Budget-aware Role Based Access Control☆

Farzad Salim[a], Jason Reid[a], Uwe Dulleck[b], Ed Dawson[a]

[a] *Institute for Future Environments, Queensland University of Technology, Brisbane, Australia*
E-mail: {f.salim, jf.reid, e.dawson}@qut.edu.au
[b] *School of Economics and Finance, Queensland University of Technology, Brisbane, Australia*
E-mail: {uwe.dulleck}@qut.edu.au

Abstract

The suitability of Role Based Access Control (RBAC) is being challenged in dynamic environments like healthcare. In an RBAC system, a user's legitimate access may be denied if their need has not been anticipated by the security administrator at the time of policy specification. Alternatively, even when the policy is correctly specified an authorised user may accidentally or intentionally misuse the granted permission. The heart of the challenge is the intrinsic unpredictability of users' operational needs as well as their incentives to misuse permissions. In this paper we propose a novel Budget-aware Role Based Access Control (B-RBAC) model that extends RBAC with the explicit notion of budget and cost, where users are assigned a limited budget through which they pay for the cost of permissions they need. We propose a model where the value of resources are explicitly defined and an RBAC policy is used as a reference point to discriminate the price of access permissions, as opposed to representing hard and fast rules for making access decisions. This approach has several desirable properties. It enables users to acquire unassigned permissions if they deem them necessary. However, users misuse capability is always bounded by their allocated budget and is further adjustable through the discrimination of permission prices. Finally, it provides a uniform mechanism for the detection and prevention of misuses.

Keywords: Role based access control, insider threat, misaligned incentives, information asymmetry, price discrimination.

1. Introduction

Access control is challenging in an organisational setting because on one hand employees need enough access to perform their jobs, while on the other hand more access will bring about an increasing risk of misuse - either intentionally, where an employee uses the access for personal benefit, or unintentionally through carelessness, losing the information or being socially engineered to give access to an adversary [2, 3, 4]. The ultimate goal and responsibility of a security administrator is to allocate each employee the precise level of access required for their job - no more and no less [2, 5]. If this goal is met the allocation can be said to be *optimal*.

At the core of existing access control approaches such as Role Based Access Control (RBAC) [6] lies the implicit assumption that the optimal security policy that assigns permission to users can be constructed a priori and maintained correctly, i.e., by a security administrator. However, the process of access control in dynamic environments such as healthcare exhibits three characteristics that challenge the practicality of this fundamental assumption. First, usually there is *information asymmetry* between the administrator and the employees. Job functions (tasks) within an organisation are performed by employees who, due to the dynamic nature of the work, have special information concerning their particular sphere of activity. Often, this information is not available to administrators (or even supervisors) to precisely determine what permissions employees may need to complete their tasks. Even if the information is available, maintaining the correctness of the security policy is challenging and resource intensive due to the dynamism that exists in organisations [7, 5]. As a result any security policy is at best an approximation of true access requirements. Second, employees are generally

☆This paper is an extended version of our paper titled "An approach to access control under uncertainty" [1].

self-interested individuals [8, 9]. The optimality of the policy depends not only on whether an employee needs the permission to perform a task, but also on whether the employee decides to use the permission for performing the task. There may be divergence of preferences between the action an employee considers optimal and that which is optimal for the organisation. This divergence arises because employees may also seek to maximize their own self-interest. This becomes particularly problematic when employees draw personal benefit from misusing permissions. Third, *perfect audit*, or verification of employees' usage of permissions, is prohibitively costly [10]. This is primarily due to the non-rivalrous property of digital resources (i.e., a resource can be used by more than one employee or for more than one purpose simultaneously), and the plethora of applications that allow employees to make use of these resources. In many cases, the misuse of permissions can go undetected for long periods. For instance, a recent Verizon data breach investigations report found that more than 70% of data breaches (within organisations in the study) are only detected weeks after their occurrence [4, p. 55]. In addition to the above common characteristics, access control in healthcare must satisfy another unique and challenging requirement: "nothing must interfere with the delivery of care" [5, 11]. Hence, an access request may not be simply denied because it is not explicitly authorised by a predefined policy.[1]

In the face of these characteristics, any access decision based on a static security policy that binds access rights to users on the basis of predefined operational needs and assumed unlikelihood of misuse is doomed to be ineffective [7, 12, 13]. This contention is supported by the results of several studies [5, 14, 15], including an empirical study of database access logs of eight Norwegian hospitals by Røstad et al., [10, 11] which suggests that classical RBAC is unduly restrictive for healthcare. Røstad et al. report that clinicians' use of an exception handling mechanism that can override access requests denied by the static RBAC policy (i.e., if staff decide such information is necessary) is widespread. Indeed, they found that 17% of all record accesses occurred through the exception mechanism. Although introducing exceptions enhances flexibility, use of exceptions must be regularly audited to ensure they are not misused. However, due to the dynamic nature of healthcare and the static nature of the policy, it turns out that the use of exceptions is hardly an exception (i.e., 74% of the staff were assigned the permission to override denied access requests and 54% of active health records accessed in a one month period had been accessed as an exception [10]). They report, the sheer number of accesses via exception has made monitoring and misuse detection/prevention an impractical task. As a result, those staff who either maliciously or inadvertently misuse their access rights are unlikely to be held accountable [16, 17].

This paper is motivated by the shortcomings of RBAC when information asymmetry is present. More precisely, we consider settings where operational needs and a user's incentives to misuse resources are only partially known by the administrator while constructing an RBAC policy. We propose a novel approach to access control by adopting an existing RBAC policy as a reference point to *discriminate* the price of permissions for users, rather than using it as the only base for making an authorisation decision. Through this, those users who based on an existing RBAC policy possess a permission would pay a *base price* for a permission, while others pay an *elevated price*. In order to pay for access, users are given a limited *budget*, allocated according to the administrator's current knowledge of each user's operational needs. However, as users interact with the system their budget may be reduced according to the information available about their *type*, i.e., their propensity to misuse their budget.

In this paper we will propose a novel Budget-aware Role Based Access Control model (B-RBAC) and show that it can improve RBAC in four major aspects. First, it makes possible the specification of an *upper-bound* on the damage any employee may inflict in any period. For example, regardless of the number of database records a role potentially has access to via its assigned permissions, only as many records can be accessed as they have budget for. The proposal can therefore prevent a large scale 'record dump'. Second, the model directly promotes employee *accountability* and takes a step towards the alignment of employees' preferences such that they choose to observe the principle of least privilege. For instance, given two alternative mediums to access a record, secure that is cheap and insecure that is more expensive, employees with limited budget have incentives to use the cheaper option to preserve their budget and consequently they choose to use a more secure alternative. Third, the model enables employees to gain permissions that have not been preassigned to them (i.e., due to the incomplete knowledge of the RBAC administrator). This feature is particularly important for access control in environments such as healthcare where the inability or untimely access to resources may have profound consequences. Fourth, the model allows for

[1] Technically, this means the existing default rule of "deny - if in doubt" is no longer acceptable.

a uniform misuse detection and prevention mechanism, through monitoring users' budget spending. Where administrator's resource to audit is constrained, they can focus primarily on accesses with high price multiplier or verify the access usage of employees whose budget is exhausted.

The rest of this paper is organised as follows. In Section 2 we provide an account of the related works. Section 3 formally introduces our Budget-aware Role Based Access Control (B-RBAC) model. The notion of value (cost) of permissions and its implications on standard RBAC is defined in Section 3.1. The implication of explicit specification of the cost of tasks in quantifying a role's weight is introduced in Section 3.2. It is shown in Section 3.3 how by using a role's weight as the multiplier for access cost we can influence users to activate less powerful roles and adhere to the least privilege principle. Section 3.4 defines the concept of escalation, through which users can acquire unassigned permissions. Sections 3.5 and 3.6 respectively introduce two mechanisms, namely access discrimination and budget allocation that together ensure: 1) the upper-bound cost each user may incur is explicitly accounted for, 2) the cost of access is determined in part by their job function, and 3) a user's budget is parameterised by the history of outcomes of their prior choices (i.e., misuses). Section 4 provides an account of the security implications of the proposed model including the attacks that can be detected/prevented. Section 5 examines implementation considerations for the proposed B-RBAC model and proposes four core sub-models that vary in terms of their overhead on administrators and users. Finally, we will discuss future directions in Section 6 and conclude with Section 7.

2. Related Works

The optimal allocation of access permissions proves to be a complex task in practice. In [18], Zhao et al. employed the empirical results from [5], the study of access control in financial institutions, to provide an intriguing discussion of these complexities. They coined the terms *over-entitlement* and *under-entitlement*, respectively referring to the employees' acquisition of more or less permissions than they actually require to perform their tasks. The authors suggested that an information governance approach is required, which provides incentives such that employees' self-interested behaviour can result in a firm's optimal use of information. Their recent formalisation [19] is directly inspired by theoretical techniques to address the principal-agent problem in the field of economics. Despite our interest in the general direction of such a solution, their model is abstract and does not directly relate to any of the existing access control models. They also make some assumptions that may be difficult to address in practice. Similar to the game theoretic access control model proposed by Salim et al. [20], they assume it is possible to quantify the benefit of opportunities that can be seized by allowing employees to escalate (increase) their access permissions.

The risk-based approach to access control has emerged recently to address the under-entitlement problem flagged by the JASON report [7]. Cheng et al. [13] proposed a Risk-Adaptive Access Control (QRAAC) based on Bell-LaPadulla's Multilevel Security model (MLS) [21]. They introduce a flexible gap between allow and deny, where transactions that are denied in the MLS model may be allowed using some additional risk mitigation mechanisms. Following the JASON report, they adopt the notion of *risk tokens* which can be traded for extra permissions. A shortcoming of their approach lies in their basing the allocation of risk tokens on a subject's clearance level - the higher the clearance level, the more risk tokens they receive. A problem arises when a high clearance individual decides to misuse such privileges. With a large allocation they can do considerable (though bounded) damage. Also, their model is not concerned with over-entitlement, to address the users misuse of preassigned permissions. Furthermore, their formal framework is based on MLS, where objects are categorised based on their sensitivity level. However, in models such as RBAC this categorisation is unavailable. Not withstanding these limitations they have proposed a novel approach.

Liu et al. [22, 23] proposed a model to address the over-entitlement problem by assigning a risk budget for jobs and rewarding those employees who perform their tasks while consuming less than the allocated budget and punishing those who exhaust their budget before completing their tasks. In this way, the risk is communicated to the user and the cost of risky actions is shifted from the organisation to them. While their proposal is related to ours, it is not concerned with RBAC model. Furthermore, in their proposal users are only charged against their risk budgets when their access is considered to be an escalation. Hence, users' misuse of their already assigned permissions cannot be accounted for. Finally, reward and punishment are external to the model, applied ex post and are assumed to be influencing users misuse decisions.

Several models have also been proposed to improve the flexibility of the RBAC model. Motivated by the suitability of trust management concepts for access con-

trol in distributed systems, Dimmock et al. [24] introduced the notion of trust to RBAC such that access decisions take into account users' trustworthiness metric in addition to their roles. While this approach is more fine grained than making access decisions based on roles alone, the users' trust level is still static and predefined. Furthermore, the access decision is still based on a predefined trust threshold and no escalation capability is defined.

Nissanke and Khayat [25] introduced a formal risk ordering relation between pairs of tasks that belong to different roles. Their model considers the relative risk associated with situations when users belonging to different roles perform a similar task. Through this, delegation of permissions from one role to another is parameterised by the associated risk. However, the primary shortcoming of their approach, which also forms part of the motivation for our work is that their notion of risk is still static and assumed to be driven from the difference between roles. It ignores the risk that arises from the human users, associated to these roles. Furthermore, the users in their model can only perform the tasks that are pre-assigned to them either through direct assignment or delegation, hence there is no concept of escalation to address under-entitlement.

Celikel et al. [16] also proposed a risk-based approach to RBAC in the context of relational database management systems. Their work is mainly concerned with the risk of role misuse, defined as repetitive query submission, i.e., query flooding. They introduce the notion of occurrence rating that binds the probability of a misuse to the number of times a query has been submitted by the user. However, the proposed approach only focuses on the over-entitlement problem and the notion of misuse is limited in this definition. Furthermore, their model is not concerned with addressing under-entitlement or providing incentives to influence the users to make misuse (sending repetitive queries) unattractive.

Yemini et al., [17, 26] proposed MarketNet, a network architecture that uses financial instruments to regulate the use of shared network resources such as storage and processing power. The framework allows each network domain to control access and direct traffic to their shared resources through setting an access price for each resource. Their work is conceptually close to the B-RBAC model proposed in this paper, however they require each node in the network to generate its own budget through providing access to its shared resources, a requirement that is impractical for access control in an organisation setting. More importantly, there is no direct mapping between their proposal and any existing access control model, e.g., RBAC.

3. B-RBAC Model

The problem that we address in this paper is how an administrator of an RBAC system can allocate a task (i.e., unassigned permissions) to a user when the undesirable consequences of the execution of the task is contingent upon the ex ante unknown type of the user. Intuitively, binding the consequence to an unknown parameter (type) introduces the *uncertainty* that exists in access control due to unpredictability about future user behaviour. For the rest of this paper we consider the notion of type to embody any *private* information affecting users' preferences over how to use the permissions. We assume the type space of users to be a spectrum from benevolent to malicious, where allocating a task to a benevolent (malicious) user will have the least (worst) undesirable consequence.

The rest of this section provides a formal approach to define and annotate the notion of *undesirable consequence* to tasks, and elucidates the implication of such explicit specification on the core RBAC model.

3.1. Task Consequences

Following standard RBAC terminology [27], the set of resources that are subject to access control is referred to by O, the set of actions that can be performed on objects by A. The set of all possible actions on objects is referred to as tasks, $\mathsf{T} = \mathsf{A} \times \mathsf{O}$, and the set of all users by U. Let the set of all possible undesirable consequences of tasks to be exhaustive and incompatible (i.e., $C = \sum_{i=1}^{n} c_i$) and let there be a total order $\succeq$ on C. We write $c_i \succeq c_j$ to mean that the consequence c_i is *costlier* than c_j. Given this, we assume that there exists a mapping function that assigns a cost value to the consequences, i.e., that is the level of "badness". Since this valuation is organisation dependant, we introduce and use an organisation specific currency represented by ♭ when referring to the cost of a consequence.[2]

$$\begin{aligned} &cost : C \to \mathbb{R}^{+} \qquad s.t. \\ &cost[c_i] \geqslant cost[c_j] \iff c_i \succeq c_j, \ \forall i, j. \end{aligned} \tag{1}$$

Let the users' *type space* to be $\Theta = \{0, ..., 1\}$. We say that a user is *benevolent* when $\theta \in \Theta = 0$ and *malicious*

[2] There may be a direct mapping between the organisation specific cost (♭) of an undesirable consequence and the actual cost, the dollar value $. However, this may not always be the case. For example, in a hospital case, the undesirable consequence of not being able to provide care (e.g., due to under-entitlement) may not be quantified by a dollar value.

when $\theta \in \Theta = 1$.[3] Given a type space Θ, the possible undesirable consequences of a task are contingent upon the type of the user - capturing the dependency of the consequences on *how* the permission is misused. Formally, the relation between task, type and consequences is defined as:

$$f : \mathsf{T} \times \Theta \rightarrow C. \tag{2}$$

Hence, since a combination of a task $t \in \mathsf{T}$ and a type $\theta \in \Theta$ will produce a particular consequence $c \in C$ (i.e., $[t, \theta] \rightarrow c = f[t, \theta]$), for each task there exists a subset of undesirable consequences based on the possible types of users that can misuse the task (eq. 2). Further, since any subset of C has a maximum cost (eq. 1), it naturally follows that for any task a *maximum cost* can be determined. Formally, we write as $\max_C[t]$ (read, the maximum cost of t):

$$\begin{gathered} \max_C : \mathsf{T} \rightarrow \mathbb{R}^+ \quad s.t. \\ \max_C[t] = c_i \Longleftrightarrow \nexists f[t, \theta] = c_j \\ where\ c_i < c_j, \forall \theta \in \Theta. \end{gathered} \tag{3}$$

The process of quantifying the maximum cost of an operation on those resources that have *intrinsic* value is intuitive. For instance, if we are designing an access control system to utilise access to a resource such as a printer, the cost of a task "print a document" can be defined as: the unit cost of print per page (e.g., ♭0.1), times the number of pages in the document. In controlling access to limited resources such as network bandwidth where quality of service is important, the cost of access may be driven from the marginal cost of the facility as well as an extra premium for the cost imposed on others using the crowded network [28].

The explicit quantification of the maximum cost of an operation on information resources that do not have an intrinsic value can also be determined by the same logic, even though it may be less intuitive. The value of these resources depends on the potential cost of misuse, for example the cost to reconstruct lost data, restore the integrity of the fabricated or intercepted data or pay the functional liabilities for public disclosure of confidential or private data [13, 7].

To clarify how the actual cost of such tasks is determined consider the example illustrated in Figure 1a. Let Alice be an administrator in a hospital, managing a database of patients' records. Further assume two query that can be executed against this database, Further consider two sample queries that can be executed against this database by hospital staff, t_1: access up to ten rows, t_2: access a table (of up to 100 rows). For now assume that Alice is only concerned about potential privacy breaches that would incur a cost of ♭2 per record (i.e., the fine specified by data breach laws). Alice knows that a staff member can be either benevolent or malicious, $\Theta = \{benevolent, malicious\}$ where benevolent does not breach the privacy policy but the malicious does. Hence, the universal set of consequences would be $C = [0, ..., 200]$. Clearly the potential consequences of each task are as follows: $t_1 = [0, \ldots, 20]$ and $t_2 = [0, \ldots, 200]$. Hence, $\max_C[t_1] = ♭20$ and $\max_C[t_2] = ♭200$.

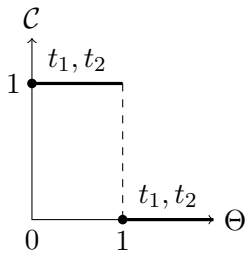
There is no restriction on the granularity of cost assignment. The granularity of associating cost to tasks is directly dependant on how much information the administrator has about the resources that are being governed. One can envisage a more comprehensive set of rules, where given a matrix of records $D_{i,j}$ and an action, a cost matrix $C_{i,j}$ is produced such that each cell has a unique cost.

The explicit assignment of (maximum) cost to tasks, even though an approximate measure, has two important advantages. It quantifies the potential upper-bound cost that the misuse of any task may incur. More importantly, it provides a basis for a relative comparison of tasks. Formally, we can extend the relation $\geq$ to also be applied on the set T:

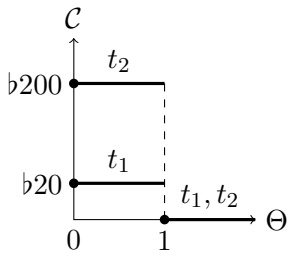
$$t \geq t' \Longleftrightarrow \max_C[t] \geqslant \max_C[t'] \quad \forall t, t' \in \mathsf{T}. \tag{4}$$

As illustrated in Figure 1a, given eq. 4 and the above example, we can deduce that $t_2 \geq t_1$ (the maximum cost of t_2 is greater than t_1). To elucidate the implications of the above exposition let us apply it to the standard RBAC model. In RBAC users can be either *unauthorised* or *authorised* ($\Theta = \{0, 1\}$), since the notion of consequence is not expressible within the model, access decisions are based on the types of users, rather than on consequences of tasks in question. This would inherently mean that consequences are binary as well, implicating *acceptable* or *unacceptable* cost ($C = \{0, 1\}$). As Figure 1a illustrates such a binary representation cannot express the distinction between tasks in any unauthorised or authorised state. The cost of t_1, t_2 used by an unauthorised user is indistinguishable.

[3] Note that the actual type space is application dependant. It captures the possible relevant misuse actions - the ones that the organisation may be concerned about their undesirable consequences, e.g., corrupting the record, publicizing private information, etc. In this sense a user's type is an abstraction of the potential harm they can cause by misusing their permissions.



(a) Binary Types and Binary Consequences



(b) Binary Types and Multiple Consequences

Figure 1: Graphical Representation of States and Consequences

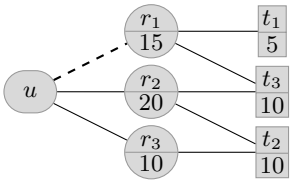


Figure 2: Role Escalation.

For example, let us assume Bob is an intern at the hospital with no authorisation to access a patient's addiction history, explicitly neither to read on the screen (t_1) nor to print (t_2). An emergency arises and Bob must access one of the patient's records. Obviously, providing an exceptional access to execute a task that limits a potential misuse opportunity (t_1) is preferable. This is what is referred to as *most-tolerable-privilege* in [29]. As shown in Figure 1b, such a distinction can be made when the upper-bound cost of tasks is explicit.

3.2. Role Weight

A role is a logical grouping of tasks, required for performing a particular job function. The set of all roles is denoted by R and the set of all predefined task-role associations is referred to as *permissions*: $\mathsf{P} \subseteq \mathsf{T} \times \mathsf{R}$. Since roles may differ both in terms of the quantity and the quality of their tasks (i.e., the extent of undesirable consequences if a task is misused), the designers of RBAC suggested in [6, p. 5] that "powerful roles can be kept dormant until they are needed, to provide an element of least privilege and safety". However, the authors only informally state the concept of a role's power (hereafter, *weight*), as the RBAC model does not allow it to be formally defined.

This notion is naturally captured when tasks are explicitly annotated with their maximum cost (eq. 3) - a role's weight corresponds to the cost of its associated tasks.[4] There are however several options available to compute a role's weight. Here we use arithmetic summation over the cost of the role's tasks.[5] This guarantees the cost of a role is at least equal to its most costly task while also reflecting the cheaper tasks that are associated with the role. Figure **??** illustrates role's cost as the aggregate cost of the tasks assigned to them (permissions). It can be seen that taking the weight of a role to be the cost of its most costly task would provide an inaccurate picture by implying that roles r_2 and r_3 to have equal weight. Formally,

$$\max_{\mathcal{W}}[r] = \sum_{\forall t \in \mathsf{T} | (t,r) \in \mathsf{P}} \max_{C}[t]. \tag{5}$$

An explicit quantification of a role's weight can be used to provide a notion of priority for administrative functions such as role activity monitoring and audit. More importantly however, a role's weight can also be utilised to construct a disincentive for users, directing them to perform their tasks through cheaper roles whenever possible. Note that, this was one of the motivations for introducing the concept of *sessions* in RBAC, which enable users to activate only those roles necessary to complete their jobs [6]. However, enforcement of such a

[4] It is important to mention that a task's cost and a role's weight do not directly represent risk, that by definition is composed of likelihood and consequence. Here cost and weight are indications of potential maximum consequence and do not capture the likelihood, which will be taken into account in the following section

[5] Another alternative is to take the cost of the most costly task of the role. This is similar to our approach in determining the cost of tasks. However, it undermines the distinction between tasks and roles: tasks have mutually exclusive consequences, while roles may be composed of many tasks inducing a union of their consequences.

practice is not trivial because only the user at the time of performing a job can determine exactly what roles they may need in a session to complete the job. Hence, so far this desire could not be enforced in RBAC as users did not have anything at stake if they simply chose to activate a session where all their roles were active in it.

The following section will show how the weight of a role can be used to adjust the price of permissions and facilitate the enforcement of this practice.

3.3. Price of Permission

In RBAC, users are assigned to roles ($\mathsf{UA} \subseteq \mathsf{U} \times \mathsf{R}$), and they are only authorised to *execute* the tasks of those roles that the administrator has already assigned to them:

$$execute[t, r] \rightarrow \exists(t, r) \in \mathsf{P} \wedge \exists(u, r) \in \mathsf{UA}. \qquad (6)$$

Since user-role and role-task relationships are many to many, a user may be able to perform a task through more than one role. For example, in Figure **??**, t_1 is available through r_1 as well as r_2. As argued in Section 3.2, it is important to motivate users towards using *cheaper* roles to perform their tasks. For instance, administrators should use their root account only when they cannot perform their tasks through their low privilege employee account. To achieve this we use the weight of a role as a multiplier for the task that is being accessed through the role. Note however, since each task has already contributed to the weight of a role (eq. 5), multiplying or summing the role's weight ($\max_{\mathcal{W}}[r]$) by the task's cost ($\max_{C}[t]$) double counts the cost of the task. To make this clear, consider a role that has only one task, for instance in Figure **??**, using t_2 through r_3. If simple multiplication (or addition) is used to derive the roles weight, the task would cost ♭100 or (♭20) even though the weight of r_3 is driven directly from t_2. The following equation addresses this by ensuring the cost of a task is increased only by the proportion of the cost of *other* tasks that are associated with the role:

$$\max_{C}[t, r] = \left(\frac{\max_{\mathcal{W}}[r]}{\max_{C}[t] + \epsilon} - 1 \right) + \max_{C}[t] \qquad (7)$$
$$s.t. \quad (t, r) \in \mathsf{P},\ \epsilon > 0.$$

Intuitively the $\max_{C}[t, r]$ is the cost of executing a task t through the role r, where ϵ is a negligible non-zero constant to ensure the validity of the operation if the cost of the task is zero. Hence, when a task can be performed through more than one role, the roles can be compared in terms of their effect on the cost of the task. For example, given eq. 7 and Figure **??**, executing t_2 would be costlier through r_2 (i.e., ♭11.5) than r_3 (i.e., ♭10).

To show the direct implication of eq. 7 on the traditional RBAC model, let us introduce the notion of *budget* with the following characteristics:[6]

- it is a virtual currency, specific to the organisation,
- it can only be allocated to users by the administrator,
- it is the only means through which users can pay for tasks,
- it is not transferable from one user to another,
- it cannot be forged, and
- it is valid only for the period for which it is being allocated (i.e., users' remaining budget expires at the end of each budget allocation round).

Given a limited budget to users, all other things being equal, it follows that the more budget a user has the more task execution capability they acquire. Hence, all users regardless of their type or intentions prefer more budget to less budget. It then follows that when the only discriminating factor between two roles is their price, users prefer the cheaper role to perform their task. Therefore, assuming the inconvenience of changing roles is negligible, the utility of a role for a user is inversely proportional to the role's weight.

This is an important achievement, as it indirectly connects the observance of the principle of least privilege with users' utility. Here users' incentives to perform their job with least budget consumption is aligned with the access provider's incentive to enforce the least privilege principle (i.e., users use less powerful roles). Furthermore, since the cost of a role is proportional to the number of its tasks, it follows that for users who are assigned to roles, unnecessary permissions, or tasks assigned to the roles, are no longer considered as "free permissions". This is in contrast to current practice where it is beneficial to users to overestimate the permissions they require and demand that administrators assign as many permissions to the roles as possible [7, 5]. Here users and administrators have incentives to cooperate and determine an optimal level of permissions for roles.

[6]The budget allocation function in the context of RBAC will be formally introduced in Section 3.6.

3.4. Escalation Capability

In RBAC users can only execute those tasks that have been preassigned to them by the administrator (eq. 6). However, as stated earlier, administrators usually have incomplete information about users' access needs and there may be situations where a user needs to perform a task they do not have permission for. These situations if untreated, lead to suboptimal access decisions (i.e. under-entitlement). Of course manual update of user-role assignments is possible, however, it is inefficient, particularly in time-critical emergency situations [10]. Furthermore, in many circumstances, users require only a transient access to a role rather than permanent access. Research shows that the approaches to satisfy such needs are ad-hoc and usually motivate administrators to allocate more access than necessary or forget to remove the "temporary" assignments [7, 18, 5]. The rest of this section is dedicated to a formal and systematic treatment of *escalation* in RBAC. Formally defined as:

$$escalate[u,t] \rightarrow \exists (t,r) \in \mathsf{P} \wedge \nexists (u,r) \in \mathsf{UA}. \quad (8)$$

Intuitively, escalation is the act of executing a task through a role that has not already been assigned to the user. This definition is very general in the sense that it enables users to theoretically use any of the existing tasks in the system regardless of their intentions, ranging from situation to stealing documents. This definition of escalation entails the well known concept of *delegation*, which becomes a special case of escalation, in which, the user u who is escalating ($escalate[u,t]$) possesses a support (e.g., in the form of a delegation certificate [30]) from another user u' who is already a member to the role: $(u',r) \in \mathsf{UA}$ where $(t,r) \in \mathsf{P}$.

Such a support is in fact a *control instrument*, used to determine whether the escalation should be allowed or not. Our definition of escalation opts for the separation of the concept from such instruments. In an access control system where users have to pay through their limited budget to acquire access to resources, the most effective control instrument at our disposal is *price discrimination* that will be introduced in the following section.

3.5. Price Discrimination

So far we have only considered a flat pricing of permissions, where all users pay the full price of the tasks they want to access (eq. 7). However, flat pricing has one important drawback when escalation is introduced to the model. That is, the users who perform jobs that involve costly tasks must be allocated a large budget. These users then pose a great risk as they can escalate and acquire any permission that costs less than their highly elevated budget that has been allocated to them.

For instance, consider a hospital employee who may need to migrate thousands of patients records to a new system in a given period. In a model where escalation is possible, the employee may use the budget allocated for this task to instead escalate his permission and access the financial records of patients. The objective is to reduce such questionable escalations while still allowing the escalations that may be needed by benevolent employees to complete their jobs.

We adopt *price discrimination*[7] concept to allow for variable pricing of permissions by using *price multiplier*, $\varphi \geqslant 1$. Conceptually, introducing the price multiplier allows an identical permission to be transacted at different prices for users based on an abstract factor that we refer to as users *likelihood* to misuse the acquired permission.

Although a precise determination of such likelihood is generally impossible, one plausible criteria is predefined operational needs. In an RBAC system, the RBAC policy provides the reference point to decide whether a user "needs" or has "competence" to perform the task, implicitly defined through user-role-permission relationships. We can therefore use RBAC policy to charge a base price for task executions (eq. 6) and elevate the price for escalations (eq. 8). Through this treatment, the price a user has to pay is adjusted to reflect users' assumed eligibility, based on the RBAC policy. Formally:

$$price[t,r] = \begin{cases} \max\limits_{C}[t,r] & if \quad execute[u,t] \\ \max\limits_{C}[t,r].\varphi & if \quad escalate[u,t] \end{cases} \quad (9)$$

Determining the multiplier rate is application specific and may be very elaborate when taking into account factors such as the roles a user already possesses and the relevance of these roles to the role that is being used to make the escalation possible. For instance, a doctor's permission escalation to read the record of a patient's parents (to search for potential genetic causes) may be considered as "relevant", hence elevated substantially less than a finance manager who is escalating to perform

[7] The concept of price discrimination is widely known in the field of economics and can be seen in everyday transactions. It captures situations when identical goods or services from a single provider are transacted at different prices for different consumers. An example related to our work is from insurance markets, where the same insurance policy is sold for different prices to customers based on their risk profile, which captures factors such as accident history or age.

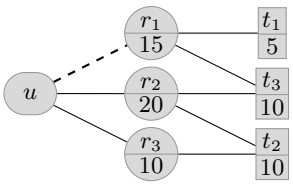


Figure 3: Role Escalation.

the same task. Furthermore, sometimes escalations may need to be prohibited or made unattractive for users. An example of such situations is when two roles r_1 and r_2 are subject to separation of duty constraint. To enforce this, when escalation violates this constraint, the cost of the escalation for employees may be raised to $\varphi = \infty$ in order to disallow such escalations. Role mining techniques such as [31] that provide quantitative notion of distance between the roles on the basis of an organisation's operational structure may be adopted for determining the rate of price multiplier.

For example, consider Figure 3 and let u (an intern) be a member of roles r_2 and r_3 but not r_1. Assume that given statistics about the number of patients admitted to the hospital, the number of medical staff, the administrator knows that a full time intern requires 200 units of organisation's access budget (♭200) to perform their tasks within a period of a week. Further let $\varphi = 5$ to say that that cost of escalations are five times the normal price. Given these assumptions, Bob can use the assigned permissions t_2 either 15 times through r_2 or 20 times through r_3 over the course of a week (using eq. 5 and eq. 9). However, let us assume that Bob wants to access the task t_1, either due to an unexpected legitimate need or to satisfy his curiosity about a patient [32]. Given Bob's budget and the cost of escalation (eq. 9), he has enough budget to perform t_1, but enacting such an escalation would exhaust a large portion (♭35) of his budget for the period. We will leave the discussion on implications and approaches to handling early budget exhaustion to Section 4.

3.6. Budget Allocation Function

The ability of users to escalate or execute a task, regardless of their intentions, is limited by their available budget. Therefore, it is clear that the appropriate allocation of budget is a powerful tool for ensuring users' can escalate their permissions (i.e., addressing under-entitlement), and also limiting users' misuse capability (i.e., addressing over-entitlement). So far we have simply assumed users are allocated a budget by the administrator. In this section we will formally introduce how such a function can be implemented given an existing RBAC policy.

In general the budget allocated to a user must be equal to the total cost of the tasks the user is supposed to perform for a given period. So the budget is directly related to the frequency, $\lambda \in N$, by which the users need to execute tasks in order to perform their job. In practice λ can be coarse grained, driven from the role itself and be the same for all users assigned to the role. Otherwise, when comprehensive workflow information is available, the allocation of budget can be user-specific, allowing users with the same role to vary in terms of the number of tasks they are entitled to execute. For instance, a medical staff member who works part-time is allocated less budget than a full-time counterpart. Regardless of the method and granularity by which λ is determined, the users' (u) allocated budget can be formally written as:

$$\mathcal{B}_{[u]} = \sum_{(u,r)\in \mathsf{UA}} \sum_{(t,r)\in \mathsf{P}} \lambda . \max_{C} [t, r], \ \forall u \in \mathsf{U}. \qquad (10)$$

As we have established so far, one characteristic of a usable access control model is its ability to capture the dynamic nature of the system. This refers to changes in any factor that is relevant in making access decisions. This may include changes in the characteristics of the resources (e.g., object classification) or changes in the need for providing access for performing jobs (e.g., emergency situations). So far, through introducing the notion of task cost, which can be dynamically determined based on any application specific requirements and adjusted through escalation price multiplier component of the proposed access discrimination instrument, the proposed model provides a degree of dynamism. Formally, using eq. 9 and eq. 10, it can be seen that the users' ability to escalate given their budget is inversely proportional to φ. Hence, users' access can be discriminated based on their assumed operational needs stated in the RBAC policy.

However, one crucial aspect of dynamism relates to changes in user *behaviour* which is not explicitly captured in existing RBAC model. Currently it is assumed that there exists an external pre-screening mechanism to determine users' credentials (i.e., trustworthiness, roles). Hence, the assignment of a user to a role inherently indicates: the user's operational access needs, their capability in performing the tasks, *and* their unwillingness to misuse the permissions associated with the role (i.e,. trustworthiness). However, even if we assume the operational needs are accurately stated in the

policy and remain unchanged, users' willingness to misuse permissions may change. For instance, in [20] we established that a selfish employee's decision to misuse permissions is dependant on their payoff which takes into account and may change with respect to the value of the resource as well as the probability and severity of punishment.

To address this we utilise the existing estimators of users' behaviour that may be available to the access control system in order to dynamically adjust the budget users will be allocated.[8] Let $\beta \in [0, 1]$ denote the probability that the user is malicious, ignoring the possibility that β may not be an accurate reflection of the users' actual type θ, we propose the dynamic budget function to be:

$$\mathcal{B}^{*}_{[u,\beta]} = \left(\sum_{(u,r)\in \mathsf{UA}} \sum_{(t,r)\in \mathsf{P}} \lambda . \max_{C} [t, r] \right) (1 - \beta) \,, \; u \in \mathsf{U}. \tag{11}$$

The above budget allocation function explicitly interlinks the available knowledge about users' types to the budget they will be allocated. Given this, regardless of the initial estimation of users' required budget to complete tasks, the allocated budget will be automatically adjusted in proportion to the perceived changes in users' behaviour as they interact with the access control system. For instance, each member of a role (e.g., nurse) may be allocated a different budget despite the fact that they are all a member of the same role with same operational access needs. Furthermore, this approach internalises the concept of punishment - as evidence regarding a user's maliciousness increases ($\beta \approx 1$) the user's budget approximates to zero ($\mathcal{B}^{*}_{[u,\beta]} \approx 0$).

In some organisations the unpredictability of users' behaviour may be an artefact of the organisation's dynamic nature. As a result, monitoring and flagging potential misuses based on users departure from normal behaviour is costly and may be unreliable because of the complexities pertaining to the definition of "normal" behaviour [33]. In such circumstances the users' allocated budget is solely based on their assumed operational access needs (i.e., $\beta = 0$ is considered to be the default value).

In Section 3.3 it was mentioned that rate of φ is influenced by users' likelihood to misuse escalations based on operational needs that can be deduced from user-role relationships in an RBAC policy. Note that the users' misuse probability (β) based on behavioral indicators can also be used to adjust the price multiplier, φ. By allowing this rate to be positively proportional to β we can ensure that a user's budget is adjusted not only at the time of budget allocation cycle, but also after the budget is allocated. For example, the price multiplier of escalation requests by those employees who receive a job termination notice may increase, preventing them from extracting patients' records with their remaining budget.

[8] Note that, even though users' type (i.e., θ) is assumed to be strictly private, the outcome of their actions may be observed through log history, audit or any real-time monitoring mechanism (e.g., intrusion detection system). Bishop et al., [33] proposed one such technique for predicting employee's potential misbehaviour.

4. Security Implications

This section will introduce the primary advantages of the proposed B-RBAC model. It will discuss in more detail how the model can assist in addressing three important risk management concepts, namely, risk communication, risk transfer and risk control. Further, it will explain how these concepts along with escalation capability can help in reducing under-entitlement and over-entitlement problems. The advantages of the model for facilitating administrative tasks i.e., monitoring and detection of employees misuse, is also discussed.

4.1. Risk Communication

In [22, 23, 20] it is argued that permission misuse can be categorised based on the intentions behind them. Inadvertent or accidental misuse is attributed to those employees with incomplete or incorrect information about the potential risk of their actions for the organisation. It is also suggested that inadvertent misuse can be reduced if the potential risk of employees' actions are effectively communicated to them. In the B-RBAC model the price of permissions allows for the explicit communication of the potential risk of actions. Hence it positively contributes to addressing inadvertent misuse.

As an example, consider a nurse who uses an RBAC-aware email client to transfer a patient's records to an external laboratory. The email can be sent via either encrypted or unencrypted emails. Without the loss of generality assume that sending an encrypted email is less efficient (i.e., abstracting for potential cost of security). With no information available about the security risks associated with each option, the nurse may choose sending unencrypted records. In contract, the B-RBAC implementation of the email client explicitly takes into account the number and the sensitivity of the attached records and presents the nurse with the price

for each alternative.[9] Now even if we assume the nurse has unlimited budget and therefore no direct incentive to choose the cheaper (encrypted) option, simply flagging the potential risks of unencrypted transfer may be enough to eliminate the inadvertent disclosure of patients' records by a benevolent nurse.

4.2. Risk Transfer

Risk communication alone may not provide enough incentive for self-interested employees, whose decisions are influenced by what maximises their utility. In [20] we argued that the interaction between the administrator (i.e., indirectly the organisation) and the self-interested employees is a form of risk marketplace wherein the risk of employees' actions should be transferred from the administrator to the employees in order to effectively influence their behaviour. In general, the risk transfer could take place by manipulating factors such as the accuracy of misuse detection mechanisms and the magnitude of external punishments (rewards) for undesirable (desirable) actions.

In the context of B-RBAC implementation of the email client, the mere knowledge of the risk of transferring patients' records unencrypted does not directly affect the decision of a selfish nurse who has an unlimited budget. However, by giving a limited budget to the nurse, since sending the email unencrypted consumes more of nurse's budget, everything else being equal, the self-interested nurse now is given an incentive to choose the cheaper encrypted transfer. Essentially, by choosing the unencrypted option the nurse is paying more for the increased risk to the hospital - risk arising from an increased likelihood of data breach that patients' records are exposed to when sent unencrypted. The nurse's incentive to preserve their budget is due to the tacit assumption that the faster the budget is exhausted, the higher the likelihood that she/he will be audited and potentially held accountable. Section 4.5 provides a detailed discussion about the relationship between budget exhaustion and usage monitoring.

4.3. Risk Control

Employees may not only be inadvertent or self-interested, but malicious. In other words, they may actively seek to inflict cost to the organisation. Therefore, by definition the price of permissions may not affect the behaviour of these employees, furthermore they may not even be deterred by increasing the expected punishment.

Given the email client example, a malicious nurse may not only try to send the unencrypted email, but if possible, publicise patients records. In an RBAC implementation of the client the malicious nurse can access all the patients' records available to a role nurse. In contrast, with the B-RBAC implementation of the client, a malicious nurse is bounded by the budget they have been allocated. The budget is therefore an upper bound on the aggregate cost that can be imposed by the nurse. Since budget is allocated individually, it is in fact a estimate of the risk the hospital has already decided to tolerate for the nurse.

4.4. Escalation Handling

By introducing the escalation capability, the B-RBAC model provides a means to address the under-entitlement problem. By the same token, it may also be used by malicious users to acquire unintended permissions. However, the extent to which escalation can be used is directly controlled through the following mechanisms:

1. the aggregate amount of damage that may be incurred is restricted by the budget allocated to users.
2. the budget allocation function proposed in Section 3.6 is parameterised by the output of online monitoring mechanisms to adjust the users' disposable budget based on their observed behaviour (β).
3. the administrator has additional control over the escalations through personalising the escalation rate (φ), which as we discussed, may take into account the application specific factors such as users' trustworthiness, operational need, and access history.

4.5. Effective Administration

Here we divide the functions that consume an administrator's time into two categories. First, the specification of (RBAC) security policy, which entails the identification of users, roles, actions, resources, conditions as well as the user-role and role-permission assignments. The second function is audit and misuse detection that usually requires access log analysis. In practice both of these administrative functions can consume considerable time. In the following sections we will discuss how the monitoring and analysis of users' budget exhaustions can provide a uniform mechanism to facilitate these functions.

[9] The approach taken for representing the price of permissions or potential warnings are implementation concerns and outside the scope of the current paper. The common risk representation mechanism where a set of colours, red, orange, yellow and green are used to represent a magnitude of risk is a simple and intuitive implementation approach.

4.5.1. Policy Maintenance

As we discussed in Section 1 and Section 2, under-entitlement and over-entitlement problems are common in practice, primarily because a correct specification and maintenance of access control policies is difficult and time-consuming. In an RBAC system, to avoid under-entitlement, users are usually allocated more permissions than they actually need, while over-entitlements mostly remain undetected or tolerated in the interests of efficiency.

Under-entitlement can also occur in a B-RBAC system when users budget is exhausted due to administrators incorrect prediction of users' access requirements for a given period. However, unlike in RBAC, over-entitlements are transparent, represented by the employees remaining budget at the end of a period. Hence, we envisage through analysis of budget spending patterns, users' budget can be quantified with sufficient accuracy to reflect the actual need. For instance, since budgets are allocated periodically, the unique characteristics of the period may influence the quantity of the required budget. As an example, statistics may suggest that doctors have more patients (and hence need more budget) during the months of December and January because less doctors are available in the hospital and more incidents occur due to higher number of road accidents.

4.5.2. Misuse Detection

The analysis of users' budget can also assist in misuse detection. Assuming that the administrator of a B-RBAC system has correctly allocated the budget users need to complete their tasks for a given period, the exhaustion of budget can be an indicator of potential misuse of permissions. Also, the ratio of users' 'remaining budget' to 'remaining duration' or an abrupt change to this ratio may be used to indicate misuse. This allows a user's potential misuses to be detected even if they deliberately avoid exhausting their budget.[10] Finally, since the price multiplier indicates the potential inappropriateness of the escalation from the administrator's perspective, the scale of the price multiplier can used as a mechanism to prioritise escalation monitoring, with larger multipliers receiving greater attention.

The analysis of users' budget for misuse detection can also assist in detecting two major types of external attacks: impersonation attack and denial of service attack.

[10] Note that when the threshold is not common knowledge (private to administrators), strategic users can no longer abuse the detection mechanism by spending their budget only up to the threshold to avoid being detected.

Impersonation Attack An outsider may acquire the credentials of an employee and access the system. The consequences of a successful impersonation attack in a traditional access control model including RBAC can be devastating as such attacks are difficult to detect or prevent. The adversary can access any and all resources for which the legitimate user held privileges without affecting the actual user's access capabilities. This is not the case in B-RBAC. Even though the attack is still possible, any access by the attacker is counted against the user's budget. Hence, users can detect the reduction in their budgets. Even if such detection does not happen, the consequences of such attacks are strictly limited by the available budget for the period.

Denial of Service Attack A malicious insider may try to prevent other users from accessing resources by performing a denial of service attack. The type of DoS attack most relevant to this proposal is *query flooding* against databases, where the malicious user sends a large number of select or update queries to a targeted database [34]. Current techniques to prevent such attacks require comprehensive real-time analysis of query log files and assumptions about normal patterns of access that suffer from a high incidence of false-positives [16, 34]. In B-RBAC these attacks will have little impact and will be easy to prevent, as a user's ability to execute tasks is bounded by their limited budget. The exhaustion of a user's budget will lead to termination of the attack and the possible detection of the malicious user.

5. Enforcement Considerations

This section will discuss implementation concerns surrounding the proposed B-RBAC model. We will first introduce the core characteristics that an access control system must possess in order to implement the B-RBAC model. Then the optional features of the B-RBAC model will be discussed. We take the original RBAC model [27] as a baseline to study the extra overhead the adoption of these features can impose on the administrator or the user in a B-RBAC system.

5.1. Enforcement Sub-Models

There are three core requirements that must be satisfied by an access control system that aims to adopt the proposed B-RBAC model: First, it must allow for

Model	Transparent Budget?	Transparent Cost?	Finite φ	Security Implication
B-RBAC0	✗	✗	✗	RBAC + Effective Monitoring and Detection + Bounded Risk
B-RBAC1	✓	✗	✗	B-RBAC0 + Risk Transfer
B-RBAC2	✗	✓	✗	B-RBAC0 + Risk Communication
B-RBAC3	✗	✗	✓	B-RBAC0 + Escalation Handling

Table 1: Enforcement Models

each user to be assigned a mutable attribute budget. Second, it must allow for objects to have a mutable attribute cost associated with them. Third, the users' acquisition of permissions must be solely conditional upon the availability of budget, rather than directly based on the RBAC security policy.

There are also three optional features. First, the transparency of budget: whether users are informed of the amount of their budget. Second, the transparency of cost: whether users are aware of the price of permissions. Third, finiteness of φ: whether escalation is allowed, which requires the administrator to explicitly specify the rate of multiplier on the price of escalation, rather than imposing an infinite multiplier on all escalations. As shown in Table 1, when the transparency of both permission prices and users' budgets, as well as the finiteness of escalation multipliers are made optional several implementation sub-models emerge.[11]

As Table 2 summarises, inclusion of each option introduces an overhead for either the administrator or the users of the system. The primary overhead for the administrator arises from the amount of effort needed to configure the B-RBAC system. For users, the main overhead are the decisions about budget expenditure. Therefore a tradeoff needs to be made between the capability of the model and these overheads. In the following section we will analyse B-RBAC sub-models with respect to these overheads.

5.1.1. B-RBAC0

As shown in Table 1, B-RBAC0 is the closest to RBAC. The infinite tax on escalations means that the model is at least as strict as the RBAC model. This means, users can only access those permission sets that have been pre-assigned to them by the administrator through user-role and role-task associations.

In comparison to RBAC, B-RBAC0 increases the effort (time) that the administrator needs to invest in implementing the policy as each task must be explicitly assigned a cost ($\max_C [t]$). Furthermore, for the budget allocation function to calculate users' budgets, it must also be provided with the task execution frequency (λ).-[12]

B-RBAC0 can also assist administration in two major aspects. It facilitates misuse detection through the monitoring of users' budget. This is because the price of permissions and budget are still used for making access decisions, even though the are not transparent to users. Furthermore, users' limited budget still ensures that there is an upper-bound on the risk they can expose the system to. However, since permission prices and users' budget are not transparent, it can neither support risk communication, nor does it support risk transfer.

5.1.2. B-RBAC1

The administrative overhead of B-RBAC1 and B-RBAC0 are equal. But, users' overhead increases in B-RBAC1. This is because budget transparency means users are faced with a decision as to whether to perform tasks or retain their budget. As shown in Table 1, in B-RBAC1 users can not determine the cost of resources in advance. This implementation may be suitable for scenarios where the cost of a permission in itself may reveal some information about the importance of the resource. This revelation may in turn influence some users to access these resources out of curiosity. For example, in a hospital scenario where the cost of a patient's record has a correlation with their fame, transparent cost of resources may trigger curiosity in hospital staff. The scandal surrounding Nadya Suleman's case [32] is a real life

[11] Note that even though there are 8 sub-models, sub-models B-RBAC4 to B-RBAC7 are not discussed as they are simply the hybrid implementation of B-RBAC0 to B-RBAC3.

[12] Note that λ can also be determined indirectly with less administrative cost: the administrator can allocate a very large budget to users and observe how much budget is left-over after each period. By observing these budget leftovers, the administrator can acquire a more accurate knowledge about users' budget need.

Model	User Invovement Cost	Administrative Cost
B-RBAC0	= RBAC	> RBAC
B-RBAC1	> B-RBAC0	= B-RBAC0
B-RBAC2	> B-RBAC0	= B-RBAC0
B-RBAC3	= B-RBAC0	> B-RBAC0

Table 2: Enforcement Models: User and Administrative Cost

example of curiosity driven misuse of permissions in a hospital.

5.1.3. B-RBAC2

In B-RBAC2 the users' budgets are not transparent. This captures implementations where users are not explicitly told that their access to resources is metered against their limited budget. Hence, the overhead for users from budget expenditure decisions is eliminated. In B-RBAC2, the transparency of the cost of permissions can communicate the potential sensitivity of the resources and provide the information required to curb some of the inadvertent misuse discussed in Section 4.1. With regard to administrative cost, both B-RBAC1 and B-RBAC2 are equal to B-RBAC0.

5.1.4. B-RBAC3

B-RBAC3 can be considered as RBAC with escalation. It enables users to acquire the permissions that have not been preassigned to them, given the user has enough budget. Since neither the quantity of budget nor the cost of permissions is transparent there is no overhead on users. In fact, they may not be aware that they have been allocated a limited budget or that their access to resources is metered against their budget. Since B-RBAC3 provides an escalation capability, the escalation multiplier needs to be customised based on the roles, or users' attributes depending on the granularity required by the application. Therefore the administrative overhead of B-RBAC3 is more than B-RBAC0, B-RBAC1 and B-RBAC2. However, B-RBAC3 can also reduce administrative overhead: monitoring of users' budget enables administrator(s) to update their knowledge of users permission needs. For instance when no permission misuse is detected but a user's budget is exhausted before the designated period, this may flag to the administrator that insufficient budget has been allocated to the user. Furthermore, users' budget monitoring can also detect users' potential misuse of assigned permissions and escalations. Hence, the effort that the administrator needs to exert on audit is reduced.

6. Discussion of Future Works

We have identified three immediate areas for future work: First, perform an empirical study of the proposed model. Second, implementation of the B-RBAC on existing databases. Third, study the suitability of the B-RBAC model to enforce obligation policies.

6.1. Empirical Study

The B-RBAC model proposed in this paper provides a unified framework to address under-entitlement and over-entitlement. In the context of RBAC this means, when alternative roles exist to perform a task, users are given incentives to choose to activate cheaper roles, execute cheaper tasks or prefer to perform their jobs through their assigned permissions over escalations. However, it is not currently clear to what extent and under what conditions users' choices would actually be influenced through the allocation of a limited budget and price discrimination between tasks. Also, it is unclear how users will react to the extra tradeoff analysis they are faced with. More specifically, does the introduction of pay for access induce some users to refrain from accessing resources, hence reducing productivity? This behaviour may be due to users' fear that before the period lapses, they may need budget to perform other tasks. Furthermore, is it possible that the proposed B-RBAC model encourages some users to go on a spending spree to ensure their budget is consumed before the end of the budget allocation cycle?

In order to test the hypothesis that the B-RBAC model is effective in reducing under-entitlement and over-entitlement in a practical setting, an empirical evaluation of the model is needed.

6.2. Implementation

There are two main concerns regarding the practicality of implementing the B-RBAC model in a real world setting for conducting a field study. First, can the proposed model be adopted by an existing access control system? To this end, we conjecture that implementing the model as a proxy to an existing relational database is a logical starting point. There are three primary reasons for this. First, the B-RBAC model is based

on the standard RBAC model currently implemented in some relational database management systems [35] (e.g., Informix, Sybase and Oracle). Second, relational databases are widely used in healthcare. Third, the information stored in databases is structured and the operations that can be performed on the information are well defined. For example, in the simplest form all the cells in a database could be assigned a uniform cost and the cost of a `SELECT` query would be based on the number of cells it returns.

Second, how practical is the configuration of the proposed model for the administrator? It has been assumed that the administrator can determine the cost of tasks and the usage frequency of these tasks. Although in environments where resources are structured, operations are well defined and the number of roles are limited this assumption may hold, the extent of administrative overhead introduced needs to be evaluated in more general settings.

6.3. Obligation Enforcement

In general, obligation policies comprise those actions that should be performed by the user at some time in the future. For example, an obligation in a hospital may state that staff should terminate the application sessions they use to access patients' records when their task is completed. The existing access control models [36, 37] that support obligations assume that the criteria (e.g., time-frame, situation) for performing obligations can be predefined and their fulfilment is enforceable [38]. This assumption may not always hold. For example, it may not be practical to predict when a staff member has to close an application session when providing care for a patient. Further, automatically disabling the session after a predefined period of inactivity may interfere with the provision of care.

We consider the fulfilment of obligations by users to be an incentive problem. The question is how to structure incentives that motivate the user to honour the obligations. Given the B-RBAC model proposed in this paper, one approach is to assign cost to obligations in a similar fashion to the assignment of cost to permissions. The price of unfulfilled obligations are then charged to user's available budget. When obligations are honoured, the cost of the obligation is credited back to the user's budget. In this way, users are provided with incentives to fulfil their obligations while providing the flexibility for users to decide when to do so. We conjecture this is a novel approach to the enforcement of obligations, however further study is needed to determine the suitability of the B-RBAC model for enforcing obligations.

7. Conclusion

This paper proposed a novel Budget-aware Role Based Access Control model (B-RBAC) where instead of making access decisions based on constructs such as roles, the access decision is based on whether the user can afford the cost of the permission. It was shown how an RBAC policy can be employed as a reference point to discriminate between users to individualise permission costs and to allocate budget to users. The proposed approach enforces an explicit upper-bound on potential damage by users regardless of their assigned roles; it allows users to gain unassigned permissions; it promotes the alignment of users' incentives to observe the principle of least privilege, and integrates a monitoring and misuse detection mechanism into the access control model. The combination of these characteristics contributes to the optimality of access control decisions. Finally, the proposed model was analysed in terms of the potential overhead that its implementation can impose on the administrator or the users. Moreover, four primary sub-models were introduced to allow for the partial adoption of the model.

References

[1] F. Salim, J. Reid, U. Dulleck, E. Dawson, An approach to access control under uncertainty, in: Proceedings of Sixth International Conference on Availability, Reliability and Security, IEEE Computer Society, 2011, pp. 1–8.

[2] B. Schneier, Real-World Access Control, Online, viewed March 2010, Link: http://www.schneier.com/blog/archives/2009/09/real-world_acce.html (September 2009).

[3] S. L. Pfleeger, J. B. Predd, J. Hunker, C. Bulford, Insiders behaving badly: Addressing bad actors and their actions, Information Forensics and Security, IEEE Transactions on 5 (1) (2010) 169 –179.

[4] W. Baker, A. Hutton, C. D. Hylender, J. Pamula, C. Porter, M. Spitler, 2011 data breach investigations report, Tech. rep., Verizon (2011).

[5] S. Sinclair, S. W. Smith, S. Trudeau, M. E. Johnson, A. Portera, Information risk in financial institutions: Field study and research roadmap, in: FinanceCom, Lecture Notes in Business Information Processing, 2007, pp. 165–180.

[6] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, R. Chandramouli, Proposed nist standard for role-based access control, ACM Trans. Inf. Syst. Secur. 4 (3) (2001) 224–274.

[7] MITRE, Horizontal integration: Broader access models for realizing information dominance, Tech. Rep. JSR-04-132, MITRE Corporation Jason Program Office (2004).

[8] R. Anderson, T. Moore, The economics of information security, Science Magazine.

[9] H. Varian, Managing online security risks, Economic Science Column, The New York Times.
URL `http: //www.nytimes.com/library/financial/columns/060100econ-scene.html`

[10] L. Røstad, O. Edsberg, A study of access control requirements for healthcare systems based on audit trails from

access logs, in: Computer Security Applications Conference, 2006. ACSAC '06. 22nd Annual, 2006, pp. 175 –186. doi:10.1109/ACSAC.2006.8.

[11] L. Røstad, Ø. Nytrø, Access control and integration of health care systems: An experience report and future challenges, in: ARES, 2007, pp. 871–878.

[12] F. Salim, J. Reid, E. Dawson, Towards authorisation models for secure information sharing: A survey and research agenda, The ISC International Journal of Information Security (ISeCure) 2 (2010) 67–85.

[13] P.-C. Cheng, P. Rohatgi, C. Keser, P. A. Karger, G. M. Wagner, A. S. Reninger, Fuzzy multi-level security: An experiment on quantified risk-adaptive access control, in: IEEE Symposium on Security and Privacy, 2007, pp. 222–230.

[14] M. E. Johnson, Data hemorrhages in the health-care sector, in: Financial Cryptography, 2009, pp. 71–89.

[15] A. Appari, M. E. Johnson, Information security and privacy in healthcare: current state of research, International Journal of Internet and Enterprise Management 6 (4) (2010) 279–314.

[16] E. Celikel, M. Kantarcioglu, B. M. Thuraisingham, E. Bertino, A risk management approach to RBAC, in: Risk and Decision Analysis, Vol. 1, IOS Press, 2009, pp. 21–33.

[17] Y. Yemini, A. Dailianas, D. Florissi, G. Huberman, Marketnet: protecting access to information systems through financial market controls, Decision Support Systems 28 (1-2) (2000) 205–216.

[18] X. Zhao, M. E. Johnson, The value of escalation and incentives in managing information access, in: Managing Information Risk and the Economics of Security, Springer US, 2009, pp. 165–177.

[19] X. Zhao, M. E. Johnson, Access governance: Flexibility with escalation and audit, in: HICSS, 2010, pp. 1–13.

[20] F. Salim, J. Reid, U. Dulleck, E. Dawson, Towards a game theoretic approach to authorisation, in: Decision and Game Theory for Security (GameSec), Vol. 6442 of Lecture Notes in Computer Science, Springer/Heidelberg, 2010, pp. 208–219.

[21] D. E. Bell, L. J. L. Padula, Secure computer systems: Mathematical foundations, Tech. rep., The MITRE Corporation (March 1973).

[22] D. Liu, X. Wang, J. L. Camp, Mitigating inadvertent insider threats with incentives, in: Financial Cryptography and Data Security: 13th International Conference, FC 2009, Accra Beach, Barbados, February 23-26, 2009. Revised Selected Papers, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 1–16.

[23] D. Liu, L. J. Camp, X. Wang, L. Wang, Using budget-based access control to manage operational risks cuased by insiders, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications 1 (1) (2010) 29–45.

[24] N. Dimmock, A. Belokosztolszki, D. Eyers, J. Bacon, K. Moody, Using trust and risk in role-based access control policies, in: SACMAT '04: Proceedings of the ninth ACM symposium on Access control models and technologies, ACM, New York, NY, USA, 2004, pp. 156–162. doi:http://doi.acm.org/10.1145/990036.990062.

[25] N. Nissanke, E. J. Khayat, Risk based security analysis of permissions in RBAC, in: 2nd International Workshop on Security In Information Systems (WOSIS), 2004, pp. 332–341.

[26] Y. Yemini, A. Dailianas, D. Florissi, G. Huberman, Marketnet: market-based protection of information systems, in: Proceedings of the first international conference on Information and computation economies, ICE '98, ACM, New York, NY, USA, 1998, pp. 181–190. doi:10.1145/288994.289032.
URL http://doi.acm.org/10.1145/288994.289032

[27] D. F. Ferraiolo, D. Kuhn, Role Based Access Control, 15th National Computer Security Conference (1992) 554–563.

[28] Y. Masuda, S. Whang, Dynamic pricing for network service: Equilibrium and stability, Management Science 45 (6) (1999) pp. 857–869.
URL http://www.jstor.org/stable/2634775

[29] S. Bartsch, A calculus for the qualitative risk assessment of policy override authorization, in: Proceedings of the 3rd international conference on Security of information and networks, SIN '10, ACM, New York, NY, USA, 2010, pp. 62–70.

[30] F. Salim, N. P. Sheppard, R. Safavi-Naini, A rights management approach to securing data distribution in coalitions, in: Proceedings of the 4th International Conference on Network and System Security, IEEE Computer Society, 2010, pp. 560–567.

[31] X. Ma, R. Li, Z. Lu, Role mining based on weights, in: Proceeding of the 15th ACM symposium on Access control models and technologies, SACMAT '10, ACM, New York, NY, USA, 2010, pp. 65–74.

[32] BBC, Octuplets' hospital privacy fine, Online, viewed January 2012, Link: http://news.bbc.co.uk/2/hi/health/8155369.stm (July 2009).

[33] M. Bishop, S. Engle, S. Peisert, S. Whalen, C. Gates, Case studies of an insider framework, in: HICSS, 2009, pp. 1–10.

[34] A. C. Squicciarini, I. Paloscia, E. Bertino, Protecting databases from query flood attacks, in: ICDE, 2008, pp. 1358–1360.

[35] C. Ramaswamy, R. Sandhu, Role-based access control features in commercial database management systems., In Proceedings of the 21st NIST-NCSC National Conference on Information Systems Security (1998) 503–511Arlington, VA.

[36] Q. Ni, A. Trombetta, E. Bertino, J. Lobo, Privacy-aware role based access control, in: Proceedings of the 12th ACM symposium on Access control models and technologies, SACMAT '07, ACM, New York, NY, USA, 2007, pp. 41–50.

[37] M. Casassa Mont, Dealing with privacy obligations: Important aspects and technical approaches, in: S. Katsikas, J. Lopez, G. Pernul (Eds.), Trust and Privacy in Digital Business, Vol. 3184 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2004, pp. 120–131.

[38] P. Gama, P. Ferreira, Obligation policies: an enforcement platform, in: Policies for Distributed Systems and Networks, 2005. Sixth IEEE International Workshop on, 2005, pp. 203 – 212.