An Empirical Framework To Detect Security Attacks On the Cloud Data Storage System

Mrs Sowmya R [1] , Mr Sundeep Kumar K[2] , Dr Jitendranath Mungara[3]

[1] CMRIT Department of CSE, Visvesvaraya Technological University Belgaum, Karnataka-560037, India

[2] CMRIT Department of CSE, Visvesarayya Technological University Belgaum, Karnataka-560037, India

[3] CMRIT Department of CSE, Visvesvaraya Technological University Belgaum, Karnataka-560037, India

Abstract— **Providing an adequate security level in Cloud Environments is currently a particularly active analysis space. a lot of specifically, malicious behaviors targeting massive scale Cloud knowledge repositories (e.g. Denial of Service attacks) could drastically degrade the general performance of such systems and can't be detected by typical authentication mechanisms. during this paper we have a tendency to propose a generic security management framework permitting suppliers of Cloud knowledge management systems to outline and enforce complicated security policies. This security framework is meant to detect and stop an oversized array of attacks outlined through an expressive policy description language and to be simply interfaced with numerous knowledge management systems. we have a tendency to show that we will efficiently defend storage system by evaluating our security framework on high of the advanced data management platform.**

Keywords-component; Cloud Computing, Security, Cloud Storage Service,Denial Of Service Flodding attacks,

I. INTRODUCTION

As Cloud computing [1] is rising as an honest means to leverage accessible remote resources in a very versatile, scalable and value-effective means because of a usage-based cost model, one amongst the important issues that directly impacts the adoption rate of the Cloud paradigm is security [2]. This currently motivates an oversized variety of analysis efforts and collaborative comes on this subject. Despite the fact that Cloud computing could be a comparatively new field, some security mechanisms are already in place, most of that are imported from the Grid computing space. However, merely applying Grid techniques to Clouds might not be enough, as Clouds introduce new assumptions and requirements: Cloud environments suppose virtualization and isolation of resources, that introduce a necessity for a distinct approach, allow us to think about the of case the Nimbus Cloud-Kit [3], that inherited the Grid Security Infrastructure (GSI) [4], widely utilized in Grids to make sure message integrity and authentication of the communicating entities. During this case, once mutual authentication is performed, a possible threat is that authenticated purchasers might behave in a very malicious means, making an attempt to break the system, consume bandwidth or decrease its overall performance through operations that they need the acceptable access rights. The main target of our analysis is that the detection of such malicious purchasers that will be performing attacks [5] like Denial of Service (DoS) attacks, flooding attacks or crawling that can't be prevented by typical security mechanisms. Addressing such security vulnerabilities proves to be non-trivial. so as to attenuate management prices and increase potency, Cloud suppliers may benefit from generic security management systems that meet 2 essential requirements: (1) they'll be interfaced with any of the assorted Cloud systems that exhibit this kind of security vulnerabilities and (2) they'll handle and detect not solely predefined attacks, however conjointly those resembling customized security policies. This paper proposes such a generic security management framework, targeted at Cloud knowledge storage systems, that permits suppliers of Cloud knowledge management systems to outline and enforce complicated security policies. The genericity of this approach comes from its flexibility: it supports custom security eventualities and may be applied to completely different Cloud storage systems.

Addressing such security vulnerabilities proves to be non-trivial. so as to attenuate management prices and increase potency, Cloud suppliers may benefit

from generic security management systems that meet 2 essential requirements: (1) they'll be interfaced with any of the assorted Cloud systems that exhibit this kind of security vulnerabilities and (2) they'll handle and detect not solely predefined attacks, however conjointly those resembling customized security policies. This paper proposes such a generic security management framework, targeted at Cloud knowledge storage systems, that permits suppliers of Cloud knowledge management systems to outline and enforce complicated security policies. The genericity of this approach comes from its flexibility: it supports custom security eventualities and may be applied to completely different Cloud storage systems. We aim to provide high-level security mechanisms for Cloud storage services, as data access operations are vulnerable against a wide range of security attacks prone to damage the system and to affect its overall data access performance and response time. This paper focuses on the policy management core. In order to have an adequate malicious client detection level, we first have to define what kind of behavior is considered inappropriate or dangerous for the system. In section 2 we give an overview of related work which identifies all the major research work being done in this area. Section 3 highlights about the proposed system. Implementation and results are discussed in Section 4 and finally in section 5 we make some concluding remarks.

II. RELATED WORK

Whereas resource management in Grid environments is enforced by system directors, matters are completely different within the context of Clouds [6] [7], where users have the management of the remote virtual resources. This raises some extra security issues regarding management policies, as purchasers need to consider the safety tools of the Cloud service suppliers. to require the instance of Nimbus [3] once more, GSI mechanisms are used to authenticate and authorize shopper requests, VM image files, resource requests, reservation and usage times for users. This approach permits for easy cluster management, identity assignment, policies enforcement, setting reservation limits and path checks. Moreover, in [8], the authors extend this mechanism by encrypting the VM pictures on the shopper aspect, permitting the user to retain information management. However, the proposed remedy is merely appropriate for the storage of VMs, as their transfer is secured through GSI and also the start-up depends on the not-always true assumption that concerned systems are often trusted. a lot of security mechanisms (e.g., intrusion detectors) are required to guard the virtual host from attacks. From a a lot of general perspective, there's a desire to detect differing types of malicious behavior through custom policy enforcement mechanisms. Hadoop Distributed File System (HDFS) [9], the default back finish for the Hadoop Map/Reduce framework [10], implements security as a rudimentary file and directory permission mechanism. regarding authorization, the permission model is analogous to different platforms like Linux, every file and directory being related to an owner and a gaggle. HDFS uses Kerberos [11] because the underlying authentication system. In distinction to Nimbus, that depends on the powerful options of GSI, the most security threats in HDFS arise from the shortage of user-to-service authentication, service-to-service authentication and also the lack of encryption when sending and storing information. Moreover, even though a typical user doesn't have full access to the file system, HDFS is liable to varied attacks that it cannot detect, like Denial of Service. In Amazon easy Storage Service (S3) [12], the info storage and management infrastructure for Amazon's Elastic Compute Cloud [13], the users will decide how, when and to whom the data stored in Amazon internet Services is accessible. Amazon S3 API provides access management lists (ACLs) for write and delete permissions on each objects and objects containers, denoted buckets. However, no high-level security mechanism is accessible to guard the setting from complicated attacks, like those that can't be prevented by authentication mechanisms. Whereas all the comes described on top of rely heavily on authentication and authorization mechanisms, none of them is ready to spot users who arrange to damage the system or to detect specific patterns of malicious behavior. we tend to address exactly this goal: we tend to propose a generic policy management system to guard Cloud services from complicated attacks which will otherwise stay undetected and have an effect on the performance perceived by the purchasers.

III. PROPOSED SYSTEM

We outlined a hierarchical format for the protection policies, therefore on befits the on top of necessities. On one hand, every policy contains a collection of template user actions that form up a pattern reminiscent of a selected security attack. additionally, the policy will specify a collection of thresholds that draw the bounds between traditional

behavior that exhibits a similar activity pattern and malicious user actions. so as for an attack to be detected, the policy should be instantiated for a selected user, that is, the activity history of that user should embody recorded actions that match the template sequence provided by the policy. As an example, a DoS attack may be outlined by a series of write operations that happen in an exceedingly short amount of your time and are initiated by a similar shopper. Therefore, the corresponding policy can describe a write operation because the required pattern and can specify a period and therefore the most variety of write operations thought-about traditional for that period. In this project we have able to identify users who attempt to harm the system or to detect specific patterns of malicious behavior. We propose a generic policy management system to protect Cloud services from complex attacks that may otherwise remain undetected and affect the overall performance perceived by the clients. The proposed system is a modeling of complex and robust architecture ensuring security of data management system in cloud services over internet. The modules will consist of:

- Cloud Controller: It will enforce the security policy towards all the active online clients in the cloud network. It will be facilitated to manage (create/delete/modify) all clients. The cloud resource and service security is ensured by implementation of IP specification in the entire client's machine, which will force the client to access only from specific system with authorized IP and MAC.
- Clients: It is a set of different customers who are assumed to be utilizing the services offered by the cloud service providers. After passing through strict procedure of authentication from cloud controller, the clients will be authorized to access their privilege services. The clients will be entitled for uploading, modifying, deleting, and accessing data. All the clients are under strict vigilance of cloud controller.
- Virtual machine: It is a completely isolated guest operating system installation within a normal host operating system. A virtual machine (VM) is a software implementation of a machine (i.e. a computer) that executes programs like a physical machine. It will act as a bridge between clients and model of cloud infrastructure in the proposed system (see system architecture in Figure 1).
- Cloud infrastructure: It is set of dispersed data centres which are networked with their respective web servers and MySQL server. The entire clients request ends in this model. The model is also integrated with proposed security framework which will manage (enable / disable) security policy written by cloud controller.
- Malicious Process (Client-Side): Three types of malicious process will be designed to operate in client's machine as:
- Flooding: Multiple duplication of requests during file upload / modification/ file accessibility.
- Crawling: While deleting it will crawl through the author blob of replicated data.
- Threshold check: It checks thread handling capacity by the data centres in order to check DoS attack.

The main security concern is that the system must be robust enough to protect the data being accessed by unauthorized users. An attacker can impersonate an authorized user by stealing its credentials and then attempt to read all stored files (crawling). The performance analysis of the proposed system will be performed by measuring parameter matrices as:

- Average throughput Vs Time with different combination of malicious clients writing.
- Average throughput Vs. Client for:
- Legitimate Client
- Malicious or Illegitimate client
- Malicious Client Vs. Time

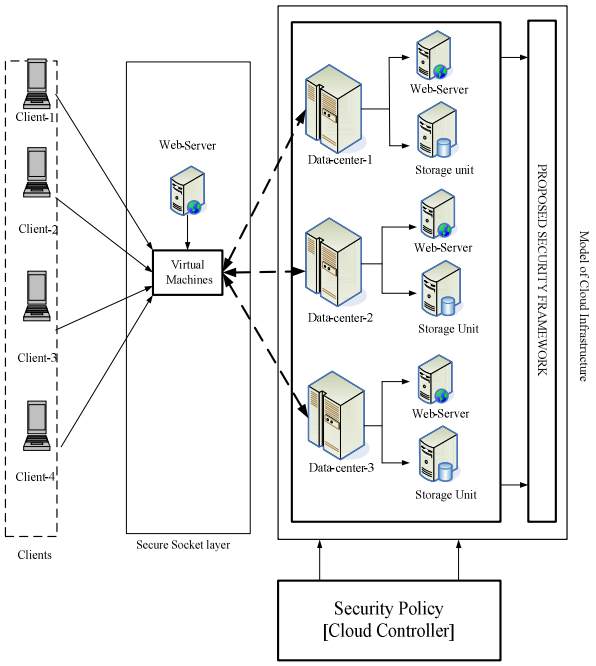

Figure 1: Proposed System Architecture

The tree structure of a security policy, which consists of four elements:

- The template set of user actions. The Preconditions element encloses the list of user actions that describe the pattern of an attack. Each user action is modeled by an Event, described through a set of attributes that identify a particular type of records in the User Activity History. To take the example of the DoS attack again, the Preconditions may contain only one event, whose Type attribute points to the list of recorded write operations in the User Activity History.
- General Parameters. General parameters will detect the clients whether they are active or which client request should get priority. Each client action will be detected by an event. Start event will detect which client has started the action (sending request)and which client has not started.
- The actions will be suggested by the security framework. These actions will be enforced based on certain constraints. If the clients sends series of request packets to the cloud for data storage. Trying to occupy more space (bandwidth) than the allocated space that time then security policy will enforce certain action based in the constraints. These constraints will define the type of action to be enforced such as blocking or monitoring ,restricting it to the limited storage.

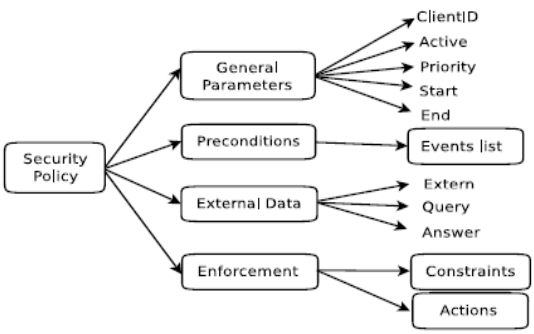

Figure 2: A high level representation of Security policy

Algorithm to Detect DDOS attack:

Step 1: Legitimate IP address of client machine is stored into an IP Address Database (IAD)

Step 2: Several statistics of incoming traffic for the current time interval N is calculated.

Step 3: A hash table is used to record the IP addresses that appeared in the current interval of time. Hash table entry contains 2 fields, the number IP packets and the time stamp of the most recent packet for that IP address.

Step 4: By comparing the current counts of the hash table with the IAD, we calculate how many new IP addresses have appeared in this time slot.

Step 5: If the number of packets per IP address is larger than a certain threshold, an alarm is set to indicate the bandwidth attack.

Step 6: By analyzing the number of new IP addresses, DOS attack can be detected.

Step 7: If an attack is detected, that particular IP address is suspended.

Detection Feature:

The key aspect of our detection scheme is that we choose a completely new detection feature compared to earlier detection proposals. We collect the IP ad-dresses during each time slot Δn (n=1, 2, 3, ...), which determines the detection resolution. We assume Δ1 =Δ2 = ... = Δn, which means the time slots are of equal length. The choice of Δn is a compromise between making Δn small so that the detection engine can quickly detect an attack, and making Δn large so

that the detection engine has less computation load because it checks the traffic less often. Let Tn represent the set of unique IP addresses and Dn represent the items of IP Address Database (IAD) at the end of the time interval Δn (n = 1, 2, 3, ...). As we discussed before, |Tn − Tn ∩ Dn| ,which represents the number of new IP addresses in Δn, can be used to detect the DDoS attack. However, |Tn−Tn∩Dn| varies according to the position of the network traffic monitoring point (NTMP) 4 Tao Peng et al. and different Δn. We can normalize this value by defining Xn = |Tn−Tn\Dn| Tn which will not be affected by the NTMP and _n. Consequently, we use Xn for our detection mechanism.

IV. IMPLEMENTATION AND RESULT

The proposed system is experimented on client server experimental test bed considering 32 bit windows OS with 1.84 GHz processor. The user interface is designed on java along with Apache Tomcat as web server. To validate our approach we

needed to see how it performs in large scale Cloud environments

Fig 3 gives the implementation flow chart of project. The Cloud controller will enforce the policy. The cloud controller will be able to create, monitor, delete the clients. After the clients are authenticated, they will be authorized to use the privileges of cloud system. This authenticated user will try to behave maliciously in terms of flooding, denial of service attacks. Such clients try to occupy more bandwidth. Our framework is designed to detect such type of attacks and block those clients which are malicious. IP address database table needs to keep updating the timestamp of each packet and the number of packets for particular instant of time.

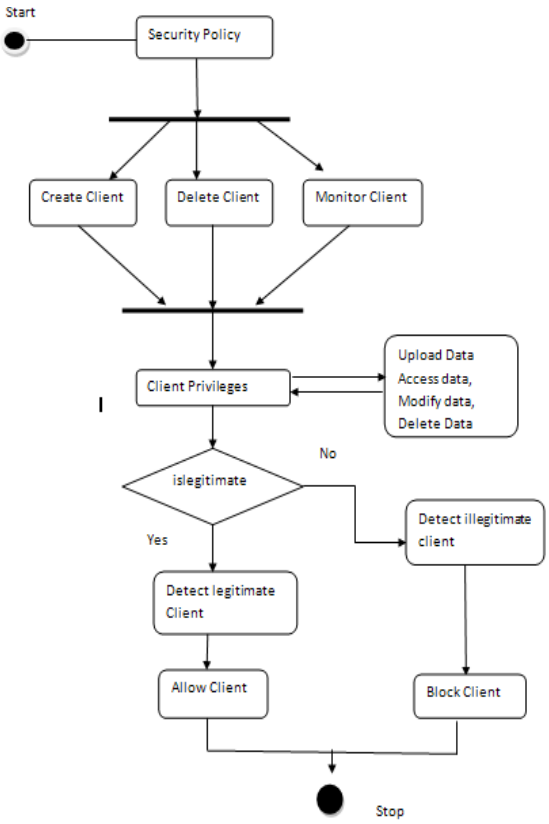

Figure 3. Implementation Plan Flow Chart

Data intensive applications can benefit from being executed in Cloud environments if the back-end storage services provide several important features, such as a scalable architecture, handling of massive unstructured data, and high throughput for data accesses or data-location transparency. We proposed a generic security management framework that enables Cloud storage providers to define and enforce flexible security policies. The Policy Management module we developed can be adapted to a wide range of Cloud systems, and can process any kind of policy that fits a given base format generated through the Policy Definition module. In this paper, we addressed a series of security issues, which expose important vulnerabilities of Cloud platforms, and, more specifically, of Cloud data management services.

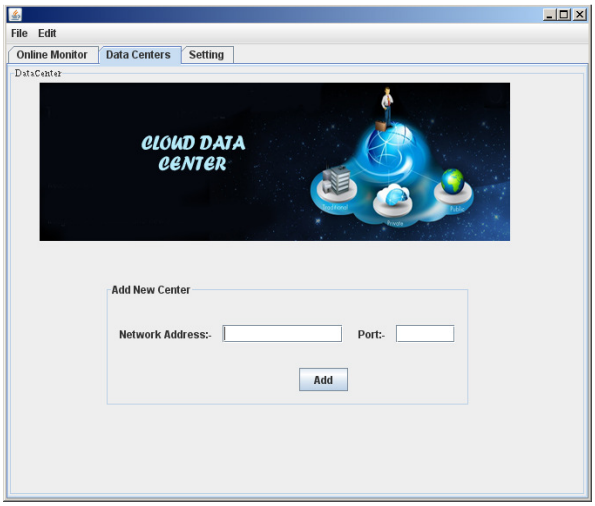

Figure 4. Main GUI of the application

The data center privilege is as shown in figure 4 above, where the user needs to add the Data Centers Add Network Address & Tomcat Port Number (Network Address means : IP address of the Data center pc IP Address. For An example: 192.168.1.101, Port: 8080). This is datacenter application with help of virtual machine application client id we should access this page.

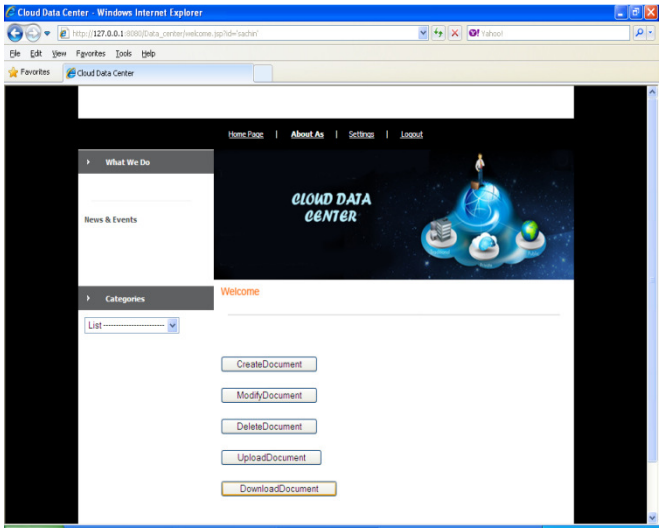

Figure 5. Sub-privileges of application

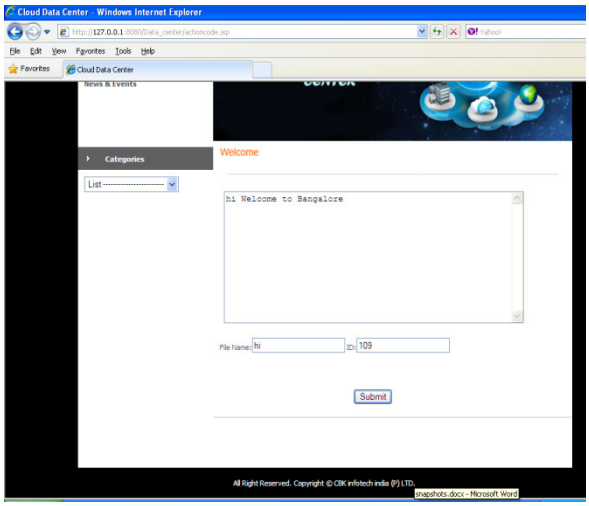

Figure 6. User interface for feeding file name and ID

.

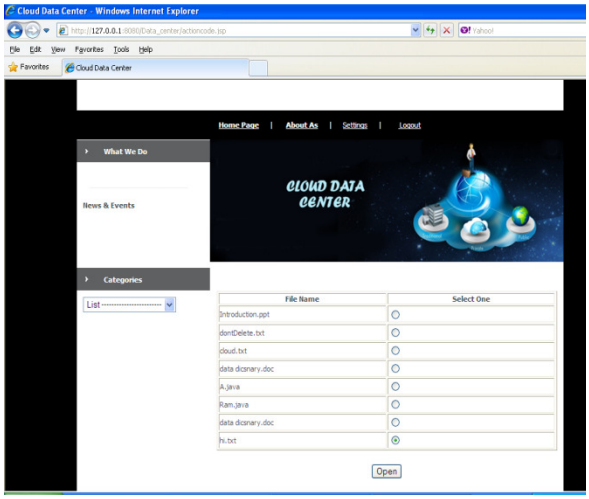

Figure 7. Visualization of digital contents in datacenter.

Figure 7 shows the availability of the digital contents in datacenters. Selecting any one it will display as below if someone want to modify the data. As shown as below; then submit the update button the data has modified

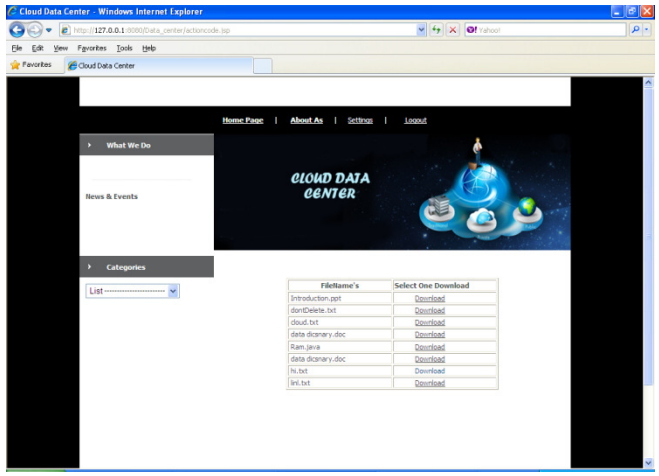

Figure 8. Digital content for downloading from datacenter.

Figure 8 shows the user interface if an user wants to download any one file it will display like as below, the files will be accessed in particular format.

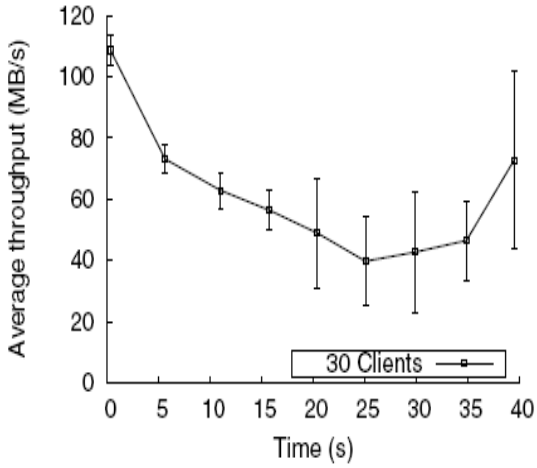

Figure 9: The evolution of the average throughput when 15 clients out of 30 perform malicious writes.

Fig 9 shows the performance (average throughput) of the cloud data center. The average throughput will decrease if the number of malicious clients increases. If the clients passes through security framework the malicious clients will be detected and it will be blocked, so that performance of the cloud data center will be retained.

V. CONCLUSION

Cloud computing has been emerged out with various benefits to an individual and corporate, and at the same time has exposed various concerns in the security aspects. In this project, the security aspects of the data management services have been considered. Various existing mechanism has been studied and analyzed. It has been found that most of the existing systems apart from data security focus on client authentication and authorization. There are always a certainty that authenticated clients can execute a process of flooding, crawling, and DoS because in cloud computing environment, the client having control of virtual machines. A framework is proposed where in local area network a cluster of computers will be formed as data centre and virtual machine to mimic actual cloud infrastructure. Authentication and authorization of an accessibility of client will be facilitated by resource ID and encryption will be handled by Secure Socket Layer (SSL), further malicious activities will be performed by the client process and then security policies will be written based on matrices such as client transaction trust system and by enforcing these policies malicious clients should be monitored and detected and further performance analysis will be done for resource utilization and Quality of Service. In this project will focus on more in-depth experiments involving the detection of various types of attacks in the same time. Moreover, we will investigate the limitations of our Security Management framework, with respect to the accuracy of the detection in the case of more complex policies, as well as the probability and the impact of obtaining false positive or false negative results. Another research direction is to further develop the Trust Management component of the security management framework and study the impact it has on the Policy Enforcement decisions for complex scenarios.

REFERENCES

[1] K. Keahey, R. Figueiredo, J. Fortes et al., "Science Clouds: Early experiences in cloud computing for scientific applications," In Cloud Computing and Its Application 2008 (CCA -08) Chicago, 2008.

[2] L. Vaquero, L. Rodero-Merino, J. Caceres et al., "A break in the clouds: towards a cloud definition," SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, pp. 50–55, 2009.

[3] K. Keahey, M. Tsugawa, A. Matsunaga, and J. Fortes, "Sky computing," IEEE Internet Computing, vol. 13, no. 5, pp. 43–51, 2009.

[4] V. Welch, F. Siebenlist, I. Foster et al., "Security for grid services," HPDC-12, p. 48, 2003.

[5] M. Jensen, J. Schwenk, N. Gruschka et al., "On technical security issues in cloud computing," in CLOUD '09, 2009, pp. 109–116.

[6] B. Nicolae, G. Antoniu, L. Bouge et al., "BlobSeer: Next generation data management for large scale infrastructures," J. Parallel Distrib. Comput., Aug 2010.

[7] B. Sotomayor, R. S.Montero, I. M. Llorente et al., "Virtual infrastructure management in private and hybrid clouds," IEEE Internet Computing, pp. 13(5):14–22, 2009.

[8] M. Descher, P. Masser, T. Feilhauer et al., "Retaining data control to the client in infrastructure clouds," Intl. Conf. on Availability, Reliability and Security, pp. 9–16, 2009.

[9] "HDFS. the Hadoop distributed file system," http://hadoop.apache.org/common/docs/r0.20.1/hdfs_design.html.

[10] D. Borthakur, The Hadoop Distributed File System: Architecture and Design, The Apache Software Foundation, 2007.

[11] B. C. Neuman and T. Ts'o, "Kerberos: An authentication service for computer networks," IEEE Communications, vol. 32(9), pp. 33–38, September 1994.

[12] Amazon Simple Storage Service (S3). http://aws.amazon.com/s3/.

[13] Amazon Elastic Compute Cloud (EC2), http://aws.amazon.com/ec2/.

[14] Managing data access on clouds: A generic Framework for enforcing security policies. Cristina B˘asescu, Alexandra Carpen-Amarie, et.al. 2011 International Conference on Advanced Information Networking and Applications

[15] Detecting Distributed Denial of Service Attacks Using Source IP Address Monitoring , Tao Peng Christopher Leckie Kotagiri Ramamohanarao

Mrs Sowmya R is presently doing Master of Technology in computer networks and engineering in CMR Institute of Technology, Bangalore Karnataka. She obtained her Bachelor of Engineering degree in computer science and engineering from University BDT college of engineering Davangere, Karnataka, India in the year 2003.

K Sundeep Kumar received the M.Tech (IT) from Punjabi University in 2003, ME (CSE) from Anna University in 2009 and pursuing Ph. D (CSE) from JNTUA. He is with the department of Computer Science & Engineering and as an Associate Professor, CMR Institute of Technology, Bangalore. He presented more than 10 papers in International and national Conferences. His research interests include OOMD, Software Engineering and Data Warehousing. He is a life member in ISTE.

Dr.M.Jitendranath is double doctorate in Electronics and Computer Science Engineering and working as Professor & Dean of Research in Computer Science Engineering department in CMRIT, Bangalore. He has published 35 papers in the area of Mobile ad hoc Networks international journals