International Journal of Advanced Technology in Engineering and Science
Vol. No.5, Issue No. 03, March 2017
www.ijates.com
ijates
ISSN 2348 - 7550

SECURITY ISSUES IN CLOUD COMPUTING

Mrs. Manisha Bharati[1],Dr. Sharvaree Tamane[2]

[1]Computer Department- ICEM/ SPP (India)

[2]IT Department- JNEC/ BAMU (India)

ABSTRACT

This paper introduces the main concepts of cloud computing systems from the unique perspectives of IT network and security. Furthermore, it introduces the cloud essential characteristics and deployment and service models. It also presents cloud security definition, the risks a cloud user should assess before committing and top threats to clouds. Some miscellaneous concepts regarding host based, network based attacks are also presented. Finally it discusses DDoS attacks and their detection in cloud systems and IDS Softwares.

***Keywords*: Cloud Computing, Security, Entropy, IDS, DDoS, Threat, Attack, Snort.**

I. INTRODUCTION

Cloud computingcurrent definitions addresses cloud systems from distinct perspectives. Here we assume the perspectives of IT network and security. According to NIST (National Institute of standards and Technology) [2], "Cloud computing ('cloud') is an evolving term that describes the development of several existing technologies and approaches to computing into something different. Cloud separates application and information resources from the basic infrastructure, and the tools used to deliver them". According to Ian Foster et al. [4], "Cloud computing is a large-scale distributed computing paradigm that is driven by economies of scale, in which a group of abstracted, virtualized, dynamically-scalable, coped computing power, storage, platforms, and services are delivered on demand to external customers over the Internet". There are a few key points in this definition. Cloud computing is a specific distributed computing paradigm; it differs from traditional ones in that

1) It is massively scalable.

2) Can be encapsulated as an abstract entity that delivers different levels of services to customers outside the cloud.

3) It is driven by economies of scale, and

4) the services can be dynamically configured via virtualization or other approaches and delivered on demand.

NIST defines cloud computing in terms of five essential characteristics, three service models, and four deployment models. They are summarized in visual form in Figure 1.1 and explained below as in [2, 4].

1.1 Essential Characteristics of Cloud Computing

Cloud services exhibit five essential characteristics that demonstrate their relation to, and differences from, traditional computing approaches [2]:

- **On-demand self-service**. A consumer can unilaterally provision computing capabilities as needed and automatically, without human interaction with a service provider.

- **Broad network access**. Computing capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g. mobile phones, laptops, and PDAs) as well as other traditional or cloud based software services.
- **Resource pooling.** A provider pools computing resources to serve several consumers using a multi-tenant model, which dynamically assigns and reassigns physical and virtual resources according to consumer demand. There is a degree of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources.
- **Rapid elasticity.** Capabilities can be rapidly and elastically provisioned, in most cases automatically and rapidly released to quickly scale out and scale in. For a consumer, the capabilities appear to be unlimited and can be purchased in any quantity at any time.
- **Measured service.** Cloud systems automatically control and optimize resource usage by leveraging a metering capability according to the type of service. Usage can be monitored, controlled, and reported, providing transparency for both the provider and the consumer.

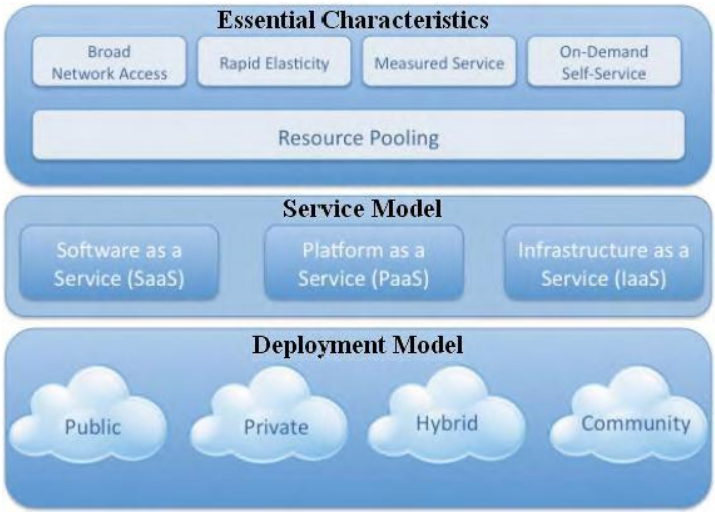

Figure 1.1: NIST Visual Model of Cloud Computing Definition

1.2 Cloud Service Models

In general, clouds offer services at three different levels [4]: IaaS, PaaS, and SaaS. However, some providers can expose services at multiple levels.

Software as a Service (SaaS) delivers software that is remotely accessible by consumers through the Internet with a usage-based pricing model. E.g., Live Mesh from Microsoft allows files and folders to be shared and synchronized across multiple devices.

Platform as a Service (PaaS) offers a high-level integrated environment to build, test, and deploy custom applications as in Google's App Engine [7]. Inside this layer resides the middleware system, a portable component for both grid and cloud systems. Examples include WSO2 Stratos [5], Windows Azure [6], and our middleware HIMAN [8, 9, and 10]

Infrastructure as a Service (IaaS) provisions hardware, software, and equipment's to deliver software application environments with a resource usage-based pricing model. Infrastructure can scale up and down dynamically based on application resource needs. Typical examples are Amazon EC2 (Elastic Cloud Computing) Service [11], Eucalyptus [12], Microsoft Private Cloud [13].

1.3 Cloud Deployment Models

There are four deployment models for cloud services, with derivative variations that address specific requirements:

- **Public Cloud.** The cloud is made available to the general public or a large industry group and is owned by an organization selling cloud services.
- **Private Cloud.** The cloud is operated solely for a single organization. It may be managed by the organization or by a third party, and may exist on-premises or off- premises.
- **Community Cloud.** The cloud is shared by several organizations to support a specific community that has shared concerns. It may be managed by the organizations or by a third party and may exist on-premises or off-premises.
- **Hybrid Cloud.** The cloud infrastructure consists of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability.

II. CLOUD COMPUTING SECURITY

Cloud computing may adopt the same control of any IT environment. However, the cloud service models, the operational models, and the supporting technologies change the risk landscape for an organization with respect to traditional IT. The next section outlines seven risks a user should consider before committing and seven top threats to cloud computing systems

2.1 Seven Risks to be analyzed before Committing

There are seven possible risks a user should assess before committing [14]:

- Privileged user access: sensitive data should be processed outside the enterprise only with the assurance that they are only accessible and propagated to privileged users.
- Data segregation: is the user data should be fully segregated from data of other users
- Regulatory compliance: a cloud provider should have external audits and security certifications and the infrastructure should comply with regulatory security requirements.
- Data location: the cloud provider should commit to storing and processing data in specific jurisdictions and to obey local privacy requirements on behalf of the customer;
- Recovery: the provider should offer an efficient replication and recovery mechanism to fully exploit the potentials of a cloud in the event of a disaster;
- Investigative support: support should to be ensured for forensics and investigation with a contractual commitment.
- Long-term viability: a user data should be accessible even when the provider is acquired by another company or the user moves to another provider.

2.2 Top Seven Threats to Cloud Computing

We briefly highlight seven threats that CSA (Cloud Security Alliance) [1] ranks and that apply across all of the different cloud computing models.

Threat #1: Abuse and Nefarious Use of Cloud Computing

The top threat that CSA identifies is the abuse and nefarious use of cloud computing. This is related to the use of botnets to spread spam and malware. Attackers can infiltrate a cloud system, by abusing the relative anonymity behind the cloud registration system and usage models. Then, they can upload malware and use the power of the cloud to attack other machines. The CSA suggests to:

1. Monitor public blacklists for one's own network blocks.
2. Use a stricter initial registration and validation processes.
3. Enhanced credit card fraud monitoring and coordination.

Threat #2: Insecure Interfaces and APIs

The CSA cautions against unsure application programming interfaces between applications for interoperability. The CSA suggests to:

1. Analyze the security model of cloud provider interfaces.
2. Ensure strong authentication and access controls are implemented in concert with encrypted transmission. Some Grid and Cloud portals can be used for this target e.g. Nubifer [15], Ubuntu Portal [16], and our HIMAN-GP [17].
3. Understand the dependency chain associated with the API.

Threat #3: Malicious Insiders

Organizations need to assess the risk on the service provider's end and demand segregation of duties to prevent a malicious insider from accessing data. The CSA suggests to:

1. Enforce strict supply chain management and conduct a comprehensive supplier assessment.
2. Specify human resource requirements as part of legal contracts.
3. Require transparency into overall information security and management practices, as well as compliance reporting.
4. Determine security breach notification processes.

Threat #4: Shared Technology Issues

Cloud users have to be aware of vulnerabilities in shared technologies, such as VMs, communications systems or key management technologies. A zero-day attack can use these technologies and quickly spread across a public cloud and expose all data within it. The CSA suggests to:

1. Implement security best practices for installation/configuration.
2. Monitor environment for unauthorized changes/activity.
3. Promote strong authentication and access control for administrative access and operations
4. Enforce service level agreements for patching and vulnerability remediation.
5. Conduct vulnerability scanning and configuration audits.

Threat #5: Data Loss or Leakage

There are several alternative ways to compromise data. Deletion or alteration of records without a backup is an obvious example. A cloud increases the risk of data compromise, due to risks and challenges which are either unique to cloud, or more dangerous because of the architectural or operational characteristics of a cloud environment.

The CSA suggests to:

1. Implement strong API access control.
2. Encrypt and protect integrity of data in transit. There are many encryption schemes for high performance systems e.g., GridCrypt [18] and our "Ultra GridSEC" [19, 20, 21].
3. Analyzes data protection at both design and run time.
4. Implement strong key generation, storage and management, and destruction practices.
5. Contractually demand providers wipe persistent media before it is released into the pool.
6. Contractually specify provider backup and retention strategies.

Threat #6: Account or Service Hijacking Cloud users need to be aware of account service and traffic hijacking. Examples for attacks that may cause these threats are: man-in-the-middle, phishing, spam campaigns, and DDoS. Cloud solutions add a new threat to the landscape. If an attacker gains access to a user credentials, then she can eavesdrop on activities and transactions, manipulate data, return falsified information, and redirect the user clients to illegitimate sites. The CSA suggests to: 1. Prohibit the sharing of account credentials between users and services. 2. Leverage strong two-factor authentication techniques where possible. 3. Employ proactive monitoring to detect unauthorized activity. 4. Understandcloud provider security policies and SLAs.

Threat #7: Unknown Risk Profile One of the tenets of cloud computing is the reduction of hardware and software ownership and maintenance costs to allow companies to focus on their core business strengths. This has clear financial and operational benefits, which must be weighed carefully against the contradictory security concerns when the migration to a cloud is driven by expected saving only by groups who may lose track of security issues. Information about who is sharing an infrastructure may be pertinent, in addition to network intrusion logs, redirection attempts and/or successes, and other logs. An IDS is the ideal tool for this threat, as it can deal with all suggestions of CSA like:

1. Disclosure of applicable logs and data.
2. Partial/full disclosure of infrastructure details (e.g., patch levels, firewalls, etc.).
3. Monitoring and alerting on necessary information.

III. HOST, NETWORK, AND DDOS ATTACKS

Attacks utilize network media and manipulate computing and/or network resources to severely degrade the performance of the services of an ICT network and eventually shutdown the entire network. We can classify attacks according to the type of penetration (inside, outside), type of interactions (passive, active) and the mechanism to launch the attack. [22, 23]

Penetration Type: Penetration can be carried out as an outsider or as an insider. Insiders are legal users that are conducting malicious activities through their accounts or by illegally using other user accounts. Instead, an outsider launches attacks from outside the network perimeter or implements probing or scanning attacks to acquire information on the network before launching the real attacks. Potential outsiders range from amateur to organized crime, cyber terrorists, and hostile governments.

Interaction Type: Attack classification should also consider the interaction between the attackers and the network environment. Based on this criterion, network attacks can be either classified as active or passive. In a passive attack (e.g., wiretapping, port scanner, idle scan), the attacker listens to the streams of traffic to gather valuable information. Thus the anomalous behaviors caused by this type of attacks are hard to observe because

they leave the minimum footprint. Active attacks aim to change the configuration of system resources or affect their operation (e.g., Denial of Service Attacks, Spoofing, Man-in-middle attack, ARP positioning). They trigger an anomalous behavior that can be observed and quantified provided that the appropriate metrics are used.

Mechanism Type: the mechanisms and techniques to launch an attack can be categorized into five classes: Denial of Service (DoS), User to Root (U2R), Remote to Local, probing, and virus/worm attacks.

Denial of Service (DoS) attack: It prevents services for the users by limiting or denying their access to system resources such as bandwidth, memory, buffers, and/or processing power. To this purpose, these attacks can target software vulnerabilities, change configuration, or exhaust the network resource to its limit. Possible examples include ICMP Nukes, Teardrop, Land Attack, the ping of death, and playing with the configuration of a compromised router. While these attacks can be easily fixed by installing proper software patches, reloading correct configuration, and limit the access to resources, they impose a critical load on network administrators that increases with the number of attacks. Section 1.9.2 describes a popular attack in this class, the Distributed Denial of Service (DDoS).

User to Root (U2R) attack: Attackers with login access can bypass authentication to gain the higher privileges of another user in controlling and accessing the system.

Remote to Local (R2L) attack: Attackers can bypass normal authentication and execute commands and programs on the target with local machine privileges.

Probe/Scanning attacks: These attacks blueprint the network and its resources to discover vulnerability or entry points that the attacker can use to penetrate or attack network resources.

Worm/virus: This attack is run by a malicious piece of code that spreads across a network and targets hosts or network resources to cause dysfunction, data loss, or data theft.

Attacks against an information system can also be classified according to the number of involved computers. An attack that may involve even a large number of computers is the DDoS ones outlined in Section 1.9.2. Attacks can also be classified into network or host ones according to the mechanism or the type of vulnerabilities they exploit. [45] presents classification criteria based on attack surfaces of the cloud computing scenario participants as shown in

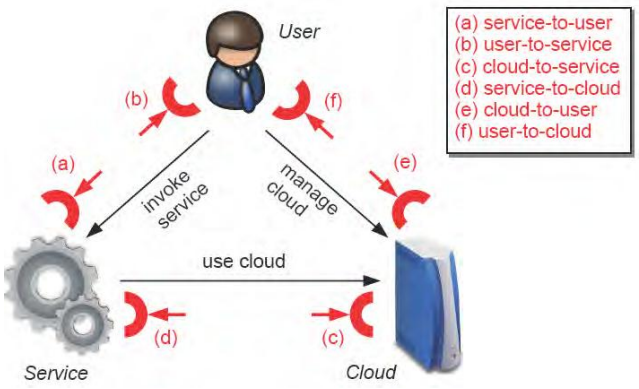

Figure 3.1

Figure 3.1: Taxonomy for attacks on cloud services (a) Service-to-User: includes attacks in common client-server architectures, e.g. buffer overflow attacks, SQL injection, or privilege escalation.

(b) User-to-Service: includes attacks in the common environment of client program, e.g. browser-based attacks, attacks on browser caches, or Phishing attacks on mail clients.

(c) Cloud-to-Service: includes attacks of a service instance against its hosting cloud system, e.g. the resource exhaustion attacks, or attacks on the hypervisor.

(d) Service-to-Cloud: incorporates attacks of a cloud provider against a service, e.g., availability reductions, privacy related attacks or even malicious interference. This category is by far the most critical one, as the provider can implement them in a rather simple way and attack impacts are tremendous.

(e) Cloud-to-User: includes user attacks against the interface of the cloud system to control the provided services and that enables the customers to add new services or change the number of service instances.

(f) User-to-Cloud: involves every kind of attack that targets a user and that originates from the cloud system. It is similar to the phishing attempts to trigger a user into manipulating cloud-provided services.

3.1 Host and Network Attacks and Their Libraries

We briefly classify attacks into network and host ones and describe the libraries that support their implementation.

Network attacks exploit vulnerabilities in the communication protocols or in the interconnection structure to attack the integrity and confidentiality of communications. As an example, since most communications adopt an unsecured or clear text format, an attacker that can access network data paths can also read and interpreter the traffic these paths transmit. Some examples of these attacks are [22]:

(1) Eavesdropping: it is also known as sniffing or snooping. This attack monitors the network traffic.

(2) Data Modification: it modifies transmitted data in a way that cannot be detected by the sender or the receiver.

(3) Identity or IP Address Spoofing: it builds IP packets that appear to originate from valid addresses to modify, reroute, or delete some data. It is supported by specialized libraries.

(4) Denial-of-Service Attack (DoS): It shuts down applications or network services by flooding them with invalid traffic. This can prevent legal user from accessing network resources.

(5) Man-in-the-Middle Attack: This attack inserts a distinct entity between two communicating components to capture and modify their communications.

Host based attacks are enabled by vulnerabilities in the host OS or in the applications. Some classes of these attacks are [Host-attack]:

- Buffer overflow: It violates memory safety to overwrite adjacent memory positions. It exploits the lack of controls on the size of a parameter
- Rootkit: It installs software components to hide a malicious processes running on the node and that grants to the attacker a privileged access to the system.
- Format string: It can crash a program or execute harmful code. It exploits the lack of control on user inputs such as the format string in some C functions.

Several libraries have been developed to support host and network attacks. As an example, Metasploit [46] is a consistent and reliable library of constantly updated exploits for network, OSs and applications. An exploit is a code fragment to automate, at least partially, an attack. Metasploit defines a complete environment to develop

new tools and automate every aspect of an attack. It simplifies the development of attack vectors to extend its exploits, payloads, encoders to create and execute more advanced and specialized attacks against a target system.

3.2 DDoS Attacks

Distributed Denial of Service (DDoS) attacks [26] are a class of attacks that disrupt the service quality of a system. It is worth considering these attacks in relation with clouds because their effectiveness increases if an attacker can use the massive amount of resources in a cloud.

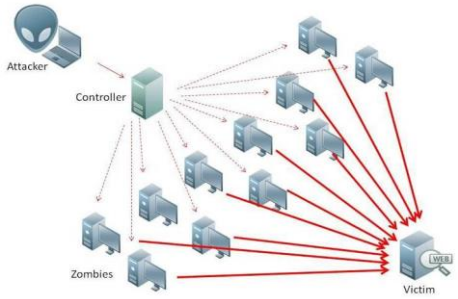

Figure.3.2: The DDoS Strategy.

Figure 3.2 shows the four elements of DDoS attacks [27] namely:

(1) The attacker machine.

(2) The handlers/controller: these are hosts controlled by the attacker as a result of a previous attack. They run some malware and act as an intermediate interfaces to control the agents and route to them the attacker commands.

(3) The agents or zombie hosts: also these hosts are controlled by the attacker. They run some malware that either implements an attack on behalf of the attacker (botnets) or generates a stream of packets towards the target system.

(4) The victim or target system.

While several kinds of DDoS attacks exist, any implementation of these attacks includes the following stages:

(1) Search of vulnerable hosts to act as handlers and zombies. This step can exploit a standard vulnerability scanner such as Nessus [28].

(2) Compromising the vulnerable hosts: The attacker exploits the vulnerabilities returned by the scanner to attack some vulnerable hosts and stealthy install some malware.

(3) Communication, broadcasting, and flooding: The attacker communicates a command to one or more handlers. Then, the handler broadcasts any received commands to hundreds or even thousands of zombies that start flooding the network of the target system until the attacker sends a stop command.

In the experiments, we implement DDoS attacks through the LOIC library. LOIC is one of the most powerful free DOS and DDOS attacking tool, it attempts to open several connections to the same target host and continuously floods it with false TCP or UDP packets, or with HTTP requests that lead to a service disruption. A DDOS attack runs LOIC through multiple zombies. Another library we have used is the CPU Ping Death library [28]. It is a DDoS attacking tool that opens multiple floods to a large number of hosts and continuously floods them with fake packets and HTTP requests to reduce their bandwidth and their performance.

3.3 Current DDoS Detection Techniques in Cloud systems.

We briefly review some IDSs that have recently been proposed to detect DDoS attacks in clouds.

[29] Investigates the effect of DDoS on clouds and proposes IDS based on the behavioral threshold. The IDS assumes that a user is attacking the system if the user requests are outside the normal user range. The threshold is automatically determined as a dynamic variable according to the network position and pressure traffic. To simplify the discovery of legal users, several solutions may be integrated with the IDS such as load balancing of the network traffic and a honeypot [30]. The latter discovers the attacker signatures by analyzing the collected data. The IDS does not correlate network events in distinct virtual zones of the cloud. Furthermore, no deployment in a real cloud system is described and the accuracy and the performance of the IDS are not evaluated[31] uses an IDS sensor such as the version of Snort [32] installed on VMware ESX [33] machine that sniffs both in-bound and out-bound traffic to detect DoS attacks. Snort analyzes in-bound packets and looks for several intrusion patterns. If at least one matches, it drops all the packets from the same IP address. The accuracy and performance of this solution is not evaluated. Furthermore, also this solution does not correlate network events to discover attacks against several virtual zones. [34] Proposescooperative IDS that reduces the impact of DoS attack in each cloud regions. Several IDS components are distributed across these regions and a cooperative module receives their alert messages. Then, a majority vote determines the trustworthiness of these alerts. This system avoids any single point of failure but its accuracy is not satisfactory. Furthermore, it has not been evaluated against a DDoS attack.

The analysis of current solutions confirms that a defense strategy for clouds against DDoS attacks introduces some further requirements with respect to those for traditional systems. To be adopted in clouds, a solution needs to:

(1) Be distributed and scalable,

(2) Avoid single points of failure,

(3) Correlate the user behaviors in distinct environments.

(4) Integrate different service models.

IV. INTRUSION DETECTION SOFTWARE

In our deployment, we used some open source IDSs and tools to detect host, network, and DDoS attacks based on the signature based analysis techniques namely, OSSEC, Snort, and OSSIM.

1) OSSEC

OSSEC [37] is an Open Source Host-based Intrusion Detection System. It performs log analysis, file integrity checking, policy monitoring, rootkit detection, real-time alerting and active response. It runs on most OSs and it has two types of installation, Local and Agent-Server. In the local installation, OSSEC only protect a local machine. Instead, the Agent-Server installation protects the machines of the network. The agents are installed in several hosts systems to report back to a central OSSEC server to aggregate the information from the agents, analyze it and fires alerts.

Figure 1.13: The Analysis flow chart of OSSEC

As shown in Figure 1.13, the analysis processes includespre-decoding, decoding, rule matching, and alerting.

- The Pre-decoding process extracts the static information such as the event message, the location or the program name.
- The Decoding process extracts non static information such the attributes of the regular expression that defines each field.
- The Rule Matching process applies the Rule Matching Engine to determine if the received event matches any stored rules to fire an alert.
- The Alerting process determines where the rules should be sent. Alerts can be emailed to the user or logged into database.

2) Snort

Snort [32] is an open source network intrusion detection system that can log network packets. It uses a rule-based language that integrates signature, protocol, and anomaly inspection methods. Snort consists of five main components, see Figure 1.14, namely, Packet Decoder, Preprocessors, Detection Engine, Logging and Alerting, and the output module.

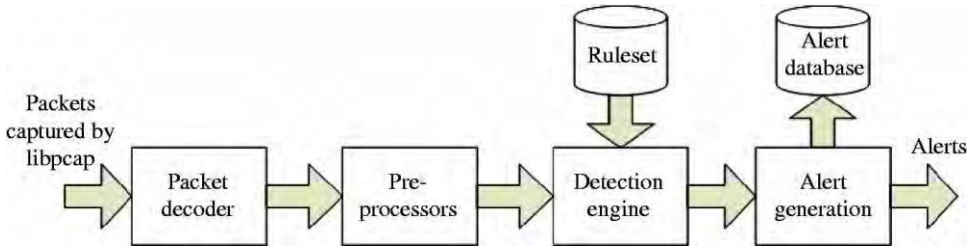

Figure 4.1: Snort Architecture

The analysis processes of Snort are summarized below:

(1) It captures packets from network using "LibPCap" component.

(2) The packet decoder component receives packets from different types of network interfaces (Ethernet, SLIP, PPP…), prepares a packet for processing and fits it at the data structure.

(3) The preprocessor component prepares data for the detection engine. It also handles defragmentation and TCP streams and detects anomalies in packet headers.

(4) The detection engine, the most important component, detects if any intrusion activity exists in a packet by applying a rule-based string matching algorithm. The algorithm dissects the packet and applies rules on different parts of the packet. If a packet matches any rule, appropriate action is taken. Otherwise no action is taken.

Finally, the Output Module processes alerts and logs and generates the final output according to the user policy and the packet content.

3) OSSIM

OSSIM [38] provides a common framework for the deployment, configuration, and management of security tools including IDS sensors. It offers event collection, normalization, correlation and incident response. We modified two modules from OSSIM, the normalization and correlation to integrate all alerts from different IDSs analyzers in the cloud i.e., OSSEC and Snort alerts by applying the IDMEF protocol.

V. CONCLUSION

Even if there is a large consensus on the benefits of cloud computing, concerns are being raised about the security issues introduced through the adoption of model and to the lack of control by the cloud users on some

architectural levels. The effectiveness and efficiency of traditional protection mechanisms can be reconsidered. Cloud computing environments are easy targets for intruders and pose new risks and threats to an organization because of their service and operational models, the underlying technologies, and their distributed nature. In particular, some kind of sharing is intrinsic to cloud computing and cannot be avoided. In turns, this blurs the traditional distinction between private and shared resources. In principle, IDSs are among the efficient security mechanisms that can handle most of the threats of cloud computing. However, several deficiencies of current IDSs technologies and solutions hinder their adoption in a cloud.

REFERENCES

[1] "Top Threats to Cloud Computing", Cloud Security Alliance, http://www.cloudsecurityalliance.org/csaguide.pdf, Version 1.0 (2010)

[2] "Security Guidance for Critical Areas of Focus in Cloud Computing", Cloud Security Alliance, http://www.cloudsecurityalliance.org/guidance/csaguide.pdf

[3] H. A. Kholidy, F. Baiadi, S. Hariri, "DDSGA: A Data-Driven Semi-Global Alignment Approach for Detecting Masquerade Attacks", in IEEE Transactions on Dependable and Secure Computing, under review in September 2012.

[4] Foster, I.; Yong Zhao; Raicu, I.; Lu, S., "Cloud Computing and Grid Computing 360-Degree Compared", Grid Computing Environments Workshop, 2008. GCE '08, vol., no., pp.1-10, 12-16 Nov. 2008

[5] http://wso2.com/cloud/stratos/

[6] "Windows Azure System" http://www.microsoft.com/windowsazure/windowsazure/

[7] Google App Engine, http://code.google.com/appengine/, 2008.

[8] Ali E. El-Desoky, Hisham A., Abdulrahman A. Azab, "A Pure Peer-to-Peer Desktop Grid Framework with Efficient Fault Tolerance", ICCES 24, Nov. 2007.

[9] A. A. Azab, H. A Kholidy, "An Adaptive Decentralized Scheduling Mechanism for Peer-to-Peer Desktop Grids", (The 2008 International Conference on Computer Engineering & Systems) 25-27 Nov, 2008

[10] A. Azab, H. Meling, "Broker Overlay for Decentralized Grid Management", JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY, January 31, 2010.

[11] Amazon Elastic Compute Cloud, http://aws.amazon.com/ec2, 2008.

[12] Eucalyptus, http://eucalyptus.cs.ucsb.edu/, 2008.

[13] Microsoft Private cloud, http://www.microsoft.com/en-us/server-cloud/private-cloud/default.asp

[14] J. Brodkin. "Gartner: Seven cloud-computing security risks", http://www.networkworld.com/news/2008/070208-cloud.html, 2008.

[15] "Nubifer Cloud Portal", http://www.nubifer.com/SaaS-pass-hass-cloud-products/top-tier-cloud-platforms.html

[16] "Ubuntu Cloud Portal", http://cloud.ubuntu.com/2010/12/announcing-ubuntu-cloud-portal/

[17] H. A. Kholidy," HIMAN-GP: A Grid Engine Portal for controlling access to HIMAN Grid Middleware with performance evaluation using processes algebra", 2nd International Conference on Computer Technology and Development (ICCTD 2010), 2-4 Nov 2010.

[18] A. Luther, RajkumarBuyya, Rajiv Ranjan, and SrikumarVenugopal, "GridCrypt: High Performance Symmetric Key Cryptography Using Enterprise Grids ", Laurence Yang and MinyiGuo (editors), Wiley Press, New Jersey, USA. Fall 2004.

[19] M. M. Mostafa, Safia H Deif, Hisham A Kholidy, "ULTRA GRIDSEC: Peer-to-Peer Computational Grid Middleware Security Using High Performance Symmetric Key Cryptography", the 5th International Conference on Information Technology-New Generations, Las Vegas, Nevada, USA, 7- 9 April 2008.

[20] Hisham A K, A. A. Azab, Safia H Deif, "Enhanced 'ULTRA GRIDSEC': Enhancing High Performance Symmetric Key Cryptography Schema Using Pure Peer-to-Peer Computational Grid Middleware (HIMAN)" in the Third International Conference on Pervasive Computing and Applications, 06-08 Oct 2008.

[21] H. A. K, K. S. Alghathbar, "Adapting and accelerating the stream Cipher algorithm "RC4" using "Ultra Gridsec" and "HIMAN" and use it to secure HIMAN Data", The Journal of Information Assurance and Security, JIAS/2009/SI4-008, Atlanta, USA, July 30, 2009. URL: http://www.softcomputing.net/jias/jias2009.html

[22] http://en.wikipedia.org/wiki/Attack (computing)

[23] G. N.; Jensen, M.;,"Attack Surfaces: A Taxonomy for Attacks on Cloud Services," Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, vol.,no., pp. 276-279, 5-10 July 2010, doi: 10.1109/CLOUD.2010.23

[24] http://www.metasploit.com/

[25] Stein, Lincoln. The World Wide Web Security FAQ, Version 3.1.2, February 4, 2002. http://www.s3.org/security/faq/ - visited on October 1, 2002.

[26] https://www.scmagazineuk.com/34-teens-arrested-by-europol-in-ddos-for-hire-crackdown/article/579090/ by Max Metzger

[27] Montoro, R.; "LOIC DDoS Analysis and Detection", URL: http://blog.spiderlabs.com/2011/01/loic-ddos-analysis-and-detection.html, 2011, Accessed December 1, 2011

[28] "NESSUS System", http://www.tenable.com/products/nessus

[29] A Mohammad Sharifi, Saeed K. Amirgholipour1, Mehdi Alirezanejad2, BaharakShakeriAski, and Mohammad Ghiami "Availability challenge of cloud system under DDoS attack", Indian Journal of Science and Technology, Vol. 5 No. 6 (June 2012) ISSN: 0974- 6846

[30] A Sardana and RJoshi, "An auto responsive honeypot architecture for dynamic resource allocation and QoS adaptation in DDoS attacked networks", Journal of Computer and Communications, July 2009, Vol. 32, P 121384-1399.

[31] ABakshi, Y. B. Dujodwala, "Securing Cloud from DDoS Attacks Using Intrusion Detection System in Virtual Machine", Proceedings of the 2010 Second International Conference on Communication Software and Networks(ICCSN '10), P 260-264

[32] Weir, J.; "Building a Debian\Snort based IDS", URL: http://www.snort.org/docs, 2011. Accessed November 28, 2011

[33] VMware cloud, http://www.vmware.com/solutions/cloud-computing/index.html

[34] Chi-Chun Lo, Chun-Chieh Huang and Ku, J, "A Cooperative Intrusion Detection System Framework for Cloud Computing Networks". In 2010 39th International Conference on Parallel Processing Workshops.

[35] http://www.citrix.com/products/xenserver/overview.html

[36] David Chappell, "THE MICROSOFT PRIVATE CLOUD", A Technical Overview Report, August, 2011.

[37] "OSSEC System", http://www.ossec.net/main/

[38] OSSIM Manual, http://www.alienvault.com/documentation/index.html