EIMJ
المجلة الإلكترونية الشاملة متعددة التخصصات
العدد السادس والاربعون شهر (٤) ٢٠٢٢

Issues of the distributing the admin’s permissions on Databases

Majed M ALmnasouri
majd@taifedu.sa

Saleh Hadi Alshahrani
shshahrani@gaca.gov.sa

Hatim Muhammad alsalmi
s44181150@st.uqu.edu.sa

Abstract:

Access control models are an essential requirements that developed for securing nowadays, data systems organizations use the access control models, particularly to define who their operators are, what they can do, which resources they can reach, and which processes they can perform and use them to manage the whole process.

Introduction:

The system administrator plays a large role in keeping the business running smoothly and maintaining its compliance with the organization's data protection requirements. System administrators have full control over the database entries and exits of the facility's databases, and in many cases, its basic physical infrastructure. For this reason, give attention must be paid to the admin’s actions to protect the database.

Administrators need high privileges to perform their daily activities and run business in a smooth, high-quality way, but these privileges also make them a potential threat - and a target for hackers. The influence of administrators on distributed databases can be minimized by implementing practices such as the principle of least privilege and segregation of tasks. The concern for protecting administrator accounts always pays off.

We consider that automated predictive criteria improve security significantly if data access permissions are appropriately consistent across systems and the right granularity.

Literature Review

1- Administration:

Like authorization of users, many administrative functions, definition, and fulfillment of semantic integrity constraints on the information, and maintenance of schema information, must be performed for any system’s

database. There are many choices on the degree of centralization and transparency of those administrative functions in an exceedingly heterogeneous distributed database system [1].
DBA stands for Database Administrator, and this is the primary user with privilege assigned to database.

2- Admin privilege permission for Production Use:

Many databases rely on the relational model and support interactive and programmed access to the SQL Server or any Open Server application. SQL is the primary query search language. Multiple SQL statements may, however, be augmented with programming constructs like conditional logic (if, else, while etc.), procedure calls and parameters, functions, and local variables. These could also be combined into one database object called a stored procedure. A procedure is an independently protected object and (as within the case of a view) can override the protection of the tables it references [2]. Thus, it is possible to grant execution privileges to a procedure but disallow direct access to the information it references.

3- Role Limitations:

In order to use the roles optimally, the Database Adminastorators should understand their field limitations.

1. Generally, a user cannot obtain a Data Munipalated Language privilege to perform a Data Definition Language operation via role. The user should be explicitly granted the necessary object of privilege. For example, if user User1 creates a package that references table T in schema User2, the privilege to select table User2.T must be granted directly and not via a role.
2. A user creating a view on other user's tables cannot receive the privilege to select from the table over a role. It is conceptually crucial to understand that the privileges assigned to a role can only be associated with a user session. Those privileges cannot be inherited by any objects (views, stored procedures) owned by a user who happens to have been granted the role. Besides, if the user wants to grant others access to his view, then the view

creator must have granted the object privilege on the underlying tables "WITH GRANT OPTION" clause.

For instance, User1 has a table. User2 wants to create a view based on this table. User2 must be explicitly granted select on the table to create a view. If User2 wants to grant his view to User3, User1 has to grant also to him select "WITH GRANT OPTION".

3. equivalently, when creating procedures, the developer who is creating a procedure must have access to the implicit objects referenced in the body of the procedure. Although those privileges cannot be granted via a role, the right to execute the procedure can. Therefore, a developer who is executing a procedure requires only EXECUTE privilege on the procedure and does not require any access to the referenced objects. This reduces the number of privileges that need to be granted to developers and enhances database security. The same applies to a developer wanting to reference another's table. The referenced privilege must be explicitly granted to the developer [3].

a. Minimal privileges:

Nowadays, hackers, insiders, and money motivated attackers will try to exploit premium accounts to gain access to sensitive application data despite multi-layered protections, and this is the misuse of premium user accounts that have leverage when it comes to the database.

The threats caused by attackers who use premium accounts are often the most difficult to detect and the most comprehensive due to the wide access granted to such accounts. Therefore, Administrative Access Usage Control was ranked 8th from the list of SANS 20 Critical Security Controls V3.0, updated in August 2011. [4]

It is very essential than ever to put more security controls in a database. However, most customers cannot afford to allocate people to manage their database security and have few database administrators to manage it. Database consolidation can improve operational efficiencies and enable fewer people to manage the database. [5]

Create protection zones that prohibit strong DBA privileges from being misused by insiders, outside hackers, or malware. This is one way to reduce the risk of a superuser accessing sensitive application data in a database.

This is particularly important considering initiatives (such as outsourcing), and the use of modern IT infrastructures (such as cloud computing) that provide efficiencies through automatic database provisioning and standardization. In these environments, premium accounts have more access to sensitive and structured application data.

Oracle Database Vault is one example of a less privileged application for database administrators [5, 6]. It is intended to be able to build critical protection zones within the database, whether it is running inside or outside the cloud.

Oracle Database Vault imposes robust operational controls within an Oracle database. These new technologies give database administrators the power to eliminate collateral damage from attacks targeting privileged accounts. Does Oracle Database Vault provide the ability to enforce controls over who has performed operations within the database? And when was it implemented? Where was it implemented? How were these operations carried out? This application-level eliminates configuration skew and prevents unauthorized changes to the database, such as adding new database accounts and copying application tables. [5]

Referring to privileges as a security attribute that's required sure enough operations. Privileges do not seem to be unique and should be held by multiple entities. This effort's motivation is that the least privilege principle: Every program, and each user should use the smallest amount of privilege necessary to complete the task [8]. Moreover, applying the principle to application design limits unintended damage resulting from programming errors. The latter two approaches aren't applicable to several Unix-like operating systems because they're developed within the C language, which lacks security type or other protection implementation. Though some systems have begun to support non-executable stack pages, which prevent many stacks overflows from being exploitable, even this straightforward mechanism is not available for many Unix platforms.

Furthermore, the Unix security model is extremely coarse-grained. Process privileges are organized in an exceedingly flat tree. At the foundation of the tree is that the super-user. Its leaves are the end-users of the system. The superuser has access to each process, whereas users might not control processes of other users. Privileges associated with classification system access have finer granularity because they grant access to support the user's identity and group memberships. Generally, privileged operations are executed via system calls within the Unix kernel, which differentiates mainly between the super-user and every-one else. This leaves defensive programming, which attempts to stop errors by checking the stability, integrity of parameters, and data structures at implementation, run time, or compile. For instance, defensive programming prevents buffer overflows by checking that the buffer is large enough to carry the information that's being copied into it. Improved library interfaces such as strlcpy and strlcat help programmers avoid buffering overflows [7].

Nonetheless, for complex programs, it is still inevitable that programming errors remain. Furthermore, even the initial carefully written application are often suffering from third-party libraries and modules that haven't been developed with identical stringency. The likelihood of bugs is very high, and an adversary will attempt to use those bugs to realize special privileges. Whether or not the principle of least privilege has been followed, an adversary should gain those necessary privileges for the application to execute[8].

b. DBA Role Problems

The privileges of a traditional database administrator (DBA) are so high, which creates an internal security threat issue, as their role is not only that they can monitor and maintain data, tables, and indexes, but they can also add, delete and modify all of the above to the core of the content.[9]

In many cases, sometimes database administrators should not have access to business data for enterprise data security but can access some administrative processes. Restricting DBA privileges is very difficult, very restrictive hence dba cannot do much work, and ultimately the burden of SYSDBA falls.

DBA permission modification is a dangerous process that may cause the system to become unavailable. Therefore, it must restrict DBA operations.

On the other hand, DBA has the authority of audits. It can delete transaction logs or records after illegal business data is displayed, causing illegal transactions to be untraceable [9].

c. Debugging Site Errors for Admin and User In Multi-Tenent DB Environment

Often using traditional database systems methods, it may display general error pages to users instead of showing a detailed error message. This approach may be problematic when developing a website by displaying a general message without identifying the real problem for website administrators, unlike guest users.

In [10], authors illustrate a method was revealed to correct site errors in a multi-tenant database system. An example method involves receiving a request from a site administrator for access to an enhanced error message in which it determines whether the site administrator is authorized to view and modify the enhanced error message by evaluating exceptions related to the enhanced error message, the request, and the site administrator based on the consent decision to display a detailed error message.

d. Database Access Method and System for User Role Defined Access

Currently, in the paper [11], mentioned the Single Organization Model, which is an access control mechanism, is cumbersome when applied to multiple tenant databases. The reason for this is that the current access licensing systems have been adapted to department data in which the authority is granted to only display records for users all public data in a specific data set is displayed to users. However, the current multi-tasked "alternative solutions" licensing access subsystems cannot segment data at the enterprise or channel level.

The database system can be a segmentable database with a set of separate virtual databases. There can also be a unique database owner for each of the separate virtual databases and access to files in the database can be granted by the database owner only for the user to view.

The authors in paper [12] suggested a model that automatically calculates permissions and access levels for all users in distributed systems to the objects, which is a more efficient decision. The authors applied the model in different data from an organization like education, health, and public datasets. They analyzed the model and compared it with the traditional access models.

Database Access Control Models Performance

1- ABAC and CBAC Models:

Attribute-based access control (ABAC) and Content-Based Access Control (CBAC) Models

HIPAA law requires healthcare providers not to share healthcare information except under very strict rules, so the degree of protection in it is considered as high. However, in some facilities, users (researchers, doctors, nurses) are often granted broader access privileges, with retroactive scrutiny to detect and punish the abuse of privileges.

Therefore, users become significantly over-privileged due to the lack of registry-level content-based access control. Excessive privilege is somehow mitigated by enforcing RBAC or MLS so that users have basic permission to access the database, and retroactive scrutiny to punish the abuse of privileges. This creates a dilemma, if the volume of data is huge, it allows the user to access an unacceptable amount of records. Meanwhile, the damage inflicted after the audit cannot be retroactively reversed, since the suspected user has already made the mistake, and Instead of allowing users to gain privileges or require excessive human intervention. The suggested model can intelligently identify a smaller subset of records related to a user's task and grant access to only that subset. Attribute-based access control (ABAC) can be used to partially alleviate the problem.

For example, it can define access control based on a set of attributes of physicians and patients: A physician can access records of patients treated in his / her department.

However, attribute-based access control may not work for unstructured text (free text) content. Also, when database structure and attributes are very complex, it can be difficult to obtain closed-form expressions for ABAC policies.

presented a content-based access control model and implementation mechanisms. proposed a two-stage hybrid solution:

(١) The validity of the substance of a set of records is given to the user by the data administrator automatic or manually.

(٢) CBAC expands the core group and makes access provisions in accordance with specific CBAC rules at runtime.

2- Top-K CBAC Model:

In the basic CBAC model, content similarity is compared to a predefined threshold, and the user is granted access to all "similar records". To address the problem, top-K similarity can be used. Instead of setting a limit for record similarity points, an administrator can preset the number of data objects to grant access.

The author used the improved K-mean collector from Oracle Data Mining to preprocess records. Collecting and storing 2M records takes approximately 3 minutes per experiment.

For the top-K CBAC form, the authors attempted to combine blocking and labeling. As shown by the "T + B" bars in Figure 1 (a), query performance has been improved again, and the overall query evaluation time for top-K CBAC is now very acceptable - only slightly slower than "no-CBAC" in Figure 2 (B). The results confirm that CBAC is fast enough to be adopted in real-world applications.

In the trials, each summary commented on relevant topics. Each topic correlates with a "confidence factor" in the range from 0 to 1, which reflects the quality of the annotation. To maintain mark quality, align all with a non-parametric distribution and note that by setting the threshold to 0.2, 80% of the marks are removed (Pareto principle, also known as the 80-20 rule). The filtered topics are added to the new CLOB attribute (with CONTEXT indexing) in the tables. In the new subject space, noise has been removed in terms of distributed data [13].

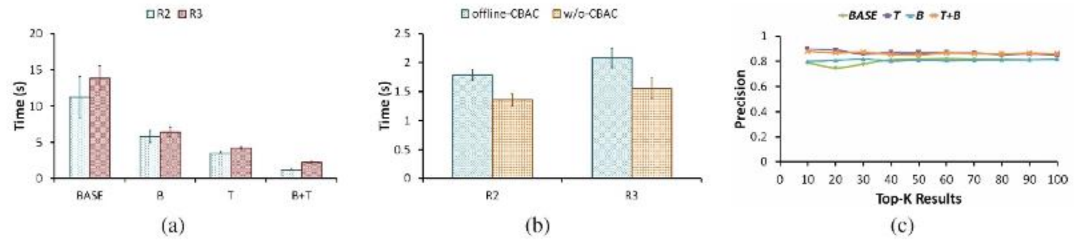


Figure 1 (a) Top-K CBAC query performance: (b) offline CBAC query performance: (c) soundness of CBAC enforcement[13]

Comparison between CEIAdmin and EnhancedRBAC

By comparing the results computed from CEIAdmin and EnhancedRBAC, they can measure the effectiveness of EnhancedRBAC using CEIAdmin indirectly as a standard. For this purpose, we compared the two models based on ‘read’ operation, ‘write’ operation, and both. For each comparison, they calculated the kappa value to measure the degree of agreement. An interesting aspect of the CEI project is that the ‘write’ permission is implanted within the ‘read’ permission (i.e., a role has the ‘write’ permission always has the ‘read’ permission). Therefore, the access permission for a specific channel can be defined again as three inconsistent outcomes: ‘no,’ ‘read-only,’ and ‘read & write’ access. To measure the agreement level based on this formulation of outcomes, we transformed the previous computation results and recalculated the kappa value. This study, with the three outcomes, provided a more accurate comparison between CEIAdmin and EnhancedRBAC.

Measuring the effectiveness

To estimate the effectiveness of EnhancedRBAC when used on the sample cases, to compare EnhancedRBAC with the gold-standard based on the previously described three outcomes (‘no’ access, ‘read-only’ access, and ‘read + write’ access). By distributing the results between the ‘no’ access and the other two outcomes, they evaluated the effectiveness of EnhancedRBAC based on the ‘read’ operation. By distributing the results between the ‘read + write’ access and the other two outcomes, they evaluated the effectiveness of EnhancedRBAC based on the ‘write’ operation. By adding up the results for both the ‘read’ and ‘write’ operations, they evaluated the overall effectiveness of EnhancedRBAC. For each comparison, we calculated sensitivity, specificity, and accuracy as specific measures. They performed the equivalent set of measurements on CEIAdmin.

Results

When formulated with three outcomes (‘no’ access, ‘read-only’ access, and ‘read + write’ access), EnhancedRBAC and CEIAdmin agreed on 4230 out of the 4576 study cases. With a kappa value of 0.80 (95% CI: 0.78–0.82), these two systems proved a high agreement level. When formulated with two outcomes (granting or denying access), the two models agreed on 4399 cases for the ‘read’ operation (kappa = 0.89, 95% CI: 0.88–0.91) and 4400 cases for the ‘write’ operation (kappa = 0.88, 95% CI: 0.86–0.90). Joining both, the two systems agreed on 8799 out of the total 9152 cases (kappa = 0.89, 95% CI: 0.88–0.90). Those comparisons have determined that EnhancedRBAC has achieved a high level of agreement with CEIAdmin. The detailed results are shown Table 3. When evaluated against the gold-standard, EnhancedRBAC had the right answer for 251 out of the 256 cases when the results were formulated with three outcomes (accuracy = 98%, 95% CI: 97– 100%). When changed to two outcomes and measured by the ‘read’ operation, EnhancedRBAC gained a sensitivity of 97% (95% CI: 94–99%), a specificity of 100% (95% CI: 100–100%), also an accuracy of 98% (95% CI: 96–100%). Based on two outcomes and ranked by the ‘write’ operation, EnhancedRBAC gained a sensitivity of 100% (95% CI: 100–100%), a specificity of 100% (95% CI: 100–100%), and an accuracy of 100% (95% CI: 100–100%). Combining both, EnhancedRBAC achieved a sensitivity of 98% (95% CI: 96–100%), a specificity of 100% (95% CI: 100–100%), and an accuracy of 99% (95% CI: 98–100%). As a comparison, they made the same set of measurements on CEIAdmin. The results showed that CEIAdmin had an overall efficacy of 76% (95% CI: 70–81%) when the results were formulated. It achieved sensitivities at the level of 100%, specificities in the range of 61–97%, and accuracies in the range of 76–99% when the results were formulated with two outcomes [14].

Comparison with 3 outcomes Kappa =0.80(95% CI: 0.78-0.82)		CEIAdmin System			
		Read+Write. No		Read Only.	Total
EnhancedRBAC Model	Read + Write	820		0 0	820
	Read Only	169		15 0	184
	No	7 3395		170	3572
	Total	996	185	3395	4576
Comparison with 2 outcomes ‘Read’ operation kappa =0.89(95% CI:0.88-0.91)		CEIAdmin System			
		Yes		No	Total
EnhancedRBAC Model	Yes	1004		0	1004
	No	177		3395	3572
	Total	1181		3395	4576
Comparison with 2 outcomes ‘Write’ operation kappa =0.88(95% CI:0.86-0.90)		CEIAdmin System			
		Yes		No	Total
EnhancedRBAC Model	Yes	820		0	820
	No	176		3580	3756
	Total	996		3580	4576
Comparison with 2 outcomes All-Operation kappa =0.89(95% CI:0.88-0.90)		CEIAdmin System			
		Yes		No	Total
EnhancedRBAC Model	Yes	1824		0	1824
	No	353		6975	7328
	Total	2177		6975	9152

Table 1 Comparisons between EnhancedRBAC and CEIAdmin.[14]

Model	Measurement with 3 outcomes	Measurement with 2 outcomes 'Read'	Measurement with 2 outcomes 'Write'	Measurement with 2 outcomes All Operations
EnhancedRBAC Model	98%	98%	100%	99%
CEIAdmin System	76%	99%	76%	87%

Table 2 Measuring the Accuracy of EnhancedRBAC and CEIAdmin

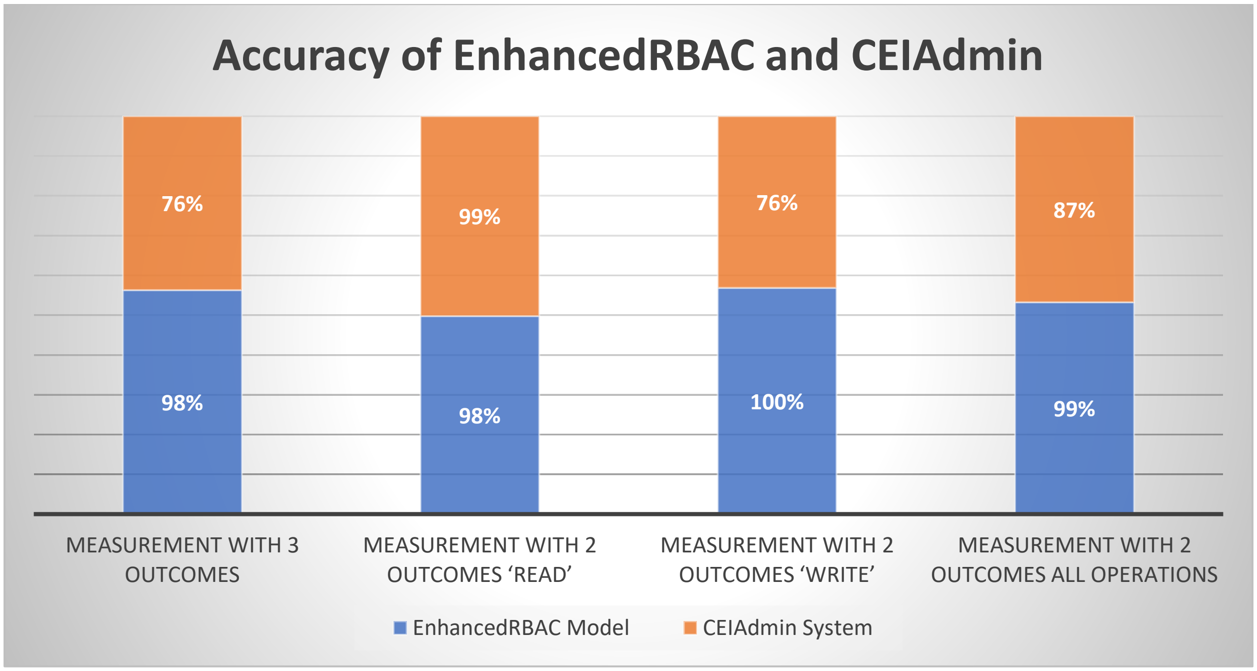


Figure 2 Accuracy of EnhancedRBAC and CEIAdmin

the A new scalable and expandable access control defined data as objects and classified users related security dimensions

Ordered dimensions:

in the ordered dimensions the values are ordered and accumulated. which means the value assigned to users content the below values also.

Unordered dimension

but in unordered dimensions, the values are not ordered and users may have multiple values in the same dimensions.

Permission Levels are various levels of related security settings for any object.
the permission level is Allowed means change the security settings of the object possible and can retrieve data of objects.
the permission level is Not Allowed means change the security settings of the object not possible.
Access Levels are various levels of related view or changed objects.
and if the access level is Read/Write, can change and show objects. and if has Read, cannot change objects but can show objects. write, can change objects but cannot show objects.
Covered, can check if the object exists or not but cannot display details of the object. if the user has access level is Not Allowed, the objects cannot show or change[12].

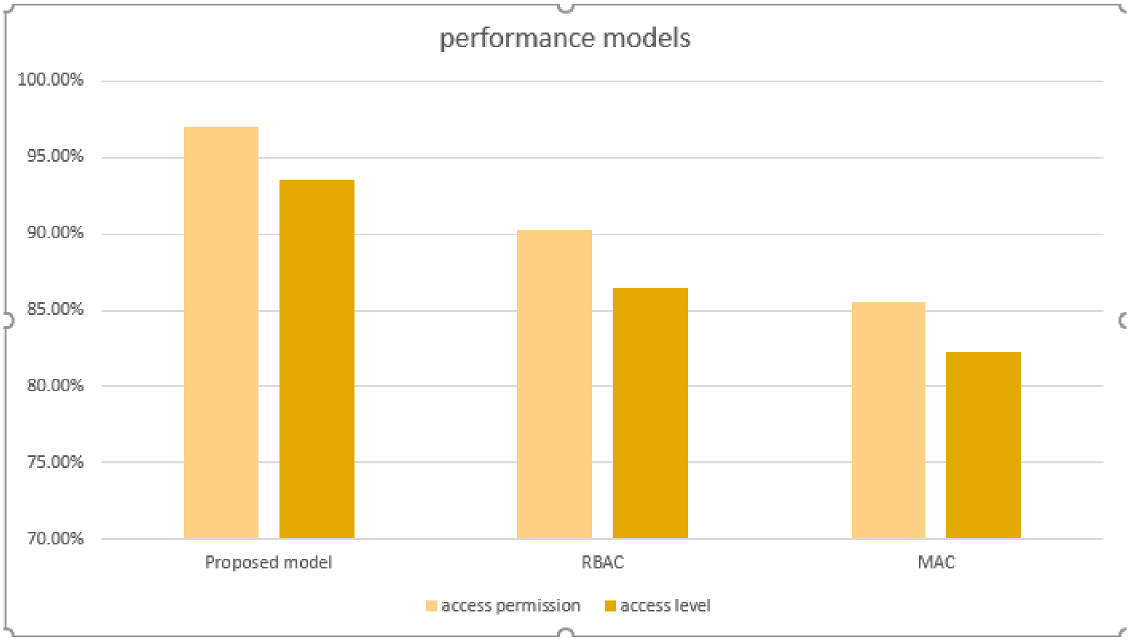
Experimental results

Compare the performance between the proposed model, RBAC, and MAC
by testing the proposed model to get 97.05% correct permission and 93.59% access level. RBAC model gets 90.22% correct permission and 86.46%.
MAC model gets 85.53% correct permission and 82.31% access level.
so the performance of the proposed model got better than RBAC and MAC performance between models.
Additional directions for future work include using various mechanisms to accelerate the access control decision-making process further and extend the model with a constant requirement to deactivate a role or revoke permission upon objection; based on sections 3.4 and 3.7 mentioned above, the EnhancedRBAC model has the highest access level accuracy with 98% when the A new scalable and expandable access control model have highest access permission accuracy with 97.05% as shown in table 4.

Models	Access permission	Access level

A new scalable and expandable access control	97.05%	93.59%
RBAC	90.22%	86.46%
MAC	85.53%	82.31%

Table 3 Measuring the Accuracy of RBAC and MAC and new scalable and expandable access control



Models	Access permission	Access level
A new scalable and expandable access control	97.05%	93.59%
RBAC	90.22%	86.46%
MAC	85.53%	82.31%
EnhancedRBAC	Null	98%
CEIAdmin System	Null	84.5%

Table 4 Comparison accuracy of Models

Conclusion:

In this paper, we recommended a better access control model that attributes may relate to users, objects, and the environment, allowing the request's context to be taken into account when making access control decisions. Unlike traditional RBAC approaches, the proposed model's permissions consist of processes and object expressions that allow control of a document's access to content. We show different stats to rate those models that can use different applications as per their requirements. In this paper, several models are reviewed according to the performance of access control to databases. After a comparison between them, we inferred that the EnhancedRBAC model has the highest access accuracy level. In contrast, a new scalable and expandable access control model has the highest accuracy of access permission.

References

[1] Sheth, A. P., & Larson, J. A. (1990). Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys (CSUR), 22(3), 183-236.

[2] Thomas, G., Thompson, G. R., Chung, C. W., Barkmeyer, E., Carter, F., Templeton, M., ... & Hartman, B. (1990). Heterogeneous distributed database systems for production use. ACM Computing Surveys (CSUR), 22(3), 237-266.

[3] Douglas Kilpatrick. Privman: A Library for Partitioning Applications. In Proceedings of the USENIX 2003 Annual Technical Conference, FREENIX track, June 2003.

[4] T. Baccam, "SANS Institute Product Review: Oracle Database Vault", (2011) August.

[5] Oracle White Paper—DBA Administrative Best Practices with Oracle Database Vault, (2010) December.

[6] H.-W. Fabry, "Database Vault: Enforcing Separation of Duties to Meet Regulatory Compliance Requirements", Proceedings of the 12th International IEEE EDOC Enterprise Computing

[7] Todd C. Miller and Theo de Raadt. strlcpy and strlcat -- Consistent, Safe, String Copy and Concatenation. In Proceedings of the 1999 USENIX Technical Conference, FREENIX track, June 1999.

[8] Jerome H. Saltzer and Michael D. Schroeder. The Protection of Information in Computer Systems. In *Proceedings of the IEEE 69*, number 9, pages 1278--1308, September 1975.

[9] Shen, M., Chen, M., Li, M., & Liu, L. (2013). Research of least privilege for database administrators. *International Journal of Database Theory and Application*, *6*(6), 39-50.

[10] Cinarkaya, B., Bulumulla, I. U., & Guest, R. (2013). *U.S. Patent No. 8,566,654*. Washington, DC: U.S. Patent and Trademark Office.

[11] Brodersen, K., Rothwein, T. M., Malden, M. S., Chen, M. J., & Annadata, A. (2004). *U.S. Patent No. 6,732,100*. Washington, DC: U.S. Patent and Trademark Office.

[12] Guclu, M., Bakir, C., & Hakkoymaz, V. (2020). A New Scalable and Expandable Access Control Model for Distributed Database Systems in Data Security. *Scientific Programming*, *2020*.

[13] Zeng, W., Yang, Y., & Luo, B. (2014, October). Content-based access control: Use data content to assist access control for large-scale content-centric databases. In 2014 IEEE International Conference on Big Data (Big Data) (pp. 701-710). IEEE.

[14] Le, X. H., Doll, T., Barbosu, M., Luque, A., & Wang, D. (2014). Evaluation of an Enhanced Role-Based Access Control model to manage information access in collaborative processes for a statewide clinical education program. Journal of biomedical informatics, 50, 184-195.