Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 6, June 2021: 805- 812

Access Control Model Based On Role And Attribute For Secured Application.

Ashwini Parkar[a] , Madhuri N. Gedam[b] , Dr. Nazneen Ansari[c] , Dr. Vaibhav E. Narawade[d]

[a] Department of Computer Engineering,Shri L R Tiwari College Of Engineering,Thane, India, parkarashwini14@gmail
[b]Department of Information Technology, Shri L R Tiwari College Of Engineering, Thane, India,madhuri.gedam@gmail.com
[c]Department of Information Technology , St. Francis Institute Of Technology,Mumbai,India,nazneenansari@sfit.ac.in
[d]Department of Computer Engineering ,Ramrao Adik Institute Of Technology,Navi Mumbai, India, vnarawade@gmail.com

Abstract

Access control mechanism is used to secure an organization from insiders and intruders. RBAC and ABAC are the most popular models at present. Yet, they both deteriorate with pitfalls. RBAC has been widely adopted due to security advantages but difficult to meet dynamic access control requirements where ABAC provides more flexibility by introducing attributes. But system complexity due to the addition of attributes is the main drawback of ABAC. This paper combines benefits of these two models to come up with a role and attribute based RABAC model. RBAC manages static attributes and ABAC manages dynamic attributes which makes it more flexible, fine grained and user friendly. Additionally, we employ RABAC model to design a secured web application framework according to the characteristics of the model. This makes the application robust to hold out against SQL injection attacks..

Keywords: RBAC, ABAC, RABAC, SQLIA

1. Introduction

Now a days, the information security of an organization has become a crucial part for it to protect their data from internal and external attacks. Access control, one of the information security principles, is used to shield data from illegitimate access and also to make decision on who is able to access what [3]. Three major access control models gained much more attention: DAC, MAC and RBAC in early 70s. With evolution of information technology, the flaws in DAC and MAC gradually emerged as they could be able to meet the requirements of small applications only. RBAC model perpetuates the security and flexibility by introducing role between users and permissions. Further growth of network environment its need complicated. Access control decision policy became much dependent on the subjects(users) and objects(resources) attributes. It then analyzed that RBAC is inadequate for the contextual attributes requirement to allow access to a user. Briefly, RBAC leads to role-permission explosion problem [14]. Also deficient in the environment where a very large number of objects exists, even for bulk of users, huge data for a bigger organization [5].The access control mechanism is being developed in the direction of fine grained and system hierarchy in the current environment and based on the attributes of subject and object authorizations are implemented [10]. The ABAC concept is based on attributes preserved by users, resources and conditions of environment-rules under which access is granted or denied [15].ABAC uses dynamically changing attributes to make decisions. But attributes are meaningless until they are associated with a subject, object or relation [24]. Both Role Based and Attribute Based Access Controls suffers from some limitations and have some specific advantages. Shortly, it can be concluded the following problems in using classic RBAC or ABAC for large organizations:

- Enlargement of an organization increases departments, number of users and data which in turn makes the association between users' access to resources complicated. If we adopt only RBAC we get more redundant roles while in ABAC we need to consider all possible combinations of the attributes of subjects, objects and environmental conditions.
- Brings burden for large organization when position of user changes frequently like transfer, getting advancement in the post etc. It is difficult to implement only RBAC using fine grained access and for pure ABAC it is difficult to manages policies too.
- Demands for new services results in increase in creation of new roles with new policies. Hence due to "role explosion" it is inadmissible for organization to adopt either RBAC or ABAC as one and only mode of access control [7].

These two models can be combined to take advantage of their strengths. So, we put forward a combined approach called RABAC mechanism to maintain security, flexibility and simplicity in access privileges. RABAC is based on static relation among users and permissions and dynamic relation among users and permissions managed by RBAC and ABAC respectively. This paper proposes RABAC, the access control model based on role and attributes and employs to a web application which sustains its security from the malicious access [4]. Moreover, hostile users become successful in getting unconditional access to personal and secrete information due to the vulnerable web application .Access to the database of web application using SQL commands make it more vulnerable to SQL injection attacks[12]. So, the defence mechanism is developed and embedded for protecting the SQLI attacks on the model.

2. Related Work

Elisa Bertino, Ravi Sandhu [8] approached need for data security. Three necessities must be practically considered for all application environments: Confidentiality, Integrity and Availability. Protecting data from illegal access [9] and inadequacy remained in the application makes it vulnerable to profound attacks [4]. For securing database, different access control models have been implemented. These models are the collection of methods and components used to protect data [13]. Three main types of access control policies are MAC, DAC and RBAC. In MAC , policy administrator defines the usage and access policy of resources while DAC policies are defined by the end users. Owners sets the permission for others who have access to these resources and programs [20]. Definition of RBAC model given by [21] which assigns the users to the roles , roles to the permissions. So, permissions are assigned through the roles to users. RBAC works in two sections: the user to role assignment and the role to permission assignment. Literature [1] proposed access control framework to solve the distributed environment security issues through I-RBAC. Attributes associated with users, roles, actions and objects are used to activate or deactivate task dependent roles. RBAC model is not suitable in the situation where it is required to have dynamic and fine grained user-permission assignment. The relationship among users and roles and then among roles and permission recurrently changes [22]. We are permitted to correlate an attribute with each resource by grouping the objects. The group attribute is used to specify the permissions , where each permission assigns to a group of objects. The number of groups increase exponentially, as the number of resource attributes increase. This policy administration job complex because for each addition of a new object in the system it has to be associated with all groups to which it is be related [14]. Additionally, ABAC is an access control mechanism with role independent uses attributes based access control rules to achieve the dynamic and flexible access control which directly determines the relation among users and permissions. But it shortfalls a set of strict rules to establish security of user to permission assignment. The security analysis becomes very difficult. [16] proposed role-centric attribute-based access control model which divides access control decision process in to two sections- 1. using RBAC to define all the privileges or permissions granted to the user in the current session.2. using permission filtering policy (PFP) to excerpt the final available permissions. The access control granularity, flexibility and decision making these three properties of RABAC can be defined on the basis of literature[5]. Prajapati B., Gurucharansingh S.[11] implemented flexible attribute enriched role based access control model for generating XACML policy by making use of generalized role and experimented advantages such as fine grained, context aware, easily auditable etc.

Literature, [19] presents the database security testing method which includes how to detect potential input points of SQL injection, automatic generation of test cases and after that by running test cases to find database vulnerabilities. The SQL attack channels in the beginning itself can be stopped by the database security testing

SQL injection is an attack in which the code is exploited by altering back end SQL queries through input manipulation. Detection of the difference between the original SQL query and the SQL injected query through parameters modified by malicious users can be done using a collective method of query processing and removal of parameters. An evolution process from web application development to its security makes the application more robust and secured. Parameters removal and comparison with the original query improves the performance of the system[18]. An advanced method proposed by literature [17] for detecting and preventing SQLIA. Static and dynamic approaches are combined to build a new method. It consists of three phases. The first phase which is static, all augmented database tables include a record has only symbols like a $ sign. So, during design of database, it should be done. The second which is a dynamic phase proposes an algorithm that is created only once but for any query it is configured to be executed. Another approach is for detecting SQLIA is to search each token in predefined word of lexicon. This makes WHERE condition always True[2]. Authors Angshuman J., Priyam B., Dipendu M.[6] proposed an input-analysis based approach to automatically detect and prevent SQLIA. The objective is to identify the malicious user inputs which are mainly consists of either some kind of special symbols or keywords or combination of the both.

3. Access Control Models

A. RBAC Model:

Role-Based Access Control, RBAC, a famous model makes a complicated system administration simpler in the organisation as when a new employee joins there will be some particular responsibilities given to the user according to his role[3]. There are less chances of exploitation if users have no more privileges. Responsibilities can be parted into statically and dynamically. Dynamically it will be completed in time and statically it will be described by different conflicting roles. RBAC can be accessed by only authorized person hence popular due to its least rights precept. But sometimes it may cause large role creations than users. Now a days fine-grained results are needed which are not provided by RBAC[15]. It does not consider the environmental constraints like time, location, day etc. It maintains the relationship between user-role mapping and role- permission mapping. It is not feasible to change access permission without changing roles.

B. ABAC Model:

The issue of access control in operating environment can be solved using ABAC mechanism. As it can hold contextual parameters as access control parameters easily makes it more flexible. The corresponding rules must be applied using attributes of the requesting subject, requested object and environmental conditions. The user is allowed to execute an operation on the object when those attributes and operating environment conditions satisfy some policy sets[15].The major drawback of ABAC is complexity increases due to addition of attributes. Also, a greater number of access control rules make the security analysis of ABAC tough[10].

The comparison based on different features is given in TABLE 1.

Table 1 Analysis of RBAC and ABAC on different features[3]

FEATURES	RBAC	ABAC
Flexibility	No	Yes
Easiness	Yes	No
Dynamicity	No	Yes
Granularity	Low	High
Changing privileges	Complex	Simple
Role explosion problem	Yes	No

C. RABAC a combining approach :

As per the above discussion it could be seen that RBAC and ABAC both possesses strengths and drawbacks. An integration of these models defeats flaws and provides scalable, flexible and easier access control.

4. Proposed Model Design

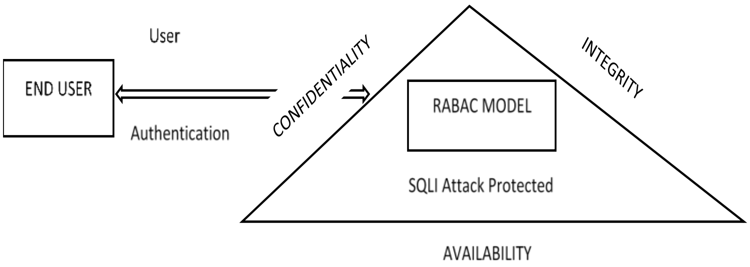

Fig. 1. Block Diagram of SQLIA protected RABAC Model

As shown in Fig.1.To perform the access control task, the following steps are needed:

- The End-User terminal requests access to the services or resources. User is authenticated through User_id and password. Passwords are encrypted using MD5 encryption algorithm[23].
- The authorized End-user is allowed to access the requested resources/services else it is denied.
- Then RABAC reserves U→R→P(RBAC) mapping and introduces ABAC into U→R and R→P mapping. Thus, roles are adjusted dynamically with the associated user and permission with the associated role by using ABAC rule.

A CIA triad block shows the security provided to the RABAC model which makes it robust to render any kind of external attack like SQL injection. Confidentiality refers to keep the data secret from illegal users. It is to control access to data to prevent illegitimate disclosure. Countermeasure taken to protect confidentiality is an authentication mechanism. Integrity assures that the data has not been tampered, so can be trusted and reliable. Strong authentication mechanisms and access controls protects data integrity. Availability ensures the legitimate users have timely, reliable access to resources when they need them.

a. RABAC Model

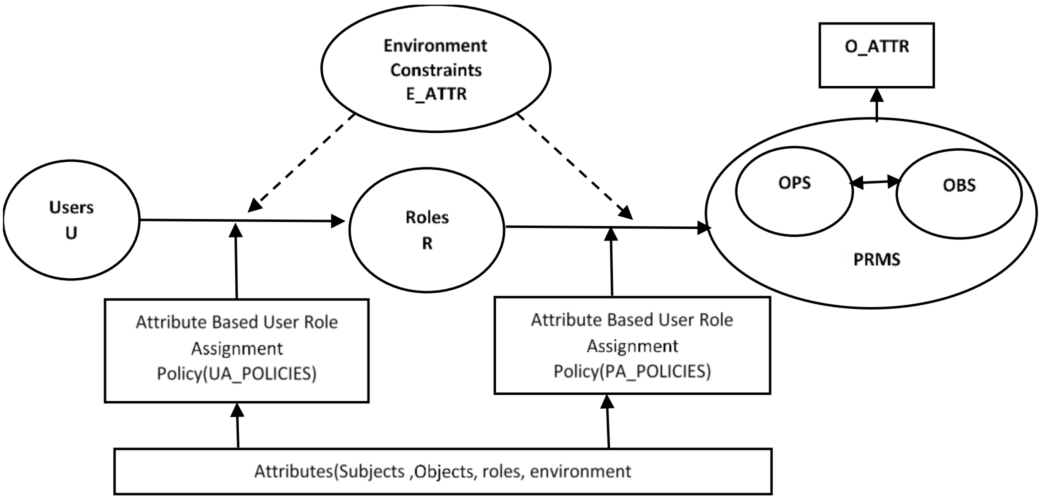
Environment Constraints E_ATTR
O_ATTR
Users U
Roles R
OPS
OBS
PRMS
Attribute Based User Role Assignment Policy(UA_POLICIES)
Attribute Based User Role Assignment Policy(PA_POLICIES)
Attributes(Subjects ,Objects, roles, environment

Fig 2. The framework of RABAC Model

As shown in Fig 2 the RBAC session is extended by proposing ABAC rule. The proposed model serves in the facets given below.

1. As per the literature [15] to U→R and R→P mapping. This provides greater flexibility for access control. Also, during a session establishment process for dynamic access control, constraints can be introduced . We employ the RABAC model to college management system which provides flexible roles settings and easier permission assessment.
2. Attribute is a property of a particular entity which represents a finite set of atomic values. For instance, branch (computer, mechanical, electronics). Subject attributes (U_ATTR) are name, gender, address. Object attributes (O_ATTR) define the properties of the resources like name, age etc. environment attributes (E_ATTR) considers external factors in which the access is needed like time, temperature which hold for multiple entities. Attributes can be static or dynamic. Static attribute values are mostly constant example designation, department

whereas dynamic attributes change frequently during the session (location, time, temperature).TABLE 2 gives different attribute types, example and values associated with it.

Table 2 Attribute table

Attribute type	Description	Attribute example	values
U_ATTR	User attribute	post_id	teaching, nonteaching, admin
O_ATTR	Object attribute	post_role	principal, teacher, hod
E_ATTR	Environment attribute	envt	time, date

3. UA_POLICIES and Roles here for instance we consider the student as a role who is allowed or disallowed to mark the attendance for the day before 10 AM and after 4 PM . UA_POLICIES may contain the following rules for role mapping shown in Fig.3.

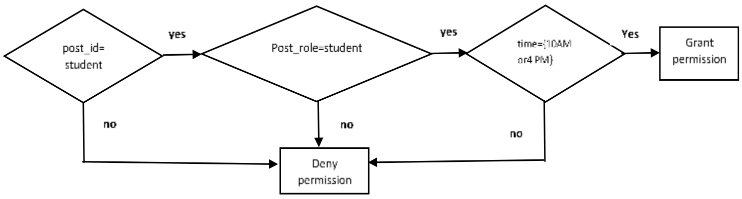

Fig. 3. UA_POLICIES and Roles

4. Many objects share common attribute values . Permissions (PRMS) is a group of objects which has common attributes for example students. Fine grained access control can be achieved by reducing number of roles associated permissions . Every permission is mate with one or more conditions. OBS are object expressions formed using attributes of object. So, each permission must be assessed to be truthful in order to grant that permission to the user. PA_Policies and permissions (PRMS) can increase the granularity of the model. Fig 4. Gives policies and permissions relationship, for example, a role-teacher has the equal operation permission (OPS) but distinct object access permission. Here the lecturer has distinct permissions in two distinct classes.

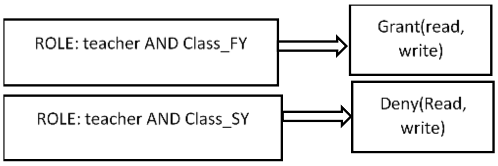

Fig. 4. PA_Policies and Permissions

These two subsets of attributes which are related to roles directly, make a direct relation with objects.

b. SQLI Attack: Detection and Prevention Techniques.

Vulnerabilities in a web application led to unrestrained access to confidential information for hostile users. As per OWASP 2010, SQLIA is the most likely and damaging[12]. Attacker inserts a malicious query to tamper data, or gains access to the back-end database. Hence it is needed to work on different SQLIA detection techniques and to apply various prevention techniques to secure an application from modification (loss of integrity) and interception (loss of confidentiality) attacks. TABLE 3 gives different methods used in SQLIA with examples.

Table 3 SQLIA types[2]

TECHNIQUE	METHOD with EXAMPLE.
Tautologies	A piece of harmful code is injected into more than one conditional statement. This always evaluated to be TRUE and the result is generated according to it. For instance, the record of the student where student_id is 101.SELECT * from student WHERE student_id = 101; This query looks like after injecting “1=1”, SELECT * from student WHERE student_id = 101 or 1=1;
Malformed Queries	The attacker injects false command into the database and obtains the error as a feedback message. It helps the attacker to extract the database structure by evaluating an error message.

Union Query	A UNION keyword is injected with some another SELECT query statement. This returns output of the 1st original query and 2nd injected query. The dataset is returned as a result from the database. SELECT * from acc WHERE id= 123 UNION select * from cr_card WHERE user=’admin’--’ and pass=’pass’.
Piggy-backed Query	These types of queries are extended after the original query by injecting additional query. These multiple SQL queries are then executed in a sequential manner. SELECT cust_info from acc WHERE login_id = ‘‘admin’’ AND pass = ‘123’ ; DELETE FROM acc WHERE Cust_Name = ’ABC’;
Interference	The attacker observes the response from the server to know about its structure after sending some data payloads. These are slower to execute but are harmful. It has two categories:1. Boolean: Attacker works on the results depend upon the type of query TRUE or FALSE. 2. Time-Based: It makes the database to wait for a period before responding to the query. The attacker checks the response time to evaluate a query TRUE or FALSE. So depending on the database he can work out if the payload he used returned TRUE or FALSE.
Stored Procedure	Database can be exploited using stored procedures by the malicious users. Exploitation can be done through injected text. For example, SELECT * FROM student WHERE studentname=' '; SHUTDOWN;--password=’ ’.

Table 4 Techniques to prevent SQLIA [17]

TECHNIQUE	METHOD
Using Prepared Statement	A parameter is used to insert a value into the database. Such SQL statements insert the values directly into database. In this way backend database is prevented from invalid SQL queries which can be harmful and not safe to the database.
Including stored procedure in the code	Store procedures are more adamant than the application code. Set of parameters are accepted which returns expected results. It assures that the important code is executed and performed similarly every time.
User Input validation	User input is supplied after the validity confirmation of it. So, the user input must be acceptable type, format or length etc. thus this method prevents data from targeting information sources like database.
Limiting access	An administrator account must not be connected to database unless you need to access the necessary piece of the database. The malicious user may gain access to the entire database. Hence it is essential to use an account with less privileges makes the system less harmful.
Data encryption	Data encryption prevents the malicious user from reading crucial data, Any change to the database would make it less affected.This can be implemented using salted hash function.

In the new model we concentrate on the above prevention techniques to secure the application from SQLI attack. Table 4 gives the various methods to withstand the application against SQLI attack which are proposed to use with RABAC model.

5. Discussion and further work

We proposed a web application to integrate features of both RBAC and ABAC. The model evaluates only those object expressions related with roles which are activated by a user in the current session. This decreases the number of rules to be evaluated. Based on roles, it evaluates a subset of policy rules. The attributes associated with subjects, objects and environment allow the request context to be considered in making access control decisions. Due to restricted access policies the resources remain protected from illegitimate access. RABAC model developed by merging best features of RBAC and ABAC models. It includes user-role and role-permission mapping policies. This model minimizes the number of roles and rules for access control without affecting the flexibility, provides more granularity and efficiency. We also presented some known SQLIA which highlights importance of database security. It shows that some simple prevention techniques prevent database from illegal access or such attacks which can be implemented in the proposed model for its security.

Future work includes to implement the proposed model and analyze the advantages exhibited by the proposed model. This application is also needed to protect from other types of attacks like Brute force attack, privilege abuse ,Denial of service etc.

References

[1] Rubina Ghazal, Ahmad Kamran Malik, Nauman Quadeer, Basit Raza, Ahmad Raza Shahid, Hani Alquhayz, “Intelligent Role-Based Access Control Model and Framework Using Semantic Business Roles in Multi-Domain Environments”, IEEE, 2020.

[2] Zar Chi Su Su Hlaing., Myo Khaing., “A Detection and Prevention Technique on SQL Injection Attacks.”, IEEE-2020

[3] Muhammad Umar A., Zhiguang Q., Zakaria., Safeer A., Pirah., Jalaluddin K., “The evaluation and comparative analysis of role based access control and attribute based access control model.” 2018, IEEE.

[4] Madhuri N. Gedam, B.B. Meshram, “Vulnerabilities & Attacks in SRS For Object-Oriented Software Development”. Proceedings of the World Congress On Engineering and Computer Science 2019, WCECS 2019, San Francisco, USA.

[5] Hui Qi, Xiaoquing Di, Jinqing Li, “Formal definition and analysis of access control model based on role and attribute”, Journal of Information Security and Applications 43(2018) 53-60, 2018,Elsevier.

[6] Angshuman J., Priyam B., Dipendu M., “Input-based Analysis Approach to Prevent SQL Injection Attacks.”, 2020 IEEE Region 10 Symposium (TENSYMP), 5-7 June 2020, Dhaka, Bangladesh.

[7] Sun L., Li Y., “RACAC: An Approach toward RBAC and ABAC combining access control.”, 2019 IEEE 5th International Conference on Computer and Communications.

[8] Elisa Bertino, Ravi Sandhu, “Database Security- Concepts, Approaches and Challenges”. IEEE transactions On dependable and secure computing, vol 2 No 1, January-March 2005.

[9] Sushil Jajodia, “Database Security and Privacy”, ACM Computing Surveys, Vol.28, No.1, 1996.

[10] Hui Q., Hongxin M., Xiaoqiang D., “Access control model based on role and attribute and its applications on space-ground integration networks.”, 2015 4th International conference on computer science and network technology (ICCSNT 2015).

[11] Prajapati B., Gurucharansingh S., “Flexible attribute enriched role based access control model”, 2017, IEEE, ICICIC-2017.

[12] Yeole A S., Meshram B B., “Analysis of Different Technique for Detection of SQL Injection”, International Conference and Workshop on Emerging Trends in Technology (ICWET 2011) – TCET, Mumbai, India

[13] Dindoliwala Vaishali J., Morena Rustom D., “Survey Of Access Control Models For its Applicability In Object Oriented Database Security.”, VNSGU Journal Of Science and Technology-V4(1)-2015-68.

[14] Mahmood Rajpoot, Qasim; Jensen, Christian D.; Krishnan, Ram, “Integrating Attributes into Role-Based Access Control”. Working Conference On Data and Applications Security and Privacy (2015). Springer. Lecture Notes in Computer Science, Vol. 9149.

[15] D.R. Kuhn, E.J. Coyne, T.R. Weil, “Adding Attributes to role based access control,” Computer, vol.43. no.6,2010; http://csrc.nist.gov/groups/SNS/rbac/documents/kuhn-coyne-weil-10.pdf.

[16] Xin Jin, Ram Krishnan, Ravi Sandhu., “A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC.”, IFIP International Federation For Information Processing 2012.

[17] Ahmad G., “A Hybrid Method for Detection and Prevention of SQL Injection Attacks.”, Computing Conference 2017 18-20 July 2017 | London, UK

[18] Rajashree A K., Swati S S., Vilas M T., “Detection of SQL Injection Attacks by Removing the Parameter Values of SQL Query.”, Proceedings of the Second International Conference on Inventive Systems and Control (ICISC 2018) IEEE Xplore Compliant.

[19] Yang Haixia, Nan Zhihong., “A Database Security Testing Scheme of Web Application.” Proceeding of 2009 4th International Conference on Computer Science & Education.

[20] Bokefode J D., Ubale S A., Apte S S., Modani D G., “Analysis of DAC MAC RBAC access control based Models for Security.” International Journal of Computer Applications (0975-8887) Volume 104-No.5, Oct.2014.

[21] Sandhu RS, Coyne EJ, Feinstein HL, Youman CE. Role-based access control models. Computer Feb. 1996;29(2):38-47.

[22] Cirio I, Cruz IF, Tamassia R. A role and attribute-based access control system using semantic web technologies. In: Proceedings of the 2007 OTM confederated international conference on the move to meaningful internet systems-volume part II; 2007.p. 1256-66.

[23] Ashwini Parkar, Madhuri Gedam, Dr. Nazneen Ansari, Dr. Shanthi Therese, “Performance Level Evaluation of Cryptographic Algorithms”, Lecture Notes in Networks and Systems volume 146, Intelligent Computing and Networking, proceedings of IC-ICN 2020 Springer, November 2020, pp. 157-167.

[24] Ed Coyne., Timothy R. Weil., “ABAC and RBAC scalable, flexible and auditable access management.”2013, IEEE