Evaluating the Risk of Adopting RBAC Roles

Alessandro Colantonio, Roberto Pietro, Alberto Ocello, Nino Vincenzo Verde

Evaluating the Risk of Adopting RBAC Roles

Alessandro Colantonio[1,2], Roberto Di Pietro[2], Alberto Ocello[1], and Nino Vincenzo Verde[2]

[1] Engiweb Security, Roma, Italy,
{alessandro.colantonio, alberto.ocello}@eng.it
[2] Università di Roma Tre, Roma, Italy,
{colanton, dipietro, nverde}@mat.uniroma3.it

Abstract. We propose a framework to evaluate the risk incurred when managing users and permissions through RBAC. The risk analysis framework does not require roles to be defined, thus making it applicable before the role engineering phase. In particular, the proposed approach highlights users and permissions that markedly deviate from others, and that might consequently be prone to error when roles are operating. By focusing on such users and permissions during the role definition process, it is possible to mitigate the risk of unauthorized accesses and role misuse.

1 Introduction

Access Control is a cornerstone of enterprise risk and security management. It represents the process of mediating requests to data and services maintained by a system, and determining whether the requests should be granted or denied [1]. Among all access control models proposed in the literature, *Role-Based Access Control* (RBAC) attracted the attention of many large-size organizations. According to this model, *roles* are created for various job functions, and permissions required to perform certain operations are assigned to specific roles. By deploying RBAC systems, companies obtain several benefits such as simplified access control administration, improved organizational productivity, and security policy enforcement. The definition of roles is one of the important issues that need to be more deeply addressed in order to increase interest toward RBAC. Role engineering [2,3] approaches can usually be classified in *top-down* and *bottom-up*. In the top-down case, a deep manual analysis is required to formulate roles that match the skills, tasks or duties of users. In the bottom-up case, also indicated with the term *role mining*, roles are automatically elicited from existing permission assignments. Several algorithms and models explicitly designed for role mining were proposed—see [4] for a brief survey on the subject. Once a role is created, its lifecycle will follow the evolution of the company: new users or new permissions can be added, old ones can be removed or replaced, users can change their job position, etc. This continuous modification of the access control system typically introduce “noise” within the data—namely, permissions exceptionally or accidentally granted or denied—, thus increasing the risk of making mistakes when managing the access control system. Moreover, it is important to create roles that administrators can easily understand and manage [5].

In this paper, we introduce a risk analysis framework that allows to evaluate the risk incurred when managing users and permissions through RBAC. A distinguishing feature of our approach is that it can be used without having already defined roles, namely in a pre-engineering phase. By evaluating the risk level of a single user or a single permission, we make it possible to produce a ranking of users and permissions, highlighting those that most deviate from others in comparison to available user-permission relationships. Consequently, we are able to identify those users and permissions that represent the most (likely) dangerous and error prone ones from an administration point of view. Having this ranking available during the role engineering phase allows data analysts and role engineers to highlight users and permissions that are more prone to error and misuse when designed roles will be operating.

The the paper is organized as follows: In Section 2 a formal modeling of the problem is provided. Section 3 introduces our risk framework, mapping typical risk-related concepts to RBAC entities. Section 4 shows the results of a test on real data, while concluding remarks are provided in Section 5.

2 Problem Modeling

In this paper we use some concepts that were introduced in [6], where a three steps methodology to identify "stable" roles is thoroughly described. First, a weight is associated to roles; second, user-permission assignments that cannot belong to roles with a weight exceeding a given threshold are identified; and third, the role-finding problem is restricted to user-permission assignments identified in the second step. Differently from [6], this paper proposes a model that is not strictly bound with the role discovery within existing user-permission relationships, but it can be adopted in a pre-mining phase in order to evaluate the risk incurred to manage users and permissions.

Before introducing the formalism required to describe our risk evaluation model, we first review some concepts of the ANSI/INCITS RBAC standard [7] needed for the present analysis. Entities of interest are: PERMS, USERS, and ROLES, that is the set of all access permissions, users, and roles, respectively; $\mathit{UA} \subseteq \mathit{USERS} \times \mathit{ROLES}$, the set of all role-user relationships; and, $\mathit{PA} \subseteq \mathit{PERMS} \times \mathit{ROLES}$, the set of role-permission relationships. A RBAC *state* is a tuple $\langle \mathit{ROLES}, \mathit{UA}, \mathit{PA} \rangle$, namely an instance of all the sets characterizing RBAC that is used to obtain a system configuration. We do not consider sessions, role hierarchies or separation of duties constraints in this paper, but in addition to RBAC concepts, we introduce: $\mathit{UP} \subseteq \mathit{USERS} \times \mathit{PERMS}$ to indicate the set of the existing user-permission assignments to analyze; $\mathit{perms}\colon \mathit{USERS} \to 2^{\mathit{PERMS}}$ to identify permissions assigned to a user, namely $\mathit{perms}(u) := \{p \in \mathit{PERMS} \mid \langle u, p \rangle \in \mathit{UP}\}$; and, $\mathit{users}\colon \mathit{PERMS} \to 2^{\mathit{USERS}}$ to identify users assigned with a permission, namely $\mathit{users}(p) := \{u \in \mathit{USERS} \mid \langle u, p \rangle \in \mathit{UP}\}$. Moreover, we define:

Definition 1 (Role Weight). *Given a role $r \in \mathit{ROLES}$, let U_r and P_r be the sets of users and permissions associated to r, that is $U_r := \{u \in \mathit{USERS} \mid \langle u, r\rangle \in \mathit{UA}\}$ and $P_r := \{p \in \mathit{PERMS} \mid \langle p, r\rangle \in \mathit{PA}\}$. We indicate with $w\colon \mathit{ROLES} \to \mathbb{N}$ the* weight *function of roles, defined as $w(r) := |U_r| \times |P_r|$.*

Definition 2 (t-stability). *Let Σ_{UP} be the set of all RBAC states that cover the user-permission assignments of UP, that is all $\langle \mathit{ROLES}, \mathit{UA}, \mathit{PA}\rangle \in \Sigma_{\mathit{UP}}$ such that $\forall \langle u, p\rangle \in \mathit{UP} \implies \exists r \in \mathit{ROLES} : \langle u, r\rangle \in \mathit{UA}, \langle p, r\rangle \in \mathit{PA}$. Given $\langle u, p\rangle \in \mathit{UP}$, let $\mathcal{R}\colon \mathit{UP} \to 2^{(\bigcup_{\langle \mathit{ROLES}, \mathit{UA}, \mathit{PA}\rangle \in \Sigma_{\mathit{UP}}} \mathit{ROLES})}$ be the function that identifies the roles which could be used to manage $\langle u, p\rangle$, that is:*

$$\mathcal{R}(\langle u, p\rangle) := \bigcup\nolimits_{\langle \mathit{ROLES}, \mathit{UA}, \mathit{PA}\rangle \in \Sigma_{\mathit{UP}}} \{r \in \mathit{ROLES} \mid \langle u, r\rangle \in \mathit{UA},\ \langle p, r\rangle \in \mathit{PA}\}\ .$$

We say that $\langle u, p\rangle$ is t-stable *if it can be managed with at least one role r with weight $w(r) \geq t$, namely $\exists r \in \mathcal{R}(\langle u, p\rangle) : w(r) \geq t$.*

If an assignment $\langle u, p\rangle \in \mathit{UP}$ is t-stable, it is also $(t - i)$-stable for each $i = 1, \ldots, t$. We are thus interested in the maximal stability of a given assignment, namely the maximum t that verifies the t-stability condition:

Definition 3 (Maximal Stability). *The* maximal stability *of an assignment $\langle u, p\rangle \in \mathit{UP}$ is the maximum t such that the assignment is t-stable. It is identified by the function $t^*\colon \mathit{UP} \to \mathbb{N}$ such that $t^*(\langle u, p\rangle) := \max_{r \in \mathcal{R}(\langle u, p\rangle)} w(r)$.*

The rational behind the introduction of the *stability* concept is that if an assignment can only be managed by roles with a limited weight, it represents an outlier. Indeed, only few users and permissions are involved in a role together with that assignment. System administrators are willing to manage roles with high weights—that is, which involve many users and many permissions—for several reasons. First, the benefits of using RBAC increase because there are fewer user-role and role-permission relationships to manage. Second, these roles represent relevant portions of the whole access control system of the company. Because of this relevance, they have a greater meaning for system administrators. Conversely, when an assignment cannot be managed with a high-weight role, it represents a portion of data which appears to be inconsistent with the remainder of that dataset. It might not be an error, but from the system administrator point of view, it is riskier than others. In other words, the risk of making mistakes when managing roles with a limited weight is higher: they are roles that are not used frequently, and are in some way obscure to administrators. We now introduce another function that will be used to evaluate the risk incurred when managing a single assignment:

Definition 4. *The function $\mathcal{N}\colon \mathit{UP} \to 2^{\mathit{UP}}$ indicates the assignments of UP which can be managed together with the given assignment, namely:*

$$\mathcal{N}(\langle u, p\rangle) := \left\{\langle u', p'\rangle \in \mathit{UP} \mid \langle u, p'\rangle, \langle u', p\rangle \in \mathit{UP},\ \langle u, p\rangle \neq \langle u', p'\rangle\right\}\ .$$

The following Lemma relates $\mathcal{N}(\langle u, p\rangle)$ with the t-stability concept:

Lemma 1. *Given an assignment $\omega := \langle u, p \rangle \in UP$, then $|\mathcal{N}(\omega)|$ is an upper bound for $t^*(\omega)$.*

Proof. By definition, all the single assignments that could be managed in the same role together with $\langle u, p \rangle$ belongs to $\mathcal{N}(\omega)$. Hence, a role that contains more than $|\mathcal{N}(\omega)|$ assignments cannot exist, concluding the proof. □

In [6] we proposed a practical approach to calculate $|\mathcal{N}(\omega)|$ for each $\omega \in UP$. We described two algorithms: a deterministic algorithm that is able to calculate the exact value for $|\mathcal{N}(\omega)|$ in $\mathcal{O}(|UP|^2)$ time, while a randomized algorithm offers an ε-approximated result with a computational complexity of $\mathcal{O}(k\,|UP|)$, where k is a parameter that can be arbitrarily chosen. Therefore, the joint usage of Lemma 1 and the algorithms described in [6] makes it possible to practically find an upper-bound for the maximal stability of each assignment belonging to UP. In the next section, we will show how to leverage this information to assign a risk level to a particular user or a particular permission.

3 The Risk Model

A typical risk management approach is made up of two parts: *risk analysis* (or *assessment*) and *risk control*. During risk analysis we identify potential risks and assess probabilities of negative events together with their consequences. With risk control we establish the tolerable level of risk for the organization, hence providing controls for failure prevention as well as actions to reduce the likelihood of a negative event. In a general risk management approach, aspects that need to be considered are: *vulnerabilities*, *threats*, and *risks*. When analyzing an access control system, possible vulnerabilities are represented by users that seem to behave differently from the majority of the users; threats are represented by errors and wrong administration actions that grant wrong permissions to users; and, risks correspond to either allowing users to execute operations that are not permitted, or to hamper their tasks by not granting required permissions.

Our target is to produce a *ranking* of users and permissions that is based on the risk of making mistakes when managing them in RBAC. By having this ranking available during a role engineering phase, it is possible to highlight users and permissions that are more prone to errors and misuse when roles are operating. At the same time, it is possible to check, and subsequently investigate the reasons why highlighted users and permissions behave as outliers. For instance, we can ask the manager of those users to confirm that the relative permissions have been granted correctly, or if there have been errors due to the omission of some internal rule. To evaluate such risks, in this paper we adopt a general risk formula that involves multiple factors with different probabilities [8], namely:

$$\mathit{Risk} := \sum_{i=1}^{n} P_i \times C_i \quad , \tag{1}$$

where P_i denotes the probability of each risk factor i, and C_i quantifies the consequences of these risk factors.

In our model, risk factors are used to evaluate the degree of importance of users and permissions. Indeed, there could be users in charge of activities that are critical for the main business of the organization, while there could be other users with a marginal importance for the business. In the same way, there could be critical permissions, and others permissions that have marginal importance only. In general, we need to assign various degree of importance to each risk factor by taking the consequence of its execution into consideration. This process requires a thorough analysis of the organization. We assume that the impact evaluation is provided by experts.

We will use the t-stability concept to give to each user-permission assignment a probability of occurrence for each risk factor. In particular, given the assignment $\omega = \langle u, p\rangle \in UP$, we define the risk probability of ω as:

Definition 5 (Risk Probability of an Assignment). *Given an assignment $\langle u, p\rangle \in UP$, the risk probability of $\langle u, p\rangle$ is a function $\mathsf{ass\_risk} \colon UP \to [0,1]$ such that:*

$$\mathit{ass\_risk}(\langle u, p\rangle) := 1 - \frac{t^*(\langle u, p\rangle)}{|UP|} \quad .$$

The rationale behind the risk probability function is the following: The more $t^*(\langle u, p\rangle)$ is close to $|UP|$, the more the risk level of the assignment ω is close to 0. Indeed, if an assignment can be managed by a single role that covers almost all assignments in UP, the user-permission assignment reflects a permission granted to the majority of the users in the dataset. Note that we are not assuming the presence of such a role among those used in the RBAC configuration, but we are only saying that such a role can exist. This consideration allows us to use our risk model in a pre-mining phase, when roles have not yet been decided on.

According to Lemma 1, we can quickly estimate an upper bound for $t^*(\langle u, p\rangle)$, and therefore a lower bound for the risk function:

Lemma 2 (Lower-Bound for the Risk Probability of an Assignment). *Given an assignment $\langle u, p\rangle \in UP$, then*

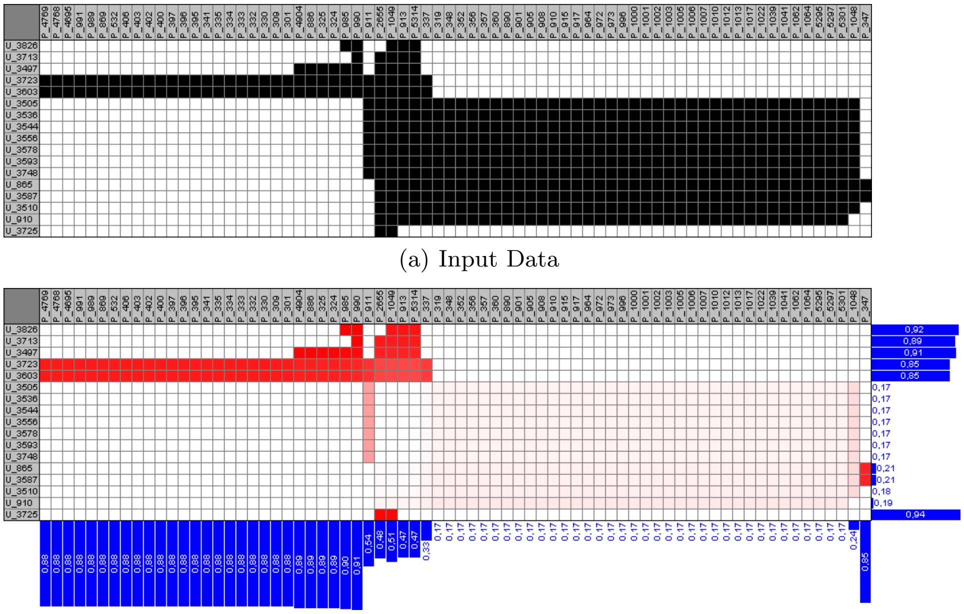
$$\mathit{ass\_risk}(\langle u, p\rangle) \le 1 - \frac{|\mathcal{N}(\langle u, p\rangle)|}{|UP|} \quad .$$

Proof. The proof immediately follows from Definition 5 and Lemma 1. □

By leveraging the above concepts, we can evaluate the risk probability for users and permissions in the following way:

Definition 6 (Risk Probability of a User). *Given an user $u \in USERS$, the risk probability of u is a function $\mathsf{user\_risk} \colon USERS \to [0,1]$ defined as:*

$$\mathit{user\_risk}(u) := \sqrt{\frac{\sum_{p \in \mathit{perms}(u)} \mathit{ass\_risk}^2(\langle u, p\rangle)}{|\mathit{perms}(u)|}} \quad . \qquad (2)$$



(a) Input Data

(b) Risk probabilities

Fig. 1. Risk probability of users and permissions in *UP*

Definition 7 (Risk Probability of a Permission). *Given a permission* $p \in$ *PERMS, the risk probability of* p *is a function* $\mathit{perm\_risk}\colon \mathit{PERMS} \to [0,1]$ *defined as:*

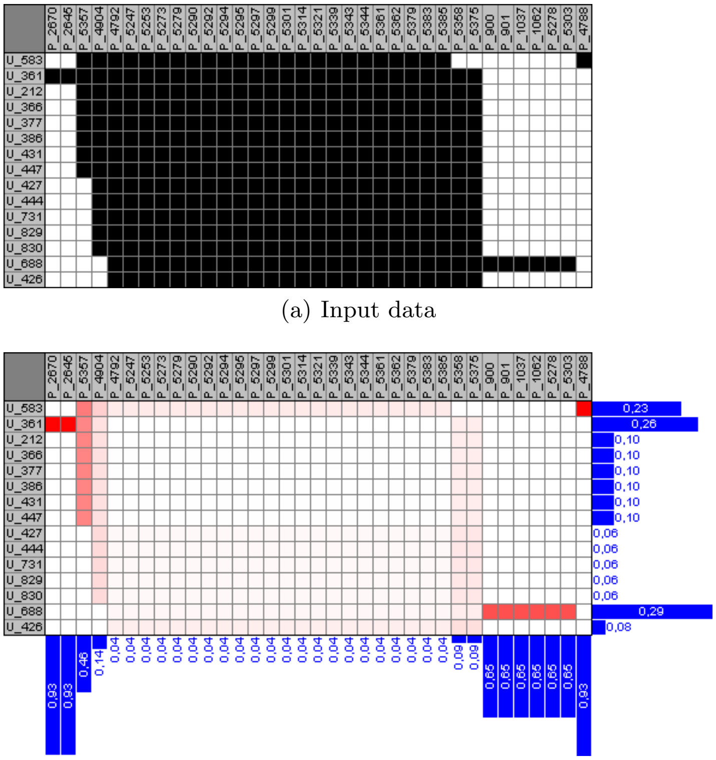
$$\mathit{perm\_risk}(p) := \sqrt{\frac{\sum_{u \in \mathit{users}(p)} \mathit{ass\_risk}^2(\langle u, p \rangle)}{|\mathit{users}(p)|}} \; . \tag{3}$$

By considering the root mean square instead of the arithmetic mean we give more importance to high risk values.

4 Experimental Results

We now show an application of our risk framework to a set of real data. Our case study has been carried out on a large private organization. Due to space limitation, we only report on a representative organization branch that contains 17 users and 72 permissions, counting 560 assignments.

Figure 1(a) depicts existing user-permission assignments in a matrix form, where each row represents a user, each column represents a permission, and a black cell indicates a user with a granted permission. Figure 1(b) depicts the same access control configuration, but the assignments colors indicate the corresponding risk probabilities. In particular, the cell color goes from red to white: Red means that the assignment has a high risk level when managed through



(a) Input data

(b) Risk probabilities

Fig. 2. Low risk users and permissions

RBAC; white means that it has a low risk level. Histograms on columns and rows borders respectively report the risk probability of managing permissions and users. Note that there are 6 users that are likely to be risky, mainly because they have a set of granted permissions that the majority of the other users do not have. This set is easily identifiable by looking at the permission histograms: Almost all the first half of the permissions are risky to manage. It is also possible to note that among the high risk users, two users have a slightly minor risk level compared to the other four. Indeed, these two users have similar permissions granted, and this is recognized as a kind of pattern within the data that reduces the overall risk.

Figure 2(a) depicts another access control configuration relative to a different branch of the same organization, while Figure 2(b) depicts the result of our risk function applied to this branch. Here, the risk levels of all the users are lower than 0.30. It means that, when adopting RBAC, the risk level is generally lower than in the previous example. In other words, role administration should make less mistakes in this second branch than in the first one.

5 Concluding Remarks

The risk management framework introduced in this paper allows role engineers and system administrators of an RBAC system to highlight those users and permissions that are more prone to error and misuse when roles are operating. A distinguishing feature of our proposal, other than that of being rooted on sound theory, is that role definition is not an input parameter for the risk analysis to be performed; indeed, our model only needs to know the access control configuration of the organization, optionally enriched with other business information. Finally, it has been applied on a real case, and results obtained showed the usefulness and viability of the proposal.

References

1. De Capitani Di Vimercati, S., Foresti, S., Samarati, P., Jajodia, S.: Access control policies and languages. Int. J. Comput. Sci. Eng. **3**(2) (2007) 94–102 Inderscience Publishers.
2. Coyne, E.J.: Role-engineering. In: Proceedings of the 1st ACM Workshop on Role-Based Access Control, RBAC '95. (1995) 15–16
3. Coyne, E.J., Davis, J.M.: Role Engineering for Enterprise Security Management. Artech House (December 2007)
4. Molloy, I., Li, N., Li, T., Mao, Z., Wang, Q., Lobo, J.: Evaluating role mining algorithms. In: Proceedings of the 14th ACM Symposium on Access Control Models and Technologies, SACMAT '09. (2009) 95–104
5. Colantonio, A., Di Pietro, R., Ocello, A., Verde, N.V.: A formal framework to elicit roles with business meaning in RBAC systems. In: Proceedings of the 14th ACM Symposium on Access Control Models and Technologies, SACMAT '09. (2009) 85–94
6. Colantonio, A., Di Pietro, R., Ocello, A., Verde, N.V.: Taming role mining complexity in RBAC. Computers & Security **Special Issue on "Challenges for Security, Privacy & Trust"** (2010) To appear.
7. American National Standards Institute (ANSI) and InterNational Committee for Information Technology Standards (INCITS): ANSI/INCITS 359-2004, Information Technology – Role Based Access Control (2004)
8. Colantonio, A., Di Pietro, R., Ocello, A., Verde, N.V.: A new role mining framework to elicit business roles and to mitigate enterprise risk. Decision Support Systems **Special Issue on "Enterprise Risk and Security Management: Data, Text and Web Mining"** (2010) To appear.