International Journal of Electrical and Computer Engineering (IJECE)
Vol. 8, No. 5, October 2018, pp. 3187~3193
ISSN: 2088-8708, DOI: 10.11591/ijece.v8i5.pp3187-3193

An Enhancement Role and Attribute Based Access Control Mechanism in Big Data

M. Menaka, K. Meenakshisundaram
Department of Computer Science, Erode Arts and Science College, India

Article Info

Article history:

Received Dec 10, 2017
Revised Jan 10, 2018
Accepted Aug 23, 2018

Keyword:

Attribute based access control
Environment state
Event-role-attribute based fine
Grained access control
Role based access control
Temporal state

ABSTRACT

To be able to leverage big data to achieve enhanced strategic insight and make informed decision, an efficient access control mechanism is needed for ensuring end to end security of such information asset. Attribute Based Access Control (ABAC), Role Based Access Control (RBAC) and Event Based Access Control (EBAC) are widely used access control mechanisms. The ABAC system is much more complex in terms of policy reviews, hence analyzing the policy and reviewing or changing user permission are quite complex task. RBAC system is labor intensive and time consuming to build a model instance and it lacks flexibility to efficiently adapt to changing user's, objects and security policies. EBAC model considered only the events to allocate access controls. Yet these mechanisms have limitations and offer feature complimentary to each other. So in this paper, Event-Role-Attribute based fine grained Access Control mechanism is proposed, it provide a flexible boundary which effectively adapt to changing user's, objects and security policies based on the event. The flexible boundary is achieved by using temporal and environment state of an event. It improves the big data security and overcomes the disadvantages of the ABAC and RBAC mechanisms. The experiments are conducted to prove the effectiveness of the proposed Event-Role-Attribute based Access Control mechanism over ABAC and RBAC in terms of computational overhead.

Corresponding Author:

M. Menaka,
Department of Computer Science,
Erode Arts and Science College, Erode.
Email: menarameac@gmail.com

1. INTRODUCTION

The large amount of data can be generated rapidly from different sources like smart phones, sensors, social networks, etc., due to the development of big data [1]. The conventional computer systems are not useful for processing such large amount of data. In addition, rapid development of data and data objects include several issues such as capabilities for data storage and security. Therefore, it is required to design access control mechanisms based on the end-users requirements for only sharing the end-user's data to authorized users. The most challenging of network security components is Access Control (AC) mechanisms [2].

Access control is one of the most powerful and basic ways of risk mitigation in any application. The access control mechanisms are used to enable the end-users for controlling the access of their own data. Among different access control mechanisms, Attribute-based Access Control (ABAC) [3] and Role Based Access Control [4] are emerging as a promising paradigm. Generally, ABAC is referred as an access control protocol wherein the access rights are allowed to users through the utilization of policies which combine attributes together. The policies may be utilize any kind of attributes such as user attributes, resource attributes, object, environment attributes, etc. RBAC is a policy neutral control mechanism defined around privileges and roles. In Big data, these mechanisms are mostly used for providing privacy and security.

The RBAC model is not sufficient for situations where the contextual attributes are required parameters in granting access to a user. Furthermore, RBAC is that the permissions are described in terms of object identifiers referring to individual objects. This mechanism is not adequate in situations where a huge volume of objects in hundreds of thousands exist and it leads to role permission explosion problem. The ABAC model is more flexible than RBAC but it more complex than RBAC. EBAC [5] is a new paradigm that adds reactive functionality to access control. The EBAC system changes the access privileges of users dynamically based on different database and temporal events. In an EBAC system, the access control policies are represented in terms of Event-Condition- Action rules, where the Action part changes the access privileges of users.

These models have feature complimentary to each other problem. These models has static boundary to provide access control. So in this paper, an Event-Role-Attribute Based fine grained Access Control mechanism is proposed which provide a flexible boundary for security policies. The security policies are developed based on the event, roles and attributes. Initially a fine grained description is defined and developed an event-Role-Attribute based access control mechanism on top it. It provides high security level to big data by changing user's, objects and security policies based on the event.

2. LITERATURE SURVEY

Amir, M., et al [6] proposed a novel token based access control based on cascading permission policy to control interactivity with big data. Token based access control is one of the access control mechanism that can comfortably handle and easily handle interactions with big data. It also enables provisioning of access to fine levels of granularity but this approach hard to manage during the rapid growth of repository. This problem is handled by proposed novel token based access control based on cascading permission policy where the access policies are extended to allow finer control granularity of visible resources along with the context-specific parameters that refine access based on the origin context. But this cascading permission policy model cannot be steered towards the security realm which provide better authentication of tokens.

Hu, V. C., et al [7] proposed a general access control scheme for distributed Big data processing clusters. In this paper introduced the definition of Big data and illustrated a big data process model, abstracted based on general distributed processing environment and presented Big data access control scheme and considerations based on trust between Big data Master System and Big data source providers and simultaneously between cooperating system and Master system. The proposed access control scheme focused on authorization that protects Big data processing and data from the insider attacks. Moreover it used attributes of actions, subjects, actions and sometimes environment conditions which determine the access permission. But the security of Big data is still challenging one.

Huang, J., et al [8] proposed a framework that integrates Attribute Based Access Control (ABAC) and Role Based Access Control (RBAC). The proposed framework used attributed based policies to create a RBAC model. This can be achieved by modeling RBAC into two levels. The above ground level is a standard RBAC model expanded with environment and it is used to support RBAC model verification. Another level is under ground level that is utilized to represent security knowledge in terms of attribute based policies that automatically generates the simple RBAC model in the above ground level. This framework tackles the problem of permission assignment for large scale applications such as big data. But the proposed framework used less number of features that reduce the performance of framework.

Khan, M. F. F., & Sakamura, K. [9] proposed a context sensitive approach to access control that developed on conventional Role Based Access Control (RBAC) and Discretionary Access Control (DAC). In addition to that, a fine grained access control mechanism is proposed to personal health information using eTRON security. The eTRON security architecture advocate use of tamper resistant chips equipped with functions for mutual authentication and encrypted communication and implemented DAC based delegation of access control rights. In order to realize access decision and authorization the RBAC model is used and implemented context verification on top of it. But this approach does not meet all healthcare domains requirements.

Kuhn, D. R., [10] applied methods to analysis the relationship among vulnerabilities in misconfigured access control rule structures. In this paper, introduced a taxonomy structure defines the rule structures and computed the detection conditions for each class of vulnerabilities in different structures. The conditions that allow the exploitation of one are sufficient for triggering other vulnerabilities downstream in hierarchies. While the number of potential flaws that result in vulnerabilities is vast, the structure of access control rules results in a hierarchy for certain classes of vulnerabilities. Thus the existence of hierarchies utilized to reduce the number of tests needed to detect the presence of faults in access control rules.

Gupta, A., et al [11] proposed conceptual storage model that solve the security problems such as data verification, transmission and data storage ion Big data. The proposed model segregates big data based on the roles in the hierarchy of organization having access to the data. The access control is provided by considering the valuable data and the number of times the data has been accessed within a certain period of time. This model is a combination of Hadoop Distributed File System and Role based Access Control. Furthermore, a normalization technique is employed to used the resources effectively through distribute the data evenly over different servers. The periodic normalization leads to overhead which is major drawback of this system.

Zeng, W., et al [12] proposed an innovative access control model named as Content Based Access Control (CBAC) for content centric information sharing. CBAC model allows approximation it does not provide a static boundary for the accessible set of records. The main contributions of this model are three fold, a data driven access control model is proposed which exploits the data content to achieve powerful and flexible access control semantics. Then developed a CBAC utilized native functions from the of-the-shelf database systems. Finally the efficiency of CBAC is improved by developing a labeling and blocking mechanism. This model has overhead problem.

Rajpoot, Q. M., et al [13] proposed a model that combines traditional two models are attribute based access control and Role based access control which unifies the benefits of two models. The proposed model provided a fine grained access control that makes the access control decisions by considering contextual information. Different request evaluation mechanism was presented which can be utilized by different applications depending on their requirements. The proposed access control model was extended by using cache cache mechanisms to further expedite the access control decision process.

3. METHODOLOGY

In this section the proposed an Event-Role-Attribute based fine grained access control mechanism for big data security is described in detail. The proposed Event-Role-Attribute based access control mechanism provides a flexible boundary for the accessible set of records in Big data based on events. It can be achieved by introducing a environmental and temporal state of an event. Initially a fine grained description for the roles and attributes of the data is presented and an Event-Role-Attribute based mechanism is built on top of the description. Then an event based access control model is introduced based on the Attribute Based Access Control (ABAC) and Role Based Access Control (RBAC) which improves the big data security.

3.1. Fine Grained Description

Data could be described as a 11 tuples $(d_0, r_0, d_a, attr, U, R, OP, EP, \text{Temstate}, URA^e, RPA^e, A)$. These 11 tuples are defined as follows:

a. d_0 represents all the objects sets of data. The objects could be anything users want to define, such as size of data, type of data, number of attributes and number of records.
b. r_0 denotes the root of objects where $r_0 \in d_0$ and it is defined to implement traversal and search for other objects.
c. d_a denotes all the object's attributes of data. The attributes of objects are secured could be security level, location and so on.
d. $attr$ denotes a binary relationship where $attr \subseteq d_a \times d_0$
e. U denotes a set of users
f. R denotes the set of role. A role reflects a job function and it is also associated with a set of permissions
g. OP denotes a set of operators
h. EP defined a set of predefined environment state patterns. It is used to model the context of a user's access. Each environment state pattern describes a set of environment states
i. Temstate defined the set of temporal state
j. URA^e denote the extended user role assignment relation which maps users with roles associated with certain environment patterns where $URA^e \subseteq U \times R \times EP$
k. RPA^e denotes the extended role permission assignment relation which used to maps roles with permissions also associated with certain environment patterns, where $RPA^e \subseteq R \times P \times EP$

3.2. Event-Role-Attribute Based Access Control

The Event-Role-Attribute Based fine grained Access Control model is shown in Figure 1. The AH is the action hierarchy which is denoted by $AH \subseteq A \times A$ and AP is the action permission assignment which is denoted by $AP \subseteq A \times P$ is a many-to-many action-to-permission assignment relation. The basic concept of Event-Role-Attribute based Access Control is described as follows:

Object is the finest granularity which a subject could access because of the characteristics like unified and atomic in security requirement. One or more objects could combine a sub data and one or more sub data could combine a big data. The set of objects are denoted as Object and it is equal to d_0. The Event-Role-Attribute based access control model with 5 tuples (Subject, Event, OT, P and Object) is used to solve the fine grained access control problem of big data.

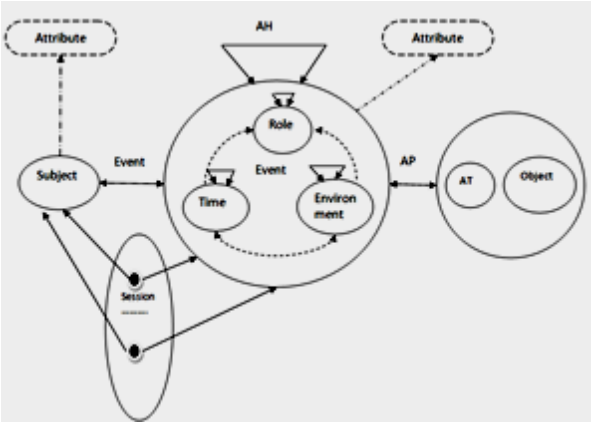

Figure 1. Event-role-attribute based fine grained access control model

3.3. Event-role-attribute based Access Control Protocol

The following access control protocol including data creation, permission specification and authority management to provide access control for big data. The symbols used in the protocol are defined as follows:

U_{id}:User's/ subject's ID
O_{id}: Objects ID
TEM_S: User's temporal state
ENV_S: User's environmental state
$start_T$: starting time of policy
end_T: end time of policy
key_o: the object's key symmetry encryption which is usually encrypted under user's private key
ot: operation type
pd: the policy's description
SK_A: A's private key

The N represents the abstract data types including role, time, event, data and environment, O denotes the set of objects and P denotes the set of policies.

a. GenData: content provider generates the structured data in specified format.
   GenData(Contents:N;out Data:N) ⊲
   Data=format (contents);⊳
b. Define object: user divides the data into sections according to their requirement for access control
   Define Object (Data: N; out O:N) ⊲
   If($o \notin O$ then O'=O∪ $\{o\}$)⊳
c. Assign Policy: user assigns the policy to the divided data and submits the results to the policy server where Policy=$\{P_1, P_2, \ldots P_k\}$
   Assign Policy (O, environment, temporal, ot, key, $start_T, end_T, pd: N; out\ Policy: N$) ⊲
   if policy $\notin Policy$
   then policy.role = role
   policy.temporal =temporal
   policy.environmet =environment
   policy.object = O
   policy.ot =ot
   policy.k =key
   policy. $start_T = start_T$
   policy. $end_T = end_T$

policy.pd = pd
POLICYS'=POLICYS∪ {$policy$}
POLICYS= POLICYS'⊳

d. verifyid: verify the user identity and certificate; if it is true then return true otherwise it returns false
Verifyid(user, certification: N; out result: BOOLEAN)
⊲ result = (userϵU)(isvalid(certification)) ⊳

e. GetReq: obtain the user's requirement R
GetReq (user, object, data, ot: N; out req:N) ⊲
req.subject =user
req.ot=ot
req.Object=object ⊳

f. GetEvent: obtain the information including role R, Time T and environment E for the policy server's further judgement.
Get Event (req, temporal, environment :N; out Event: N) ⊲
Event.role=req.subject
Event.temporal=temporal
Event.environment=environment ⊳

g. VerifyTEM: when the event information Event and user's requirement req are obtained, the policy server will validate the user's temporal information by using the following function. If it is valid return true otherwise it return false
VerifyTem (Event, validTEM : N;out result: BOOLEAN) ⊲
action.temporal∈Temstate
validTEM⊆ Temstate
result= ($\exists t_1, t_2 \in$ validTEM. $t_1 \leq Event.temporal \leq t_2$) ⊳

h. VerifyE: when the user's requirement req and event information event are obtained, the policy server will validate the user's environmental information by this function. I f valid then return true otherwise it returns false
VerifyE (Event, validE: N; out result :BOOLEAN) ⊲
Event.environment ∈EP;
Valide⊆EP
result=($\exists e_1, e_2 \in$ validE. $e_1 \leq Event.environment \leq e_2$) ⊳

i. VerifyR: when the user's requirement req and event information event are obtained, the policy server will validate the user's Role by using this function. if the function is valid it retrns true otherwise it returns false.
VerifyR (Event, validR : N; out result: BOOLEAN)
Event.role ∈ R
ValidR⊆R
result=($\exists r_1, r_2 \in$ validr. $r_1 \leq Event.role \leq r_2$) ⊳

j. JusUsage: the policy server will search the corresponding policy for user according to his event and requirement.
JusUsage (event, req: N; out policy: N) ⊲
policy'=($\exists policy' \in POLICY(policy'.object = req.object \wedge policy'.ot = req.ot)$)
if verifyR (event, policy'.role) then
if verify(event,policy'.temporal) then
if verifyE(event, policy'.environment) then
policy=policy'⊳

The above protocol provides the fine grained Event-Role-Attribute based access control to big data based on environment, temporal, role and attributes in big data.

4. RESULTS AND DISCUSSION

In this section the experiment conducted in healthcare data which contain the health events data of patients. For the experimental purpose, the ABAC, RBAC, EBAC, Event-Role-Attribute based access control are tested in terms of computation overhead to prove the effectiveness of the proposed Event-Role-Attribute based access control model.

4.1. Computation Overhead

Computational overhead is analyzed based on the computation time of the access control models. Table 1, shows the comparison of computation overhead. It shows the computation overhead of Attribute

Based Access Control (ABAC), Role Based Access Control (RBAC), Event Based Access control (EBAC), Event-Role-Attribute Based Access Control (ERABAC). For 5 number of attributes the existing RBAC has computation time of 25 milliseconds, ABAC has computation time of 20 milliseconds, EBAC has 15 milliseconds and the proposed ERABAC has computation time of 10 milliseconds. Similarly the following Table 1, shows the computation overhead of each access control methods for 6 different number of attributes. From the Table 1, it is proved that the proposed ERABAC has better computation time than the other methods.

Table 1. Comparison of Computation Overhead

No. of attributes	Access Control Methods			
	RBAC	*ABAC*	*EBAC*	*ERABAC*
5	25	20	15	10
10	30	25	20	15
15	32	27	22	17
20	35	30	25	20
25	40	35	30	25
30	45	40	35	30

Figure 2 shows the graphical representation of computation overhead between RBAC, ABAC, EBAC and ERABAC access control methods. X axis represents the Number of attributes and Y axis represents the computation time in milliseconds. When number of attribute is 10, the computation time of RBAC is 30 ms, ABAC is 25 ms, EBAC is 20 ms and ERABAC is 15 ms. When the number of attributes is 15, the computation time of RBAC is 32 ms, ABAC is 27 ms, EBAC is 22 ms and ERABAC is 17 ms. When the number of attributes is 20, the computation time of RBAC is 35 ms, ABAC is 30 ms, EBAC is 25 ms and ERABAC is 20 ms. When the number of attributes is 25, the computation time of RBAC is 40 ms, ABAC is 35 ms, EBAC is 30 ms and ERABAC is 25 ms. When the number of attributes is 30, the computation time of RBAC is 45 ms, ABAC is 40 ms, EBAC is 35 ms and ERABAC is 30 ms. From the analysis it is proved that the proposed ERABAC has better computation overhead than the other methods.

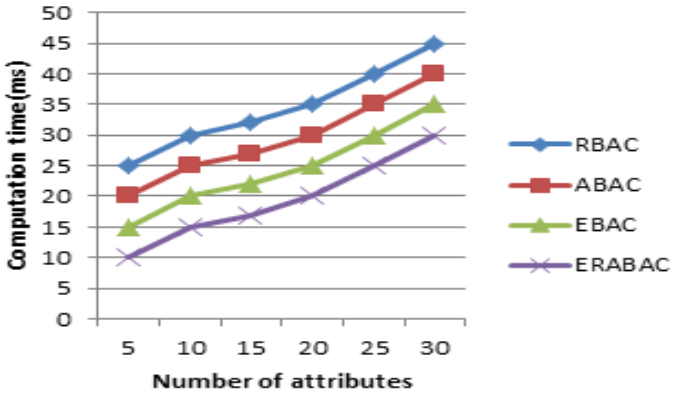

Figure 2. Comparison of computation overhead

5. CONCLUSION

In this paper, a fine grained event based access control mechanism is proposed which consider both the role and attributes in dataset. Initially a data in the dataset is defined through fined grained description. Then an event based fine grained access control is developed on top of the description which considers the temporal state, environmental state, roles and attributes of data. This provides a flexible boundary for access control based on an event. It improves the security of big data. The experiments are conducted in healthcare data in terms of computational overhead.

REFERENCES

[1] Yang, K., Han, Q., Li, H., Zheng, K., Su, Z., & Shen, X. An efficient and fine-grained big data access control scheme with privacy-preserving policy. *IEEE Internet of Things Journal*, *4*(2), 563-571, 2017.

[2] Reddy, Y. B. Access control for sensitive data in hadoop distributed file systems. In *Third International Conference on Advanced Communications and Computation, INFOCOMP*, 17-22, 2013.
[3] Longstaff, J., & Noble, J. Attribute Based Access Control for Big Data Applications by Query Modification. In *Big Data Computing Service and Applications (BigDataService) IEEE Second International Conference on* IEEE, 58-65, 2016.
[4] Aedo, I., Díaz, P., & Sanz, D. An RBAC Model-Based Approach to Specify the Access Policies of Web-Based Emergency Information Systems. In International Journal Of Intelligent Control and Systems, 2(4), 272-283, 2006.
[5] Bhide, M., Pandey, S., Gupta, A., & Mohania, M. Dynamic access control framework based on events: a demonstration. In *Data Engineering, 2003. Proceedings. 19th International Conference on* IEEE, 765-767, 2003.
[6] Amir, M., Pillai, P., & Hu, Y. F. Cascading Permissions Policy Model for Token-Based Access Control in the Web of Things. In *Future Internet of Things and Cloud (FiCloud) 2014 International Conference on* IEEE, 238-245, 2014.
[7] Hu, V. C., Grance, T., Ferraiolo, D. F., & Kuhn, D. R. An access control scheme for big data processing. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on* IEEE, 1-7, 2014.
[8] Huang, J., Nicol, D. M., Bobba, R., & Huh, J. H. A framework integrating attribute-based policies into role-based access control. In *Proceedings of the 17th ACM symposium on Access Control Models and Technologies* ACM, 187-196, 2012.
[9] Khan, M. F. F., & Sakamura, K. (2015, May). Fine-grained access control to medical records in digital healthcare enterprises. In *Networks, Computers and Communications (ISNCC) 2015 International Symposium on* IEEE, 1-6, 2015.
[10] Kuhn, D. R. Vulnerability hierarchies in access control configurations. In *Configuration Analytics and Automation (SAFECONFIG), 2011 4th Symposium on* IEEE, 1-9, 2011.
[11] Gupta, A., Pandhi, K., Bindu, P. V., & Thilagam, P. S. (2016). Role and Access Based data Segregator for Security of Big Data. *Procedia Technology*, *24*, 1550-1557.
[12] Zeng, W., Yang, Y., & Luo, B. Content-based access control: Use data content to assist access control for large-scale content-centric databases. In *Big Data (Big Data), 2014 IEEE International Conference on* IEEE, 701-710, 2014.
[13] Rajpoot, Q. M., Jensen, C. D., & Krishnan, R. Attributes enhanced role-based access control model. In *International Conference on Trust and Privacy in Digital Business* Springer International Publishing, 3-17, 2015.