"The final publication is available at springerlink.com"
In Proceedings of the 14th International Conference on Computer Aided Systems Theory (EUROCAST), Springer, LNCS 8111, pp. 507-514(2013)

Delta Analysis of Role-Based Access Control Models

Maria Leitner

University of Vienna, Austria
Faculty of Computer Science
maria.leitner@univie.ac.at

Abstract. Role-based Access Control (RBAC) is de facto standard for access control in Process-aware Information Systems (PAIS); it grants authorization to users based on roles (i.e. sets of permissions). So far, research has centered on the design and run time aspects of RBAC. An evaluation and verification of a RBAC system (e.g., to evaluate ex post which users acting in which roles were authorized to execute permissions) is still missing. In this paper, we propose delta analysis of RBAC models which compares a prescriptive RBAC model (i.e. how users are expected to work) with a RBAC model (i.e. how users have actually worked) derived from event logs. To do that, we transform RBAC models to graphs and analyze them for structural similarities and differences. Differences can indicate security violations such as unauthorized access. For future work, we plan to investigate semantic differences between RBAC models.

Keywords: Access Control, Delta Analysis, Organizational Mining, RBAC, Security

1 Introduction

Process-aware Information Systems (PAIS) support the automated execution of tasks in business processes (cf. [23]). Authorization and access control are key challenges when it comes to security in PAIS (cf. [4, 14]). Role-based access control (RBAC) models (e.g., [11]) are the de facto standard for access control in PAIS. RBAC uses the concept of roles to restrict access; a role consists of a set of permissions, i.e. authorizations to do certain actions such as executing tasks in a business process. For example, only users having the role *Doctor* are allowed to execute task *retrievePatientRecords* to get patient records.

Furthermore, process mining techniques extract and examine process-related information from (process) event logs [1]. Process mining can be used for delta analysis by comparing the discovered process model (i.e. the actual process represented by a process model obtained through process mining) with a prescriptive process model [2] (i.e. how the process model is expected to work). Furthermore, organizational mining techniques extract and derive organizational structures with organizational models (e.g., [19]). These techniques can be suitable to derive RBAC models from event logs (further called *current-state* RBAC models)

[15]. Current-state RBAC models reflect the operational reality; they provide information on which users acting in which roles have actually executed which tasks. Hence, these access snapshots can be used for analysis and evaluation.

This paper investigates delta analysis of RBAC models which compares a prescriptive RBAC model with a current-state RBAC model and analyzes and evaluates the models for structural similarities and differences. Therefore, we transform RBAC models into labeled graphs (cmp. [13]) and compare them with e.g., the graph edit distance (cf. [12, 9]). With delta analysis, we hope to discover differences between the prescriptive and the current-state RBAC models. These deviations can indicate e.g., security and compliance violations or an outdated configuration of the RBAC model. Furthermore, a case study shows that the evaluation of a snapshot of RBAC models can be complex and can only be performed by domain experts.

The remainder of this paper is structured as follows. Section 2 outlines delta analysis of RBAC models by comparing the structure of RBAC models. Furthermore, structural differences of RBAC models are evaluated in a case study in Section 3. Section 4 reviews related work and Section 5 concludes the paper.

2 Delta Analysis of Role-based Access Control Models

Delta analysis aims to discover differences between descriptive/prescriptive and discovered (current-state) RBAC models as shown in Figure 1. Current-state RBAC models contain which users in which roles have actually invoked which permissions i.e. reflect operational reality. Hence, these access snapshots can be used for an ex post analysis and evaluation of RBAC implementations. As shown in Figure 1, delta analysis uses this operational knowledge and compares the existing original, conceptual (prescriptive) models with reality to detect violations such as unauthorized access.

Delta analysis of RBAC models contains of three steps: (1) obtain prescriptive and discovered RBAC models, (2) compare RBAC models for e.g., structural similarities and differences and (3) analyze and evaluate differences to detect violations. In the following sections, we will define basic concepts and outline the structural matching of RBAC models.

2.1 Preliminaries

In this paper, we specify an RBAC model based on the NIST standard RBAC model defined in [11] and its administrative operations (further called edit operations). Specifically, our approach uses the following edit operations of [11]: addUser, deleteUser, addRole, deleteRole, addPermission, deletePermission, assignUser, deassignUser, grantPermission, revokePermission, addInheritance and deleteInheritance. A RBAC model contains a set users, roles and permissions ($PRMS$) and three relations exist: users can be assigned to roles (UA), permissions can be assigned to roles (PA) and roles can be associated with roles (RH) i.e. an inheritance relation exists between roles. To compare the structure

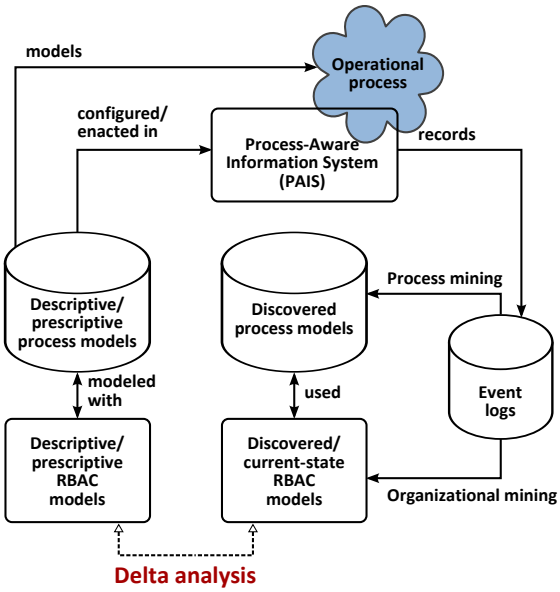

Fig. 1. Delta Analysis Overview (adapted from [2])

of RBAC models we transform RBAC models into labeled graphs. Therefore, we will give a definition of a directed acyclic graph (DAG) and describe the graph-transformed RBAC model.

An labeled DAG is denoted as $G = (N, E, \alpha, \beta)$ where N is a set of nodes, E ($E \subseteq N \times N$) is a set of edges, α is a node labeling function; $\alpha : N \rightarrow L_N$, and β is a labeling function for edges: $\beta : E \rightarrow L_E$. Let $(n, m) \in E$ be an edge, (n, m) is incident from node n and enters node m. The number of edges entering a node is called in-degree (id), and the out-degree (od) signifies the number of edges leaving a node. There exist no isolated nodes in G; $\forall n \in N : id(n) + od(n) > 0$. The label representation $\rho(G)$ of G is given by $\rho(G) = \{L, C, \lambda\}$ where $L = \{\alpha(n) | n \in N\}$ and $C = \{(\alpha(n), \alpha(m)) | (n, m) \in E\}$ and $\lambda : C \rightarrow L_E$ with $\lambda(\alpha(n), \alpha(m)) = \beta(n, m)$ for all $(n, m) \in E$.

Let a graph-transformed RBAC model be $G = (N, E, \alpha, \beta)$. Then, $N = \{USERS \cup ROLES \cup PRMS\}$ is a set of nodes and $E = \{UA \cup PA \cup RH\}$ is a set of edges. The labeling functions are identically defined as in the DAG.

2.2 Structural Matching of RBAC Models

We can identify three (use) cases of the comparison of RBAC models. In the first case, the current-state RBAC model equals the predictive RBAC model as shown in Figure 2 (A) and (B). This case is probably the rarest case. It seems likely that this case may only happen if a predictive model has been newly

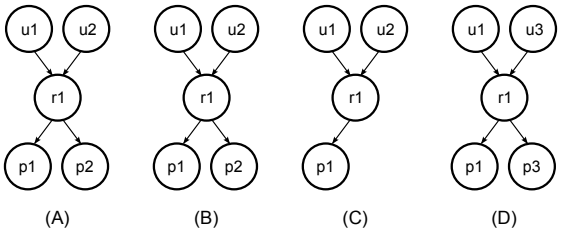

Fig. 2. Examples for a (A) Prescriptive RBAC Model and Current-State RBAC Models (B), (C) and (D)

implemented or recently adapted. Delta analysis can be used to verify the latest RBAC implementation.

Furthermore, the current-state RBAC model is a subgraph of the predictive model in the second case (cmp. Figure 2 (A) and (C)). Current-state RBAC models reflect the operational reality; they do not reflect a full state i.e. certain users, roles, and permissions are not included because they are not listed in event logs. For example, certain permissions are defined in RBAC models such as that a president has permission to launch nuclear weapons. However, the president might have the permission but he or she might not exercise it often. Hence, the current-state model is still valid and certain permissions are not included.

In the third case, the current-state RBAC model differs from the predictive RBAC model as shown in Figure 2 (A) and (D) e.g., new users, permissions or roles are included. For example, the role structure in RBAC depends on the applied technique such as role mining (e.g., finding a minimal descriptive set of roles [20]) or organizational mining (cmp., [15]). In order to minimize deviations in role structure, we recommend to use the same mining algorithm for the current-state RBAC model as the predictive model.

Based on these cases, we evaluated inexact graph matching techniques (e.g., [6, 8]). We assume that RBAC models use unique labels i.e. users, roles and permissions have unique IDs. Given this structural requirement and based on an extensive literature review, graph matching techniques for graphs with unique node labels as specified in [9] are the most suitable for this domain. In fact, the problems graph isomorphism, subgraph isomorphism, the maximum common subgraph (cf. [7]) and the graph edit distance as specified in [9] can cover all three cases:

Let a graph-transformed RBAC model be $P = \{N_1, E_1, \alpha_1, \beta_1\}$, another graph-transformed RBAC model be $S = \{N_2, E_2, \alpha_2, \beta_2\}$ and their label representations be $\rho(P) = \{L_1, C_1, \lambda_1\}$ and $\rho(S) = \{L_2, C_2, \lambda_2\}$.

- Graph isomorphism between P and S is a bijective mapping $f : N_1 \to N_2$ such that $\alpha_1(n) = \alpha_2(f(n)), \forall n \in P$ and $\beta_1(n, m) = \beta_2(f(n), f(m)), \forall (n, m) \in E_1$. Graph S is isomorphic to graph P if $\rho(P) = \rho(S)$ i.e. $L_1 = L_2$, $C_1 = C_2$ and $\lambda_1 = \lambda_2$.

- Subgraph isomorphism between P and S is an injective mapping $f : N_1 \rightarrow N_2$ if there exists a subgraph $S \subseteq P$. Graph S is subgraph isomorphic to graph P if $L_2 \subseteq L_1$, $C_2 \subseteq C_1$ and $\lambda_2 \subseteq \lambda_1$.
- Let G be a graph with $\rho(G) = \{L, C, \lambda\}$ such that $L = L_1 \cap L_2$, $C = \{(n,m)|(n,m) \in C_1 \cap C_2\}$ and $\lambda_1(n,m) = \lambda_2(n,m)$ and $\lambda(n,m) = \lambda_1(n,m)$ for all $(n,m) \in C$. Then, the maximum common subgraph of P and S is G.
- The graph edit distance $d(P,S)$ measures the minimal number of graph edit operations necessary to transform one graph P into another graph S. Please note that we consider the RBAC edit operations outlined in Section 2.1 as graph edit operations. The graph edit distance between P and S is $d(P,S) = |L_1| + |L_2| - 2\,|L_1 \cap L_2| + |C_1| + |C_2| - 2\,|C_0| + \left|C_0'\right|$ where $C_0 = \{(n,m)|(n,m) \in C_1 \cap C_2\}$ where $\lambda_1(n,m) = \lambda_2(n,m)$ and $C_0' = \{(n,m)|(n,m) \in C_1 \cap C_2\}$ where $\lambda_1(n,m) \neq \lambda_2(n,m)$.

All problems are deployed and tested in a prototypical implementation. Using unique node labels, the computational complexity for all problems is $O(n^2)$.

3 Case Study

Figure 3 displays a graph representation of a (A) predictive and a (B) current-state RBAC model. As can be seen from the figures, the predictive RBAC model differs from the current-state model. The distance of both models is measured by the graph edit distance ($d(A,B) = 14$). The edit operations necessary to transform model (A) to (B) are shown in Figure 3.

Differences in the structure of the two RBAC models in Figure 3 can indicate violations. For example, user `u7` is included in the predictive model (A) but is not shown in the current-state model (B). This could signify that `u7` was not active during that time (e.g., on vacation) or that he or she is not a user any more (e.g., retired). However, it seems that `u5` who was assigned to role `r3` in (A) has `r4` in (B). This could indicate that `u5` changed roles (e.g., promotion) or that he or she violated permissions (e.g., acquired unauthorized access to access rights). In the case of `u5`, delegations of roles in RBAC or tasks in business processes can cause these deviations.

Furthermore, the models in Figure 3 differ in the role hierarchy. For example, role `r6` is created in (B) and inherits permissions of `r3` and `r2`. Interestingly, the inclusion of `r6` adds only an additional role layer but does not change the semantics (e.g., user `u6` can still perform the same set of permissions). In (A), `u1` is assigned to the roles `r1` and `r3`. These assignments are not included in (B) as `r1` inherits permissions of `r3` (e.g., reduced by the mining algorithm).

It can be seen from the case study that the evaluation of structural differences is complex. Deviations can be caused of the underlying mining technique (e.g., additional role `r6`), security violations (e.g., `u5`), inactivity or absence (e.g., `u7`) and misconfiguration (e.g., outdated assignments). Given these multifaceted causes, the evaluation has to be performed or monitored by experts with domain knowledge. Due to the large size of RBAC systems with thousands of roles (cf.

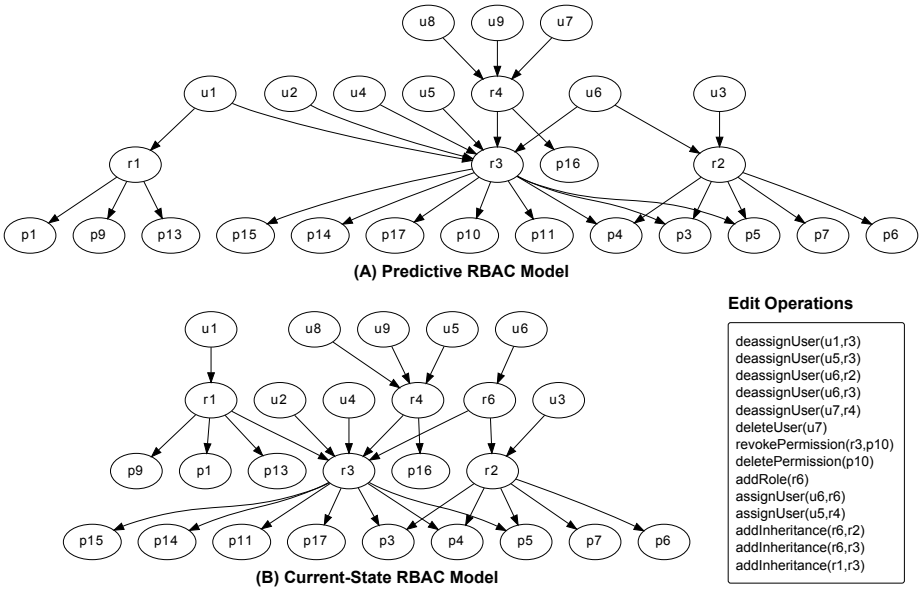

Fig. 3. Graph Representation of a Graph-based (a) Prescriptive RBAC Model and a (b) Current-State RBAC Model

[18]) an automated evaluation is preferred and can be cumbersome. By comparing a section of a RBAC model or by analyzing the RBAC model of a certain business process can reduce the size of the models.

4 Related Work

In recent years, many extensions of the NIST RBAC model [11] have been proposed to include aspects of PAIS. For example, the W-RBAC model [21] extends the NIST model for cases and organizational units. Further examples integrate process changes or structural and operational aspects in an RBAC model (e.g., [22, 17, 16]).

A graph-based formalism in [13] specifies static and dynamic consistency conditions in graphs. Moreover, graph optimization for role engineering is shown in (e.g., [24]). In this paper, we use graphs to compare the structure of RBAC models. A comparison of RBAC models proposed in [5] aims to migrate an existing RBAC model into a desired model (a designated state). The goal of delta analysis is to compare both RBAC models to analyze differences to detect security violations.

In business processes, current research provides similarity metrics for e.g., structural or label matching of processes (e.g., [10]). However, in this paper we center on RBAC models which have different structural requirements such

as role hierarchies or unique node labels. The suitability of process mining for security audits such as evaluating authorization constraints is shown in [3]. In fact, the evaluation is performed manually and with respect to roles. As the event logs provide only user and task information, the analysis is cumbersome and an organizational model is needed. Delta analysis already considers the organizational model (e.g., roles) and enables an automated evaluation.

5 Conclusion

This paper described delta analysis of RBAC models which compares a prescriptive RBAC model with a current-state RBAC model. This approach aims to identify security violations such as unauthorized access. In this paper, we transform RBAC models into labeled graphs and analyze and compare the structure of RBAC models to identify differences which can indicate security violations. Furthermore, we show in a case study that the comparison of RBAC models can be complex and can only be performed by domain experts. For future work, we plan to include and examine semantic distance measures to compare RBAC models. As structure of RBAC models is an aspect, we want to examine semantic differences of RBAC models such as the impact on users having more or less access.

References

1. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer (2011)
2. Aalst, W.M.P.v.d.: Business alignment: using process mining as a tool for delta analysis and conformance testing. Requirements Engineering 10(3), 198–211 (Aug 2005)
3. Accorsi, R., Stocker, T.: On the exploitation of process mining for security audits: the conformance checking case. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. p. 1709–1716. SAC '12, ACM, New York (2012)
4. Atluri, V., Warner, J.: Security for workflow systems. Handbook of Database Security p. 213–230 (2008)
5. Baumgrass, A., Strembeck, M.: An approach to bridge the gap between role mining and role engineering via migration guides. In: 2012 Seventh International Conference on Availability, Reliability and Security (ARES). pp. 113–122. IEEE (2012)
6. Bunke, H., Allermann, G.: Inexact graph matching for structural pattern recognition. Pattern Recognition Letters 1(4), 245–253 (May 1983)
7. Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph. Pattern Recognition Letters 19(3–4), 255–259 (Mar 1998)
8. Conte, D., Foggia, P., Sansone, C., Vento, M.: THIRTY YEARS OF GRAPH MATCHING IN PATTERN RECOGNITION. International Journal of Pattern Recognition and Artificial Intelligence 18(03), 265–298 (May 2004)
9. Dickinson, P.J., Bunke, H., Dadej, A., Kraetzl, M.: Matching graphs with unique node labels. Pattern Analysis and Applications 7(3), 243–254 (Dec 2004)
10. Dijkman, R., Dumas, M., van Dongen, B., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Information Systems 36(2), 498–516 (Apr 2011)

11. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R.: Proposed NIST standard for role-based access control. ACM Trans. Inf. Syst. Secur. 4(3), 224–274 (2001)
12. Gao, X., Xiao, B., Tao, D., Li, X.: A survey of graph edit distance. Pattern Analysis and Applications 13(1), 113–129 (Feb 2010)
13. Koch, M., Mancini, L., Parisi-Presicce, F.: A formal model for role-based access control using graph transformation. In: Cuppens, F., Deswarte, Y., Gollmann, D., Waidner, M. (eds.) Computer Security - ESORICS 2000, Lecture Notes in Computer Science, vol. 1895, pp. 122–139. Springer Berlin Heidelberg (2000)
14. Leitner, M.: Security policies in adaptive process-aware information systems: Existing approaches and challenges. In: 2011 Sixth International Conference on Availability, Reliability and Security (ARES). pp. 686–691. IEEE (2011)
15. Leitner, M., Baumgrass, A., Schefer-Wenzl, S., Rinderle-Ma, S., Strembeck, M.: A case study on the suitability of process mining to produce current-state RBAC models. In: Rosa, M., Soffer, P. (eds.) Business Process Management Workshops. Lecture Notes in Business Information Processing, vol. 132, pp. 719–724. Springer Berlin Heidelberg (2012)
16. Leitner, M., Mangler, J., Rinderle-Ma, S.: SPRINT-Responsibilities: design and development of security policies in process-aware information systems. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA) 2(4), 4–26 (2011)
17. Leitner, M., Rinderle-Ma, S., Mangler, J.: AW-RBAC: access control in adaptive workflow systems. In: 2011 Sixth International Conference on Availability, Reliability and Security (ARES). pp. 27–34. IEEE (2011)
18. Schaad, A., Moffett, J., Jacob, J.: The role-based access control system of a european bank: a case study and discussion. In: Proceedings of the sixth ACM symposium on Access control models and technologies. p. 3–9. SACMAT '01, ACM, New York (2001)
19. Song, M., van der Aalst, W.M.: Towards comprehensive support for organizational mining. Decision Support Systems 46(1), 300–317 (2008)
20. Vaidya, J., Atluri, V., Guo, Q.: The role mining problem: finding a minimal descriptive set of roles. In: Proceedings of the 12th ACM symposium on Access control models and technologies. p. 175–184. SACMAT '07, ACM, New York (2007)
21. Wainer, J., Barthelmess, P., Kumar, A.: W-RBAC - a workflow security model incorporating controlled overriding of constraints. International Journal of Cooperative Information Systems 12(4), 455–485 (2003)
22. Weber, B., Reichert, M., Wild, W., Rinderle, S.: Balancing flexibility and security in adaptive process management systems. In: Meersman, R., Tari, Z. (eds.) On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, Lecture Notes in Computer Science, vol. 3760, pp. 59–76. Springer Berlin Heidelberg (2005)
23. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer (2007)
24. Zhang, D., Ramamohanarao, K., Ebringer, T.: Role engineering using graph optimisation. In: Proceedings of the 12th ACM symposium on Access control models and technologies. p. 139–144. SACMAT '07, ACM, New York (2007)