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

International Journal of Innovative Research in Computer and Communication Engineering
*Vol. 1, Issue 1, March 2013*

# Privacy Preserving & Access Control to Intrusion Detection in Cloud System

Swati P. Ramteke[1], Priya S. Karemore[2], S. S. Golait[3]

Student, Priyadarshini college of Engineering, Nagpur, India[1]
Assistant Professor, Priyadarshini college of Engineering, Nagpur, India[2]
Assistant Professor, Priyadarshini college of Engineering, Nagpur, India[3]

**Abstract**: As Cloud Computing becomes common, more and more responsive information are being centralized into the cloud. For the protection of data privacy, sensitive data usually have to be encrypted before outsourcing, which makes effectual data utilization, a very challenging task. We propose a new model for data storage and access in clouds. Our scheme avoids storing multiple encrypted copies of same data. In our framework for secure data storage, cloud stores encrypted data (without being able to decrypt them). The main innovation of our model is addition of key distribution centers (KDCs). In this paper, we propose a solution which removes the trusted central authority, and protects the users privacy by preventing the authorities from pooling their information on particular users, thus making ABE more usable in practice. To handle large scale network access traffic and administrative control of data and application in cloud, a new multi-threaded distributed cloud IDS model has been proposed. Our proposed cloud IDS handles large flow of data packets, analyze them and generate reports efficiently by integrating knowledge and behavior analysis to detect intrusions.

**Keywords**: Access control, attribute based encryption, privacy preserving, Storage in Clouds.

## I. INTRODUCTION

We often identify people by their attributes. In 2005, Sahai and Waters proposed a system (described in more recent terminology as a key-policy attribute-based encryption (ABE) system for threshold policies) in which a sender can encrypt a message specifying an attribute set and a numbered, such that only a recipient with at least d of the given attributes can decrypt the message. However, the deployment implications of their scheme may not be entirely realistic, in that it assumes the existence of a single trusted party who monitors all attributes and issues all decryption keys. Instead, we often have different entities responsible for monitoring different attributes of a person, e.g. the Department of Motor Vehicles tests whether you can drive, a university can certify that you are a student, etc. Thus, Chase gave a multi-authority ABE scheme which supports many different authorities operating simultaneously, each handing out secret keys for a different set of attributes. However, this solution was still not ideal. There are two main problems: one concern of security of the encryption, the other the privacy of the users.

## II. CLOUD COMPUTING

Cloud computing is internet based computing where virtual shared servers provide software, infrastructure, platform, devices and other resources and hosting to customer as a service on pay-as you-use basis. All the info that a digitized system has to offer is provided as a service in the cloud computing model. Users can access these services available on the “internet cloud” without having any previous know-how on managing the resources involved. Cloud users do not own the physical infrastructure; rather they rent the usage from a third- party provider. They consume resources as a service and pay only for resources that they use. What they only need is a personal computer and internet connection. Cloud computing has revolutionized the IT world with its services provisioning infrastructure, less maintenance cost, data & services availability assurance, rapid accessibility and scalability. Cloud computing has three basic abstraction layers i.e. system layer (which is a virtual machine abstraction of a server), the platform layer (a virtualized operating system of a server) and application layer (that includes web applications) . Hardware layer is not included as it does not directly offer to users.
Cloud computing also has three service models namely Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Software as a Service (SaaS) models. PaaS model facilitates users by providing platform on which applications can be developed and run. IaaS deliver services to users by maintaining large infrastructures like hosting servers, managing networks and other resources for clients. SaaS model makes user worry free of installing and running software services

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

International Journal of Innovative Research in Computer and Communication Engineering
*Vol. 1, Issue 1, March 2013*

on its own machines. Presently, Salesforce.com, Google and Amazon are the leading cloud service providers who extend their services for storage, application and computation on pay as per use basis. Data, application and services non-availability can be imposed through Denial of Service (DOS) or Distributed Denial of Service (DDOS) attacks and both cloud service provider and users become handicap to provide or receive cloud services . For such type of attacks Intrusion Detection System (IDS) can be emplaced as a strong defensive mechanism. IDSs are host-based, network-based and distributed IDSs. Host based IDS (HIDS) monitors specific host machines, network-based IDS (NIDS) identifies intrusions on key network points and distributed IDS (DIDS) operates both on host as well as network. IDSs produce alerts for the administrators which are based on true positives or true alarms when actually intrusion takes place and false positive or false alarms in case of a wrong detection by the system. IDSs can detect intrusion patterns by critically inspecting the network packets, applying signatures (pre-defined rules) and generating alarms for system administrators. IDS uses two method of detection i.e. anomaly detection, that works on user behaviour patterns and suspicious behaviour. Other method is misuse detection that can detect through renowned attack patterns and matching a set of defined rules or attack against system vulnerabilities through port scanning .
Since Cloud infrastructure has enormous network traffic, the traditional IDSs are not efficient enough to handle such a large data flow. Most known IDSs are single threaded and due to rich dataset flow, there is a need of multi-threaded IDS in Cloud computing environment. In a traditional network, IDS monitors, detects and alert the administrative user for network traffic by deploying IDS on key network choke points on user site. But in Cloud network IDS has to be placed at Cloud server site and entirely administered and managed by the service provider. In this scenario, if an attacker manages to penetrate and damage or steal users data, the cloud user will not be notified directly. The intrusion data would only be communicated through the service provider and user has to rely on him. The cloud service provider may not like to inform the user about the loss and can hide the information for the sake of his image and repute. In such a case, a neutral third party monitoring service can ensure adequate monitoring and alerting for cloud user. In this report, we have proposed an efficient multi-threaded cloud IDS, administered and monitored by a third party ID monitoring service, who can provide alert reports to cloud user and expert advice for cloud service provider. In order to resolve the issues which traditional IDSs cannot resolve, an efficient and reliable distributed Cloud IDS model is proposed.

## III. ATTRIBUTE BASED ENCRYPTION

We now give an overview of Attribute-Based Encryption (ABE) algorithms. The Sahai-Waters (ABE) cryptosystem as implemented in this paper is specifically detailed. We focus our efforts on providing the description of the scheme and intuition for its construction. For the proof of security see Sahai and Waters . Attribute-Based Encryption can be viewed as a generalization of Identity-Based Encryption (IBE) . A party in the system can encrypt a message to this particular user with only the knowledge of the recipient's identity and the system's public parameters. In particular the encryption algorithm does not need to have access to a separate public key certificate of the recipient. In Attribute-Based Encryption a user's identity is composed of a set, S, of strings which serve as descriptive attributes of the user. For example, a user's identity could consist of attributes describing their university, department, and job function. A party in the system can then specify another set of attributes S0 such that a receiver can only decrypt a message if his identity S has at least k attributes in common with the set S0, where k is a parameter set by the system. Like traditional Identity-Based Encryption, a party in an Attribute-Based Encryption system only needs to know the receiver's description in order to determine their public key. However, the expressiveness of an ABE system is potentially much more powerful. For example, there could be several different recipients that are able to decrypt a message encrypted for a set S0. ABE-based systems can also leverage "threshold constructions" where a user with identity S will be able to decrypt a message if it has at least k attributes that overlap with a set S0 chosen by the encryptor. Although there could in theory exist even more expressive ABE systems, the threshold constructions described and illustrated in the following sections are sufficiently semantically deep that we can define complex and precise encryption policies.

### *A. ABE ALGORITHMS*

We now informally specify a threshold Attribute-Based Encryption system as a collection of four algorithms:

• **Setup(k):** The Setup algorithm is run by an authority in order to create a new ABE system. Setup takes as input a threshold value, k and outputs a master key MK and a set of public parameters PK.

• **Key-Gen(S,MK):** The authority executes the Key-Gen algorithm for the purpose of generating a new secret key SK. The algorithm takes as input the user’s identity, S, as a set of strings representing a user’s attributes and the master-key MK and outputs S’s secret key SK.

• **Encrypt(M, S0,PK):** The Encrypt algorithm is run by a user to encrypt a message M, with a target set S0, and the public parameters. It outputs a ciphertext, C.

• **Decrypt(C, S0, S, SK):** The Decrypt algorithm is run by a user with identity S and secret key SK to attempt to decrypt a ciphertext C that has been encrypted with S0. If the set overlap |S \ S0| is greater than or equal to k the algorithm will output the decrypted message M.

## IV. INTRUSION DETECTION SYSTEM

Intrusion Detection Systems help information systems prepare for, and deal with attacks. They accomplish this by collecting information from a variety of systems and network sources, and then analyzing the information for possible security problems. Intrusion detection provides the following:

- Monitoring and analysis of user and system activity
- Auditing of system configurations and vulnerabilities
- Assessing the integrity of critical system and data files
- Statistical analysis of activity patterns based on the matching to known attacks
- Abnormal activity analysis
- Operating system audit

There are three main components to the Intrusion detection system

- Network Intrusion Detection system (NIDS) – performs an analysis for a passing traffic on the entire subnet. Works in a promiscuous mode, and matches the traffic that is passed on the subnets to the library of knows attacks. Once the attack is identified, or abnormal behavior is sensed, the alert can be send to the administrator. Example of the NIDS would be installing it on the subnet where you firewalls are located in order to see if someone is trying to break into your firewall

- Network Node Intrusion detection system (NNIDS) – performs the analysis of the traffic that is passed from the network to a specific host. The difference between NIDS and NNIDS is that the traffic is monitored on the single host only and not for the entire subnet. The example of the NNIDS would be, installing it on a VPN device, to examine the traffic once it was decrypted. This way you can see if someone is trying to break into your VPN device

- Host Intrusion Detection System (HIDS) – takes a snap shot of your existing system files and matches it to the previous snap shot. If the critical system files were modified or deleted, the alert is sent to the administrator to investigate. The example of the HIDS can be seen on the mission critical machines, that are not expected to change their configuration

IDSs are software or hardware systems that automate the process of monitoring the events occurring in a computer system or network, analyzing them for signs of security problems. IDSs are one of widely used security technologies. An IDS alerts to system administrators, generate log about attack when it detects signature of accident according to host or network security policy. IDS can be installed in a host or a network according to purpose. Thus, the aim of the IDS is to alert or notify the system that some malicious activities have taken place and try to eliminate it. According to the method of the collection of intrusion data, all the intrusion detection systems can be classified into two types: host-based and network-based IDSs. Hostbased intrusion detection systems (HIDSs) analyze audit data collected by an operating system about the actions performed by users and applications; while network-based intrusion detection systems (NIDSs) analyse data collected from network packets. IDSs analyse one or more events gotten from the collected data. According to analysis techniques,

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

International Journal of Innovative Research in Computer and Communication Engineering
*Vol. 1, Issue 1, March 2013*

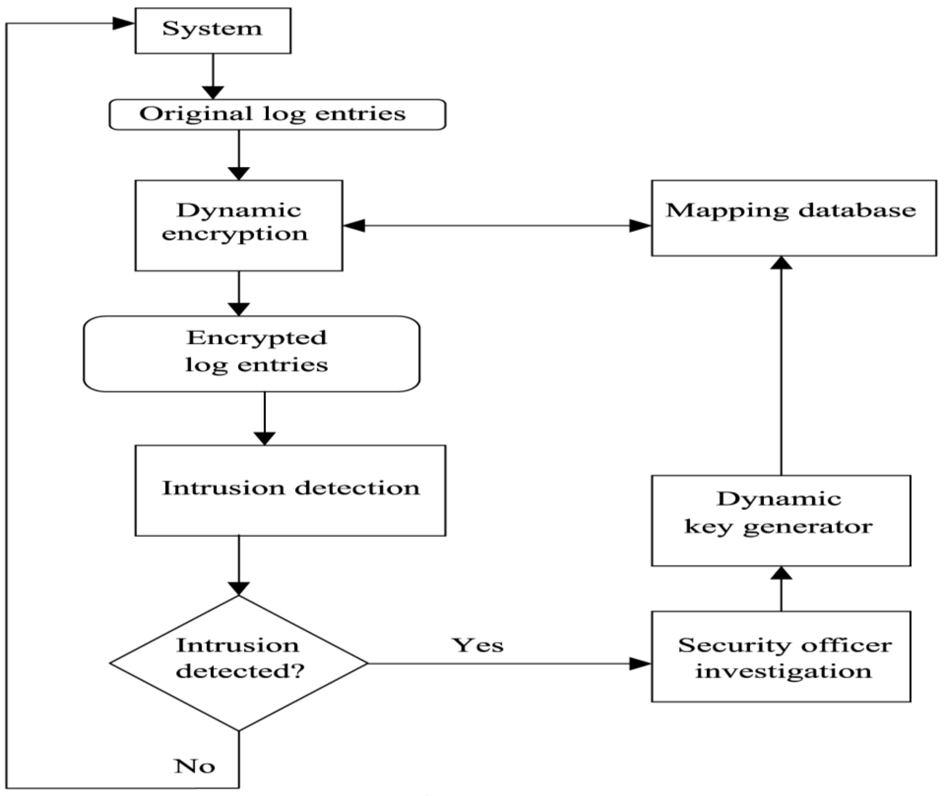

Fig. 1. Flowchart of Intrusion Detection System

IDS system is classified into two different parts: misuse detection and anomaly detection. Misuse detection systems use signature patterns of exited well-known attacks of the system to match and identify known intrusions. Misuse detection techniques, in general, are not effective against the latest attacks that have no matched rules or pattern yet. Anomaly detection systems identify those activities which deviate significantly from the established normal behaviors as anomalies. These anomalies are most likely regarded as
intrusions. Anomaly detection techniques can be effective against unknown or the latest attacks. However, anomaly detection systems tend to generate more false alarms than misuse detection systems because an anomaly may be a new normal behavior or an ordinary activity. While IDS detects an intrusion attempt, IDS should report to the system administrator. There are three ways to report the detection results. They are notification, manual response, and automatic response. In notification response system, IDS only generates reports and alerts. In manual response system, IDS provides additional capability for the system administrator to initiate a manual response. In automatic response system, IDS immediately respond to an intrusion through auto response system.

## V. CLOUD INTRUSION DETECTION SYSTEM SERVICE

Developing IDS mechanism in the Cloud environment is highly motivated for both Cloud users and Cloud providers. There are some factors which instruct user for this tendency. Apart from the normal features which are obtained when using a service provided by a Cloud such as fast access to best business applications, eliminating the rule of trained new personnel, or licensing new software, some aspect of IDS in Cloud would be tempting for this migration.Accessing to Network-based IDS on the Cloud to protect a network gives the possibility of exploiting different type of IDS detection methods on a single segment based on user demands almost instantly. It gives the illusion of a Network-based IDS which try to protect all segments of a network which are communicating over the internet with each other. It uses dedicated resources on the cloud for IDS functionalities which are isolated from any host on user network.

### *A. CLOUD INTRUSION DETECTION SERVICE ARCHITECTURE*

We introduce a Cloud Intrusion Detection System Service (CIDSS) to overcome the critical challenge of keeping the client secure from cyber attacks. It is designed based on software as a service model for security of any Cloud based user. The CIDSS is composed of three components: Intrusion Detection Service Agent, Cloud Computer Service

Component (CCSC), and Intrusion Detection Service Component (IDSC). Figure 2 represents each component in the proposed structure.

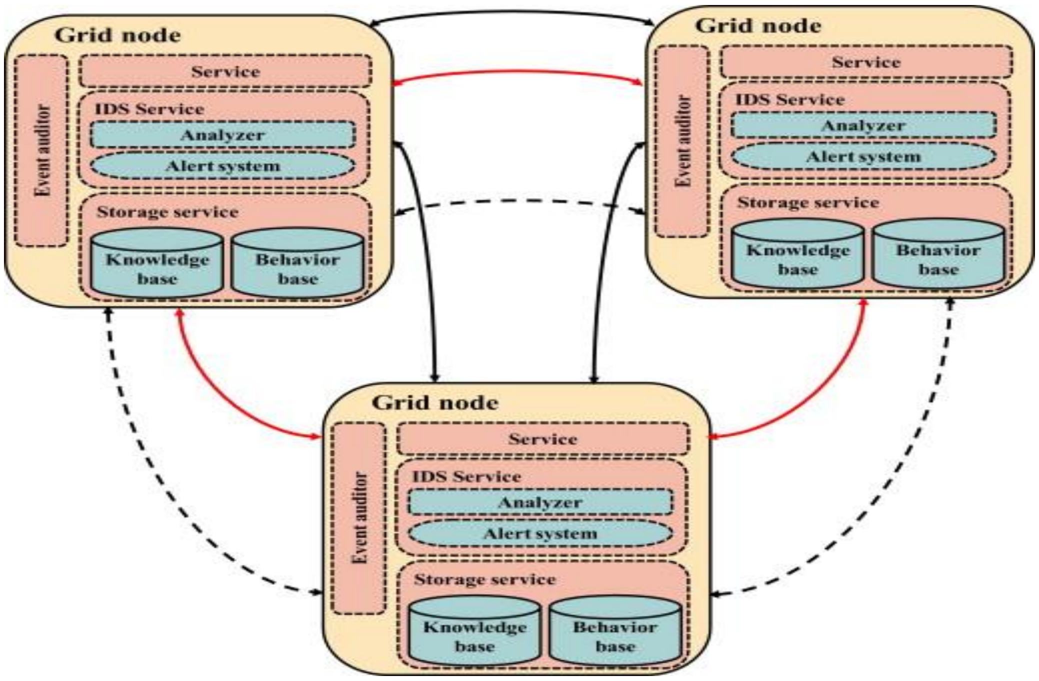

Fig.2. Cloud Intrusion Detection Architecture

• **Intrusion Detection Service Agent:** The agent is a light weight, single purpose equipment – dedicated hardware or software - integrated inside the user network to collect necessary information. According to the location of the agent, the CIDSS could protect a segment of the network or the whole network. Agents are grouped based on rule-sets and thresholds or network traffic to improve the service efficiency and protection flexibility.

• **Cloud Computer Service Component:** The CCSC collects messages from agents. It formats all messages and send them to the IDSC according to grouping constrains defined for messages. A secure connection path should be established by CCSC to absorb information gathered by agents otherwise the system behavior could be tainted by external intrusion.

• **Intrusion Detection Service Component:** The IDSC is responsible for intrusion detection. There are four subcomp-onents playing major rule in IDSC. *Collector:* Collector is responsible for reading all information received by CCSC, selecting items of interest, and forwarding them to the appropriate analysis engine.

*Analysis Engine:* Analysis engine is a sophisticated decision and pattern matching mechanism. It analysis data came from the collector and matches it to known patterns of activity stored in the signature database. It identifies malicious behavior and generates alerts through the event publisher.

*Event Publisher:* It is a standardized interface to provide a unified view of result report for users as the analysis engine could be an independent process which can be implemented using any IDS, e.g. Snort. Intrusion Detection Message Exchange Format (IDMEF) [14] would be used as standard representation of IDS alerts.

*IDS Controller:* It is responsible for remote configuration and control of all agent groups. It has access to IDSC configuration for fine tuning its operation based on user demands.Fig. 2 Cloud Intrusion Detection System Architecture.

### *B. CLOUD INTRUSION DETECTION SERVICE REQUIREMENTS*

There are a number of challenges that must be considered when implementing CIDSS. Some of these challenges are inherent in what an IDS does and others are simply part of the way that a network is configured. Normally, the construction of a network would include switches and routers. The collision domains which is defined as the extend to which a signal can be propagated inside a network would be separated by these devices . An agent needs to be able to look at all of the traffic on a protected network segment even on different collision domains. There are a number of ways to achieve this goal including: adding a hub inline at a choke point, connecting a network Test Access Port (TAP) to allow passive access to all the traffics, using Switched Port Analyzer (SPAN) port on the switch being monitored.High-speed networks are a problem for any IDS solutions. A normal IDS would do the processing of every packets on the network. The speed of the network could increase ahead of the processing power available to pull every packet off the wire and process it . An agent does not demand high resources as it would not scrutinize the network traffic and limits itself to rudimentary processing to capture packet of interest. Nevertheless, the only restriction on agent deployment is disparity between the traffic of network segment and the connection speed with the Cloud. CIDSS mostly targets the Small and Medium Businesses (SMB) as users and they do not typically have very high networking bandwidth as soon as they become available. Anyhow, agent grouping would provide the flexibility to cope with this issue. Three basic rules are applied when grouping the agents, listed in Table 1. According to the rules, the agent utilized to protect whole user network from internet attacked should be in a separate group while the agents deployed to protect two small segments of user network with similar rule-sets could be in one group if their traffics are negligible.

Table 1: Agent Grouping Basic Rules

| *Network Segment Configuration* | *Group* |
|---|---|
| Segments with different rule -set | Separate Group |
| Segments with similar rule-set | Similar Group |
| Segments with high traffics | Separate Group |

The security of agents is a preliminary factor while implementing and embedding one inside the user network. They are operating in stealth mode and a distinct connection form user network internet connection would be employed for communication with the cloud. Even though it is not considered as a requirement for agents operation but single purpose internet connection could enforce more security as the respective firewall could be configure more strictly for a single service.

### *C. CAPTURING INTERESTED PACKETS*

Agent should do the early filtering to reduce the load on the Analysis engine and on the overall system, as the process of sending packets from the agent to the CCSC can often be avoided. Considering an internet connection with n Mbps bandwidth, at most a channel with 2n Mbps (upstream plus downstream) bandwidth should be sniffed by an agent. For almost real time IDS, the captured packets should be discarded to 50% of total packets in the worst case. Otherwise, detection delay should be introduced. Some unique feature of network traffics can be exploited to this end. Refer to Figure 3 for internal structure of agent.

Since a fraction of packets is only subject to header analysis, this could be performed by agent and only packets prone to intrusion effects would be transferred . Packets with no payload which do not show any indication of intrusion are simply discarded

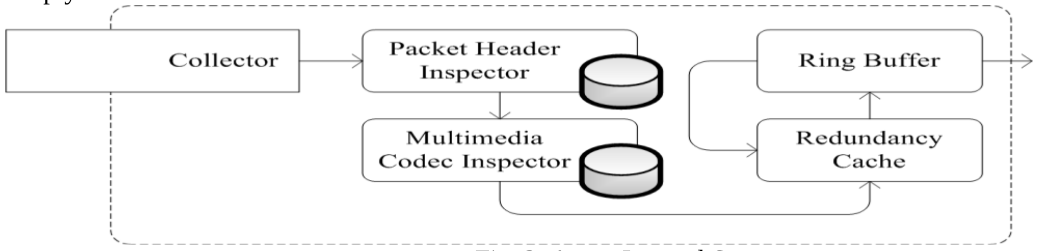

Fig. 3. Agent Internal Structure.

.

According to the researches 30% of all incoming traffic and 60% of all outgoing traffic is redundant by measures . Therefore, agent and the respective CCSC should cooperate to eliminate the redundant data in packets. Caches are place at both agent and CCSC, which are used to hold the most recent packets. The cache at agent end would replace the redundant data inside the packet into light weight tokens and passes the encoded, smaller packet to CCSC, where the original packet would be reconstructed. The technique developed by Manber for finding similar files in a huge file system utilizing Robin fingerprint would be adopted to implement a network traffic tunable duplication detection strategy . Packet losses should be prevented in order to preserve the required consistency between caches. Multimedia data which constitutes most of the traffic in the internet communications could be inspected by agent. Multimedia traffic can be uniquely identified by certain characteristics, e.g. the header in an AVI file, and once the traffic has been recognized the remaining packets for the stream containing the multimedia content can bypass the analysis engine. Ring buffer is a method for adapting the packet stream speed in a way that prevent packet loss . The latency introduced by ring buffers is not particularly important for a CIDSS operation where the latency does only affect detection delay and no packet forwarding happens as no prevention is intended by CIDSS. The ring buffer should tune the performance of redundancy cache. In the best case it would be disabled if the detection latency is negligible.

### *D. SECURE CONNECTION AND GROUP MANAGEMENT*

As the CIDSS intends to protect the user network from the cyber attacks, the agent should sniff all internet inbound and outbound traffics. A whole user network can be protected by assigning an agent group. Similarly, different network segments of a user network which are connected to each other using a VPN over the internet can be protected by assigning a single agent group for all segments. Therefore, independent to the network structure, the user network can be protected against cyber attacks using a central IDS in a cloud. Figure 4 demonstrates sample agent configurations.

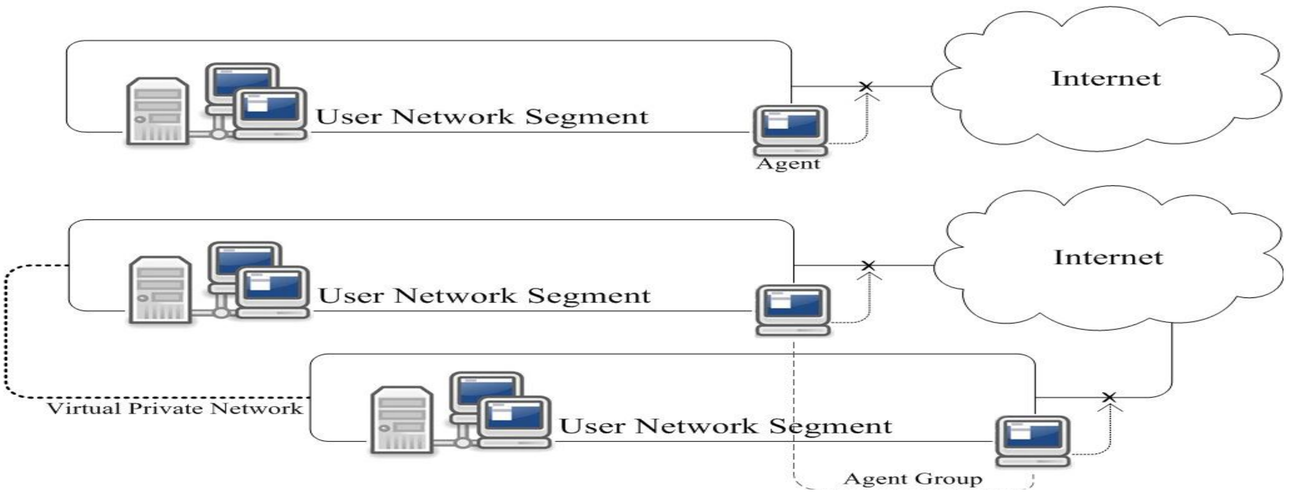

Fig. 4 Sample Agent Configuration in User Local Network.

All of the agents in a single agent group would communicate with a single instance of CCSC. Therefore, similar analysis engine an ultimately same rule-sets would be applied for all agents in an agent group.

### *E. IMPLEMENTATION*

The Proof-of-Concept is implemented as a first step in the direction of a mature CIDSS. All components are implemented in a local test bed to evaluate the concept of the proposed structure. The service would allocate two VMs for CCSC and IDSC components of the CIDSS. A CCSC would be allocated for each agent group. CCSCs would access a database containing all issued credentials while the service ordered by users. These credentials are used by CCSCs while the VPN is established among a agent group members and CCSC to check the validity of a user.
There is single IDSC VM for each user irrespective to the total number of agent groups. It would be equipped with a normal Network-based IDS installation which is working in promiscuous mode on a virtual NIC. All IDSC of all users would access a single database containing pre-configured ready-to-used rule-sets which can be requested by users when a service is requested or can be configured using the agent.

Three steps should be proved to be feasible in the practical situation:
• Capturing packets from the protecting network.
• Secure communication with remote node.
• Detection of Intrusion on the remote node.

Table 2: Virtual Network Interfaces Configurations

| | *NICs* | *NICs Information* | |
|---|---|---|---|
| **VM One** | **1** | IP (eth0) | 192.168.0.10 |
| | | Subnet | 255.255.255.0 |
| | | Network | 192.168.0.0 |
| **VM Two** | **3** | **(eth0) / (tap0) Bridge** | |
| | | IP (eth1) | 192.168.1.10 |
| | | Subnet | 255.255.255.0 |
| | | Network | 192.168.1.0 |
| **VM Three** | **3** | IP (eth0) | 192.168.1.11 |
| | | Subnet | 255.255.255.0 |
| | | Network | 192.168.1.0 |
| | | **(eth1) / (tap0) Bridge** | |
| **VM Four** | **1** | IP (eth0) | 192.168.2.11 |
| | | Subnet | 255.255.255.0 |
| | | Network | 192.168.2.0 |

The testing environment consists of four VMs. The specifications for each VMs is represented in Table 2.According to the table three networks would be deployed for this implementation 192.168.0.0, 192.168.1.0 and 192.168.2.0. ethX are VMware virtual network interfaces which are configured using static IP addresses in Linux operating system. tapX are the Linux Kernel virtual tap interface used by VPN software to interfere with normal multiple layers of ISO OSI model. Using tapX the VPN software could simulate the data-link and physical layer of the communication channel as a virtual medium over the internet. In the remaining part of this section the steps taken is listed:
1. Install a VMware workstation on a PC.
2. Create four VMs with Linux as operating systems.
3. Configure the virtual network using VMware Virtual Network Editor to comply with the information represented in Table 2.
4. Install OpenVPN on VM two and VM three and create the tapX interfaces.
5. Configure the OpenVPN client and server on VM two and VM three.
6. Activate bridging between eth0 and tap0 on VM two and between eth1 and tap0 on VM three.
7. Disable Linux kernel MAC table.
8. Install Snort on VM four.
9. Configure Snort to monitor traffic in promiscuous mode on eth0 on VM four.

Finally to test the proposed structure the attack should be initiated on VM one and the Snort should be capable of generating the intended attack results.

## VI. FUTURE ENHANCEMENTS

In the future, we‘ll implement our IDS, helping to improve green (energy-efficient), white (using wireless networks), and cognitive (using cognitive networks) cloud computing environments. We also intend to research and improve the security features in cloud computing environment.

## VII. CONCLUSION

Facing the complexity of Cloud architecture, this paper focuses on proposing deployment architecture of Intrusion Detection Systems in the Cloud. We discuss and list several existing threats for a Cloud infrastructure and are motivated to use Intrusion Detection Systems (IDS) and its management in the Cloud. We propose the deployment of integrated and layered IDS on cloud that designed to cover various attacks. This IDS integrates knowledge and behavior analysis to increases a cloud‘s security. The two intrusion detection techniques are distinct. But the deficiency of one technique will be complimented by other one. Layered IDS offers effective resource and log management.

## VIII. ACKNOWLEDGMENT

At the outset I whole heartedly thank the Almighty, who has been my strength in times of weakness and hope in time of despair, the sole creator of all the creations in this world and hence this paper. I thank my parents and guardian who have encouraged me with good spirit by their incessant prayers to complete this paper. I would like to express my sincere thanks to our department for providing me various formalities needed for successful completion of my project. It is my deep sense of gratitude and honor to acknowledge sincere thanks to my internal guides Prof. Priya S. Karemore & Prof. S. S. Golait, for his valuable directions, suggestions and exquisite guidance with enthusiastic encouragement ever since the commencement of the project.

## REFERENCES

[1] H. Debar, M. Dacier, and A. Wespi, ―Towards a Taxonomy of Intrusion Detection Systems,Int‘l J.Computer and Telecommunications Networking, vol. 31, no. 9, pp. 805–822, 1999.
[2] S. Axelsson, Research in Intrusion-Detection Systems: A Survey, tech. report TR-98-17, Dept. Computer Eng.,Chalmers Univ. of Technology, 1999.
[3] S. Kenny and B. Coghlan, ―Towards a Grid-Wide Intrusion Detection System, Proc. European Grid Conf. (EGC 05), Springer, pp. 275–284,2005.
[4] M. Tolba et al., ―Distributed Intrusion Detection System for Computational Grids,Proc. 2nd Int‘l Conf. Intelligent Computing and Information Systems (ICICIS 05), 2005.
[5] F-Y. Leu et al., ―Integrating Grid with Intrusion Detection,Proc. Int‘l Conf. Advanced Information Networking and Applications
[6] S. Roschke, F. Cheng, and C. Meinel, "Intrusion detection in the cloud," in 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing. IEEE, 2009, pp. 729–734.
[7] K. Vieira, A. Schulter, C. Westphall, and C. Westphall, "Intrusion detection for grid and cloud computing," It Professional, vol. 12, no. 4, pp. 38–43, 2010.
[8] C. Mazzariello, R. Bifulco, and R. Canonico, "Integrating a network ids into an open source cloud computing environment," 2010.
[9] V. Marinova-Boncheva, "A short survey of intrusion detection systems," Problems of Engineering Cybernetics and Robotics, vol. 58, pp. 23–30, 2007.
[10] R. Bace and P. Mell, Intrusion detection systems. US Dept. of Commerce, Technology Administration, National Institute of Standards and Technology, 2001.
[11] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," Computer Networks, vol. 31, no. 8, pp. 805–822, 1999.
[12] R. Buyya, J. Broberg, and A. Goscinski, Cloud Computing Principles and Paradigms. Wiley, 2011, vol. 81.
[13] P. Mell and T. Grance, "The nist definition of cloud computing," National Institute of Standards and Technology, vol. 53, no. 6, 2009.
[14] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," Journal of Network and Computer Applications, vol. 34, no. 1, pp. 1– 11, 2011.
[15] L. Youseff, M. Butrico, and D. Da Silva, "Toward a unified ontology of cloud computing," in Grid Computing Environments Workshop, 2008. GCE'08. IEEE, 2008, pp. 1– 10.