Effective Analysis of Cloud Based Intrusion Detection System

Sanjay Ram M
Adhiyamaan College of Engg.
Hosur, India

Velmurugan N
Saveetha Engg. College
Chennai, India

Thirukumaran S
Adhiyamaan College of Engg.
Hosur, India

ABSTRACT

The goal of IDS is to analyze events on the network and identify attacks. The increasing number of network security related incidents makes it necessary for organizations to actively protect their sensitive data with the installation of intrusion detection systems (IDS). People are paid more attention on intrusion detection which as an important computer network security technology. According to the development trend of intrusion detection, detecting all kinds of intrusions effectively requires a global view of the monitored network, Here, discuss about new intrusion detection mechanism based on cloud computing, which can make up for the deficiency of traditional intrusion detection, and proved to be great scalable.

Keywords

Cloud computing; GRID computing; intrusion detection system; services; security issues; attack exposure; denial-of-service.

1. INTRODUCTION

Cloud computing is a new and emerging information technology that changes the way IT architectural solutions are put forward by means of moving towards the theme of virtualization: of data storage, of local networks (infrastructure) as well as software [1] [2]. The success of modern day technologies highly depends on its effectiveness of the world's norms, its ease of use by end users and most importantly its degree of information security and control. International Data Corporation (IDC) conducted a survey of IT executives and their line-business colleagues to gauge their opinions and understand their companies' use of IT cloud services. Security ranked first as the greatest challenge or issue of cloud computing.

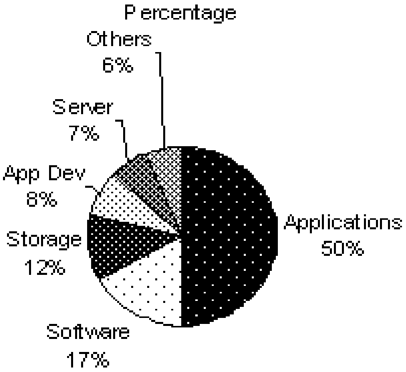

Figure 1: Worldwide IT Cloud Spending on 2012

In the vendor perspective of driving of cloud computing is easier for application vendors to reach new customers; low cost way of delivering and supporting applications; ability to use commodity server and storage hardware; ability to drive down data centre operational costs, and the customer perspective of cloud computing is faster, simpler and cheaper to use cloud applications; no upfront capital required for servers and storage; no ongoing expenses for running data centre; applications can be accessed from anywhere and anytime. Some common cloud computing challenges are: data protection, data recovery and availability, management capabilities, and the regulatory and compliance restrictions. Some of the cloud computing typical benefits are: reduced cost, increased storage, quick and easy implementation, and flexibility. To satisfy the requirements of next generation computing will need to mean more than just externalized data centre's and hosting models.

The intrusion detection technology is the process of identifying network activity that can lead to a compromise of security policy. Domestic research institutions and network security Products Company Also carried out related research, but the domestic intrusion detection products are less. However, current intrusion detection systems have several drawbacks: insufficient detection rates, too many intrusions detected or missed. In The new intrusion detection mechanism based on cloud computing, With it, on any network site, a local detection engine analyses the data collected by cloud computing centre to find intrusion patterns. Afterwards, all the generated alerts are processed by a global intrusion Detection engine to find more complex intrusions and to give a global view of the network security.

In this paper described in section 2 cloud computing overview like cloud computing services and security issues, and section 3 cloud based intrusion detection system - intrusion detection system methods and intrusion detection system services with evaluation of approaches and conclusion.

2. CLOUD COMPUTING OVERVIEW

2.1 Cloud computing services

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [3]. There are three delivery models; Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

2.1.1 Securing Infrastructure-as-a-service

The hosting of hardware in an external data centre is sometimes called as infrastructure as a service. In this model lets user's lease compute, storage, network, and other resources in a virtualized environment. The user doesn't manage or control the underlying cloud infrastructure but has control over the OS, storage, deployed applications, and possibly certain networking components. Amazon's Elastic Compute Cloud (EC2) is a good example of IaaS. At the cloud infrastructure level, CSP can enforce network security with intrusion-detection systems (IDS), firewalls, antivirus programs, distributed denial-of-service (DDoS) defenses, and so on.

2.1.2 Securing Platform-as-a-service

Platform services also called as middleware as a service. Cloud platforms are built on top of infrastructure service with system integration and virtualization middleware support. Such platforms let users deploy user-built software applications onto the cloud infrastructure using provider-supported programming languages and software tools (such as Java, Python, or .NET). The user doesn't manage the underlying cloud infrastructure.

Table 1: Cloud computing models

Categories	Services	Examples
IaaS	In this model is pay per use model, services like storage, database management and compute capabilities are offered on demand.	3 Tera, Go Grid, Amazon Web Services
PaaS	The platform used to design, develop, build and test applications are provided by the cloud infrastructure	Azure service platform, Google App Engine
SaaS	Highly scalable internet based applications are hosted on the cloud and offered as services to the end user.	Google Docs, acrobat.com, Amazon Docs

Popular platforms include the Google App Engine (GAE) or Microsoft Windows Azure. This level requires securing the provisioned enforcing security compliance, managing potential risk, and establishing trust among all cloud users and providers.

2.1.3 Securing Software-as-a-service

Application hosting is sometimes called as software as a service. This service employs browser-initiated application software to serve thousands of cloud customers, who make no upfront investment in servers or software licensing. From the provider's perspective, costs are rather low compared with conventional application hosting. Software service as heavily pushed by Google, Microsoft, Salesforce.com, and so on — requires that data be protected from loss, distortion, or theft. Transactional security and copyright compliance are designed to protect all intellectual property rights at this level. Data encryption and coloring offer options for upholding data integrity and user privacy.

2.2 Security issues

Cloud computing security issues identified seven issues that need to be addressed before enterprises consider switching to the cloud computing model. They are as follows:

- Privileged user access - information transmitted from the client through the Internet poses a certain degree of risk, because of issues of data ownership; enterprises should spend time getting to know their providers and their regulations as much as possible before assigning some trivial applications first to test the water.
- Regulatory compliance - clients are accountable for the security of their solution, as they can choose between providers that allow to be audited by third party organizations that check levels of security and providers that don't.
- Data location - depending on contracts, some clients might never know what country or what jurisdiction their data is located.
- Data segregation - encrypted information from multiple companies may be stored on the same hard disk, so a mechanism to separate data should be deployed by the provider.
- Recovery - every provider should have a disaster recovery protocol to protect user data.
- Investigative support - if a client suspects faulty activity from the provider, it may not have many legal ways pursued an investigation.
- Long-term viability - refers to the ability to retract a contract and all data if the current provider is bought out by another firm.

Table 2: Cloud computing and Grid computing characteristics

Characteristics	Cloud computing	Grid computing
Service oriented	Yes	Yes
Strong fault tolerant	Yes	Half
TCP/IP based	Yes	Half
High security	Half	Half
Loose coupling	Yes	Half
Virtualization	Yes	Half
Ease use	Yes	Half
Commercial pattern	Yes	No

Given that not all of the above need to be improved depending on the application at hand, it is still paramount that consensus is reached on the issues regarding standardization. The comparable characteristics of cloud computing and grid computing are listed in Table 2. The "yes" and "no" stand for cloud computing or grid computing has the special characteristic or not. The "half" means not owning the whole characteristic to a certain extent. This paper doesn't pay much attention on the similarities and difference between them and focuses on the essential characteristics of cloud computing [4]. The inherent or perceived risk of cloud computing is likely to be the most restricting factor in its possible success. Risk can occur in areas of availability, privacy, legislation, and data theft and user security.

The intrusion detection system meets two themes of requirements, such as functional and performance requirements [5].

The functional requirements are: IDS must continuously monitor and report intrusion; IDS should have a very low false alarm rate; IDS should provide enough information to repair the system in the case of detection of intrusion. This characteristic depends on intrusion detection system goals. In fact many intrusion detection system solutions focus only on alerting administrators without suggesting any corrective actions. Intrusion detection system must detect and react to distributed and coordinated attacks. This detection feature is one of the most difficult because it needs huge distributed amount of information in addition to the hard task of synchronization between different hosts. The IDS should adaptive to network topology and configuration changes.

The performance requirements are: intrusion should be detected in real time as it should be reported immediately in order to minimize network damage; the IDS must be scalable in order to handle additional computational and communication loads.

The most common IDS limitations include the following: high number of false positives; lack of efficiency: usually when an IDS is faced with a very large number of events in the network, it slows down a system or drops network packets; vulnerability to attacks: hierarchical structures, attackers the opportunity to harm the IDS by cutting off a control branch or even by tacking out the root command.

3. CLOUD BASED IDS

The Grid and Cloud Computing Intrusion Detection System integrates knowledge and behavior analysis to detect intrusions. Because of their distributed nature, grid and cloud computing environments are easy targets for intruders looking for possible vulnerabilities to exploit. By impersonating legitimate users, the intruders can use a service's abundant resources maliciously. To combat attackers, intrusion-detection systems (IDS) can offer additional security measures for these environments by investigating configurations, logs, network traffic, and user actions to identify typical attack behavior [1]. However, IDS must be distributed to work in a grid and cloud computing environment. It must monitor each node and, when an attack occurs, alert other nodes in the environment. This kind of communication requires compatibility between heterogeneous hosts, various communication mechanisms, and permission control over system maintenance and updates—typical features in grid and cloud environments [6]. Cloud middleware usually provides these features, so we propose an IDS service offered at the middleware layer (as opposed to the infrastructure or software layers). An attack against a cloud computing system can be silent for a network-based IDS deployed in its environment, because node communication is usually encrypted. Attacks can also be invisible to host-based IDS, because cloud-specific attacks don't necessarily leave traces in a node's operating system, where the host-based IDS reside. In this way, traditional IDS can't appropriately identify suspicious activities in a grid and cloud environment [7]. The client system is the system which wants to get service or response from a server by forwarding request to the server.

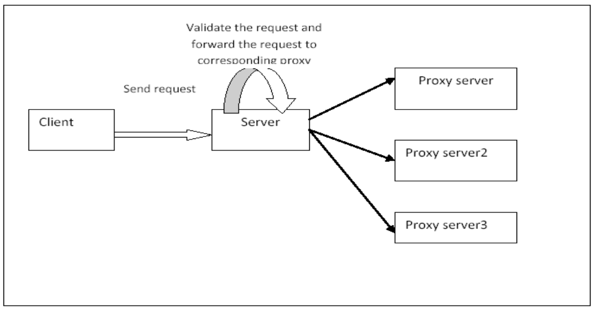

Figure 2: Client request with proxy server

An anonymous proxy serves as a middleman between your web browser and an end server. Instead of contacting the end server directly to get a Web page, the browser contacts the proxy, which forwards the request on to the end server. When the end server replies to the proxy, the proxy sends the reply on to the browser. No direct communication occurs between the client and the destination server; therefore it appears as if the HTTP request originated from the intermediate proxy server.

3.1 Intrusion detection system methods

The Intrusion Detection Service (IDS) service increases a cloud's security level by providing two methods of intrusion detection. First approach is performance approach which orders how to compare recent user actions to the usual behavior. The second approach is information approach that notices known trails left by attacks or certain sequences of actions from a user who might represent an attack. The audited data is sent to the IDS service core, which analyzes the behavior using artificial intelligence to detect deviations. This has two subsystems namely analyzer system and alert system. The analyzer uses a profile history database to determine the distance between a typical user behavior and the suspect behavior and communicates this to the IDS service. The rules analyzer receives audit packages and determines whether a rule in the database is being broken. It returns the result to the IDS service core. With these responses, the IDS calculate the probability that the action represents an attack and alerts the other nodes if the probability is sufficiently high. This subsystem will work when intrusion is detected. If any node among the cloud system is affected by intrusion then this alert system will alert the remaining nodes about the intrusion.

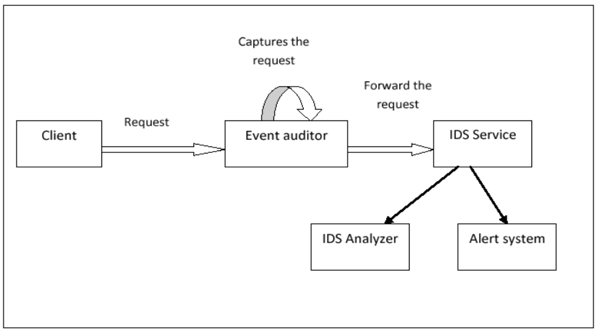

Figure 3: Client request with IDS Service

The storage service is a database system which contains two types of services namely information based service and performance based service. Whenever a node gets requests or responses, the analyzer system compares the node information in the storage service.

This paper used audit data from both a log system and the communication system to evaluate the information based system. The created a series of rules to illustrate security policies that the IDS should monitor. The information service is nothing but set of rules which is formed from previous attacks.

Following things comes under this category:

- Password cracking and access violation,
- Trojan horses,
- Interceptions most frequently associated with TCP/IP stealing and interceptions that often employ additional mechanisms to compromise operation of attacked systems (for example by flooding) man in the middle attacks.
- If any packets come with .exe extension
- Packets containing worms

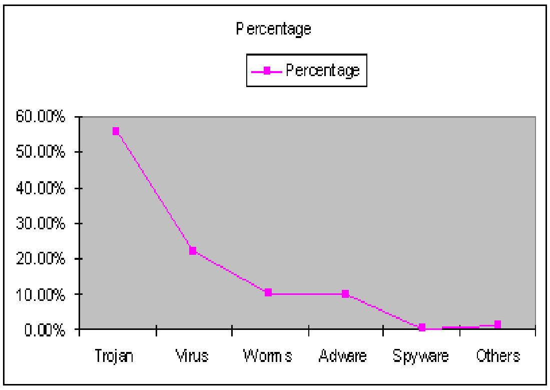

Figure 4: Recent Breakdown of the types of malware programs

In our solution, each node identifies local events that could represent security violations and alerts the other nodes. Each individual intrusion detection system cooperatively participates in intrusion detection.

The *node* contains the resources, which are accessed homogeneously through the middleware. The middleware sets the access-control policies and supports a service-oriented environment. The *service* provides its functionality in the environment through the middleware, which facilitates communication. The *event auditor* is the key piece in the system. It captures data from various sources, such as the log system, service, and node messages.

The IDS service analyzes this data and applies detection techniques based on user behavior and knowledge of previous attacks. If it detects an intrusion, it uses the middleware's communication mechanisms to send alerts to the other nodes. The middleware synchronizes the known-attacks and user-behavior databases.

The *storage service* holds the data that the IDS service must analyze. It's important for all nodes to have access to the same data, so the middleware must transparently create a virtualization of the homogeneous environment.

3.2 Intrusion detection system services

The IDS service increases a cloud's security level by applying two methods of intrusion detection. The *performance approach* orders how to compare recent user actions to the usual performance. The *information approach* notices known trails left by attacks or certain sequences of actions from a user who might represent an attack.

The audited data is sent to the IDS service core, which analyzes the performance using artificial intelligence to detect deviations. The analyzer uses a profile history database to determine the distance between a typical user performance and the suspect performance and communicates this to the IDS service.

The rules analyzer receives audit packages and determines whether a rule in the database is being broken. It returns the result to the IDS service core. With these responses, the IDS calculate the probability that the action represents an attack and alerts the other nodes if the probability is sufficiently high.

To detect an intrusion, need audit data describing the environment's state and the messages being exchanged. The event auditor can monitor the data that the analyzers are accessing.

The first component monitors message exchange between nodes. Although audit information about the communication between nodes is being captured, no network data is taken into account only node information.

The second component monitors the middleware logging system. For each action occurring in a node, a log entry is created containing the action's type (such as error, alert, or warning), the event that generated it, and the message. With this kind of data, it's possible to identify an ongoing intrusion.

3.2.1 Performance Approach

Performance approach is normal or expected performance extracted from reference information is compared with the current activity, any deviation observed, is detected as an intrusion [8].

The advantages of using performance approach are: detect attempts to exploit new and unforeseen vulnerabilities and contribute to the automatic discovery of new attacks; do not face the generalization issue; they help detect abuse of privileges types of attacks that do not actually involve exploiting any technological vulnerability.

The disadvantages of using performance approach are: high false alarm rate; periodic online retraining of the performance profile is required which results in the either unavailability of the intrusion detection system or the additional false alarms.

Numerous methods exist for performance based intrusion detection, such as data mining, artificial neural networks, and artificial immunological systems. This paper use a feed-forward artificial neural network, because—in contrast to traditional methods—this type of network can quickly process information, has self-learning capabilities, and can tolerate small performance deviations. These features help overcome some IDS limitations [9] Using this method, need to recognize expected performance (legitimate use) or a severe performance deviation. Training plays a key role in the pattern recognition that feed-forward networks perform. The network must be correctly trained to efficiently detect intrusions. For a given intrusion sample set, the network learns to identify the intrusions using its retro propagation algorithm.

However, focus on identifying user performance patterns and deviations from such patterns. With this strategy, cover a wider range of unknown attacks.

3.2.2 Information Approach

Information approach contains information about specific attacks and vulnerabilities and looks for attempts to exploit these vulnerabilities. When such an attempt is detected, an alarm is triggered. Accuracy depends on the regular update of information about attacks [8].

The advantages of using information approach are: the potential for very low false alarm rates; contextual analysis proposed by the intrusion detection system is detailed, making it easier to take preventive or corrective action.

The disadvantages of using information approach are: maintenance of the information base of the intrusion detection system and maintaining it up to date; information about attacks is much focused causing it to be closely tied to an environment; detection of insider attacks is difficult.

Information based intrusion detection is the most often applied technique in the field because it results in a low false-alarm rate and high positive rates, although it cannot detect unknown attack patterns. It uses rules (also called signatures) and monitors a stream of events to find malicious characteristics. Using an expert system, describe a malicious behavior with a rule. One advantage of using this kind of intrusion detection is that add new rules without modifying existing ones.

In contrast, performance approach is performed on learned performance that can't be modified without losing the previous learning. Generating rules is the key element in this technique it helps the expert system recognize newly discovered attacks. Creating a rule consists of defining the set of conditions that represent the attack.

3.2.3 Increasing attack exposure

The two intrusion detection techniques are distinct. The performance approach intrusion detection is characterized by a high hit rate of known attacks, but it's deficient in detecting new attacks. Therefore, the complemented it with the performance technique, which can discover deviations from acceptable use and thus help identify privilege abuse.

Rapid increase in the number of vulnerabilities has resulted in an exponential rise in the number of attacks. According to the Computer Emergency Response Team (CERT), the number of vulnerabilities in software has been increasing and many of them exist in highly deployed software [10], [11]. Considering that it is near to impossible to build 'perfect' software, it becomes critical to build effective intrusion detection systems which can detect attacks reliably. The prospect of obtaining valuable information, as a result of a successful attack, subside the threat of legal convictions. The inability to prevent attacks furthers the need for intrusion detection. The problem becomes more profound since authorized users can misuse their privileges and attackers can masquerade as authentic users by exploiting vulnerable applications.

The volume of data in a cloud computing environment can be high, so administrators do not observe each user's actions they observe only alerts from the IDS.

3.2.4 Experimental analysis

In testing our prototype, it has a low processing cost while still providing a satisfactory performance for real-time implementation. Sending data to other nodes for processing didn't seem necessary. The individual study performed in each node reduces the complexity and the volume of data in comparison to previous solutions, where the audit data is concentrated in single points. In the future, implement our IDS, helping to improve green (energy-efficient), white (using wireless networks), and cognitive (using cognitive networks) cloud computing environments. And also intend to research and improve cloud computing security.

Created data tables to perform the experiments with audit elements coming from both the log system and from data captured during node communications.

- Created data representing legitimate action by executing a set of known services simulating a regular behavior.
- Created data representing behavior anomalies. To represent anomalous sequences of actions, we altered the services and their usage frequency.
- Finally created data representing policy violation. This was prepared with a set of audit packages containing a series of elements violating base rules.

The event auditor captures all requests received by a node and the corresponding responses, which is fundamental for performance approach. For each action a node performs, a log entry is generated to register the methods and parameters invoked during the action.

In the experiments with the performance based IDS, considered using audit data from both a log and a communication system. Unfortunately, data from a log system with the exception of the message element has a limited set of values with little variation. This made it difficult to find attack patterns, so opted to explore communication elements to evaluate this technique.

In the Evaluated performance technique using artificial intelligence enabled by a feed forward neural network [12]. Increasing the sample period for the learning phase improved the results.

3.2.5 Evaluating the performance approach

To measure IDS efficiency [13] considered accuracy in terms of the system's ability to detect attacks and avoid false alarms. A system is imperfect if it accuses a legitimate action of being malicious. So, measured accuracy using the number of false positives (legitimate actions marked as attacks) and false negatives (the absence of an alert when an attack has occurred).

Anomaly detection models operate by building a model of system performance based upon the standard operation of the network or component under observation. After this model of \normal system performance has been created, current activity is compared to it. When the deviation grows greater than a threshold level, an alert is triggered [14]. Such a system has the advantage of being able to detect attacks that are not currently known. The drawback of such systems is that they often have a high false positive rate, which can lead to a lack of trust in the software.

The performance test designed also evaluated the analysis technique's cost. The performed a load tests where the program analyzed 1 to 1000 actions. The simulation involving 1000 actions is hypothetical. It surpasses the usual data volume and served as a base for understanding system performance in an overloading condition. An action took approximately 0.0003 seconds to be processed with our setup.

The training time for an input samples behavior took 2.03 seconds. However, the training was sporadic to plan updates to the performance profile database according to a routine in the execution environment (since a user's behavior tends to change with time). This helped us identify a convenient period of days for determining the profile of a legitimate user. Artificial neural networks aren't deterministic, so the number of false positives and false negatives didn't represent a linear decreasing progression. The neural network tended to avoid identifying legitimate actions as attacks there were always more false negatives than false positives when using the same quantity of input data.

No false alarms occurred during the training with simulation periods, although the uncertainty level was still high, with several outputs near zero. The algorithm showed a low number of false positives, but after several repetitions, the quantity of false positives varied, again representing the nondeterministic nature of neural networks.

3.2.6 Evaluating the information approach

In contrast to the performance approach, used audit data from both a log system and the communication system to evaluate the information based system. The created a series of rules to illustrate security policies that the IDS should monitor. Collected audit data referring to a route discovery service, service discovery and service request and response. The series of policies created tested the system's performance, although our scope didn't include discovering new kinds of attacks or creating an attack database.

Information-based intrusion detection techniques apply the knowledge accumulated about specific attacks and system vulnerabilities. The intrusion detection system contains information about these vulnerabilities and looks for attempts to exploit them. When such an attempt is detected, an alarm is raised. In other words, any action that is not explicitly recognized as an attack is considered acceptable. Therefore, the accuracy of information-based intrusion detection systems is considered good. However, their completeness depends on the regular update of information about attacks.

Our goal was to evaluate our solution's functionality and the prototype's performance. The rule below characterizes an attack in any message related to the storage service. The functions of the rule are as follows:

- At start-up, the rules stored in an XML file are loaded into a data structure.
- The auditor starts to capture data from the log and communication systems.
- The data is preprocessed to create a data structure dividing log data from communication data to provide easy access to each element.
- The corresponding policy for the audit package is verified.
- An alert is generated if an attack or violation occurred.

The performed a load test for this algorithm simulating the approach of 10 to 10000 rules for an action. Then, verified the textual or numerical field in comparison to the rules.

The analyzer performed two primary functions: it searched for nominal content, and it compared numerical intervals. Comparing 1,000 rules for an action consumed 0.0072 seconds; comparing a million rules consumed 2.03 seconds. This suggests that real-time analysis is possible up until a certain limit in the number of rules.

In testing our prototype learned that it has a low processing cost while still providing a satisfactory performance for real-time implementation. Sending data to other nodes for processing didn't seem necessary [9]. The individual analysis performed in each node reduces the complexity and the volume of data in comparison to previous solutions, where the audit data is concentrated in single points. When an intrusion-detection system is deployed, it becomes the natural primary target of hostile attacks, with the aim of disabling the detection feature and allowing an attacker to operate without being detected. Disabling the intrusion-detection system can happen in the following ways:

Denial-of-service attacks are a powerful and relatively easy way of temporarily disabling the intrusion-detection system. The attack can take place against the detector, by forcing it to process more information than it can handle (for example by saturating a network link). This usually has the effect of delaying detection of the attack or, in the worst case, of confusing the detector enough so that it misses some critical element of the attack. A second possibility is to saturate the reaction capability of the operator handling the intrusion-detection system. When the operator is presented with too many alarms, the person can easily miss the important one indicating penetration, even if it is present on the screen.

Several techniques have been developed to evade detection of an attack by intrusion-detection systems. Network-based tools, the most popular tools today, particularly suffer from these attacks involving hand-crafted network packets:

- Attack by IP fragmentation. Intrusion-detection systems have difficulties reassembling IP packets. Therefore, splitting an attack artificially into multiple packets creates a mismatch between the data in the packet and the signature, thus hiding the attack.
- Attack via the TTL (Time To Live). By altering the TTL of IP packets, it is possible to make the intrusion-detection system see packets that will not arrive at the target of the attack. By inserting fake data into the communication stream, an attacker can interleave the attack with bogus information, thus hiding the attack from the intrusion detection system while the target correctly reconstructs this attack data and reacts to it.

Intrusion-detection systems are beginning to protect themselves from these attacks, but little information is released by vendors as to the effectiveness of these protection measures. It is often difficult to assert the configuration of an intrusion-detection system, as in most cases there is no easy way to check the configuration and the proper detection of the attacks.

In the future, implement our IDS, helping to improve green (energy-efficient), white (using wireless networks), and cognitive (using cognitive networks) cloud computing environments. And also intend to research and improve cloud computing security.

4. CONCLUSION

Intrusion detection currently attracts considerable interest from both the research community and commercial companies. This paper is providing a satisfactory performance for real-time implementation. In this system implement a best remedial technique to overcome the drawbacks in the existing cloud and grid system. The individual analysis performed in each node reduces the complexity and the volume of data in comparison to previous solutions, where the audit data is concentrated in single points.

This approach increases the detection speed which meets the requirements of network communication. It improves the interactive performance of intrusion detection system for enhancing the security of the whole system. It is relatively low cost.

In the future, implement our intrusion detection system, helping to improve energy-efficient, using wireless networks, and using cognitive networks cloud computing environments. We also intend to research and improve cloud computing security.

5. REFERENCES

[1] Tim Mather, Subra Kumaraswamy, Shahed Latif, "Cloud Security and Privacy," O'Reilly, pp26-27, September 2009.

[2] M.Ali Aydin,A.Halim Zaim,K.G.khan Ceylan, "A hybrid intrusion detection system design for computer network security," Computers and Electrical Engineering, 517-526, February 2009.

[3] National Institute of Standards and Technology (NIST) Definition of Cloud Computing, http://csrc.nist.gov/groups/SNS/cloud-computing/

[4] Abdoul Karim Ganame, Julien Bourgeois, Renaud Bidou, Francois Spies, "A global security architecture for intrusion detection on computer networks", computers & security, pp 30-47, March 2008.

[5] W. Jansen, P. Mell, T. Karygiannis, and D. Marks "Applying mobile agents to intrusion detection and response", Technical report, NIST Interim Report - 6416, October 1999.

[6] Foster et al., "A Security Architecture for Computational Grids" *Proc. 5th ACM Conf. Computer and Communications Security*, ACM Press, 1998, pp83–92.

[7] S. Axelsson, "*Research in Intrusion-Detection Systems: A Survey"*, Technical report 98-17, Dept. Computer Eng., Chalmers University of Technology, 1999.

[8] R.Race and Mell, "Intrusion detection systems" NIST special reports on IDS, August 2001.

[9] A. Schulter et al., "Intrusion Detection for Computational Grids," *Proc. 2nd Int'l Conf. New Technologies Mobility, and Security*, IEEE Press, 2008.

[10] CERT / CC Statistics, http://www.cert.org/stats/.

[11] Thomas A. Longstaff, James T. Ellis, Shawn V. Hernan, Howard F. Lipson, Robert D. Mcmillan, Linda Hutz Pesante, and Derek Simmel. Security of the Internet. Technical Report The Froehlich / Kent Encyclopedia of Telecommunications Vol.15, CERT Coordination Center, 1997, http://www.cert.org/encyc_article/tocencyc.html.

[12] N.B. Idris and B. Shanmugam, "Artificial Intelligence Techniques Applied to Intrusion Detection," *Proc.2005 IEEE India Conf. (Indicon) 2005 Conf.*, IEEE Press, 2005, pp 52–55.

[13] H. Debar, M. Dacier, and A. Wespi, "Towards a Taxonomy of Intrusion Detection Systems," *International Journal Computer and Telecommunications Networking*, vol. 31, no. 9, 1999, pp 805–822.

[14] SANS Institute, "Intrusion detection FAQ", http://www.sans.org/newlook/resources/IDFAQ/ID FAQ.htm, 2000.

[15] Mihai Christodorescu, Reiner Sailer, Douglas Lee Schales, "Cloud Security is Not(Just) Virtualization Security," IBM T J.Watson Research, September 2009.

[16] Igor Muttik,Chris Barton, "Cloud security technologies," information security technical report, Elsevier Ltd, April 2009.

[17] Borje Ohlman, Anders Eriksson, "What Networking of Information Can Do for Cloud Computing," 2009 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, Nov. 2009.

[18] P.F. Da Silva and C.B. Westphall, "Improvements in the Model for Interoperability of Intrusion Detection Responses Compatible with the IDWG Model," *International Journal Network Management*, vol. 17, no. 4, 2007, pp 287–294.