Volume 7, Issue 5, May 2017 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Dynamic Access Control in a Document Data Store

Kriti Srivastava
PhD Scholar, Computer Engineering,
Dwarkadas J Sanghvi College of Engineering,
Mumbai, India

Priyal Shah
Student, Information Technology,
Dwarkadas J Sanghvi College of Engineering,
Mumbai, India

Khushali Shah
Student, Information Technology,
Dwarkadas J Sanghvi College of Engineering,
Mumbai, India

Dr. Narendra Shekokar
HOD,
Computer Engineering,
Dwarkadas J Sanghvi College of Engineering,
Mumbai, India

DOI: 10.23956/ijarcsse/V7I5/0107

Abstract— The wide expansion of the Internet has set new requirements for access control policy specification. Due to the rapid changes and continuous communication between organizations applications can no longer be standalone or isolated from each other. In a large dynamic and heterogeneous environment applying policies to control the access provided o the users becomes very challenging. All policies that are created always go through various minor or major changes. Applying access control policies becomes challenging in a large, heterogeneous, and dynamic environment. There has been a rapid evolution in computing paradigm that has created the need for data storage as agile and scalable as the applications they support. NoSQL has emerged as the latest technology to be used by various organizations hence the provision of security is a major concern. In order to implement a security framework, it is mandatory to provide real time and on demand access control management approach that should take care of: Data integration and sanitation, multi-tenancy, user identity along with relation between the different users and the objects being accessed. With the aim of encapsulating persistent goals of policies we introduce extensions in the form of policies.

Keywords- Large-scale Systems, Security, Policy-based access control, No SQL databases

I. INTRODUCTION

Ecommerce applications focus on transforming the national economies from manufacturing-based paradigms to digitally available knowledge-based systems. Their most important asset is to store and process vast quantities of sensitive data and ensures that the application provides scalability and the users who requested to access the resource is authorized user. E-commerce security has its own particular shades and is one of the highest visible security components that affect the end user through their daily payment transactions.

In traditional ecommerce applications relational databases were used to provide better security to applications, but with the increase in the number of users and increase in the number of transactions to access the resources, scalability is of major concern. For achieving better scalability and availability they need to be dependent on NoSQL but all the NoSQL stores are having issues with respect to security.

In NoSQL, there is no in-built feature for authentication and access control. Ecommerce business model which used NoSQL data store needs to have a access control module implemented separately on the data layer provided by the NoSQL database. [1]

This access control mechanism should be dynamic in nature to adapt all the changes in the system. We consider the security requirements which include authentication, authorization and privacy, and discuss how these requirements can be met with access control model. A secure-ecommerce framework includes functions for authentication, dynamic access control and trust management for clients as well as service providers or companies.

Various policies are needed to ensure that users access only the information they are authorized to. The policies should be expressed in an expressive manner which can contribute to the e-commerce model of business. The model of access control is thus presented as a disjoined solution and needs to be implemented as a separate layer of security. To resolve the security issues in ecommerce applications, this paper presents the access control based on policies in NoSql database (MongoDB) and it is used for implementing security as well as scalability. It will provide policy management and generates dynamic policies will take into consideration the permission and access control rules. PBAC is more flexible than RBAC and supports dynamism.

The rest of the paper is organized as follows; Section II describes the need for PBAC in ecommerce. Section III discusses the .Section IV describes the database used. Section V describes statement for performance evaluation. Section VI is the conclusion and future scope. document is a template. An electronic copy can be downloaded from the Journal website. For questions on paper guidelines, please contact the journal publications committee as indicated on the journal website. Information about final paper submission is available from the conference website.

II. NEED FOR PBAC IN ECOMMERCE WEBSITE

As mentioned in the above section ecommerce websites are switching to NoSQL database servers to provide scalability to their users but this scalability can cost their security which can be handled then by access control mechanisms.

The main aim of the model is to protect resources from unauthorized access, and ensuring authorized access. Resources are generally accessed through some application, which in our case is an ecommerce website, which enforces access control restrictions by allowing only authorized access to its users. The access restriction of the subject is implemented and configured the sharing of information and resources is minimized which in turn increases the security of the entire model.

Focusing on different requests of security, researchers design many access control models, such as Mandatory Access Control (MAC), Discretionary Access Control (DAC), Attribute-based Access Control (ABAC) and Rolebased Access Control (RBAC), and. The RBAC model has received the most popularity amongst all of the models. Though these models have been able to provide security to their fullest yet some limitations are there which needs to be overcome. [1]

Firstly, the session is not extensively taken into consideration. All the models take the session as subject oriented that is restrictions are put only on the subject. The session being subject oriented doesn't allow the developer to consider the other elements like object and environment of the session. Secondly, none of these models support dynamism because all the logic written is hard coded and changes to such models is difficult especially if the changes are at runtime. [1]

Because of the above disadvantages PBAC should be used for NoSQL servers. PBAC combines attributes from the resource, the environment, and the requester with information on the particular set of circumstances under which the access is request, and uses rule sets that specify whether the access is allowed under policy for those attributes under those circumstances. [2]

Policy Based Access Control is a strategy for managing user access to system resources, where business classification of users is combined with policies to determine what access privileges a user should have. The support for PBAC is incorporated in our proposed architecture to introduce access control and contribute towards a privacy framework in the relatively new NoSQL databases so that it can be used by ecommerce applications.

III. IMPLEMENTATION

System Framework: As shown in figure 1.user interacts with the Ecommerce website. User requests access to the data object to perform operation on the object and along with it the RTC (RunTimeContext) will be set for that particular session which tells about whom the user is and what work does it have with the system. It is verified if the session belongs to that particular user with its credentials for logging in. The request will be sent to the MongoDB who will in turn contact the policy control module to decide whether to grant or deny the access. The module will accept the parameters from the ecommerce website and check whether the user is an authenticated user or not and Access Computation Engine will take the decision whether to grant access or deny access depending upon the access control policies.

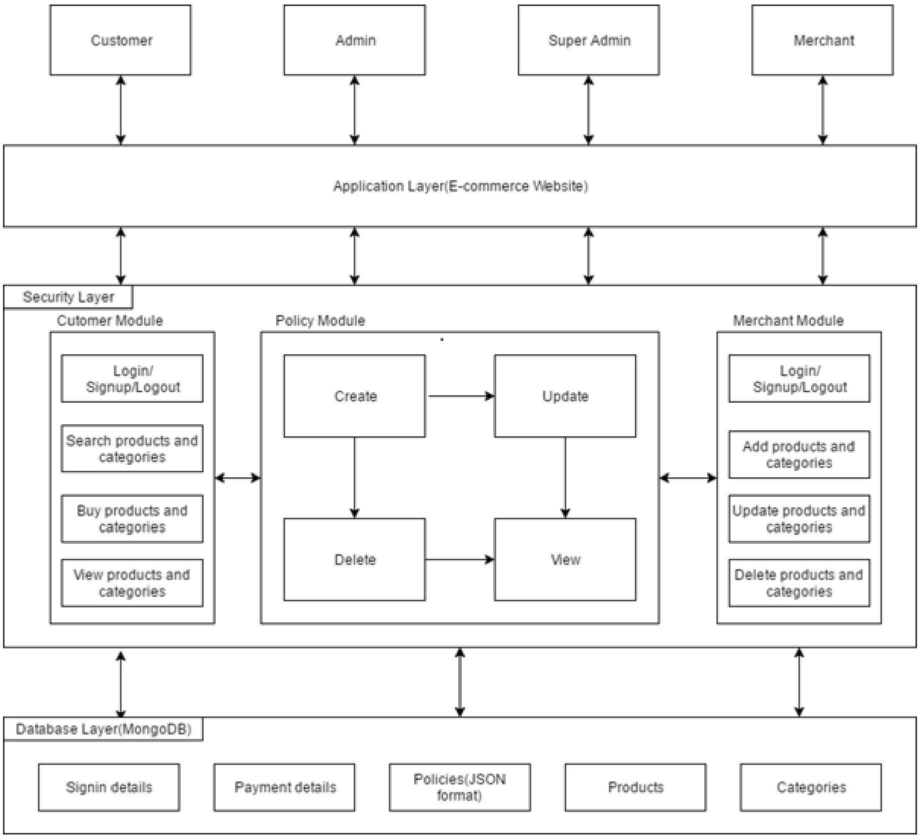

Fig. 1 Propose System Architecture

1. Ecommerce Application Layer

1.1 User interacts with the Ecommerce website.

1.2 User requests access to the data object to perform operation on the object.

1.3. It is verified if the session belongs to that particular user.

1.4. The request will be sent to the MongoDB who will in turn contact the agent to decide whether to grant or deny the access.

2. Security Layer (Access Computation Engine (ACE) Module)

The Security Layer will accept the parameters from the ecommerce website and will ask the Access Computation Engine (ACE) whether the user is an authenticated user or not and Access Computation Engine will take the decision depending upon the access control policies.

Access Computation Engine consists of Access control PoliciesACP={R, P, C} where

R= {Rn,Rctx,S}

Rn- Role name,

Rctx- Enable, Disable or active Role context i.e. context for enabling disabling or active state, State, checks whether it is in enable, disable or active state.

P= {O, OP} where P is the target policy and it consists of the operation performed on that object.

C= Context for that object.

3. Database (MongoDB)

MongoDB is an open source database that uses a document-oriented data model. MongoDB is one of several database types to arise in the mid-2000s under the NoSQL banner. Instead of using tables and rows as in relational databases, MongoDB is built on architecture of collections and documents. Store policy information like search policy information, retrieve policy information, information regarding different users. d. Policies: The policies described in the above section should have all the attributes related to the decision making so that real time decision can be made to provide access to the requesting users. Many parameters should be considered before creating and implementing the policies like [7]:

a. Access Control Procedures: All Business Systems must create, and adopt to a formal, documented access control procedure that addresses purpose, scope, roles, responsibilities, management commitment, and coordination among entities.

b. User Account Management: They must have the following attributes to take the decision:

i. Identify role types.

ii. Identify authorized users of the information asset and specifying access privileges.

iii. Require appropriate approvals for requests to established roles.

iv. Establish, activate, modify, disable, and remove roles.

v. Specifically authorize and monitor the use of guest/anonymous accounts.

vi. Grant access to the system based on

(1) valid access authorization,

(2) intended system usage, and

(3) other environment attributes as required by the security layer.

c. Least Privilege: if a particular user has more than one role, the role with the least privilege according to the context of the session should be considered while giving the authority.

d. Locking the session: If the session is not being accessed for 120 minutes since the last request of the user the access should be prevented to the information or resources. And to add more restriction the entire validation and authentication of the user should be done again.

IV. DATABASE

The data structures used by NoSQL databases are by default different from those used in relational databases. NoSQL database's suitability depends on the problem it must solve. The data structures used by NoSQL databases are more flexible than relational database tables.[9]

Key aspects in NoSQL adoption

1. Sparse and semi-structured data are allowed by dynamic schemas.
2. Dynamic ecommerce operations allows schema-less designs.
3. Works best for read-heavy scenarios.
4. Instead of expensive, monolithic architecture, it is best suited for scalable, efficient, scale-out and distributed architecture.
5. It consists of features such as Auto sharding, integrated caching and replication.
6. It is open source that makes a biggest advantage leading to cost effectiveness.
7. It offers faster business analytics and reporting strategies.
8. It provides improved performance, scalability and high availability.[3]

Advantages of NoSQL:

1. Elastic scalability: Expansion of new nodes gives an advantage for NoSQL databases. The databases are designed for use with low-cost hardware. NoSQL databases are a better fit as upward scalability is being replaced by outward scalability

2. Big data applications: As there is need to store massive sets of data, the transaction rates are also growing at a greater extent. It is difficult to handle large sets of data in RDBMS. These large sets of data are easily handled by NoSQL databases. [4]
3. Database administration: NoSQL databases require less management, with data distribution and auto repair capabilities as well as fewer administration and tuning requirements whereas in RDBMS databases require expensive installation and complex administration. There is always a need of taking into consideration the performance and availability of databases.
4. Economy: In RDBMS installation of storage systems and servers is expensive while NoSQL databases can be easily installed in cheap hardware clusters. Therefore storing and processing of data can be done at much less cost.

Disadvantages of NoSQL:

1. Less Mature: Many key features of NoSQL databases are yet to be implemented as still there are pre-production versions of the database.
2. Less Support: Only a few firms offer support for each NoSQL database, these companies often are small start-ups without the global reach.
3. Less Secure: NoSQL technologies cannot be secure with the increasing adoption in Big Data and its numerous applications alone. In order to incorporate security framework access control management should take care of:
    a) User identity- User’s identity should be authenticated against different policies and depending upon that user should grant access or deny access on a particular data object.
    b) Multi-tenancy- Multi-tenancy refers to a principle in software architecture where a single instance of the runs on a server serving multiple customers (tenants).This is a really important part of important feature. This is important because in multi-tenant environment customers do not share or see each other’s data.
    c) Relation between different users and the resources-User session should verify whether the session belongs to that particular user and whether he/she can access the data[5][6]

V. STATEMENT FOR PERFORMANCE EVALUATION

The In this section, we present the performance prediction module of our model. The main goal is to achieve all the requirements necessary for the proposed ACL module component.

1. Muti-tenancy: Each user can have different roles and the role that is applicable in that particular session is being considered and then access is provided.
2. Application based ACL: Queries are created and processed by the ACL engine that takes the decision in order apply the policies. Instead of reading entire database, it only reads ACL policy and generates only the applicable policies keeping the ecommerce website in mind.
3. Involvement of stakeholder: Organization has environment parameters, specific rules, operations and other related stakeholder's information that are stored. In policy decision making, ACL engine considers only the related stakeholder’s information.
4. Real-Time and On-Demand Access Control: Access Computation engine takes decision based on inclusive as well as exclusive policies based on heuristic data. [8]
5. Availability of resources: Any application can use any of the service for its particular purpose at any time and this component provides access control policies as endpoints.

As a NoSQL database is used, the performance needs to be evaluated using large amounts of data to check whether the policy module is being affected by them. As the products are divided into various categories each category can handle approximately 1 GB of data which corresponds to 10 Lakhs products in each category and then a lag of 3-7 seconds arrives. But the policy module where the access control is provided does not get affected by the size of the data as the module has been isolated from all other modules and only the required data is passed between these modules.

Many attacks can take place in a working ecommerce website, where the user continuously interacts with the website. A few have been taken care of in our project like the landing pages attack where if the URL is copied and pasted on another tab and the access is not provided. Just like the regular SQL injections to gain access to the website by using tautologies and union queries the website is tampered, same is the case of NoSQL injections and the attack is not possible in our project.

VI. CONCLUSION AND FUTURE SCOPE

Traditional RBAC is not able to guarantee a sufficiently fine-grained access control or specify constraints that should be applied to an access policy for non-structured data. In this paper we have incorporated the support for PBAC to introduce access control and contribute towards a much needed privacy framework in the relatively new NoSQL database. Policies will take into consideration the access control list/matrix as well as different parameters. The endorsement of attributes in policies will be used where a security level changes on the same data object depending upon various elements at runtime.

In our future version of model we plan to incorporate mechanism to address context-sensitivity when the requestor is a service outside the system by extending single domain to multi domain as well as incorporation of encryption algorithm for providing more security in the authorization process. [9]

REFERENCES

[1] Mansura Habiba, Md. Rafiqul Islam, A B M Shawkat Ali, *Access Control Management as a Service for NoSQL Big Data*, Computer Science and Engineering (APWC on CSE), 2015 2nd Asia-Pacific World Congress on, May 2016

[2] A Survey of Access Control Model- http://csrc.nist.gov/news_events/privilege-management-workshop/PvM-Model-Survey-Aug26-2009.pdf

[3] Scalable ecommerce with NoSQL Databases http://echidnainc.com/scalable-ecommerce-with-nosql-databases/

[4] Big data, http://www.mongodb.com/solutions/big-data

[5] Top five advantages and disadvantages of NoSQL-http://bigdata-madesimple.com/top-five-advantages-and-disadvantages-of-nosql/

[6] MongoDB, Inc. http://www.mongodb.com/

[7] Information Security Access Control Policy-https://www.maricopa.gov/technology/pdf/TEMPLATE_Information_Security_Acc

[8] Oracle NoSQL Database White Paper-http://www.oracle.com/technetwork/database/nosqldb/learnmore/nosql-database-498041.pdf

[9] An Access Control Model for NoSQL Databases-http://ir.lib.uwo.ca/cgi/viewcontent.cgi?article=3294&context=etd