Ochrana informace I. - otázky ke zkoušce

sepsali František Princ, Jindřich Vodrážka

ZS 2006/2007

1 Auditovatelnost

Jakákoliv akce ohrožující bezpečnost je vystopovatelná ke konkrétnímu autentizovanému subjektu. Je třeba vést záznamy o tom, kdo co dělal s kterými položkami nejen pro to, abychom byli schopni sledovat přístupy a změny, ale i pro dlouhodobé sledování uživatelů a následné rozhodování, zda vyhovět žádosti. Je nutné zvolit vhodnou granularitu - bloky, záznamy, položky. Zde přistupuje tzv. pass through problem - uživatel smí přistupovat k objektu, ale tento mu nesmí být předán (např. při vyhledávání). Uživatel může zjistit hodnotu položky i bez přímého dotazu - nestačí log žádostí o přístup k odhadu toho, co ví.

2 Autentizace pro počítače

Vhodnou metodou pro autentizaci počítačů je např. metoda jednorázových hesel. V podstatě se jedná o Challenge-response systém. Počítač, který se chce autentizovat obdrží náhodný dotaz, který zpracuje (např. zašifruje tajným klíčem) a odešle výsledek. Výsledek je ověřen a pokud je správný, autentizace je uskutečněna.

3 Autentizace v prostředí databáze

Každý, komu je povolen přístup k databázi, musí být pozitivně identifikován. SŘBD[1] potřebuje přesně vědět, komu odpovídá. Protože však zpravidla běží jako uživatelský proces, nemá spolehlivé spojení s jádrem OS a tedy musí provádět vlastní autentizaci.

4 Autentizace v síti

Protože síťové prostředí zpravidla není považováno za bezpečné, je třeba využívat autentizační mechanismy odolné vůči odposlechu, resp. aktivním

[1] System Řízení Báze Dat

útokům. Často bývá žádoucí řešit jednotné přihlášení (single sign on).

- cookies
- tickety
- certifikáty, PKI
- čipové karty

S procesem integrace autentizačních mechanismů souvisí nutnost zavedení centrální správy uživatelů nebo alespoň synchronizace záznamů o uživatelích.

5 Autorizace

Existují různé úrovně ochrany objektů:

1. *Žádná ochrana* - Je nutná alespoň samovolná časová separace procesů.
2. *Izolace* - Procesy o sobě vůbec neví a systém zajišťuje ukrytí objektů před ostatními procesy.
3. *Sdílení všeho nebo ničeho* - Vlastník objektu deklaruje, zda je objekt **public** nebo **private** (tedy jen pro něho).
4. *Sdílení s omezenými přístupy* - OS testuje oprávněnost každého přístupu k objektu. U subjektu i objektu existuje záznam, zda má subjekt právo přístupu k objektu.
5. *Sdílení podle způsobilosti* - rozšíření předchozího - Oprávnění dynamicky závisí na aktuálním kontextu.
6. *Limitované použití objektů* - Kromě oprávnění přístupu specifikujeme, jaké operace smí subjekt s objektem provádět.

6 Bell-LaPadula model

Popisuje povolené přesuny informací takové, aby bylo zajištěno jejich utajení. Pro každý subjekt S a objekt O je v systému definována bezpečnostní třída $C(S)$, $C(O)$.

- *Vlastnost jednoduché bezpečnosti:* Subjekt S může číst objekt O právě tehdy, když $C(O) \leq C(S)$.
- **-vlastnost:* Subjekt S mající právo čtení k objektu O může zapisovat do objektu P právě tehdy, když $C(O) \leq C(P)$.

Obyčejně nepotřebujeme tak silná omezení, která klade *-vlastnost. Často je tato vlastnost poněkud oslabena v tom smyslu, že systém povolí zápis do objektu nižší bezpečnostní třídy, pokud zapisovaná data nezávisí na čtených údajích. Model je používán v systémech, které paralelně zpracovávají informace různého stupně utajení.

7 Bezpečnost fyzické přenosové vrstvy

Bezpečnost je do určité míry závislá na použitém přenosovém médiu. Útok proti komunikačním linkám může být pasivní (pouze odposlech), nebo aktivní (vkládání dalších informací do komunikace).

- **Kabely** - Častým útokem je tzv. napíchnutí (wiretaping). Proti tomuto způsobu útoku jsou obzvláště bezbranné metalické vodiče, je však možné monitorovat i optické kabely. Obecně lze mezi metalickými kabely považovat za bezpečnější kabely koaxiální. Vyrábí se celá řada kabelů s omezeným vyzařováním, případně s detekcí napíchnutí. K napíchnutí jsou náchylnější pevné linky (leased lines). Obecně je útok pravděpodobnější u některého z konců linky.

- **Mikrovlny** - Svazek není možno zcela přesně směrovat, navíc se mírně rozbíhá. Komunikace může být zachycena kdekoliv mezi vysílačem a přijímačem nebo v prostoru za přijímačem. Obdobné nedostatky mají mikrovlny i z hlediska možnosti aktivního útoku.

- **Satelitní přenos** - Poznamenejme, že ta samá technologie je používána k **šíření** TV signálu.

- **Celulární rádio** - Nebezpečí útoku je velké. Zejména pasivní útok je snadno proveditelný.

8 Bezpečnostní architektura OS (ring, vrstvový model)

- **Vrstvový model** (Layered design) - Již operační systémy s kernelem obsahují několik vrstev - hardware, kernel, zbytek OS, uživatelské procesy. Tyto vrstvy lze dále dělit. Např. na uživatelské úrovni můžeme oddělit semi-systémové programy jako různé databázové systémy, shelly apod. Vrstvy lze chápat jako soustředné kruhy. Čím blíže je vrstva středu, tím je důvěryhodnější a bezpečnější. Ne všechny bezpečnostní funkce (např. autentizace uživatele) jsou implementovány uvnitř bezpečnostního jádra. Bezpečnostní jádro spolupracuje s okolními spolehlivými vrstvami, které by měli být formálně ověřeny a přinejmenším dobře otestovány. Každá vrstva používá služby nižších vrstev a sama vyšším vrstvám

poskytuje služby jisté úrovně bezpečnosti. Stejná funkce může být implementována v několika vrstvách zároveň.

- **Kruhová struktura** (Ring structured) - Kruhy jsou číslovány od 0 (kernel). Čím důvěryhodnější proces je, tím nižší je číslo kruhu, do kterého patří. Kruhy jsou soustředné a překrývající se - proces patří do kruhu k a všech dalších. Ve středu je HW počítače. Každá procedura nebo oblast obsahující data se nazývá segment. Ochrana segmentu je založena na trojici $< b_1, b_2, b_3 >, b_1 \leq b_2 < b_3$, nazývané závora kruhu (ring bracket). (b_1, b_2) nazýváme přístupová závora (access bracket), (b_2, b_3) je závora volání (gate extension, call bracket). Nechť programová rutina patří do kruhu k, pokud $k = b_1$, může pracovat přímo s daty tohoto segmentu. Pokud $b_1 < k \leq b_2$, může pracovat přímo s kopií dat a pokud $b_2 < k \leq b_3$, může k datům přistupovat pouze prostřednictvím definovaného rozhraní (gate). Tento základní mechanismus, nazývaný též nondiscretionary nebo mandatory control může být dále doplněn o další doplňkové (discretionary) mechanismy. Např. k daným datům smějí přistupovat pouze jmenovité procesy. Procesy patřící do okruhu přístupové závory mohou volně číst, ale zapisovat pouze za specifických podmínek apod.

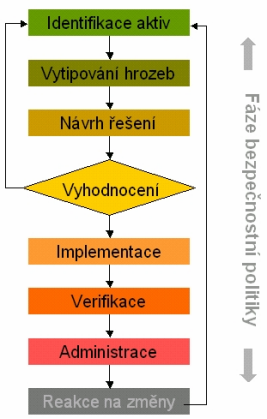
9 Bezpečnostní kernel

Bezpečnostní kernel poskytuje základ pro vybudování bezpečnostního mechanismu, často bývá implementován uvnitř kernelu. Uzavřít bezpečnostní funkce systému do security kernelu má několik důvodů:

- Oddělení od zbytku systému zjednodušuje ochranu mechanismu.
- Všechny bezpečnostní funkce jsou shromážděny v jednom kusu kódu $\Rightarrow$ implementace bezpečnosti je kompaktní.
- Kernel nebývá velký $\Rightarrow$ implementace je snadno ověřitelná.
- Je snazší provádět testování a změny bezpečnostního mechanismu.
- Přes kernel procházejí veškeré žádosti o přístup ke všem objektům (volání odpovídajících modulů) $\Rightarrow$ je možno zachytit každý přístup.

10 Bezpečnostní modely obecně

První fází tvorby bezpečného IS je volba vhodného bezpečnostního modelu. Základní požadavky bezpečnosti: *utajení, integrita, dostupnost, anonymita.* Předpokládejme, že umíme rozhodnout, zda danému subjektu poskytnout přístup k požadovanému objektu. Modely poskytují pouze mechanismus pro rozhodování!



Obrázek 1: Fáze bezpečnostní politiky

- **Jednoúrovňové modely** jsou vhodné pro případy, kdy stačí jednoduché ano/ne rozhodování, zda danému subjektu poskytnout přístup k požadovanému objektu a není nutné pracovat s klasifikací dat.
- **Víceúrovňové modely** - Může existovat několik stupňů senzitivity a "oprávněnosti". Tyto stupně senzitivity se dají použít k algoritmickému rozhodování o přístupu daného subjektu k cílovému objektu, ale také k řízení zacházení s objekty. Víceúrovňový systém "rozumí" senzitivitě dat a chápe, že s nimi musí zacházet v souladu s požadavky kladenými na daný stupeň senzitivity. Rozhodnutí o přístupu pak nezahrnuje pouze prověření žadatele, ale též klasifikaci prostředí, ze kterého je přístup požadován.

11 Bezpečnostní politika

Bezpečnostní politika je dokument, ve kterém se naplánuje, jakým způsobem se budou řešit všechny oblasti bezpečnosti, kdo je za co zodpovědný a jak to bude implementováno a provozováno. Bezpečnostní politika říká, jak zvládnout problém zajištění IS proti incidentům. Důležitější než rozsah je, aby pokrývala všechny důležité okruhy problémů formou, která je srozumitelná všem, kterých se týká. Materiál by měl popisovat konkrétní IS. Organizace pohybující se ve stejné branži budou mít podobné nároky na bezpečnost. Fáze bezpečnostní politiky popisuje obrázek **??**. Bezpečnost je proces - bez soustavného přizpůsobování se změnám vnějšího prostředí a vývoji vlastního IS je to celé k ničemu. Stejně jako každá činnost, i provozování systému pro správu informací je spojeno s jistým rizikem (chyba zařízení, obsluhy, programu, vandalismus, krádež). Provedení kvalifikovaného odhadu rizik přináší výhody - viz otázka **Odhad rizik**. Bezpečnostní poli-

tika vyjadřuje vůli pracovat na dosažení jistého stupně bezpečnosti. Bývá rozdělena do více dokumentů:

1. Statement (záměr bezpečnosti) - cca 1 strana základního záměru, podepsaná top managementem
2. Politika a principy bezpečnosti - podle potřeby až 100 stran

Bezpečnostní politika by měla obsahovat:

- Popis celkových cílů bezpečnostních aktivit - např. ochrana dat před katastrofami, před úniky mimo organizaci, apod.
- Kdo má zodpovědnost za udržení bezpečnosti - pověřený pracovník, vedení, všichni
- Závazky organizace na udržení bezpečnosti - počet vyčleněných pracovníků, minimální výdaje do této oblasti

12 Biba model

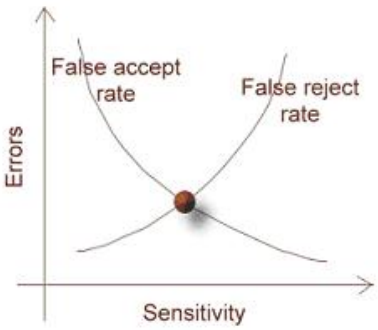
Biba model je duálním modelem k Bell-LaPadula modelu. Bell-LaPadula model se vůbec nezabývá integritou dat. Nechť pro každý subjekt S a objekt O je v systému definována integritní bezpečnostní třída $I(S)$ a $I(O)$. Obdobně jako v Bell-LaPadula modelu definujeme:

- *Vlastnost jednoduché integrity:* Subjekt S může modifikovat objekt O právě tehdy, když $I(O) \leq I(S)$.
- *Integritní *-vlastnost:* Subjekt S mající právo čtení k objektu O může zapisovat do objektu P právě tehdy, když $I(O) \geq I(P)$.

Biba model se zabývá zajištěním integrity a tedy i důvěryhodnosti dat. Bezpečnostní třída entity popisuje míru její důvěryhodnosti pro ostatní. Tento model vůbec neřeší utajení dat.

13 Biometriky

Jde o techniky identifikace lidí na základě jejich osobních charakteristik. Navzájem se odlišují různou mírou spolehlivosti, ceny a v neposlední řadě i společenské přijatelnosti. Hledáme charakteristiky mající dostatečnou meziosobní variabilitu při zachování vnitro-osobní reproducibility. Kvalitu biometriky lze charakterizovat četností nesprávných odmítnutí autorizovaného subjektu a četností nesprávných přijetí neautorizovaného subjektu (útočníka). Situaci vystihuje obrázek **??**.



Obrázek 2: Graf chybovosti biometrik

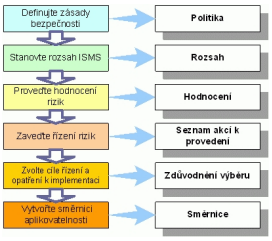
- **Verifikace hlasu** - Testovaný subjekt přečte systémem náhodně zvolenou frázi. Je proveden rozbor zvuku na základě jednotlivých složek zvuku. Výsledek je vhodným způsobem komprimován (1-2 kB) a porovnán se srovnávacím vzorkem. Výhodou je přirozenost a možnost provádět verifikaci např. prostřednictvím telefonu.

- **Verifikace dynamiky podpisu** - Sledují se změny tlaku, zrychlení v jednotlivých částech, celkový průběh zrychlení, zarovnání jednotlivých částí podpisu, celková rychlost, celková dráha a doba pohybu pera na a nad papírem apod. Ze získaných hodnot je opět vytvořen vzorek, který je porovnán se srovnávacím vzorkem. Výhodou je opět přirozenost a sociální akceptovatelnost, nevýhodou malá mechanická odolnost snímačů a značná variabilita podpisu u některých lidí.

- **Verifikace otisků prstů** - Systém provádí statistický rozbor výskytu tzv. markant - hrbolků, smyček a spirál v otisku prstu a jejich vzájemné polohy. Často se provádí testování uživatelem zvoleného výběru několika prstů. Výhodou je vynikající mezi/vnitro-osobní variabilita, a dobrá zpracovatelnost vstupních dat, nevýhodou jsou možné negativní asociace uživatelů a vyšší cena technologie.

- **Geometrie ruky** - Metoda zkoumá délku a šířku dlaně a jednotlivých prstů, boční profil ruky apod. Výsledkem je velmi malý vzorek - cca 18 bytů. Metoda je poměrně spolehlivá, avšak poněkud dražší. Možnost podstrčení odlitku ruky.

- **Obrazy sítnice** - Zařízení pořídí obraz struktury sítnice v okolí slepé skvrny, tento obraz je digitalizován a převeden na vzorek délky přibližně 40 bytů. Obrázky sítnice mají stejné charakterizační vlastnosti jako otisky prstů. Výhodnou metody je značná spolehlivost a velmi obtížná napodobitelnost. Proto jde o metodu vhodnou k nasazení v prostředí



Obrázek 3: Předpokládaný proces tvorby bezpečnosti

nejvyššího utajení. Nevýhodou jistá subjektivní nepříjemnost a velmi drahá technologie.

- **Další biometriky** - rysy obličeje, Bertillonovy míry, rytmus psaní na klávesnici, EEG, EKG, otisky dlaní a chodidel, otisky chrupu, genetické rozbory ...

14 BS 7799

BS 7799 je organizační norma, která popisuje obecně, jaké činnosti musí organizace vykonávat pro zajištění bezpečnosti IS. Nestanovuje kvalitativní kritéria. Je založena na myšlence budování bezpečnosti shora dolů, tj. od bezpečnostní politiky po implementaci protiopatření. Předpokládaný proces tvorby bezpečnosti a z něho odvozené bezpečnostní dokumentace ukazuje obrázek **??**. BS 7799 pokrývá tyto oblasti: bezpečnostní politika, klasifikace a řízení aktiv, personální bezpečnost, fyzická bezpečnost a bezpečnost prostředí, řízení provozu a komunikací, řízení přístupu, vývoj a údržba systémů, řízení kontinuity operací, soulad s požadavky (právní, technické, audit). Pro podporu budování bezpečnosti a návrhu implementace bezpečnostních mechanismů podle BS7799 existuje standardní metodika a automatizovaný nástroj CRAMM.

15 Certifikace

Certifikace je proces potvrzení toho, že daný IS prošel evaluací a že splňuje daná kritéria, případně že se shoduje s konkrétní normou. Certifikát je zárukou kvality pro pořizovatele certifikovaného bezpečnostního systému. Certifikáty stanovují srovnatelná kritéria pro porovnávání bezpečnostních systémů.

16 Co je to hrozba?

Hrozba je skutečnost, která potenciálně může být způsobit bezpečnostní incidentu. Zdaleka nejstrašnější hrozbou jsou vlastní uživatelé.
Další příklady hrozeb: povodně a záplavy, požáry, zloději, rozvědky, konkurence, hackeři, vandalové, nešikové s bagrem, viry a červi, závady techniky, výpadky napájení, teplota, vlhkost, vibrace. Možné hrozby jsou tyto:

1. **Přerušení** - některá část systému je ztracena nebo nedosažitelná
2. **Zachycení** - neautorizovaný subjekt získá přístup k nějakému objektu systému
3. **Modifikace** - neautorizovaný subjekt získá možnost pozměňovat některé části systému
4. **Fabrikace** - neautorizované vytvoření nového objektu

Seznam relevantních hrozeb si musí každý sestavit sám. Někdy se tomu učeně říká **model ohrožení**. Naplněním hrozby vznikne **bezpečnostní incident**, jehož finančnímu vyjádření se říká **dopad**. Rozsah hrozeb spolu s pravděpodobností jejich realizace udává celkovou míru rizika.

17 Common criteria

Je metanorma stanovující principy a postupy, jak odvozovat konkrétní technické normy pro vývoj, testování, výsledné vlastnosti a provoz technických bezpečnostních protiopatření v různých prostředích. Formalizuje proces vyhodnocování: evaluační kritéria → evaluační metodologie → evaluační schéma → evaluace → výsledky evaluace → certifikace → registr certifikátů. Odděluje funkcionalitu od "jistoty". Sada konkrétních funkčních a "jistotních" požadavků tvoří profil zabezpečení.

- **Funkční třídy:** bezpečnostní audit, komunikace, kryptografická podpora, ochrana uživatelských dat, identifikace a autentizace, bezpečnostní management, soukromí, ochrana bezpečnostního mechanismu, využívání prostředků, přístup, důvěryhodná cesta/kanál
- **Jistotní třídy:** správa konfigurací, dodávka a provoz, vývoj, dokumentace a návody, podpora životního cyklu, testování, vyhodnocení slabin
- **Vyhodnocení kvality bezpečnostního mechanismu:** vyhodnocení profilu bezpečnosti, vyhodnocení cíle hodnocení

- **Úrovně vyhodnocení dle kritérií:** funkční testování, strukturální testování, metodické testování a kontroly, metodický návrh + testování a ověření, semiformální návrh a testování, semiformálně verifikovaný návrh a testování, formální návrh a testování

18 Dvou - třífázový update

- **Dvoufázový update:** Problém v konzistenci dat může vzniknout během provádění modifikace např. pádem výpočetního systému. Proces modifikace tedy rozdělíme na dvě části.

  1. *Záměr* (intent) - Provede se načtení relevantních dat, uzamčení záznamů, vytvoření záznamů a kalkulace výsledků. Poslední událostí první fáze je operace commit.
  2. *Zápis* - V rámci druhé fáze jsou prováděny trvalé změny dat s ohledem na data získaná v první fázi.

  Pokud některá z fází nedoběhne, může být snadno opakována. Po ukončení fáze je systém opět konzistentní.

- **Třífázový update:** Pokud máme distribuovanou databázi, musíme řešit problém, aby všechny systémy měly konzistentní data. Zavedeme si pojem **quorum**, což je číslo, které danému systému udává, zda může alespoň číst nebo i zapisovat ($\frac{1}{3}$ pro čtení a $\frac{2}{3}$ pro zápis). Jde o to, že když vypadne spojení na jeden ze serverů, tak aby si tento server "nešel vlastní cestou." Třífázový zápis:

  1. Záměr (všechny servery)
  2. Quorum (pokud je jich více jak $\frac{2}{3}$, pak OK)
  3. Zápis na fyzická média (disk)

19 Dynamické metody testování

Jedná se o testování běhu programu - analýza výsledků zkušebních běhů a simulací (testování a prototypování).

Několik příklady dynamických metod:

- Funkční testování - Spuštěním systému se zkoumá, zda jsou naplněny požadavky uživatelů.
- Analýza mezních hodnot - Detekce chyb na mezních vstupech (nuly, prázdná pole a řetězce, první a poslední položka, ...).
- Testování rozhraní - Ověření reakcí na vstupy, hledání chyb v I/O.

- Testování výkonu - Porovnává skutečný výkon s požadavky. Zkoumá zatížení komponent.

20 Evaluace

Proces, při kterém je bezpečnostní systém (Target of evaluation) podroben sérii testů, za účelem ověření jeho funkčnosti a míry naplnění požadovaných vlastností vzhledem k předem stanoveným evaluačním kritériím. Evaluací se detailně zabývá dokument Common Evaluation Methodology.

21 Granularita oprávnění

Kontrola přístupu může být implementována na různých úrovních (byte, věta, soubor, adresář, ...). Je potřebné volit mezi režií kontroly a dostatečně jemným rozlišením. Náročnost implementace roste s zároveň s granularitou.

Granularita autorizace:

- Ochrana po skupinách - Uživatelé jsou podle svého zaměření, pracovního zařazení, atd. vhodně rozděleny do skupin. Pro účely ochrany objektů je svět rozdělen na vlastníka souboru, skupinu, do které vlastník patří a ostatní uživatele. Předpokládá se, že uživatelé v rámci skupiny potřebují sdílet data. Při vytvoření objektu vlastník specifikuje, jaká práva přiděluje sobě, uživatelům ve stejné skupině, ostatním. Metoda je jednoduchá, snadno implementovatelná leč neposkytuje dostatečně jemné rozlišení. Navíc je většinou nutné, aby každý uživatel byl právě v jedné skupině, jinak nastávají problémy s přidělováním práv skupinám.

- Hesla a jiné tokeny - Při vytvoření objektu vlastník specifikuje hesla, potřebná pro jisté módy přístupu k objektu. Heslo zašle uživatelům, kteří mají mít přístup. Systém splní žádost o přístup k objektu pouze tomu, kdo se prokáže odpovídajícím heslem. Nevýhodou je, že v případě zapomenutí není možno zjistit, jak heslo vypadalo. V případě, že dojde k vyzrazení hesla, je složité nastavit nové. Stejně obtížné je odejmout právo přístupu.

- Dočasné propůjčení oprávnění - Mechanismus známý ze systému UNIX. Funguje stejné přidělování práv jako v případě ochrany po skupinách. Navíc je možno stanovit, že (spustitelný) soubor smí být prováděn s oprávněním vlastníka. Prostřednictvím rutin běžících s oprávněním vlastníka lze řízeně přistupovat k objektům, ke kterým uživatel přímý přístup nemá.

- VAX VMS/SE - Ke každému souboru může uživatel vytvořit *Seznam oprávnění*, udávající kdo má jaká práva. Každý uživatel je členem

jedné skupiny, navíc administrátor může vytvořit skupinu typu *obecný identifikátor* a tuto skupinu mohou uživatelé uvádět v Seznamech oprávnění. Seznamy oprávnění mohou být též použity pro přidělování přístupu k ostatním systémovým zdrojům.

- Systém rolí a skupin - Oprávnění jsou sdružována do ucelených souhrnů (tzv. rolí), které odpovídají svým obsahem okruhu práce, kterou vykonává pracovník na určitém zařazení (správce uživatelů, finanční účetní, skladník, ...). Uživatel nezískává oprávnění "po jednom", ale přidělením role. Pro zjednodušení práce bývá k dispozici systém kompozitních rolí, odvozených rolí atd. Namísto práce s jediným uživatelem může být možné definovat a hromadně spravovat celé skupinu uživatelů, majících stejná oprávnění.

22 Hesla

Charakteristika dobrého hesla:

- Obsahuje kromě velkých a malých písmen též číslice a další na klávesnici se vyskytující znaky.
- Je dostatečně dlouhé.
- Nejde o obvyklé slovo nebo známou frázi.
- Je nepravděpodobné, nelze jej odvodit ze znalosti osoby vlastníka.
- Mělo by být často obměňované.
- Není nikde po okolí poznamenáno (ideálně není NIKDE poznamenáno).

Alternativou k heslu je **passphrase**, což je delší sekvence znaků (např. část přísloví, písně, říkanka). Pokud použijeme vhodný kompresní algoritmus, lze passphrase transformovat ve velmi kvalitní heslo **Skupinová hesla** jsou málo bezpečná a bývají často vyzrazena. **Piny** jsou číselné řetězce standardní délky sloužící jako hesla v souvislosti s kreditními a platebními kartami, mobilními telefony.

23 Chalenge-Response systémy

Heslo může být zachyceno v průběhu vkládání nebo při přenosu cílovému uzlu. Časté změny hesla jsou pro uživatele zatěžující. Vhodnější je, pokud systém zašle výzvu v podobě náhodné zprávy a uživatel jako heslo vrátí správnou reakci na tuto zprávu (např. její zašifrování tajným klíčem).

24 Chráněné objekty OS

- Paměť
- Procesor
- Spustitelné programy
- Sdílená zařízení typu disky
- Sériově znovupoužitelná zařízení (tiskárny, pásky)
- Sdílená data

25 Identifikace hodnot

Identifikace hodnot je prvním krokem při odhadování rizik. Jde o určení hodnot jednotlivých komponent systému. Hodnota komponenty se odvozuje z její ceny, ale také z její důležitosti pro správnou funkci systému. Př.: Napájecí kabel k serveru nestojí moc v porovnání se ztrátou v případě jeho poškození spojeného s výpadkem serveru ⇒ jeho hodnota se následkem této skutečnosti zvyšuje. V podstatě se jedná o inventarizaci systému s přihlédnutím k širším souvislostem.

Výsledku lze snáze dosáhnout sčítáním po kategoriích:

- hardware (počítače, monitory, tiskárny, disky)
- software (operační systém, koupené programy, vlastní zdrojové kódy)
- data (vlastní uložená data, logy, archivní kopie, listingy)
- lidé (pracovníci potřební ke správnému chodu systému, administrátoři, programátoři)
- dokumentace (programů, technického vybavení, systému, administrativních postupů)
- spotřební materiál (papír, tonery, datová média)

26 Integrita dat při přenosu

Přenos zpráv je řízen přenosovými protokoly zajišťujícími integritu dat, pořadí doručených částí, detekci duplicit apod. To však nestačí pro účely ochrany dat samotných. Tyto informace jsou v plaintextu bez potřebné detekce modifikací. Různé zabezpečovací kódy je snadné replikovat. Vhodnější je použití kryptografických kontrolních součtů → do každého šifrovaného bloku zprávy přidat jeho pořadové číslo, aby útočník nemohl provádět záměny pořadí notarizace zpráv. Každou posílanou zprávu je možné nechat ověřit centrální autoritou.

27 Integrita databáze

Spolehlivost a integrita v prostředí databází jsou pojmy ještě důležitější, než obvykle. Jde nám o zachování těchto tří vlastností:

1. Integrita databáze - celková správnost, ochrana před technickými závadami a poničením globálních struktur.
2. Elementární integrita - Změny a záznamy mohou provádět pouze autorizované entity.
3. Elementární správnost - Jsou přijata jen korektní data odpovídající typem, hodnotou.

Správce databáze musí zajistit, že změny dat mohou provádět pouze oprávnění uživatelé. Systém musí obsahovat prostředky překonávající nedostupnost položky nebo dokonce celé DB. Z hlediska OS a správce systému jde o ochranu relevantních souborů a programů, zálohy, kontroly zařízení atp. Z pohledu SŘBD přistupuje systém transakcí a logů, umožňující rekonstruovat stav databáze.

28 ITSEC

ITSEC = Information Technology Security Evaluation Criteria.

Je to mezinárodní sada kritérií, která jsou rozdělena na třídy funkčnosti (F) a korektnosti (E). Kritéria hodnocení funkčnosti jsou rozdělena na hodnocení následujících oblastí:

- Integrity systému (F-IN)
- Dostupnost systémových zdrojů (F-AV)
- Integrita dat při komunikaci (F-DI)
- Utajení komunikace (F-DC)
- Bezpečnost v rámci celé sítě (F-DX)

Každé z těchto kritérií může být vyhodnocováno nezávisle. Třídy funkčnosti F-D, F-C1, F-C2, F-B1, F-B2 a F-B3 zhruba co do funkčnosti odpovídají třídám C1 až B3 hodnocení TCSEC (= Orange Book).

Kritéria pro hodnocení korektnosti jsou přidána pro zvýšení důvěryhodnosti systému. Požadavky vyšší třídy korektnosti jsou vždy nadmnožinou všech předchozích.

- E1 - testování
- E2 - kontrola konfigurace a distribuce

- E3 - ověření detailního návrhu a zdrojového kódu
- E4 - zevrubná analýza slabin systému
- E5 - důkaz, že implementace odpovídá detailnímu návrhu
- E6 - formální modely, formální popisy a jejich vzájemná korespondence

Tyto třídy odpovídají požadavkům na důvěryhodnost kladeným třídami C2 až A1 hodnocení TCSEC.

29 Jak uchovávat autorizace

- **Adresář** (directory) - Metodu popíšeme pro případ uživatelů systému v roli subjektů a souborů coby objektů, lze ji však snadno rozšířit na libovolné objekty a subjekty. Každý soubor má svého vlastníka, který k němu vlastní veškerá práva včetně práva určovat rozsah oprávnění ostatních uživatelů k tomuto souboru. S každým uživatelem je spojena speciální struktura (**adresář**) obsahující odkazy na všechny soubory, k nimž má daný uživatel nějaké oprávnění, včetně popisu tohoto oprávnění. Žádný uživatel nesmí zapisovat do svého adresáře. Nevýhodou může být velký rozsah adresářů a velmi obtížná správa a úpravy takto přidělovaných oprávnění. Rovněž udržení přehledu o tom, kdo k danému souboru má jaká práva může být problematické.

- **Seznam oprávnění** (Access Control List) - Využívá opačný přístup k problému. Tentokrát je s každým objektem udržován seznam informací, které subjekty k němu mají jaká oprávnění. Metoda umožňuje snadno přidělovat implicitní práva subjektům případně skupinám subjektů. Při vhodném označení subjektů a použití expanzních znaků může být tato metoda dostatečně pružná. Seznamy zpravidla bývají udržovány setříděné tak, že záznamy s expanzními znaky jsou na konci. Tak stačí hledat první shodu s identifikací subjektu a použít tímto záznamem specifikované oprávnění.

- **Přístupová matice** (Access Control Matrix) - Řádky matice odpovídají jednotlivým subjektům, sloupce objektům. V políčku daném řádkem a sloupcem je záznam o úrovni oprávnění odpovídajícího subjektu k příslušnému objektu. Přístupová matice je zpravidla velmi velká záležitost, zhusta řídká.

- **Způsobilost** (Capability) - Způsobilost budeme chápat jako nefalšovatelný token, jehož vlastnictví dává vlastníkovi specifická práva k danému objektu. Lze chápat jako lístek do kina. Jednou z metod zajištění nefalšovatelnosti je, že tokeny se nepředávají přímo subjektům, ale jsou udržovány v chráněné oblasti paměti, přístupné pouze systému.

Při přístupu k objektu tak systém zkontroluje existenci příslušného tokenu, tento postup lze urychlit tím, že zvlášť udržujeme seznam způsobilostí právě běžícího procesu. Výhodou metody je, že dovoluje definovat nové dosud neznámé způsoby používání objektů a přidělovat odpovídající oprávnění. Nevýhodou je opět poněkud obtížná správa těchto tokenů, zejména odebrání způsobilosti je netriviální operace.

- **Security Labeling** - S každým subjektem a objektem asociujete bezpečnostní label popisující pověření/klasifikaci entity Podpora víceúrovňové modely.
- **Procedurálně orientovaný přístup** - Namísto přidělování obecného přístupu k subjektu (čtení, zápis) můžeme přidělovat právo používat některých funkcí z rozhraní, prostřednictvím kterého je objekt zpřístupňován. Metoda podporuje koncept skrývání a zapouzdřování informací popsaný v minulé lekci. Nevýhodou je jistá ztráta efektivity a rychlosti přístupu.

30 Jednorázové heslo

Řeší problémy úschovy hesel. Namísto konstantní fráze má uživatel přiřazenu konstantní funkci (vhodný matematický výpočet, dešifrování soukromým klíčem). V procesu autentizace obdrží od systému náhodně zvolené vstupní parametry a odpoví výsledkem. Metoda je obzvláště vhodná pro vzájemnou autentizaci strojů.

31 Kupujete IS na senzitivní informace

Při práci se senzitivními informacemi je zapotřebí zajistit spolehlivost a utajení na požadované úrovni. Zvládnout tento problém je velice obtížné. Pro zjednodušení byly vyvinuty normy, standardy a doporučení, jejichž použití zaručuje dobrý výsledek. Patří sem BS7799 (organizační norma), TCSEC (Orange book - první ucelená technická norma), ITSEC (mezinárodní sada kritérií) a Common Criteria (metanorma stanovující postupy pro odvození konkrétních technických norem). Použití norem také usnadňuje jednání s obchodními partnery a auditory.

32 Metody kontroly vstupu

Autorizovaní uživatelé mohou vkládat data, ale činí chyby, které musí SŘBD zachycovat a vyžádat si opravu. Metody:

- Field checks - test vhodnosti vkládaných dat: zda je to číslo, zda jde o jméno, ...

- Kontrola přístupu - mechanismus řešící, kdo co může měnit, jak naložit s kolizními případy, následnost úprav
- Log změn - záznam o všech provedených změnách, obsahuje původní a novou hodnotu
- Kontrola čtyř očí

33 Metody útoku po síti

V případě jednotlivých strojů může být útočníkem člověk. V prostředí sítí však již útočník může používat počítač a pokoušet se aktivně poškozovat systém ochrany dat.

- **Odposlech sítě** - Na základě odposlechu síťové komunikace může být útočník schopen vydávat se za někoho jiného. Může též přímo zachytávat senzitivní data, pokud nejsou dostatečně šifrována.
- **Playback starších zpráv** = pokusy o znovupoužívání starších zachycených zpráv. Vhodnou metodou ochrany jsou časová razítka v kombinaci s šifrováním, různé tokeny s omezenou časovou platností notarizace, nebo ofsetování zpráv.
- **Narušení služeb** - Velmi snadným způsobem útoku je přetěžování sítě nesmyslnými zprávami. Rovněž účinnou metodou je pokusit se pozměňovat routovací informace. Také je možné zachycovat nebo alespoň poškozovat zprávy zasílané určitému uživateli. Proti mnohým útokům je možno se bránit vytvořením duplicitních linek, po kterých mohou být zprávy posílány. Pokud je to možné, lze se omezit pouze na důvěryhodné uzly.
- **Vkládání poškozených zpráv** - Útočník může vkládat poškozené zprávy, při jejichž zpracování může dojít ke zhroucení službu konajícího stroje nebo k jeho nesprávné funkci.

34 Model pro komerční organizace

Potřebám komerčních organizací dobře vyhovuje např. Clark-Wilson model, který přejímá postupy běžné v účetnictví.

Základní principy Clark-Wilson modelu:

1. Dobře formované transakce (konzistentní data → konzistentní data)
2. Separace operací - žádnou operaci nesmí být schopen korektně provést jediný subjekt.

Pravidla modelu jsou dále rozdělena na požadavky na vynucení (E) a korektnost (C).

- E1 – Systém musí zajistit, že pouze procedury vyhovující požadavku C2 mohou pracovat s chráněnými objekty.
- E2 – Systém musí udržovat seznam relací popisující, který subjekt smí spouštět které transformační procedury a musí zajistit dodržování těchto relací.
- E3 – Systém musí autentizovat každý subjekt pokoušející se spustit transformační proceduru.
- E4 – Pouze administrátor provádějící certifikaci entit může provádět změny relací. V žádném případě nesmí mít právo spustit žádnou z procedur, které administruje.
- C1 – Všechny procedury testující validitu dat musí zajistit, že pokud doběhnou, všechna chráněná data jsou korektní.
- C2 – Všechny používané transformační procedury musí být certifikovány, že po zpracování korektních chráněných dat zanechají chráněná data opět v korektním stavu.
- C3 – Seznam popsaný v E2 musí splňovat pravidlo separace operací.
- C4 – Všechny transformační procedury musí zapisovat do append-only objektu (log) veškeré informace nezbytné pro rekonstrukci povahy provedené operace.
- C5 – Každá transformační procedura zpracovávající nechráněná data musí buď skončit s tím, že chráněná data jsou v korektním stavu, nebo nesmí provést žádnou změnu.

35 Očekávaná ztráta

Rozsah hrozeb spolu s pravděpodobností jejich realizace udává celkovou míru rizika. Očekávanou ztrátou rozumíme riziko, vztažené k určitému časovému období.

36 Odhad aktiv

Informační systém je tvořen souborem tzv. **aktiv**. Jejich společným cílem je poskytovat služby v požadované kvalitě. Mezi aktiva patří mimo jiné: záznamová media; počítače; tiskárny; programy; konfigurace; vlastní informace; sklad spisů; napájení; komunikační linky; administrátoři; uživatelé;

zálohy; provozní prostory. Svůj soupis aktiv si každý musí provést sám. Základem je zjistit, co vlastně ve svém informačním systému máme a k čemu je to dobré. Přesnější výsledek docílíme sčítáním po jednotlivých kategoriích.

- Hardware - počítače, monitory, pásky, tiskárny, disky, komunikační media
- Software - operační systém, koupené programy, vlastní zdrojové kódy
- Data - vlastní uložená data, logy, archivní kopie, listingy
- Lidé - pracovníci potřební k správnému chodu systému, správci, programátoři, technici
- Dokumentace - programů, technického vybavení, systému, administrativní postupy
- Spotřební materiál - papír, diskety, tonery, pásky do tiskáren

V podstatě v tomto kroku provedeme zevrubnou inventarizaci celého systému. Cena některých částí může být pouze velmi přibližně odhadnuta a i takový odhad může být velmi obtížný.

37 Odhad rizik

Provozování systému pro správu informací je vždy spojeno s určitým rizikem (př. chyba zařízení, obsluhy, programu, vandalismus, krádež). Přínos odhadu rizik:

- Zlepšení obecného povědomí - Pracovníci si problém uvědomí a mají šanci jej pochopit.
- Identifikace hodnot, slabin a možných kontrol celého systému - Ne vždy je jasné, které části systému mají největší hodnotu a odkud pramení největší nebezpečí.
- Zlepšení východiska pro strategická rozhodnutí - některé ochranné a kontrolní mechanismy velmi snižují produktivitu systému, přičemž jejich přínos není zřejmý, různé druhy nebezpečí jsou různě reálné a představují mnohdy daleko větší hrozbu, než by se dalo očekávat.
- Lepší rozložení výdajů na bezpečnost - některé velmi drahé ochranné mechanismy poskytují pouze malé zvýšení bezpečnosti a popřípadě i naopak.

Provedení odhadu rizik lze rozdělit do několika kroků:

1. Identifikace hodnot
2. Určení slabin
3. Odhad pravděpodobnosti zneužití
4. Výpočet očekávaných ročních ztrát
5. Přehled použitelných ochranných mechanismů
6. Nástin ročních úspor ze zavedení ochranných mechanismů

38 Ochrana objektů OS

Metody ochrany objektů v operačních systémech:

- Fyzická separace - Procesy pro vykonávání operací různého stupně utajení používají oddělená zařízení.
- Časová separace - Procesy různého stupně utajení jsou prováděny v různém čase.
- Logická separace - Operační systém zajišťuje oddělení jednotlivých procesů tak, že pro každý vytváří iluzi, že má celý počítač pro sebe.
- Kryptografická separace - Použitím kryptografických metod procesy provádějí ukrytí svých dat (např. sdílení komunikačních linek).

Je možná kombinace několika metod separace. Metody jsou řazeny dle rostoucí implementační složitosti a zároveň dle klesající spolehlivosti.

39 Ochrana paměti

Ochrana paměti je základním požadavkem pro zajištění bezpečnosti. Má-li být spolehlivá, je nutná hardwarová podpora. HW podpora navíc poskytuje dostatečnou efektivitu ochrany.

- **Ohrada** (fence) - Stanoví se hranice. Operační paměť na jednu stranu od této hranice používá OS, na druhou stranu aplikační programy. Metoda je vhodná pro jednoduché jednouživatelské systémy, umožňuje však pouze chránit OS. Nemůže být použita pro vzájemnou ochranu uživatelů většího systému. Implementace je velmi jednoduchá. Stroj má tuto hranici buď pevně zabudovanou nebo má tzv. *fence register*, jehož hodnotu porovnává s každou adresou, kterou aplikační program vygeneruje.

- **Relokace** - Programy jsou vytvořeny tak, jako by v paměti ležely od adresy 0. V rámci procesu spouštění programu je ke všem adresám v programu připočten *relokační faktor*. Aplikace tak nemůže zasahovat do oblastí, v nichž leží systém. Metoda má stejné nevýhody, jako předchozí způsob ochrany paměti.

- **Base/Bound registry** - Realizují přenesení myšlenek předchozích metod do prostředí multiuživatelských systémů. K adresám generovaným programem je připočítávána hodnota bázového registru. Každý odkaz je porovnáván s hodnotou bound registru, zda je menší. Program má tak shora i zdola omezen prostor, v němž může pracovat. Metoda umožňuje vzájemné oddělení jednotlivých uživatelů, nechrání však kód aplikačního programu před chybou. Možným rozšířením je používat dva páry registrů, jeden pro vymezení oblasti pro kód procesu a druhý pro datovou zónu. Nevýhodou je nemožnost selektivního sdílení pouze některých dat.

- **Značkovaná (Tagged) architektura** - S každou adresou (slovem) v paměti stroje je spojeno několik tag bitů, jejichž obsah určuje typ zde uložených dat a povolené operace. Obsah tag bitů je testován při každém přístupu k obsahu této adresy. Tag bity mohou být měněny pouze privilegovanými instrukcemi. Alternativou může být používání jednoho tagu pro celý blok slov.

- **Segmentace** - Celý program sestává z několika bloků (segmentů), které mohou být nezávisle uloženy do paměti. Program potom generuje odkazy ve tvaru <jméno segmentu><ofset>. Jméno segmentu je pomocí systémem udržovaného segmentového adresáře převedeno na adresu počátku segmentu, ke které je přičten ofset. Často je též ofset porovnán s velikostí segmentu, aby se zajistilo, že program nesahá "za segment". Metoda již poskytuje dostatečné prostředky pro sdílení dat a navíc umožňuje ochranu kódu programu, případně i vybraných dat. Rovněž je schopna chránit uživatele navzájem.

- **Stránkování** - Metoda je velmi podobná segmentaci, jen předpokládáme segmenty konstantní velikosti (= **stránky**). Opět používáme dvousložkové adresování <číslo stránky><ofset>. Ochrana proti adresování za stránku je vyřešena samovolně tím, že nelze udělat větší ofset, než je velikost stránky. Možnost ochrany obsahu stránek je poněkud slabší než v případě segmentů, neboť není příliš jasná vzájemná souvislost obsahů stránek a rozdělení programu a dat do stránek.

40 Ochrana perimetru sítě

Od dávných dob vytvářeli lidé kolem svých sídel obranné prvky v podobě hradeb nebo vodních příkopů. Jejich cílem bylo jednak odolat případnému nájezdu útočníků a jednak přimět cizince, aby dovnitř vstupovali pouze branami, tedy místy, kde se dal jejich pohyb kontrolovat. Stejné principy jsou používány při zabezpečování počítačových sítí. Komunikaci mezi nedůvěryhodným vnějším prostředím a vnitřní chráněnou sítí řídí firewall. Pokud si to síťový provoz směřuje dovnitř otevřenými branami v podobě otevřených portů firewallu jsou jeho čisté úmysly ještě dále důkladně prověřeny pomocí síťových systémů detekce a prevence narušení a síťových antivirových bran, případně dalších síťových bezpečnostních zařízení. Teprve potom je vpuštěn. Stejně tak je důležité monitorovat a filtrovat provoz směrem ven ze sítě, aby nedocházelo k nekontrolovaným únikům senzitivních dat. Ochrana perimetru sítě však neřeší problém útoku zevnitř sítě, který je statisticky častější.

41 Ochrana prostoru

Základem prostorové ochrany je **princip plotu**. Veškerý pohyb přes tento plot by měl být monitorován a autorizován. Musíme mít prostředky chránící prostor s klíčovými komponentami systému před vstupem potenciálního útočníka, případně před jinými nežádoucími událostmi (vnášení nebezpečných předmětů, vynášení senzitivních informací, krádeže zmíněných komponent atd.). Prostorovou ochranou se také rozumí schopnost detekovat skutečnost, že k podobnému bezpečnostnímu incidentu došlo.

Předmětem útoku mohou být: počítače (nebo jejich části), záznamová média, části síťové technologie, klíčoví pracovníci.

Metody prostorové ochrany:

- **Stráž** - Je nepřetržitě k dispozici. Zná všechny pracovníky nebo je schopna je identifikovat např. podle tokenu. Provádí záznam o pohybu všech osob. Alternativou (nebo spíše doplňkem) jsou turnikety nebo přechodové prostory s identifikačními mechanismy osob nebo tokenů).
- **Elektronická ochrana**
  - dveřní a okenní kontakty detekují otevření
  - otřesové hlásiče - detekují rozbití nebo proražení střežené plochy (skla, příčky, přepážky, . . .)
  - vodičové desky, drátěné sítě - slouží k detekci průrazů ve stěnách, podlahách apod.
  - kontaktní matice - při instalaci pod podlahové krytiny slouží k detekci vstupu osob do chráněného prostoru

- mikrovlnné, ultrazvukové, infračervené detektory - reagují na změnu resp. přerušení svazku příslušného záření
- zvukové hlásiče - reagují na specifické zvuky jako řezání, vrtání, šroubování apod.
- kyvadlové hlásiče - reagují na otřesy a vychýlení z původní roviny
- k ochraně jednotlivých předmětů lze použít obrazových vah, závěsů apod.
- vhodným prostředkem v mnoha případech je průmyslová televize, zejména v kombinaci se záznamem snímaného obrazu

42 Ochrana proti elektromagnetickému vyzařování

Elektromagnetické vyzařování je způsobeno změnou proudu ve vodiči. Mezi problematické součásti systému patří především výpočetní technika a přenosové linky. Informace mohou unikat i naindukováním v napájecích vodičích. Odposlech elmag. vyzařování většinou není kriminalizován. Způsoby ochrany mohou být následující:

- Vzdálenost - Intenzita záření klesá se čtvercem vzdálenosti.
- Zmatení - Množství podobných signálů komplikuje případný odposlech. Rušivý signál je možné generovat uměle nebo umístit větší množství podobně vyzařujících zařízení na jedno místo.
- Speciální vybavení - Speciálně vyvinuté součásti, jejichž vyzařování nepřekračuje určitou mez.
- Vhodné umístění - Umístění do prostor, zabraňujících šíření elmag. záření (speciální odstíněné schránky nebo i celé místnosti).

43 Personální politika

Soubor pravidel v oblasti vztahů organizace a zaměstnanců, ovlivňujících efektivnost zaměstnanců a tím i celé organizace. Tato pravidla se uplatňují např. při přijímání nových zaměstnanců. Provoz bezpečnostního systému může záviset na malém počtu lidí, kteří ho umí udržovat. Tito lidé by měli být vybráni s opatrností přímo úměrnou vzhledem k významu jejich budoucí pozice. Z pohledu bezpečnosti je kladen důraz především na důvěryhodnost zaměstnanců.

44 PKCS

Public Key Cryptography Standarts je ucelený soubor technických norem popisující implementaci různých nástrojů asymetrické kryptografie. Tyto

normy jsou spravovány a aktualizovány laboratořemi firmy RSA Security. Obsahují např. D-H algoritmus (PKCS#3) nebo RSA algoritmus (PKCS#1).

45 Problém odvoditelnosti

Jde o možnost odvodit senzitivní informace ze znalosti (velkého množství) nesenzitivních.

- Přímý útok - Útočník se snaží získat informace přímými dotazy, jejichž odpověď závisí na velmi malém počtu vět, které vyhověly podmínkám dotazu. Takový dotaz může obsahovat velké množství uměle vložených nesplnitelných podmínek.
- Nepřímý útok - Často je z databáze obsahující např. senzitivní osobní údaje povoleno zveřejňovat statistické údaje, které není třeba považovat za tajné. Z těchto údajů lze ovšem za vhodných okolností získat původní utajované informace.
- Součet - Vhodný dotaz na součet určitých položek může vést k vyzrazení utajované položky. (Součet jedné položky je sama tato položka.)
- Počet - Dotaz na počet může být kombinován s dotazem na součet.
- Medián - Utajované hodnoty lze získat z dotazu na medián.
- Tracker attack - Účinnou obranou je, pokud správce databáze odepře odpověď na dotazy, jejichž výsledek závisí na malém množství záznamů. Útočník však může získat informace porovnáním výsledků několika dotazů. Odečtením výsledků pak získá výsledek dotazu, který správce nevydal.

45.1 Ochrana proti odvoditelnosti

- Rozbor dotazů i s ohledem na minulost - velice komplikovaný, nákladný a málo spolehlivý, efektivní pouze proti přímým útokům.
- Ochrana vlastních dat - pasivní ochrana, hlavními metodami jsou potlačení (suppression) a skrytí (concealing) výsledků.
  - Potlačení - systém nevydá odpověď na všechny dotazy, ale případné odpovědi jsou přesné.
  - Skrytí - systém odpovídá na všechny dotazy, ale odpovědi jsou záměrně nepřesné.

46 Průnik do OS

1. Místem největšího počtu průniků je mechanismus zpracování I/O operací.
   - Mnohá I/O zařízení jsou do značné míry inteligentní a nezávislá na zbytku systému. Provádějí optimalizaci své činnosti. Jejich řadiče často spravují více takovýchto zařízení.
   - Kód I/O operací je často velmi rozsáhlý, je těžké jej řádně testovat a někdy je dokonce nutné používat kód dodaný výrobcem zařízení.
   - V zájmu rychlosti a efektivity I/O operace občas obcházejí bezpečnostní mechanismy operačního systému, jako stránkování, segmentaci apod.
   - Velká část I/O operací je znakově orientovaná. V zájmu efektivity se často příslušné kontroly neprovádějí s každým přijatým znakem, ale pouze při startu operace.
2. HW současných procesorů poskytuje rozsáhlé prostředky pro vytvoření kvalitní bezpečnosti (úrovně oprávnění, privilegované instrukce, trasovací režimy, systém obsluhy výjimek, systém ochrany segmentů a stránek, ...). Ne vždy jsou však využívány.
3. Dalším problémem je hledání kompromisu mezi důkladnou izolací uživatelů a nutností umožnit sdílení dat. Tento kompromis zhusta bývá obtížně formalizovatelný. Nejasnosti návrhu pak mohou být příčinou "děr" v implementaci.
4. Ne vždy je možné provádět kontroly oprávněnosti s každou operací. Často je kontrola prováděna pouze jednou během provádění celého bloku akcí. Pokud se v této době uživateli podaří změnit parametry, může dojít k průniku.
5. Další skulinu v bezpečnosti může způsobit snaha o obecnost možného nasazení systému. Aby bylo možno systém používat pro nejrůznější úkoly, ponechají návrháři často mechanismus, pomocí kterého si uživatel může systém přizpůsobit. Tento mechanismus může být zneužit.

47 Rozdíl hrozba vs. riziko

Hrozba existuje nezávisle na zavedených bezpečnostních opatřeních. Riziko je pravděpodobnost, že dojde k naplnění určité hrozby. Můžeme ho snížit pomocí vhodných bezpečnostních opatření.

48 Senzitivita informací

Za senzitivní považujeme takové informace, které by neměly být veřejně známé. Lze rozlišovat několik stupňů senzitivity (např. Přísně tajné, Tajné, Důvěrné, Vyhrazené, Neklasifikováno). Přístup k senzitivním informacím by měly mít pouze osoby s patřičným stupněm prověření.

Důvody senzitivity informací:

- Přirozeně senzitivní - Informace sama o sobě je utajovaná (např. výše platů některých státních zaměstnanců).
- Ze senzitivního zdroje - např. z informace je patrné, kdo ji poskytl.
- Deklarované jako senzitivní - Informace jsou prohlášené správcem nebo majitelem databáze za utajované.
- Senzitivní atribut nebo záznam - V DB může být určitý řádek nebo sloupec prohlášen za tajný.
- Senzitivní ve vztahu k dříve vyzrazeným skutečnostem - např. byla-li vyzrazena zeměpisná šířka utajované vojenské základny → údaj o z. délce je senzitivní.

49 Šifrování komunikace

Šifrovaná komunikace se používá za účelem utajení obsahu zpráv procházejících nezabezpečenou cestou.

Základními pojmy:

- Zašifrování - proces zakódování zprávy tak, aby její obsah nebyl zřejmý.
- Dešifrování - opačný proces k zašifrování.
- Kryptosystém - systém umožňující šifrovat a dešifrovat zprávy.
- Otevřený text - originál nezašifrované zprávy (plain text).
- Šifrovaný text - zašifrovaná zpráva.

K provedení procesu šifrování a dešifrování je zpravidla nutná znalost **klíče**. Mezi hlavní problémy spolehlivého kryptosystému patří utajení šifrovacího klíče. Mezi nejrozšířenější kryptosystémy patří D-H algoritmus a RSA algoritmus. Bezpečnost šifrovacích metod využívajících šifrování pomocí klíče je založena na výpočetní složitosti různých matematických problémů (rozklad na prvočinitele, výpočet logaritmů nad GF(p)). Jediná šifrovací metoda, jejíž neprolomitelnost byla exaktně dokázána, je tzv. *One-time pad* algoritmus využívající klíč o stejné délce jakou má přenášená zpráva (to činí

tuto metodu značně nepraktickou). Aby bylo možné využívat bezpečnou šifrovanou komunikaci, je nutná existence nějaké důvěryhodné autority, která zajistí bezpečnou výměnu klíčů (ochrana proti tzv. man-in-the-middle attacku[1]).

Šifrování v rámci sítě lze provádět na různých úrovních:

- Na úrovni linky (Link encryption) - Data jsou šifrována těsně před vstupem do komunikačního média a dešifrována těsně po výstupu. Tento proces je pro uživatele transparentní. Další výhodou tohoto způsobu je rychlost a snadná přenositelnost.
- End-to-End šifrování - Šifrování probíhá na úrovni aplikační, nebo prezentační vrstvy referenčního modelu. Není transparentní a je třeba ho do systému vhodně zakomponovat. Výhodou je možnost šifrovat pouze vybranou komunikaci. Tento způsob dokáže zajistit autentizaci a integritu.

Se zavedením šifrování souvisí nutnost existence mechanismu distribuce a správy nezbytných šifrovacích klíčů, potřebných centrálních autorit pro zajištění provozu systému kryptografické ochrany a vhodných kryptografických zařízení zajišťujících základní funkce kryptografické ochrany.

50 Spojení dvou sítí

- Gateway
- Filtrování datového toku
- Spojení sítí přes veřejné komunikační sítě

51 Spolehlivá síťová komunikace

51.1 Spolehlivé síťové rozhraní (trusted network interface)

Každý uzel sítě musí být "opatrný" vůči ostatním uzlům. Měl by zajistit, že spojení naváže pouze s dalším uzlem, který má spolehlivé síťové rozhraní.

Funkce spolehlivého síťového rozhraní:

1. Zajištění bezpečnosti vlastního uzlu před útoky zvenčí.
2. Veškerá výstupní data musí být označena příslušnou bezpečnostní klasifikací.
3. Před uvolněním dat je provedena verifikace oprávněnosti žadatele a jeho autentizace.

[1] "Man in the middle" je útočník schopný dešifrovat, změnit a přeposlat zprávu bez vědomí obou komunikujících stran.

4. Ověření konzistence došlých dat.
5. Nesmí docházet k míchání dat různého stupně utajení nebo k samovolnému předávání informací ostatním uzlům.
6. Bezpečnost dat nesmí záviset na bezpečnosti linky.

51.2 Bezpečná komunikace

Vlastní síť včetně příslušných řídících modulů není uvažována jako bezpečná. Bezpečnou komunikaci zajišťují samy komunikující procesy ve spolupráci s operačním systémem. Jsou dodržována pravidla Bell-LaPadula bezpečnostního modelu. Pokud chceme zavést potvrzování zpráv, je nutné, aby na každém uzlu běžel pro každou bezpečnostní úroveň komunikační server, který, posílá-li proces zprávu procesu vyšší úrovně na jiném uzlu, zašle ji tamějšímu kom. serveru své úrovně, od kterého obdrží potvrzení a který ji předá. Posílání zpráv procesům nižší úrovně probíhá prostřednictvím spolehlivé centrální autority (network manažera,) který zkoumá, zda nedochází k únikům klasifikovaných informací.

51.3 Bezpečné síťové spojení

Spolehlivá síťová rozhraní rozdělíme na moduly se vstupnímy a výstupnímy sokety. Pokud u daného modulu můžeme dokázat, že jeho výstupy závisí pouze na některých vstupech → multilevel modul - může mít výstupní sokety různých úrovní citlivosti. Jinak má modul výstupy odpovídající nejvyšší citlivosti vstupu. Opět budeme dbát na zachování pravidel Bell-LaPadula modelu, tzn. výstup modulu může být připojen pouze na vstup jiného s nejméně stejným stupněm citlivosti. Každý proces prohlásíme rovněž za modul se specifickým stupněm citlivosti. Takto lze definovat povolená spojení v rámci celé sítě.

52 Spolehlivý front-end

Tato otázka je zpracována v rámci otázky *Víceúrovňová bezpečnost v DB*.

53 Správa verzí (konfigurací)

Cílem je zajištění dostupnosti a používání správných verzí software.

Hlavní funkce:

- zajištění integrity programů a dokumentace + vyhodnocování a zaznamenávání provedených změn
- prevence proti úmyslným změnám odzkoušených programů (vkládání trapdoors, logických bomb)

Důvody pro zavedení správy verzí:

- Zabraňuje nechtěným ztrátám předchozích verzí software
- Odstraňuje komplikace při vývoji několika podobných verzí (např. pro různé platformy) zároveň
- Poskytuje mechanismus pro kontrolované sdílení modulů, z nichž je skládán vytvářený systém

Za účelem udržení přehledu je nutné vést důkladnou dokumentaci všech kopií. Správou konfigurací se zpravidla zabývá vyčleněný pracovník. Programátor pracuje na vlastním exempláři modulu, který po ukončení etapy předá správě konfigurací včetně popisu provedených úprav a celkové charakteristiky a dále již v něm nemůže činit změny. Je vhodné, aby správce konfigurací přijímal programy výhradně ve zdrojové formě se soupisem a popisem provedených změn. Nutné vést detailní log co, kdo, kdy dělal.

54 Srovnání jedno- a víceúrovňových modelů

Jednoúrovňové modely jsou vhodné případy, kdy stačí jednoduché ano/ne rozhodování, zda danému subjektu poskytnout přístup k požadovanému objektu a není nutné pracovat s klasifikací dat.

Obecně však může být několik stupňů senzitivity a oprávněnosti. Tyto stupně senzitivity se dají použít k algoritmickému rozhodování o přístupu daného subjektu k cílovému objektu, ale také k řízení zacházení s objekty. Víceúrovňové modely "rozumí" senzitivitě dat a chápou, že s nimi musí zacházet v souladu s požadavky kladenými na daný stupeň senzitivity. Rozhodnutí o přístupu pak nezahrnuje pouze prověření žadatele, ale též klasifikaci prostředí, ze kterého je přístup požadován.

55 Statické metody testování

Přímo zkoumají struktury a formu produktu bez jeho spuštění (reviews, inspekce, data-flow).

Příklady statických metod:

- Analýza algoritmů - Zpětným přepisem do běžného jazyka nebo formalismu je ověřována logika a správnost návrhu. Zahrnuje znovuodvození vztahů, vhodnost návrhu, stabilitu, časování, přetypování, ...
- Čtení kódu - Expertní pročítání cizího kódu.
- Rozhodovací tabulky - Analýza komplexních logických vztahů a rozhodování.
- Analýza rozhraní - Ověření správnosti rozhraní modulů, reakcí na vstupy, dodržení norem.

56 TCSEC

TCSEC = Trusted Computer Security Evaluation Criteria, neboli Orange Book.

- První ucelená technická norma, tvůrce Ministerstvo obrany USA
- Systémy jsou rozděleny do čtyř základních tříd a dále na podtřídy.
  - **Třída D** - žádná ochrana
  - **Třída C** - Optional protection - ochranný mechanismus musí být k dispozici, ale uživatel zvolí zda ho chce používat
    - Třída C1 (volná ochrana)- Musí existovat metody umožňující uživatelům chránit vlastní data před ostatními.
    - Třída C2 (kontrolovaný přístup) - Granularita musí být až na úroveň jednotlivých uživatelů. Vedení access logu je povinné. Nutná je ochrana proti *residuím* (obsah paměti, registrů apod. nesmí být poté, co je proces přestane používat přístupný nikomu jinému).
  - **Třída B** - Mandatory protection - odpovídající mechanismus musí být k dispozici a uživatel ho nemůže obejít ani deaktivovat.
    - Třída B1 (značkovaná ochrana) - Každý kontrolovaný objekt a subjekt musí mít přiřazen stupeň utajení (a být tímto stupněm označen). Každý přístup musí být ověřen dle Bell-LaPadula modelu. Musí existovat popis implementovaného formálního modelu. Systém je podrobován testování.
    - Třída B2 (strukturovaná ochrana) - Musí být k dispozici verifikovatelný globální návrh systému. Systém musí být rozdělen do dobře definovaných nezávislých modulů. Návrh zohledňuje princip nejmenších možných oprávnění. Bezpečnostní mechanismy musí být uplatňovány vůči všem objektům a subjektům. Musí existovat analýza možných skrytých kanálů. Vlastní systém musí provádět kontroly své integrity a běžet v rámci své bezpečnostní domény.
    - Třída B3 (bezpečnostní domény) - Systém musí být podrobitelný testování. Musí existovat úplný popis návrhu systému (jednoduchý). Každý přístup k objektu musí být testován (na úrovni typu přístupu). Systém musí být vysoce odolný vůči průnikům.
  - Třída A1 (verifikovaný návrh) - Návrh systému musí být formálně verifikován. Existuje formální model bezpečnostního mechanismu s důkazem konzistentnosti; formální specifikace systému s ověřením, že odpovídá formálnímu modelu; ověření, že implementace není odchylná od formální specifikace; analýza skrytých kanálů.

57 Testy - životní cyklus

Je obtížné prokazovat správnost programu. To, že nebyly nalezeny chyby, může pouze znamenat, že byla použita nevhodná metodika testování. Testování by měl provádět nezávislý tým. Snižuje se tak nebezpečí, že výsledný program obsahuje nežádoucí kód, či že nebyl dodržen původní cíl projektu.

Testování během vývoje programu:

1. Validace požadavků na software - ověření, zda požadavky nejsou v rozporu s platnými standardy, zda nejsou vnitřně sporné.
2. Verifikace a validace návrhu - ověření, jestli návrh úplně a bezchybně implementuje požadavky.
3. Verifikace a validace kódu - ověření, použití předepsaných standardů a postupů vývoje.
4. Testování - (modulů, integrace, systému, instalace) na jednotlivých úrovních se zkoumá, jestli se výsledný program chová přesně podle zadání, zvláště pod tlakem.
5. Verifikace a validace při používání programu - co dělat při změně v systému.

58 Tokeny a autentizace

Token je předmět, který pomáhá autentizovat svého nositele. Zpravidla pouhé předložení tokenu k autentizaci nestačí - může být spojeno se zadáním PINu nebo hesla. Token by měl být jedinečný a nepadělatelný. Obvyklou implementací jsou magnetické nebo čipové karty. Nevýhody jsou přenositelnost, statičnost.

Typy tokenů:

- Pouze s pamětí - obdoba mechanického klíče (obsahem paměti je jednoznačný identifikační řetězec).
- Tokeny s heslem - po zadání jednoduchého hesla vygenerují dlouhý klíč (příklad silného bezpečnostního mechanismu chráněného slabším bezpečnostním mechanismem).
- Tokeny s logikou - Jsou schopné zpracovat jednoduché podněty. Např. vydat následující klíč nebo cyklickou sekvenci klíčů. Pomocí takových tokenů lze implementovat systém s one-time hesly.
- Inteligentní tokeny (smart cards) mohou disponovat vlastním vstupním zařízením pro komunikaci s uživatelem, zajišťovat šifrování, generovat náhodná čísla. Jsou ideálním doplňkem Challenge-Response systémů (autentizace pomocí dotazu a odpovědi na něj).

59 Úrovně ochrany objektu

Tato otázka se skládá z otázek *Autorizace* a *Granularita*.

60 Útoky prostřednictvím programů

Programy mohou poškodit či zcela zničit data, způsobit kompromitaci utajovaných skutečností, omezit a popřípadě vyloučit funkčnost celého systému.

- **Trapdoors** - Tímto pojmem označujeme nedokumentovaný vstup do programového modulu. Nejčastěji bývá vytvořen v rámci tvorby tohoto modulu za účelem snazšího ladění (doplnění příkazu na ladící tisky apod.). Může se stát, že program má špatně ošetřené vstupy a na nepřípustných datech se nechová korektně. Trapdoors jsou obvykle odstraněny před dokončením modulu, ale mohou být opomenuty, ponechány za účelem ladění dalších modulů, či snažší správy dokončeného programu. V neposlední řadě mohou být ponechány záměrně pro získání neoprávněného přístupu k běžícímu programu.

- **Trojský kůň** - Je to program, který kromě svých "řádných" funkcí vykonává ještě další skryté akce. Objevit Trojského koně v programu o stovkách tisíc či miliónech řádků je velmi obtížné, navíc v tomto smyslu může být program upraven až následně po otestování a zařazení do provozu.

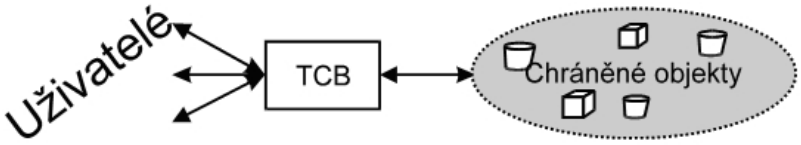
- **Salámový útok** - Jde o programy, které se snaží využívat ve svůj prospěch malých zaokrouhlovacích chyb na hranici přesnosti počítače. Salámový útok je opět velmi těžko detekovatelný, neboť bývá prováděn v rozsáhlých SW systémech, navíc nepůsobí viditelné problémy.

- **Skryté kanály** - V prostředích spravujících klasifikované informace zpravidla aplikační programátoři po ukončení vývoje nemají přístup k běžícím programům. Chtějí-li získat přístup ke spravovaným informacím, vytvoří skrytý kanál. Implementace je naprosto obecná: zdánlivé chyby na výpisech, existence či neexistence specifických souborů, vznik jistých systémových událostí, nepatrné změny front-endu. Jsou zvláště vhodné pro únik malého množství informací, opět prakticky nedetekovatelné.

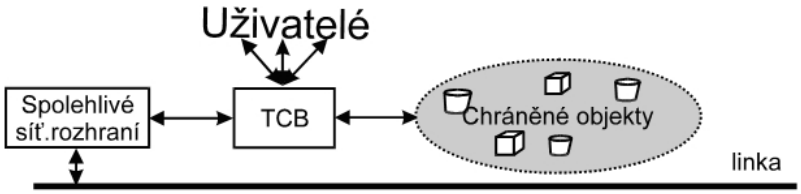
- **Hladové programy** - Na mnoha počítačích běží s velmi nízkou prioritou programy, které vykonávají nejrůznější zdlouhavé výpočty, které "nepospíchají". Nechtěné či úmyslné zvýšení priority může znamenat zahlcení celého systému. Dalším případem jsou procesy generující velké množství synovských procesů (mnohdy chyba programu), programy běžící v nekonečné smyčce. Operační systémy často obsahují obranné

mechanismy, které násilně ukončí programy, jež běží příliš dlouho. V době provádění I/O operace neběží virtuální čas procesu, tedy procesy generující velmi velké množství I/O operací mohou běžet takřka neomezeně dlouho.

- **Viry** jsou malé programy s autoreprodukční schopností.
  - Zpravidla se připojí nebo nahradí část kódu napadeného programu. Při spuštění tohoto programu se nejprve provede kód viru, který se nainstaluje do paměti a převezme nebo pozmění některé funkce systému.
  - Viry často obsahují obranné mechanismy proti detekci, jsou schopny instanci od instance podstatným způsobem měnit vlastní kód, po nainstalování do paměti provedou operace, jež ostatním programům učiní použitou oblast paměti nedostupnou.
  - Velmi často po určitou dobu pouze provádějí reprodukci bez jakýchkoliv vedlejších projevů. V době odhalení tak může být napadena drtivá většina programů.
  - Šíření virů částečně omezuje nástup systémů poskytujících ochranu paměti a dokonalejší správu prostředků. Tyto systémy však bývají komplikované a dokáží tak poskytnout úkryt mnohem komplexnějším virům.
  - Následky virové nákazy mají nejrůznější podobu - od převážně neškodných zvukových či obrazových efektů, které pouze rozptylují obsluhu a zatěžují počítač, až po rozsáhlá poškození či zničení veškerých dat a programů.
  - Podmínkou šíření virů je neopatrná manipulace s programovým vybavením, přenášení většinou nelegálního software atp.
  - Dobrou obranou je kromě proškolení personálu též rozdělení veškerých programů a souvisejících dat do oddílů, které jsou navzájem dostatečně odděleny, tak aby nemohlo docházet k šíření virů z jednoho oddílu do druhého.
- **Červy** jsou síťovou obdobou virů. Mají schopnost se šířit prostřednictvím komunikačních linek z jednoho počítače na druhý. Obecně vzato mají stejné zhoubné účinky jako viry, avšak díky schopnostem samovolně se šířit v dnes již celosvětovém měřítku je jejich expanze daleko rychlejší (hodiny) a dopad tedy daleko větší. Obranou je rovněž kvalitní správa programového vybavení, používání pouze dobře otestovaných programů a rozdělení sítě na domény, mezi kterými dochází k minimálnímu sdílení informací, které je navíc podrobeno důkladné kontrole.



Obrázek 4: TCB



Obrázek 5: TCB v síti

- **Logická bomba**- kus kódu, který se spustí při dosažení specifických podmínek (konkrétní datum, dosažený obrat společnosti..) a nějakým způsobem poškodí systém, nebo uložená data.

61 Vícefaktorová autentizace

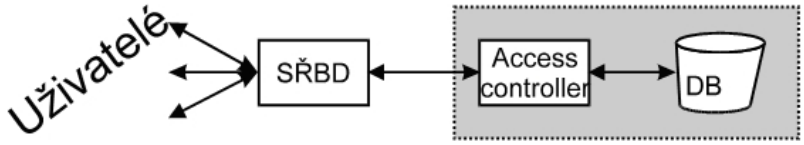
Kombinací několika autentizačních postupů (např. pin + smart karta) získáme vyšší úroveň bezpečnosti.

- Několik nezávislých bezpečnostních mechanismů aplikovaných paralelně.
- Aktivace silnějšího mechanismu a následná autentizace za použití tohoto mechanismu.

62 Víceúrovňová bezpečnost obecně

63 Víceúrovňová bezpečnost v sítích

Rovněž v počítačových sítích mohou pracovat uživatelé s různým stupněm prověření. Síť obsahuje data různých stupňů utajení. Nejčastěji se používá nějaká modifikace military security modelu. Operační systémy, navrhované pro vysokou bezpečnost bývají rozděleny na moduly, na jejichž bezpečnost nejsou kladeny nároky, které přistupují k chráněným objektům prostřednictvím spolehlivých modulů, jež tvoří spolehlivou výpočetní bázi (TCB = Trusted Computing Base viz. obrázek **??**). Zařazení principu TCB do sítí ukazuje obrázek **??**.



Obrázek 6: Access controller

64 Víceúrovňová bezpečnost v DB

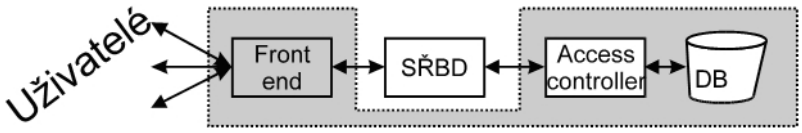
- **Parcelizace** (partitioning) - Databáze je rozdělena dle stupně citlivosti informací na několik subdatabází, což vede ke zvýšení redundance s následnou ztíženou aktualizací a neřeší problém nutnosti současného přístupu k objektům s různým stupněm utajení.

- **Šifrování** - Senzitivní data jsou chráněna šifrováním před náhodným vyzrazením. Zná-li útočník doménu daného atributu, může snadno provést chosen plaintext attack (zašifrováním všech hodnot z domény). Těžko totiž někoho zbavíme znalosti šifrovacího klíče. Řešením je používat jiný klíč pro každý záznam, což je však poměrně náročné. V každém případě nutnost neustálého dešifrování snižuje výkon systému.

- **Integrity lock** - Každá položka v databázi se skládá ze tří částí: < vlastní data : klasifikace : checksum >. Vlastní data jsou uložena v otevřené formě. Klasifikace musí být nepadělatelná, nepřenositelná a skrytá, tak aby útočník nemohl vytvořit, okopírovat ani zjistit klasifikaci daného objektu. Checksum zajišťuje svázání klasifikace s daty a integritu vlastních dat. Model byl navržen jako doplněk (access controller) komerčního SŘBD, který měl zajistit bezpečnost celého systému. Šedá oblast na obrázku **??** vyznačuje bezpečnostní perimetr systému. Access controller nevidí na výstup databáze a není schopen zajistit, na výstupu označovala data stupněm senzitivity.

- **Spolehlivý front-end** (guard) - Systém je opět zamýšlen jako doplněk komerčních SŘBD, které nemají implementovánu bezpečnost. Uživatel se autentizuje spolehlivému front-endu, který od něho přebírá dotazy, provádí kontrolu autorizace uživatele pro požadovaná data, předává dotazy k vyřízení SŘBD a na závěr provádí testy integrity a klasifikace výsledků před předáním uživateli. SŘBD přistupuje k datům prostřednictvím spolehlivého access controlleru. Tento způsob ukazuje obrázek **??**.

- **Komutativní filtr** (Commutative Filter) - Jde o proces, který přebírá úlohu rozhraní mezi uživatelem a SŘBD. Filtr přijímá uživatelovy dotazy, provádí jejich přeformulování a upravené dotazy posílá SŘBD k vyřízení. Z výsledků, které SŘBD vrátí, odstraní data, ke kterým



Obrázek 7: Spolehlivý front-end

uživatel nemá přístupová práva a takto upravené výsledky předává uživateli. Filtr je možno použít k ochraně na úrovni záznamů, atributů a jednotlivých položek. V rámci přeformulování dotazu může např. vkládat další podmínky do dotazu, které zajistí, že výsledek dotazu závisí jen na informacích, ke kterým má uživatel přístup.

- **Pohled** (View) - Pohled je část databáze, obsahující pouze data, ke kterým má daný uživatel přístup. Pohled může obsahovat i záznamy nebo atributy, které se v původní databázi nevyskytují a vznikly nějakou funkcí z informací původní databáze. Pohled je generován dynamicky, promítají se tedy do něho změny původní DB. Uživatel klade dotazy pouze proti svému pohledu - nemůže dojít ke kompromitaci informací, ke kterým nemá přístup. Záznam / atribut původní databáze je součástí pohledu, pokud alespoň jedna položka z tohoto záznamu / atributu je pro uživatele viditelná, ostatní položky v tomto jsou označeny za nedefinované. Uživatel při formulování dotazu může používat pouze omezenou sadu povolených funkcí. Tato metoda je již návrhem směřujícím k vytvoření bezpečného SŘBD.

65 Víceúrovňové modely

V jednoúrovňových modelech jsme měli jednoduché vztahy (objekt je/není senzitivní, subjekt má/nemá přístup k danému objektu). Obecně však může být několik stupňů senzitivity a "oprávněnosti". Tyto stupně senzitivity se dají použít k algoritmickému rozhodování o přístupu daného subjektu k cílovému objektu, ale také k řízení zacházení s objekty. Víceúrovňový systém "rozumí" senzitivitě dat a chápe, že s nimi musí zacházet v souladu s požadavky kladenými na daný stupeň senzitivity (např. tajná data ukládat pouze na konkrétní diskové pole, přísně tajná data posílat mimo systém výhradně zašifrovaná HW šifrátorem, . . .). Rozhodnutí o přístupu pak nezahrnuje pouze prověření žadatele, ale též klasifikaci prostředí, ze kterého je přístup požadován (tj. uživatel je prověřen na vyhrazená data, ale sedí u stanice, která nemá klasifikaci "na vyhrazená data" a tudíž přístup není povolen).

- Military security model
- Svazový model (Lattice model)

- Bell-LaPadula model
- Biba model
- Clark-Wilson model
- Chinese wall model
- Graham-Denning model
- Take-Grant system

66 Vlastnosti bezpečného OS

Na kvalitě operačního systému závisí bezpečnost celého mechanismu ochrany dat. OS kontroluje chování uživatelů a programů a v konečném důsledku zpřístupňuje utajované informace. Proces vývoje bezpečného OS lze rozdělit do několika fází:

- Bezpečnostní modely - Vytvoří se formální modely prostředí a zkoumají se způsoby, jak v tomto prostředí zajistit bezpečnost.
- Návrh - Po zvolení vhodného modelu je hledán vhodný způsob implementace.
- Ověřování - Je třeba ověřit, že navržená implementace skutečně odpovídá teoretickému modelu.
- Implementace - praktické a důkladné provedení shora uvedených teoretických úvah.

Implementace bezpečnostních mechanismů je v přímém rozporu s efektivitou systému OS vykonává několik s bezpečnostní úzce souvisejících činností:

- Autentizace uživatelů
- Ochrana paměti - mezi uživateli i v rámci jednoho uživatelského prostoru
- Řízení přístupu k souborům a I/O zařízením - ochrana před neautorizovaným přístupem
- Alokace a řízení přístupu k obecným objektům - zajištění bezproblémového současného přístupu více uživatelů k stejnému objektu
- Zabezpečení sdílení - zejména zajištění integrity a konzistentnosti
- Zajištění spravedlivého přístupu - o HW prostředky se opírající mechanismus zajišťující, že všichni uživatelé dostávají přidělen procesor a ostatní systémové zdroje

- Meziprocesová komunikace a synchronizace - systém poskytuje mechanismus pro bezpečné předávání zpráv mezi procesy, procesy nekomunikují přímo ale via systém

Bezpečnost musí být brána v potaz ve všech aspektech návrhu systému a musí být zapracována již v prvotním návrhu. Je velmi obtížné ji "přidat" do hotového návrhu. Následující principy je vhodné mít na paměti:

- Nejmenší práva - Každý subjekt by měl mít pouze nezbytná práva.
- Ekonomický návrh - Bezpečnostní systém má být malý a jednoduchý, pak je testovatelný a věrohodný.
- Otevřený návrh - Bezpečnostní mechanismus by měl být veřejně známý (a oponovaný) a měl by záviset na bezpečnosti co nejméně objektů.
- Úplné zprostředkování - Veškeré přístupy k objektům zprostředkovává a testuje OS.
- Povolování operace - Co není výslovně povoleno, je zakázáno.
- Rozdělené oprávnění - Přístup k objektům by měl záviset na více podmínkách (např. správná autentizace a vlastnictví klíče).
- Nejmenší sdílené prostředky - Sdílené moduly jsou potenciálním kanálem pro únik informací, mělo by jich být co nejméně.
- Snadná použitelnost - Mechanismus není obcházen, když se neplete více, než je únosné.

67 Výběr dat ze statistické databáze

Statistická databáze obsahuje velké množství položek (statistická data), přičemž jednotlivé položky jsou senzitivní. Není možné číst pouze jednotlivé položky.

- Potlačení malých výsledků (limited response suppression) - Systém nevydá odpověď, pokud výsledná hodnota nepřekračuje stanovený limit, případně závisí na příliš malém, nebo příliš velkém oboru (tzn. $< k$ nebo $> n - k$ řádků z celkových n pro zvolené k).
- Kombinování výsledků - Systém nevydává výsledky týkající se jednotlivých hodnot z dané domény, ale pouze souhrnné výsledky pro intervaly těchto hodnot.
- Modifikace výsledků(response modification) - Systém spočítá přesný výsledek, který následně mírně poškodí možností je zaokrouhlování výsledků, navracení průměru výsledků pro interval hodnot z dané domény apod.

- Náhodný šum - Před vyhodnocením výsledku je ke každé použité položce připočtena malá náhodná chyba, chybové hodnoty volíme jako náhodnou veličinu se střední hodnotou 0.
- Náhodný výběr (random sample) - Systém neodpovídá na základě všech hodnot v databázi ale pouze na základě provedeného náhodného výběru ze všech relevantních položek, aby nebylo možné počítáním průměrných hodnot výsledků opakovaných dotazů získat přesné hodnoty, měl by se pro ekvivalentní dotazy používat stále stejný výběr.
- Náhodné zmatení (random data perturbation) - Ke každé položce v databázi přičteme náhodnou chybu e, přičemž narozdíl od náhodného šumu pro opakované dotazy systém zajišťuje, že použitá chyba je stále stejná pokud je chyba z okolí nuly, je vliv této úpravy na statistické výsledky typu součet, průměr malý. Metoda je vhodnější než předchozí, neboť lze snáze zajistit stejné výsledky ekvivalentních dotazů.

68 Zbytkové riziko

Není reálně možné vytvořit dokonalý bezpečnostní systém s nulovým rizikem bezpečnostního incidentu. Dojde-li k bezpečnostnímu incidentu, je nutno považovat tento stav za jeden z možných provozních stavů systému. Riziko, které hrozí informačnímu systému i přes zavedená bezpečnostní opatření nazýváme zbytkovým rizikem. S pomocí bezpečnostních opatření můžeme toto riziko minimalizovat dokud "se to vyplatí" (náklady na zavedení bezpečnostních opatření jsou menší než dopad který riskujeme).

69 Zotavení po havárii

Pro případ poruchy by měl být vypracován podrobný plán, popisující co je třeba udělat pro zotavení systému. Tento plán by měl být důkladně prověřen a otestován. Měl by být stále k dispozici (nejlépe v tištěné podobě). Schopnost provést co nejrychlejší zotavení systému je často velice důležitá (porucha systému je bezpečnostní incident, který může mít obrovský dopad). Většina dodavatelů hardwaru je schopna dodat náhradní technické vybavení během jednoho dne.

- **Cold site** je zařízení vybavené zdroji el. energie, klimatizací, komunikačními linkami atd. Systém zde může být rychle nainstalován a uveden do provozu.
- **Hot site** je zařízení vybavené též nainstalovaným systémem, připraveným ke spuštění - stačí pouze dodat zálohu dat a programů. S pomocí Hot site může být zcela zničený systém obnoven během několika málo hodin.

70 Způsobilost (capability)

Způsobilost budeme chápat jako nefalšovatelný token, jehož vlastnictví dává vlastníkovi specifická práva k danému objektu. Lze chápat jako lístek do kina. Jednou z metod zajištění nefalšovatelnosti je, že tokeny se nepředávají přímo subjektům, ale jsou udržovány v chráněné oblasti paměti, přístupné pouze systému. Při přístupu k objektu tak systém zkontroluje existenci příslušného tokenu. Tento postup lze urychlit tím, že zvlášť udržujeme seznam způsobilostí právě běžícího procesu. Výhodou metody je, že dovoluje definovat nové dosud neznámé způsoby používání objektů a přidělovat odpovídající oprávnění. Nevýhodou je opět poněkud obtížná správa těchto tokenů, zejména odebrání způsobilosti je netriviální operace.