Balancing Access and Control: Identity Management with Segregation of Duties

SHARAD SHARMA
(CISSP, Sr. Member IEEE, Alumni IIT Kanpur)

Abstract:

Segregation of Duties (SoD) enforcement is crucial to avoid fraud, mistakes and mis-compliance in organizations. However, enforcing SoD correctly is a challenging task in identity management given the presence of orphaned accounts, exploding number of agents, roles and dynamic access needs in hybrid IT environments using identity governance technologies. In this paper, we discuss research challenges and solutions to enforce correct SoD in hybrid IT environments using identity governance technologies. We compare static rule-based SoD detection using predefined conflict matrices with Artificial Intelligence (AI)-supported SoD detection methods such as role mining and predictive detection of conflicting roles. Our evaluations show that SoD violations can remain undetected for a long time because of orphaned accounts (accounts of users that still exist after changes in role or termination of employment including accounts that are not linked to or owned by users). While role mining reveals new insights, it also uncovers hidden conflicts that must be continuously fine-tuned to avoid false positives. While a rule-based system provides easy audit-ready compliance for Sarbanes-Oxley Act (SOX) and the General Data Protection Regulation (GDPR) for complex SoD scenarios, it monitors adaptive risk. To take full advantage of this recommendation, regular access recertification, automating provisioning and deprovisioning workflows and using compensating controls where perfect SoD separation is not possible can help.

Identity Management; Segregation of Duties; Orphaned Accounts; Role Mining; Rule-Based Detection; AI-Driven Detection; Compliance; SOX; GDPR

1. Introduction

The identity and access management (IAM) market has rapidly evolved over the past two decades. Initially focused on authentication—verifying user identity, it expanded to include access management with primary focus on provisioning and deprovisioning access. Simple identity management evolved into Identity Governance (IdGov) as corporations grew. IdGov is a strategic discipline managing user identities throughout their life cycle (onboarding, offboarding, role changes, etc.). It involves user provisioning/de-provisioning, periodic access rights recertification, enabling access request workflow for “Access by exception” and enforcing organizational rules (business policies and processes). IdGov addresses not only “Who is this user?” but also “What rights should this user have, for how long, how he should get these rights and under what circumstances?” Separation of duties (SoD) is crucial internal control to prevent and detect fraud and errors. It ensures no individual or group has all roles and authorities to perform risky transactions. For instance, one might create a transaction but not post or approve payment, or the person adding a vendor isn't the one approving payment(Mpamugo and Ansa,

2024; Ghadge, 2024). SoD prevents financial fraud, protects sensitive information, and ensures compliance. Enforcing SoD policies is challenging. Legacy SoD approaches with static policies and periodic reviews worked in monolithic environments. Today's hybrid IT environments with on-premises software, SaaS, IaaS, and PaaS are vulnerable to sophisticated attacks from multiple devices and locations. Roles and permissions don't map across systems, and temporary access often remains long after its purpose ends (Uddin et al., 2019).

Enforcing dynamic Separation of Duties (SoD) constraints in hybrid environments is a non-trivial task for organizations today. Currently there is a lack of effective practices to enforce dynamic SoD constraints in hybrid environments. This is in addition to the static approaches provided by tools such as Automated Business Intelligence (ABHI) and Access Certifier that generate conflict matrices, as well as periodic SoD reviews where tools such as the Team Discovery for SoD Enforcement approach perform access reviews every quarter (Obuse et al., 2025). Thus, organizations are looking for SoD enforcement that operates in real-time or near real-time, or even better that learns and adapts to access patterns without significant disruption to normal business activities. Although Identity Governance technologies have become ubiquitous, SoD enforcement is still far from being fully addressed due to the challenges of handling inactive/orphaned accounts and the so-called role mining problem (Filho, 2025; Angela et al., 2024). In addition to discussing current technological solutions to SoD enforcement, this paper compares rule-based and AI-based approaches for defining feasible combinations of roles to enforce effective SoD. The implications of both rule-based and AI-based approaches on important compliance use cases, such as those required by SOX as well as newly introduced GDPR, are discussed. This paper provides a practical reference for identity governance based on effective SoD controls.

2. Theoretical Framework of Identity Management

Identity management is about ensuring that the right people have access to the right resources for the right reasons at the right time and in the right way. It is the domain for enforcing the principles of least privileges. Theoretical identity management models describe the Identity Management (IdM) framework based on the identity lifecycle, authoritative identity sources, and advanced access models. This article will focus on these three components of the identity management theory (Wang et al., 2004; Yutaka et al., 2019).

2.1 The Identity Lifecycle: Provisioning, Reconciliation, and Deprovisioning

Every digital identity follows a predictable lifecycle, beginning before the user ever logs in and ending long after they leave an organization.

- **Provisioning:** Provisioning involves creating user accounts and assigning roles and permissions for appropriate access to organization assets. This can be done manually by administrators or automatically via workflow-based software. Many organizations allow self-service provisioning for low-risk actions like password resets for certain applications. In mature identity governance, provisioning follows the least privilege principle, granting users minimal access needed for their job functions(Yutaka et al., 2019; Zhang et al., 2020).

- **Reconciliation:** Reconciliation verifies identity information against a repository and allows for creating a single source of record for all digital identities access. As identities are provisioned, access may be set manually without updating the repository, resulting in "orphaned" identities with no owner, "ghost" identities provisioned but unassociated with any user, and privilege drift where access increases over time. Regular reconciliation is crucial to enforce Same Level of Access by identifying conflicts missing by automated provisioning (Yutaka et al., 2019) and identifying any rogue access or managing any "Access by Exception".
- **Deprovisioning**: Remove access when no longer needed (e.g., terminated employee, role/responsibility change, transferred, expired contractor agreement, changing user types from contractor to employee or vice versa). Many Separations of Duty failures occur when deprovisioning is delayed, or users retain irrelevant or conflicting privileges from a prior role. Best practice is to immediately deprovision a terminated employee or conditionally deprovision for role changes, ensuring all rights are removed before granting new ones (Yutaka et al., 2019).

3. **Authoritative Sources:** Human Resource (HR) Systems**,** Lightweight Directory Access Protocol (LDAP), and System for Cross-domain Identity Management (SCIM)

An authoritative source is the single source of truth for identity attributes. Without authoritative sources, identity management becomes fragmented and unreliable. The source of identities for an organization can be fragmented and spread across multiple sources like HR systems, contractor/vendor management tools or a custom HR source.

- HR Systems (e.g., Workday, SAP SuccessFactors, Oracle HCM) are the main source of employee identities and hire/termination data (e.g., hire date, termination date, job code, department, manager, role name). Identity governance solutions integrate with these systems to manage user accounts and access. For instance, if an Accounting Clerk is promoted to Accounts Payable Supervisor, the identity system recalculates access rights and checks for Same-Duty conflicts (Bhatt et al., 2017; Servos and Osborn, 2015).
- LDAP including Microsoft Active Directory (AD) and OpenLDAP, manages authentication and authorization, storing access rights. HR systems are the source of truth for user identity, while LDAP handles provisioning/deprovisioning. LDAP is not authoritative for identity attributes, as most changes occur outside HR workflows (Jabal et al., 2020; Narouei et al., 2017).
- SCIM (System for Cross-domain Identity Management) defines RESTful interfaces for managing user information on cloud applications like Salesforce and Office 365. Unlike directory-centric LDAP, SCIM is application-centric. In hybrid environments, SCIM provisioning synchronizes identities between on-premises stores (e.g., Active Directory) and cloud apps (e.g., Entra, Office 365) in real-time (Jabal et al., 2020; Narouei et al., 2017).

3.1 Advanced Concepts: Just-in-Time Access and Attribute-Based Access Control (ABAC)

Traditional identity management relies on static role assignments, but modern environments demand greater flexibility. Two advanced concepts address this need.

- **Just-In-Time (JIT):** Just-In-Time (JIT) Access grants users' temporary access for specific tasks, reducing standing privileges that pose SoD risks. JIT Access is common for administrators and urgent scenarios. In the example, access is for 60 days with SoD workflow approval (Zhu et al., 2019).
- **Attribute-Based Access Control (ABAC):** Attribute-Based Access Control (ABAC) bases decisions on user, resource, and environmental attributes, unlike traditional Role-Based Access Control (RBAC). ABAC uses dynamic conditions for access, e.g., approving purchase orders if the user's department matches the vendor's, its business hours, and the amount is within limits. This offers fine separation of duties, which is not possible with RBAC. Despite added administrative complexity, in high-risk environments, the benefits to risk reduction and security outweigh costs(Wang et al., 2004; Yutaka et al., 2019).

4. Segregation of Duties: Formal Models and Conflict Matrices

Segregation of Duties (SoD), a crucial business policy, is usually a corporate policy rather than a formal constraint. We discussed modeling SoD (following sub-sections) as a constraint satisfaction problem using Role-Based Access Control (RBAC) models, and the use of conflict matrices and static versus dynamic rules (Li et al., 2007; Li et al., 2004; Gligor et al., 1998).

4.1 SoD as a Constraint Satisfaction Problem

In formal terms, SoD violation occurs when a single user is assigned two or more permissions that together enable a high-risk transaction. The goal is to satisfy the constraint that no user should hold conflicting permissions. This is a constraint satisfaction problem because:

- **Variables** = Users, roles, and permissions
- **Domains** = Possible role assignments
- **Constraints** = Mutually exclusive permissions or roles

If assigning Role A and Role B to the same user violates a SoD rule, the system must either prevent the assignment or require compensating control (e.g., managerial override with logging) (Li et al., 2007; Li et al., 2004).

4.1 Formal Definitions Using RBAC Models

The foundational RBAC model proposed by Sandhu et al. (1996) defines SoD using two key constructs:

- **Static Separation of Duty (SSD):** This object type enforces the rule that a user cannot have two conflicting roles. Example: A user cannot be assigned the role Purchase Requisition Creator and also the role of Purchase Order Approver. SSD is enforced during role assignments (Uddin et al., 2019).

- **Dynamic Separation of Duty (DSD):** A constraint that allows a user to hold both roles but is denied the ability to exercise both roles' permissions within a single transaction or session. Example: A single user can hold both roles, but once that user executes the first permission of a role, that user is then denied the ability to execute the first permission of the other role for that transaction or session (Thakare et al., 2020).

4.2 SoD Conflict Matrices

The conflict matrix (SoD matrix) is commonly used to outline incompatible roles and clearly define permissions and access conflicts. The conflict matrix is organized in a table format with the role names (accounts, functions etc.) as the headers for both the rows and the columns. An "X" is placed in the cells representing a conflict between the roles (Joshi et al., 2005).

Role	Create Vendor	Approve Payment	Release Payment	Reconcile Account
Create Vendor	-	X	X	X
Approve Payment	X	-	X	X
Release Payment	X	X	-	-
Reconcile Account	X	X	-	-

Table 1. Example of Simple SoD Conflict Matrix (Financial Process)

An "X" marks a forbidden combination. For example, the same user cannot either Create Vendor or Approve Payment. A blank cell (-) means the combination is allowed, such as Release Payment and Reconcile Account (Yang and Hu, 2024). This can be explained using the example of Create Vendor permission. A user who submits and onboards new vendors can fraudulently approve payments to a bogus vendor and also release payments to it if the above defined SoDs are not enforced.

5. Technical Challenges in SoD Enforcement

Models and conflict matrices show challenges with Segmentation of Duties (SoD) and issues from incorrect duty assignment but not the practical difficulties in implementing SoD. This article discusses four additional technical challenges.

5.1 Role Explosion

Traditional Role-Based Access Control (RBAC) suffers from "role explosion," where every new set of permissions requires a new role. But as a company grows and adds new applications, it also goes through changes in business processes and must meet new regulatory requirements, meaning the number of roles can explode. For example, the organization has 3 organization dimensions (10 job functions,5 geographic locations and 4 product lines) and the client started their calculations with 10 x 5 x 4 = 200 roles. However, once you add in the SoD rules, the client quickly realized they needed a more advanced approach to organize. The overhead of too many roles makes management difficult, leading administrators to use workarounds where they assign direct permissions instead of using role. This can be mitigated by using ABAC or role mining to reduce number of roles to a subset of similar roles (Thakare et al., 2020).

5.2 Transitive Conflicts Across Multiple Systems

Transitive conflict arises when an identity is granted conflicting permission through role assignments across systems. For example, an identity gets the "Create Purchase Order" role in System A and the "Approve Purchase Order" role in System B. The user seems to have a single set of permissions until accessing a system where both roles can be executed, violating SoD principles. Current SoD solutions only detect conflicts within a system and do not consider conflicts from a single identity across multiple applications. Unified identity governance is needed to identify and mitigate these conflicts, enforcing SoD policy across all systems with a unified engine (Uddin et al., 2019). SoD spread across multiple systems are commonly referred to as cross application SoD and enforcing cross application SoD encompassing multiple enterprise application which themselves have SoD is one of the most challenging problems today. Identifying these SoDs itself is a big challenge which is further complicated when we define a process to prevent and detect these SoD violations.

5.3 Mitigating Controls: Supervisory Override with Logging

Ideally, teams and users should have separate duties, but exceptions can lead to conflicts. Organizations can allow supervised actions with logging. A mitigating control is an "approve" override, letting a request be blocked and reviewed by management. The supervisor can approve, log the action, and record identities and reasons for auditors. In a small accounting department, one person might prepare and approve checks, needing a manager's override code, logged for monthly review. Effective control requires overrides to be exceptions. If daily overrides occur, organizations should reconsider their SoD design (Uddin et al., 2019). Effectively identifying these exceptions, logging the mitigating controls and tracking the audit of these exceptions and timely mitigation of these exceptions is a major challenge in today's environments where organizations have to ensure they adhere to multiple industry standards and laws including but not limited to SOX and GDPR.

5.4 Ghost Users and Service Accounts Bypassing SoD

SoD rules enforce permissions for human users and administrators but miss two types of "non-human" accounts: service accounts and "ghost" users. Service accounts, used for software/applications integration, APIs, and automation, may have overlapping privileges necessary for function, which would be suspicious for human users. When employees leave, their accounts might remain active, creating "ghost users" who can go unnoticed for months and potentially engage in detrimental actions. An example of a SoD exception involves a data integration tool's service account with read permissions for financial transactions and write permissions for user roles. Human employee exceptions with less sensitive permissions are typically denied. For service accounts, we document exceptions due to technical infeasibility and perform quarterly reconciliations against HR data to identify and deactivate "ghost" users. We also require quarterly manager attestation for active accounts (Uddin et al., 2019; Atlam and Yang, 2025).Within the realm of this issue, agents within Artificial systems are recent additions and are also treated as non-human identities.

6. Compliance and Regulatory Landscape

Segregation of Duties (SoD) is a requirement in many regulatory and compliance frameworks. In this guide we will look at the SoD requirements for SOX, also highlighting the requirements for other major frameworks including COSO, GDPR, HIPAA and NIST SP 800-53. We will also cover some common audit failures and potential compensation controls (Olajide et al., 2024).

6.1 SOX Section 404

The SOX Section 404 rule mandates publicly traded companies to have adequate internal controls over financial reporting. Segmentation of Duties (SoD) is often associated with SOX Section 404, though not specifically mentioned. SoD involves splitting tasks so one employee cannot complete a whole transaction. For instance, an employee posting vendor payments should not approve them. SoD failures are reported as material weaknesses and disclosed publicly, risking employees and companies. Breaches of SoD are often simple to commit, such as when one employee handles both posting and approving payments (Moeller, 2012).

6.2 COSO Framework

In the COSO framework, Segregation of Duties (SoD) falls under Control Activities. SoD involves dividing conflicting roles and responsibilities among personnel to reduce fraud and error risk. Auditors assess if such conflicting roles exist and are managed through controls. Failing to address SoD indicates a deficiency in the control environment (Alrahamneh, 2024; Thabit, 2019; Espinosa-Jaramillo, 2024).

6.3 GDPR

Adhering to GDPR, access to employees' personal information should match their role and enable effective function. Implementing an access strategy requires a least privilege model and duty separation. Allowing one-person full access to sensitive customer data and permission to alter access rights would breach regulations, potentially costing €20 million or 4% of global revenue (Wang et al., 2024; Folorunso et al., 2024).

6.4 HIPAA

Protecting protected health information (PHI) is a responsibility of a HIPAA-covered entity enforced through policy. The implied Split Duty (SoD) roles are those who grant access to PHI and those who audit access for monitoring and reporting. The OCR has cited failures in separating these roles by HIPAA regulations (Moeller, 2012).

6.5 NIST SP 800-53

AC-5 (Separation of Duties) and AC-6 (Least Functionality/Least Privilege) in NIST SP 800-53 require the documentation of incompatible functions, separation of duties to the degree practicable, and a periodic review of current privileges and roles. Federal agencies must implement SoD in order to receive authorization to operate (ATO) (Wang et al., 2024; Roy, 2020).

6.6 Audit Failures Due to SoD Violations

SoD violations are a leading cause of poor audit findings. A material weakness was identified at a telecommunications company where the same person recorded and approved journal entries. A

retailer was fined for SoD violations when a system administrator could add user accounts and approve access to financial systems. Root causes include orphaned/rogue accounts, users skipping manual steps, and unidentified transitive conflicts (Wang et al., 2024; Alrahamneh, 2024).

6.7 Compensating Controls

When complete separation of duties is unfeasible, compensating controls like supervisory review, transaction sampling, automated logging with alerts, and mandatory duty rotation mitigate risk. If consistent, documented, and effective, auditors accept them as alternatives. Continued reliance on these controls may suggest the need to re-engineer business processes (Faruq, 2025).

7. Implementation Models and Maturity Levels

SoD enforcement is never 100% effective initially. As an organization matures in its use of SoD enforcement, it will move through four levels of maturity. This section describes these four levels of maturity from manual to predictive and then AI-powered detection and enforcement.

- **Level 1: Manual Segregation Checklists**

A Security of Data (SoD) low maturity level involves manual enforcement. This is initially done by maintaining a spreadsheet or paper checklist detailing conflicting roles or permissions. When a new user is added or a role updated, the administrator manually checks the list to avoid conflicts. There is no automated enforcement, so compliance relies entirely on manual checklist adherence.

Characteristics: Errors in setting roles and responsibilities are easily made and bypassed, with no audit trails for verification, and this method is not scalable for large organizations. An example is assigning conflicting roles because the administrator did not double-check the list (Uddin et al., 2019).

- **Level 2: Rule-Based Identity Analytics**

Level 2 automated identity analytics tools monitor user activity and report SoD violations. Static conflict matrices (Section 4) dynamically compare user assignments to SoD rules, and reports are reviewed by Security or compliance officers to decide on revoking permissions or assessing acceptable risk with current controls.

Characteristics: This tool scans and reports violations, possibly weeks or months later. Although not as effective as hoped, it is more effective than a manual checklist, provides an audit trail, and suits organizations with moderate risk, but not high risk (Uddin et al., 2019). Auditing the identified SoD violations adds additional complexity to this process.

- **Level 3: Automated Preventive SoD During Access Request**

At Level 3 enforcement, the focus is preventive, preventing Manager/Administrators from assigning SoD blocked roles/permissions. The system validates requests in real-time for SoD conflicts. If conflicts are found, the request fails, and an error message is provided. The requestor

can assign a different role, request a supervisory override through a workflow, or cancel the request.

Characteristics: Prevents unwanted identity provision to prevent malicious activity, requiring integration with Identity Governance and provisioning systems (HR, LDAP, SailPoint, Saviynt.). These are minimum expectations for compliance (e.g., SOX) and other regulated organizations (Filho, 2025). This level still requires to record any transactions performed by users with SoD violations

- **Level 4: AI/ML Predictive Conflict Detection**

Predict unseen SoD conflicts. Manually defining all conflicts between role pairs is challenging, especially in complex systems with many roles. Unknown conflicts can be overlooked. At higher maturity level, access patterns, user behavior, and transaction logs train an AI/ML model to predict anomalous role combinations indicating unknown conflicts. Guru, a machine learning tool provider, analyzed data breaches and found that breached users had both "Export Customer Data" and "Modify Access Logs" permissions. The system flagged these as potential SoD conflicts without human setup.

Characteristics: Our predictive and adaptive method detects potential conflicts and risks, identifying not only traditional transitive conflicts but also previously unrecognized ones. However, training on large datasets, fine-tuning to avoid false positives, and using it incur significant overheads, making it mainly suitable for large enterprises in high-risk industries like finance, healthcare, and government (Filho, 2025). This level comes with the inherent challenges that come with implementing an AI tool for example model poisoning, higher cost, authentication and authorization challenges. The table below summarizes the maturity levels:

Maturity Level	Description	Characteristics	Benefits
Level 1: Manual Segregation Checklists	Human-driven checks of SoD via static lists and audits	Labor-intensive, error-prone, low scalability	Simple to implement for small setups
Level 2: Rule-Based Identity Analytics	Automated application of static SoD rules to identity data	Post hoc conflict detection, reactive compliance	Improved accuracy, efficient audits
Level 3: Automated Preventive SoD Enforcement	Integrated real-time access control preventing conflicts during requests	Dynamic SoD, workflow-aware, policy-enforced at decision point	Proactive risk mitigation, improved governance
Level 4: AI/ML Predictive Conflict Detection	Predictive analytics using AI/ML to forecast and prevent SoD violations	Adaptive, context-aware, behavioral anomaly detection	Enhanced accuracy, anticipatory control, regulatory transparency

Table 2. Summary Table of SoD Maturity Model

8. Conclusion & Future Directions

- **Conclusion:** This paper talks about how identity management works with Segregation of Duties. The paper explains the big ideas. The paper gives some examples. The paper

points out some technical problems. The paper lists rules that people need to follow. The main finding is clear. No single technical fix can handle SoD enforcement alone. We need three things to make SoD work well. The policy lists any roles that can overlap. The policy explains how much risk is allowed. The policy tells what to do if someone does not follow the rule. If the policy is not clear, even the best technology gets ignored. Second, identity governance gives the main tools and systems people need. This has trusted sources, ways to manage accounts over time, and one place to see what happens in different systems. Third, when the team checks often, the SoD controls keep working as the company changes. The team can find orphaned accounts. The team can see role drift. The team can spot new conflicts fast. Technical problems like role explosions, transitive conflicts, service accounts that get around controls, and the limits of set rules show that one way does not fit every place. Companies need to see the risks, the rules, and the limits at work. This helps companies pick the best way to do the work. Companies can use paper checklists or use AI that predicts things. Most groups that follow rules pick Level 3. Automated checks help spot SoD issues when someone asks for access. Level 4 uses AI or machine learning to find issues before they show up. Level 4 is new. Only some large companies with good data use Level 4 right now (Mpamugo and Ansa, 2024; Ghadge, 2024). The paper highlights Identity governance and administration tools can be used to enforce potential preventive and detective SoD.

- **Future Research Directions:** Future research should include federated separation of duty (SoD) across multi-cloud platforms. Organizations typically use both traditional on-premises applications and modern cloud services. For instance, financial applications might remain on-premises, while AWS, Azure, or Google Cloud Platform (GCP) handle customer-facing applications, and Workday manages Human Resources Information Management Systems (HRIMS). Other organizations use a mix of on-premises and cloud-based services like Salesforce, Netsuite, ServiceNow, or other SaaS providers. Each cloud manages user and API identities differently, making it hard to detect SoD conflicts across clouds with current solutions. Future work should explore decentralized identity formats like Verifiable Credentials, cross-cloud policy management frameworks, and zero-trust architectures to enforce SoD at a resource level. Explainable AI models are also needed to detect SoD conflicts, reduce false positives, and build trust in predictive models for InfoSec practitioners. SoD is not a one-time task but a discipline essential for reducing fraud risk, passing audits, and maintaining security posture. It must scale with the organization and support multiple cloud platforms (Uddin et al., 2019).

9. References

1) Mpamugo, E., & Ansa, G. (2024). Enhancing Network Security in Mobile Applications with Role-Based Access Control. Journal of Information Systems and Informatics, 6(3), 1872–1899. https://doi.org/10.51519/journalisi.v6i3.863
2) Ghadge, N. (2024). Enhancing Identity Management: Best Practices for Governance and Administration. 219–228. https://doi.org/10.5121/csit.2024.141119
3) Uddin, M., Islam, S., & Al-Nemrat, A. (2019). A Dynamic Access Control Model Using Authorising Workflow and Task-Role-Based Access Control. IEEE Access, 7, 166676–166689. https://doi.org/10.1109/access.2019.2947377

4) Obuse, E., Ayanbode, N., Cadet, E., Essien, I., & Etim, E. (2025). Privacy-First security models for AI-integrated identity governance in multi-access cloud and edge environments. Computer Science & IT Research Journal, 6(8), 506–524. https://doi.org/10.51594/csitrj.v6i8.2012
5) Filho, W. L. R. (2025). THE ROLE OF AI IN ENHANCING IDENTITY AND ACCESS MANAGEMENT SYSTEMS. International Seven Journal of Multidisciplinary, 1(2). https://doi.org/10.56238/isevmjv1n2-011
6) Angela, O., Atoyebi, I., Soyele, A., & Ogunwobi, E. (2024). Enhancing fraud detection and prevention in fintech: Big data and machine learning approaches. World Journal of Advanced Research and Reviews, 24(2), 2301–2319. https://doi.org/10.30574/wjarr.2024.24.2.3617
7) Servos, D., & Osborn, S. L. (2015). HGABAC: Towards a Formal Model of Hierarchical Attribute-Based Access Control (pp. 187–204). Springer. https://doi.org/10.1007/978-3-319-17040-4_12
8) Narouei, M., Khanpour, H., Takabi, H., Parde, N., & Nielsen, R. (2017). Towards a Top-down Policy Engineering Framework for Attribute-based Access Control. 103–114. https://doi.org/10.1145/3078861.3078874
9) Abu Jabal, A., Bertino, E., Lobo, J., Law, M., Russo, A., Calo, S., & Verma, D. (2020). Polisma - A Framework for Learning Attribute-Based Access Control Policies (pp. 523–544). Springer. https://doi.org/10.1007/978-3-030-58951-6_26
10) Bhatt, S., Patwa, F., & Sandhu, R. (2017). ABAC with Group Attributes and Attribute Hierarchies Utilizing the Policy Machine. 17–28. https://doi.org/10.1145/3041048.3041053
11) Yutaka, M., Zhang, Y., Sasabe, M., & Kasahara, S. (2019). Using Ethereum Blockchain for Distributed Attribute-Based Access Control in the Internet of Things. 1–6. https://doi.org/10.1109/globecom38437.2019.9014155
12) Wang, L., Wijesekera, D., & Jajodia, S. (2004). A logic-based framework for attribute-based access control. 45–55. https://doi.org/10.1145/1029133.1029140
13) Zhu, Y., Yu, R., Ma, D., & Cheng-Chung Chu, W. (2019). Cryptographic Attribute-Based Access Control (ABAC) for Secure Decision Making of Dynamic Policy with Multiauthority Attribute Tokens. IEEE Transactions on Reliability, 68(4), 1330–1346. https://doi.org/10.1109/tr.2019.2948713
14) Zhang, Y., Yutaka, M., Sasabe, M., & Kasahara, S. (2020). Attribute-Based Access Control for Smart Cities: A Smart-Contract-Driven Framework. IEEE Internet of Things Journal, 8(8), 6372–6384. https://doi.org/10.1109/jiot.2020.3033434
15) Li, N., Tripunitara, M. V., & Bizri, Z. (2007). On mutually exclusive roles and separation-of-duty. ACM Transactions on Information and System Security, 10(2), 5. https://doi.org/10.1145/1237500.1237501
16) Li, N., Bizri, Z., & Tripunitara, M. V. (2004). On mutually exclusive roles and separation of duty. 42–51. https://doi.org/10.1145/1030083.10300
17) Joshi, J. B. D., Bertino, E., Latif, U., & Ghafoor, A. (2005). A generalized temporal role-based access control model. IEEE Transactions on Knowledge and Data Engineering, 17(1), 4–23. https://doi.org/10.1109/tkde.2005.1

18) Thakare, A., Lee, E., Kumar, A., Nikam, V. B., & Kim, Y.-G. (2020). PARBAC: Priority-Attribute-Based RBAC Model for Azure IoT Cloud. IEEE Internet of Things Journal, 7(4), 2890–2900. https://doi.org/10.1109/jiot.2019.2963794

19) Yang, B., & Hu, H. (2024). Resiliency Analysis of Role-Based Access Control via Constraint Enforcement and Mathematical Programming. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 54(7), 4089–4100. https://doi.org/10.1109/tsmc.2024.3373567

20) Gligor, V. D., Gavrila, S. I., & Ferraiolo, D. (1998). On the formal definition of separation-of-duty policies and their composition. 430, 172–183. https://doi.org/10.1109/secpri.1998.674833

21) Atlam, H. F., & Yang, Y. (2025). Enhancing Healthcare Security: A Unified RBAC and ABAC Risk-Aware Access Control Approach. Future Internet, 17(6), 262. https://doi.org/10.3390/fi17060262

22) Roy, P. P. (2020, February 1). A High-Level Comparison between the NIST Cyber Security Framework and the ISO 27001 Information Security Standard. https://doi.org/10.1109/ncetstea48365.2020.9119914

23) Alrahamneh, S. (2024). ENHANCING INTERNAL AUDIT QUALITY IN JORDANIAN INSURANCE COMPANIES A COSO FRAMEWORK PERSPECTIVE. EDPACS, 69(6), 1–27. https://doi.org/10.1080/07366981.2024.2307068

24) Espinosa-Jaramillo, M. T. (2024). Internal Control in Companies from the Perspective of the COSO. Management (Montevideo), 2, 28. https://doi.org/10.62486/agma202428

25) Thabit, T. (2019). Determining the Effectiveness of Internal Controls in Enterprise Risk Management based on COSO Recommendations. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3401199

26) Moeller, R. R. (2012). Brink's Modern Internal Auditing. Wiley. https://doi.org/10.1002/9781118371558

27) Faruq, M. O. (2025). A META-ANALYSIS OF CYBERSECURITY FRAMEWORK INTEGRATION IN GRC PLATFORMS: EVIDENCE FROM U.S. ENTERPRISE AUDITS. Journal of Sustainable Development and Policy, 01(01), 224–249. https://doi.org/10.63125/kwhkmb57

28) Wang, W., Sadjadi, S. M., & Rishe, N. (2024). A Survey of Major Cybersecurity Compliance Frameworks. 19, 23–34. https://doi.org/10.1109/bigdatasecurity62737.2024.00013

29) Folorunso, A., Wada, I., Samuel, B., & Mohammed, V. (2024). Security compliance and its implication for cybersecurity. World Journal of Advanced Research and Reviews, 24(1), 2105–2121. https://doi.org/10.30574/wjarr.2024.24.1.3170

30) Olajide, J., Otokiti, B., Nwani, S., Ogunmokun, A., Adekunle, B., & Fiemotongha, J. (2024). A Regulatory Compliance Model for Financial Reporting Across Global Supply Chain Functions. International Journal of Scientific Research in Science and Technology, 11(4), 619–635. https://doi.org/10.32628/ijsrst241151217