Architektura bezpečnosti

systémy a aplikace zpravidla procházejí určitými fázemi

1. návrh
2. vývoj
3. testování
4. nasazení
5. údržba
6. vyřazení

opatření ideálně již ve fázi návrhu – je to levnější a funkčně optimální
opatření třema implementovat tak, aby nešla vypnout, obcházet, odstranit apod.
normy
ISO/IEC 19249 Information technology – Security techniques – Catalogue of architectural and design principles fo secure products, systems and applications
STRIDE (spoofing, tampering, repudiation, inf. disclosure, DoS, elevation)
DREAD (damage, reproducibility, exploitability, awarenes, discoverability)
PASTA – process of attack simulation and threat analysis
…

Obecné principy

Separace domén

doménou rozumíme soubor souvisejících komponent se společnými bezpečnostními atributy
každé má odpovídající bezpečnostní požadavky
komunikace omezená na dobře definované kanály se striktní kontrolou přístupu, toku informací, auditní stopou, …

Rozvrstvení (layering)

hierarchické strukturování systému
vyšší vrstvy závisí výhradně na službách poskytovaných nižšími vrstvami
jako každá dekompozice napomáhá srozumitelnosti, lepšímu testování, spolehlivosti
snažší implementace bezpečnostních mechanismů

Zapouzdření

objekty nezpřístupňuí data, ale metody (neplatí jen pro programování)
lepší kontrola přístupu, lepší integrita
kontrolované změny objektů, uniformní přístup

Redundance
replikace komponent, paralelní zpracování, vyšší odolnost

Virtualizace
další úroveň separace, snazší zotavení, lepší zvládání dostupnosti, kontejnerizace

Nejmenší oprávnění (least priviledge)
služby a informace dostupné pouze na základě aktuální potřeby, ideálně just-in-time princip .. jedná se o implementaci **principu need-to-know**

Minimalizace plochy pro útok
plochou pro útok (attack surface) rozumíme souhrn všech expozic
„hardening“ systému – vypnutí nepotřebných služeb, odstranění standardních účtů, redukce otevřených portů a aplikací, zabránění možnosti úprav
redukce komplexity

Centralizovaná validace parametrů
validace vstupů, aplikační firewall, kontrola poškozených dat, …

Centralizace bezpečnosntích opatření
centralizace různýc bezpečnostních funkcí

- autentizace
- řízení přístupu
- autorizace,
- logování,
- správa klíčů,
- …

Správa výjimek a chyb
detekce a spuštění odpovídající akce, zabránění úniku informací, zachování systému v bezpečném stavu

Bezpečné výchozí hodnoty
systém musí po spuštění / instalaci být bezpečný, bezpečnost nesí záviset na tom, že dojde k aktivaci mechanismů
ideální je maximálně bezpečné výchozí nastavení a naásledně po úvaze omezovat

Bezpečné havarování

jde o to, jak se systém zachová v případě neočekávané události a havárie
fail-open: snaha o zachování dostupnost služeb
fail-safe: implicitně se blokuje přístup

Separace odpovědností

jde o zajištění, že určitou operaci, proces či úkon nemůže provést jeden člověk subjekt

Keep-it-simple

čím je věc jednodušší, tím snadněji se testuje, pochopí, provozuje
vypadá to hloupě, ale je to zcela fundamentální princip

Důvěřuj, ale prověřuj (trust but verify)

ověřujte cokoliv, co přichází z vnějšího prostředí (vzhledem k aktuální bezpečnostní doméně), audituj
tj. věříte, že vaše okolí funguje rozumně, ale ověřujete si to

Nulová důvěra (zero trust)

nevěříte vůbec ničemu – veškeré vstupy se validují, věškeré přístupy autentizují, de facto nerozlišujete mezi vnějším a vnitřním prostředím
součástí této strategie je předpoklad, že došlo k vyzrazení (assume breach)

Bezpečnost v návrhu (privacy by design)

bezpečnost musí být integrální součástí návrhu systému

- proaktivní, ne reaktivní
- utajení jako výchozí nastavení
- mechanismy mají být integrální součástí řešení
- end-to-end bezpečnost … po celou dobu životního cyklu
- transparentnost – tj dobrá zdokumentovanost mechanismů
- respekt k soukromí

Bezpečnost do hloubky (defence in depth)

jedná se o soubor praktik, jaknavrhovat bezpečnost

- vícevrstvý (vícenásobný) přístup k aplikaci bezpečnostních mechanismů
- rozdělení systému na vrstvy se striktním oddělením
- navzájem se podporující mechanismy

- předpoklad průniku (tj. systém mysí být navržen tak, aby zůstal bezpečný v případě překonání jednoho či více bezpečnostních opatření)

Formální modely bezpečnosti

první fází tvorby bezpečného IS je volba vhodného bezpečnostního modelu
připomeňme dodržení základních požadavků bezpečnosti:

utajení, integrita, dostupnost, anonymita, ...

dále budeme předpokládat, že umíme rozhodnout, zda danému subjektu poskytnout přístup k požadovanému objektu, modely poskytují pouze mechanismus pro rozhodování

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 !***všechna data stejná***!

Monitor model

též reference monitor

- subjekt při přístupu k objektu vyvolá tzv. *monitor* a předá mu žádost jakou akci s kterým objektem chce provést
- monitor žádost vyhodnotí a na základě informací o přístupových právech vyhoví či nikoliv

výhodou jednoduchost a snadná implementovatelnost
nevýhodou je, že proces poskytující služby monitoruje volán při každém přístupu k libovolnému objektu, což systém velmi zatěžuje
další nevýhodou je, že tento model je schopen kontrolovat pouze přímé přístupy k datům, ale není schopen zachytit např. následující případ

```
if profit <= 0
   then delete file F
   else
      write file F, "_zpráva_"
endif
```

subjekt mající legitimní přístup k souboru *F* může získávat informace o proměnné *profit*, k níž by přístup mít neměl

Information flow model

odstraňuje posledně jmenovanou nevýhodu předchozího modelu
autoři si všimli, že uživatel může získávat i jiné informace, než na které se explicitně ptá

již ve fázi vývoje je prováděno testování všech modulů, zda jejich výstupy závisí na interakcích se senzitivními daty a případně jakým způsobem
z těchto dílčích výsledků je sestavován celkový graf závilostí
veškeré požadavky na systém procházejí inteligentním filtrem, který zjišťuje, zda nedochází k nežádoucí kompromitaci informací

Modely pro specifické účely

Clark-Wilson model

dobře odpovídá potřebám komerčních organizací, přejímá postupy běžné v účetnictví
základní principy:

1. dobře formované transakce (konzistentní data → konzistentní data)
2. separace odpovědností – žádnou operaci nesmí být schopen korektně provést jediný subjekt

pravidla modelu rozdělujeme obvykle na požadavky na korektnost „C“ a na vynucení „E“
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.
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.
C3 – Seznam popsaný v E2 musí splňovat pravidlo separace odpovědností.
E3 – Systém musí autentizovat každý subjekt pokoušející se spustit transformační proceduru.
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.
E4 – Pouze administrátor provádějící certifikaci entit může provádět změny relací. V žádném případě nesmí mít právo spustit žádnou z procedur, které administruje.

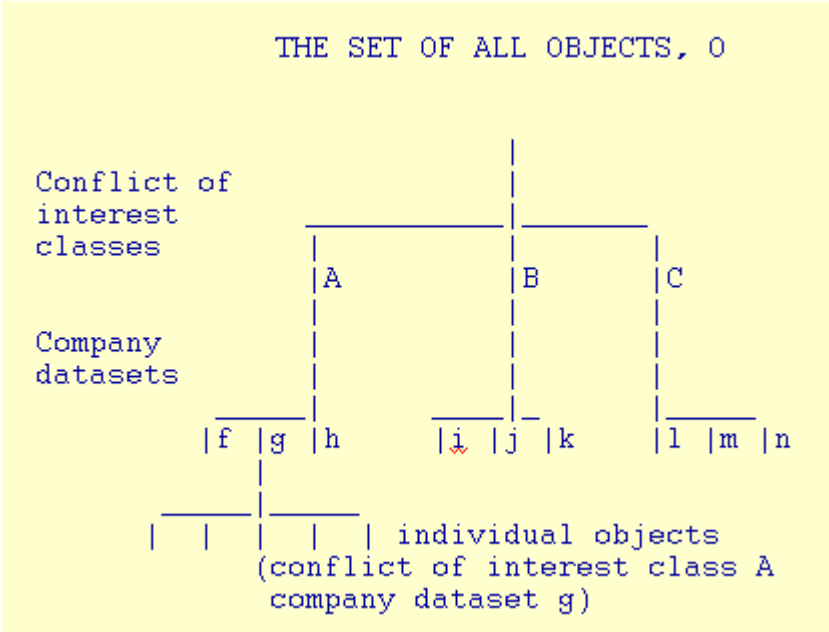
Chinese wall model (Brewer-Nash)

Představuje příklad dynamického modelu – pravidla jsou generována až v okamžiku používání řízených objektů

„Konzultant musí zachovávat diskrétnost informací získaných v různých firmách, tj. nesmí radit konkurenční firmě na základě vnitřních znalostí jiné korporace. Může ale radit nekonkurenčním firmám, případně může dávat rady na základě obecných informací“.

objekty jedné organizace tvoří dataset, datasety rozčleněny do tříd (conflict of interest classes)

sanitizovaná informace – odstraněny ty části, které umožňují identifikovat konkrétního vlastníka



subjekt na počátku univerzální práva (ke všem objektům)

Vlastnost jednoduché bezpečnosti

Přístup je povolen pokud požadovaný objekt:

1. je ve stejném datasetu jako objekt, ke kterému subjekt již přistupoval, nebo
2. náleží do jiné třídy

**-vlastnost*

Zápis je povolen pouze v případě že:

1. přístup je možný podle vlastnosti jednoduché bezpečnosti, a zároveň
2. není čten žádný objekt obsahující nesanitizované informace náležející do jiného datasetu než toho, do kterého se zapisuje

Graham-Denning model

popisuje proces tvorby objektů a subjektů a bezpečného přidělování oprávnění i v distribuovaném prostředí

model pracuje s množinou subjektů *S,* množinou objektů *O*, množinou práv *R* a přístupovou maticí *A*.

Každý objekt má přiřazen jeden subjekt nazývaný *vlastník,* každý subjekt má přiřazen jiný subjekt nazývaný *kontroler*.

Model definuje následující práva:

- *vytvořit objekt* - povoluje subjektu vytvořit v systému nový objekt
- *vytvořit subjekt*, *zrušit objekt*, *rušit subjekt* - obdobně jako předchozí
- *číst přístupová práva* - povoluje subjektu zjistit aktuální přístupová práva jistého subjektu k určitému subjektu
- *přidělit přístupová práva* - dovoluje vlatníku objektu přidělit jistá práva k objektu určitému subjektu
- *zrušit přístupová práva* - dovoluje vlastníku objektu resp. kontroleru subjektu odebrat danému subjektu jistá práva k objektu resp. subjektu
- *předat přístupová práva* - dovoluje subjektu předat některé ze svých práv jinému subjektu (každé oprávnění může být předatelné či nikoliv, obdrží-li subjekt předatelné právo, může jej dále předat jako předatelné či nepředatelné).

Následující tabulka uvádí podmínky nutné pro vykonání operací s přístupovými právy.

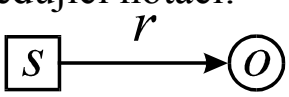
vytvořit objekt *o*	-
vytvořit subjekt *s*	-
zrušit objekt *o*	vlastník je v $A[x,o]$
zrušit subjekt *s*	vlastník je v $A[x,s]$
číst přístupová práva *s* k *o*	kontroler je v $A[x,s]$, nebo vlastník v $A[x,o]$
zrušit přístupové právo *r* subjektu *s* k *o*	kontroler je v $A[x,s]$, nebo vlastník v $A[x,o]$
přidělit *s* právo *r* k objektu *o*	vlastník je v $A[x,o]$
předat přístupové právo *r* nebo *r** k objektu *o* subjektu *s*	*r** je v $A[x,o]$

*r** označuje předatelné právo

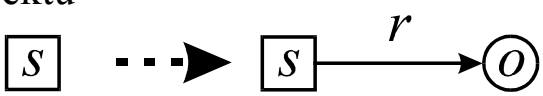
Take-Grant systém

model popisuje systém přidělování a odebírání oprávnění, dovoluje vyhodnocování efektivních práv v lineárním čase

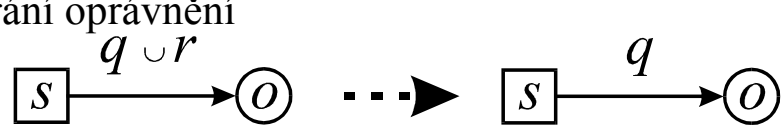
model pracuje s čtyřmi základními primitivami: *create, revoke, take, grant.*
předpokládáme, že systému obsahuje množinu subjektů *S,* množinu objektů *O*, objekty dělíme na aktivní (zároveň i subjekty) a pasivní (nejsou subjekty) a množinu práv *R*
Pro popis operací použijeme následující notaci:



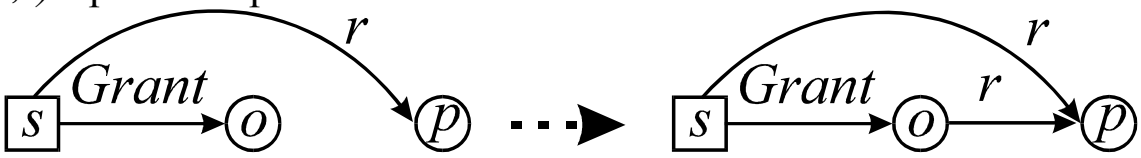
Subjekt *s* má k objektu *o* oprávnění *r*.
create(o,r) - vytvoření objektu



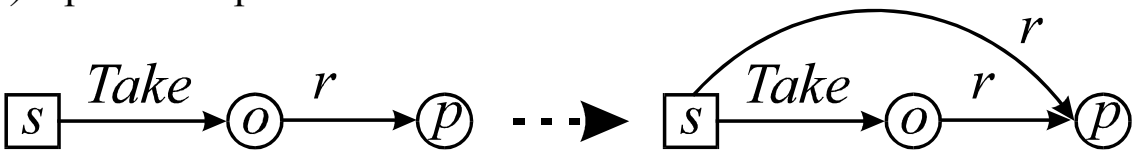
revoke(o,r) - odebrání oprávnění



grant(o,p,r) - předání oprávnění



take(o,p,r) - převzetí oprávnění



Výhodou popsaného systému je, že umožňuje v subpolynomiálním čase řešit dotazy na dostupnost jistého objektu pro daný subjekt.

Víceúrovňové modely

v předchozích 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 ukládáat 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

u zelených mozků je každá informace zařazena do některé z kategorií utajení (např. *unclassified, confidental, secret, top secret*), které jsou disjunktní
silné uplatnění zde má *princip nejmenších privilegií* - každý subjekt má mít pouze taková oprávnění, aby mohl konat svoji práci
všechny chráněné informace jsou rozděleny podle obsahu do *oblastí* (compartments), informace může být i několika oblastech zároveň
klasifikací informace potom rozumíme dvojici <*stupeň_utajení*, *oblasti*>
aby subjekt mohl používat požadovanou informaci, musí mít dostatečné *oprávnění*.
oprávnění má stejný tvar jako klasifikace - <*stupeň_utajení*, *oblasti*>, tedy daný subjekt smí používat informace až do *stupeň_utajení* v těchto *oblastech.*

$$O \leq S \Leftrightarrow st\_utaj_O \leq st\_utaj_S \wedge oblast_O \subseteq oblast_S$$

Relace $\leq$ odpovídá *oprávnění* subjektu *S* k danému objektu *O*.
Požadavky na stupeň utajení bývají označovány jako hierarchické, rozdělení na oblasti jako nehierarchické omezení.

Svazový model (Lattice model)

předchozí military model je specielním případem tohoto modelu
relace $\leq$ je částečným uspořádáním, množina klasifikací všech informací v systému tvoří svaz, stejně tak množina oprávnění všech subjektů
v různých oblastech se používá různých svazů, např. v komerční oblasti jsou obvyklé stupně utajení *public, company confidental, high security*, rovněž rozdělení do oblastí se liší případ od případu …
svazový model je často používaným modelem v mnoha prostředích

dále popíšeme dva modely, zabývající se tokem informací uvnitř systému

Bell-LaPadula model

model popisuje povolené přesuny informací, takové, aby bylo zajištěno jejich utajení pro každý subjekt *S* resp. objekt *O* v systému nechť je definována bezpečnostní třída $C(S)$ resp. $C(O)$
bezpečné přesuny informací mají následující vlastnoti:
Vlastnost jednoduché bezpečnosti (Simple Security Property):

Subjekt *S* může číst objekt *O* právě když

$$C(O) \leq C(S)$$

*-*vlastnost* (*-Property):

Subjekt *S* mající právo čtení k objektu *O* může zapisovat do objektu *P* právě 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 byl je používán v systémech, které paralelně zpracovávají informace různého stupně utajení.

Biba model

předchozí model se však vůbec nezabývá integritou dat, Biba model je duálním modelem k Bell-LaPadula modelu
Nechť pro každý subjekt *S* resp. objekt *O* v systému je definována integritní bezpečnostní třída *I*(*S*) resp. *I*(*O*). Obdobně jako v předchozím případě definujeme:
Vlastnost jednoduché integrity (Simple Integrity Property):

Subjekt *S* může modifikovat objekt *O* právě když

$$I(O) \leq I(S)$$

*Integritní *-vlastnost* (Integrity *-Property):

Subjekt *S* mající právo čtení k objektu *O* může zapisovat do objektu *P* právě když

$$I(O) \geq I(P)$$

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

Tímto výčet modelů zdaleka nekončí:

- Model nezasahování (noninterference) – ochrana proti vzájemnému rušení mezi akcemi na různých stupních senzitivity (tzn. úniky informací)
- Goguen-Meseguer model – zajišťuje integritu prostřednictvím parcelizace a zásad pro zabránění rušení
- Sutherland model - zajištění intagrity prostřednictvím definice přípustných stavů systému a možných tras přesunu ze stavu do stavu zabraňujícím rušení

- Harrison-Ruzzo-Ullman model – definuje možné operace nad definovanou sadou objektů oprávnění, zajišťuje, že se nikdo nemůže oprávnění neautorizovaně rozšířit
- …

Bezpečnostní rámce (frameworks)

Pro vlastní postup implemetnace bezpečnosti je vhodné zvolit si rámec – zpravidla dává nějakou kategorizaci bezpečnostních témat a mechanismů, stanovuje postup a poskytuje seznam, co vše je třeba řešit

- ISO 27001
- NIST 800-37: Risk management framework
- NIST CyberSecurity Framework
- COBIT
- …

není nutné omezit se na jeden, pro různé aspekty bezpečnosti lze vhodně kombinovat

studium rámce usnadní pro jednotlivé okruhy problémů zvolit vhodná opatření – a pak přichází nekonečný řetězec údržby:

zdroj: thequalityinsider

Své plány byste měli revidovat nejméně:

- po bezpečnostním incidentu nebo úniku
- významných změnách v organizaci
- u příležitosti zavedení nového produktu/procesu či ukončení starého
- při významných změnách ohrožení
- vři významných změnách v informačním systému
- při zavedení zpracování nového typu informací
- při změně pravidel bezpečnosti
- při významné sociální či společenské změně