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 6, June 2016

Security in Cloud Computing Environment by Decentralized Information Flow Control

Neeta R. Somavanshi[1], Prof. Y. B. Gurav[2]

M.E. Student, Dept. of Computer Engineering, TSSM's PVPIT, Pune, India[1]

Associate Professor, Dept. of Computer Engineering, TSSM's PVPIT, Pune, India[2]

ABSTRACT: The secrecy and integrity parameters about the sensitive data have been gaining lot of importance these days. Information flow control (IFC) has been gaining acceptance as a better methodology for protecting privacy without unduly restricting access to sensitive data. Information flow control is acquiring the grip of practical way for protecting the information related to programming languages and operating systems. Applications connect with database through the drivers but the database does not have any control how applications use the information.

We are proposing the scheme of information flow control based on decentralize model. The model is based on information flow. This model works on query by label concept and database includes the tags and labels. The principals are also defined in database. IFM uses a simpler model for security of information with a standard principal-based structure for authority management. This technique is expected to improve the time required to process the queries. Also programming overhead is minimized as most things are moved to database in the form of procedures and functions.

KEYWORDS: Information flow control, DIFC, Query by Label model, IFM.

I. INTRODUCTION

Online applications are becoming cosmopolitan while sharing and processing different data for diverse sources and providing security is a challenging task. The applications are becoming more complex, there is a need to deal with security problems. Different industries are dealing with this problem in different ways. Such as medical systems must take care of the protection of the patient's data from been misused, while social networking sites must provide better tools for providing security to the user's data.

The hypothesis that information flow control (IFC) provides convenient security than access control is given. Instead of using restricted access controls, the system follows the sensitive data and prevents data from being used or liberated inappropriately. The sensitive data is processed in both, the application as well as the database because the IFC allows it. The IFC enhances the security as it enforces end-to-end security policies.

Now a day we operate sensitive data which can be social, medical or related to financial or commercial through internet. The software which is organized to handle this data can be accessed through desktops or smart phones. If the software can be constructed, designed and monitored securely then only it is accepted trustily.

1.1 Security in Heterogeneous Systems

Heterogeneity is a main obstacle and bottle-neck of security in software. Multiple components are used to create a computing system. The components can be evaluated in various programming languages or it operates on separate platform. It has some limitations and the chances are to break the system. It is essential to ensure the security of the system.

1.2 Information-flow Control

Various methods for information flow control have been proposed and adopted, with some advantages and disadvantages. Type systems and language design have been common methods to ensure secure flow [3], but they generally require non trivial effort from the programmer to approve thenew ideas. Still these techniques traditional, as

DOI: 10.15680/IJIRCCE.2016. 0406097

they are accepted through worldwide. They reject safe programs as ill-typed and the current research is in improving this false positive rate.

The theoretical platform of IFC considers arbitrary lattice of labels. Decentralized Label Model (DLM) [2] includes a policy owner's concept, which is not similar with readers, followers, or trainers. It can enable the programmers to control who can use the labelled queries and who can change their labels by de-limitation. Insecure information flow can be the main reason for confidentiality and integrity attacks in computing systems. Confidentiality attacks can occur if the information leaks from sensitive databases to an attacker. Integrity at-tacks can occur if the data flows from entrusted sources to trusted descends. Many of the times the applications require to access confidential information to accomplish tasks. Ordinary access control mechanism if allow a program to read information, it cannot control the distribution of the information what it has read. Consider an example, if user is using a spread-sheetapplication about his investments. This spreadsheet application needs access to the confidential data of user to perform functions. This application must be a web application. In this paper further we will see: Section II talks about related work studied till now on topic. Section III shows proposed methodology and Design Section IV show System Architecture, Mathematical Model, Section V shows result and discussion while at the end we conclude the proposed work.

II. Related work

IFDB [1] is a database management sys-tem which supports Decentralized Information Flow Control to protect data with least over-head. It discovers query by label model which is very easy to use for developers. Suh et al. [4] describes a hardware mechanism which tracks flow of information. The authors adapt in what way the standard instructions propagates tags and store those tags in an extra cache. Information Flow Control tracing is done at the process level in operating system if it enforced IFC. Asbestos [5] is IFC-capable operating system. It is an operating system with high-security which is constructed upon information flow control. The labels are maps from principals to security levels. . Labels are composed of a finite map from principals to levels, plus a default level for the principals (categories) that are not mentioned in the map. The DStar system [6] and the further latest version of HiStar [10] use a very simple form of labels that is appropriate for our framework. Their labels are pairs of finite sets of schemas out of which one is for secrecy, one is for integrity.

Flume[8] proves that how the DIFC can benefits to the operating systems and applications. The programmers can apply strong security UNIX applications by using Flume. The la-belling and isolation in Asbestos [9] is used to bound software defects Laminar [12] proceeds a related methodology to Flume at the Operating System level with a security module applied to a standard Linux. Airavat [13] running Laminar, is a good example of cloud services; it brands the MapReduce and Hadoop file sys-tem. DIFC aware. Aeolus [7] uses a DIFC plat-form which is designed to build a distributed application like a web service where multiple users data is stored on multiple servers. Two main security issues are handled by Aeolus i.e. confidentiality and integrity. It explores the feasibility of deploying IFC as part of the next generation of secure cloud infrastructures. We review research on information flow tracking and enforcement and estimate data-centric security models.

Without getting affect for the efficiency of sharing objects, DEFCON system [11] improves strong object isolation to Java. The Decentralized Event Flow Control (DEFC) is introduced to focus the IFC requirements for event-based systems. DEFCON is applied to Java. IFC is used in SafeWeb [14] to track flow of data flows through all layers of the web application structure to ensure confidentiality and integrity.

III. PROPOSED METHODOLOGY AND DESIGN

The main objective of the application is to pro-vide to extension to the existing RDBMS model so that the application cannot breach through Database. We have to use the model of Query by label so that application component should not disclose more information than necessary.

A. Problem Definition

The security of cloud is very challenging be-cause of its widespread use. The information Flow Control and Decentralized information flow control both have the potential to increase the security of cloud. These mechanisms can track the flow of information in many ways. DIFC allows applications to define its own in-dependent terminology for security. In today's world the applications are built to serve different types of devices and based on one centralize database. To serve these devices application build different component and there is possibility of losing information through these component boundaries.

The main concern is the confidentiality and integrity of data. The scripts often revel more information to the private world than it should. Our proposed method addresses this problem which is caused by insecure information flow control model.

B. System Design

The main objective of the application is to pro-vide to extension to the existing RDBMS model so that the application cannot breach through Database. WE have to use the model of Query by label so that application component should not disclose more information than necessary.

The existing work has been done in the field of Information flow control model uses centralize platform. Our model is basically de-pends on IFDB [1] which is based on decentralized approach of information flow control. Security server authenticates the entity based on the query requested by the process. The existing work has some limitations based on the usability and functionality. The first one is Authority Server is introduced on different server, which overheads each authentication. This needs more execution time and message exchange. The second is related to compound tags. It supports compound tags but at static level and we need to define this earlier in our work. This can be overcome and we can create compound tags based on user requirement later in project development.

Query by Label: The proposed method makes use of Query by label model [1] for information flow control. Each label is the combination of tags representing secrecy and integrity tags. We are using the tuple level labelling method so each table is being introduced with label columns. These columns will be hidden to avoid confusion for program developers.

C. Assumption and Entities

The entities involved are DBA, end users and CSP. The CSP entity is just use as storage. This works on SAAS, one of the service model of cloud computing. The DBA task will be to manage complete IFM. This will also include the communication with authority server. End users are the entities who will use the data through

IV. IMPLEMENTATION DETAILS

D. System Overview

We are proposing to implement the project under JAVA and MySQL environment. MySQL will be uses for main database. The detail Plan is explained below.

Module 1: Dataset Gathering, Framework building, and UI: In this module we have to gather the information which needs to store in database and needs to build the framework which can have proper connection with UI and can connect with RDBMS.

Module 2: Decentralize Database storage: In this module we need to perform decentralized model of database and need to build web service clients which can communicate to database for user request on behalf of application.

Module 3: Query by Label: In this module, actually query by label model will be implemented and each SQL and database constructs will be labelled by the tags and label. We have to block the loopholes of information leaks through multiple channels and individual communication should be performed by individual client.

Module 4: Analysis Testing: The final module which will display the graphical results for analysis purpose and testing of the application

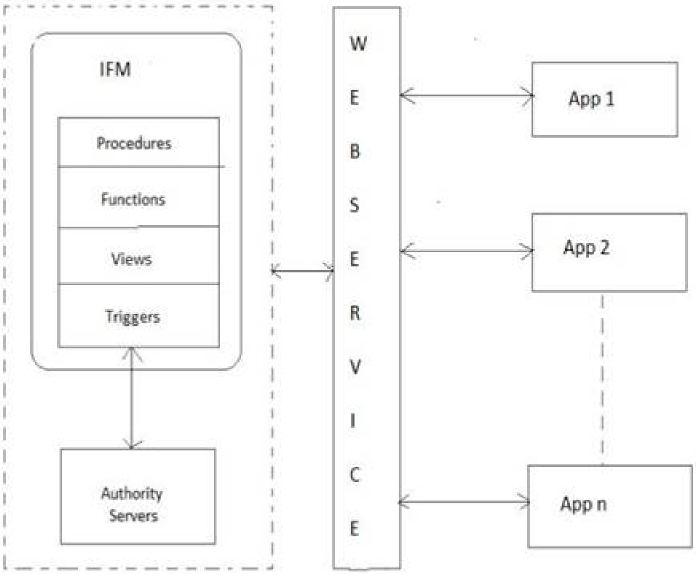

Fig.1: System Architecture

E. **Mathematical Model**

Tags
Every data has a tag assigned to it which has a secrecy and integrity restrictions. For example, for every medical record of Alice a tag Alice-medical is assigned and for every financial record a tag Alice-finance is assigned.

Label
Each label can be a set of either no tag or any number of tags. Every object of data and every process in the system have a secrecy and integrity label i.e.LS and LI. The data flows from X to Y only if the secrecy tags of X are a subset of the secrecy tags of Y, and the integrity tags of Y are a subset of the integrity tags of X.
Label L is represented as L = {o1: r1, r2; o2:r2, r3} where o1, o2, r1, r2 are principals.

Information Flow Rule
Data can flow from X to Y only if X.LS ⊆ Y.LS and X.LI ⊇ Y.LI.

Read Rule
Tuples which are with the label of LTj are selected by query of label LP if LTj⊆ L. for instance Alice’s records only be retrieved by process if its label contaminated for Alice. Any label with another tag or empty label is not received by process.

Write Rule
If a process has a label LP then it can write a tuple of label LT if LT ⊇ LP.

Integrity condition
Thread T can read object O only if O. label ⊆ T. label

Secrecy Condition
Thread T can write object O only if T. label ⊆ O. label

DOI: 10.15680/IJIRCCE.2016. 0406097

Communication Rule

The process p can send to data to the processq iffSp⊆Sq If X.LS ⊂ Y.LS and X.LI ⊃ Y.LI, then a Labels pair (X.LS, X.LI) is less restrictive than another label pair (Y.LS, Y.LI), i.e. we can say that information can flow from X to Y safely. X's label pair is no more restrictive than Y's label pair. (X.LS, X.LI) ≤ (Y.LS, Y.LI)

Indexing

Generating the index by merging of a columns, after that you can creating an index on a combination of columns, then you have to list the column names within the parentheses, separated bycommas:

Query - CREATE INDEX Index name ON table name (ColumnName1, Column Name2)

F. Experimental Setup

The system is built using Java framework (version jdk 1.8) on Windows platform, we also use MySQL 5.0 and Sqlyog 5.0 for database.NetBeans (8.0.2) is use as development tool. The system doesn't require any specific hard-ware to run, any standard machine is capable of running the application.

V. Result and Analysis

To improve the search result for input query we proposed indexing of database. A database index is an data structure that enhances the speed of data retrieval operations on a database table at the expense of extra composes and storage space to keep up the index data structure. Indexes are utilized to rapidly find data without searching each row in a database table each time a database table is accessed. Indexes can be made utilizing one or more column of a database table, giving the premise to both quick random lookups and efficient access of requested records. Also to improve the performance of system we used distributed cloud architecture, the use of cloud allows us to access the data anywhere, and anytime which enhance the availability of data or resource.

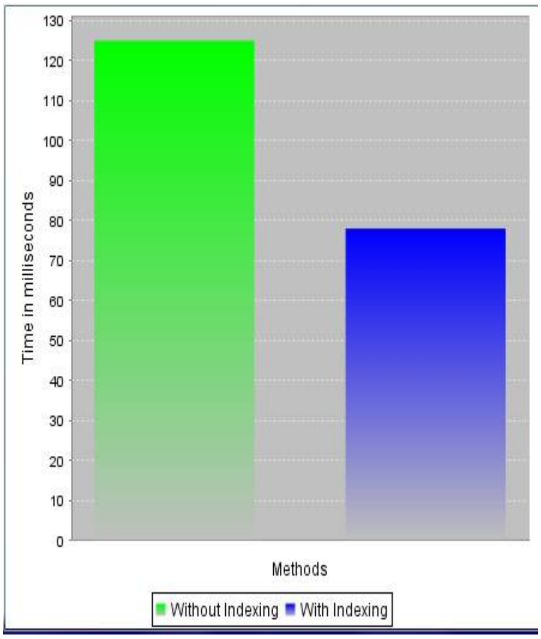

Fig.2: Time Comparison Graph.

DOI: 10.15680/IJIRCCE.2016. 0406097

VI. Conclusion and Future Work

We have studied the nature of information flow control through the various web application and we have introduced secure channel of data flow in applications. We have also explored that indexing on database by which the response time of search query is increased.In the future we are expecting to look into implementation of more SQL constraints like sequence. We will also try to work into the area of placing Authority Server into IFDB.

References

1. D. Schultz and B. Liskov, "IFDB: Decentralized Informa-tion Flow Control for Databases," in 8th ACM European Conference on Computer Systems (Eurosys). ACM, 2013, pp. 43-56.
2. A. Myers and B. Liskov. A decentralized model for in-formation flow control. In Proc. SOSP, Saint-Malo, France, 1997. ACM.
3. D. E. Denning, "A lattice model of secure information flow," Commun. ACM, vol. 19, no.5, pp. 236-243, May 1976.
4. G. E. Suh, J. W. Lee, et al., "Secure program execution via dynamic information flow tracking," ACM SIGOPS OSR, vol. 38, no. 5, pp. 85-96, 2004
5. P. Efstathopoulos, M. Krohn, et al., "Labels and event processes in the Asbestos operating system," in Proc. 2005 ACM SOSP, pp. 17-30.
6. N. Zeldovich, S. Boyd-Wickizer, and D. Mazieres, "Securing distributed systems with information flow control," in 2008 USENIXNSDI, pp. 293-308. 94
7. W. Cheng, D. R. K. Ports, et al., "Abstractions for usable information flow control in Aeolus," in 2012 USENIX ATC. -95
8. M. Krohn, A. Yip, et al., "Information flow control for standard OS abstractions," in 2007 ACM SOSP
9. P. Efstathopoulos, M. Krohn, et al., "Labels and event processes in the Asbestos operating system," in Proc. 2005 ACM SOSP, pp. 1730. 49
10. N. Zeldovich, S. Boyd Wickizer, E. Kohler, and D. Mazires.Making information flow explicit in HiStar. In Proc. OSDI,Berkeley, CA, 2006. USENIX.
11. M. Migliavacca, I. Papagiannis, et al., "Information flow guarantees in multi-domain distributed applications," in 2010 ACM/IFIP Middleware. 98
12. I. Roy, D. E. Porter, M. D. Bond, K. S. McKinley, and E. Witchel, "Laminar: practical fine-grained decentralized information flow control," SIGPLAN Not., vol. 44, no. 6, pp. 63-74, 2009. 96
13. I. Roy, S. T. V. Setty, A. Kilzer, V. Shmatikov, and E.Witchel, "Airavat: security and privacy for MapReduce," in 2010 Usenix NSDI. 97
14. P. Hosek, M. Migliavacca, et al., "SafeWeb: a middle-ware for securing Ruby-based web applications," in 2011 ACM/IFIP Middleware. 59