Martin Zukal

RBAC

(ROLE BASED ACCESS CONTROL)

Vznik a standardizace

- Dříve jen DAC, MAC -> potřeba nového bezpečnostního modelu se snadnou správou
- Koncept rolí se v softwaru používá už od poloviny 70. let.
- Myšlenka z uživ. skupin OS, databázových systémů
- Vznik 1992 – Ravi Sandhu, David F. Ferraiolo, D. Richard Kuhn
- 90. léta – vývoj, nestandardizované implementace
- Formální standardizace NIST – r. 2000

Ve standardizaci poprvé zavádí RBAC model + pojmy:
Subjekt – aktivní prvek (člověk, počítač, proces)
(reference monitor)
Objekt – pasivní prvek (soubor, adresář, tiskárna, paměť)
Transakce – operace autorizovaná pro roli
Identifikace a autentizační proces (např. login) není považován za transakci.

Aktivní role subjektu
Autorizovaná role subjektu

Důvody a přínos RBAC

Důvody zavedení

•Správa rozsáhlých systémů je komplikovaná
•V systémech řádově desítky práv (SELinux 30)
•nemožnost efektivní distribuce práv k systému mezi více administrátorů.
•Hrubé přidělování práv uživatelům -> uživ. má víc práv než potřebuje
•Jemné přidělování práv uživatelům -> komplikovaná správa
•Požadovány dynamické změny práv

Výsledkem je vznik rolí

Přínos RBAC modelu

•Vyhneme se používání nízkoúrovňových přiřazování práv jednotlivcům
•Identita uživatele může být nahrazena rolí (utajena) – vhodné hlavně ve webových aplikacích
•Možnost hromadně změnit práva všem uživatelům v dané roli, potomkům role.
•Snížení možnosti vzniku chyby (např. neúplný výčet práv).
•Role korespondují s pozicemi, které uživatelé zastávají v organizaci, jsou popisnější.

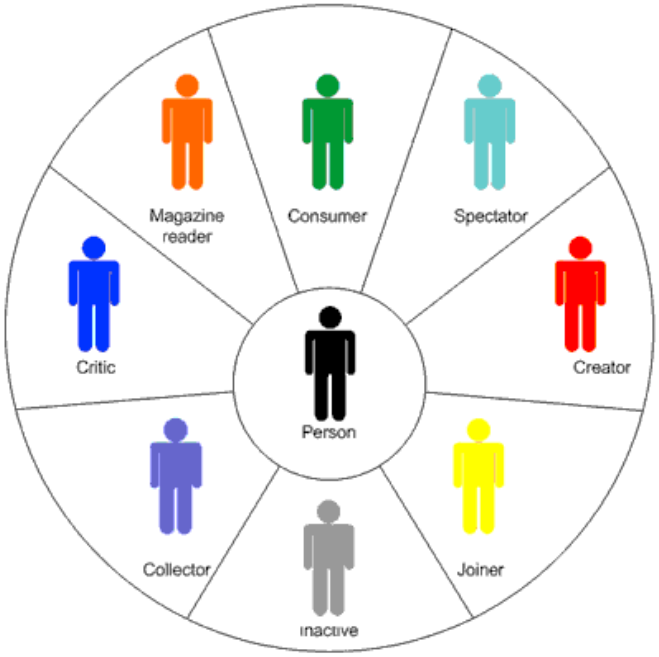
->Vede k snížení nákladů a času na správu systému

Charakteristika RBAC

RBAC rozvíjí mechanizmy MAC -> dodržuje pravidlo:
Uživatel má k dispozici minimální množství privilegií, které potřebuje k výkonu své práce.

2 zákl. přiřazení - uživatel - role
- práva - role

Role – typicky odvozena od pracovní funkce v organizaci, má přiřazenou informací týkající se autority a odpovědnosti v této funkci.



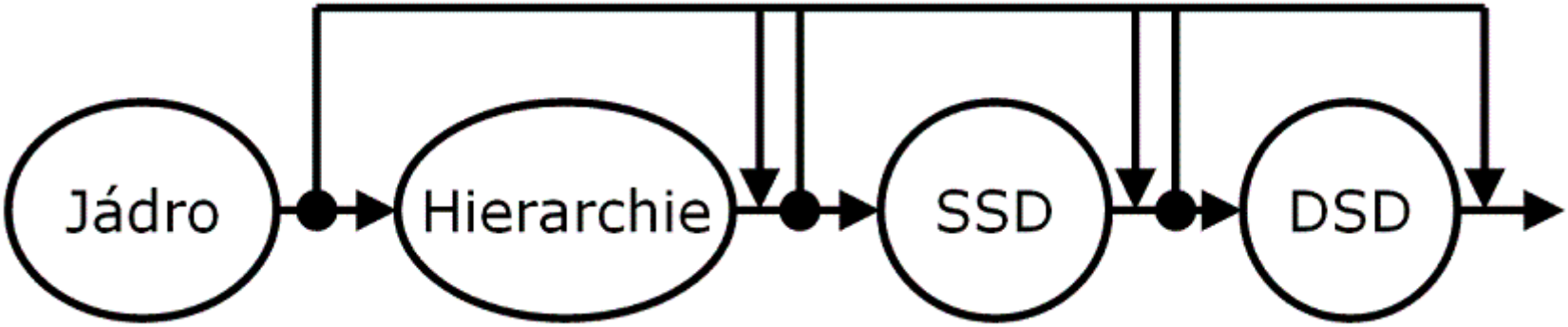
- Roli přiřadíme uživateli nebo skupině
- Uživatelem dnes může být i stroj
- Uživatel vystupuje pouze v jedné roli v každém systému, nebo ve více rolích současně v rámci jedné sítě

- Roli přiřadíme práva
  - Přístupová práva k souborům
  - Systémová práva

RBAC model

Celkem 4 stupně RBAC modelů: RBAC0, RBAC1, RBAC2, RBAC3

- RBAC0 – role nemají žádnou hierarchii, definovány pouze komponenty jádra RBAC
- RBAC1 – zavádí hierarchii (dědění práv) rolí
- RBAC2 – zavádí statická omezení (SSD)
- RBAC3 – zavádí dynamická omezení (DSD)

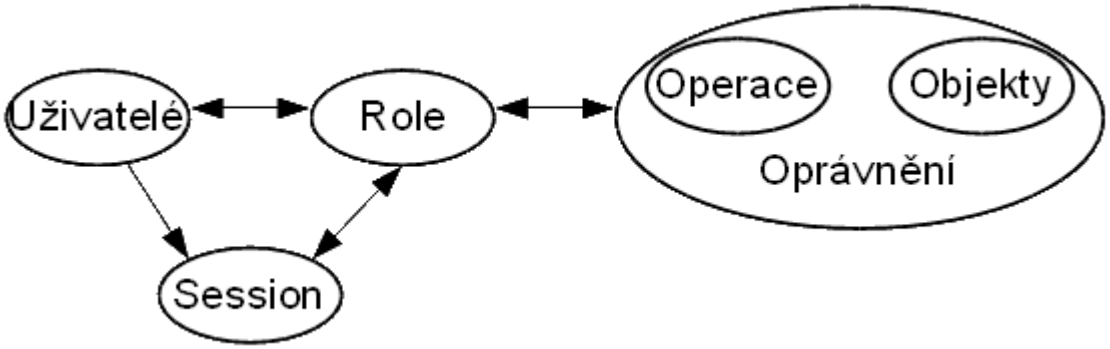


- Pokročilé modely
- Packages

Komponenty modelu RBAC

Jádro (Core) RBAC

- Nikdy nepřidělujeme oprávnění přímo uživatelům
- Oprávnění přiřadíme rolím, role uživatelům
- Role – soubor oprávnění
- Sessiony umožňují selektivní aktivaci a deaktivaci rolí pro daného uživatele
- Možnost vytváření, mazání rolí



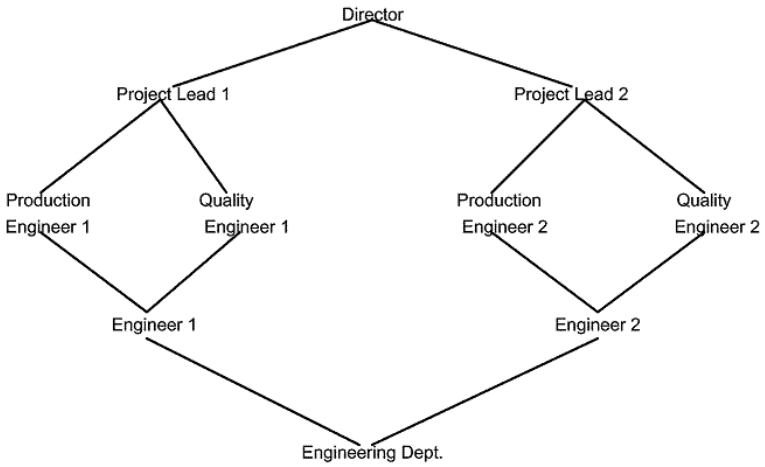
- Oprávnění – určuje jaké operace smí daná role vykonávat nad objekty
- Objekt – soubory, složky, systémové zdroje, řádky tabulky, …
- Dynamické přiřazování: role – uživatel, role - oprávnění

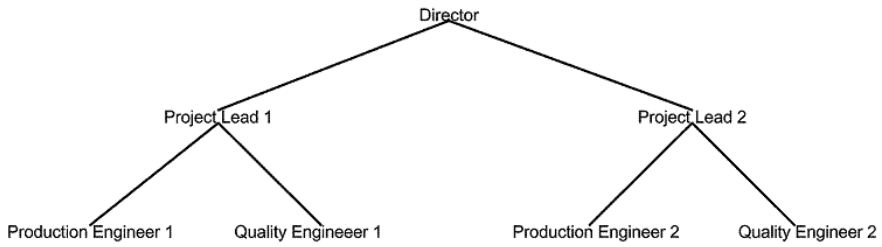
Komponenty modelu RBAC

Hierarchie rolí

- Zavádí *senior role* a *junior role*
- Senior role získávají práva od svých juniorů
- Junior role dědí uživatele svých seniorů (záleží na implementaci)
- Role mohou být členy dalších rolí

1. General Hierarchical RBAC – mnohonásobná dědičnost vpřed i vzad
2. Limited Hierarchical RBAC – omezení, např. stromové grafy





Komponenty modelu RBAC

Je třeba jednoznačně oddělit úlohy jednotlivých uživatelů.
oprávnění jejich rolí – nesmí být konfliktní, musí odpovídat jejich pozici v organizaci (uživatel má oprávnění, které by jeho role vůbec neměla mít)

Static Separation of Duty (SSD)

- RBAC implementuje striktní rozdělení rolí v organizaci.
- Oprávnění rolí jednoho uživatele nesmí být v konfliktu.

SSD udává pevně nastavená pravidla pro příslušnost k jednotlivým rolím
Např. má-li uživatel přiřazenou roli A, nesmí vystupovat v roli B
Dvojice (role set, n)

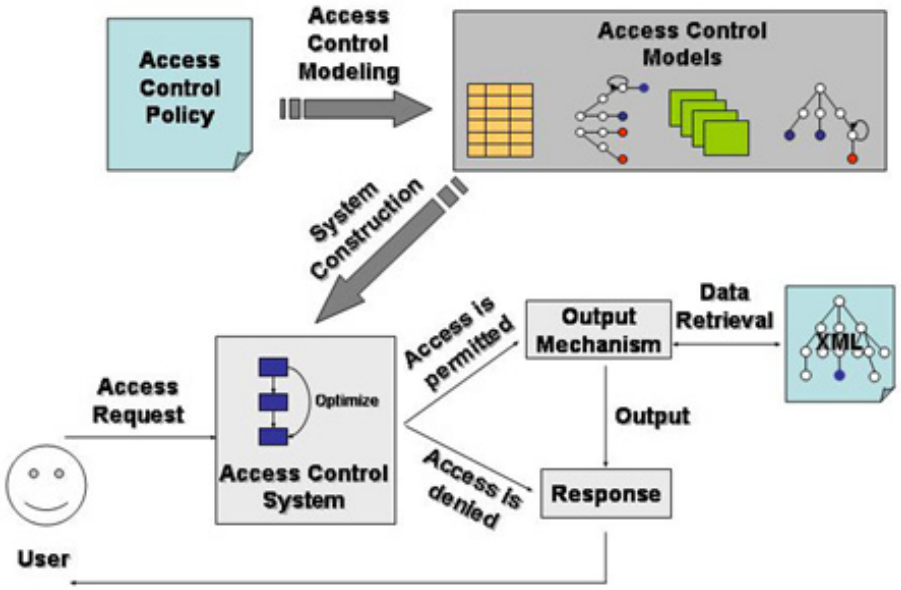
Dynamic Separation of Duty (DSD)

- DSD umožňuje vystupovat v konfliktních rolích
- Zajišťuje, aby sessiony konfliktních rolí neběžely současně
- Dvojice (role set, n>=2)

SSD a DSD pravidla určují systémoví administrátoři

Příklad inicializace systému s RBAC

- Přihlášení uživatele do systému -> inicializace nové session
- První proces session inicializován seznamem rolí uživatele
- První proces je předkem všech procesů v session
- Zahájení nové session -> přihlášení uživatele může rozeznat jádro systému (volání *setuid() v případě Linuxu*)
- Po úspěšném přiřazení uživatel-role následuje přiřazení domény
- Role aktivní -> EnableRole



Příklad práce s RBAC v SELinux

V SELinuxu (i v grsecurity) je třeba RBAC explicitně spustit:

```
# seedit-rbac on
# reboot
```

Defaultně tyto 3 role:

- sysadm_r
- staff_r
- user_r

Změna role:

```
# newrole -r sysadm_r
Authenticating root
Password:
```

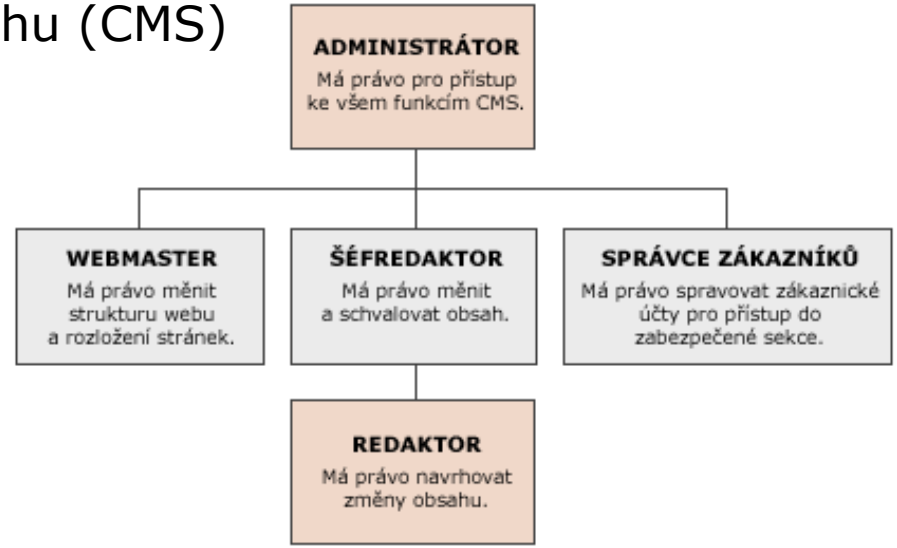
Nastavení uživatele, role, práv (zápis do složených závorek):

```
role webmaster_r;
user webmaster;
allow /var/www/** r,w,s;
```

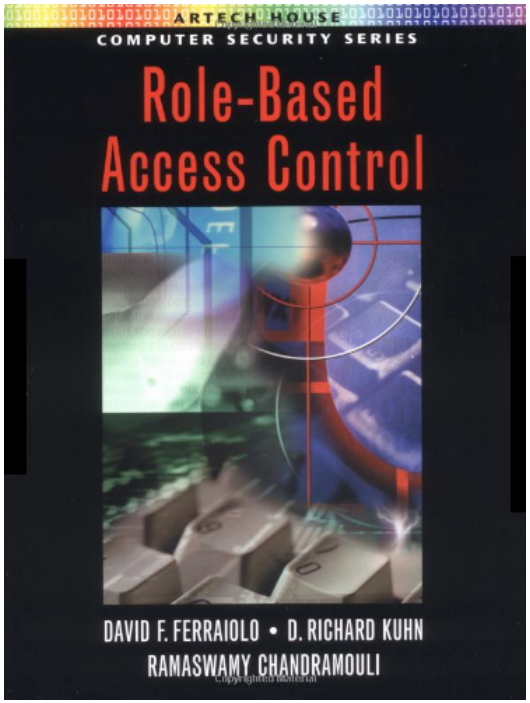
Kde se RBAC používá?

- SELinux (Security Enhanced Linux) - vyvinutý v NSA (National Security Agency)
- Grsecurity – od verze 2.x
- LSM (Linux Security Module) – v jádru od verze 2.6.x
  - Obsahuje framework pro implementaci vlastního bezpečnostního modelu
- MS - Authorization Manager, role-based API (Windows Server)

Příklad webové aplikace s RBAC obsahující vlastní systém pro správu obsahu (CMS)



Literatura



Role-Based Access Control
David F. Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli

Použité zdroje:
1.Ferraiolo David F., Sandhu Ravi, Kuhn D. *Richard. Proposed NIST Standard for Role-Based Access Control*, c2000
2.Nakamura Yuichi. *SELinux Policy Editor RBAC Guide*, c2006
3.Web National Institute of Standards and Technology, dostupné z www: <http://www.nist.gov>