EC- RBAC Model: secured access control model for Exigent Scenarios in Defense Systems.

Abdullah Sharaf Alghamdi
Software Engineering Department,
CCIS, King Saud University,
Riyadh, Saudi Arabia,
ghamdi@ksu.edu.sa

Syed Shah Amanullah Quadri
Software Engineering Department,
CCIS, King Saud University,
Riyadh, Saudi Arabia,
aquadri@ksu.edu.sa

Abstract:

With the advent of the exponential enterprise system growth in defense systems two areas have emerged as points of serious concerns viz., the abuse of privileges and control of the system as a whole with a super access that overrides the underlying access controls especially in exigent and emergency scenarios. In this paper we propose a RBAC model for exigency control (EC-RBAC) in emergency scenario which is expected to ensure a secured super access by a defense authority in the defense system, using Rational Software Modeler by IBM, a UML CASE tool. The EC-RBAC not only controls the whole system but also ensures that such access is competently secured.

Keywords: C4I Systems, Security, RBAC, UML.

1. INTRODUCTION

Role based access control has been enjoying the fame and acceptance as a one stop solution to the business, enterprise and e-commerce’s information security and access control requirements since its inception. The NIST model of RBAC incorporates reference model and functional specifications defined at the core. This is mainly intended for software engineers and product development manager for designing of access control features (1).

RBAC functionalizes using basic elements like users, roles, permission, operations and objects and is divided into four model components – Core RBAC, Hierarchical RBAC, Static Separation of Duties (SOD) and Dynamic Separation of duties (D-SOD) (1).

Core RBAC defines the element sets and is responsible for activation of user session. This model component is compulsory entity of any system that implements access control. However, the rest of the three model components are independent of each other and can be implemented separately. Hierarchical RBAC is a collection of mapping sets which map the inheritance of permission among the roles and is best represented using mathematical representations. Static Separation of Duties (SOD) component is key element to eradicate inconsistencies among the role permissions which in turn contributes to exclusivity to user assignment. Finally, the Dynamic Separation of Duties (D-SOD) concerns the precision of the relations of roles activated in the users’ session (2).

2. BACKGROUND

Duoqiang Wang et al., in their work [3] pointed out that the D-SOD policies do not enforce SOD policies as they don’t prevent users from activating mutually exclusive roles across multiple sessions. Thus, D-SOD has been assumed as requirement qualifying policy wherein users can carry out their steps in a sensitive task which requires to be verified for quantitative requirements check [3].

3. PROPOSED EC-RBAC MODEL

In this paper we present a RBAC mechanism that can efficiently handle the exigencies and its adverse effects on the defense system. Having noted that in exigent scenarios in defense system, there is bright window possibility of the authority over lapse and which in turn can provide access in the defense system and its critical levels of security. This is not necessary an allowed access therefore the authority and the roles can be wrongfully utilized. In the EC-RBAC model we assume that only a few yet trustworthy high ranking defense authorities are in function, which can be trusted with the whole command of the complete defense functionalities.
The immediate task at hand is to allow access to proper defense authority into the system and let the RBAC know that: (1) the authority logging on is to be treated with critical superiority, (2) all the other critical accesses in the defense system will be rolled back and overridden by the current superior access, (3) The superior authority logging in has raised an EXG flag to his logon and his logon is confirmed using a special security mechanism and (4) Once this authority has assumed control over the defense system no other logon is with the exigency flag or superiority is allowed to access the system.

In Fig. 1 we represent that, firstly the defense authority logs in to the defense system using a login which is attached with an exigent flag (Exg), this user login details are pre-available in the RBAC system for verification in a remote storage facility for security purposes. The instance the RBAC identifies the login as an exigent logon it will divert the user login to two different consoles of pass key repositories (Rep 1 and Rep 2) to acquire the passkeys which are stored in a remote facility again for security purposes invoked only by RBAC. The user will have to logon to the consoles using two sets of usernames and passkeys to acquire the single passkey to access the exigent permission repository of the RBAC.

Secondly, once the user has successfully acquired the passkeys the RBAC will be informed the same and a copy of the passkeys will be mirrored to main RBAC pass key repository. Then, RBAC will divert the user console back to main logon of the exigent permission repository. Once user logs on using the passkeys acquired, RBAC will verify the passkeys using its mirrored copy. If the keys match RBAC will proceed further else the login will be cancelled and the Exigent Logon details i.e., the passkeys and login id will be purged.

Further, In Fig. 2 we show that once the user has been successfully authenticated as an exigency super user by RBAC the user will be granted access to the Exigent Defense Permission Repository which will be dynamically instantiated, invoked and granted access to the Super user by RBAC.

The Exigent Defense Permission Repository (ExDPR) is not a static one, this ensure that no access in the regular functioning of the defense system will have control of the critical operations of the different verticals of the Defense system. The ExDPR will be dynamically created by RBAC by following the sequence of the operations as follows:

A. **Formation of Repository**
   From the permission repository copies of the RBAC all operations of all verticals of the defense system marked critical will be shortlisted and assigned to super user represented as:
   
   \[ \text{SupU} \rightarrow \text{SLPerm} \]

B. **Revoke Access**
   Revoke access of all the logins in the RBAC from the shortlisted operations.

C. **Alert Issuance**
   Issue system wide alert of exigent super user access in the system.

D. **ExDPR Formulation**
   Formulate dynamically the ExDPR repository with the shortlisted operations.

E. **Accessing ExDPR**
   Allow access to the ExDPR only by the Exigent Super User.

F. **Restriction further access**
   Disallow any further login to the system using exigent flag.
3.1 Specification and Verification

For the practical Specification, implementation and verification we are making use of UML as it is convenient to express constraints, sequential flow of functionality and also has the ability to depict the pre and post conditions along with Alternatives, Artifacts and Exceptions.

3.2 Dynamic ExDPR Formulation

The most important and challenging issue of the model is dynamic compilation of the ExDPR whose definition and design should be a result of sound Software Engineering techniques in general and the pivotal focus being the compilation process modeling in particular. The main focus should be modeling the auto-generation of views to compile the permission repository under emergency situation. This is to ensure that exigent permission repository is not susceptible to intruders and the dynamic compilation of ExDPR will ensure that the permission repository will exist in the system only after the Exigent Login has been authorized successfully. Thus, even if somehow the intruders using specialized intrusion techniques have compromised the remote logon mechanism, until the second level RBAC has not cleared the entry using passkeys the ExDPR will not be available in the system anywhere. Only a genuine exigent login acknowledged by second level RBAC will run the view in an executable format to assemble the critical operations from the permission repositories to formulate the ExDPR only after satisfying the preconditions as represented in figure 4.

3.3 Formalization of EC-RBAC Policies

The formalization of the EC-RBAC Policies needs to ensure the maintainability, efficiency and scalability of the overall functionality of the Defense system. This implies that the presence of the Exigent access in the system which overrides and controls all the critical functions of the system, should ensure that the critical functions in process are not forced into deadlocks. Thus, formalization of policies by combining authorization constraints to avoid above mentioned side effects. The formal verification techniques like theorem proofer, proof assistants and like thereof, though cumbersome yet can contribute.
great help to guarantee the avoidance of such issues up to a competent level of control and acceptance.

Fig. 4: State diagram of the ExDPR formulation process.

In Figure 4 we have shown the state diagram as an activity model reflecting the stepwise flow of the operations in the following steps:

A. Authorization
   Authorization of the exigent login including the RBAC execution at 2\textsuperscript{nd} level of the repository view generation process.

B. Formulation
   Formulation of ExDPR, repository acting on itself by acquiring core system functions critical to the defense system.

C. Revocation
   Revoking access on any current core system functions being acquired by the ExDPR.

D. Post security measures

D.1. Purging and Resetting View
   The current settings of the operational view which formulated the ExDPR to be purged and reset.

D.2. Purging and Resetting Pass Keys
   The current pass keys that granted access to the ExDPR generating view and resetting the RBAC pass keys.

The Operations carried out in the Post Security measures are ensure that in case of system compromise, the original view settings and passkeys are never available once the ExDPR has been generated and pass keys have been utilized to gain access to the core system functions of the defense architecture.

The whole scenario includes objects also known as participants in Unified Modeling Language, represented as life lines which communicate among each other. This can be via object referencing or via interface.

To demonstrate the participant wise understanding of the EC-RBAC model elaborating the focus of the detailed description of the each participant involved,
EC-RBAC can be represented using sequence diagram shown in figure 5.

IV. CONCLUSION AND FUTURE WORK

In this paper we presented a model for Exigency control in defense system which is capable of securing the critical operations of the defense system by a secured exigent super login contrivance and assigning the permissions to control those core functions as a super sole user.

This mechanism can help the defense system in two ways. (i) as a super control overriding the control on the core operations in the defense circumference and (ii) can also serve as a monitoring tool for the critical operations to ensure the confidentiality, integrity and security of the defense system.

The future work for this research is panoramic, in our forth coming works we intend to proceed on a modular basis. Initially, mainly focusing the policy designing of the contents of the ExDPR and then a secured mechanism, implementing the first and the second level of RBAC as discussed in the paper. At the secondary and tertiary levels we will look into implementation of the mechanism of the dynamic generation of the ExDPR repository via the available technologies that can aid the purpose.

REFERENCES


2) Zhichao Wen; Bo Zhou; Di Wu; "Three-Layers Role-Based Access Control Framework in Large Financial Web Systems," Computational Intelligence and Software Engineering, 2009. CISE


