International Journal of Cyber-Security and Digital Forensics (IJCSDF) 10(4): 112-119
The Society of Digital Information and Wireless Communications (SDIWC), 2021 ISSN: 2305-0011

Policy of Least Privilege and Segregation of Duties, their Deployment, Application, & Effectiveness

Joseph C. Brickley and Kutub Thakur
Professional Security Studies,
New Jersey City University, Jersey City, NJ, USA
jbrickley@njcu.edu and kthakur@njcu.edu

ABSTRACT

With the risk of insider threats on the rise, organizations should deploy the policy of least privilege and Segregation of Duties (SOD) as a safeguard against malicious exposure of information from disgruntled employees. Effectively deploying the policy of least privilege will also decrease the damage dealt if an outsider compromises an account. SOD consists of a process in which request, and approvals are divided into two separate roles or duties. This will ensure that employees can not commit fraud and cover up their tracks, especially when dealing with money. This paper explores the five least privilege and SOD deployment models consisting of discretionary access control (DAC), role-based access control (RBAC), rule-based access control (RuBAC), attribute-based access control (ABAC), and mandatory access control (MAC). The policy of least privilege and SOD is not a "set it and forget it" defense against fraud, as employees often switch roles and are granted more privileges, commonly referred to as privilege creep. Monitoring and auditing tools should be put in place to assist in identifying and preventing privilege creep. It is recommended to use automated auditing tools, specifically Creeper, as it is fast and more accurate than a human auditor and a competitor.

KEYWORDS

Least Privilege, Segregation of Duties, Access Control Model, discretionary access control (DAC), role-based access control (RBAC), rule-based access control (RuBAC), attribute-based access control (ABAC), and mandatory access control (MAC), Creeper, SELinux, SAM Jupiter.

1 INTRODUCTION

With the threat of malicious insiders on the rise, precautions will need to be taken to protect an organization's information [1]. Organizations rely heavily on their information and the technology that protects and uses it. Due to the dependency's organizations place on information and technology, they become vulnerable to cyber-threats: Managers of these organizations therefore need to take proper precautions and deploy appropriate best practices to mitigate the risk of insider threats and other cyber threats [2]. When a user's account is created on a network, the last thing an organization wants to do is grant that user privileges to the entire network, or in other words, hand them the keys to the kingdom Employees should not have full access, especially if they only need a basic account to perform a specific task. Granting users privileges to the entire network can be crucial as employees who become disgruntled may become insider threats who can exploit poor security policies and bypass all layers of security an organization has in place [3]. Influences from outside an organization also occur as cyber-criminals are also focusing on susceptible people within organizations to target and disgruntled employees are easy targets [4]. Deploying strong security policy can be an overlooked process and is often not considered until it is too late.

A strong security policy that can be put in place to protect organizations against insider threats and other cyber-threats is the policy of least privilege. As referenced by Saltzer and Schroeder [5], the policy of least privilege is when every user's account has minimal privileges to complete the assigned tasks delegated to them. This includes not only general user accounts but also admin accounts and every other account on a network. The policy will limit and control employees' access to information and allow only abilities that they need to perform a task. In theory, it will stop the possibility of employees accidentally altering, exposing, destroying, and exploiting information that they do not need access to. It can help assist in the

separation of duties from employees and reduce a compromised account's effects.

Segregation of duties (SOD), also known as separation of duties, is a policy that is required to be implemented by many types of organizations. Whether it is public companies that must meet Sarbanes Oxley Act (SOX) or federal organizations required to meet National Institute of Standards and Technology (NIST) special publication 800-53 SOD must be met and maintained. SOD is the separation of work so that one employee cannot commit fraud or errors and conceal them in their daily task [6].
The classic example of SOD is dividing a task up into two parts consisting of requests and approvals. An organization will undoubtedly be at risk if employees can make the request and approve them. By implementing SOD, it will add a layer of protection to an organization's processes. According to Islam et al. [7], "Segregation of duties (SOD) is a cornerstone of effective internal controls and the deterrence of fraud" [3]. SOD will add a set of eyes on approvals and increase efficiency in detecting fraud. It is essentially an assurance of checks and balances.

The five deployment models that will be assessed are discretionary access control (DAC), role-based access control (RBAC), rule-based access control (RuBAC), attribute-based access control (ABAC), and mandatory access control (MAC). Recommendations for effective deployment and a conclusion will be offered following the study.

2 DEPLOYMENT MODELS

2.1 Discretionary Access Control

The first deployment option is DAC, which according to Hu et al. [8], leaves granting access to subjects up to the owner of an object. The owner, who creates the object can grant and revoke privileges to any subject they deem fit. Once the owner has granted a subject privileges or rights, the subject may perform the following actions consisting of writing, executing, etc., depending on the granted privileges. DAC is not the ideal model for deploying the policy of least privilege and SOD due to many vulnerabilities. When an owner of an object grants a subject unrestricted permission over their object, the subject has permission to copy the contents of the object assigned to them and duplicate it as their own separate object that they are the creator/owner of in which they can grant privileges to whom they so choose. This makes the read right transitive and will undoubtedly present many vulnerabilities in least privilege/SOD implementation and present problems with fraud detection. Having the owner of an object determine privileges presents many areas of concern as privileges should be defined and determined by an organization's security policy or senior leaders. The DAC model presents issues with privilege creep, where if a subject switches their role or gets assigned a different task, they accumulate more and more privileges without the old privileges being revoked, which could potentially defeat the purpose of SOD and help influence fraud. The DAC model presents too many vulnerabilities to an organization and deployment of this model should be avoided.

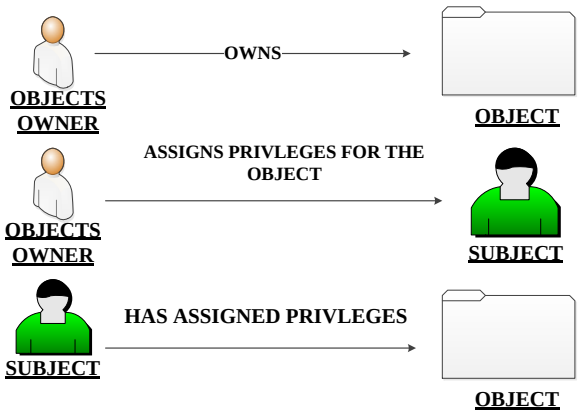

Figure 1: Example of the DAC model in use

2.2 Rule-Based Access Control

The third deployment option is RuBAC, which according to Hu et al. [8], uses a pre-configured and predetermined rule set that is referenced to grant users' access. RuBAC is a generic term and typically refers to an organization's defined rules. RuBAC is typically combined with DAC or RBAC because it compares an access request to the subject's rights and the rules set on the object. RuBAC can provide flexibility for

assigning privileges, but it lacks the constraints between objects, actions (Read, Write, Delete, etc.), and subjects that other access control options offer. It is the administrator's responsibility to assign each subject the necessary actions that they can perform on an object. It can be very time consuming for the administrator and why RuBAC is typically paired with RBAC.

According to Debreceni et al. [9], RuBACs single rule may grant or deny access to a larger number of objects in a folder. For example, a rule might be set for a user to access an object, but inside that object there could be hundreds of folders containing information that the user may not need to have access to complete their assigned task or job function. Due to RuBACs rule-based nature, an implicit nature exists. This implicit nature suggest that many rules would conflict with one another and assign contradicting permissions. The implicit rule may also be used as beneficial when setting rules. According to Bergmann et al. [10] fine-grained policies such as implicit deny may be set as the last rule of the predetermined rule set for RuBAC. Implicit allow will allow access to all users not listed in the predetermined rules. If the last rule set is implicit deny, all users that are not listed to have access granted will be denied access. An example of this may be seen in Figure 3.

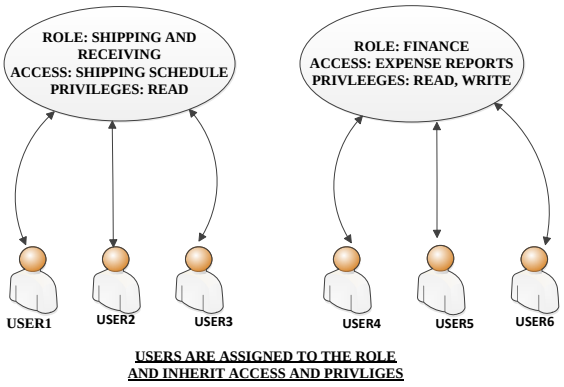

Figure 2: Example of the RBAC model in use

2.3 Role-Based Access Control

According to Ferraiolo et al. [11], RBAC prevents users from having discretionary access to objects. This is done by creating groups of roles and assigning users to these role groups. The roles should be designed only to grant access and privileges for the designated task that those roles need to perform. The users who would be members of these groups would then inherit only the privileges that the group was designed to have; an example of this can be seen in Figure 2. This allows for great flexibility as users can be assigned new roles. Roles can be granted new privileges, and roles can have previous privileges revoked if necessary. Users may also be assigned to more than one role. Users being assigned to more than one role can be beneficial for users who are actively performing two or more roles simultaneously. RBAC can also present many issues concerning SOD and privilege or permission creep.

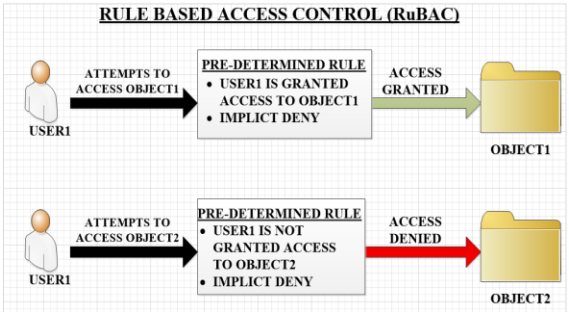

Figure 3: Example of the RuBAC model in use.

2.4 Attribute-Based Access Control

The fourth model, ABAC, is the most granular model option available as it can further limit privileges based on specific attributes. According to Weil and Coyne [12], the organizations which deployed RBAC eventually noticed that a more granular model was needed. The RBAC needs to expand and include attributes consisting of location, day, time of day, etc. Thus, ABAC was used as a replacement for RBAC. ABAC uses user attributes and assigns object labels to determine privileges, increasing the flexibility of RBAC. An example of ABAC is shown in figure 4. RBAC can be combined with ABAC as roles can be used as another attribute type. The configuration of ABAC can be extremely granular and auditing can become challenging. It is recommended to have an expert staff member configure and

administer ABAC, especially if roles are included.

According to Panende et al. [13], When ABAC is implemented as a security policy it will increase the security level of an organization due to the extremely granular and flexible nature of ABAC. Which is extremely beneficial from a security standpoint, however acquiring an expert staff member may be the challenge. Many organizations that deployed RuBAC have also been found to switch to ABAC to increase the flexibility that RuBAC does not offer [14]. With such flexibility and granularity ABAC is a valid option when deploying Least privilege and SOD.

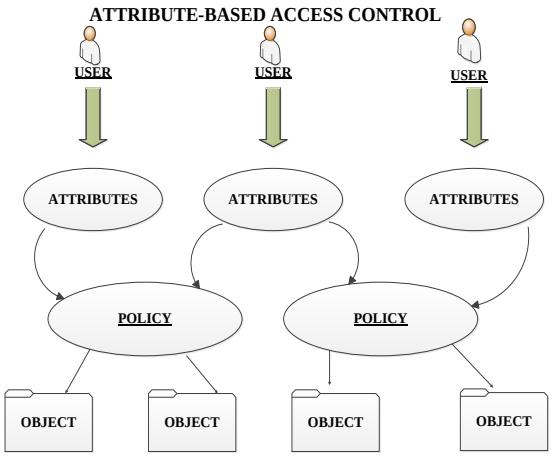

Figure 4: Example of the ABAC model in use

2.5 Mandatory Access Control

According to Joint Task force Transformation Initiative et al. [15], MAC grants all objects and subjects security labels, an example being top secret, secret, and confidential. For a subject to be granted access to an object, the subject must have the same security label as the object, a need to know, and formal access approvals. If the subject has the same security label as an object but does not have the same need to know or access approvals, the subject cannot access the object. If the subject has a higher security label than the object and a need to know, the subject will be granted access to the object. However, if the object has a higher security label than the subject, the subject will not be granted access to the object, an example of this can be seen in in Table 1. The subject who is granted access to an object is prohibited from showing unauthorized subjects or objects. The MAC model is designed to take away the transitive trust flaw that the DAC model presents. The subject who is granted access to an object is also prohibited from changing the access control rules of an existing object/subject, granting the security labels, need to know, and access approvals for modified or newly created documents. The MAC model has been adopted and enforced by many government agencies.

Table 1: Shows the MAC model in practice.

SUBJECT	OBJECT	NEED TO KNOW	ACCESS
SECRET	TOP SECRET	NO	NOT GRANTED
SECRET	SECRET	YES	GRANTED
TOP SECRET	SECRET	YES	GRANTED
SECRET	SECRET	NO	NOT GRANTED
TOP SECRET	SECRET	NO	NOT GRANTED
SECURITY LABELS	SECURITY LABELS		

3 DEPLOYMENT

When the deployment of RuBAC is considered alone, Kern and Walhorn [16], suggest that it does not seem feasible for an organization-wide deployment. It can be difficult to maintain RuBAC long term due to the ever-changing impact of rule changes. Using RuBAC in combination with RBAC offers the most advantages to an organization. The combination allows an organization to assign a set of rules to each role and then assign subjects to said roles. If the subject changes roles, the subject is required to be taken out of the old role group and added to the new role group. If an organization changes its rules for a specific role, it can alter that role's rules, and all the subjects in that group's privileges are adjusted automatically.

SAM Jupiter is recommended for use by Franco et al. [17], as it combines RuBAC with RBAC. Once SAM Jupiter is installed and systems are connected, their data will be uploaded to a central repository where administrators will

work out of. All administrative changes will be made from SAM Jupiter, changes will be automatically deployed in the underlying systems. All access privileges are administered using RBAC and have been extended to support Enterprise Roles (ER). ER are all privileges needed for a specific role and span across different systems. After a user is assigned a role, the privileges are then applied to the target system (TS), as shown in Figure 5. With SAM Jupiter's RBAC and ER deployment, it can reduce administrators' time through automation and simplify managing a large organization's privilege.

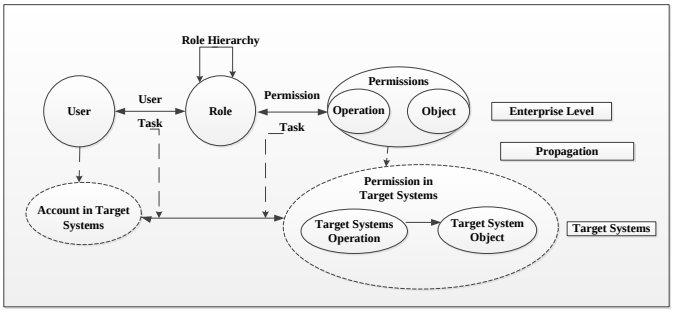

Figure 5: Enterprise Role-Based Access control (redrawn) [17]

A combination of MAC and RBAC is possible with Security Enhanced Linux (SELinux) and is a great deployment option adopted by more than just the military and health information systems (HIS). In a study conducted by Franco et al. [17], SELinux was used to deploy MAC on HIS operating systems. SELinux was chosen as the operating system for HIS because of its many security features that protect sensitive information. The U.S. National Security Agency originally developed SELinux, and in 2000 it was made freely available to the public for implementation. MAC was required for HIS to minimize the DAC flaws by not allowing users to grant privileges to objects and minimize damage if a compromise occurred. SELinux deploys Type Enforcement (TE), a type of MAC that uses a domain for subjects with object classes and types. SELinux also builds upon TE and adds RBAC to group types with users, but ultimately the decisions are made from the TE. SELinux also increases user accountability but manages user identification between Linux User Identification (UID) and SELinux identifiers. SELinux offers flexible policy configuration, fine-grained access controls, and orthogonal user identifiers to make SELinux a great deployment option. However, its configuration can be a grand task and should be done by someone with a high level of expertise, which is not commonly found in HIS.

According to Kuhn et al. [18], combining RBAC and ABAC benefits can provide effective access control for rapidly changing environments. A combined approach was chosen because many attributes are considered static, while others are dynamic. Dynamic attributes consisting of time of day and static being office location, position, etc. An organization should choose a role structure that focuses on the static attributes instead of more dynamic ones, to avoid awkward designs. By focusing on static attributes, employing them in the role decisions can drastically reduce the number of dynamic rules, which maintains the benefits of simplicity, convenience, and time efficiency in determining a subject's privileges.

Any of these options can be a great deployment option when privilege or permission creep is addressed. A tool has been developed by Parkinson et al. [19], that addresses this issue by monitoring and auditing for cases where privilege creep has occurred. The auditing tool's name is Creeper, and it was developed for Microsoft systems, but offers great flexibility and scalability as it can be deployed in other systems as well. In an empirical analysis, Creeper was compared to a competitor called ntfs-r and a human auditor that had 10 years of security audit experience. Table 2 shows the characteristics of the real-world file systems used in the analysis and Table 3 presents a comparison of the findings. The results indicated that the accuracy rates of correctly identified privilege creep instances were 55% for the human analysis, 60% for ntfs-r, and 98% for Creeper. The human analyst correctly identified every instance of privilege creep but failed to audit the remaining cases due to the study's time constraints. That is where Creeper excels, as it is an automated process that is faster and more accurate than some competitors and human analysts.

Many smaller companies struggle to implement SOD, according to a study conducted by

Gramling et al. [20]. Out of 700 companies, with market values less than 75 million, 358 had SOD problems; Where companies with a market value greater than 75 million were categorized as large companies, only 30 large companies out of a similar sample size had SOD problems. Identifying that smaller companies struggle to implement SODs. Of the 358 smaller companies identified to have SOD problems, 116 were used as the sample size. The 116 smaller companies' management reports were analyzed to determine why SOD is a weakness. The findings presented in Table 4 show that 90 companies identified SOD problems due to being short-staffed. With the specific areas mentioned all involving financial payments, where SOD issues can consequentially result in unauthorized purchases and/or fraud. Of the 116 companies, 18 of them mentioned compensating controls that involved a form of review, either internal or 3rd party.

Table 2: Characteristics of the real-world file systems (redrawn) [19]

Dataset	Total Directories	ACLs Inspected	Total Permission Levels
1	168	42	8
2	254	24	3
3	570	74	6
4	3,517	499	23
5	11,654	870	15

Table 3: Comparison of the findings (redrawn) [19]

	Creeper			ntfs-r			Human Auditior		
Dataset	C	I	M	C	I	M	C	I	M
1	0	0	0	0	0	0	0	0	0
2	14	1	0	8	7	7	8	0	7
3	0	0	0	0	0	0	0	0	0
4	10	0	0	6	0	4	2	0	8
5	3	0	0	3	79	0	0	0	3
C = Correct		I = Incorrect		M = Missed					

To solve the identified issues of smaller companies and SOD, Gramling et al. [20], suggests the obvious solution of adding more people. From a financial standpoint, adding more people may not be feasible for smaller companies, and that may be why some companies have added compensating controls. Although compensating controls are better than no controls at all, they should not be relied on. Relying solely on compensating controls will consume more resources and require more time to recover, investigate, and prevent issues, than applying SOD would take. Management oversight, job rotation, and mandatory vacation are recommended when adding employees are not feasible. Management oversight can be implemented to review, approve, and oversee functions that would otherwise be mitigated with SOD.

Table 4: Identified SOD Weaknesses (redrawn) [20]

Segregation of Duties Vulnerabilties Summary for 116 Companies	
Causes for Segregation of Duties Vulnerabilites*	**Companies**
Lack of employees	90
Nonspecific Segregation of duties issue	22
Limited to 1-2 directors or officers	7
Other	2
Accounts Mentioned or Specific Areas	
Cash disbursements	6
Cash disbursements	5
Invoice approval/accounts payable	3
Purchases	3
Period-end closing process	3
Compensating Controls Mentioned	
Independent reviews, management/board review, and reconciliations	14
Third-party review	4
*** Some companies are included in multiple categories.**	

According to Snyder and Dietz [21], implementing job rotation and mandatory vacation policies, where employees are required to take a minimum of five vacation days in a row annually, is a smart policy. While the employee is off on vacation, a new employee should be assigned their task. There are multiple benefits of deploying this, as it gives assurance that if an employee was to leave the organization, another could fill the gap. It also helps prevent fraud as the employee on vacation is aware that their work is being reviewed and exposed if fraud has occurred. Even if fraud is not occurring, many policies will help identify errors and highlight areas of a process that need improvement.

When it comes to implementing job rotation in small companies Snyder and Dietz [21], suggest that even though it may seem difficult to implement, it is feasible. It is recommended to focus on areas that involve money as those are the areas that fraud is most likely to occur. By rotating simple tasks, such as making cash deposits and protecting checks that have been received in the mail, a company can have any

employee rotate without much previous knowledge of the task

4 CONCLUSION

When it comes to protecting an organization, one can never take too many precautions. The policy of least privilege paired with SOD should always be implemented to protect from insider threats and reduce the effects of a compromised account. When choosing a deployment model, DAC should be avoided as there are numerous vulnerabilities consisting of the owner being able to grant privileges instead of the organization. Some of these vulnerabilities can be seen through the transitive read trust and privilege creep. RBAC has many benefits, as after roles are created, it is simple to add a subject to a role but must be monitored for privilege creep as subjects switch roles within an organization. RuBAC can be very time consuming for administrators and should be avoided unless paired with RBAC. MAC model is a valid option as it takes away the transitive trust downfall that the DAC model presents. It is widely adopted in many military organizations and even in HIS when paired with RBAC. ABAC offers a more granular option than RBAC but should be combined with RBAC by having roles as an attribute if role rules are focused on static rules and not dynamic.

No matter what least privilege and SOD deployment model is chosen, an organization should take the time to audit access privileges because no matter what model is deployed, it must be maintained as subjects switch roles over time. Automated auditing tools, such as creeper, should be installed to monitor privilege creep. Automated tools should be implemented instead of human auditors, specifically Creeper, as it is much faster and more accurate than a human auditor. It will reduce time and help maintain SOD for large and small companies alike.

Many smaller companies struggle to deploy SOD as hiring more employees may not be finically possible. It is recommended that smaller companies implement compensating controls consisting of mandatory vacation and job rotation to better assist. Mandatory vacation and job rotation will help prevent fraud by having employees cover for one another while they are away for a minimum of five days. This will assure the company that if an employee leaves the organization, the position can be covered until additional help is hired, and it will shed light on areas of concern. Smaller companies that still might struggle to implant job rotation and mandatory vacation should prioritize implementing these compensating controls in areas that involve money. Implementing them in a simple task, for example, deposits and receiving checks in the mail, employees with no prior experience should be able to successfully rotate.

In future studies, it can be recommended to expand on non-traditional deployment models of least privilege and SOD as well as the deployment methods for different operating systems. Researching what fields of business benefit from which model more would be beneficial. Determining the best software to deploy least privilege and SOD will also benefit large and small companies. Researching other privilege Creep auditing tools will benefit the deployment.

REFERENCES

1. Claycomb, W., Huth, C. L., Flynn, L., McIntire, D. M., & Lewellen, T. (2012). Chronological examination of insider threat sabotage: Preliminary observations. J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl., 3, 4-20.
2. Al-Ahmad, W. (2013). A detailed strategy for managing corporation cyber war security. International Journal of Cyber-Security and Digital Forensics, 2(4), 1-9.
3. Soluade, O. A., & Opara, E. U. (2014). Security breaches, network exploits and vulnerabilities: A conundrum and an analysis. International Journal of Cyber-Security and Digital Forensics, 3(4), 246-261.
4. Brickley, J. C., Thakur, K., Kamruzzaman, A. S. (2021). A comparative analysis between technical and non-technical phishing defences. International Journal of Cyber-Security and Digital Forensics, 10(1), 28-41.
5. Saltzer, J. H., and Schroeder, M. D. (1975). "The protection of information in computer systems," in *Proceedings of the IEEE*, 63(9), 1278-1308., https://doi.org/10.1109/PROC.1975.9939.
6. segregation of duties: Integrating theory and practice for manual and IT-supported processes. International Journal of Accounting Information System, 15(4), 304-322.304-322, https://doi.org/10.1016/j.accinf.2014.05.003 .
7. Islam, A., Corney, M., Mohay, G., Clark, A., Bracher, S., Raub, T., & Flegel, U. (2011). Detecting collusive

fraud in enterprise resource planning systems. (eds) Advances in Digital Forensics VII. DigitalForensics 2011. IFIP Advances in Information and Communication Technology, vol 361. 143-153. https://doi.org/10.1007/978-3-642-24212-0_11.

8. Hu, V. C., Ferraiolo, D. F. and Kuhn, D. R. (2006). Assessment of access control systems, NIST Report 7316, http://www.csrc.nist.gov/publications/nistir/7316/NISTIR-7316.pdf
9. Debreceni, C., Bergmann, G., Ráth, I. Z., & Varró, D. (2016). Deriving effective permissions for modeling artifacts from fine-grained access control rules. In: 1st International Workshop on Collaborative Modelling in MDE, Saint Malo, France. ACM
10. 2017). Towards efficient evaluation of rule-based permissions for fine-grained access control in collaborative modeling.
11. Ferraiolo, D. F., Cugini, J. A., & Kuhn, D. R. (1995). Role-Based Access Control (RBAC): Features and Motivations. Proceedings of the 11th Annual Computer Security Applications Conference, pp. 241-248.
12. Weil, T., & Coyne, E. (2013). "ABAC and RBAC: Scalable, Flexible, and Auditable Access Management" in IT Professional, vol. 15, no. 03, pp. 14-16, https://doi.org/10.1109/MITP.2013.37
13. Panende, M. F., Prayudi, Y., & Riadi, I. (2018). Comparison of Attribute Based Access Control (ABAC) Model and Rule Based Access (RBAC) to Digital Evidence Storage (DES). Int. J. Cyber-Security Digit. Forensics, 7(3), 275-282.
14. Bhatt, S., Patwa, F., & Sandhu, R. (2016). An attribute-based access control extension for openstack and its enforcement utilizing the policy machine. In 2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC) (pp. 37-45). IEEE. https://doi.org/10.1109/CIC.2016.019.
15. Joint Task Force Transformation Initiative & National Institute of Standards and Technology (U.S.). (2013). Security and privacy controls for federal information systems and organizations https://doi.org/10.6028/nist.sp.800-53r4
16. Kern, A., & Walhorn, C. (2005). Rule support for role-based access control. In Proceedings of the tenth ACM symposium on Access control models and technologies (pp. 130-138).DOI: https://doi.org/10.1145/1063979.1064002
17. Enhanced Linux to Enforce Mandatory Access Control in Health Information Systems," in Proceedings of the 2 Australasian Workshop on Health Data and Knowledge Management - Volume 80, Wollongong, NSW, Australia, pp. 27-33.
18. Kuhn, D. R., Coyne, E. J., and Weil, T. R. (2010). "Adding Attributes to Role Based Access Control," Computer, vol. 43, no. 6, pp. 79–81.
19. Parkinson, S., Khan, S., Bray, J., & Shreef, D. (2019). Creeper: a tool for detecting permission creep in file system access controls. Cybersecurity, 2(1), 1-14.https://doi.org/10.1186/s42400-019-0031-1
20. Gramling, A., Hermanson, D., Hermanson, H. M., & Ye, Z. (2010). Addressing Problems with the Segregation of Duties in Smaller Companies.
21. risk of financial fraud. Family Practice Management, 13(2), 45-48.