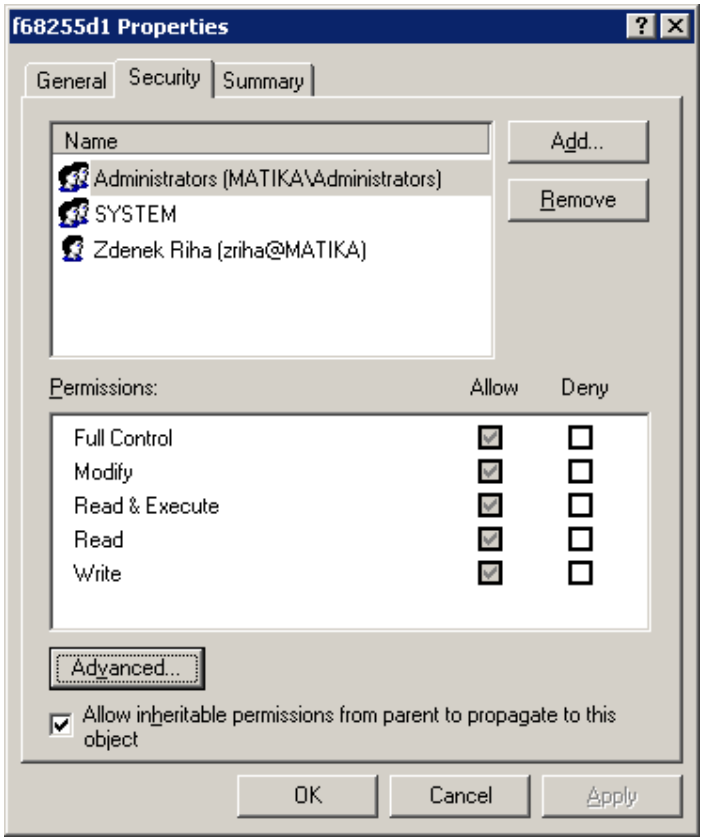
Řízení přístupu



Řízení přístupu

- Funkce pro řízení, který subjekt (uživatel, proces …) má jaký přístup k určitému objektu (souboru, databázi, tiskárně …).
- Implementovaná bezpečnostním mechanismem řízení přístupu.
  - Hierarchie
    - ✓ Hardware
    - ✓ Operační systém
    - ✓ Middleware (např. databázový systém)
    - ✓ Aplikace (např. informační systém)
  - Předpoklad implementace
    - ✓ je bezpečně implementovaná funkce „Identifikace & autentizace“
- Kategorie mechanismů řízení přístupu
  - fyzické typicky ochrana off-line uložených archivních kopií
  - logické typicky ochrana on-line uchovávaných dat

Pojmy

- **vlastník dat** (owner)
  - subjekt, statutární autorita odpovědná za daný typ dat,
  - za konkrétní data daného typu, za datový objekt

- **správce** (dat, objektu) (custodian)
  - subjekt, autorita pověřená odpovědností za bezpečnost konkrétních dat, za bezpečnost konkrétního objektu

- **uživatel** (user)

  také autorizovaný uživatel, oprávněný uživatel
  - subjekt, mající právo přístupu ke konkrétním datům, ke konkrétnímu objektu

Typy omezení přístupu k objektu

- R, Read Only
  - ano: kopírování, prohlížení, tisk, ...
  - ne: rušení, vytváření, modifikace, …
- RW, Read/Write
  - ano: kopírování, prohlížení, tisk, … , rušení, vytváření, modifikace
- X, Execute
  - ano: řízení běhu procesu
  - ne: kopírování, prohlížení, tisk, … , rušení, vytváření, modifikace
- daný omezením
  - časem pouze v danou dobu
  - místem pouze z …
  - obsahem interval hodnot (bankovní automat)
  - typem služby e-mail ano, telnet ne

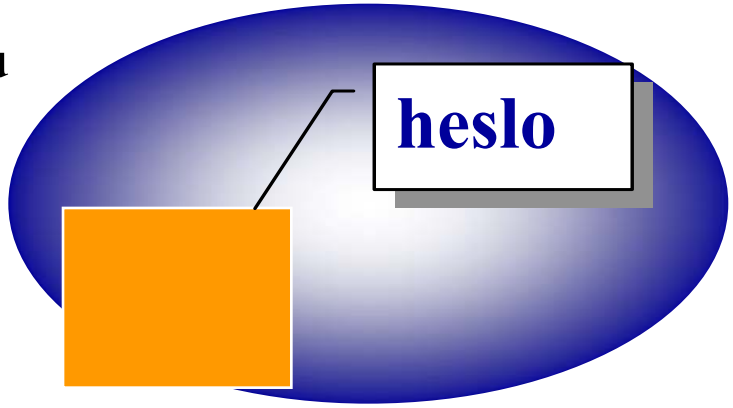
Správa řízení přístupu – modely

- centralizovaná správa řízení přístupu
  - 1 správce všech objektů (IS, Informačního systému)
  - mnoho vlastníků, mnoho uživatelů
  - klad: přísné řízení, konzistentnost
  - zápor: vysoká (časová) režie v (distribuovaných) IS

- decentralizovaná správa řízení přístupu
  - objekt spravuje jeho vlastník – správce,
  - mnoho vlastníků – správců, mnoho uživatelů
  - klad: snadno dosažitelná vysoká odpovědnost
  - zápory:
    - ✓ obtížnost udržení konzistence komunikace mezi správci
    - ✓ není dostupný okamžitý celkový přehled stavu
    - ✓ hůře se prosazuje bezpečnostní politika IS

Heslo

- heslo, šifrovací klíč
  - všeobecné
  - samostatné pro
    - čtení, modifikaci, rušení, …
- každý **objekt** má přiděleno(-a) svým vlastníkem heslo (hesla)
- právo přístupu má subjekt znající heslo
- použito například ke sdílení disků/adresářů v některých verzích MS Windows
- negativa:
  - nelze zjistit kdo všechno má právo přístupu
  - heslo musí být dostupné (uloženo v otevřeném tvaru nebo se slabým maskováním v programu, pom. souboru...)



Matice přístupových práv

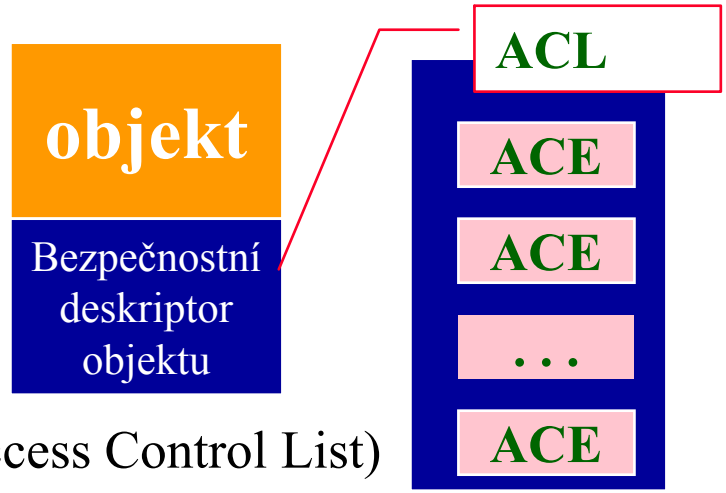
- Matice udávající přístupová práva subjektů k objektům

	objektA	objektB	objektC
subjekt1	rw	rx	–
subjekt2	rwx	rwx	rwx

Matice přístupových práv

- Může být i trojrozměrná
- 3. rozměrem je program
- Máme trojice (subjekt, program, objekt) a jim odpovídající přístupová práva
  - (Alice, účetní program, účetní data) má práva {r,w}
  - (Alice, editor, účetní data) má práva {}
  - (Alice, editor, /etc/passwd) má práva {}
  - (Alice, passwd, /etc/passwd) má práva {r,w}
- Dvourozměrná i třírozměrná matice je v praxi tak velká, že je problém s ní efektivně pracovat (ukládat ji, vyhledávat v ní)
  - Tisíce až miliony objektů (souborů)
  - Tisíce až miliony subjektů (uživatelů)
- Matici můžeme ukládat po řádcích či po sloupcích

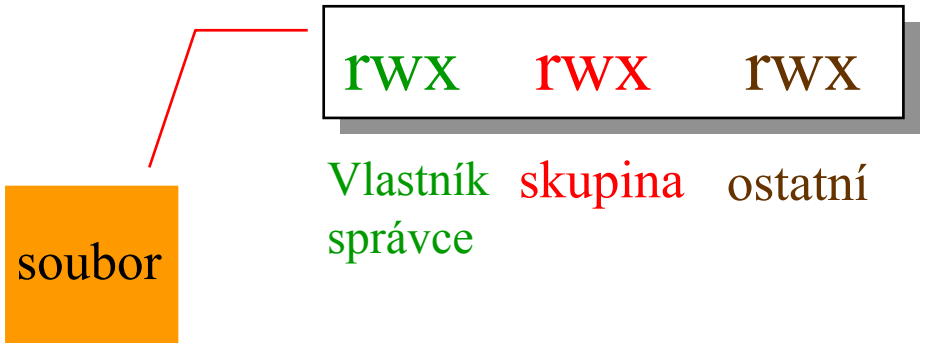
Seznamy přístupových práv



- Ukládáme matici přístupových práv po objektech
- Matice je často řídká, proto ukládáme jen neprázdné prvky
- Seznam přístupových práv objektu – ACL (Access Control List)
- Element přístupových práv – ACE (Access Control Element)
  - prvek ACL
  - přidělení práv přístupu jednotlivému subjektu, skupině subjektů, …
- Výhoda seznamu přístupových práv
  - modifikaci/zpřístupnění lze omezit na jednotlivce ve skupině
- Nevýhoda seznamu přístupových práv
  - obtížná správa, neefektivní kontrola přístupových práv při přístupu k objektu
  - nesnadné zjistit k jakým objektům má určitý uživatel přístup (například při změně pracovního zařazení)

Implementace ACL – UNIX

- Nelze zamítnout přístup jednotlivci ze skupiny vlastníka
- Ve starších verzích UNIXu mohl uživatel patřit do pouze jedné skupiny
- Nutnost vytvářet nové skupiny uživatelů (může jen root)
- Nelze zajistit důvěrnost:



110 100 000 originál
110 100 000 kopie člena skupiny

110 100 100 zrádný člen skupiny upraví

SUID/SGID programy

- Matice přístupových práv jen dvourozměrná.
- Povolení práv pro určité programy je třeba řešit pomocí SUID/SGID bitů.
- Vlastník programu může nastavit, že program po spuštění bude běžet s právy uživatele/skupiny vlastníka.
- Například program pro změnu hesla potřebuje přístup zápisu do souboru /etc/passwd resp. /etc/shadow. Běžný uživatel však nemá k těmto souborům přístup pro zápis. Proto je program passwd nastaven jako SUID na uživatele root, který do těchto souborů zapisovat může.
- Přístup SUID/SGID není příliš intuitivní. Leniví programátoři píší hromadu programů, které musí běžet jako SUID root. SUID programy musí být napsány bezpečně, jejich vstupy (parametry, stdin, proměnné prostředí) nejsou důvěryhodné.

Implementace ACL – moderní UNIX

- Administrátor (root, resp. uživatel s UID 0) má neomezený přístup ke všem objektům, může měnit libovolné soubory, upravovat logy apod.
- Nejen skutečný administrátor, ale i hacker apod.
- Snaha o omezení možností administrátora
  - Nové vlastnosti souborového sytému UNIXových verzí odvozených od Berkeley větve
    - ✓ Možnost nastavit dodatečné „flagy“ pro soubory
      - Append-only: lze pouze přidávat data – vhodné pro logy
      - Immutable: soubor není možné modifikovat – vhodné pro systémové soubory
      - Undeleteable – nesmazatelné
    - ✓ Možnost nastavit pro uživatele i skupiny
    - ✓ Nastavuje se při bootu, potom ani root nemůže provádět změny
    - ✓ Je i v Linuxu: chattr pro ext2/ext3 (lze však měnit kdykoliv)

Atributy souborů v ext2/ext3

- Prohlížíme příkazem **`lsattr`**
- Nastavujeme příkazem **`chattr`**
- Atributy
  - A – čas posledního přístupu není aktualizován (vyšší výkon)
  - a – do souboru lze pouze přidávat data (nelze smazat nevhodný záznam např. v logovacím souboru)
  - c – soubor bude na disku komprimován
  - d – soubor nebude zálohován programem dump
  - i – soubor nemůže být nijak modifikován (bezpečnost)
  - j – určuje jakým způsobem se ukládají žurnálová data
  - s – při mazání souboru jsou uvolňované bloky vynulovány
  - S – při každém zápisu je provádí sync
  - t – zakazuje částečné fragmenty
  - u – i při smazání souboru se obsah ponechává (pro undelete)
- Atributy může nastavovat jen administrátor (root), může je kdykoliv jakkoliv změnit, proto je jejich význam pro bezpečnost jen omezený

Implemetace ACL – moderní UNIX

- Jen trojice rwx pro vlastníka, skupinu a ostatní není dostatečně jemné
- Moderní UNIXové systémy se snaží tyto trojice doplnit skutečnými ACE
  - Tyto ACE obsahují výjimky ke standardním přístupovým právům (výše uvedené trojici)
  - Můžeme tak odebrat právo jednotlivci ve skupině, případně přidat právo jinému uživateli
- Tento přístup implementují mnohé komerční UNIXové systémy (např. VAX, …) a dnes i většina open source systémů
- ACL pro souborový systém extX se stalo standardní součástí jádra Linuxu >2.5

POSIX ACL

- Klasická práva pro vlastníka (ACL_USER_OBJ), skupinu (ACL_GROUP_OBJ) a ostatní (ACL_OTHER) zůstávají
- Nově je možné přidávat výjimky pro
  - uživatele (ACL_USER)
  - skupiny (ACL_GROUP)
- a nastavit masku (ACL_MASK), která dále omezuje práva udělená uživatelům (ACL_USER) a skupinám (ACL_GROUP_OBJ a ACL_GROUP) [vlastníka se netýká]
- a u adresářů lze nastavit defaultní ACL pro nově vytvářené objekty
- ACL zapisujeme ve formě typ_subjektu:subjekt:práva
  - např. user:zriha:rw-

POSIX ACL - příklad

- ACL zjistíme příkazem **`getfacl`**
- ACL nastavíme příkazem **`setfacl`**
- Příklad:
  - **`getfacl soubor`**

```
user::rw-                                (vlastník)
user:lisa:rw-        #effective:r--      (uživatelka lisa)
group::r--                               (skupina)
group:soft:rw-       #effective:r--      (skupina soft)
mask::r--                                (maska)
other::r--                               (ostatní)
```

  - Zkráceně lze psát tato práva jako:
    - g:soft:rw,u:lisa:rw,u::rw,g::r,o::r,m::r
  - Zde maska omezuje w právo uživatelky lisa a skupiny soft

Omezení práv uživatele „root“

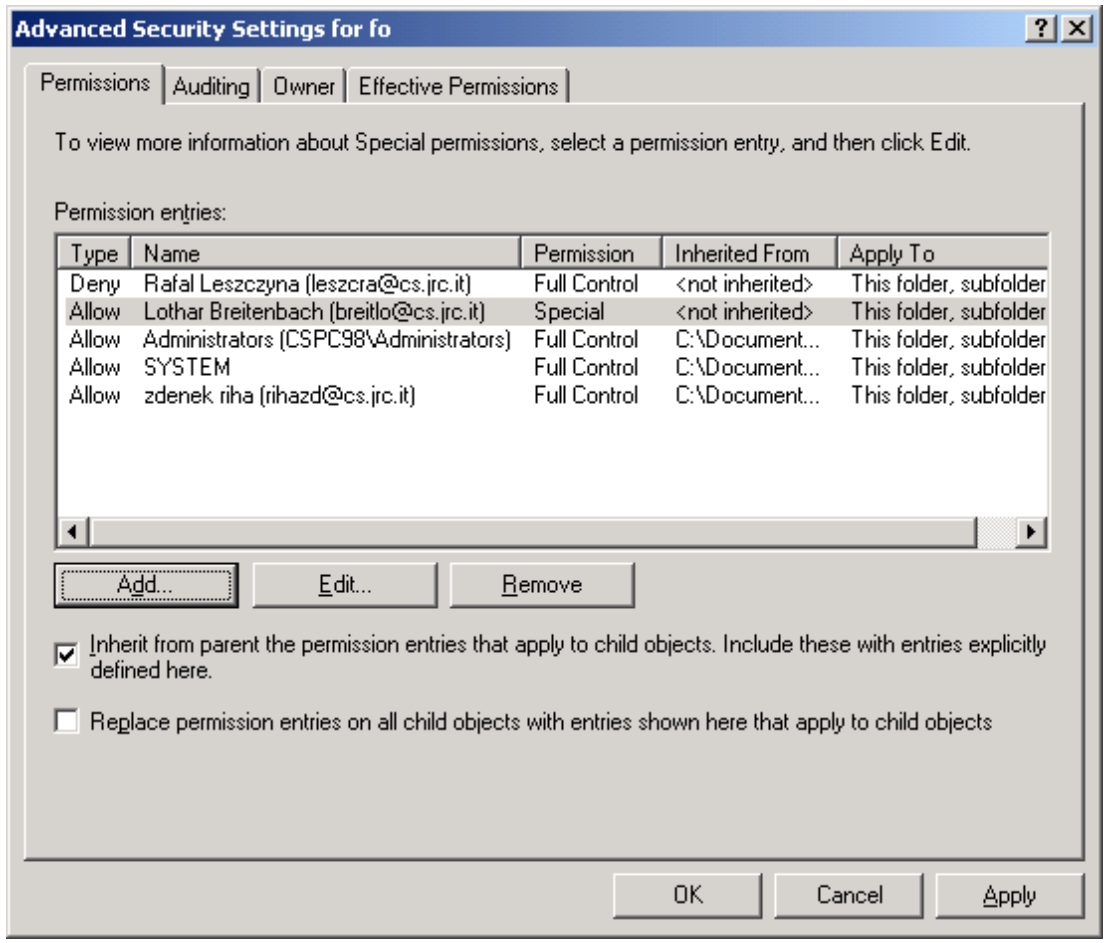
- V klasickém UNIXu:
  - Uživatel s UID 0 – může „vše“, kontrola práv se neprovádí
  - Uživatel s UID jiným než 0 – každá operace podrobena kontrole práv
  - Řada aktivit vyžaduje UID 0
- Snaha o omezení supermocného UID 0
- POSIX capabilities porcují práva do jednotlivých „capabilities“
- V Linuxu od 2.2 (pro procesy, pro soubory neimplementováno)

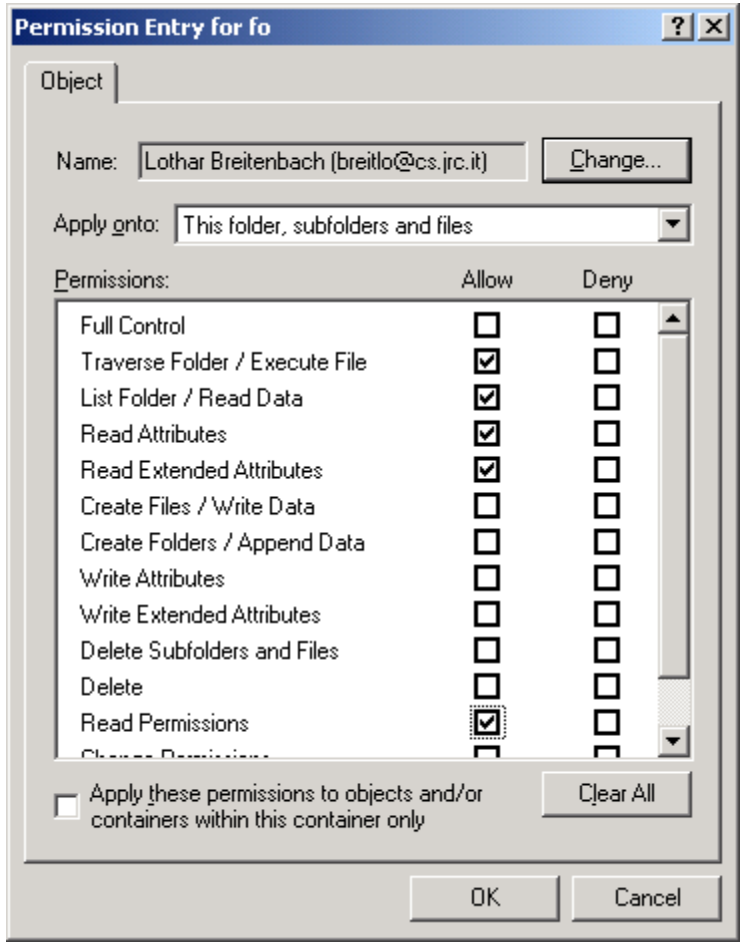
```
CAP_AUDIT_CONTROL (since Linux 2.6.11)
CAP_AUDIT_WRITE (since Linux 2.6.11)
CAP_CHOWN
CAP_DAC_OVERRIDE
CAP_DAC_READ_SEARCH
CAP_FOWNER
CAP_FSETID
CAP_IPC_LOCK
CAP_IPC_OWNER
CAP_KILL
CAP_LEASE
CAP_LINUX_IMMUTABLE
CAP_MKNOD
CAP_NET_ADMIN
CAP_NET_BIND_SERVICE
CAP_NET_BROADCAST
CAP_NET_RAW
CAP_SETGID
CAP_SETPCAP
CAP_SETUID
CAP_SYS_ADMIN
CAP_SYS_BOOT
CAP_SYS_CHROOT
CAP_SYS_MODULE
CAP_SYS_NICE
CAP_SYS_PACCT
CAP_SYS_PTRACE
CAP_SYS_RAWIO
CAP_SYS_RESOURCE
CAP_SYS_TIME
CAP_SYS_TTY_CONFIG
```

Implementace ACL – Windows

- Nejen číst data/zobrazovat obsah složky, zapisovat data/vytvářet soubory, spouštet soubory/procházet složku, ale také
  - Číst atributy,
  - Číst rozšířené atributy,
  - Připojovat data/vytvářet složky
  - Zapisovat atributy,
  - Zapisovat rozšířené atributy,
  - Přebírat vlastnictví,
  - Číst oprávnění,
  - Měnit oprávnění,
  - Odstraňovat
- a to pro uživatele nebo skupiny uživatelů.
- Atributy nejen Povolit/Odepřít, ale také možnost auditování úspěšného či neúspěšného pokusu o přístup.
- Větší možnosti nastavování přístupových práv znamenají možnost přesněji postihnout/implementovat bezpečnostní politiku.
- V praxi však často uživatel přijde, naloguje se jako administrátor (aby mohl instalovat aplikace apod.) a jako administrátor pracuje do ukončení své práce.

Windows ACL – příklad





Řízení přístupu na čipových kartách

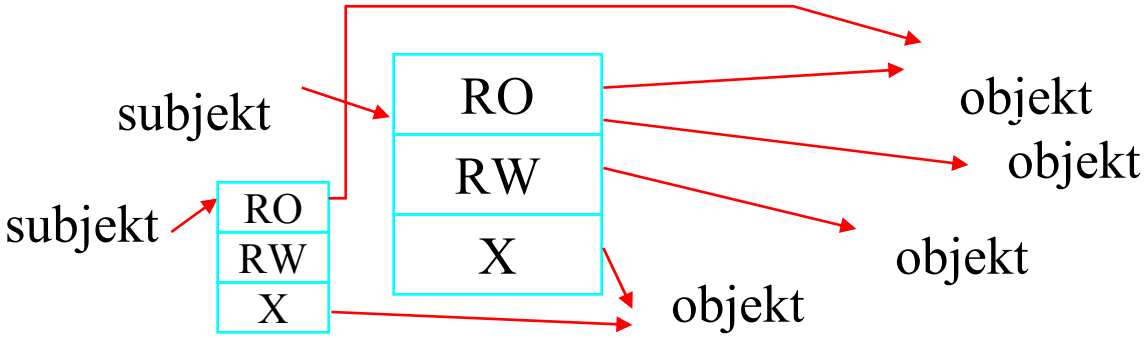
- Řízení přístupu k datům na kartě je tvořeno především řízením přístupu k souborům.
- S každým souborem je svázána hlavička souboru, která určuje přístupová práva k souboru.
- Základním principem řízení přístupu je zadávání PINů a jejich management.
- Přístup k souboru může být například vázán na splnění některé z těchto podmínek:
  - ALW (vždy povolen přístup)
  - CHV1 (nutné zadat PIN uživatele 1)
  - CHV2 (nutné zadat PIN uživatele 2)
  - NEV (přístup nepovolen)

Čipové karty – PIN management

- PINy jsou ukládány v samostatných souborech (EF). Přístupová práva k těmto souborům určují možnost změny těchto PINů.
- Při změně PINu je požadavek provázen starým a novým PINem.
- Počet neúspěšných pokusů bývá omezen. Po překročení limitu (3 – 5) je PIN blokován.
- Pro odblokování je třeba zadat PIN a odblokovací PIN (u SIM karet nazýván PUK).
- I počet neúspěšných odblokování je omezen.

Seznam přístupových oprávnění

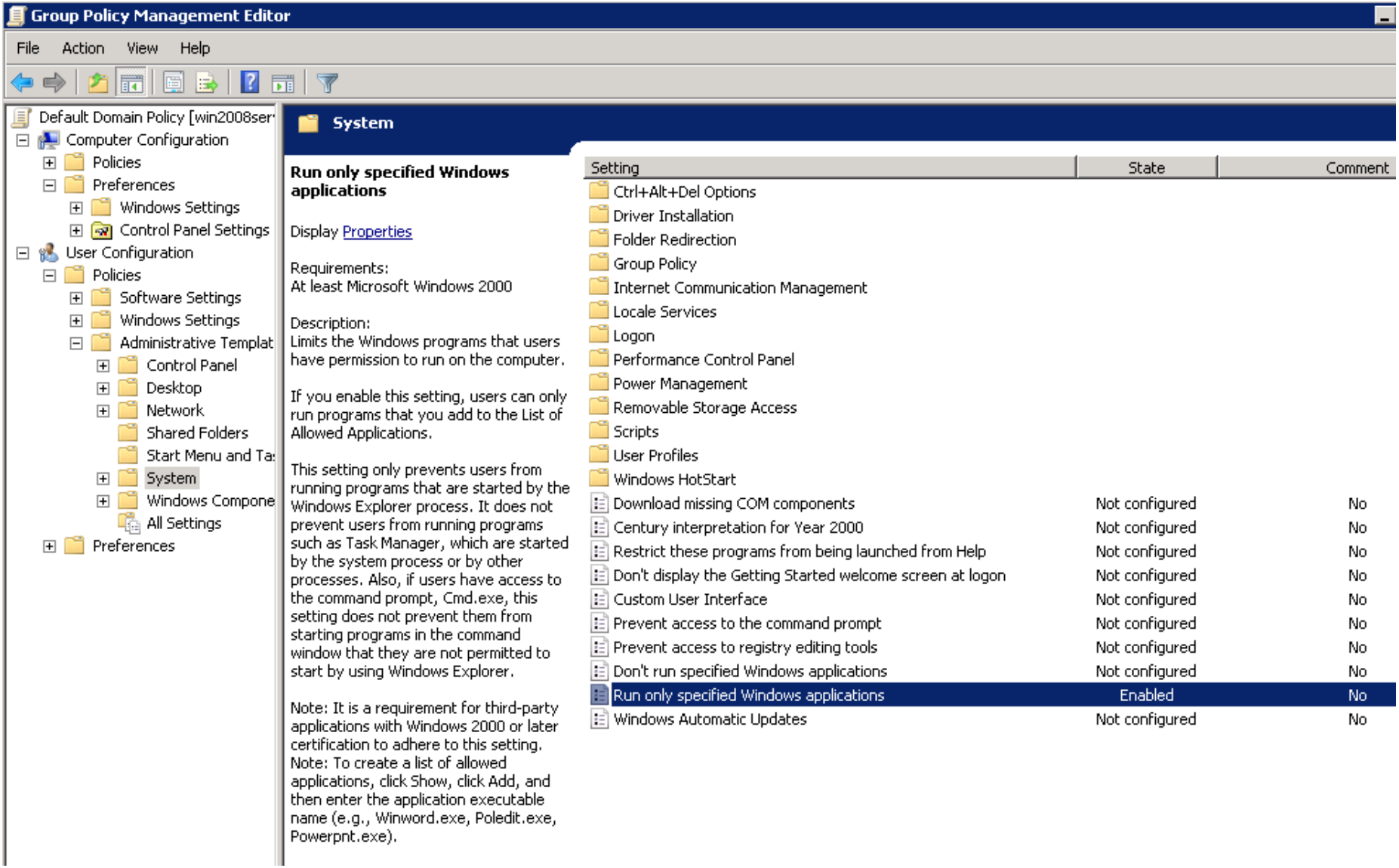
- Seznam přístupových oprávnění (capabilities)
- Ukládáme matici přístupových práv po **subjektech**
- Není žádnou novinkou
- Například: model Multics, IBM AS/400
- Dnes často ve formě certifikátů
- Výhoda seznamu přístupových oprávnění
  - Efektivní kontrola přístupových práv při přístupu k objektu
- Nevýhoda seznamu přístupových práv
  - Nesnadné zjistit kdo má k určitému objektu přístup



Skupinové politiky

- Windows (2000 a výše; omezeně i dříve) implementují nejen ACL, ale i přístupová práva ve formě přístupových oprávnění.
- Tato bezpečnostní oprávnění mohou převážit nebo doplnit přístupová práva ve formě ACL.
- Bezpečnostní politika je svázána se **skupinami** uživatelů [skupiny (groups) jsou definovány v aktivním adresáři (active directory)].
- „Skupinové politiky“ (“Group policy”) [dříve „system policy“] se vztahují na domény, počítače, celé organizace.
- Příklad: Skupinová politika obsahuje seznam aplikací, které skupina uživatelů nemá oprávnění spouštět (např. internet explorer, outlook express, …). Přístupová práva programu na disku (internet explorer) jsou nastavena na [everybody: rx]. Skupinová politika převáží přístupové práva souboru (aplikace) → spuštění aplikace je zakázáno.

Skupinové politiky



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

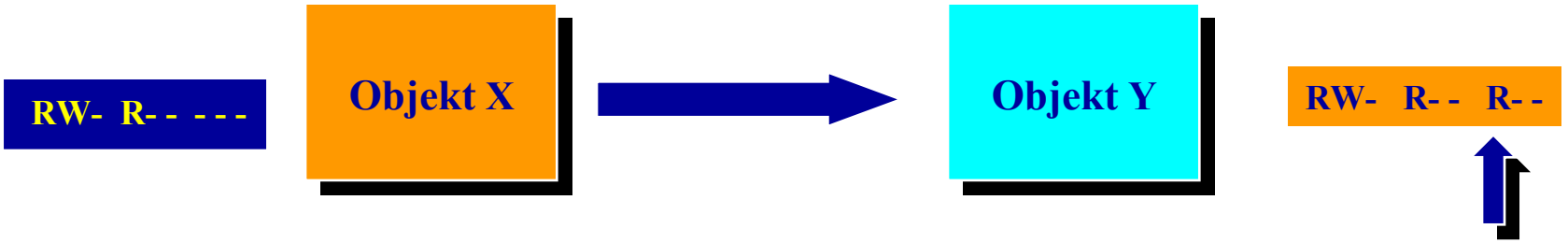
- **povinný přístup** (mandatory)
  - systémová politika nezávislá na vůli subjektů rozhoduje o tom, kdo má k objektu přístup

Volitelné řízení přístupu – výhody

- Jednoduchost, proto malá režie
- Velká vyjadřovací schopnost
- Někdy možné vázat udělení přístupových práv na splnění dodatečných časových, místních apod. 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 koňům"
- Systém se nestará o využití jednou získaných dat
  - např. dám skupině právo čtení na můj důvěrný soubor, nějaký člen si ho zkopíruje a nesprávně nastaví přístupová práva



Spouštění nedůvěryhodného kódu

- Jak se chovat k nedůvěryhodnému kódu, získanému např. z internetu?
  - Autentizace kódu, viz např. Authenticode
  - Speciální jazyk neumožňující škodlivou činnost (JavaScript)
  - Kontrolní programy, které před spuštěním kontrolují kód na škodlivou činnost
    - ✓ Obecně nerozhodnutelné
    - ✓ V praxi heuristiky antivirových programů
  - Spuštění kódu s minimálními uživatelskými právy (např. unixový nobody)
  - Omezené prostředí, ve kterém nemá kód přístup k určitým prostředkům („Sandbox“)
    - ✓ Např. interpret Javy – Java Virtual Machine – applety stažené z webu (bez přístupu na disk, komunikace možná jen s původním serverem)

Podpora řízení přístupu v HW

- Přístup do paměti, ke strukturám OS:
  - Úkolem je zamezit komunikaci/ovlivňování procesů jinak než explicitně povoleným způsobem
  - Např. „fence address“ – limit paměti, do nižších adres má přístup jen operační systém
  - Např. „segmentové adresování“ – Adresování formou segment+offset. Segment může měnit jen operační systém (referenční monitor)
  - Např. dva režimy procesoru – autorizovaný a neautorizovaný. V neautorizovaném režimu není možné měnit segmentové registry.
  - Např. „Rings of protection“ – několik režimů činnosti s různými právy. Měnit ring možné jen v režimu ringu 0 (operační systém).
    - ✓ Volání rutin operačního systému – změna ringu: volání brány (GATE), přerušení.

Objektové programování

- Přístup k datům může být omezen pouze na explicitně uvedené metody
- Příklad v C++

```
class example {
private:
int counter;
protected:
void add_substract(int);
public:
void decrease(void);
void increase(void);
};
```

Řízení přístupu

- Jedna z nejdůležitějších komponent bezpečnosti jakéhokoliv systému
- Bohužel ne vždy je kód řízení přístupu (referenční monitor) bezchybný. (typicky nedostatečná kontrola nedůvěryhodného vstupu a následný „buffer overflow“)
- Příliš mnoho kódu operačního systému je označeno za důvěryhodný (jádro obsahuje ovladače nejrůznějších zařízení napsaných programátory řady firem/organizací)
- „Race condition“ – operace ověření přístupových práv a použití práv není atomická (zneužitelné u programů s většími právy – SUID/SGID programy, webové CGI aplikace)
- Trojští koně

Řízení přístupu

- Separace oprávnění – pro vykonání určité akce je potřebný souhlas několika osob (např. velkou bankovní transakci musí podepsat dva bankovní úředníci)
  - Řízení přístupu na úrovni OS (často ani middleware) toto nepodporuje
  - Nutná podpora přímo v aplikačním SW

- Princip nejmenších privilegií – uživatel má právo přístupu jen k takovým objektům, ke kterým z titulu svého pracovního zařazení přístup potřebuje. Počátečně je množina oprávnění malá a postupně se rozrůstá. Žádný uživatel nemá přístup k objektům, ke kterým přístup nepotřebuje.

Otázky?

Příští přednáška 10. 5. 2011 v 10:00

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