Proceedings of the Fifth International Conference on Intelligent Computing and Control Systems (ICICCS 2021)
IEEE Xplore Part Number: CFP21K74-ART; ISBN: 978-0-7381-1327-2

Identifying Considerable Anomalies and Conflicts in ABAC Security Policies

K.Vijayalakshmi
Vels Institute of Science, Technology and Advanced Studies, Chennai, India
Arignar Anna Government Arts College, Cheyyar
Email: vijiyuvavelan@gmail.com

Dr.V.Jayalakshmi
School of Computing Sciences
Vels Institute of Science, Technology and Advanced Studies
VISTAS, Chennai, India
Email: jayasekar.scs@velsuniv.ac.in

2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS) | 978-1-6654-1272-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICICCS51141.2021.9432162

***Abstract*— Nowadays security of shared resources and big data is an important and critical issue. With the growth of information technology and social networks, data and resources are shared in the distributed environment such as cloud and fog computing. Various access control models protect the shared resources from unauthorized users or malicious intruders. Despite the attribute-based access control model that meets the complex security requirement of todays' new computing technologies, considerable anomalies and conflicts in ABAC policies affect the efficiency of the security system. One important and toughest task is policy validation thus to detect and eliminate anomalies and conflicts in policies. Though the previous researches identified anomalies, failed to detect and analyze all considerable anomalies that results vulnerable to hacks and attacks. The primary objective of this paper is to study and analyze the possible anomalies and conflicts in ABAC security policies. We have discussed and analyzed considerable conflicts in policies based on previous researches. This paper can provide a detailed review of anomalies and conflicts in security policies.**

***Keywords*— Access control models, Anomalies, Attribute based access control model, Big data, Conflicts, Policy validation, Security Policy.**

I. INTRODUCTION

New and advanced computing technologies are developed rapidly through the easy availability and speedy connection of the internet. As information technology is growing well, the range of users and resources is also increased exponentially [1]. Due to many security issues, threats, and attacks, protecting privacy and resources is a critical job. The users' information and resources are shared in this new distributed computing environment. Securing data and resources and maintain users' privacy in this dynamic and distributed computing era is a big challenging task [2]. The intrusion detection security system implements access control models to protect the shared resources. Various access control models have been developed in past decades. Some of them give better performance while some are failing in securing the resources. The discretionary access control model (DAC) maintains an Access Control List (ACL) for each resource to specify the access rights of the resource. ACL is the list of records and each record specifies the user and his access rights on the resource. In DAC, the owner of the resource is capable of specifying ACL for his resource. DAC is good when the number of users and resources is small [3]. The mandatory access control model also performs well in securing shared resources if the communications between users and resources are low [4]. Multilevel database systems, military, and government organizations use MAC to secure the resources. MAC uses security labels for both user and resource and it allows the user to do the requested operation on the resource if the labels are matched only. The role-based access control (RBAC) model consists of two mappings. The first mapping is role-permission which assigns the access rights for the role (admin or manager) and the second is role-user which assigns the role to the user. Hence a user can get the access rights that are eligible for his role only, no other permissions are allowed [5]. Despite RBAC protects shared resources even the number of users and resources is high, there was difficult to address the complex security requirements of today's dynamic and growing computing environment. The attribute-based access control (ABAC) model specifies the security policies with the attributes of the subject (who requests the access), object (shared resource), and the environment conditions (date, time, and others) [6]. Based on our previous research, the ABAC model satisfies the security issues of these emerging advanced technologies such as cloud, fog, edge, and IoT computing [7]. ABAC is an efficient and convenient access control model to specify and update the security policies dynamically.

Anomalies and conflicts in security policies degrade the efficiency of the security system. The detection of possible anomalies and conflict in policy is an important and difficult task. Even many machine learning algorithms are proposed to detect anomalies; the detection of anomalies or conflicts is still imperfect due to the dynamic and fast generation of security policies. This paper describes the anomalies and existing approaches used to detect the anomalies. The redundancy of policies and conflict in the security policies are considerable anomalies. The redundancy of more policies decreases the performance of the access control model. The conflict in policies may give the wrong decision that the restricted user can avail the access to the resources. Hence Detecting and eliminating anomalies and conflicts is a critical task. We presented the previous research works done on anomaly detection in section-II. We use section-III to discuss the fundamental concepts of the ABAC model and section-IV to identify, describe, and analyze the anomalies in ABAC security policies. We made a conclusion in section V.

II. RELATED WORK

Since the importance of security systems, many research works have been done and doing to detect and address the anomalies in policies. Randa Aljably and Yuan Tian proposed an approach in machine learning to detect anomalies. This dynamic approach consists of services of machine learning concepts to detect anomalies and enhance the performance of the access control model [2]. This approach combines both supervised and unsupervised learning concepts. In this approach, the anomalies and abnormalities are monitored and updated quickly if any issues are detected. Maryem Ait El Hadj and his team have described more about the anomalies and proposed a model to detect and rearrange the security policies [8]. They use the clustering technique to cluster similar rules and proposed the detection and resolution algorithm to detect and eliminate or modify the clusters. The resolving process of clusters rather than each rule increases the performance of their approach. E. Lupu and M. Sloman proposed the first approach in this concept [9]. They detected two types of conflicts modality-conflict and application-specific- conflict. The conflict modality occurs when there are positive and negative security rules and this conflict can be resolved by updating the rules. The application-specific conflict occurs due to the external or environmental activities that can be addressed by specifying meta-policies. They used the relationship of subjects to address the overlapping requests. Their approach detects all possible conflicts at design time, not at run time.

Jonathan D. Moffett and Morris S. Sloman made research on distributed management policies not in ABAC security Policies. They categorized several conflicts in management policies where some of the conflicts are applicable in the access control security policies [10]. They have discussed elaborately all possible conflicts with clear examples. They proposed a framework to analyze conflicts in policies with theoretical evidence.

TABLE I. PREVIOUS RESEARCH CONTRIBUTIONS IN IDENTIFYING AND VALIDATING POLICY ERRORS.

References	Proposed work
R. Aljably, et al. 2020 [2].	Proposed a machine learning approach that incorporates both supervised and unsupervised learning concepts. This approach detects and resolves the errors in policies quickly.
M. Ait El Hadj, et al. 2018 [8].	Proposed a cluster-based approach to validate possible and potential errors in ABAC security policies. They used XACML (eXtensible Access Control Markup Language) to specify the ABAC policies. They detect and give resolution for each cluster of rules rather than every rule. The detection of anomalies in only clusters increases the performance of the approach
E. C. Lupu, et al. 1999 [9].	One of the early proposed approaches. This research categorized the errors in policies as conflict-modality and application-specific conflict. This approach validates the first category by changing the syntax of policies and validates the second anomaly by specifying meta-policies. Validation of policies is done at design time
J. D. Moffett, et al.1994 [10].	Made an analysis in distributed management policies. They discussed and analyzed many conflicts in management policies and gave the resolution to validate all conflicts
M. Erradi, et al. 2018 [11]	This method detects and resolves anomalies in only doubted rules, not in all rules. The suspicious or doubted rule is identified by using access-log. The detection and addressing conflicts in only doubted rules increase the performance of the system
R. A. Shaikh, et al. 2017 [12]	Proposed a new approach to validate policy sets. They list two types of conflicts inconsistency (same rules have different action or permission) and incompleteness (no rule is defined for a particular operation)
Vijayalakshmi, K, et al. 2020 [13]	Proposed and priority-based approach to cluster the ABAC policies. Discussed anomalies and conflicts in ABAC policies
M. Yahiaou, et al. 2018[14]	Proposed a framework to identify and resolve policy conflicts in the ABAC system
C. Morisset, et al. 2019) [15]	Presented a framework for policy validation and used Binary Decision Diagram (BDD) to represent the policies. This work is the implementation of three real-world policies
This paper	This paper aims to identify, describe and analyze all possible errors in policies and the strategies used to validate the policies.

Yahya Benkaouz and Ahmed Khoumsi have proposed an efficient approach to detect anomalies and conflicts in ABAC policies. This method detects and resolves anomalies in only doubted rules, not in all rules [11]. The suspicious or doubted rule is identified by using access-log. The detection and addressing conflicts in only doubted rules increase the performance of the system. Riaz Ahmed Shaikh and his team proposed an approach to detect errors or anomalies in policy sets[12]. They categorize the errors in policies into two categories inconsistency and incompleteness. Their approach validates both the two categories of anomalies simultaneously. Their approach has the capacity to validate redundancy rules. In our previous work, we proposed a clustering algorithm to cluster similar rules based on the similarity value of rules and priority of policies. We have discussed briefly considerable conflicts and anomalies in ABAC policies [13]. The scope of this work is to identify, define, describe policy errors and express equation for all considerable anomalies. In this paper, we have described and analyzed all possible anomalies and conflicts in ABAC policies and also we discussed some strategies to detect and address this issue. This is a contribution to our research framework to enhance the ABAC model. Table I. summarizes what the works are done previously in analyzing anomalies and conflicts in policies.

III. BACKGROUND

This section presents the basic concepts of security systems and various access control models. We discussed the working mechanism of the ABAC model and the specification of ABAC security policies.

A. *Access Control Models*

As data and other resources are shared in this new advanced distributed computing environment to provide ease of access, availability of resources, and everything to the user, preserving confidentiality, the integrity of data, and other resources is a big and important task. The access control model is a protection mechanism that monitors, identifies, and prevents unauthorized intrusion based on security policies or constraints. Fig. 1 illustrates the role of the access control model.

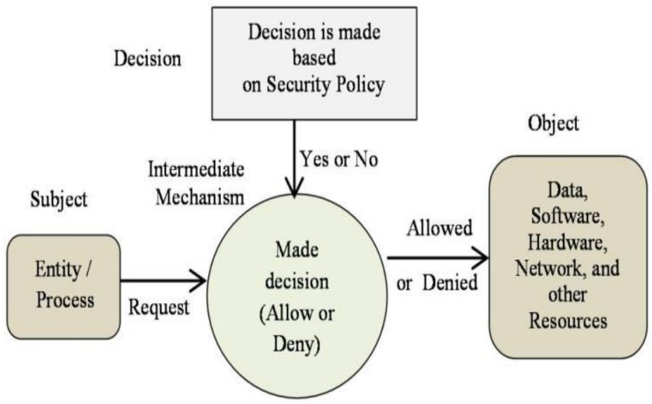

Fig. 1 The role of the access control model

The security system consists of Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) to secure the shared resources and big data. The role of IDS is a continuous process and it monitors the traffic in the network and identifies and alerts the security system when there is any suspicious user or activity occurred [16]. While the IDS detects the attacks, the IPS prevent or block the suspicious attacks of unwanted intruders [17]. The security system implements any one of the access control models including a traditional authentication system to increase efficiency.

Many access control models have been proposed. The DAC allows the owner of the object to specify the policy set. This model gives full control to the owner of the object. Based on his own discretion, he/she can give access rights to the user. Thus the user or others cannot grant privileges of other's object. DAC maintains an access control list (ACL) for each object. Thus the ACL of each object contains a set of records and each record specifies the access rights of the particular subject. It is good in small-scale applications and organizations. The mandatory access control model (MAC) establishes security labels for each subject and object. This model allows the subject's request if the label of the subject and the object matches only. Despite the MAC give a better protection mechanism than the DAC, but it also fails to satisfy todays' computing security issues.

The Role-based access control model uses two mappings between permission-role and role-user. The RBAC assigns possible access rights to the roles first. The user can get the access rights that are applicable for his role only. The RBAC mechanism performs well and meets the security requirements of large-scale enterprises and organizations. The RBAC is inefficient in expressing policies in a dynamic and fast-growing distributes environment. The ABAC is a dynamic, fine-grained, and next-generation security model. It fulfills the security needs of new computing technologies. The ABAC sets the security rules with the attributes of the subject, object, and environmental conditions. This model allows or denies the subject's request based on the attributes specified in the policies only. The errors in security policies are referred to as anomalies or conflicts. The policy errors cause the intrusion of malicious users and attacks. Hence it is an important task to detect and give resolution for policy errors. This process is called policy validation.

In our previous work, we analyzed based on the features granularity, flexibility, efficiency, and security level. The term granularity refers to how accurately the model protects unauthorized access. The property flexibility defines how the admin can easily generate or update the policy set. The term efficiency refers to how fast and rightly the decision is made. The security level refers to how the resources are secured and protected efficiently and reliably. We summarized the analysis of discussed four primary access control models in Table II.

TABLE II. SUMMARY OF THE ANALYSIS FOUR ACCESS CONTROL MODELS (DAC, MAC, RBAC, AND ABAC)

Access control model	Granularity	Flexibility	Efficiency	Security level
DAC	Good at small scale applications	Good	Poor	Low
MAC	Good at small scale applications	Good	Poor	Better than DAC
RBAC	Good at large scale applications	Good	Good	Good
ABAC	Good at today's computing technologies and big data	Good	Good	Good

B. Attribute Based Access Control Model

ABAC specifies policy sets that contain the security rules for a protection mechanism. ABAC allows or denies the users' or applications' request for access to resources based on attributes specified in security policies. The security rule is expressed with the attributes of the subject, object, and environmental conditions. The traditional authentication system verifies the originality or identity of the subject with limited information (username and password), that process is not sufficient to protect the shared big data or resources in the well growing and large distributed environment. ABAC stores and uses more attributes of all categories to validate the identity of the original users that improves the level of security.

The definition of the ABAC model comprises the following important terms. An attribute is specified with two pieces of information name of the attribute and the value of the attribute. For example 'department' is the attribute name and 'hematology' is the value of the attribute 'department' {department=hematology}. A subject is an application or user who requests access to a shared resource. The attributes of the subject are determined by the properties or characteristics of the subject. For example, department and designation are the attributes of the subject. An object is a shared resource that is demanded by the subject to perform a desired operation on the resource. Some examples of the attributes of the object are resource-name and resource-type and it can be expressed as {resource-name=pat005-blood-report, resource-type=file}.

Environmental conditions: Environmental attributes are determined by the characteristics of situation, subject, and object. the operational or situational context in which access requests occur. Environment conditions are detectable environmental characteristics that may include the current time, day of the week, or location of a user. For example date or time is an attribute of the Environmental category. An operation is the execution of a function at the request of a subject upon an object. Operations include read, write, edit, delete, copy, execute, and modify. Security rule is constructed with the attributes of the categories subject, object, and environmental conditions. The policy is the representation of rules or relationships that makes it possible to determine if requested access should be allowed, given the values of the attributes of the subject, object, and possibly environment conditions.

C. Specification of security rules, policy, and policy sets

Generally, the ABAC security model consists of security policy sets. The ABAC security rules are constructed with majorly three categories subject, object, and environmental conditions. The ABAC security policy is a set of security rules and the ABAC policy set is a set of policies [18]. The following example shows how to express the ABAC security policy using XACML (Extensible Access Control Markup Language).

```
<PolicySet>
  <Policy PolicyID="P1">
   <Rule RuleID="R1" Decision="Allow" >
        <Operation>
          <Operation-1>read</Operation-1>
        </Operation>
      <Subject>
           <Department>Hematology</Department>
          <Designation> Nurse <Designation>
        </Subject>
       <Object><ResourceName>
           PatID_007_Blood_CBC_Report
             </ ResourceName >
        </Object>
         <EnvironmentalCondition>
           <Duration>7:19</Duration>
          </EnvironmentalCondition>
       </Rule>
             ……….    // more rules can be specified
    </Policy>
      ……….. // more policies can be specified
  </PolicySet>
```

In the above example, the policy set contains many policies including Policy P1 which has many rules along with the rule R1. The rule R1 allows the persons who are all working as a 'nurse' in the department 'Hematology' to read a file 'PatID_007_Blood_CBC_Report' during the working hours 7:19. In literally, the security rules can be expressed in the following understandable form. We used the additional parameter PriorityLevel that indicates the priority value of the rule and it is used to solve the problem conflict-demand.

R1= {$Allow_{read}$ | Designation = {Nurse}, Department = {hematology}, File-Name = { PatID_007_Blood_CBC_Report}, Time= {9:19}, PriorityLevel=2}

We express the security rule as follows [18].

R= {Xop | PR}

The decision X (allow, deny) is made for a request of operation (read, write) based on the predicate PR.

PR= {atr_1 € $Vatr_1$, atr_2 € $Vatr_2$, …., atr_n € $Vatr_n$ ^ Priority-level = non-empty- integer-value}

So rule R can be written as

R= {Xop | $Vatr_1$, atr_2 € $Vatr_2$, …., atr_n € $Vatr_n$ ^ Priority-level = non-empty-integer-value}

atr_1,atr_2,…,atr_n are the list of attributes belonging to any category(Subject, Object, Environmental Conditions). $Vatr_1$, $Vatr_2$, $Vatr_n$ are the set of permitted values of the attributes atr_1,atr_2,…,atr_n respectively.

For example, in the following rule,

R= {$allow_{read}$ | Designation = {Nurse, LabTechnician}, Department={Hematology}, FileName = { PatID_007_Blood_CBC_Report }, Time={07.00-19.00}^Priority-level=1}

The attributes Designation and Department are attributes of Subject, Filename is the attribute of Object, and Time is the attribute of Environmental conditions. This rule has priority 1.

IV. ANOMALIES AND CONFLICTS

This section discusses considerable anomalies and conflicts in ABAC policy sets. We identify, define, and describe with sample examples of possible and critical policy errors. The anomalies and conflict in the ABAC security policies is the main problem that should be resolved to improve the performance of the access control model. ABAC model is a flexible and efficient one to meet today's computing technologies. This is good in protecting the big data and other shared resources in clouds. Despite this model addresses the security requirements of today's complex environment, the anomalies, and conflicts in the security policies degrade the performance of this mechanism. Hence it is an important and critical issue and must be rectified. In this paper, we analyzed the following anomalies and conflicts.

A. Rules-Redundancy

Definition 1: The anomaly rules-redundancy occurs if and only if the decisions of the two rules R1 and R2 are the same (X(R1)=X(R2)), request of the task or operations are same (T(R1)=T(R2)) and either R1 is a subset of R2 or R2 is a subset of R2 (R1 ⊆ R2 or R2 ⊆ R1). This can be expressed by equation-1.

$$\text{Rules-Redundancy (R1,R2)} = \begin{cases} \text{True} & X(R1) = X(R2) \text{ and } T(R1)=T(R2) \text{ and } (R1 \subseteq R2 \text{ or } R2 \subseteq R1) \quad \text{-----(1)} \\ \text{False} & \text{Otherwise} \end{cases}$$

X(R1) is the decision of the rule R1 and X(R2) is the decision of R2. Thus the decision of the two rules are the same, the request of operations are same, and the values of the attributes of R1 is a subset of R2 or the values of the attributes of R2 is a subset of R1 causes the rules-redundancy anomaly. If the rule R1 is a subset of R2, then the rule R1 is removed, otherwise, R2 is removed from the policy set. The rule-redundancy anomaly results in storage-space consumption and degrades the performance of the model. Table. III illustrates the rules-redundancy anomaly.

Example 1: In Table. III, when comparing the rules R1 with R3, The decisions X(R1)=X(R2) (allow=allow), the operations T(R1)=T(R2) (read=read), and R1 is a subset of R2 ({Hematology}⊆ {Hematology}, {Nurse}⊆ {Nurse, LabTechnician }, {PatID_007_Blood_CBC_Report} ⊆ { PatID_007_Blood_CBC_Report }, and {07:19}⊆{07:19}), the anomaly rule-redundancy is occurred and the minimum (R1⊆R2) rule R1 is removed from the policy set. Likewise There is rule redundancy between R4 and R2, The minimum (R4⊆R2) rule R4 is removed.

TABLE III. SAMPLE RULES - EXAMPLE OF RULES-REDUNDANCY

Rules	Decision (X)	Operations to be performed	Attributes			
			Department	Designation	FileName	Time
R1	allow	read	Hematology	Nurse	PatID_007_Blood_CBC_Report	07:19
Resolution : - T(R1)=T(R2) and X(R1)=X(R2) and R1⊆R2 - Rule_redundancy - R1 is removed						
R2	allow	read	Hematology	Nurse, Lab Technician	PatID_007_Blood_CBC_Report	07:19
Resolution: No redundancy						
R3	allow	read	Cardiology	Dietician	PatID_007_Blood_CBC_Report	08:18
Resolution: No redundancy						
R4	allow	read	Hematology	Lab Technician	PatID_007_Blood_CBC_Report	07:19
Resolution : - T(R4)=T(R2) and X(R4)=X(R2) and R4⊆R2 - Rule_redundancy - R4 is removed						

B. Rules-Discrepancy

Definition 2: The anomaly rules-discrepancy occurs if and only if the decisions of the two rules R1 and R2 are not the same (X(R1)=X(R2)), request of the task or operations are same (T(R1)=T(R2)) and either R1 is a subset of R2 or R2 is a subset of R2 (R1 ⊆ R2 or R2 ⊆ R1). This can be expressed by equation-2.

Proceedings of the Fifth International Conference on Intelligent Computing and Control Systems (ICICCS 2021)
IEEE Xplore Part Number: CFP21K74-ART; ISBN: 978-0-7381-1327-2

$$\text{Rules-Discrepancy }(R1,R2) = \begin{cases} \text{True} & X(R1) \neq X(R2) \text{ and } T(R1)=T(R2) \text{ and } (R1 \subseteq R2 \text{ or } R2 \subseteq R1) \quad \text{----- (2)} \\ \text{False} & \text{Otherwise} \end{cases}$$

$$\text{Rules-Inadequacy}(P) = \begin{cases} \text{True} & \{a_1,a_2,..,a_n\} \varepsilon\ S \subseteq \{R1 \text{ or } R2 \text{ or } \ldots Rm\} = \theta \quad \text{---- (3)} \\ \text{False} & \text{Otherwise} \end{cases}$$

X(R1) is the decision of the rule R1 and X(R2) is the decision of R2. Thus the decision of the two rules are not the same, but the request of operations are the same, and the values of the attributes of R1 is a subset of R2 or the values of the attributes of R2 is a subset of R1 causes the rules-discrepancy anomaly. In this case, the admin should verify the truth and facts, based on the facts, either R1 is reconstructed or R2 is reconstructed. The rule-discrepancy anomaly results in a critical security issue that should be eradicated.

TABLE IV. SAMPLE RULES – EXAMPLE OF RULES-DISCREPANCY

Rules	Decision (X)	Operations to be performed	Attributes			
			Department	**Designation**	**FileName**	**Time**
R1	Deny	read	Hematology	Nurse	PatID_007_Blood_CBC_Report	07:19
Resolution: - X(R1)≠X(R2) and T(R1)=T(R2) and R1⊆R2 - Rule_discrepancy - Either R1 is reconstructed or R2 is reconstructed						
R2	allow	read	Hematology	Nurse, Lab Technician	PatID_007_Blood_CBC_Report	07:19
Resolution: - X(R1)≠X(R2) and T(R1)=T(R2) and R1⊆R2 - Rule_discrepancy - Either R1 is reconstructed or R2 is reconstructed						

Example 2: In Table. IV, when comparing the rules R1 with R3, The decisions X(R1)≠X(R2) (deny=allow), the operations T(R1)=T(R2) (read=read), and R1 is a subset of R2 ({Hematology}⊆{Hematology}, {Nurse}⊆ {Nurse, LabTechnician}, {PatID_007_Blood_CBC_Report} ⊆ {PatID_007_Blood_CBC_Report}, and {07:19}⊆{07:19}), the anomaly rule-discrepancy is occurred. In this case, either R1 is reconstructed or R2 is reconstructed based on the facts by the admin or cloud service provider.

C. *Rules-Inadequacy*

Definition 3: Suppose the authorized subject S with the values of attributes $\{a_1,a_2,..,\ a_n\}$ requests to perform a particular operation on the object, there are no rules specified in the policy set to make a decision, Thus the rules are not defined adequately and completely. This can be expressed by equation-3.

Equation-3 states that there are no rules specified in the policy set P to make a decision when a subject S requests access for an object. {R1 or R2 or …Rm} is the set of rules in the policy set P.

TABLE V. SAMPLE RULES - EXAMPLE OF RULES-INADEQUACY

Rules	Decision (X)	Operations to be performed	Attributes			
			Department	**Designation**	**FileName**	**Time**
R1	allow	read, share	Hematology	Duty-doctor, Chief-Doctor	PatID_007_Blood_CBC_Report	08:18
R2	allow	read	Hematology	Nurse, Lab Technician	PatID_007_Blood_CBC_Report	07:19
R4	allow	read, write, share	Hematology	Clinical-Analyst.	PatID_007_Blood_CBC_Report	07:19
Resolution : • No rules specification for a request with the values of the attributes {Hematology, Surgeon, 08:18} • Rules_Inadequacy • A new rule is constructed to make a decision for this request						

Example 3: Consider Table V, the subject S with attributes {Department="Hematology", Designation ="Surgeon", Time="08:18"} requests to read the file 'PatID_007_Blood_CBC_Report'. In this case, there is no rule specification to make a decision for this request. This rule inadequacy causes rules-insufficient problems, hence the user cannot get his rights despite he is an authorized user.

D. *Conflict-Decision-Positive-Negative*

Definition 4: The conflict-decision-positive-negative will occur if and if only there are three categories(Subject, Object, and Operation) of overlapping. Thus this conflict occurs if the subject is both allowed and denied for the same request of operation on the same object. In this case, both decisions are required or not required sometimes. This is defined in equation- 4.

$$\text{Conflict-Decision-Positive-Negative}(R1,R2) = \begin{cases} \text{True} & T(R1)=T(R2) \text{ and } (R1 = R2) \text{ and } ((X(R1)=\text{allow and } X(R2)=\text{deny}) \text{ Or } (X(R1)=\text{deny and } X(R2)=\text{allow})) \\ \text{False} & \text{Otherwise} \end{cases} \quad \text{---- (4)}$$

TABLE VI. SAMPLE RULES- EXAMPLE OF CONFLICT- DECISION-POSITIVE-NEGATIVE

Rules	Decision (X)	Operations to be performed	Attributes			
			Department	Designation	FileName	Time
R1	deny	share	Hematology	Nurse	PatID_007_Blood_CBC_Report	07:19
R2	allow	share	Hematology	Nurse	PatID_007_Blood_CBC_Report	07:19
Resolution • The subject Nurse is allowed and denied the same request for sharing the same file. • Conflict-Decision- Positive-Negative. • The rule should be reconstructed with more attributes.						

Example 4: In TABLE VI, the subject 'Nurse' is allowed and denied for the same request to share the same file 'PatID_007_Blood_CBC_Report'. In this case, the subject Nurse is not allowed all the time and also not denied all the times to share that file. Thus the Nurse is allowed to only read the file, but he/she can share if permitted by the admin or proper authority. Hence this is a serious security issue, the rules should be reconstructed. For the above example, the rules are modified by adding an additional attribute shown in TABLE VII.

TABLE VII. SAMPLE RULES - RESOLVED THE CONFLICT-DECISION-POSITIVE-NEGATIVE

Rules	Decision (X)	Operations to be performed	Attributes				
			Department	Designation	Permitted by the authority	FileName	Time
R1	deny	share	Hematology	Nurse	No	PatID_007_Blood_CBC_Report	07:19
R2	allow	share	Hematology	Nurse	Yes	PatID_007_Blood_CBC_Report	07:19

E. *Conflict-Demand*

Definition 5: The conflict-demand will occur when multiple rules or multiple requests arise for access to the same shared resource. This can be solved by applying priority scheduling. Let R1, R2 are not redundant rules and S(R1) and S(R2) are the objects of R1 and R2 respectively. This conflict can be expressed in the equation.5

$$\text{Conflict-Demand }(R1,R2) = \begin{cases} \text{True} & S(R1) = S(R2) \text{ and } R1 \neq R2 \\ \text{False} & S(R1) \neq S(R2) \text{ and } R1 \neq R2 \end{cases} \quad \text{---- (5)}$$

Thus there are two requests from the different subjects for access to the same single shared resource. Table VIII illustrates the example of conflict-demand anomaly.

TABLE VIII. SAMPLE RULES – EXAMPLE OF CONFLICT-DEMAND

Rules	Decision (X)	Operations to be performed	Attributes			
			Department	Designation	FileName	Time
R1	allow	read	Hematology	Nurse	PatID_007_Blood_CBC_Report	08:13
R2	allow	read	Hematology	Dietitian	PatID_007_Blood_CBC_Report	08:13
Resolution : • Two subjects request access to the same object at the same time. • Conflict-Demand • The rules should be reconstructed or resolved.						

TABLE IX. SAMPLE RULES – RESOLVED CONFLICT-DEMAND

Rules	Decision (X)	Operations to be performed	Attributes				Priority-level
			Department	Designation	FileName	Time	
R1	allow	read	Hematology	Nurse	PatID_007_Blood_CBC_Report	08:13	1
R2	allow	read	Hematology	Dietitian	PatID_007_Blood_CBC_Report	08:13	2
Resolution : • The subject Dietitian get access to read first • Conflict-Demand resolved							

Example 5: In table VIII, the two subjects 'Nurse' and 'Dietitian' request to read the same file 'PatID_007_Blood_CBC_Report'. This can be resolved by adding priority to the rules. For the above example, the rules are modified by adding the additional parameter Priority-level shown in table IX. The rule or subject which has the highest priority value gets access first. Hence the subject 'Dietitian' gets access to read the file first.

V. CONCLUSION

Cloud service providers use various access control models to protect the shared big data and other resources in the well-communicated and distributed computing technologies. Cloud service providers should satisfy the security requirements of the owners of the shared resources. Most intrusion detection systems use the ABAC model to address the complex security needs of today's computing technologies. Nowadays ABAC is a widely used protection mechanism due to the flexibility, efficiency, and reliability of the mechanism. Despite the ABAC model is efficient in protecting the resources, policy errors or anomalies are the biggest issues that degrade the performance and efficiency of the model. The previous research work fails to identify, express, and resolve considerable policy anomalies and conflicts. We identified, defined, expressed, and described five policy errors:1)Rules-redundancy 2) Rules-Discrepancy 3) Rules-Inadequacy 4) Conflict-Decision-Positive-Negative and 5) Conflict-Demand. In this paper, we explained each anomaly with the proper definition and examples. In this work, we have identified all serious and considerable policy errors which degrade the protection level of the ABAC model.

Our future work is to propose an approach to resolve rules-redundancy anomaly at the time of clustering the rules based on the similarity values to decrease the complexity of cluster generation and time complexity. And also we aim to extend our research to propose an approach to detect and resolve all other anomalies to improve the performance and efficiency of the ABAC model. In the future, a framework may be developed to evaluate and validate the security policies.

REFERENCES

[1] K. Vijayalakshmi and V. Jayalakshmi, "Big Data Security Challenges and Strategies in Cloud Computing : A Survey, International Conference on Soft Computing and Optimising Techniques,August 2019," no. 11824, pp. 11824–11833, 2020.

[2] R. Aljably, Y. Tian, and M. Al-Rodhaan, "Preserving Privacy in Multimedia Social Networks Using Machine Learning Anomaly Detection," *Secur. Commun. Networks*, vol. 2020, 2020, doi: 10.1155/2020/5874935.

[3] E. Conrad, S. Misenar, and J. Feldman, "Domain 5: Identity and Access Management (Controlling Access and Managing Identity)," *CISSP Study Guid.*, pp. 293–327, 2016, doi: 10.1016/b978-0-12-802437-9.00006-0.

[4] P. B. Tarigan, "Encyclopedia of Cryptograpghy and Security, Second Edition, Springer," *Journal of Chemical Information and Modeling*, vol. 53, no. 9. pp. 1689–1699, 2013, doi: 10.1017/CBO9781107415324.004.

[5] M. Markandeyulu, B. V. R. R. Nagarjuna, A. R. Babu, and A. S. K. Ratnam, "A Study of Role Based Access Control policies and Constraints," *Soft Comput.*, no. 1, pp. 279–282, 2012.

[6] V. C. Hu *et al.*, "Guide to attribute based access control (abac) definition and considerations," *NIST Spec. Publ.*, vol. 800, p. 162, 2014, doi: 10.6028/NIST.SP.800-162.

[7] K. Vijayalakshmi and V. Jayalakshmi, "Shared Access Control Models for Big data : A Perspective Study and Analysis," *I Pandian A.P., Palanisamy R., Ntalianis K. Proc. Int. Conf. Intell. Comput. Inf. Control Syst. Adv. Intell. Syst. Comput. vol 1272. Springer, Singapore. https//doi.org/10.1007/*, 2021.

[8] M. Ait El Hadj, A. Khoumsi, Y. Benkaouz, and M. Erradi, "Formal Approach to Detect and Resolve Anomalies while Clustering ABAC Policies," *ICST Trans. Secur. Saf.*, vol. 5, no. 16, p. 156003, 2018, doi: 10.4108/eai.13-7-2018.156003.

[9] E. C. Lupu and M. Sloman, "Conflicts in policy-based distributed systems management," *IEEE Trans. Softw. Eng.*, vol. 25, no. 6, pp. 852–869, 1999, doi: 10.1109/32.824414.

[10] J. D. Moffett and M. S. Sloman, "Policy conflict analysis in distributed system management," *J. Organ. Comput.*, vol. 4, no. 1, pp. 1–22, 1994, doi: 10.1080/10919399409540214.

[11] M. Ait, E. Hadj, M. Erradi, and A. Khoumsi, "Validation and Correction of Large Security Policies : A Clustering and Access Log Based Approach," *2018 IEEE Int. Conf. Big Data (Big Data)*, no. 1, pp. 5330–5332, 2018, doi: 10.1109/BigData.2018.8622610.

[12] R. A. Shaikh, K. Adi, and L. Logrippo, "A Data Classification Method for Inconsistency and Incompleteness Detection in Access Control Policy Sets," *Int. J. Inf. Secur.*, vol. 16, no. 1, pp. 91–113, 2017, doi: 10.1007/s10207-016-0317-1.

[13] V. Vijayalakshmi, K. and Jayalakshmi, "A Priority-based Approach or Detection of Anomalies in ABAC Policies using Clustering Technique," no. Iccmc, pp. 897–903, 2020, doi: 10.1109/iccmc48092.2020.iccmc-000166.

[14] M. Yahiaoui, A. Zinedine, and M. Harti, "Deconflicting Policies in Attribute-Based Access Control Systems," *Colloq. Inf. Sci. Technol. Cist*, vol. 2018-Octob, pp. 130–136, 2018, doi: 10.1109/CIST.2018.8596576.

[15] C. Morisset, T. A. C. Willemse, and N. Zannone, "A framework for the extended evaluation of ABAC policies," *Cybersecurity*, vol. 2, no. 1, 2019, doi: 10.1186/s42400-019-0024-0.

[16] H. Alazzam, A. Sharieh, and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer," *Expert Syst. Appl.*, vol. 148, 2020, doi: 10.1016/j.eswa.2020.113249.

[17] R. Guan, S. Wang, S. Xiong, and X. Nie, "Preliminary Study on Intrusion Prevention System of Small Aircraft Applied to Large Hydropower Station," *J. Phys. Conf. Ser.*, vol. 1549, no. 5, 2020, doi: 10.1088/1742-6596/1549/5/052107.

[18] V. K.Vijayalakshmi, V.Jayalakshmi, "A priority-based approach for detection of anomalies in ABAC policies using clustering technique," *2020 Fourth Int. Conf. Comput. Methodol. Commun. (ICCMC), IEEE,* pp. 897–903, 2020.