PV157 – Autentizace a řízení přístupu

Řízení přístupu II.

Politiky řízení přístupu

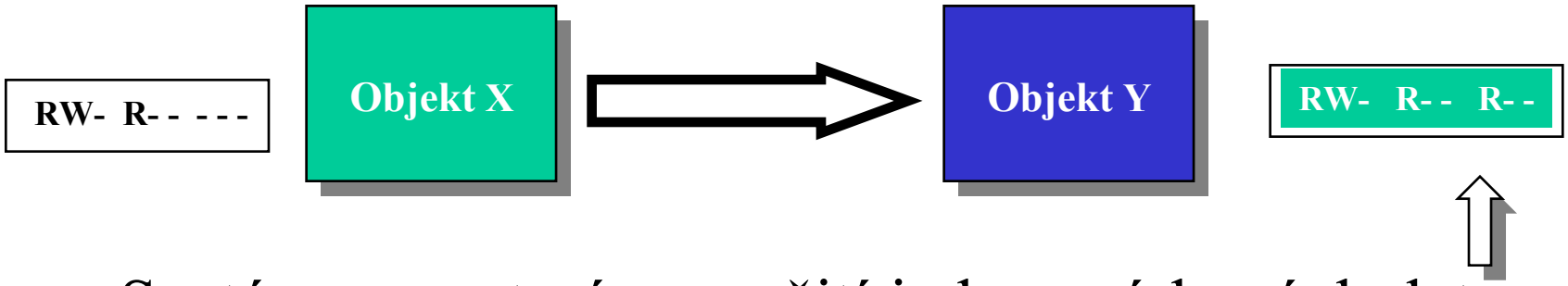
- **volitelný přístup** (*discretionary*)
  - subjekt (vlastník objektu) rozhoduje o tom, kdo má k objektu přístup
  - volitelná = určuje subjekt–vlastník objektu
  - typicky politika podporovaná operačním systémem
    - podporuje i operace změny vlastníka objektu
- **povinný přístup** (*mandatory*)
  - systémová politika nezávislá na vůli subjektů rozhoduje o tom, kdo má k objektu přístup

Volitelné řízení přístupu – výhody

- Jednoduchost = malá režie
- Velká vyjadřovací schopnost
- Lze relativně jednoduše vázat udělení přístupových práv na splnění dodatečných časových, místních aj. podmínek
- Flexibilita

Volitelné řízení přístupu – nevýhody

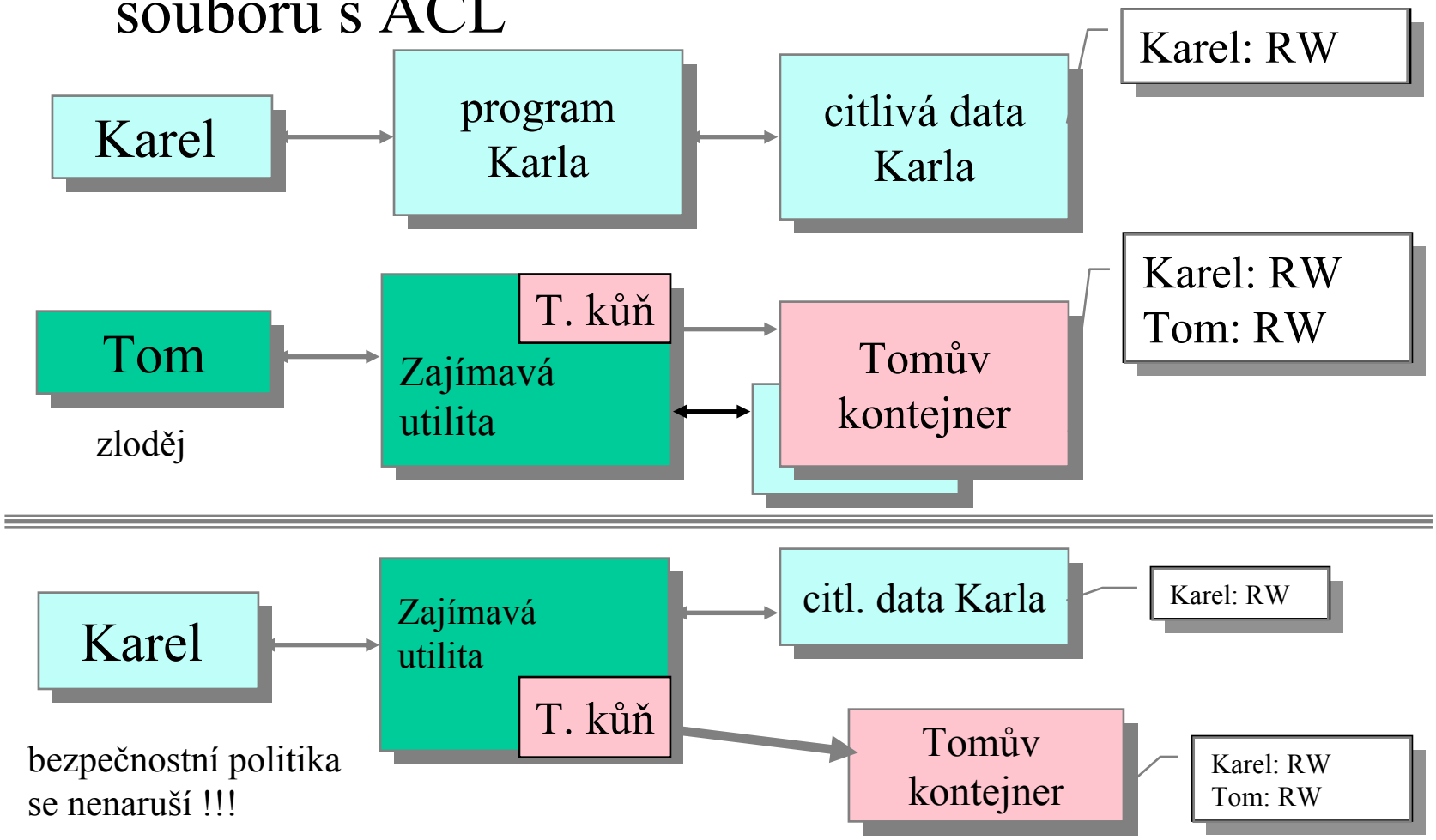
- Nedostatečná bezpečnost
  - použití pouze přístupových práv není dostatečné v situacích, kdy klademe důraz na bezpečnost
- Není odolné vůči "Trojským koním"



- Systém se nestará o využití jednou získaných dat
  - např. dám skupině právo čtení na důvěrný soubor, nějaký člen si ho zkopíruje a nesprávně nastaví přístupová práva

Volitelné řízení přístupu

- Příklad útoku Trojským koněm na správu souborů s ACL



Víceúrovňové systémy

- MLS (Multilevel systems)
- Koncept podporovaný mnohaletým výzkumem sponzorovaným vládami (zvláště USA a UK)
- Původně podpora důvěrnosti, později i integrita (komerční systémy)
- Primární model bezp. politiky – Bell-LaPadula
  - 1973, pro US AirForce
  - ochranné schéma klasifikací a oprávnění po 2. světové válce

Povinné řízení přístupu

- systémová politika nezávislá na vůli subjektů rozhoduje o tom, kdo má k objektu přístup
- zavedeme
  - kategorie subjektů (proces, uživatel) = důvěryhodnost
  - klasifikace objektů (data) = důvěrnost
- definujeme uspořádání klasifikací objektů
- definujeme množinu kategorií subjektů
- **referenční monitor** (monitor odkazů)
  - implementace funkce prosazující bezpečnost *řízení přístupu*
  - při každém přístupu subjektu k objektu kontroluje, zda tento přístup odpovídá zásadám bezpečnostní politiky
  - bezpečnostní politika: pravidla toku dat mezi objekty a subjekty

Politiky nejsou exkluzivní!

- Při povinném přístupu (prosazovaném systémem) lze (a často je to žádoucí) obvykle podporovat i volitelný přístup!!!
  - Bezpečnost (daná politikou povinného přístupu) s určitou flexibilitou (podporující bezpečnost) vyjadřovacími prostředky volitelného přístupu

Model Bell-LaPadula (1)

- paradigma objekt – subjekt
- uživatel
  - Má počáteční bezpečnostní úroveň uživatele, resp. bezpečnostní **oprávnění** (*clearance*)
  - Přihlašuje se na aktuální bezpečnostní úrovni uživatele, s právy přístupu k objektům nepřevyšujícími práva daná bezpečnostním oprávněním
- subjekt
  - aktivní element – proces činný na pokyn uživatele
  - provádí akce:
    - read-only, append (zápis bez čtení), read-write, execute
  - bezpečnostní úroveň procesu = bezp. úroveň jeho uživatele
    - Je daná „důvěryhodností“ subjektů vlastnících proces a „důvěrností“ (citlivostí) zpracovávatelných objektů (klasifikací)

Model Bell-LaPadula (2)

- objekt
  - pasivní, chráněný element
  - obsahuje informace
  - soubor dat, prostor paměti, program
  - **klasifikace** objektu = bezpečnostní úroveň objektu
    - daná důvěrností (citlivostí) informace obsažené v objektu
    - definuje/mění ji vlastník objektu, vlastnictví objektu je nepřenositelné

Bell-LaPadula – Klasifikace / kategorie

- Bezpečnostní úroveň L= (C, $\underline{S}$)
  - C – klasifikace objektů

TS	top secret	přísně tajné
S	secret	tajné
C	classified	pouze pro vnitřní potřebu (důvěrné.)
U	unclassified	neklasifikováno

    Definice uspořádání: TS > S > C > U
  - $\underline{S}$ – podmnožina množiny kategorií subjektů
    - množina kategorií subjektů je dána aplikací
      - {odbor obrany, ekonomický odbor, vnitřní odbor}
      - {ekonomický odbor, vnitřní odbor}
      - {odbor obrany, vnitřní odbor}
      - {vnitřní odbor}
- Upořádání bezpečnostních úrovní – dominance

  L1=(C1, $\underline{S1}$), L2=(C2, $\underline{S2}$), $L1 \geq L2 \Leftrightarrow C1 \geq C2 \wedge \underline{S1} \supseteq \underline{S2}$

Bell-LaPadula – příklad

- bezpečnostní úrovně

  L1 = (S, {ekonom.}) tajné, {ekonom. odbor}

  L2 = (C, {ekonom.}) pro vnitřní potřebu, {ekonom. odbor}

  L3 = (TS, {obrana}) přísně tajné, {odbor obrany}

  L4 = (TS, {ekonom., obrana}) přísně tajné, {odbor obrany, ekonomický odbor}

- uspořádání (dominance) bezpečnostních úrovní

  $L1 \geq L2$ S > C, {ekonom.} ≡ {ekonom.}, L1 dominuje L2

  L1, L3 L1 neporovnatelné s L3 : {ekonom.} {obrana}

  $L1 \leq L4$ S < TS, {ekonom.} $\subseteq$ {ekonom, obrana}

  L2, L3 L2 neporovnatelné s L3 : {ekonom.} {obrana}

  $L2 \leq L4$ C < TS, {ekonom.} $\subseteq$ {ekonom., obrana}

  $L3 \leq L4$ TS = TS, {vnější vztahy} $\subseteq$ {ekonom., obrana}

Klasifikace v ČR a NATO

NATO	ČR	Německo
cosmic top secret	přísně tajné	streng geheim
NATO secret	tajné	geheim
NATO confidential	důvěrné	VS-vertraulich
NATO restricted	vyhrazené	VS-nur-für den Dienstbegrauch

Bell-LaPadula – stav systému

- Stav systému, $\Sigma = (b, \underline{M}, f)$

  b – množina aktivních (právě realizovaných) přístupů
  - trojice (subjekt, objekt, právo)

  $\underline{M}$ – matice přístupových práv
  - M[s, o] přístupová práva subjektů s k objektům o

  f – úrovňová funkce: $\underline{O} \cup \underline{S} \rightarrow L$,
  - množiny: O – objektů, S – subjektů, L – bezpečnostních úrovní
  - udává bezpečnostní úroveň každého subjektu a objektu
    - objekty, každý má jedinou klasifikaci (bezp. úr. obj.): f_o
    - subjekty, každý vlastní dvě „bezpečnostní úrovně subjektu“:
      - » bezpečnostní oprávnění, clearance f_p
      - » aktuální bezpečnostní úroveň subjektu f_a, $f_a(s) \leq f_p(s)$
- bezpečnost systému je chápána jako vlastnost <u>stavů systému</u>

Bell-LaPadula – bezpečnost stavu

- Stav systému se mění operacemi změny stavu systému
  - uplatnění přístupových práv, změny přístupových práv
- Stav systému je bezpečný pouze tehdy, když jsou splněny všechny bezpečnostní vlastnosti
  - omezení daná vztahy bezpečnostních úrovní subjektů a objektů
- Operace změny stavu systému se povolí pouze tehdy, když výsledný stav systému po jejím provedení bude bezpečný – kontroluje referenční monitor
- Důvěryhodnost subjektu
  - důvěryhodný subjekt – smí porušovat bezpečnostní politiku povinnou pro nedůvěryhodné subjekty
    - Ví, co smí a nesmí, kdy komunikuje s jinými důvěryhodnými subjekty, ...
  - nedůvěryhodný subjekt – jeho chování je třeba hlídat doplňkovými omezeními podle zavedené bezpečnostní politiky

Bell-LaPadula – operace změny stavu

- právo přístupu – read-only, append, execute, read-write
- operace změny stavu
  - získat/vrátit právo přístupu, zahájit/ukončit operaci s objektem
    - mění množinu aktivních přístupů **b**, doplňuje / ruší trojici (s, o, p)
  - dát / odebrat právo přístupu subjektu k objektu, modifikace **M**
    - musí být v souladu s politikou definovanou axiomy povinné bezpečnostní politiky
  - změnit aktuální bezpečnostní úroveň subjektu
    - musí se zachovat dominance bezpečnostního oprávnění subjektu, mění se **f**
  - změnit bezpečnostní úroveň objektu, jeho klasifikaci
    - pouze pro „neaktivní“ (se kterým nikdo nepracuje) objekt, mění se **f**, lze ji pouze
      - použít oprávněně – nová bezpečnostní úroveň *objektu* musí dominována bezpečnostní úrovní *subjektu* provádějícího změnu
      - a zesilovat – nová úroveň objektu musí dominovat předchozí úrovni

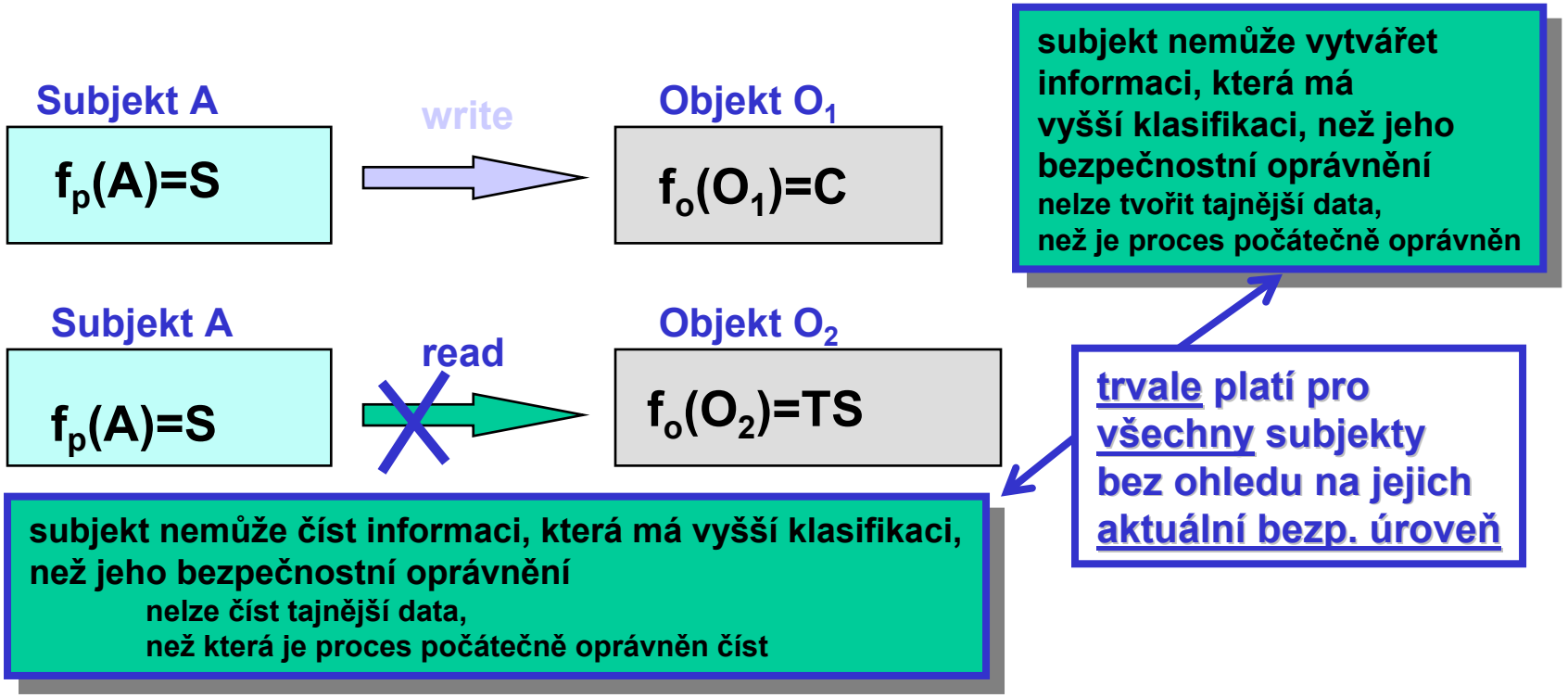
Vlastnosti (axiomy) modelu Bell-LaPadula

- Procesy nesmějí číst data na vyšší úrovni (tzv. základní bezpečnostní vlastnost – *ss property*, též *NRU - no read up*).

- Procesy nesmějí zapisovat data do nižší úrovně (tzv. *-vlastnost, též *NWD - no write down*).

ss-vlastnost

- subjekt může přistupovat (read/write) pouze k objektům s bezpečnostní úrovní (klasifikací) dominované jeho bezpečnostním oprávněním (clearance)
- Pak $\forall s \in S\ o \in O\ read \in M[s,o] \vee write \in M[s,o] \Rightarrow f_p(s) \geq f_o(o)$



Nedostatečnost ss-vlastnosti

- subjekt A nižší bezpečnostní úrovně l_0 ,
  než je klasifikace objektu o_1, může vytvořit Trojského koně, který bude spuštěn s vyšší bezpečnostní úrovní $l_v > o_1$

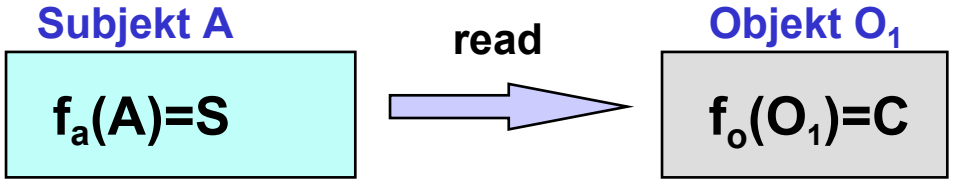
- Trojský kůň může přečíst obsah objektu s klasifikací o_1

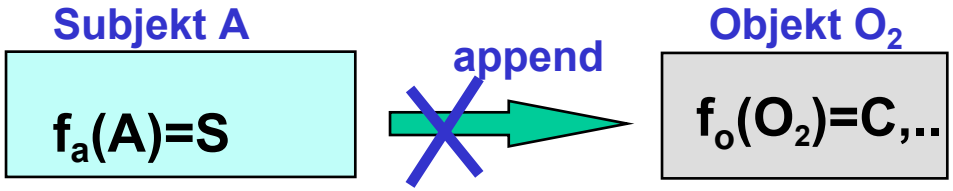
- Trojský kůň může vytvořit objekt s okopírovanou informací s klasifikací objektu $o_2 < o_1$

- subjekt A může číst objekt s klasifikací o_2

*–vlastnost

- pouze pro *nedůvěryhodné subjekty*
- Pak $\forall s \in S' \; \forall o \in O$
  - read $\in M(s,o) \Rightarrow f_a(s) \geq f_o(o)$
  - write $\in M(s,o) \Rightarrow f_a(s) = f_o(o)$
  - append $\in M(s,o) \Rightarrow f_a(s) \leq f_o(o)$
- splnění tohoto axiomu implikuje splnění předchozího axiomu; opak ale neplatí





nedůvěryhodný subjekt může číst informaci, jestliže její klasifikace je dominovaná aktuální b. ú. subjektu

číst lze jen méně tajná data

nedůvěryhodný subjekt může zapisovat informaci, jestliže její klasifikace je shodná s aktuální b. ú. subjektu

tvořit lze jen stejně tajná data

nedůvěryhodný subjekt může doplňovat informaci, jestliže její klasifikace dominuje aktuální b. ú. subjektu

doplňovat lze stejně tajná data nebo tajnější data

nelze poslat zprávu procesu s nižším bezpečnostním oprávněním

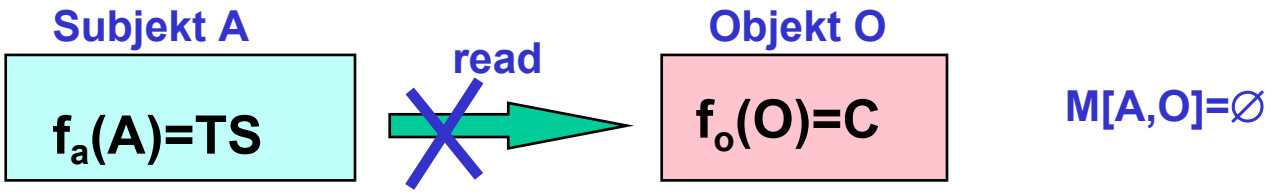
důvěryhodné subjekty mohou porušovat *-vlastnost

Volitelný přístup v Bell-LaPadula

- Také volitelný přístup (discretionary access property)
  - pro povolení přístupu je nutné mít patřičná práva v matici přístupových práv **M**, subjekt musí být pro danou operaci autorizován

$$\forall s \in S\ \forall o \in O\ \forall a \in A\ \ <s,o,a> \in b \Rightarrow a \in M[s,o]$$

    - model B-P je rozšířením modelu s maticí přístupových práv
  - Příklad – subjekt s bezp. úrovní TS **nemusí** mít právo čtení k jistému objektu O, byť tento má klasifikací C



- Stav systému – je považován za bezpečný, jsou-li splněny všechny 3 vlastnosti

Problémy víceúrovňových systémů

- Jak klasifikovat data?
- Tendence k příliš striktní klasifikaci!
- Jak propojit MLS jednoho typu MLS jiného typu (neekvivalentní klasifikací) – např. US vs. UK?
- Vývoj MLS bývá příliš komplikovaný a drahý
- Administrace je náročná
- Uživatel bývá při své práci příliš omezován
- Jak řešit snížení klasifikace?

Skrytý kanál

- *Covert channel* - mechanismus, který není primárně určen pro komunikaci, ale může být využit (zneužit) pro komunikaci mezi jednotlivými úrovněmi
- Typické je využití nějakého sdíleného prostředku
  - zaplnění disku
  - pozice hlavičky na disku
  - zamykání souborů
  - čas posledního přístupu k souboru
  - aktuální zátěž procesoru
- Obrana – snížení šířky pásma komunikačního kanálu
  - diskové kvóty, nucené nastavení hlaviček disku
  - zavedení šumu

Interference

- Chráníme existenci informace na vyšší úrovni
- Uživatel na nižší úrovni chce vytvořit soubor, který již existuje na úrovni vyšší
  - Můžeme zakázat ⇒ prozradíme existenci souboru
- *Noninterference* = vlastnost, kdy akce uživatele na vyšší úrovni nijak neovlivní to, co vidí uživatel na nižší úrovni
- Souborový systém: zavedeme konvence pro pojmenování
- Databáze: netriviální problém (smyšlený příběh vs. zatajení)
- Př.: USA:

klasifikace	Účel skladu
C	Sklad atomových zbraní
U	Sklad uniforem

UK:

klasifikace	Účel skladu
C	Sklad atomových zbraní
U	klasifikováno

Model Biba

- K zajištění integrity
  - „převrácený“ model Bell-LaPadula
  - Integrita a důvěrnost jsou svým způsobem doplňující se koncepty (někdo musí zapsat = změnit integritu, aby šlo vůbec číst)
- Číst lze jen data vyšší úrovně (důležitější, přesnější, spolehlivější)
- Zapisovat lze jen „dolů“ (podřízeným)
- Např. systém pro informování cestujících bere data od signalizačního systému, ale nemůže jeho data měnit.

Dopad MLS

- Velké množství bezp. projektů a výzkumu
- Koncepty pro ne-MLS systémy, jako např.
  - Důvěryhodná cesta (*Trusted Path*) – bezpečný kanál pro komunikaci komponent
  - Důvěryhodná distribuce (*Trusted Distribution*) – bezpečná distribuce systému
  - Důvěryhodná správa zařízení (*Trusted Facility Management*) – bezpečná administrace

Skutečné MLS systémy

- Upravené verze běžných systémů
  - Trusted Solaris
  - HP Virtual Vault
  - TrustedBSD
  - SE Linux
  - AppArmor

SE Linux

- vyvinuto za pomoci NSA
- součásti jádra Linuxu od verze 2.6.0
- je nadstavbou POSIX capabilities (tj. práva je možné nastavovat jemněji než jen běžný uživatel vs. root)
- od verze 2.6.12 obsahuje i MLS
- aktivní (enforcing) vs. pasivní (permissive) režim

SE Linux (2)

- Při přístupu subjektu k objektu pomocí systémového volání se kromě běžných přístupových práv kontroluje splnění bezpečnostní politiky
- Tuto kontrolu provádí *bezpečnostní server* vůči sadě pravidel
- Rozhoduje se na základě trojice identita:role:typ a klasifikace v MLS (volitelné)
  - identita (user_u, system_u nebo speciální identita pro některé uživatele), odlišná od UID
  - role (sysadm_r, system_r, user_r)
  - typ – typ objektu (file_t, default_t, user_home_dir_t)

SE Linux (3) – příklad

```
# ls -Z /
drwxr-xr-x  root     root     system_u:object_r:bin_t          bin
drwxr-xr-x  root     root     system_u:object_r:boot_t         boot
drwxr-x---  root     root     root:object_r:user_home_dir_t    root
drwxr-xr-x  root     root     system_u:object_r:sbin_t         sbin
drwxr-xr-x  root     root     system_u:object_r:file_t         selinux
-rw-r--r--  root     root     system_u:object_r:net_conf_t     yp.conf

# ls --scontext
system_u:object_r:etc_t          shadow

# chcon system_u:object_r:httpd_sys_content_t index.html

# id -Z
root:staff_r:staff_t
```

Klasifikace (MLS) v SE Linux

- MLS v SE Linux je implementací modelu Bell-LaPadula
- Úrovně - sensitivity (s)
  - s0 až s15
- Kategorie – category (c)
  - c0 až c1023

s0 bez kategorie až
s0 se všemi kategoriemi

```
SELinux User    Labeling Prefix  MLS/ MCS Level  MLS/ MCS Range                SELinux Roles

git_shell_u     user             s0              s0                            git_shell_r
guest_u         user             s0              s0                            guest_r
root            user             s0              s0-s0:c0.c1023                staff_r sysadm_r system_r unconfined_r
staff_u         user             s0              s0-s0:c0.c1023                staff_r sysadm_r system_r unconfined_r
sysadm_u        user             s0              s0-s0:c0.c1023                sysadm_r
system_u        user             s0              s0-s0:c0.c1023                system_r unconfined_r
unconfined_u    user             s0              s0-s0:c0.c1023                system_r unconfined_r
user_u          user             s0              s0                            user_r
xguest_u        user             s0              s0                            xguest_r
```

```
drwxr-xr-x.  13 system_u:object_r:usr_t:s0       root root  4096 Jan 17 16:35 usr
drwxr-xr-x.  27 system_u:object_r:var_t:s0       root root  4096 Jan 17 17:15 var
```

Integritní úrovně Windows Vista

- Každý objekt má přiřazenou integritní úroveň znamenající jeho důvěryhodnost
- 6 hierarchických integritních úrovní
  - Untrusted – procesy přihlášené anonymně
  - Low – procesy pracující s Internetem
  - Medium – běžná úroveň
  - High – administrátor
  - System – systémové procesy, služby jádra
  - Installer – instalace a odinstalace

Integritní úrovně Windows Vista

- Subjekty na nižší úrovni nemohou modifikovat objekty na vyšší úrovni
  - Na čtení a spouštění se to nezvtahuje (vs. Biba)
- Integritní úroveň není dynamická
  - Čtením méně důvěryhodných objektů se úroveň procesu nesnižuje (vs. Biba)
- Integritní politika zabraňuje přístupu k objektům, ale nesleduje informační toky

Role based access control (RBAC)

- Není ani volitelné ani povinné řízení přístupu, ale samostatná kategorie
- Uživatelům jsou přiřazeny role (uživatel může mít více rolí)
- Role znamenají práva k provedení určitých akcí, tato práva mohou být specifikována velice jemně (např. přidat záznam, upravit nějakou položku apod.), jemněji než pomocí ACL
- Role jsou však specifické pro každý IS, v heterogenní organizaci není snadné vytvořit jasný systém rolí, jím odpovídajících práv a rozdělení uživatelů do rolí

Role based access control (RBAC)

- Příklad RBAC – oracle
  - create role vyuka;
  - grant CREATE SESSION, ALTER SESSION, CREATE PROCEDURE, CREATE SEQUENCE, CREATE SYNONYM, CREATE TABLE ... to vyuka;
  - grant vyuka to zriha;

Příští přednáška 17. 5. 2011 v 10:00

matyas@fi.muni.cz
zriha@fi.muni.cz