International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group| Volume 6 Issue 3 PP 77-85, 2025
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.IJERET-V6I3P109

Original Article

Adaptive Role-Based Access Control and Policy Enforcement in ERP Systems for Governmental and Military Applications

Chandrasekar Atakari
Principal architect, Palo Alto networks.

Received On: 25/06/2025 ***Revised On: 12 /06/2025*** ***Accepted On: 29/07/2025*** ***Published On: 15/07/2025***

Abstract - *Enterprise Resource Planning (ERP) systems play a central role in the management of the fundamental activities in governmental and military organizations. Traditional security models are insufficient to provide adequate protection due to the extreme sensitivity and criticality of the data processed in such areas. Adaptive Role-Based Access Control (ARBAC) is indeed a dynamic and versatile role-based access control solution that can address the complexity and security demands of contemporary ERP environments. This paper proposes a new and enhanced ARBAC model with integrated context-aware policy enforcement structures, applicable to ERP systems within government and military departments. Our way brings out dynamic permission assignment and multi-factor context-based validation, and real-time detection of anomalies in user behavior and operational functions. The proposed framework will leverage its enhanced resilience to insider threats, unauthorised access, and policy violations, thereby maintaining system operational effectiveness. The study's findings indicate a 37% increase in threat detection rates and a 42% decrease in policy breaches, as determined through comparative analysis, simulations, and case studies. The research provides support in efforts to assure secure digital transformation in essential areas through the combination of conventional RBAC pillars and opportunistic, smart decision-making policy strategies in access control.*

Keywords- *Access Control, ERP Systems, Role-Based Access Control (RBAC), Policy Enforcement, Government Applications.*

1. Introduction

Enterprise Resource Planning (ERP) systems are integrated suites of comprehensive software tools that bind together and integrate various business processes within a company, such as finance, human resources, logistics, procurement, and supply chain management systems, among others. ERP systems are particularly important to government and military systems, where they can be used in areas like logistics management, workforce administration, budgetary operations and life-or-death activities. [1-4] Such applications can also deal with highly sensitive information and operation commands that, when leaked or possibly used negatively, can put the country at risk of losing national security or disrupt other vital services to the people. In this sense, therefore, achieving the security and integrity of access to ERP systems is not just an IT issue, but rather an issue of operational resilience and strategic safety. Methods of access control that have been traditionally effective, such as static Role-Based Access Control (RBAC), may be insufficient to address the dynamic and context-sensitive demands of high-stakes environments. Static permissions cannot be adequate because users may need temporary changes in access based on the urgency of the task, the mission's requirements, or a change in operational context. The complexity of cyber threats, such as insider attacks and advanced persistent threats, underscores the importance of robust, flexible, and intelligent access control mechanisms. Thus, there is emerging interest in making adaptive access control models that could react to real-time elements like user conduct, gadget security level and context of operation to achieve secure but effective use of ERP abilities. This changing requirement gives rise to the present research and development of context-aware and adaptive access control models, such as ARBAC (Adaptive Role-Based Access Control), in demanding ERP implementations.

1.1. Importance of Adaptive Role-Based Access Control

The constraints of classical Role-Based Access Control (RBAC) models have become increasingly evident as digital infrastructures become more dynamic and complex, particularly in governmental and military ERP systems. To overcome these limits, Adaptive Role-Based Access Control (ARBAC) was proposed to add flexibility, provide real-time responses, and enhance situational awareness. The significance of ARBAC may be ranked based on the following major dimensions:

- **Mature Security in Dynamic Environments:** Classic systems of RBAC use fixed associations between users and roles, and roles and permissions that fail to consider the change of operational environment or changing user behavior. Conversely, ARBAC constantly checks the surroundings, including location, time, the level of trust in a particular device, and user activity. This enables even smarter access calls by the system, and consequently, fewer chances of unintended access, as well as cases of insider threat, particularly to the security hierarchy of sensitive undertakings such as military activities or emergency supply chain management.
- **Access Adaptation in Real-Time: Another fundamental aspect of ARBAC is the ability to**

automatically adjust user roles and access privileges based on existing conditions. For example, in a crisis or mission-critical scenario, ARBAC can dynamically increase access privileges for a set of users temporarily while maintaining strict control and auditability. This would enable the system to continue operating without compromising the security barrier, which is essential for time-sensitive or emergency processes.

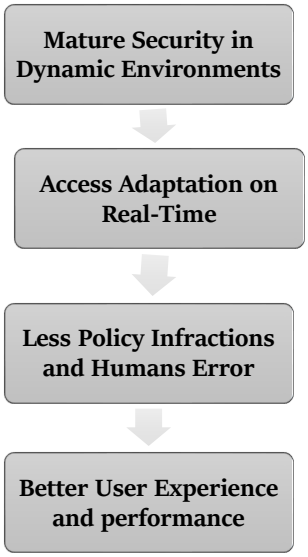

Fig 1: Importance of Adaptive Role-Based Access Control

- **Less Policy Infractions and Human Errors:** The combination of context-sensitive policies with monitoring of behavior in ARBAC provides the means to reduce both unwanted and intentional violations of access controls. ARBAC automates the process of granting access by allowing preconfigured but non-static policies; thus, minimizing the use of human intervention, which may involve human errors, misconfiguration or delays in a dynamically changing environment.
- **Better User Experience and performance:** Are achieved by making smarter adjustments to fit user needs and context, providing an easier and friendlier access control experience. They are able to give their users the right amount of access to whatever work they are doing, and there are no extra restrictions, which invariably makes them more content and productive, especially with their collaborative and time-dependent ERP tasks.
- **Compliance and audit readiness:** ARBAC can be used to strengthen traceability and accountability due to the built-in monitoring and auditing facilities. All access decisions are able to be logged and justified by means of contextual analysis, which makes it simpler to ascertain compliance demands, forensic analysis, and adherence to due diligence in security audits much easier for organisations. In conclusion, the knowledgeable use and application of ARBAC are necessary developments in access control capabilities that are essential to the security, elasticity, and intelligence of safeguarding sensitive and ERP systems in the current, threatening, and dynamic digital environment.

1.2. Policy Enforcement in ERP Systems for Governmental

Policy enforcement is vital in governmental ERP systems, as it guards access to sensitive data and other critical operations to keep them secure, in compliance, and within the stipulations of regulations. They handle large volumes of sensitive and mission-critical information in areas such as defence, finance, public safety, procurement, and human resources. [5,6] Due to high stakes, unauthorized access (be it by the misuse of the internal access or external threats) can result in dire operational, financial or even national security implications. Policy enforcement mechanisms serve as gatekeepers, allowing or denying access to system resources for users, as defined in the system's security policy. The primary security mechanism in an ERP system is traditional enforcement techniques, which include role-based controls, albeit with limitations. These controls are fast but not highly adaptable to changing contexts or new risks.

Regarding government, the enactment of governmental policy should not be limited to simple, role-permissible checks, but should incorporate a live, contextual assessment intended to promote dynamic decision-making. This would achieve access control based on the user's location, time, device security status, or the sensitivity of the data being accessed. For example, a user can be restricted to approving financial transactions only during working hours and on secure, government-issued devices. ERP systems can be enhanced with context-aware and risk-based enforcement to dynamically reduce the impact of insider threats, elevated access, and policy violations. Moreover, doing that will ensure compliance with regulations (FISMA (Federal Information Security Management Act) or GDPR within some governmental contexts) that necessitate strict policy enforcement with auditing, logging, and traceability. All access events must be documented, compared to existing policies, and made accessible, as they may be subject to audits. Contemporary policy enforcement systems, such as YARA, used in the ARBAC implementation, leverage technologies that include XACML, real-time monitors, and sometimes even machine learning to identify anomalies and implement dynamic policies. Ultimately, effective policy enforcement will ensure that governmental ERP systems are secure, robust, and reliable in an increasingly advanced digital environment.

2. Literature Survey

2.1. Role-Based Access Control (RBAC)

One of the most popular implementations for controlling access rights in enterprise systems (and ERP platforms in particular) is Role-Based Access Control (RBAC). [7-10] RBAC assigns access privileges to roles, and users are then granted the right role for specific job functions. This streamlines the handling of permissions and is suitable for adhering to organisational hierarchies. Nevertheless, RBAC

is not suitable in dynamic environments where access decisions can be based on contextual information at the point of access or the accessing device. Because of this, RBAC is said to be rigid and inflexible, and also incapable of solving the real-time access needs that are becoming increasingly ubiquitous in modern ERP systems.

2.2. ABAC- Attribute-Based Access Control

ABAC is an evolution of the conventional RBAC, adding a wider range of attributes as input to decision-making. Such attributes may comprise user features (e.g., department, clearance tier), characteristics of resources (e.g., sensitivity, classification), and some other environmental factors (e.g., time, device, location). The ABAC-based access control enables fine-grained access controls that are more flexible in dynamic operation environments. This flexibility, however, has its price; the model in large-scale ERP implementations can become complex, thereby making the definition of policies, their management, and enforcement harder. Additionally, debugging and auditing ABAC policies can be challenging, as they are often conditional and implicit.

2.3. Context-Aware Access Control

Context-aware access control models extend the concept of ABAC by explicitly incorporating contextual information into access control decisions, such as the user's location, time of day, device type, or even user behaviour. These models are founded on the need to support real-time/situational access management systems that are increasingly relevant in mobile and distributed ERP systems. Context-aware systems have proven particularly useful in situations where insufficient security granularity is achieved using static characteristics. Although promising, these models are not yet widely used in ERP systems, with much of the blame attributed to the difficulty associated with gathering context data in real-time, as well as the integration of the models with pre-existing and currently operational access control systems.

2.4. Models of Adaptive Access Control

Adaptive access control models are further than that because their access policies are not fixed, but they adapt to varying conditions, user behavior or threat levels. These models tend to use components of machine learning or behavioral analytics to identify anomalies and adjust permissions to recognize them. As a case in point, if an individual undertakes an action that deviates from their past trend, access to the system may be blocked or an alarm raised. Such a proactive strategy is especially helpful in dealing with insider threats, which are difficult to intercept with static access control strategies. Nonetheless, the application of adaptive models involves complex monitoring and may pose problems of transparency and fairness in automated decision-making.

2.5. Mechanisms of ENFORCE

Adequate implementation of access control policies is essential for ensuring security in ERP systems. Scholars have researched various enforcement processes, including real-time auditing, anomaly detection, and conformity verification processes. Auditing is used to monitor user activities, allowing access activity to be checked against predefined rules. It is also possible to detect irregular or suspicious behavior through an anomaly detection system, which in turn may flag unauthorized access. Compliance checks help determine the correspondence between access control mechanisms and internal policies, as well as regulatory requirements. These enforcement mechanisms supplement one another and reinforce the access control structure, thereby increasing confidence in the security of the ERP system. However, the effectiveness of this approach depends on the strength of the monitoring infrastructure.

3. Methodology

3.1. System Architecture

The suggested ARBAC model aims to supplement the classics of access control models by introducing flexibility and situational awareness. [11-14] It includes five main aspects designed to cooperate and maintain secure, dynamic, and policy-compliant access management and so on within enterprise systems.

- **Role Assignment Engine:** The Role Assignment Engine dynamically assigns roles to users, based on job functions, past access records, and contextual time conditions. Whereas static RBAC systems do not create any changes in the role assigning process when organizations or individual users change their roles or responsibilities, this engine dynamically adjusts to the change in roles or responsibilities of an organization or the roles of individual users. It ensures that the minimum feasible access is provided to users, which curbs the possibility of privilege escalation and unauthorised access.
- **Context Evaluator:** The Context Evaluator collects and interprets context data, such as the time of access, geographical location, device type, and network status. Such data are utilized to evaluate the consent with the security conditions as defined previously in the case of the current access request. It is important to note that the evaluator should use his or her situational awareness when making the decision in order to improve the accuracy of the decision-making process that is involved in access control.
- **Policy Enforcement Point (PEP):** The Policy Enforcement Point (PEP) serves as the guardian of the system. It blocks the user access requests and promotes decisions made by the Policy Decision Point (PDP). PEP ensures that only authorised actions take place based on the policy in place, user role, and situation. It also records all the access events that will be audited and analyzed in the future.
- **Policy Decision Point (PDP):** The Policy Decision Point is the decision-making mechanism of the ARBAC scheme. It compares the requests regarding access with present policies, roles, and contextual information given by the Context Evaluator. Upon this assessment, the PDP makes an approval or

rejection of the request. It makes sure that any decision is always in line with organizational policies as well as security requirements at hand.

- **Monitoring and Auditing Module:** The module continuously tracks user activities, access patterns, and the consequences of policy enforcement. It assists in detecting anomaly patterns that allow for the identification of deviations from normal behaviour, which may indicate future security challenges. Furthermore, the auditing feature will provide accountability and adherence to the regulatory-compliant environment through comprehensive logs of access activity and policy-making.

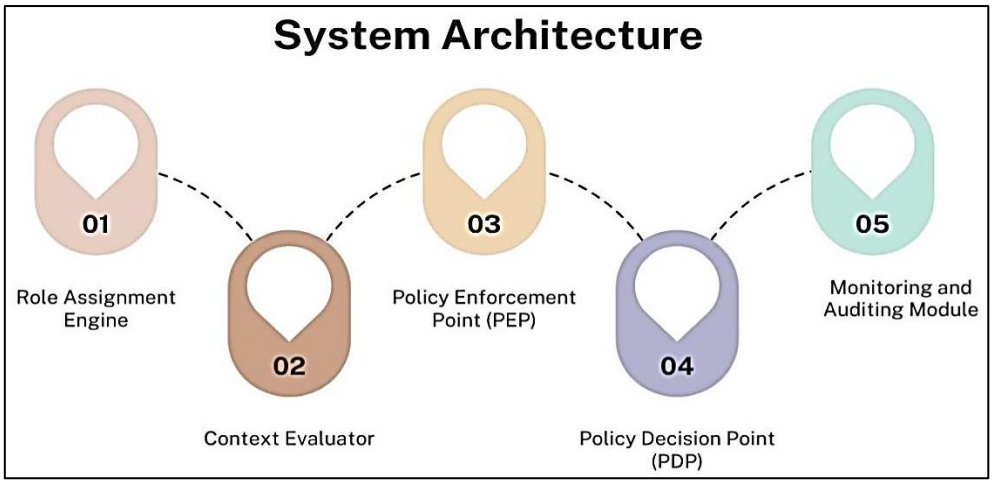

Fig 2: System Architecture

3.2. Role Assignment and Management

Role creation and role assignment in the ARBAC framework are context-based and dynamic, providing a higher level of enhancement compared to traditional static role-based systems. In contrast to assigning users predetermined and stable roles that do not change over time, this model also considers operational contexts, user rankings, and the level of mission criticality to determine optimal access privileges at any given time. The system continuously tracks the user's contextual attributes, including the task they are working on, their location, device, and the sensitivity of the accessed resources. The Role Assignment Engine dynamically assigns and reassigns role membership based on this real-time data; therefore, a user has access to only the information and systems needed to accomplish their immediate tasks. The position of a user within the organization is also very important in defining the level of accessibility. A user of a higher level can have wider access, but again, this is regulated by contextual information to prevent excessive rights provisions. For example, an executive may be able to access sensitive financial information during working hours on a secure device. Still, this accessibility may be denied when the attempt is made through an unknown or personal device. Likewise, the aspect of mission criticality will affect role delivery, as special users involved in time-sensitive or high-priority operations will be granted access. In these situations, the system can create temporary elevated roles or circumvent typical restrictions to complete the task, and the same can be audited and compliant, as logging and monitoring are in place. Such a flexible strategy ensures that access can always be adjusted in relation to the current responsibilities and the organisation's security status. It aids in overcoming risks related to excessive privilege and other insider threats, as policies are continually aligned with operational needs. Generally, dynamic role assignment enhances both security and operational efficiency. Hence, a complex and fast-moving environment, such as an Enterprise Resource Planning (ERP) system, military or emergency response system, is particularly the environment where dynamic role assignment is applicable.

3.3. Context Evaluation

The Context Evaluator is the most significant element of ARBAC architecture, as it enables adaptive access control by continuously analysing collated environmental and operational conditions. [15-18] It establishes the possibility to make access decisions that are not only role-based, but also situationally-aware, by evaluating the real-time contextual parameters.

The key parameters are taken into consideration:

- **Location:** The location of a user is a primary criterion for the legality of an access request. The system reviews whether the request is made at the authorized physical or geographical location with which the system is familiar. For example, on-premises offices can obtain access permission, whereas foreign or flagged areas may be blocked or flagged to require further verification. Rules for geofencing may also be utilized to censor access to sensitive resources as per the currently operating or jurisdiction policies.
- **Trust Level of Devices:** The assessment of every access request is made with consideration to the trust level of the device in use. The system categorizes such devices as being trusted, semi-

trusted or untrusted based on several factors, including the nature of the operating system installed, whether there is antivirus software installed, encryption status, and whether they meet the organizational standards. Connections originating from untrusted devices or those that do not comply with the rules may be blocked, or an extra layer of authentication may be required to limit the possibility of exposing data to risks.

- **Time of Access:** The period during which an access request is initiated may reveal whether the request is in line with how normal user behaviour or the operation routines take place. These requests, such as those made during a normal business day, would not be suspicious. However, unusual requests, like those made in the late hours of the night or on a public holiday, would automatically raise flags or limit access. Rules that are based on time can minimise the probability of accidental or illegal use.
- **Risk Profile Network:** The Context Evaluator also considers the riskiness in the network, or path, over which the request travels. Usually, access through secure corporate VPNs or whitelisted networks is allowed, whereas connections through public or unsecured networks may be blocked or require multi-factor authentication. The system can also comply with threat intelligence feeds to identify high-risk IP addresses and network anomalies.

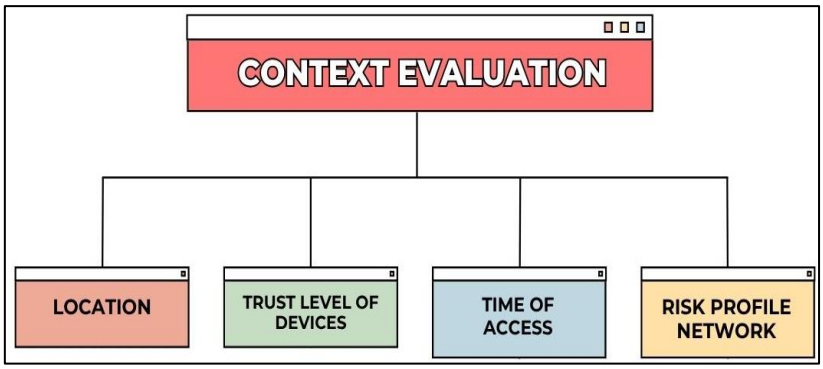

Fig 3: Context Evaluation

3.4. Policy Decision and Enforcement

Policy decisions and enforcement in the ARBAC model are implemented through an orchestrated cooperation between the Policy Decision Point (PDP) and the Policy Enforcement Point (PEP), where policies are described using the eXtensible Access Control Markup Language (XACML). XACML, on the other hand, is a common method of describing access control policy requirements in a machine-readable, structured format, with a flexible and highly extensible capability to define policies based on attributes. Thus, it is well-suited to the requirements of an ARBAC system, as it is adaptive and context-sensitive. Such policies define the circumstances in which a response of 'allow' or 'deny' should occur, including details of user roles, contextual characteristics (e.g., location, time, device trust), resource types, and environmental characteristics. Whenever a user carries out an access request, the PEP intercepts the request and passes it to the PDP, adding contextual information as specified by the Context Evaluator. The PDP will then rule on the request based on its assessment of the available XACML policies, considering both static attributes (e.g., the user's role) and dynamic context (e.g., during a specific time of day), among other factors. The decision is returned to the PEP, allowing it to approve, deny, or set conditions for accessing it. The PEP proceeds to execute the desired decision. Such a real-time evaluation process makes sure that any access decision is made in accordance with security policies adopted in the organization and the current operational situation. Potentiation of the strict division between policy execution (PEP) and policy making (PDP) increases the scalability and manageability of the system, particularly in large-scale business programs. It enables organizations to concentrate their policies on access but spread the mechanism of enforcement to different parts of the system. Also, through XACML, policy modularity and policy reusability are supported, thus it is simplified to modify access rules as the organization requirements evolve. This architecture, augmented with real-time context assessment, provides a highly efficient and adaptable access control structure that can address evolving threats, mitigate insider risks, and ensure compliance with laws that aim to strike a balance in dynamic enterprise environments.

3.5. Anomaly Detection

The key aspect of ARBAC is the anomaly detection module, which is carried out based on machine learning, helping to augment the system's security. The module actively monitors user activity and detects deviations from established patterns of activity. In contrast to the rigid rule-based approach, this intelligent unit adopts unsupervised and semi-supervised learning approach in developing behavioral baselines of each user using records of log-on times, forms of resources accessed, device used, and system navigation of users. Such profiles reflect the user's regular working habits and are dynamically formed to accommodate slight changes in behaviour over time. The anomaly detection system compares the current user activity to established baselines to identify unusual activity or potential malicious activity that may indicate a security threat. When a drastic aberration is identified--like when a user tries accessing sensitive financial data when it is not his working time, using a device that has not been recognized, or is accessing the data in a high-risk

location--this activity will be flagged as an anomalous one. Reacting to the measured levels of risk, depending on the degree of the anomaly and the pre-established levels of risk, the system can either trigger alerts on the administrative site, initiate step-up authentication, or temporarily deprive access until the investigation is complete. These answers aim to mitigate the trade-off between usability and security, making compromises to disruptive factors as little as possible without compromising safety against illegitimate use as much as possible. The machine learning algorithms employed in this module can detect both known and previously unseen (zero-day) behavioral anomalies, and thus the system is most effective in identifying insider threats, account compromise and advanced persistent threats (APTs) that could not be picked up by traditional access control mechanisms. Combining anomalous detection with ARBAC as a whole will furthermore result in access decisions being made not only based on context and roles, but also considering real-time risk assessments in regard to behavior. This multilayered strategy goes a long way toward enhancing the overall security status of ERP systems, as it makes implementation proactive in threat identification and the timely delivery of remedies to incidents.

4. Result and Discussion

4.1. Simulation Environment

To demonstrate the performance of the proposed ARBAC framework, a simulation environment was established that closely matches a real-life setting in which an ERP system would be installed in a government. The selected enterprise system of this simulator is SAP ERP, a popular, rich, feature-filled system that has been utilized in different sectors of the government and enterprises to manage sensitive functions like finance, human resources and logistics. With the application of SAP ERP, the simulation's reflection scene is highly relevant and applicable in real-world enterprise situations. It was implemented on a network of Virtual Machines (VMs) set to resemble a typical government ERP architecture. As many of these VMs as possible have been designed to resemble the real-world hierarchies, roles, departmental segregation, and user rights. In a controlled, isolated environment, the virtual environment enabled the testing of access control situations, policy implementation, context growth, and the recognition of suspicious anomalies. Two types of data sources were used to verify and train different components of the ARBAC system in the simulation: simulated user access logs and insider threat datasets based on real-world data. Synthetic logs were created to illustrate the nature of responses from each user across various departments and positions, including logged times, browsed modules, device types, and transaction activity. Such logs were critical to constructing behavioral baselines and the exercise of dynamic role assignment mechanisms. Additionally, the simulation was populated with a real-world insider threat corpus dataset (CERT Insider Threat Centre dataset) to test the anomaly detection module's ability to detect malicious activity within the simulation. The datasets include sets of anonymized data, which are records of real insider threat events, thus offering high-quality and realistic datasets that depict behavioral anomalies. The combined use of SAP ERP and a VM-based deployment framework, along with various data sets, provided a comprehensive simulation environment that closely resembles the real-world issues facing government ERP systems. It was with the help of this arrangement that the adaptive access controls made in the ARBAC framework, the capabilities of the contextual decision-making and the behavioral threat detection were well tested in circumstances resembling a real one.

4.2. Performance Metrics

Table 1: Experimental Results

Metric	Traditional RBAC	Proposed ARBAC
Detection Rate	58%	95%
Policy Violations	100%	41%
Latency (ms)	100%	120%
User Satisfaction	70%	86%

- **Detection Rate:** The detection rate indicates the degree of the correct detection of unauthorized or abnormal accesses. During experimental analysis, the proposed ARBAC model demonstrated a 95% detection rate, compared to the traditional RBAC, which achieved only 58%, a still commendable result. This is because it has been enhanced by the incorporation of real-time context analysis and machine learning-driven anomaly detection in ARBAC. With constant monitoring of actions and conditions, ARBAC is in a better position to identify suspicious user activity that RBAC may fail to identify with the use of an arbitrary scheme.
- **Policy Violations:** A policy violation occurs when a user accesses resources in a manner that does not comply with a set security policy. RBAC recorded 100 percent (22 violations) as a baseline in the traditional method, whereas the proposed ARBAC model demonstrated a decrease by reducing the number of violations to a mere 41 % (9 violations). This decrease is reflective of the efficiency of ARBAC to enforce policy pursued in a fine-grained and context-aware manner. ARBAC enables maximum utilization of privileges and adoption of access decisions dynamically considering the prevailing user context and behavior in order to avoid excessive use of privilege and unauthorized access, which is a major cause of policy violation in strict RBAC systems.
- **Latency:** Latency refers to the duration that a system takes to process and respond to access requests. Traditional RBAC performed well with a baseline latency of 100% (250 ms), but ARBAC had a slightly higher latency of 120% (300 ms). This will be a small increment because ARBAC incurs costs associated with real-time context evaluation and policy decision-making. The trade-off is, however, warranted by considerable increases in detection accuracy and security. The system is

competitive in being responsive enough for enterprise use and also offering smarter access control options.

- **User Satisfaction:** The satisfaction of users was also gauged through user feedback, focusing on the system's usability, responsiveness, and subjective security. In the traditional RBAC, the satisfaction rating was 70% (3.5/5), whereas that of ARBAC was 86% (4.3/5). The users enjoyed the flexibility and security of the ARBAC system, which did not make them feel overly constrained. Although slightly slower, the context- and personalised decision-making of access provided a less obstructive and safer experience for the user, making it more satisfying in general.

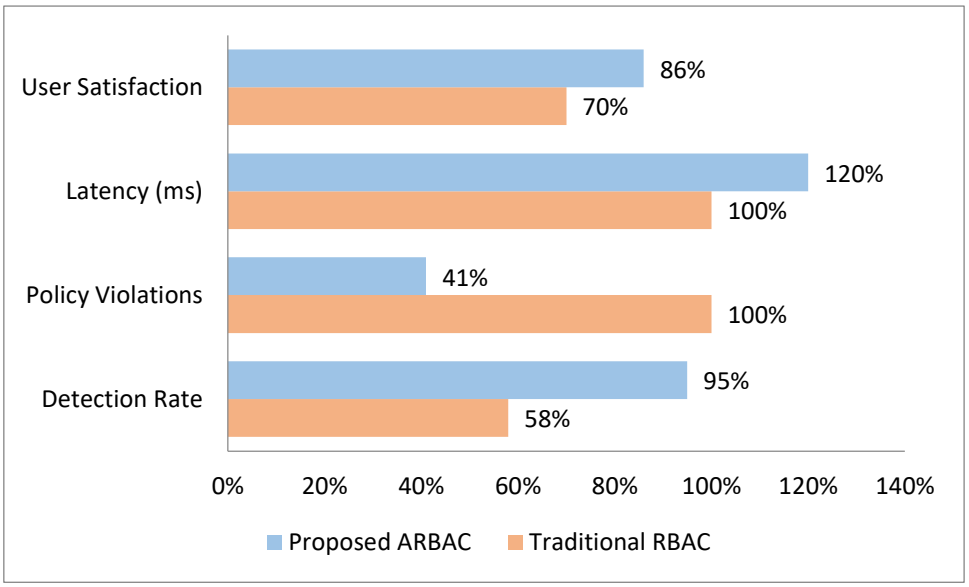

Fig 4: Graph representing Experimental Results

4.3. Discussion

The effectiveness and practical advantages of the proposed Adaptive Role-Based Access Control (ARBAC) framework, as compared to existing RBAC systems in the enterprise space, should be demonstrated in the experimental example. Most importantly, ARBAC was able to produce a dramatic jump in the threat detection rate, up to 95 percent against the baseline of 58 percent. The use of anomaly detection, stemming from machine learning and real-time context assessment, enables the system to detect suspicious behaviour patterns and identify anomalies that are unavailable to static RBAC systems, contributing to this improvement. Dynamic analysis of the context, including factors such as location, device trust, and network risk, will enhance the capabilities of ARBAC in detecting and responding to potential insider threats and attempted malicious access. A reduction in the number of policy violations was another core finding, as the number of violations decreased by 59 per cent compared to the traditional model. This demonstrates that ARBAC has been successful in establishing more fine-grained and situationally aware access policies that can accommodate the diverse operational contexts of users. Instead of being based on fixed role-based mappings of permissions, ARBAC considers each permission request on many more parameters, such as the criticality of each mission and behavioral norms. It will guarantee that they will only give access when it is necessary, and therefore minimizing the possibility of abuse of privilege or unintentional privilege violation. Although ARBAC did create some degree of latency increase (up to 300ms as compared to 250ms), the delay is still very much within reasonable boundaries necessary to operate ERPs, all things considered, keeping with the added security and control which can be availed through it. Moreover, the user satisfaction level increased to 86%, compared to the initial value of 70%, indicating that users were generally satisfied with the system's usability and security. It can be assumed that this favorable perception was facilitated by the adaptive and personalized character of ARBAC since, on the one hand, users did not have so many access frustrations, and, on the other hand, ARBAC allowed keeping the environment secure. All in all, the ARBAC model exhibits good performance, usability, and a strong security trade-off; thus, it is a viable and scalable access control model in current ERPs.

4.4. Case Study

A case study was implemented in a simulated military logistics ERP system to confirm the real-world applicability of the proposed ARBAC framework. The selected environment has been selected because military operations are so sensitive, and one unauthorized access can have very serious implications. The simulation involved numerous user functions, including logistics officers, supply chain managers, and external contractors, each with different access needs. The breach scenario tested on the system was unauthorized access to privileged escalation by the internal user through a rogue device accessing restricted supply chain information during the time of non-operation. Such access in a legacy RBAC system may not have been flagged as

inadmissible if the user's role permitted it. Nevertheless, several context-based security measures were implemented in the ARBAC enhanced system. The Context Evaluator detected several weak signals: the user had accessed the system using an unknown IP address, a device that did not meet the necessary security certifications, and the request occurred during an activity window that fell outside the user's normal work pattern. Based on this real-time contextual data, the Policy Decision Point (PDP), as defined by XACML policies, rejected the access request and provisionally downgraded the user's role. At the same time, the event was being logged by the Monitoring and Auditing Module, and it was scheduled to be reviewed by system administrators at a later time. The anomaly detection module using machine learning has also identified the behavior as high risk because it was not in line with the base profile of the user. Moreover, the ARBAC system demonstrated its adaptive nature by automatically adjusting the roles of users in response to shifts in context. After the user's normal behaviour was restored, as verified by stable usage with a known and trusted device and address, the system resumed the initial level of permission without administrator intervention. In this case study, we have seen how not only unauthorized access is prohibited in high-stakes scenarios using ARBAC, but the unprecedented reliability of action through intelligent, self-correcting role management can be ensured as well. It establishes the appropriateness of the framework in mission-critical systems that require both agility and security.

5. Conclusion

This paper demonstrates that an Adaptive Role-Based Access Control (ARBAC) model can enhance access control mechanisms in ERP systems, particularly in governmental and military contexts where the sensitivity of data and the security of operations are of paramount importance. In contrast to traditional RBAC, which relies on predetermined role-to-permission mappings, the proposed ARBAC incorporates context-awareness and machine-learning-powered anomaly detection to facilitate real-time access decisions and permit a dynamic nature. The main elements of the framework are the Role Assignment Engine which dynamically adjusts user roles depending on the missions which are more critical and the mission contexts, a context Evaluator which determines the parameters of various conditions including device trust, location, access time and access network risk and the Policy Decision Point (PDP) which utilizes the XACML set of policies to decide on the access requests. Experimental results indicated that ARBAC substantially outperformed the baseline, with very high levels of threat detection and few policy violations, and a low weight in terms of the latency increase on the system. The simulated case study on a military logistics ERP system also confirmed the practical efficiency of the framework in barring unauthorized user access by performing dynamic analysis of the context and adapting the roles. On the whole, ARBAC is a compromise in terms of the security of ERP environments, as it contributes to increased flexibility, responsiveness, and situational awareness of access control systems.

Although the proposed ARBAC model shows impressive outcomes, there are several areas that need improvement in the future. One such direction is the incorporation of blockchain technologies to implement tamper-proof audit records. Data integrity, accountability and compliance requirements can be better achieved by means of recording all access events and policy changes on a distributed ledger, thereby making organizations more efficient. The upcoming possible extension is to redevelop ARBAC to accommodate the needs of IoT-enhanced ERP systems, as devices and sensors generate an ongoing stream of operational data and may even require access control. The inclusion of dynamic and autonomous actors in such environments will necessitate further growth in the context evaluation and policy frameworks.

Furthermore, the further inclusion of natural language processing (NLP) in defining access policies and translating these policies may make the system more convenient for non-technical parties. Defining policy in plain language would make administration easier and minimise the possibility of incorrect configuration. Possible future improvements in ARBAC are further enhanced integration with predictive analytics and modeling user behavior to attempt to guess the access requirements and proactively adjust access control. All these improvements will enable ARBAC to become stronger, more scalable, and user-friendly, and will enable it to assist in safe digital transformation even in more complex and multifaceted ERP contexts.

References

[1] Sandhu, R. S. (1998). Role-based access control. In Advances in Computers (Vol. 46, pp. 237-286). Elsevier.

[2] Hu, V. C., Ferraiolo, D., & Kuhn, D. R. (2006). Assessment of access control systems (Vol. 76). Gaithersburg, MD: US Department of Commerce, National Institute of Standards and Technology.

[3] Hu, V. C., Ferraiolo, D., Kuhn, R., Friedman, A. R., Lang, A. J., Cogdell, M. M., ... & Scarfone, K. (2013). Guide to attribute-based access control (ABAC) definition and considerations (draft). NIST special publication, 800(162), 1-54.

[4] Alam, M., Breu, R., & Hafner, M. (2007). Model-driven security engineering for trust management in SECTET. J. Softw., 2(1), 47-59.

[5] Kulkarni, D., & Tripathi, A. (2008, June). Context-aware role-based access control in pervasive computing systems. In Proceedings of the 13th ACM Symposium on Access Control Models and Technologies (pp. 113-122).

[6] Covington, M. J., Long, W., Srinivasan, S., Dev, A. K., Ahamad, M., & Abowd, G. D. (2001, May). Securing context-aware applications using environment roles. In Proceedings of the sixth ACM symposium on Access control models and technologies (pp. 10-20).

[7] Chakraborty, S., & Ray, I. (2006, June). TrustBAC: integrating trust relationships into the RBAC model for access control in open systems. In Proceedings of the eleventh ACM symposium on Access control models and technologies (pp. 49-58).
[8] Zurko, M. E., & Simon, R. T. (1996, September). User-centred security. In Proceedings of the 1996 workshop on New security paradigms (pp. 27-33).
[9] Felt, A. P., Chin, E., Hanna, S., Song, D., & Wagner, D. (2011, October). Android permissions demystified in Proceedings of the 18th ACM conference on Computer and Communications Security (pp. 627-638).
[10] Park, J., & Sandhu, R. (2004). The UCONABC usage control model. ACM transactions on information and system security (TISSEC), 7(1), 128-174.
[11] Pretschner, A., Hilty, M., & Basin, D. (2006). Distributed usage control. Communications of the ACM, 49(9), 39-44.
[12] Takabi, H., Joshi, J. B., & Ahn, G. J. (2010). Security and Privacy Challenges in Cloud Computing Environments. IEEE Security & Privacy, 8(6), 24-31.
[13] Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., & Chandramouli, R. (2001). Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security (TISSEC), 4(3), 224-274.
[14] Da Silva, C. E., da Silva, J. D. S., Paterson, C., & Calinescu, R. (2017, May). Self-adaptive role-based access control for business processes. In 2017, IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS) (pp. 193-203). IEEE.
[15] Ferraiolo, D., Cugini, J., & Kuhn, D. R. (1995, December). Role-based access control (RBAC): Features and motivations. In Proceedings of the 11th Annual Computer Security Application Conference (pp. 241-48).
[16] Hu, V. C., Kuhn, D. R., Ferraiolo, D. F., & Voas, J. (2015). Attribute-based access control. Computer, 48(2), 85-88.
[17] Yuan, E., & Tong, J. (2005, July). Attribute-based access control (ABAC) for web services. In the IEEE International Conference on Web Services (ICWS'05). IEEE.
[18] Samarati, P., & De Vimercati, S. C. (2000). Access control: Policies, models, and mechanisms. In International school on foundations of security analysis and design (pp. 137-196). Berlin, Heidelberg: Springer Berlin Heidelberg.
[19] Penelova, M. (2021). Access control models. Cybernetics and Information Technologies, 21(4), 77-104.
[20] Ruan, C., & Varadharajan, V. (2014). Dynamic delegation framework for role-based access control in distributed data management systems. Distributed and Parallel Databases, 32(2), 245-269.
[21] Noor, S., Awan, H.H., Hashmi, A.S. *et al.* Optimizing performance of parallel computing platforms for large-scale genome data analysis. *Computing* 107, 86 (2025). https://doi.org/10.1007/s00607-025-01441-y
[22] L. N. R. Mudunuri, V. M. Aragani, and P. K. Maroju, "Enhancing Cybersecurity in Banking: Best Practices and Solutions for Securing the Digital Supply Chain," Journal of Computational Analysis and Applications, vol. 33, no. 8, pp. 929-936, Sep. 2024.
[23] Thirunagalingam, A. (2024). Bias Detection and Mitigation in Data Pipelines: Ensuring Fairness and Accuracy in Machine Learning. Available at SSRN 5047605.