Validation Policy Statement on the Digital Evidence Storage using First Applicable Algorithm

Achmad Syauqi[1]
Department of Information System
Universitas Peradaban
Brebes, Indonesia

Imam Riadi[2]
Department of Information System
Universitas Ahmad Dahlan
Yogyakarta, Indonesia

Yudi Prayudi[3]
Department of Informatics
Universitas Islam Indonesia
Yogyakarta, Indonesia

***Abstract*—Digital Evidence Storage is placed to store digital evidence files. Digital evidence is very vulnerable to damage. Therefore, making digital evidence storage need access control. Access control has several models, one of them is ABAC (Attribute-Based Access Control). ABAC is a new access control model. ABAC model has a flexible function, allows intersect with many attributes. This will be very complex and causing inconsistency and incompleteness. Access control testing is a must before access control is implemented because it is the main key in the security of a system. Especially in digital evidence storage because the data in it is very vulnerable to damage either intentionally or not. The type of access control that is widely used is ABAC because this ABAC model has a flexible function. This ABAC model intersects with many attributes, it is necessary to test the policy statement. This test is carried out to avoid inconsistencies and incompleteness in the policy statement. An example tool for testing policy statements is ACPT (Access Control Policy Testing). At ACPT there are various algorithms for creating and testing policy statements. This study uses the first applicable algorithm to test policy statements in digital evidence storage. This research has successfully tested the policy statement properly and found no inconsistencies and incompleteness.**

Keywords—Testing; policy statement; rule; ABAC; digital evidence

I. Introduction

Storing a file is important and has rules. [1] ne example is digital evidence. Digital evidence is very vulnerable to damage and very possible data changes [2]. A system must be safe, especially from attacks [3] and avoid cybercrime [4]. Therefore, to making digital evidence storage (DES), special security is needed. The digital evidence contained within them will be guaranteed safety with security. [5] Redfield has researched the process of recording and storing digital evidence. They introduced the Gringgotts scheme to a system to maintaining integrity in the recording process, transfer and data storage with the digital signatures as data security from the digital evidence file. The digital evidence file is guaranteed authenticity and can be accounted for before the court.

DES must be made to make it easier for users. data storage, a web-based graphical user interface, and an API endpoint so practitioners can analysis the schedule actions on clients and view and process data. [6].

Digital evidence cannot be accessed by any user, so the DES needs to be added with access control. Access control has several models, one of them is ABAC (Attribute-Based Access Control). ABAC is a new access control model. ABAC model has a flexible function. So ABAC is used as an access control model that is widely used in security systems now and for years to come.

ABAC model has a flexible function, allows intersect with many attributes. This will be very complex and causing inconsistency and incompleteness. ABAC model is possible for a user who should have permit access rights to change to deny, and vice versa. Therefore the implementation of ABAC must be supported by appropriate policies and validated properly in order to the security in DES runs well.

There are many models for validating ABAC, one of them is model checking. This model while discussing the elements in the system to the errors can be identified easily. There are many tools for validating policy statements, one of them is Access Control Policy Testing (ACPT). ACPT has various methods for creating and testing policy statements [7]. As well as being complete ACPT is widely used for ABAC validation research that has been done before. ACPT has several algorithms that used to test policy statements.

II. Literature Review

Literature review takes reference based on studies that have been done first. The literature review can be seen in Table I.

TABLE. I. LITERATURE REVIEW

Main paper	**application**	**Validation model used**	**Data used**	**Tools**
Catherine MS Redfield, Hiroyuki Date (2014) [5]	Digital evidence storage	Skema Gringgot	Digital evidence	-
Dianxiang Xu, Yunpeng Zhang (2014) [8]	Application content	Model checking	Policy	ACPT
Ang Li, Qinghua Li, Vincent C Hu, Jia Di (2015) [9]	Database system	Model Checking	Policy	ACPT
Nariman Ammar, Zaki Malik, Abdelmounaam Rezgui, Elisa Bertino (2016) [10]	Data Repository	Dinamic privacy management	Policy	SunXACML
Nuo Li, JeeHyun Hwang, Tao Xie (2008) [11]	Web application	-	Policy	SunXACML
Muhammad Aqib, Riaz Ahmed Shaikh (2015) [12]	-	Formal methods, Model checking methods, matrix based approaches, mining technique, mutation testing technique, others	Policy	-
M Fadly Panende, Imam Riadi, Yudi Prayudi (2017) [13]	Digital evidence storage	-	Policy	UMU
Research				
Solution	Digital evidence storage	Model Checking	Policy	ACPT

III. THEORY

Access control has several types of models that used from the first to the lastest. Access control models are MAC, DAC, RBAC, and ABAC:

1) MAC (Mandatory Access Control): MAC giving access depends on the document owner.

2) DAC (Discretionary Access Control): The DAC will restrict access to objects based on the identity of the subject.

3) RBAC (Role-Based Access Control): RBAC is an approach that limits access to a system for users who have authority in the system.

4) ABAC (Attribute-Based Access Control): ABAC is one of model access control that applies policies.

According to Dianxiang Xu and Yunpeng Zhang [8], the latest generation of access control models is the ABAC model because this model has better features than the previous generation access control model. These features are:

1) ABAC can provide grant access control trought the attributes of authorization elements such as subject, resources, actions, and environment into an access control decision. This also allows the subject to access the widest possible resources without the existence of individual relationships between each subject.

2) ABAC can facilitate the administration of collaborative policies in large organizations. This policy can be prepared by policymakers from various departments.

3) ABAC can also facilitate the decoupling of access control from certain application business logic.

How to work from ABAC according to Hu et al. [3] can be illustrated in Fig. 1:

From the picture it can be explained that there are three main steps in implementing ABAC, namely:

1) Subject accept request from object

2) Give a decision given through an evaluation mechanism for (a) Rules, (b) Subject Attributes, (c) Object Attributes and (d) environmental conditions.

3) The subject is given a decision: reject or allow access to the object.

ABAC is an access control method in which subjects will only make requests to perform operations on objects based on the attributes that have been pinned on the subject, object, environmental conditions, and some policies, included in the attributes and conditions. The authorization element In the ABAC model is defined in the attribute. According to Shandu [14] there are 4 aspects of attributes in ABAC, namely:

1) Subject: The subject is a user, whether human or not (eg device or software) requesting an access request. Examples of this subject include name, address, position, etc. While requests can use subject attributes with unique properties.

2) Resources: Resources are protected targets such as devices, networks, files, applications, etc.

3) Operation: An operation is the implementation of a function that requests a subject for resources.

4) Environment attribute: Is an operational and situational characteristic, such as current time, ip address, etc

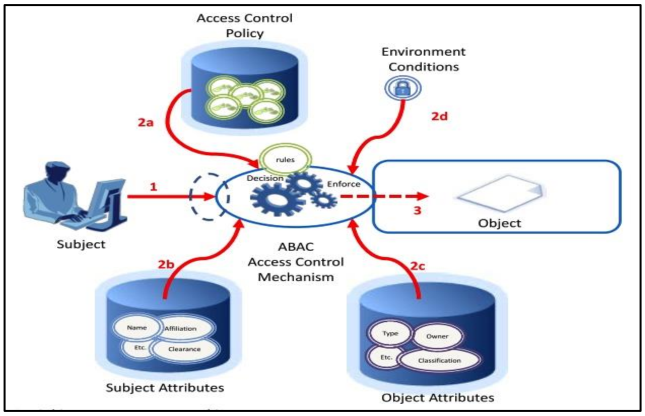

Fig. 1. How to Work from ABAC.

Attributes can be determined through identifiers (variables), data types, and domains where finite sets containing the value of the data type are given. Data types of attributes can be used in computer systems such as integers, strings, and booleans. Data types or domain attributes in ABAC can be specified explicitly or implicitly.

ABAC policy is a function representation that determines whether access requests are permitted based on the given attribute value. Formally the ABAC policy will contain three (X, Y, f). where:

- X is the finite set of attributes with domain D1 ... Dn
- Y is the finite set of access control decisions (for example: permit, deny, undefined)

F:= D1 x D2 x … Dn → Y; this is an access control function

The ABAC policy is said to be complete if and only if f is a total function, where for the value given by each attribute then f always produces deterministic decisions. In this case, a different ABAC system will use a different set of access control decisions, for example {permit, deny, undefined} *or* {permit, deny, NotApplicable, Intermediate}. According to Aqib [12], there are two problems faced in implementing access control solution, namely.

1) Inconsistency: Inconsistency it is a condition where there are 2 rules that give the result of contradiction. If S, O and A is Subject, Object and Actions. If given $a \in$ A, $s \in S$, $o \in O$, then given $d \in D$ namely set Decision D = { permitted, denied, undefined} and $r \in R$ in the form of a three tuple rule (s,o,a) → d. A policy is said to inconsistency if for every two rules r_i and $r_j \in R$, where $i \neq j$ then $r_i \rightarrow d_i$ and $r_j \rightarrow d_j$ where $i \neq j$ then r_i and r_j will give the results of contradictory decisions.

2) Incompleteness: It is a condition where there are rules that have not been accommodated in a set of rules that have been previously set. That is, there is r for a condition where $r \notin R$.

The purpose of validating access control policies is to make sure that the inconsistency and incompleteness system was not happening. If the problem still exists, the security system becomes invalid or not safe. Accordance with Aqib [12] to validate the access control policy with the ensure that there are no incidents and incompleteness. There are 5 methods to do validation namely: Mining technique, model checking technique, formal methods, matrix-based approaches, mutation testing, and other technique.The method in this study can be summarized as in Fig. 2:

Based on the summary of the access control validation method from Aqib [12] the author tries to apply an examination model for validating digital evidence access. Model-checking method is a method that uses Linear Temporal Logic (LTL) to describe the nature and model of SPIN examiners to be used in the verification and validation of existing policies. The checking model checks the components in a system, in this case, a policy statement. The policy statement in the DES will be checked one by one so that if there are differences in policy, the errors will be known and can be corrected again.

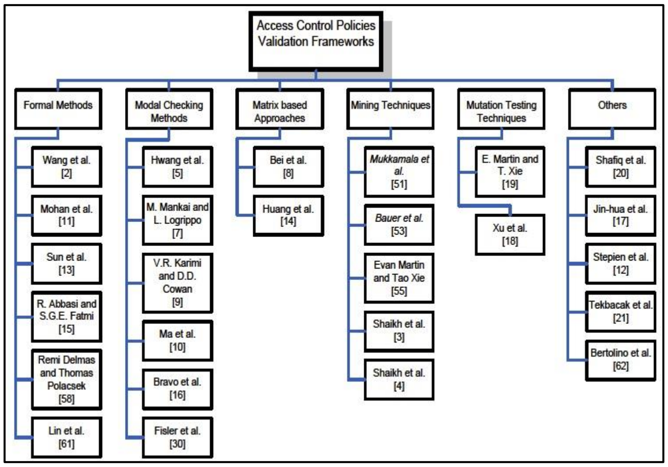

Fig. 2. Summary of Validation Access Control Methods.

There are many tools for making ABAC policies, one good tool is ACPT [7]. Some previous studies used ACPT to create and test ABAC access control. ACPT provides 3 main functions, namely:

1) Help find and combine policies based on what is already known by the existing policy model.

2) ACPT analyzes and converts policies (based on policy models) into formats that can be run like XACML.

3) To ensure that the policy is correct, ACPT conducts static and dynamic verification of a policy.

ACPT has several algorithms for creating and testing ABAC access control, namely:

1) First applicable: First applicable is the condition when the policy statement that has been compiled and given access rights becomes the first effect. While policy statements that have not been granted access rights will be given other access rights.

2) Deny overrides: Deny overrides is an algorithm that combines decisions in such a way that if there is a missing decision. Then the decision will win.

3) Permit Overrides: Permit Overrides are the opposite of deny overrides, an algorithm that combines decisions in such a way that if there is a permit decision. Then the decision will win.

This study uses the first applicable algorithm because it is more suitable for making and testing ABAC access control for DES.

IV. Methodology

1) Policy Statement: The policy statement in this study takes the example of a policy statement in the DES [13].

2) Software: The software used to test the policy statement is ACPT (Access Control Policy Testing).

3) Model: The model used used for this study is a checking model with methods provided by ACPT, namely, first applicable.

V. Result and Analisys

A. DES Policy Statement

The policy statement in this research takes the example of a policy statement in the DES [13], the policy statements namely:

1) First responder: The First Responder has the task of processing the scene to find evidence and obtain electronic evidence and upload digital evidence to DES. So that the DES has permission to access rights, namely: upload digital evidence, create a cabinet, create a rack, create a bag, input data evidence. In addition to the access rights granted, First Responder is not permitted to access it.

2) Examiner: The Examiner has the task of processing digital evidence so that the DES has permission to access rights, namely: download digital evidence, input data case. In addition to the access rights granted, Examiner is not permitted to access it.

3) Officer: The officer has the task of managing everything in the DES so that he has permission, namely: create a username, create a password, create a signature, delete digital evidence, validate digital evidence, validate data evidence, validate data case, create form COC, download form COC. In addition to the access rights granted, Officer is not permitted to access it.

4) External: Externals are parties that go beyond official participation. Which includes external in the DES, namely lawyers and external examiners. Externals who are given access rights namely: download digital evidence and download form COC. In addition to the access rights granted, External is not permitted to access it.

Rules for DES-based on the policy statement above are divided into four, namely, rules for first responders, rules for testers, rules for officers and final rules for external. The rule for the first responder on DES can be seen in Table II.

The rule for examiner on DES can be seen in Table III.

The rule for an officer on DES can be seen in Table IV.

The rule for external on DES can be seen in Table V:

TABLE. II. Rule for First Responder on DES

Subject	Resource	Action	Environment	Decision
First Responder	Digital Evidence	Upload	Fulfilled	Permit
	Cabinet	Create	Fulfilled	Permit
	Rack	Create	Fulfilled	Permit
	Bag	Create	Fulfilled	Permit
	Data Evidence	Input	Fulfilled	Permit

TABLE. III. The Rule for Examiner on DES

Subject	Resource	Action	Environment	Decision
Examiner	Digital Evidence	Download	Fulfilled	Permit
	Data Case	Input	Fulfilled	Permit

TABLE. IV. Rule for Officer on DES

Subject	Resource	Action	Environment	Decision
Officer	Username	Create	Fulfilled	Permit
	Password	Create	Fulfilled	Permit
	Signature	Create	Fulfilled	Permit
	Digital Evidence	Delete	Fulfilled	Permit
	Digital Evidence	Validate	Fulfilled	Permit
	Data Case	Validate	Fulfilled	Permit
	Form COC	Create	Fulfilled	Permit
	Form COC	Download	Fulfilled	Permit

TABLE. V. Rule for External on DES

Subject	Resource	Action	Environment	Decision
External	Form COC	Download	Fulfilled	Permit
	Digital Evidence	Download	Fulfilled	Permit

B. Testing Policy Statement

Test the policy statement using the ACPT tool and use a combination of the first applicable algorithm. This test is done 30 times because in statistics 30 is the minimum sample for a large population. In one test, there were 60 different policy statement combinations. This combination resulted in two decisions, namely, permission and rejection. The complete results of this test can be seen in Table VI.

This study has the main objective to examine the policy statement in DES to avoid inconsistencies and incompleteness and by the rules that have been prepared previously with the checking model. This inspection model checks the elements in the policy statement in DES whether the policy statement complies with the rules or not and there are no inconsistencies and incompleteness.

TABLE. VI. Test Result

Testing to	Test Result		Testing to	Test Result	
	Decision Permit	**Decision Deny**		**Decision Permit**	**Decision Deny**
1	3	57	**16**	1	59
2	2	58	**17**	2	58
3	1	59	**18**	1	59
4	2	58	**19**	2	58
5	2	58	**20**	2	58
6	3	57	**21**	4	56
7	3	57	**22**	1	59
8	3	57	**23**	2	58
9	3	57	**24**	1	59
10	2	58	**25**	3	57
11	1	59	**26**	1	59
12	1	59	**27**	1	59
13	4	56	**28**	5	55
14	3	57	**29**	1	59
15	2	58	**30**	2	58

1) Inconsistency: Inconsistency it is a condition where there are 2 rules that give the result of contradiction. If S, O and A is Subject, Object and Actions. If given $a \in$ A, $s \in S$, $o \in O$, then given $d \in D$ namely set Decision D = { permitted, denied, undefined} and $r \in R$ in the form of a three tuple rule (s,o,a) → d. A policy is said to inconsistency if for every two rules r_i and $r_j \in R$, where $i \neq j$ then $r_i \rightarrow d_i$ and $r_j \rightarrow d_j$ where $i \neq j$ then r_i and r_j will give the results of contradictory decisions. Examples of inconsistencies from this study are if the first responder is the subject, digital evidence as an object, upload as action and allow as a decision. However, in rule 1 and other rules, the decision of the subject of the first responder must be permitted but has a decision deny.

This study found no inconsistencies in the policy statement for the DES after testing using the model checking. Policy statements are prepared by existing rules.

2) Incompleteness: It is a condition where there are rules that have not been accommodated in a set of rules that have been previously set. That is, there is r for a condition where r $\notin$ R. An example of the incompleteness of this study is that the subject of the first responder must have 5 rules, but in preparation, there are only 4 rules so that there is still 1 rule that has not been accommodated in the rules set for the first responder subject.

This policy statement in DES after testing with the checking model did not find any rules that had not been accommodated from the set of rules that had been made. The rules in the DES policy statement have been accommodated properly without being incomplete.

VI. Conclusion

Security in a system is the main thing in making the system itself. Especially systems that have very important and easily damaged data such as DES. These problems form the basis of the DES system which must be equipped with access control. Access control is what restricts users from accessing the entire system. The access control model that is widely used today is the ABAC model. Developing an ABAC model of access control for DESs must also consider policy statements and rules to be made and tested as a final step before access control is applied to the DES system. Testing models for testing access control now vary. This study uses the ACPT tool and a combination of the first applicable algorithm in compiling and testing DES policy statements. The tests conducted in this study did not find inconsistency and incompleteness problems. The combination of policy statements from the test results runs according to the rules that have been prepared previously.

References

[1] F. Albanna dan I. Riadi, “Forensic Analysis of Frozen Hard Drive Using Static Forensics Method,” International Journal of Computer Science and Information Security (IJCSIS), Vol. %1 dari %2Vol. 15, No. 1, 2017.

[2] R. Umar, I. Riadi dan G. M. Zamroni, “A Comparative Study of Forensic Tools for WhatsApp Analysis using NIST Measurements,” (IJACSA) International Journal of Advanced Computer Science and Applications , Vol. %1 dari %2Vol. 8, No. 12, 2017.

[3] A. Yudhana, I. Riadi dan F. Ridho, “DDoS Classification Using Neural Network and Naïve Bayes Methods for Network Forensics,” (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. %1 dari %2Vol. 9, No. 11, 2018.

[4] I. Riadi, A. Fadlil dan A. Fauzan, “A Study of Mobile Forensic Tools Evaluation on Android-Based LINE Messenger,” (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. %1 dari %2Vol. 9, No. 10, 2018.

[5] C. M. Redfield dan H. Date, “Gringotts: Securing Data for Digital Evidence,” IEEE Security and Privacy Workshops, 2014.

[6] S. Sunardi, I. Riadi dan A. Sugandi, “Forensic Analysis of Docker Swarm Cluster using Grr Rapid Response Framework,” (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. %1 dari %2Vol. 10, No. 2, 2019.

[7] J. Hwang, T. Xie, V. Hu dan M. Altunay, “ACPT: A Tool for Modeling and Verifying Access Control Policies,” IEEE International Symposium on Policies for Distributed Systems and Networks, 2010.

[8] D. Xu dan Y. Zhang, “Specification and Analysis of Attribute-Based Access Control Policies: An Overview,” Eighth International Conference on Software Security and Reliability - Companion, 2014.

[9] A. Li, Q. Li, V. C. Hu dan J. Di, “Evaluating the Capability and Performance of Access Control Policy Verification Tools,” IEEE Military Communications Conference, 2015.

[10] N. Ammar, Z. Malik, A. Rezgui dan E. Bertino, “XACML Policy Evaluation with Dynamic Context Handling,” IEEE Transactions on Knowledge and Data Engineering, 2016.

[11] N. Li, J. Hwang dan T. Xie, “Multiple-Implementation Testing for XACML Implementations,” TAV-WEB '08 Proceedings of the 2008 workshop on Testing, analysis, and verification of web services and applications , 2008.

[12] M. Aqib dan R. A. Shaikh, “Analysis and Comparison of Access Control Policies Validation Mechanisms,” I.J. Computer Network and Information Security, 2015.

[13] M. F. Panende, I. Riadi dan Y. Prayudi, Model ABAC pada Lemari Penyimpanan Bukti Digital, Yogyakarta: Universitas Islam Indonesia, 2018.

[14] R. Sandhu, Security Models: Past, Present and Future, San Antonio: Institute for Cyber SecurityUniversity of Texas at San Antonio, 2009.