COMPUTERS & SECURITY 39 (2013) 237–254

Available online at www.sciencedirect.com

SciVerse ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

An adaptive risk management and access control framework to mitigate insider threats

*Nathalie Baracaldo**, *James Joshi*

University of Pittsburgh, 706A IS Building, 135 N. Bellefield Avenue, Pittsburgh, PA 15260, United States

ARTICLE INFO

Article history:
Received 23 June 2013
Accepted 1 August 2013

Keywords:
Trust
Risk management
Insider threat
Access control
Role based access control
Inference threat

ABSTRACT

Insider Attacks are one of the most dangerous threats organizations face today. An insider attack occurs when a person authorized to perform certain actions in an organization decides to abuse the trust, and harm the organization. These attacks may negatively impact the reputation of the organization, its productivity, and may produce losses in revenue and clients. Avoiding insider attacks is a daunting task. While it is necessary to provide privileges to employees so they can perform their jobs efficiently, providing too many privileges may backfire when users accidentally or intentionally abuse their privileges. Hence, finding a middle ground, where the necessary privileges are provided and malicious usage are avoided, is necessary. In this paper, we propose a framework that extends the role-based access control (RBAC) model by incorporating a risk assessment process, and the trust the system has on its users. Our framework adapts to suspicious changes in users' behavior by removing privileges when users' trust falls below a certain threshold. This threshold is computed based on a risk assessment process that includes the risk due to inference of unauthorized information. We use a Coloured-Petri net to detect inferences. We also redefine the existing role activation problem, and propose an algorithm that reduces the risk exposure. We propose a methodology to help administrators managing inference threats. We present experimental evaluation to validate our work.

1. Introduction

According to the Computer Crime and Security Survey, insider attacks accounted for 33% of the total incidents reported in 2010 (C. S. Institute, 2010). An insider attack is performed by people who are legitimately authorized in the system to perform certain tasks. The consequences of insider attacks may be devastating, and may include monetary losses, negative impact on the reputation, loss of customers, among others. According to Moore et al. (2008), the monetary losses due to insider attacks ranged from five hundred dollars to tens of million dollars, around 75% of the organizations had a negative impact to their business operations, and 28% experienced a negative impact to their reputations.

Some of these attacks could be avoided if access control systems were able to react when a user is performing actions that are not appropriate for their normal job functions. These attacks are typically preceded by *technical precursors* that include download and use of hacker tools, unauthorized access of customers' or coworkers' systems, system access after termination, inappropriate Internet access at work, and the setup or use of backdoor accounts (Moore et al., 2008). Hence, if the system is able to identify such inappropriate behaviors, it is possible to mitigate the misuse of permissions. For this reason, having an access control system integrated with an appropriate monitoring system can significantly help reduce potential insider attacks. The monitoring module can alert the access control module of a user's suspicious behavior, so

* *Corresponding author.*
E-mail addresses: nab62@pitt.edu (N. Baracaldo), jjoshi@pitt.edu (J. Joshi).
0167-4048/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cose.2013.08.001

that it can respond by restricting privileges to the suspicious users.

Although Role Based Access Control (RBAC) model has proved to be a promising approach for different types of organizations (Osborn and Sandhu, 2000), it is not able to cope with the changing behavior of the users. As long as a user is authorized for a role, the system grants him access. RBAC is appropriate in environments where users are well-behaved, where they can be trusted to perform actions according to their roles. Unfortunately, as the statistics show, insiders do perform attacks! Moreover, even if the users could be trusted, malware can be inadvertently installed and a user account be compromised. Thus, it is necessary to include the behavior of the users in the access control loop. The trust the system has on a user should be updated according to the user's behavior. When a user's behavior falls out of the expected pattern in a suspicious fashion, the trust the system has on him should be reduced. If a user is no longer trusted, the system should react by denying access to key resources. Several researchers have recognized the advantages of adding trust to access control models, e.g., Chakraborty and Ray (2006), Feng et al. (2008), Dimmock et al. (2004) and Nissanke and Khayat (2004). Adding trust to RBAC helps the system react to changes in the behavior of users. A trust threshold is typically used to limit access to resources based on their importance to the organization. However, existing approaches do not present a comprehensive analysis of the way in which trust thresholds should be assigned, nor do they include the separation of duty constraints or hybrid hierarchy. They also do not specify how to enforce such policies or reduce the risk exposure automatically.

In this paper, we propose a framework that integrates RBAC with the notion of *risk* and *trust*. Our trust-and-risk aware RBAC approach differs from existing ones in that it allows the integration of the risk assessment results in the access control policy. This permits the establishment of a threshold that represents the minimum amount of trust the system needs to have on a user in order to allow him to acquire the permissions associated with a particular role. Our model also supports cardinality and separation of duty constraints (Ahn and Sandhu, 2000), as well as the hybrid hierarchy (Sandhu, 1998). Including these components of RBAC allows us to inherit their well-known advantages. The main contributions of this paper are as follows:

1. We propose a model that includes risk and trust in RBAC systems that adapts to anomalous and suspicious changes in users' behavior.
2. We propose a comprehensive approach to calculate the risk values associated with permissions and roles. In particular, we introduce the notion of inference of unauthorized permissions when calculating the risk of activation of a set of roles. For this purpose, we present a formulation of a Coloured Petri-net (Jensen, 1987) to identify when a particular user may infer unauthorized permissions, and subsequently adjust the trust threshold required to activate needed roles.
3. We propose a refinement methodology to reduce the amount of information stored and the performance of the CP-net used to identify the risk exposure due to inference of unauthorized information.
4. We define an optimization problem to enforce the policy and reduce the risk exposure. To the best of our knowledge this is the first work that attempts to reduce the risk exposure in this way.
5. We present a role activation algorithm to solve the optimization problem, and evaluate its performance using well-formed policies and prove its correctness.
6. In order to improve the risk management process related to inference threats, we propose a simulation strategy that allows administrators to identify active inference threats before a policy is deployed. In addition, an administrator can determine the effect of adding a user-to-role assignment before he modifies the access control policy in the production system. This methodology reduces undesirable inference threats.

The rest of the paper is organized as follows. In Section 2, we overview RBAC, risk, trust and Coloured Petri-nets. In Section 3, we present the requirements of the system and an overview of the proposed framework. The details of the risk calculations are presented in Section 4. The formal definition of the role activation problem and the proposed algorithm is presented in Section 5. In Section 6, we present a CP-net based technique to find and manage the inference risk. We show the experimental results in Section 7. We present the related work in Section 8 and the conclusions in Section 9.

2. Preliminaries

In this section, we overview RBAC, risk, trust and Coloured Petri-nets.

2.1. RBAC, constraints and hybrid hierarchy

Our work is based on Role Based Access Control (RBAC) model (Ferraiolo et al., 2001), because of its benefits. It encompasses discretionary and mandatory access control models and supports organization or user-specific requirements. In addition, RBAC uses roles which are a natural abstraction for most organizations, and it provides organizations with economic benefits due to the reduction on the administration cost (Osborn and Sandhu, 2000).

In RBAC, *permissions* are assigned to *roles*, and roles are assigned to *users*. In order to obtain the permissions authorized for a role, users need to *activate* the role in a *session*. Sets U, R, and P represent the set of users, roles and permissions in the system, respectively. Separation of duty constraints (SoD) are used to avoid fraudulent activities within an organization by preventing a unique user from assuming two or more conflicting roles. There are two types of SoD constraints: Static (SSoD) and the Dynamic (DSoD). SSoD restricts the authorization of users to conflicting roles (Ahn and Sandhu, 2000). Each constraint is denoted as $\text{SSoD}(RS,k) \in \text{SSoD}$, where $RS \subseteq R$ with $2 \leq k \leq n$. This constraint states that a user can be *authorized* to at most $k-1$ roles in RS. Similarly, a DSoD constraint $\text{DSoD}(RS,k) \in \text{DSoD}$ states that a user can *activate* at most $k-1$ roles in RS simultaneously.

We consider two types of cardinality constraints in our model. An *activation cardinality constraint* restricts the number