# ,Authorisation and Access Control of Application Data in Workflow Systems

Klucove slova – business proces

Workflow (strana 2)

Potreba definavie prístupu vo workflow (strana 3 -> od polovice)

RBAC – 4strana

Alturi a Huang s work … petri nets … pozriet (AH96A)

DAC – strana 5 … moc sa tam o tom nepíše.. vraj je to rozoberané jedine v práci od autorou vyššie, ale chýba im tam process instance-based group (nerozlišujú medzi typmi dát)

Dalej sat u zaoberá najma medicínskymi aplikáciami

Potom nejaké obrázky hierarchie rolí

Process instance-based user group –strana 9 dole

-popisuje to, že o každého pacienta sa stará určitá skupina lekárov , sestričiek a iba taký môžu k tomuto pacientovi nahliadnut a iný nie ( v našom systeme je to riešené cez referencie .. ale nie je to úplne dokonalé)

Data content - strana 10

-tomuto celkom nerozumiem… ide o to, že ked má pacient veľa záznamov , tak sa tažko na každá záznam nastavujú právomoci … vraj je na to dobré riešenie cez CFSM (predicate-based access control)… .som z toho jeleň

Task – 10 dole

Tu je povedané, že nejaká rola má odlišné právomoci vzhladom na úlohy (niekde može čítat, niekde iba zapisovať)… to máme ošetrene cez formuláre, kde je kolonka “editable”

Privilege propagation- 11-stred

Wtf? … pozrieť ešte

Rola- regulácia 4 … strana 11

… blbost… zase tie veci, ktoré ošetrujeme cez formuláre a process

Dynamic authorization

Regulácia 6…strana 12 ak su vytvorené nejaké grupy , tak by sa mali dat za pochodu menit… napriklad pacient vymení lekára

Dalsie info k process instance-based groups strana 13 … že je to potrebné a blab la bla… je to smutné lebo take nemám

Task 13

14 - … definovanie takéhoto funkcneho modelu

15- komponenty --- role hierarchy, task hierarchy

Dalej už implementačný ballast ktorý sa mi nechce čítať

# ROLE BASED ACCCESS CONTROL

# The Application of Petri Nets to Workflow Management

# Unifying Petri Nets: Advances in Petri Nets

# The Economic Impact of Role-Based Access Control

405 stranová bichľa… skúsim ju zhrnúť

Pozrieť: chapter 12.1 ,13.3 ,13.5, 9, 6, 4, 1.3

Výhody: commercial application , reducing complexity and cost of security. Može zahrnat mac a dac

Strana 23

Information security risks can be broadly categorized

into the following three types, *confidentiality, integrity,* and *availability,*

which can be remembered with the convenient mnemonic “CIA.” These

categories are described as follows:

◗ *Confidentiality* refers to the need to keep information secure and private.

This category may include anything from state secrets to confidential

memoranda, financial information, and security information

such as passwords.

◗ *Integrity* refers to the concept of protecting information from being

improperly altered or modified by unauthorized users. For example,

most users want to ensure that bank account numbers used by financial

software cannot be changed by anyone else and that only the user or an

authorized security administrator can change passwords.

◗ *Availability* refers to the notion that information is available for

use when needed. Attacks that attempt to overload corporate

Web servers, widely reported in the popular press, are attacks on

availability.

Bell and LaPadula [ strans 28)

Strana 32/11

Three basic

rules were required:

1. *Role assignment:*Asubject can execute a transaction only if the subject

has selected, or been assigned to, a role. The identification and

authentication process (e.g., login) is not considered a transaction.

All other user activities on the system are conducted through transactions.

Thus, all active users are required to have some active role.

2. *Role authorization:* A subject’s active role must be authorized for the

subject. With rule 1, this rule ensures that users can take on only

roles for which they are authorized.

3. *Transaction authorization:* A subject can execute a transaction only if

the transaction is authorized for the subject’s active role. In concert

with rules 1 and 2, this rule ensures that users can execute only

transactions for which they are authorized.

!!!strana 12//33 ... formálna reprezentácia rbac + obrázky

15 /36 základné a zložitejšie rbac modely

RBAC represents a

major advancement in flexibility and detail of control from the existing

standards of DAC and MAC

dac

A means of restricting access to objects based on the identity of subjects or

groups, or both, to which they belong. The controls are discretionary in the

sense that a subjectwith a certain access permission is capable of passing that

permission (perhaps indirectly) on to any other subject (unless restricted by

MAC).

20/41 matematicky znázornené , kedy je výhodné pouzit rbac

22/43 zobrazené že systém rolí zabezopečuje oprávnenia pre viac aplikácii (hlavne rieši problém s pridávaním aplikácii)... v našom prípade procesy7 ... veľmi zaujímavé ... určite to spomeniem

2.2.2 32/53

Princípy bezpečného prístupu

Salzer and Schroeder identified several design principles pertaining to protection

mechanisms [15]. Although first proposed in 1975, before the rise of

networking and the Internet, these design principles continue to hold today.

(Some of the principles in fact can be traced to the rules for military ciphers

2.2 Access control: core entities and principles 31

proposed by Auguste Kerchoffs in the nineteenth century.) They promote

simplicity, isolation, confinement, and ease of use:

1. *Least privilege:* Every user and process should have the least set of permissions

or privileges necessary in conducting the task at hand. The

implementation of this principle has the effect of limiting damage

that can result from system error or malicious events. When considering

a user, permissions should be carefully assigned and

periodically reviewed to ensure they minimally fit the user’s functional

needs. When considering a process, a minimum subset of the

evoking user’s security attributes with associated permissions should

be activated during the course of a session.

2. *Economy of mechanism:* The design should be sufficiently small and

simple so that it can be evaluated and shown to be correct. Simple

means that less can go wrong and when errors do occur, they are easier

to identify and fix. The application of this principle usually entails

implementing the protection mechanism at the lowest and most

protected levels of the system possible, where the higher levels (e.g.,

applications) are controlled by the lower levels.

3. *Fail-safe defaults:* Access decisions should be based on inclusion rather

than on exclusion. The default should be lack of access. If the protection

mechanism should fail, then legitimate access is denied, but

illegitimate access is also denied. Note that there may be a trade-off

between this rule and the security goal of *availability*.

4. *Complete mediation:* Every request for access by a subject should be

checked for authorization. If permissions change, different results

are computed. Any caching of results should not be permitted.

5. *Open design (Kerckhoffs’ law):* The premise of security should not

depend on the design being secret. If the design is sound the system

should be secure. The more eyes the greater the likelihood of success.

This principle has traditionally been applied to cryptographic system

where the algorithms are subject to public scrutiny.

6. *Separation of privilege:* Where possible, a protection mechanism

should depend on multiple conditions being satisfied, such as requiring

cooperation from two independent entities, or requiring

cosigners.

32 Access Control: Properties, Policies, and Models

7. *Least common mechanism:* Minimize the sharing of mechanisms by

multiple users. The implementation of this principle includes isolation

through physically separate systems (sandboxes) or though

logically through virtual machines.

8. *Psychological acceptability:* The protection system interface should be

easy to use so that users accept the protection mechanism correctly.

The complexity of the protection system should be transparent to the

user. The user should not have to logoff and back on in performing

normal tasks.

The RBAC model taxonomy consists of four models—core RBAC, hierarchical

RBAC, static constrained RBAC, and dynamic constrained RBAC

3.2 61/82 Core RBAC

Core RBAC recognizes five administrative elements: (1) users, (2) roles, and

(3) permissions, where permissions are composed of (4) operations applied

to (5) objects. Central to RBAC is the concept of role, where a role is a

semantic construct around which access policy is formulated. The most basic

of these relations are user and permission assignments. In RBAC, permissions

are associated with roles, and users are made members of roles,

thereby acquiring the roles’ permissions. Figure 3.1 shows the relationship

between users, roles, and permissions. Figure 3.1’s use of double-headed

arrows indicates a many-to-many relationship. For example, a single user

can be associated with one or more roles, and a single role can have one or

more user members.

This arrangement provides great flexibility and granularity of assignment

of permissions to roles and users to roles. Any increase in flexibility in controlling

access to resources also strengthens the application of the principle

of least privilege.

As an alternative to providing these conveniences, it is often the practice

to establish user permissions based on a concept of “cloning.” Cloning is the

practice of assigning permissions to a user based on the duplication of permissions

of a second user who performs a similar function to that of the first

user. Cloning is usually performed without regard to the details of the permissions

that are assigned to users. Although cloning may be a quick and

efficient method for the establishment of permissions, due to the coarse

nature of permission assignment, cloning is generally considered to be a

dangerous practice.

3.2.2 Role activation 64/85

Hovorí o potrebe priradit role operácie a objekty ... čiže u nás tasky a v taskoch sa definuje čo može robit

4.3.2 85/106 Organization chart hierarchies .. .departmenty

Chapter 5 ... pravidlo dvoch ludí ... kvoli security strana 97/ 118

6.2 132/153

Simulating DAC on RBAC is not a straightforward process, despite DAC’s

apparent simplicity. Part of the difficulty is that RBAC is by nature a

6.2 Enforcing MAC on RBAC systems 131

nondiscretionary approach to access control. Perhaps because of its

nondiscretionary nature, RBAC can be configured to implement MAC much

more easily than DAC. This problem has been studied by a number of

authors, including Nyanchama and Osborn [1], Sandhu [2], and Osborn [7].

A compilation of results by Osborn, Sandhu, and Munawer [3] shows how

to configure RBAC to support lattice-based access control policies, including

several varieties of MAC.

As with DAC, a number of different rules have been proposed for MAC.

These rules have been introduced in Chapters 1 and 3, but are reviewed

here. Recall that a security label combines both a level (e.g., secret,

top-secret) and a set of security categories. One rule common to all multilevel

secure MAC systems is the simple security property, for subjects and

objects with security levels given by *L*(*s*) and *L*(*o*), respectively.

Strana 189/211

9-administrácia rolí - model administratívnych a uzivatelských rolí

Strana 273 / 293

12.1.1

Based on the definition provided by the Workflow Management Coalition

(WFMC) [1], an international organization of workflow vendors, users, and

research groups, a workflow is a representation of an organizational or

272 Integrating RBAC with Enterprise IT Infrastructures

business process in which “… documents, information, or tasks are passed

from one participant to another in a way that is governed by rules or procedures.”

A workflow separates the various activities of a given organizational

process into a set of well-defined tasks. Hence, typically, a workflow (often

synonymous with a process) is specified as a set of tasks and a set of dependencies

among the tasks. The various tasks in a workflow are usually carried

out by several users in accordance with organizational rules relevant to the

process represented by the workflow.