2009

AUTOMATING PERIODIC ROLE-CHECKS A TOOL-BASED APPROACH

Ludwig Fuchs
Universität Regensburg

Christian Müller
Universität Regensburg

AUTOMATING PERIODIC ROLE-CHECKS A TOOL-BASED APPROACH

Ludwig Fuchs, Christian Müller[1]

Abstract
The use of roles in Identity Management has proven to be a solution for reorganising and securing the access structures of organisations. One critical challenge companies face after they implemented roles is the maintenance of the role system itself. This includes sophisticated duties like periodically verifying the valid roles. We argue that due to the high complexity, periodic role-checks need to be automated. However, as a result of lacking theoretical foundation, no approaches to leverage the level automation have been published so far. In this work we develop a catalogue of use cases that affect the role definitions within an organisation. We propose checkROLE, a tool for automated role-checking on basis of the defined use case catalogue.

1. Motivation

In today's business environment companies provide access to resources to a greater number of users, and more types of users, than ever before. Major IT security problems arise because of employees gaining unauthorised access to resources as a result of manually handling user accounts ([4], [12]). This situation results in the so called identity chaos. National and international regulations like Basel II [1], the Sarbanes-Oxley Act [19], and the EU Directive 95/46 [5] force businesses to audit the actions within their systems. In-house Identity Management (IdM) is a means to solve the aforementioned identity chaos. It deals with the storage, administration, and usage of digital identities during their lifecycle [8]. Role-based IdM in particular is seen as a means to get compliant in general and to easily manage identities and their access to resources ([6], [10]). However, the central challenge after the implementation of roles is the management and maintenance of the role system itself in order to assure its timeliness. This includes the operative administration of the user-role assignments as well as strategic tasks including, e.g., the administration of role-permission assignments. Several developments within an organisation might affect the role definitions and require the role catalogue to be adapted. With thousands of users and millions of authorisations in big companies this task can't efficiently be carried out manually. Besides the lacking theoretical foundation, no approaches to leverage the level of automation during the process of role-checking have been published so far. In this work we develop a catalogue of use cases that affect the role catalogue within an organisation. In order to show the applicability and advantages of our approach we furthermore propose *checkROLE*, a tool for the automatic

[1] Universität Regensburg, D- 93053 Regensburg, Universitätsstr.31

detection of the defined use cases. Using checkROLE organisations can automatically identify changes in their role definitions and keep their role catalogue up-to-date.

This paper is structured as follows. After an introduction to Role System Management and periodic role-checks in the related work section, we introduce a catalogue of use cases as the theoretical foundation for automating role-checks in section 3. After a detailed description we analyse selected use cases showing their influence on role definitions. In section 4 we then consecutively propose checkROLE. Section 5 gives conclusions and points out future work.

2. Related Work

2.1. Role System Management

Operative and strategic management of roles and the role system in general are essential tasks to keep the implemented role definitions usable. In order to avoid misunderstandings we define the terms used in the following: Role System Management is the umbrella term for operative Role Management and strategic Role Management (Role Maintenance). Operative Role Management includes routine administration duties like user-role-assignment or role-permission-assignment according to the given administration model. Various authors investigated this area proposing several role administration publications like [2], [14], [15], [16], [17], [18], and [21] as well as role system lifecycles ([9], [11], [20]). However, operative Role Management can only be carried out effectively on basis of correct role definitions. Besides the maintenance of the underlying role concept and -model, the most important Role Maintenance duty is the up-to-date keeping of the role catalogue on basis of strategic Role Management processes. In contrast to its operative counterpart, this task heavily depends on organisational and operational structures (OOS) within a company. With dozens of business processes, thousands of users and millions of authorisations in big organisations, strategic Role Management is a seemingly difficult task. Only a few authors theoretically touched this issue on the brink ([11], [20]) while hardly any of them consider automation of Role Maintenance processes. The goal is to face Role Maintenance challenges by analysing existing role- and permission structures in order to provide suggestions for necessary changes in the role definitions, role-permission-, and user-role assignments. This way inconsistent permission assignments endangering compliance with security principles can be cleansed.

2.2. Periodic Role-Checks

The importance of periodic role-checks as central part of the Role Maintenance duties has recently been pointed out by [9]. The goal of periodic role-checks is to evaluate the elements of a role system at a certain point of time in order to identify changes within a company that affect the role definitions. Values of role model elements of the last valid state are compared with their actual values in the productive systems. Periodic role-checks have to rely on existing user and access information as well as role definitions stored in the Identity Management Infrastructure or other user repositories. Together with organisational structures and the basic employee information coming from various directories this is a reliable and permanently available source for adequate user information. After the input information has been gathered the comparison of the output OP(t) of the productive user management environment at time *t* against the last valid role catalogue state OR(t-1) on basis of predefined use cases indicates events that affect the role definitions (see *Figure 1*). A consecutive impact analysis provides further assistance for resolving the discrepancies. Modifications in the organisational structure and user population are highlighted, for example when employees move to another department or a large number of new employees are hired. On basis of

this assistance a decision for resolving the open issues has to be made manually in an analytical way together with business- and IT representatives. In general, one can say that this duty is an iterative process, reducing inconsistencies all along the overall role-check process loop. As seen in *Figure 1*, periodic role-checks need to be carried out on basis of a well-defined use case catalogue. To the best of our knowledge no such basis has been proposed yet and no insight about underlying use cases as well as practical implementation issues is given in literature. In the following we close this gap by developing a use case catalogue that can be used as theoretical basis for role-checks.

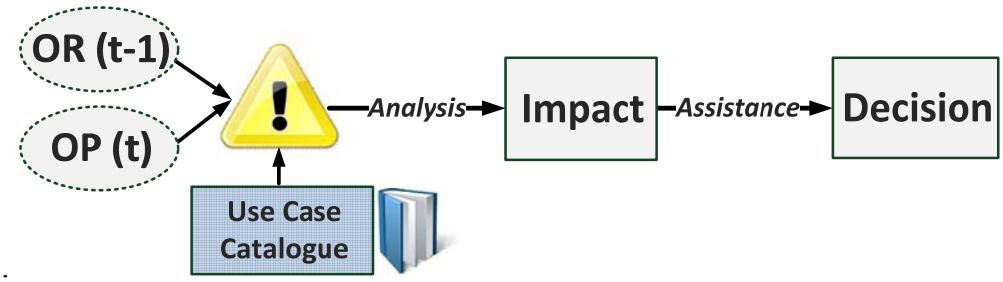

Figure 1: Role-checking on basis of a given use case catalogue

3. Automatic Role-Checking

3.1. Input Elements

Before designing a comprehensive use case catalogue we identified the various elements within organisations affecting role-checks and the defined role catalogue using busiROLE [7] as underlying role model. BusiROLE supports the usage of various types of roles, namely Basic Roles, Organisational Roles, and Functional Roles and is applicable in complex role environments. Basic Roles bundle common access rights, Organisational Roles represent employees' positions, and Functional Roles correspond to the task bundles of employees. Basic Roles can be regarded as special Organisational Roles. Our research was carried out on basis of practical experience with various partners from industry on the one hand and scientific publications in the business administration- and role-based user management area ([3], [11], [13], and [20]) on the other hand. The initial analysis revealed a number of components influencing the valid set of roles (see tables 1-6): *Employees*, *Organisational Hierarchies*, *Positions*, *Task Bundles*, and *Permission (Bundles)*. Employees e.g. could leave the company or be assigned to a different hierarchical element. A hierarchical element is defined as a unit in the organisational structure of an enterprise, for example a business unit, a department, or a unit within a department. In other cases employees might be promoted and assigned to a new Position and new Task Bundles. Examples for change of Organisational Hierarchies include mergers or major restructuring efforts within a business area. Additionally, Permission Bundles of users might change over time. New IT systems might be implemented and the respective rights assigned to the employees while old system might no longer be used. Various constraints, e.g. security policies restricting the user-role- and user-permission-assignments can also affect Role Maintenance. However those constraints are usually not stored in a way that they could automatically be integrated into the role-checking process. Hence we focus on the previously introduced elements because information about those elements is likely to be available in an appropriate format. A definition of every element is given in the following to found the basis of standardised use case catalogue description in section 3.2.

Tables 1-6: Definition of input elements for role-checks

Employees	
EMPS	A set of all employees
$EMP_a \in$ EMPS	A human being working for a certain enterprise

Organisational Hierarchies	
OH	A set of all Organisational Hierarchies within an enterprise with OH = {$OHtype_a$, $OHtype_b$, …, $OHtype_n$}
$OHtype_a$	One specific organisational hierarchy type within an enterprise, e.g. the line organisation
$OHEtype_a$	A set of all hierarchical elements within $OHtype_a$
$OHE_a \in OHEtype_a$	An element of a certain organisational hierarchy type $OHype_a$

Positions	
POSITIONS	The set of Positions within OH
$POS_a \in$ POSITIONS	A Position within an $OHE_a \in OHEtype_a$

Task Bundles	
TB	Set of all Task Bundles
$TB_a \in$ TB	A bundle of tasks serving to fulfil a certain business goal

Permission Bundles	
PB	Set of all Permission Bundles
$PB_a \in$ PB	A specific Permission Bundle serving to fulfil a certain business goal

Organisational and Functional Roles	
ORG_Roles	Set of Organisational Roles
FUN_Roles	Set of Functional Roles
$ORG\_Role_a \in$ ORG_Roles	An Organisational Role defined to represent one Position
$FUN\_Role_a \in$ FUN_Roles	A Functional Role defined to represent one Task Bundle

3.2. Use Case Catalogue Definition

After the theoretical definition of input elements for role-checks we derived a comprehensive set of input element-specific use cases that influence the existing role definitions (see *Figure 2*). The elements are classified according to their respective layer of origin: Permissions and Permission Bundles are representing the existing access rights and hence are directory-specific while the other elements are related to operational and organisational structures within a company (OOS-specific). For each element various operations are possible, representing one single use case. Regarding *Organisational Hierarchies*, e.g., "new", "delete", "split", and "merge" are possible operations. Hierarchical elements can be created or deleted as a result of restructuring efforts within a company. The splitting and merging operations can be seen as a combination of the previously mentioned ones. In terms of *Positions* companies might e.g. define new or delete old ones. Splitting an existing Position into two separate units can again be modelled as definition of two new Positions, re-assigning the old Task Bundles appropriately, and the consecutive deletion of the old Position. The same holds for the "merge" or "relocate" operation. Organisations might carry out new *Task Bundles*, delete old ones, or relocate existing Task Bundles to a new Position. The "split" and "merge" operations can again be seen as a special combination of the basic operations. Taking a user-centric view one can easily recognise the various changes an *Employee* can go through when working for a company. He can either change his Position or the assigned Task Bundles, or be

relocated to other hierarchical elements. Directory-specific use cases, finally, are dealing with *Permission (Bundle)* changes. Companies might install new resources, abandon old IT systems, or update existing software. The various examples regarding the "merge", "split", and "relocate" operations already have shown the interdependencies between single events. In practical settings it is likely that they do not occur isolated but combined. We are thus now going to identify complex use cases consisting of a number of the various events shown in *Figure 2*.

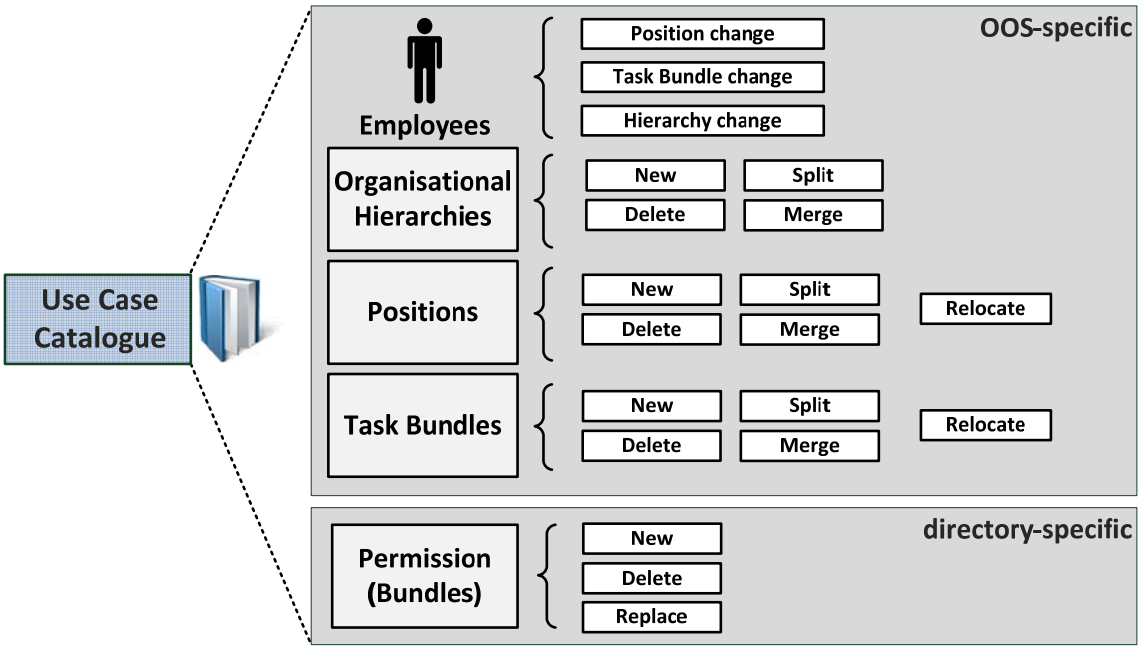

Figure 2: Theoretical use case catalogue

3.3. In-Depth Analysis of Selected Use Cases

In order to foster a clear differentiation and standardised usage we designed a three-layer scheme describing every use case in terms of its *structure*, its *impact* on the role system, and possible *detection* mechanisms. An analysis together with user partners from industry has shown the applicability of this schema as a means of easing communication. We are thus using it during the in-depth analysis of selected use cases in the following.

3.3.1. Selected Use Case 1: Employee Changes Hierarchical Element

Structure (Figure 3):
An employee EMP_c is assigned to a new hierarchical element OHE_i and Position POS_{i2} in the organisation. Due to this fact his work pattern changes and consists of new Task Bundles TB_c and TB_d. Previously he has been working in hierarchical element OHE_j incorporating position POS_{j1} and related Task Bundles TB_e, TB_f, and TB_g.

Impact:
To enable correct execution of TB_c and TB_d according permissions are granted to EMP_c. This means that he needs to be assigned to a certain number of roles corresponding to POS_{i2}, TB_c, and TB_d. However, access right allocation is mostly done manually by IT administrators and only rarely on basis of automated processes that ensure that the permissions requested are granted compliant with current role system policies. Imagine the promotion of an employee that rapidly needs access rights

for his new daily work and thus requests them calling several IT administrators. This usually results in a direct assignment of requested rights, probably on basis of the privileges of EMP_b. Taking a closer look at necessary de-provisioning tasks unused Organisational- and Functional Roles corresponding to the old Position POS_{j1} and Task Bundles TB_e, TB_f, and TB_g have to be revoked. Usually no automatic de-provisioning processes are in place so that this duty is not carried out at all or at most manually. Hence the employee is likely to accumulate a number of excessive rights within the organisations' IT systems, violating the principle of the least privilege [6].

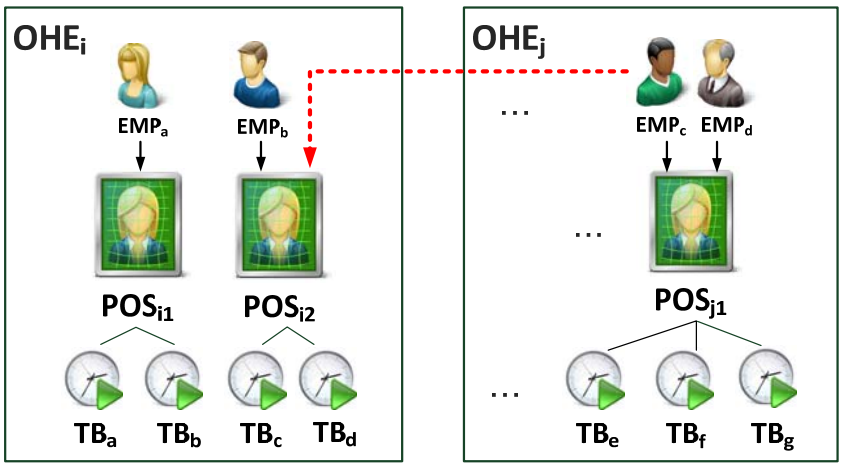

Figure 3: Employee changing organisational hierarchy element

Detection and Actions:
The first step in this scenario is to detect all employees $EMP_i \in EMPS$ which have been assigned to a new hierarchical element. This can be done by analysing the according user attributes e.g. in the global user directory. In a second step, Role Maintenance must on the one hand examine if these employees have been assigned to the correct permissions and roles according to their new OHE_i assignment. In case of any incorrect assignments this situation has to be resolved together with responsible executives. Role Maintenance has to check if old and thus unused access rights and role memberships have been correctly revoked. Statistical analysis and the integration of policies that e.g. define that a user is not allowed to be member of more than one Position in a certain organisation hierarchy type $OHtype_a$ can be facilitated to address this problem.

3.3.2. Selected Use Case 2: New Hierarchical Element is Created

Structure (Figure 3):
A new hierarchical element OHE_k is created within a company due to certain organisational changes. It e.g. emerges from splitting an existing OHE_l into two separate hierarchical elements. Scientific publications in the business administration area and in the area of organisational behaviour contain additional insight into reasons for change of hierarchical structures within an organisation ([3], [13]). In this scenario the creation of a hierarchical element results in the definition of a new position POS_{k1} and assignment of Task Bundles TB_d, $TB_e \in TB$. Employees are newly hired (EMP_{new}) or move from existing hierarchical elements to the newly created one (EMP_a).

Impact:
After the creation of a new hierarchical element appropriate Organisational- and Functional Roles have to be defined. Role development has to be carried out for the new hierarchical element. However, in real-life settings employees quickly need permissions in order to fulfil their workload.

Imagine the newly created department OHE_k in a large company. Two employees EMP_a and EMP_{new} assigned to the defined Position POS_{k1} need to work on a highly critical project as soon as possible. In such a scenario administrators would likely assign permissions manually. This situation is subverting the goals of the existing role system. A further disadvantage is that employees who later on join this specific department, will also not be provided with the correct roles but rather gain their permissions manually. Similar to the previous use case the employees that already worked within the company in a different department still might have a large number of their old permissions due to the lack of correct de-provisioning processes.

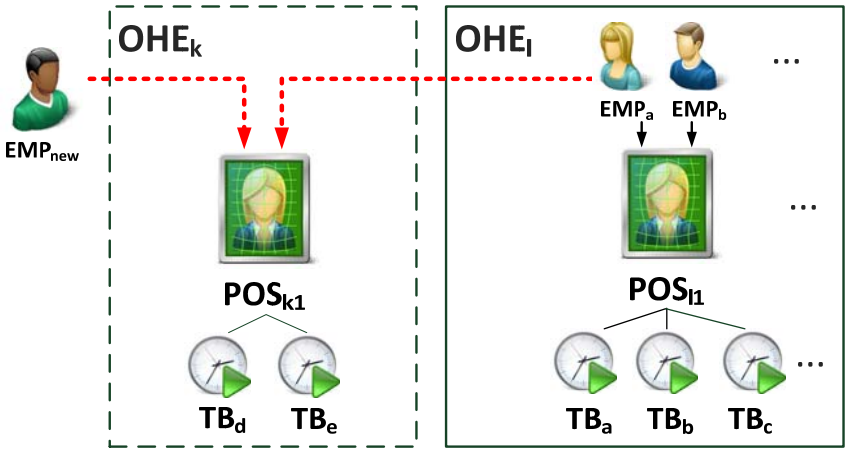

Figure 4: Creation of new hierarchical element

Detection and Actions:
To detect this use case and react appropriately any newly created or deleted hierarchical elements have to be detected at first. This can be done by an analysis of available organisational charts, a list of valid hierarchical elements, or user attributes regarding the assignment of employees to hierarchical structures. Consecutively the permissions of employees working in the newly created hierarchical elements have to be examined. Respective Functional- and Organisational Roles must be implemented and the role catalogue extended accordingly if they are not in place yet. After the creation of the various roles within the new hierarchical elements the permissions of employees have to be correctly provisioned. Directly assigned permissions have to be revoked and membership has to be granted using roles. Unused roles might have to be deleted.

4. checkROLE – A Tool for Automated Role Checking

As tool support for the automatic detection of the various use cases is mandatory we developed *checkROLE* an open-source application that is able to detect events affecting the role definitions. On this basis it provides responsible managers assistance in their decision process from *Figure 1*. Input information from various user repositories at a certain point of time *t* is copied to the Role Maintenance environment using a LDAP connector or a .csv-file import. Additionally, an image of the currently valid role catalogue is extracted from the Role Management System. CheckROLE provides interfaces to an Active Directory storing the role catalogue, synchronised identity information, as well as access rights. Moreover a connector to other external Role Mining tools gives us the ability to execute non-statistical analysis of user data.

4.1. Detecting Use Cases in CheckROLE

Figure 5 presents the main interface of checkROLE. Besides the detection scenarios (*Role Maintenance*) the *Import to LDAP* tab provides the ability to set up the data basis contained in the underlying repository. To support additional Role Mining tasks, an *Export to DB* tab was implemented providing export functionality of user and permission data. For real-life applicability we narrowed down the theoretically defined use cases according to their outcome to nine so called *detection scenarios* that checkROLE is able to identify. These detection scenarios occur in various combinations, dependent on the underlying use cases from section 3:

- $\{EMP_a, EMP_b, \ldots, EMP_n\}$ have directly assigned Permission Bundles $PB_a \in PB$
- $\{EMP_a, EMP_b, \ldots, EMP_n\}$ have a set of wrong or unused Organisational Roles $\{OR_a, OR_b, \ldots, OR_n\} \in ORG\_Roles$
- inconsistent assignment of Permission Bundle $PB_a \in PB$ to $FUN\_Role_a \in FUN\_Roles$
- inconsistent assignment of Functional Role $FUN\_Role_a \in FUN\_Roles$ to Organisational Role $ORG\_Role_a \in ORG\_Roles$
- detection of new Permission Bundle $PB_{new} \in PB$
- detection of changes in Position definitions
- detection of changes in Organisational Hierarchies

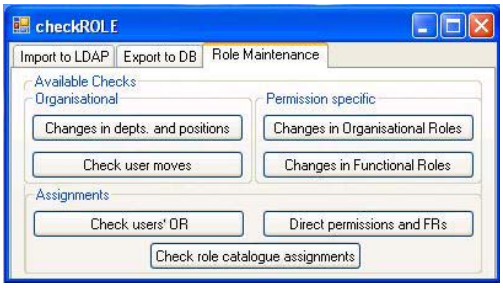

Figure 5: CheckROLE interface

4.2. Test Scenario

In order to demonstrate the application of checkROLE, we are going to present a short test scenario in a small industrial company with about 10 departments, 45 employees and 15 different business roles granting membership to about 50 different Active Directory groups (permissions). The company has implemented busiROLE as role model. For the role-checks user information and the existing role catalogue have been imported to the Role Maintenance environment using a LDAP connection to the global Active Directory within the company. Applying the *Check user moves* and the *Check users OR* functionality of checkROLE generates the output windows shown in *Figure* .

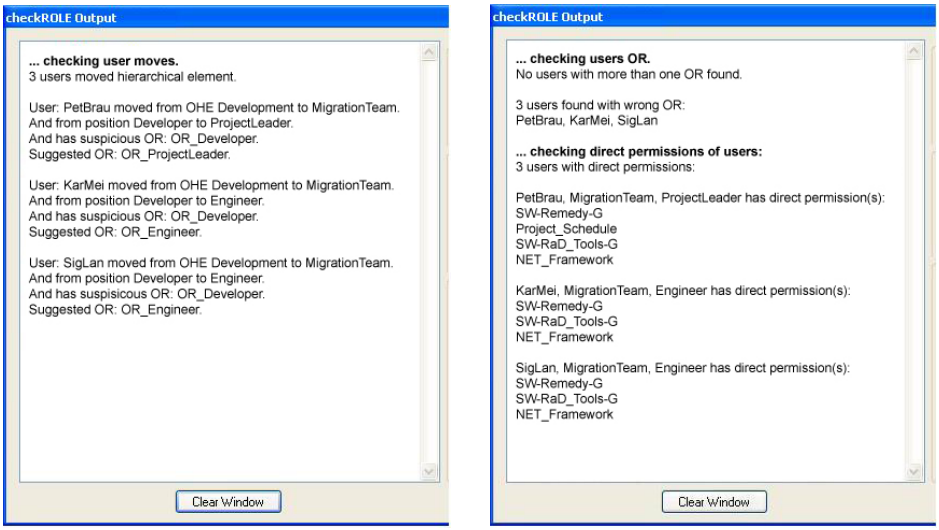

Figure 6: CheckROLE test scenario output

The window on the left lists the employees that have changed their hierarchical element(s) and/or Position since the last role-checking loop. Positions of employees are stored in the *title* attribute within the Active Directory of the company. One can see that three users previously working in the Development department have joined the MigrationTeam. Their title attribute has changed, e.g. from *Developer* to *ProjectLeader* in case of employee *PetBrau*. Security policies prevent users to have more than one Position and Organisational Role at a certain point of time. CheckROLE identified that *PetBrau* is still connected to his old Organisational Role (*OR_Developer*) and that his title attribute has a different entry not corresponding to the Organisational Role assigned. Investigating the valid roles in the department MigrationTeam *OR_ProjectLeader* is suggested as the correct role. This needs to be done on basis of cluster analysis of existing rights within the respective department, closely interwoven with Role Development issues. The output window on the right side of *Figure* reveals directly assigned permissions. Even though all the permissions imported to the Role Maintenance environment have to be granted on basis of role membership, checkROLE identified several direct user-permission assignments. Employee *PetBrau* is e.g. on the one hand still able to access the various resources connected to his old Organisational Role. To fulfil the tasks related to his new Position *ProjectLeader* he was on the other hand assigned to the necessary permissions *SW-Remedy-G, Project_Schedule, SW-RaD_Tools-G, and NET_Framework* directly. With this output responsible role managers now are able to take adequate measures to resolve the found issues. This short scenario has shown that checkROLE is able to support strategic Role Management and provide information that might not even have been identified at all.

5. Conclusions and Future Work

In this paper we have seen that companies require support in ensuring the timeliness of the implemented roles in their IT infrastructure. Several changes within an organisation might affect the role definitions and require the role catalogue to be updated. This requires an automated overview over possible changes to be integrated in a Role Maintenance solution. Up to now, to the best of our knowledge, no assistance is provided by researchers in this area. Consequently we defined a use case catalogue that acts as theoretical basis for role-checks. An in-depth analysis of single use cases has revealed their structure, impact on the role definitions, and possible detection mechanisms. We furthermore presented checkROLE a tool for automatic detection of various use cases. It facilitates statistical analysis and data mining algorithms to compare a valid role catalogue with the present situation within a company in order to identify discrepancies. A short test scenario

has shown the applicability of checkROLE within a small company. For future work we are going to add new functionalities for detecting combined use cases and migrate towards a process-oriented role checking workflow. Up to now checkROLE only provides a simple text-based presentation of results. For a seamless integration into future Role Maintenance and Role Development solutions it needs to be extended in order to allow for an automatic correction of detected anomalies. Moreover the text-based visualisation is very limited in terms of readability in more complex test scenarios. We hence are implementing an adequate solution that presents the found results in a more structured way allowing for interaction with the checkROLE users, i.e. the Role Maintenance team.

References

[1] BANK FOR INTERNATIONAL SETTLEMENTS, BIS: International Convergence of Capital Measurement and Capital Standards: A Revised Framework - Comprehensive Version, http://www.bis.org/publ/bcbs128.pdf, 2006.

[2] CRAMPTON, J., LOIZOU, G., Administrative scope: A foundation for role-based administrative models, ACM Transactions on Information and System Security (TISSEC) 6, pp. 201–231, 2003.

[3] DAFT, R., Organization Theory and Design. 2nd ed. West, St. Paul, Minn. 1986.

[4] DHILLON, G.: Violation of Safeguards by Trusted Personnel and Understanding Related Information Security Concerns. Computers & Security 20 (2), pp. 165-172, 2001.

[5] EUROPEAN UNION: Directive 95/46/EC of the European Parliament and of the Council. Official Journal of the European Communities L (28-31), http://ec.europa.eu/justice_home/fsj/privacy/docs/95-46-ce/dir1995-46_part1_en.pdf, 1995.

[6] FERRAIOLO, D. F., KUHN, R. D., CHANDRAMOULI, R., Role-Based Access Control. Artech House, Boston, Mass./London 2007.

[7] FUCHS, L., PREIS, A., BusiROLE: A Model for Integrating Business Roles into Identity Management, Proc of the 5th Int. Conference on Trust, Privacy, and Security in Digital Business (TrustBus), Torino, Italy 2008.

[8] FUCHS, L., PERNUL, G., Supporting Compliant and Secure User Handling – a Structured Approach for In-house Identity Management, Proc. of the 2nd Int. Conference on Availability, Reliability and Security (ARES '07), pp. 374-384. IEEE Computer Society, Vienna, Austria 2007.

[9] FUCHS, L., PERNUL, G., proROLE: A Process-oriented Lifecycle Model for Role Systems, Proc. of the 16th European Conference on Information Systems **(**ECIS), Galway, Ireland 2008.

[10] GALLAHER, M. P., O'CONNOR, A. C., KROPP, B.: The economic impact of role-based access control. Planning report 02-1, National Institute of Standards and Technology, http://www.nist.gov/director/prog-ofc/report02-1.pdf, Gaithersburg, MD 2002.

[11] KERN, A., KUHLMANN, M., SCHAAD, A., MOFFETT, J., Observations on the role life-cycle in the context of enterprise security management, Proc. of the 7th ACM Symp. on Access Control Models and Technologies (SACMAT '02), pp. 43-51, ACM, New York 2002.

[12] LARSSON, E. A.: A case study: Implementing Novell Identity Management at Drew University. Proc. of the 33rd ACM SIGUCCS conference on User services (SIGUCCS'05), pp. 165-170, ACM, New York 2005.

[13] MINTZBERG, H., Structuring of Organizations. Prentice Hall, Englewood Cliffs, N.J. 1979.

[14] NYANCHAMA, M., OSBORN, S., The role graph model and conflict of interest, ACM Transactions on Information and System Security (TISSEC) 2, pp. 3–33, 1999.

[15] OH, S., SANDHU, R., A model for role administration using organization structure, Proc. of the 7th ACM Symp. on Access Control Models and Technologies (SACMAT '02), pp. 155–162, ACM, New York 2002.

[16] OH, S., SANDHU, R., Zhang, X., An effective role administration model using organization structure, ACM Transactions on Information and System Security (TISSEC) 9, pp. 113–137, 2006.

[17] SANDHU, R., BHAMIDIPATI, V., MUNAVER, Q., The ARBAC97model for role-based administration of roles, ACM Transactions on Information and System Security (TISSEC) 2, pp. 105–135, 1999.

[18] SANDHU, R., MUNAVER, Q., The ARBAC99 Model for Administration of Roles, Proc. of the 15th Annual Computer Security Applications Conference, pp. 229–238, Phoenix, USA 1999.

[19] SARBANES, P. S., OXLEY, M.: Sarbanes-Oxley Act of 2002, also known as the "Public Company Accounting Reform and Investor Protection Act of 2002", http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_bills&docid=f:h3763enr.tst.pdf, 2002.

[20] SCHIMPF, G., Role-Engineering Critical Success Factors for Enterprise Security Administration, Position Paper for the 16th Annual Computer Security Application Conference, New Orleans, USA 2000.

[21] ZHANG, Y., JOSHI, J. B., ARBAC07: A Role-based Administration Model for RBAC with Hybrid Hierarchy, Proc. of the IEEE Int. Conference on Information Reuse and Integration, pp. 196–202, Las Vegas, USA 2007.