protokol, ktory autentifikuje klientov na sluzby a naopak
- po nezabezpecenej sieti je mozne bezpecne overit identitu medzi dvomi ucastnikmi
- postaveny na symetrickom sifrovani
- pouziva sa port 88
- vydavany pod licenciou podobnou BSD
- vyuziva koncept listka ako tokenu, ktory reprezentuje identitu pouzivatela
- listky su digitalne dokumenty, ktore drzia session kluce. Tieto kluce su typicky vydane pocas login session a potom mozu byt pouzite namiesto hesiel pre rozne Kerberizovane sluzby.

Kerberos Servers

K dosiahnutiu bezpecnej autentifikacie, Kerberos pouziva doverihodnu tretiu stranu znamu ako disctribucne centruk klucov (**key distribution center - KDC**), ktore je tvorene 2 komponentami typicky integrovanimi do jedneho servera:

- autentifikacny server (**authentication server - AS**), ktory vykonava autentifikaciu
- udelovaci server pre listky (**ticket-granting server - TGS**), ktory udeluje listky pouzivatelom

Autentifikacny server udrzuje databazu ulozenych skrytych klucov od pouzivatelov a sluzieb. Skryty kluc pouzivatela je typicky vygenerovany pomocou funkcie ktora vygeneruje jednocestny hash z pouzivatelovho hesla.

Princip Kerberos autentifikacie

- pri sifrovani vystupuju tieto entity:
  - AS – autorizacny server
  - SS – servisne stredisko
  - TGS – ticket-granting server – riadiaci server
  - TGT – ticket granting ticket – ticket opravnujuci komunikaciu s TGS

1. Uzivatel zada login a heslo, vygeneruje na zaklade neho hash (**kluc1**), ten nikam neposiela
2. Poziada AS o pristup k sluzbe a odosle mu svoje IDcko (nezasifrovane)
3. AS skontroluje ci taky user existuje, ak ano, posle mu 2 spravy:

- sprava A - TGS kluc (**kluc2**) zasifrovany klucom **kluc1**
- sprava B - TGT obsahujuci ID, sietovu adresu, zivotnost ticketu a **kluc2**, toto vsetko je zasifrovane klucom TGS (**kluc3**)

4. klient obdrzi tieto 2 spravy, spravu A desifruje hashom, ktory ziska zo svojho loginu a hesla a pouzije na desifrovanie spravy s klucom 1
5. spravu B nie je schopny desifrovat pretoze user nema **kluc3**
6. ziadanim o pristup posiela user 2 spravy serveru

- sprava C - obsahuje spravu B a ID sluzby, ktorej pristup ziada
- sprava D - autentifikator (ID klienta a casova znacka) sifrovany klucom **kluc2**

7. TGS rozkoduje C a z nej ziska B, ktoru desifruje pomocou **kluc3**, z toho ziska **kluc2**
8. pomocou kluc2 desifruje D a odosle klientovi dalsie 2 spravy:

- sprava E - klient/server ticket obsahujuci ID klienta, jeho sietovu adresu, dobu platnosti a klient/server kluc (**kluc4**), to vsetko zasifrovane pomocou kluc5
- sprava F - klient/server kluc (**kluc4**) sifrovany klient/TGS klucom (**kluc2**)

9. klient ma teraz dost informacii k autentizacii voci SS. Klient sa k nemu pripoji a posle 2 spravy:

- sprava E - len preposle existujucu spravu E, ktoru uz dostal v kroku 8
- sprava G - novy autentifikatror, obsahuje ID klienta, casovu znacku, zasifrovane pomocou kluc4

10. SS desifruje E pomocou **kluc5** a ziska **kluc4**
11. SS desifruje G, z nej ziska autentifikator a posle klientovi spravu aby potvrdil svoju identitu a ochotu posluzit

- sprava H - inkrementovana casova znacka klientovho autintifikatora, zasifrovane **kluc4**

12. klient desifruje H pomocou **kluc4** a skontroluje ci je casova znacka spravne inkrementovana, pokial ano, moze doverovat serveru a moze zacat posielat ziadosti o sluzby
13. server poskytuje sluzby

Vyhody

- Kerberos protokol je navrhnuty byt bezpecny aj v nezabezpecenej sieti

- Pretoze je prenos sifrovany s pouzitim tajneho kluca, utocnik nemoze sfalsovat platny listok k ziskaniu neautorizovanemu pristupu k sluzbam
- Vyuziva symetricke sifrovanie ktore je vypoctovo efektivnejsie ako asymetricke

Nevyhody

- musi byt nepretrzity beh centralneho servera - pokial nebezi, nikto sa neprihlasi
- musi byt prisna synchronizacia casu - nesmie sa lisit o viac ako 5 minut, pretoze sa pouzivaju casove znamky
- ak sa utocnik dohodne s KDC, vsetky autentifikacne informacie o pouzivateloch a serveru na sieti budu odhalene