ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

An Automatic Mechanism for Converting Existing RBAC System's Roles into ABAC System Policies Using Machine Learning

Palak J Manjrawala, Prof. Gurucharan Sahani

M.E Student, Dept. of C.E., SVIT, Vasad, India

Asst. Professor, Dept. of C.E., SVIT, Vasad, India

ABSTRACT : Recently, there has been considerable interest in attribute-based access control (ABAC) to overcome the limitations of the classical access control models (i.e, discretionary-DAC, mandatory-MAC and role based-RBAC) while unifying their advantages. The general idea of ABAC is to determine access control based on the attributes of involved entities. Example user attributes are department, clearance and role and example object attributes are size, create_Time and owner. Authorization results are computed based on subject and object attributes and authorization policies. As attributes can be engineered to reflect appropriately detailed information about users, subjects and objects, ABAC ensures great flexibility in expressing fine-grained policies which are increasingly required by applications. Till today no method has been defined which guides for automatic conversion of RBAC to ABAC, So in this paper efforts for automotive conversion of Role Base Access Control Model to Attribute Base Access Control Model has been done. Using Log Files of RBAC system, the transactional records are analysed using Machine Learning Algorithms – Navies Bayes Theorem, to generate access patterns from which various needed attribute for generation of XACML policy is fetched and from that XACML policies are generated.

KEYWORDS : Role Based Access Control Model, Attribute Based Access Control Model, Log Files. XACML Policy, Machine Learning-Navies Bayes Algorithm.

I. INTRODUCTION

In the era of Information and resources to be highly secured the access control plays an important role. Access Control means performing selective restriction to a place or to any resource of the system for system users. Granting permission to access any of the resources of the system to the users is called authorization. Access control can also be defined as the process by which users are granted access and certain privileges are assigned to access the information and various resources of the system.

In an access control system users are assigned various credentials, prior they are granted access to the system. In information security, access control has various mechanisms such as authorization, authentication and audit of the entity trying to gain access which ensures the security of the system. Access control models have a subject - referred to human user, the one trying to gain access and an object is usually the software or system resource. In the world of information security, an access control list includes various parameters such as list of permissions and the users to whom these permissions are associated. Data which is confidential or have certain security associated with it can be viewed by certain people and not by other people and is controlled by access control. Due to this mechanism administrator has rights to secure information and set privileges for what information can be accessed, who can access it and at which time it can be accessed.

DAC : DAC also known as discretionary access control is an access control mechanism that gives grant or keep restriction for an access to an object by an access policy which is defined by an object's owner group or by the

subjects [15].DAC mechanism controls are defined by user identification with supplied credentials during authentication, such as username and password. DACs are discretionary because the subject (owner) can transfer

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

authenticated objects or information access to other users. In other words, the owner determines object access privileges [15]. In DAC, each system object (file or data object) has an owner, and each initial object owner is the subject that causes its creation. Thus, an object's access policy is determined by its owner. A typical example of DAC is Unix file mode, which defines the read, write and execute permissions in each of the three bits for each user, group and others. In DAC user have the ability to transfer ownership of the object to other user and he can also regulate the access pattern of the other users of the system. Though DAC is easy to Implement but it has vulnerabilities for Trojan horse and has limited negative authorization power.

MAC : Media access control (MAC) is a sublayer of the data link layer (DLL) in the seven-layer OSI network reference model. users have the authority to decide whether to grant access to any other user. To allow that, all users have clearances for all data. MAC is responsible for the transmission of data packets to and from the network-interface card, and to and from another remotely shared channel [16].The basic function of MAC is to provide an addressing mechanism and channel access so that each node available on a network can communicate with other nodes available on the same or other networks. Sometimes people refer to this as the MAC layer [16]

ACL : An access control list (ACL) is a table that tells a computer operating system which access rights each user has to a particular system object, such as a file directory or individual file. Each object has a security attribute that identifies its access control list. The list has an entry for each system user with access privileges. The most common privileges include the ability to read a file (or all the files in a directory), to write to the file or files, and to execute the file (if it is an executable file, or program).

RBAC : Role-based access control (RBAC) is a method of access security that is based on a person's role within a business. Role-based access control is a way to provide security because it only allows employees to access information they need to do their jobs, while preventing them from accessing additional information that is not relevant to them. An employee's role determines the permissions he or she is granted and ensures that lower level employees are not able to access sensitive information or perform high-level tasks. Role-based access control is another mechanism that is used to authorize access to operations based on a caller's role membership and is mostly used in Web applications requiring scalability.

ABAC : Attribute-based access control (ABAC) is a different approach to access control in which access rights are granted through the use of policies made up of attributes working together. ABAC uses attributes as the building blocks to define access control rules and access requests. This is done through a structured language called the eXtensible Access Control Markup Language (XACML), which is as easy to read or write as a natural language.

II. RELATED WORK

Classical access control model such as DAC, MAC and RBAC are not suitable for all application areas. They are not so much flexible and scalable as required in large, distributed, open and cloud computing application environments. Recently, there has been considerable research carried out in the area of access control. Recently, there has been considerable interest in attribute-based access control (ABAC) to overcome the limitations of the classical access control models (i.e, discretionary-DAC, mandatory- MAC and role based-RBAC) while unifying their advantages. Research on ABAC has been carried out in various areas.

Yu et al [5] proposed access control for digital health records in clouds. For example, in storing the digital health records, an owner is a patient who decides the access policy. In the Setup and KeyGen phases, the KDC gives attributes and secret keys to users, who register in the system. There can be more than one KDC (Details in the next section). The owner encrypts (Encrypt algorithm) the data using the access policy she has and the public parameters. The encrypted data is then stored in clouds. A user with a valid set of attributes can download the encrypted data from the cloud and decrypt it using its secret keys (Decrypt algorithm). When new users join the system, they register with the CPS and receive attributes and keys from the user. When a user is revoked or the access policy is changed, then the KDC redistributes keys in such a way, that the revoked users are not able to decrypt any stored data. Thus the success of an access control mechanism rests on not only how efficient the encryption and decryption mechanisms are, but also on the revocation schemes and distributed structure of KDC.

Kuhn et al [4] suggested that combining the best features of RBAC and ABAC can provide effective access control for distributed and changing applications. They compared the strength and weakness of RBAC and ABAC with respect to simplicity of security administration, easiness of reviewing permissions assigned to users, and flexibility to adapt to rapid changing applications; then they presented a spectrum of possible ways to combine RBAC and ABAC; finally,

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

they revealed that standards organizations are developing a policy-enhanced RBAC standard to accommodate attribute based features. They presented a very interesting picture of the landscape of combining RBAC and ABAC. However, the attributes they considered are limited to user-centered attributes. The attributes of objects and environment are also important to access control, and should be considered.

Earlier, Al-Kahtani and Sandhu [1] proposed a model of rule-based automatic user-role assignment for RBAC, called RB-RBAC, to overcome the difficulty of manual user-role assignment for service-providing enterprises which typically have a huge number of users. In this model, users are dynamically assigned roles by using rules, based on users' attributes. This model also has the limitation of considering only the attributes of users; furthermore, attributes are expressed in propositional logic, thus being less expressive than what we permit (first order logic). In addition their approach to representing mandatory access control is to create roles of read and write for every node in a security lattice. This approach can lead to a large number of roles; more importantly, the roles are created based on the general security classification lattice rather than specific job functions; this makes it difficult to realize the principle of least privilege.

Similar to [1], Kern and Walhorn [3] also adopted a rule based approach to user-role assignment, supporting automated administration of roles in large organizations. They pointed out that dynamic user-role assignment as in [1] creates difficulties in reviewing permission assignments and in evaluating the impact of a new rule or revision of rules. To overcome these problems, they proposed static user-role assignment. In this way, the rule-based user-role assignment is separated from run-time RBAC system. This approach has been applied in a bank and identity management solution of an IT service provider.

Chae and Shiri [2] proposed a variant of RBAC to categorize objects in hierarchical classes (more exactly, groups), to enable association of an object group rather than an individual object with an operation in permission, and to allow authorization propagation through object group hierarchy. Compared to [1, 3, 4], this model deals with the difficulty of handling a large number of objects.

III. PROPOSED WORK FLOW

As discussed in related work, numerous efforts have been done to convert existing access control mechanism to attribute based access control model. But all research work up till now throws light on doing this conversion manually. For manual conversion of Role Based Access control system to attribute based access control system all attributes have to be taken into consideration.

Different attributes of user, subject, object and various environmental conditions are grouped together and various possible combination of all the attributes leads to generate an XACML policy based on the authorization rights of the user. Here for generating XACML policy for access control also includes various Environmental conditions like time_of_access, location_of_ user and many more information are fetched from system file, user database file, and from Data dictionary.

As the work flow indicates initially the input to the system is the transactional record information stored in log files of Role Based Access Control enabled system. Then each record for the file are categorized on the basis of role and related to access pattern the fine granularity is obtained by generating a nested categorization for each user following in some fixed group of user or say some particular role. From this categorization and fetching more information from system file, user data base, data dictionary a master table is created. From this categorization various attributes like user attributes, subject attributes, object attributes and environmental condition are listed down. This information works as Meta data for the next step, which is to generate XACML files for policy which is helpful in access control.

Here after generating XACML file automatically, admin has rights for the policy pruning, admin can add new policy, can delete redundant policy, can add new user, can truncate older details of the employee and then finally can configure the final policy.

As the proposed workflow indicates the categorization of the log file on the basis of Role that the user has been assigned, the object he is accessing, the operation the he is performing on the object, the date and time on which such objects are accessed are all categorized with the help of Machine Learning.

From various Machine Learning algorithm Naive Bayesis used. Naive Bayesis was used for text retrieval community in the early 1960s and remains a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including support vector machines.

Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers. Here Naive Bayesis algorithm is used for categorizing the file on the basis of probability that various records have been accessed. More the frequency of record higher te probability that record will be considered for access pattern and that access pattern will be converted in XACML Policy .

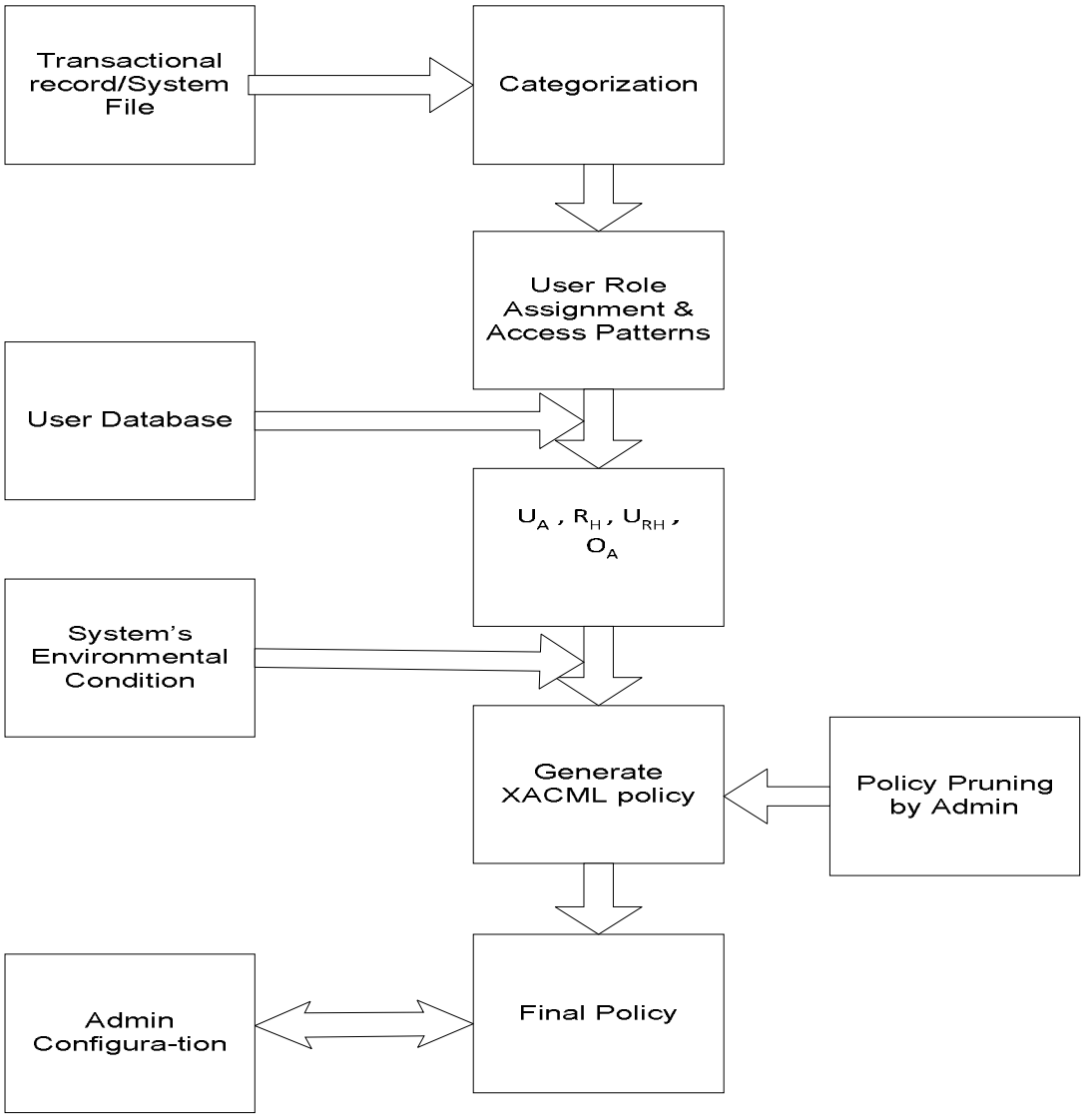

Figure 1 : Proposed work flow

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

IV. PROGRESS WORKFLOW

Below Screen shots shows how our proposed approach will work. Transaction records contained in the log file of Role Based Access Control model has been categorized on the basis of role every user has been assigned, then on the basis of object that the user is accessing, the operation he performs on the accessed object, and then carrying out the various other attributes like at which date and time such actions have been done.
By this various attributes like user attributes, subject attributes, object attributes can be derived from log file of Role Base Access Control model. This categorization has been done using machine learning algorithm – Navie's Bayes Classifier. By Using this algorithm initially probability of each Role is obtained. After it probability of other attributes are found. Then by using the joint probability the access pattern are generated. Generated access pattern turns out to be the attributes for XACML policy

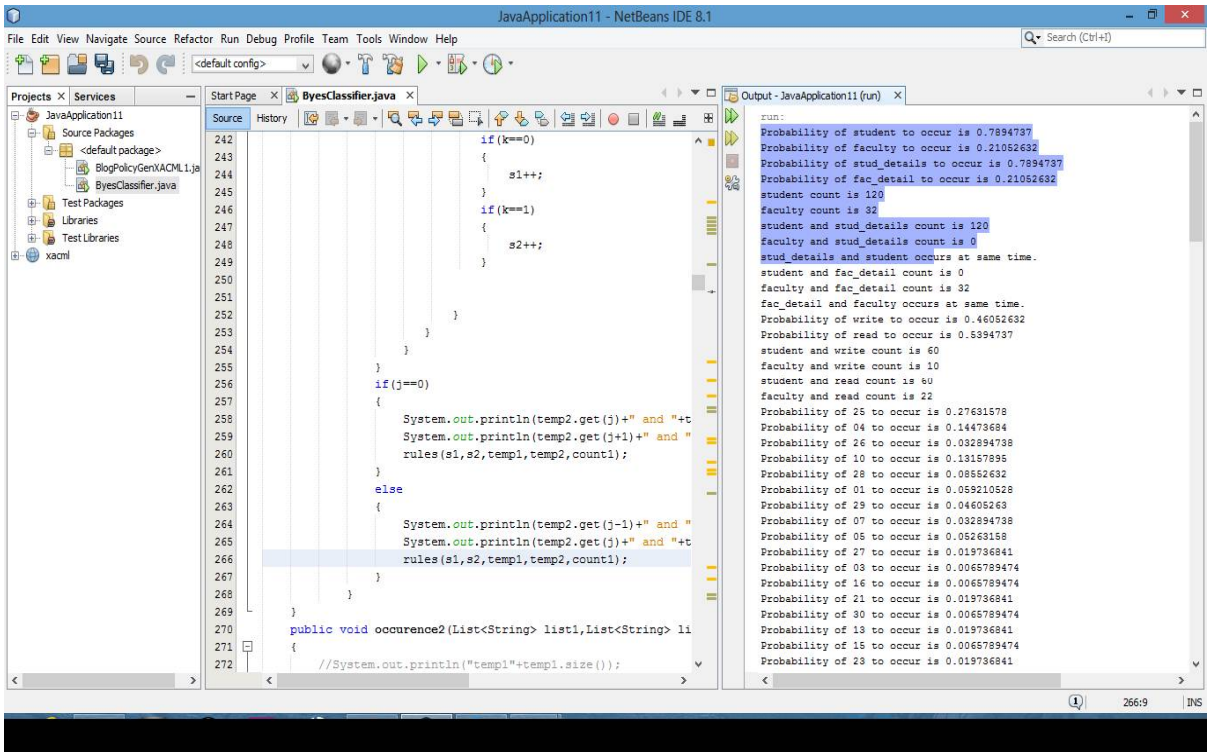

Fig 2 : Showing Probablity of each role

Above figure 2 shows the probability of each role of Role Based Access Control model on the basis of log file, for the resources they are accessing.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

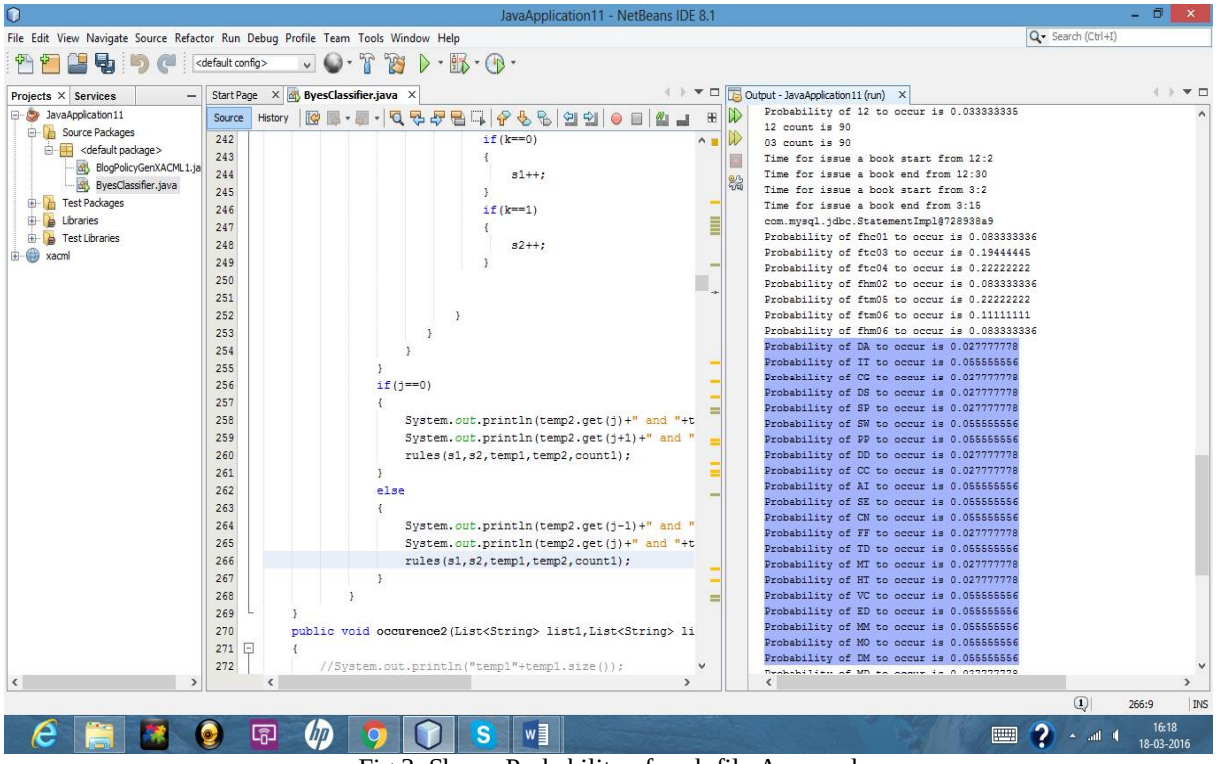

Fig 3: Shows Probability of each file Accessed

Above figure 3 shows the probability of each resource accessed of Role Based Access Control model on the basis of log file, by various roles of Role Based Access Control Model.

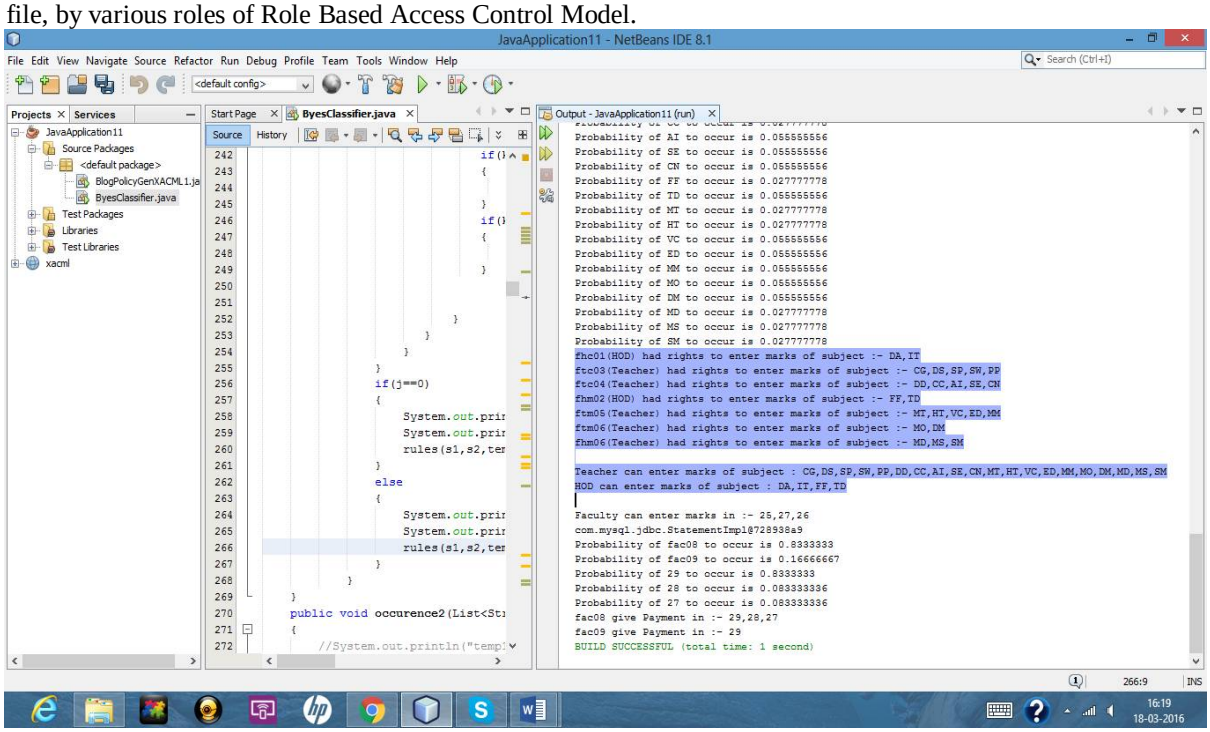

Fig 4: Shows Probability of action performed on the basis of date and time.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Above figure 4 shows the probability of each resource accessed of Role Based Access Control model on the basis of date and time of access.

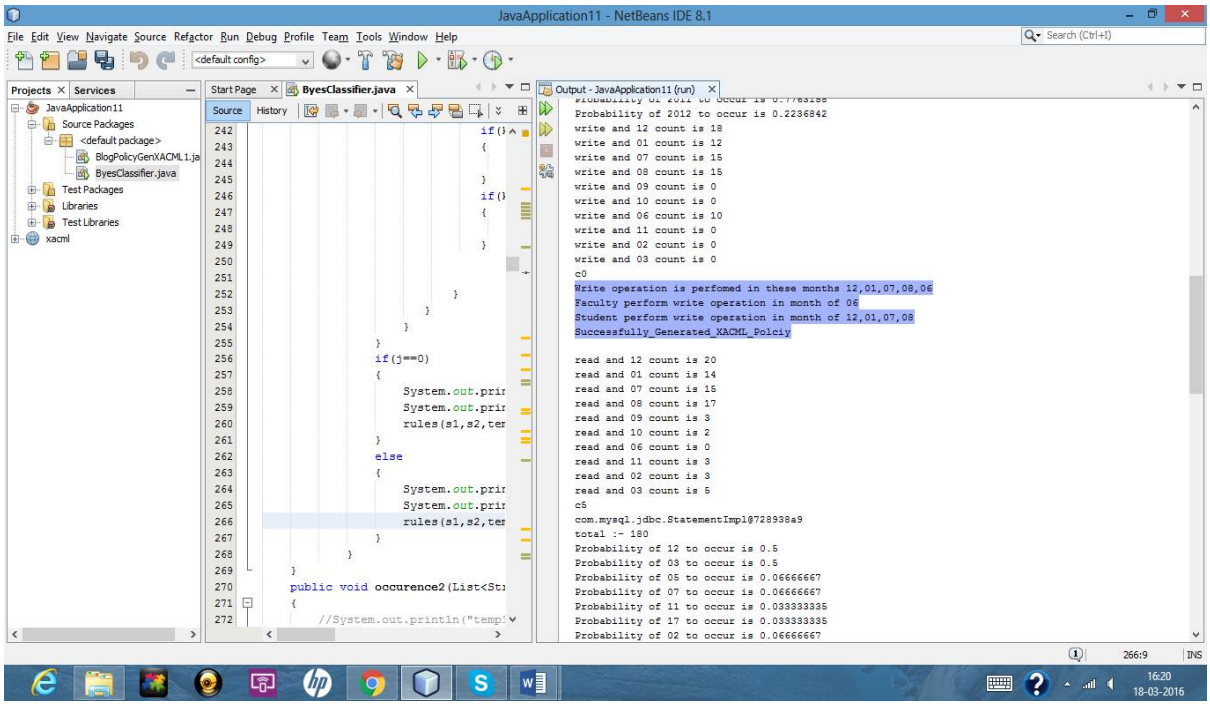

Fig 5: Show the access Pattern Generated.

Above figure 5 shows the access pattern generated on the basis of log file by finding the joint probability of resources being accessed by the different users on some specific time.

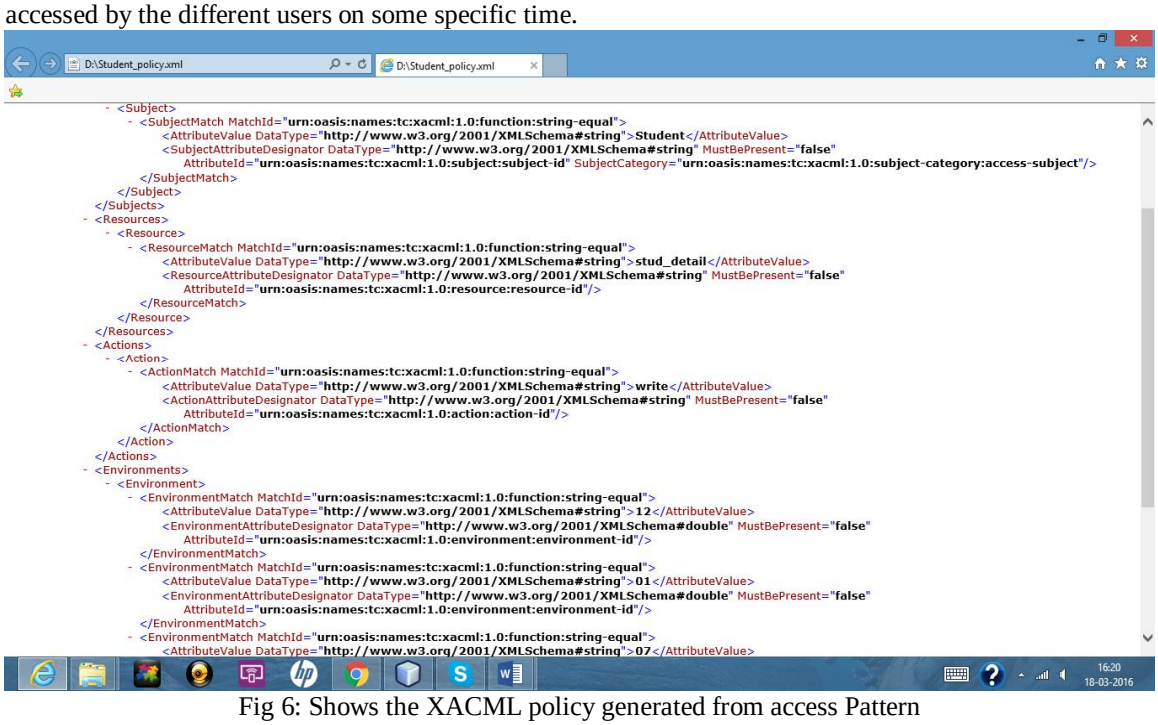

Fig 6: Shows the XACML policy generated from access Pattern

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Above figure 6 Shows the XACML policy generated from access Pattern on the basis of access pattern generated by applying Machine learning algorithm on the log file of Role Based Access Control Model

V. ANALYSIS

The above described worked flow is able to generate the XACML policy from the log records of Role Based Accessed Control Model. This portion ensures that the proposed scheme is able to generate similar XACML policy in the same way the role based access control would allow the authorization of authenticated user. Consider for example RBAC system in which certain roles has been defined and each role has certain permissions on various objects. Assume these roles have been defined as per table 1.

Role_id	Computer Graphics	Distributed System	Parallel Processing	Information technology	Software Engineering
R1	r/w	r/w	r/w	r/w	r/w
R2	r/w	r/w	r	r	R
R3	R	R	r/w	r	r/w

Table 1: Role definition in RBAC system

Figure 7: Generated XACML Policy

Suppose, role R2 has been assigned to faculty with Id " Ftc03". Here as shown in table 1 the faculty with Id " Ftc03 " has read/write permission to the subject file Computer Graphics and Distributed System according to Role Based Access Control Model. XACML policy generated in Figure 7 also decides the authorization of user Ftc03 in the same way. Machine Learning Concepts to the log file of Role Based Access Control Model fetches the needed attribute for XACML Policy and generates the policy that grants the access to the user on the basis of predefined access control mechanism..

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

VI. CONCLUSION

Attribute Based Access Control is one type of access control mechanism. There are many access control mechanism but attribute based access control mechanism provides more flexible environment as access control policy are based on attributes.

As this dissertation leads to generate Attribute based access control model automatically from existing Role based access control model. Here transactional records of Role based access control model are used to fetch attributes for access control policy. Using this Attributes XACML policy file is generated and admin has rights to modify policy and then configures the final policy. Till date a proto-type for this model have been created where log file has been categorised using Machine Learning Algorithm- Navie's Bayes Classifier. Access pattern obtained from machine learning algorithm serves attributes needed for creating XACML Policy. An XACML policy has also been generated.

REFERENCES

[1] M. A. Al-Kahtani and R. Sandhu. A model for attribute-based user-role assignment. ACSAC '2002.
[2] J. H. Chae and N. Shiri. Formalization of rbac policy with object class hierarchy. In Proceedings of the 3rd international conference on Information security practice and experience, ISPEC'07, pages 162–176,
Berlin, Heidelberg, 2007. Springer-Verlag.
[3] A. Kern and C. Walhorn. Rule support for role-based access control. SACMAT '2005, pages 130–138. ACM, 2005.
[4] D. R. Kuhn, E. J. Coyne, and T. R. Weil. Adding attributes to role-based access control. Computer,
43(6):79 –81, June 2010.
[5] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in ACM ASIACCS, D. Feng, D. A. Basin, and P. Liu, Eds. ACM, 2010, pp. 261–270.
[6]. Wang, L., Wijesekera, D., Jajodia, S.: A logic-based framework for attribute based access control. In: 2nd ACM Workshop on FMSE (2004)
[7]. Yong, J., Bertino, E., Toleman, M., Roberts, D.: Extended RBAC with role attributes. In: 10th Pacific Asia Conf. on Info. Sys. (2006)
[8]. Giuri, L., Iglio, P.: Role templates for content-based access control. In: ACM Workshop on RBAC (1997)
[9]. Jajodia, S., Samarati, P., Sapino, M.L., Subrahmanian, V.S.: Flexible support for multiple access control policies. ACM Trans. Database Syst. (2001)
[10]. Evered, M.: Supporting parameterised roles with object-based access control. In: HICSS (2003)
[11]. Fischer, J., Marino, D., Majumdar, R., Millstein, T.: Fine-Grained Access Control with Object-Sensitive Roles. In: Drossopoulou, S. (ed.) ECOOP 2009. LNCS, vol. 5653, pp. 173–194. Springer, Heidelberg (2009)
[12]. Abdallah, A.E., Khayat, E.J.: A formal model for parameterized role-based access control. In: Formal Aspects in Security and Trust (2004)
[13]. Al-Kahtani, M.A., Sandhu, R.S.: A model for attribute-based user-role assignment. In: ACSAC (2002)
[14]. Bertino, E., Catania, B., Ferrari, E., Perlasca, P.: A logical framework for reasoning about access control models. In: SACMAT (2001)
[15] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. IEEE Computer, 1996.
[16] Ravi S. Sandhu and P. Samarati. Access control: Principles and practice. IEEE Com. Mag.,1994.
[17] M.J. Moyer and M. Abamad. Generalized role-based access control. In International Conference on Distributed Computing Systems, 2001.

BIOGRAPHY

Palak Manjrawala is a M.E student in the Computer Engineering Department, Sardar Vallabdhbhai Patel Institute of Technology, Vasad and Pursuing Master of Engineering (ME) degree in 2016 from SVIT, Vasad, Anand, India. Research interests are Information Security, Role Based Access Control Model, Attribute Base Access Control Model.