Design and Performance Evaluation of a Role-Based Access Control Model for Secure Web Applications

Mr. Vedant Pradipkumar Thorat,

Mr. Swaraj Sanjay Malve (Assistant Professor*)*

Department of Master of Computer Application,

JCEI's Jaihind Institute of Management and Research, Vadgaon Sahani

Email: vedantthorat938@gmail.com , swarajmalve79@gmail.com

Abstract: Web application security poses significant challenges to researchers and practitioners in computer science. One of the commonly used approaches is role-based access control (RBAC), where permissions are granted to users based on their role. However, even in RBAC, it is possible to encounter unauthorized access attempts, data exposure, and abuse of privileges. The purpose of this study is to design and evaluate an RBAC model which can be used in a secure web application. Specifically, we define users, roles, and permissions, and authorize access to protected resources and web APIs on this basis. We attempt to build a prototype web application which can be used to test how access control decisions are implemented for different roles, including administrators, staff, and users. Our evaluation methodology focuses on the accuracy of authorization decisions, prevention of unauthorized requests, response time, and overhead due to permission checking. We will also implement security test cases which verify that a given set of users cannot access unauthorized resources and methods. By demonstrating the role of access control in web application security, we hope to contribute to the knowledge in this area and provide practical insight into building such systems.

Key Words: Keywords: Role-Based Access Control, Web Application Security, Authorization, Access Management, User Permissions, API Security, Security Testing.

1. Introduction:

Nowadays, web applications are considered as an integral part of all spheres in human life such as education, healthcare, banking, business, and commerce. These applications comprise a significant amount of data regarding users' information that requires a high level of security to be protected from unauthorized access. Given the growing number of users and various web-based programs, it is critical to provide access control to maintain the confidentiality of each individual's data and authorized activities. In this regard, developing an efficient access control system is of great significance to ensure that users are granted only appropriate information and functions. The access control mechanism allows restricting and regulating users' access to elements, operations, or resources based on their credentials.

The conventional access control models applicable to web applications may impose some limitations in terms of managing a high number of users and their multiple roles. At the same time, the application's administrator should customize and configure permissions for each user who is engaged in numerous activities. Such a strategy creates a challenging task for developers due to the necessity of considering a number of constraints. From this perspective, Role-Based Access Control (RBAC) can be implemented as an efficient mechanism to manage permissions according to a user's role in an organization or a project. For instance, RBAC can be utilized in a web application by determining a specific set of permissions for Administrators, Staff, and Users. In this way, a web application's Administrator can control and monitor all users' activities, the staff managing department can perform only managerial tasks, and users can only operate on their data.

Based on this information, the current research aims to design a Role-Based Access Control model for secure web applications. This application allows developers to customize the roles and permissions of users in accordance with their duties. The proposed model can be utilized in web applications where

access control is the primary objective. At the same time, the research will involve the process of prototyping a web application to evaluate the performance of an access control model. Within the framework of the current study, a web application will be designed and utilized for testing purposes to observe the behavior of the developed access control model. By determining the set of permitted operations for each user role, the research will assess the authorization accuracy, prevention of illegal requests, response time, and permission overhead. This study will demonstrate how role and permission-based data management can be employed to ensure the efficiency of authorization in web applications. In addition, the current research will highlight recommendations for developers of web applications with a high number of users and multiple roles.

1.1 Problem statement:

The growing use of web applications in different fields raises the need to address the issue of restricting access to achieve information security. Such software allows several users to perform their activities and access particular data. However, this expediency may be compromised by a weak strategy for managing permissions. One of the common security problems in web apps is the inadequate or absent authorization mechanism. In this case, users may gain access to restricted parts of the application or violate the integrity of stored information. For example, unsecured access to a particular page is prohibited, but users can still change its contents. Another concern is the lack of authorization to modify APIs that process sensitive data and require appropriate roles to access. The issue of interest in this paper is to design and evaluate an efficient Role-Based Access Control model that will enforce permissions in a web application. The study will address the central question of how the developed model can ensure protection against unauthorized changes in the application and its databases A series of security tests will be conducted to evaluate how effectively the proposed system enforces access permissions and performs under various testing conditions.

1.2 Objectives:

- The main purpose of this research is to design and evaluate the role-based access control model that will enhance the security and management of users' permissions in web applications.
- To design the RBAC model: Develop a web application with a set of built-in access control elements that will provide the role-based access control to users.
- To implement the secure authorization: Enforce the server-side authorization to ensure that users can only access those application features and information that are allowed by their roles.
- To prevent unauthorized access: Test the implementation for unauthorized access attempts to examine if the application is capable of denying unauthorized requests and protecting sensitive information.
- To evaluate the authorization process: Perform a series of tests to examine if the application's authorization process prevents unauthorized users from performing prohibited actions.
- To evaluate the system's performance: Measure the time required for the application to authorize requests and ensure the system's performance is not impacted significantly.
- To improve permission management: Examine how the role-based access control model affects the overall management of permissions and if there is room for optimization.

1.3 Contributions:

The research was conducted to establish a Role-Based Access Control system for the security of web applications. Below are the main points advanced by the researchers :

Structured role and permission system: The research provides a structured model for the assignment of roles and permission in a web application, according to which users are provided with access rights depending on their roles.

Server-side authorization enforcement: The study suggests that the control of access rights should be done on a server-side to avoid user interference with restrictions set by the application.

Testing of unauthorized access: The research establishes tests for unauthorized access to ensure that users are only able to undertake roles and tasks that are arguable with their assigned roles.

Analysis of performance: The analysis of the performance of the web application under test conditions was done to establish the amount of computation involved with enforcing role-based access controls.

Security enhancement recommendations: Lastly, the research recommends ways through which weaknesses in enforcing permissions can be eliminated in a web application based on the experimental results.

2. Literature Review:

1. Role-Based Access Control Models The article by Sandhu, Ferraiolo, and Kuhn (2000) introduces NIST's model for Role-Based Access Control, which consolidated different aspects relating to RBAC in a unified theory. In particular, the authors describe the variations of trust and responsibility in role hierarchies and authorization constraints in role assignments and activations. The study offers a theoretical basis for developing systems with the controlled data access policy that relies on roles rather than individual permissions. The present research uses this idea to define the users, roles, permissions, and authorization relations in a web application.

2. Authorization Enforcement and Web Security The Authorization Cheat Sheet by the Open Worldwide Application Security Program (OWASP) Foundation offers guidelines on implementing the security measures for enforcing user permissions. The cheat sheet distinguishes the authorization controls from authentication mechanisms, clarifying that the two terms refer to different concepts. While authentication verifies the user's identity, authorization defines what this user is allowed to do in the application. The guidelines promote the secure coding practices by advising developers to always check the user's permissions on the server side. In addition, the OWASP report suggests that the absence of access control measures may lead to the unintended data exposure or privilege escalation. Therefore, the present study relies on this source to design a web application with the restricted access to its components.

2.1 Current Challenges:

They include the failure to appropriately set roles, granting too many privileges, and improper checks of access. Moreover, the application owner will be ineffective if they intend to manage access to data based solely on the client-side. The problem is that the app's users will be able to forge requests to the server to gain access to unauthorized information. Besides, as the application grows, the list of roles and users' privileges may become too large and require more thoughtful organization. The described issues can be solved by implementing an efficient server-side authorization mechanism for users' access to the web app. Such a system will provide high security and require fewer server resources if designed correctly.

2.2 Techniques, Strengths and Limitations:

Strengths: RBAC offers several benefits such as reduced administrative complexity, increased enforcement consistency throughout the system, elimination of redundant tasks, and mitigation of accidental disclosure of information when designed correctly. The control is particularly beneficial for web applications which provide various services to different users.

Limitations: The control may be associated with some shortcomings due to the need to configure roles and periodically re-evaluate permissions. The problem may occur when the privileges are incorrectly set or overly extensive leading to the generation of security weaknesses. Additionally, when scaling applications, managing a large number of roles may become challenging. The control is not suitable for all systems since there are cases when the context of a request is significant and should be considered. Therefore, the use of this technique requires appropriate security assessments and the implementation of other security measures.

3. Research Methodology:

The research method consists of developing, implementing, and testing an access control model of a web application. In particular, the plan is based on defining users' roles and permissions, implementing the chosen authorization model, and completing relevant performance and security tests.

3.1 Proposed system design There are three types of users in the system Administrator, Staff, and Regular Users. Each user type is associated with a set of allowed actions, ranging from full control over the system for the Administrator to the ability to access specific tools for Staff and Regular Users.

3.2 Implementation An application with Role-Based Access Control will be implemented in the web-based application, which will restrict all requests to the server-side authorization procedures. In particular, each user will gain access to the system only after confirming their roles and permissions.

3.3 Testing and Evaluation The authorization model will be subjected to security testing by implementing test cases with different user roles and checking whether unauthorized actions and access attempts are denied. In addition, the application's performance will be evaluated by measuring the response time and analyzing the impact of request verification on the system's overall performance. Specifically, testing will include attempts to access the application through direct addresses by different user roles.

3.4 The obtained results will be analyzed to evaluate the effectiveness of the implemented role-based authorization model and identify the components that require optimization. The findings will contribute to the development of recommendations for improving the performance of the web application.

3.1 Research Approach:

This research follows an implementation-based approach to evaluate the effectiveness of role-based access control. The study will focus on identifying users' roles and the permissions associated with them, implementing the authorization mechanism in the server-side code, and testing the approach to ensure that the determined threats are addressed. The checks will be done by testing different roles to ensure that only the authorized actions are permitted. The time taken when making requests will also be evaluated to establish the effect of the described security measures on performance. The gathered information will then be utilized to identify the security strengths and weaknesses of the role-based access control approach and propose enhancements to web application security.

3.2 Data Collection:

Data for this study will be gathered by carrying out a set of experiments on the proposed web application. Various tests will be done to observe the permission levels of different user categories including the Administrator, Staff, and Regular User. The results collected will consist of permitted and denied access requests, access control decisions, failed access attempts, and response time of the application. The information gathered will be used to examine the effectiveness of the proposed role-based access control model. The experiments will be carried out in a controlled environment by using set scenarios to observe and record data.

3.3 Metrics Selected for the Study: The proposed Role-Based Access Control model will be assessed using the following metrics:

- **Authorization Accuracy**: the ability of the application to correctly approve or deny user requests based on access rights.
- **Unauthorized Access Prevention**: preventing unauthorized users from gaining access to any data and performing any actions.
- **Response Time**: the duration that the application takes to process instructions and respond to requests.
- **System Performance**: general application behavior and its responsiveness to requests during the test, including the application's performance with a varying number of requests.
- **Permission Enforcement Consistency**: the uniform application of permissions to all aspects across all application components like pages, APIs, and operations.

3.4 Static Code Analysis:

Static code analysis will be applied to the application's codebase to evaluate possible security weaknesses, common programming defects, and inadequate authorization practices in the detected Role-Based Access Control. The codes will be subjected to role validation, permission checks, access control mechanisms, and sensitive information handling analyses. The assessment will be performed using tools like SonarQube and relevant programming language static code

analysis tools to reveal code quality and possible security issues. The results will be examined to prioritize code refinement and enhancement according to the application's security and quality needs.

3.5 Data Analysis:

The collected data during security and performance testing will be analyzed to assess the effectiveness of the suggested role-based access control model. The response-time measurements will also be analyzed to observe how the permission validation process impacts the performance of the application. The analysis will be visually represented using graphs and tables to illustrate the obtained results clearly. Additionally, the occurrence of improperly allowed or denied requests will be identified to reveal some common implementation issues and threats. Finally, the obtained data will be used to formulate a conclusion about the security and performance of the selected model and its ability to withstand attacks.

3.6 Results:

The results of this study will be used to evaluate the effectiveness of the proposed Role-Based Access Control (RBAC) model, which will be based on the adherence to accesses permissions, prevention of unauthorized operations, and the application's performance. The tests will be conducted using different scenarios to ensure that authorized users receive the necessary accesses and permissions after being authenticated and that those who do not have required roles will be prohibited from unauthorized entry and operation.

1. This section of the analysis will be based on the test scenarios, which will depict how the application will respond to requests from users with different roles, such as Administrator, Staff, and Regular User. It will be determined if allowed accesses will be granted and unauthorized operations will be denied. The results will be used to identify errors or omissions with permission assignment and overall authorization control.

2. **Security Evaluation:** The security assessment will be conducted to verify if denied accesses to specific pages, data, and operations will be appropriately restricted and if unauthorized interaction with the application will be prevented. This evaluation will be based on the results of testing unauthorized attempts to utilize restricted features and access data. The findings will provide justification for additional recommendations on improving the security performance of the application.

3. **Performance Evaluation:** The authorization time, as well as the overall performance of the tested application under different workloads, will be analyzed and compared to measure performance changes, which will provide a basis for concluding on the RBAC model efficiency.

> This table shows how permissions can be assigned to different user roles in web application.

Application operation	Admin	Staff	Regular User
View records	✓	✓	✓
Create records	✓	✓	✗
Edit records	✓	✓	✗
Delete records	✓	✗	✗
Manage user roles	✓	✗	✗
View own profile	✓	✓	✓

✓ = permitted; ✗ = denied. These are proposed permissions for your prototype, not measured results.

4. Discussion:

The results of the final tests will be analyzed to evaluate the efficiency of the created RBAC model in enforcing the permissions and protecting the restricted resources. The qualitative and quantitative results of the security tests will be used to identify the main strengths and weaknesses of the model. These findings will be used to discuss the efficiency of the selected model in different real-life cases and provide recommendations for improvement.

5. Conclusion:

Role-Based Access Control (RBAC) is an effective method for managing users' access to resources. The present research aims to design an RBAC model that will assign users specific roles and utilize server-side checking to ensure the proper use of permissions. The study will demonstrate how the authorization tool can help eliminate unauthorized access and enhance management functions in terms of permissions and roles. In addition, the research will also assess the security and performance aspects of the model to evaluate its effectiveness and identify potential vulnerabilities. By doing this study, one will understand how the application can be fortified against illegal intrusions and what weaknesses should be taken into consideration when developing software. Future studies may build on the findings of this research by elaborating on the best practices for creating resilient web applications. One of the potential directions of the investigation is the creation of a standard RBAC model that will also include other security measures, such as multifactor authentication, attribute-based access control, and intrusion detection mechanisms.

6. Future Work:

The suggested model of Role-Based Access Control can be improved in the following ways in the future:

1. The implementation of multiple factors of authentication to enhance account security;

2. The addition of the Role-Based Access Control to attribute-based access control;

3. Testing of the system's scalability;

4. The addition of the system to cloud-based and distributed web applications;

5. The introduction of novel automated techniques of auditing the system for permissions errors and weaknesses;

6. Monitoring of users' activities for detecting illegal or authorized access;

7. Enhanced management of permissions and roles to ensure the ongoing nature of duties and responsibilities.

7. References:

1. The NIST Model for Role-Based Access Control: Towards a Unified Standard. Proceedings of the Fifth ACM Workshop on Role-Based Access Control, 47–63. https://doi.org/10.1145/344287.344301
2. Barkley, J., Kuhn, D. R., Rosenthal, L. S., Skall, M., & Cincotta, A. V. (1998). Role-Based Access Control for the Web. CALS Expo International and 21st Century Commerce 1998: Global Business Solutions for the New Millennium. National Institute of Standards and Technology. https://www.nist.gov/publications/role-based-access-control-web
3. Ferraiolo, D. F., Barkley, J., & Kuhn, D. R. (1999). A Role-Based Access Control Model and Reference Implementation Within a Corporate Intranet. ACM Transactions on Information and System Security, 2(1), 34–64. https://doi.org/10.1145/300830.300834

4. Barkley, J., Cincotta, A. V., Ferraiolo, D. F., Gavrila, S. I., & Kuhn, D. R. (1997). Role Based Access Control for the World Wide Web. Proceedings of the 20th National Information Systems Security Conference, 331–340. National Institute of Standards and Technology.
5. Gavrila, S. I., & Barkley, J. (1998). Formal Specification for Role Based Access Control User/Role and Role/Role Relationship Management. Proceedings of the Third ACM Workshop on Role-Based Access Control, 81–90. https://doi.org/10.1145/286884.286902
6. Jansen, W. (1998). A Revised Model for Role-Based Access Control (NIST Interagency Report 6192). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.6192
7. OWASP Foundation. (n.d.). Authorization Cheat Sheet. OWASP Cheat Sheet Series. https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html
8. OWASP Foundation. (2021). A01:2021 – Broken Access Control. OWASP Top 10. https://owasp.org/Top10/2021/A01_2021-Broken_Access_Control/