Virtual Machine-based Intrusion Detection System Framework in Cloud Computing Environment

Huaibin Wang
Key Laboratory of Computer Vision and System, Ministry of Education Tianjin University of Technology, Tianjin, China
Email: hbwang@tjut.edu.cn

Haiyun Zhou and Chundong Wang
Key Laboratory of Computer Vision and System, Ministry of Education Tianjin University of Technology, Tianjin, China
Email:haiyun313@163.com

***Abstract*—Cloud computing an emerging approach by sharing infrastructure is an overwhelming trend. While in the process of cloud deployment, the security issues can not be underestimated. Traditional Intrusion Detection System (IDS) because of lower detection rate and higher false rate couldn't be suitable the cloud here. Extensibility is the main requirement for IDS framework of cloud environment in this paper as follows. First the cross-platform and strong isolation properties of virtualization have been fully reflected here, that is to say, an extensible VM-based multiple IDSs are deployed in each layer to monitor specific virtual component. Moreover, during the process, we also propose the cloud alliance concept by the communication agents exchanging the mutual alerts mainly to resist Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) - the single point attack of failure. On this basis, we have the identity certification of the communication agents to improve the reliability of the alerts. Through the comparison of simulation results, the proposed system framework has a great advantage for monitoring VMs on the detection rate.**

***Index Terms*—cloud computing, VM-based IDS, cloud alliance, communication agent, detection rate**

I. Introduction

Cloud computing the new IT concept is being spread rapidly around the world. Cloud computing is named by the model of service delivering and using. It is regarded as on-demand scalable way to obtain the necessary service which can be Internet-based software services, bandwidth services or any other services. This is our so called cloud. The services provided by cloud computing could be divided into three types:

1) Cloud Computing Software as a Service (SaaS) provided by the cloud service provider could be accessed by the interfaces of a variety of clients. The underlying infrastructure of the cloud including networks ,servers, operating systems, storage or even a single application functionality need not to be managed by the user. All customers share a single instance of the hosted application, with the virtualization system managing access. Its typical case is Google's app engine [1], and so on.

2) Cloud Computing Platform as a Service (PaaS), this application development environment created by the tools (such as Java, python, .Net) provided by provider are automatically developed to the cloud computing infrastructure. The underlying cloud infrastructure including networks, servers, operating systems and storage needn't to be managed and controlled by the user. The consumer could control and deploy the application and environment. For example, this type of service could be provided by Windows Azure [2].

3) Cloud Computing Infrastructure as a Service (IaaS) computing power, storage capacity, network rent provided by provider is available to users. Any software including operating system and application configuration could be deployed by users. The underlying infrastructure are not be controlled or managed by users. Amazon [3] is a typical IaaS service provider.

Intrusion detection system has been developed for a long time as a security mechanism for monitoring, and resisting the intrusion. The IDS can be divided into host-based intrusion detection system (HIDS), which is used to monitor the software application of the single host by the means of verifying the operation system and checking the log file, the file system message and the connections of network as in [4]. Network-based intrusion detection system (NIDS) is often used as the non destructive way. The flow of information in the LAN area could be captured by the system and compared with the known attack signatures as in [5]. Distributed intrusion detection system (DIDS), according to the scope of intrusion, is a kind IDS designed to discover attacks on single host as well as the network which is used to aggregate data generated by individual intrusion detection systems [6]. The IDS technology could also be divided into signature-based detection and anomalous-based detection. The

doi:10.4304/jcp.7.10.2397-2403

former is used to describe the known attack and intrusion model and form the corresponding event model. When the audited events match the known attack, then the alert is generated as in [7]. The measurement parameters including the number of audit events, time interval, resources consumption, are often used by the latter method as in [8]. The average of measuring property will be used to compare with the behavior of network and system. Any observation outside the normal range is considered to invasion by NIDS. Owing to different deployment mechanisms, IDS can be divided into software-based IDS, hardware-based IDS and VM-based IDS [9]. Strong isolation, fast recovery, and cross-platform are the strengths of the virtualization technology. So the newly emerged VM-based IDS implementations are usually more strong, practical and convenient. In the proposed virtual cloud infrastructure, due to the highly heterogeneous architecture, the VM-based IDS new structure is the core of the paper.

Owing to the combination of the means of cloud service and the different deployment of cloud computing, it brings about the security problem generated by each layer during executing the system. In addition, new security challenges emerge such as how to resolve the deployment of the virtual infrastructure in cloud platform when virtual technology provides the flexible deployment of resource for cloud computing platform. In order to ensure the above issues are more reasonably controlled, it's necessary to deploy IDS sensors to monitor the separated VM at each layer which is controlled by the VM management unit. In order to integrate and analyze the alerts generated by multiple distributed sensors deployed in cloud, a plug-in-concept is proposed in core management unit including collector component, database, threshold compare. They are mainly used to integrate, unify and analyze a large number of security-related events generated by sensors. Furthermore, in the paper, the single point attack of failure must be considered, that is to say, to realize cloud alliance concept by the communication agents. The alert information must be mutual exchanged by communication agents reducing the impact of Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS). While in order to ensure the communicate agents' robustness so that they could not be disguised by intruders, we introduce a third party certificate module here. All above are proposed in proof-of-concept to realize the architecture.

The rest paper is arranged as follows. II describes the cloud computing involved related work. III introduces the proposed infrastructure. IV simulates about the infrastructure. V lists the future work. VI gives a brief summary.

II. THE RELATED WORK

The simple cloud model and its associated threats, virtualization technology, DoS and DDoS attack are mainly introduced here.

A. The Cloud Model,Technologies-related and Threats Vulnerably Suffered at Each Layer.

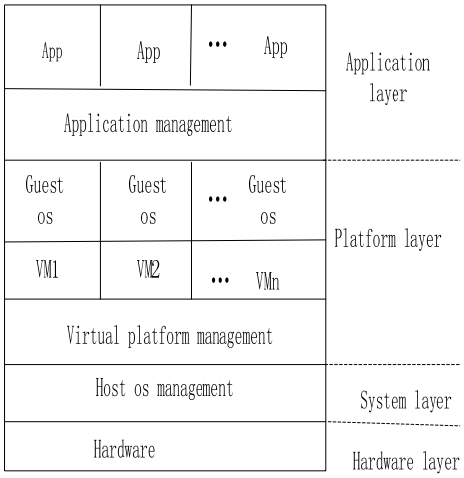


Figure 1. Architecture of cloud computing

As shown in Fig. 1, the cloud architecture can be divided into four layers: the hardware layer, the system layer, the platform layer and the application layer.

1) Hardware layer: include a variety of storage facilities, network facilities, computing facilities, software, host and server. Cloud computing abstracts the infrastructure by using virtualization technology, and formats the corresponding resource pool for the user to call and all this is transparent to users.

2) System layer: the second layer in the cloud architecture includes virtualized hosts and networks. One example for this layer is the Amazon EC2 service [2], which provides virtual hosts and network to the customers.

3) Platform layer: the third layer of the cloud architecture includes virtualized operating systems as well as runtimes and Application Program Interfaces (APIs). A famous example is the Platform Windows Azure [3], which provides the users with several APIs to storage and management.

4) Application layer: the top layer of the architecture provides virtual applications. Google App Engine [1] is a known infrastructure on this layer.

So as [10] described, the specific service which could be directly used by user deployed in each layer is provided by provider with the computing technique described specifically as follows: Grid Computing which is the virtualized combination of computing power from multiple domain getting high computing resource; Utility Computing that consumers pay for computing resources as much as they use without buying them, Server based Computing that any applications and data exist in server. Clients access the server and utilize them using server's computing power and so on.

The hardware layer which is not directly provided to users is not considered here. The system layer threats suffered easily are affected by the traditional

security methods. The user can simultaneously run on several virtual hosts which provide Linux-based and Windows-based web page and ftp file-sharing [11]. Take advantage of this opportunity, the activities of Internet users and pages viewed may be spied through cookies by some commercial companies. In addition, other computers could be controlled by using corpse programs to master what service could be used in cloud. The threats are suffered easily by platform layer especially the hacker attacks. It is also possible that the reliability of the system itself may not fully be guaranteed. The example of application layer suffering the attacks is as follows. Google Gmail service interrupted for up to four hours in February 2009 [12]. The failure is due to the data center in Europe makes another data center overload and spreads the data to other data centers when routine maintenance.

Visibly each layer of the cloud structure may be subjected to some different degrees of damage and attacks.

Another important secure threat of cloud computing is the concept of multi tenancy in VMs. Multi tenancy can be viewed as a hierarchical model, where appropriate policies are enforced on the VMs at every level leading to better governance and segmentation of the consumers. Enforcing different policies at different levels of hierarchy also leads to a secure environment for the consumers to store and access their confidential data. When the VMs are deployed on the physical server security threats always play a major role. Even during the everyday routine utilization there is always a way for the attackers to consolidate their VMs and gain control over the OS. VMs could be moved over from one host to another and has a major threat of being collapsed. While the VM is copied over the network, the state of the VM could be On, Off or suspended. In this research, different VMs need to be assigned to different users because of the possible security threats in a server virtualized environment.

In a word, the following aspects cloud-related securities need to be considered thoroughly such as credibility, reliability, confidentiality and privacy.

B. Virtualization

Virtualization technology has developed rapidly because of the rapid decrease in hardware cost and concurrent increase in hardware computing power. Several features of the virtualization are used in deploying multiple sensors in cloud environment process. In order to avoid a promised IDS sensor to be used to attack other sensors, the virtualization of IDS sensor is described in [13]. A variety of snort-based sensors in the format of plug could be used to effectively monitor virtualization components. The cloud provider wants to identify the components running in a virtual host as directly as running in their underlying architecture. It's necessary that the virtualization technology including VM state, VM work-platform and the monitoring of the configuration information of IDS in VM need to be joined together to integrate the cloud infrastructure with the virtual context e.g. [14] virtualized the system which makes the system hard to be promised, as well as with the self-reflection of memory .

C. DoS and DDoS Attack

In order to make the computer and network deny to serve the normal service by consuming bandwidth and hosting system resources, mining program defects and providing false DNS information. For example, network communication is blocked and access to service is denied, server crashes or service have been damaged. The denial service capability of DDoS is increased through depending on client and server technology with multiple computers together as an attack platform to launch DoS attack to one or more targets and generate more attacks traffic than DoS [15].

Two primary aspects about a typical DDoS attack are as follows:

1) Initial mass-intrusion: look for the puppets. That is to say, make the internet systems that easy to damage compromised, and then install attack tools in these vulnerable systems.

2) Denial of service attack: the target system will be paralyzed, once the puppets receive an attacking commend issued by the attacker through a secure channel.

Many research papers have give taxonomy on DoS and DDoS [16-20]. All of them have analyzed DoS and DDoS attack and defense in different perspectives.

III. Proposed Architecture

A. VM Management Unit

The security of IDS sensors which are endowed with specific VM component of different layer must be ensured by provider [21]. Network-based or host-based sensors of different layers in virtual environment are managed by VM management unit which is a part of the core management unit. The state of VM such as start, shutdown, stop, continue, reset or update and weather the VM is running, how its platform are involved in virtual environment. The attacks related to the virtual component could be recognized by the provider with the VM management unit. The attack also could be interrupted immediately by automatic counter-measures. Multiple IDS sensors are deployed in specific layer as shown in Fig. 2 by which each virtual component security is insured.

B. Collector

Alerts generated from multiple sensors are collected by the component. Then the alert with the format IDMEF [22] has been proposed as a standard to enable interoperability among different IDS approaches. IDMEF defines a unified format for communication between IDS sensors, response systems, and the core management unit. IDMEF-Message is the top-level class for all IDMEF-Messages. Every type of message belongs to the subclass of the top-level. Alerts and HeartBeats are mainly two types of message. Within each message, the detailed information is provided by the subclass of the message class. It specifies an Extensible Markup

Language (XML)-based data model to represent the exchanging data between IDSs [23]. The Extensible Markup Language (XML) is a simplified version of the Standard Generalized Markup Language (SGML), the syntax for specifying text markup defined by the ISO 8879 standard. XML is gaining widespread attention as a language for representing and exchanging documents and data on the Internet, and as the solution to most of the problems inherent in HyperText Markup Language (HTML). XML is met language-a language for describing other languages – that enables an application to define its own markup. XML allows the definition of customized markup languages for different types of documents and different applications. A message with the type of IDMEF message is the part of IDMEF library which is based IDMEF XML scheme by RFC [23]. However due to the highly heterogeneous architecture, especially VM-based new structure IDMEF is inevitable

C. Analyze & Compare and Threshold Computing

The alerts unified by the front component are passed to database to compare with the signatures. If the type of packet is correspondence with the one listed in the database, then the alert is considered to be anomaly and to be dropped. Otherwise the alert is continued passed to threshold computing. Thus, the time comparing is saved and the detection efficiency is promoted.

The threshold computing formula is described as follows:

$$\text{Threshold}=W+U*V\ . \tag{1}$$

W means the average of the same type alerts received during a specific time interval. U means the standard deviation. V is a constant dynamically determined by administrator. If the result computed is larger than the specific threshold, then the packet is considered to be anomaly. Last the alert forwards two directions: firstly, the rule of the alert is added into the database shown in Figure 2. Second in order to avoid the single point attack of failure, the alert is continued to be transmitted to communication agent communicating with other cloud region's communication agent. Reducing the communication overheads among IDSs and improving the accuracy of the alerts are the main priorities of the component. The reliability of the system will be improved owing to that false alerts and communication overheads caused by message exchanging are reduced remarkably.

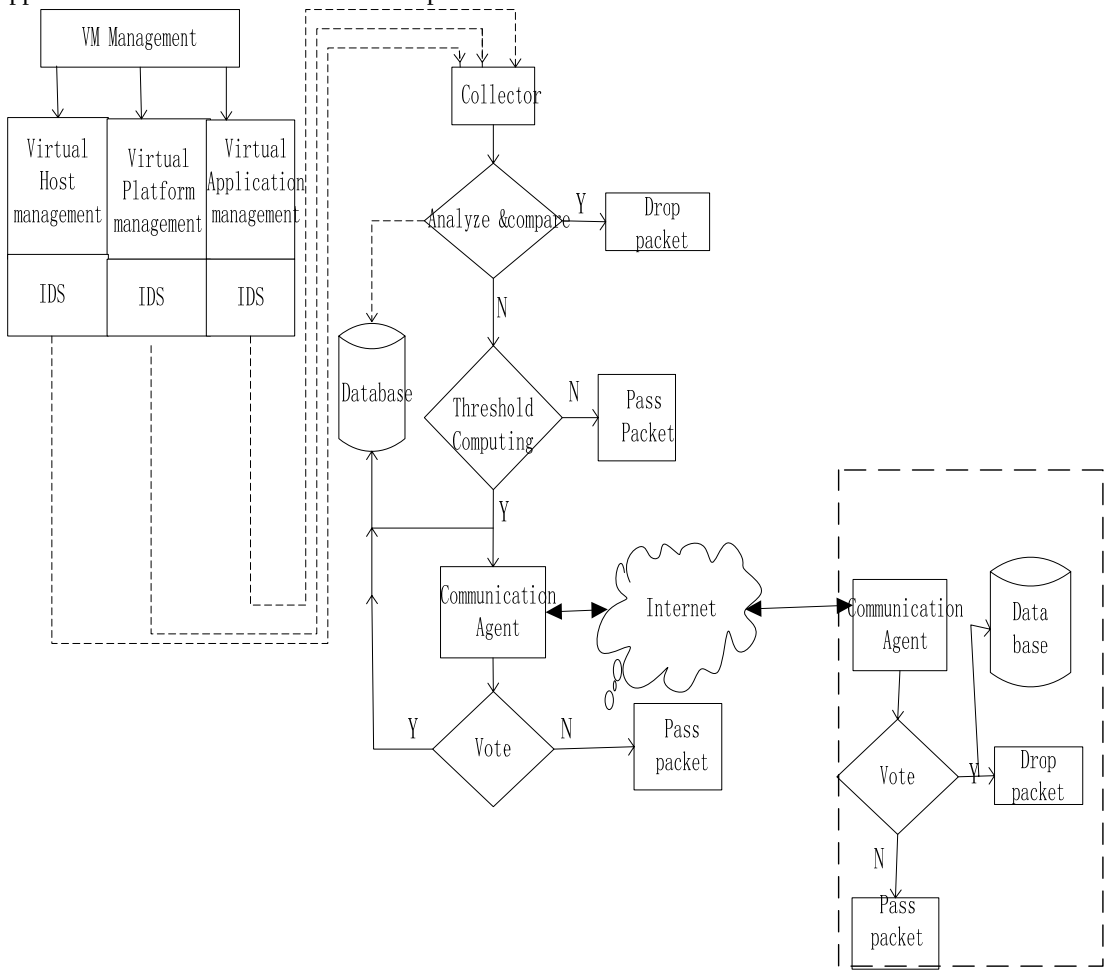


Figure 2. The core management unit

D. Certification Module

Before the alerts are exchanged among communication agents of different cloud regions through the internet, we introduce an improved Kerberos protocol certification [24] module as the third party in figure.3 to ensure the communication agents' true identities.

First of all, the key components of the module are described specifically. Followed by is the introduction of the entire workflow.

Certificate Authority (CA) which is the third party trusted organization provides information security service for the network agents and equipments. Authentication Server (AS) could confirm the agent's identity when the agent logs by sharing a key with each user. Ticket Granting Service (TGS) distributes the ticket for the communications among agents. So that the application server believes that the holder of TGS is as same as it claims.

The specific workflow of the module is as follows (the detailed description of (1), (2)….. in Fig. 3 is accordance with the followed *1), 2)* ……) :

1) The agent requests the key certificate of TGS for CA by sending its public key, agent's server name and the server name of TGS.

2) After checking the agent's legitimate identity, CA issues the public key certificate attached with the key certificate of the information server of agent's TGS and issuance time for agent and TGS, through the public key.

3) Each time when agent needs to use the service in application server, generate an authentication code containing the user name, time stamp and additional session key from the AS.

4) After receiving the request, TGS decrypts the license ticket with its own private key for attaining session key, decrypted Authentication Code (AC) and verifying the validity of the timestamp. While, TGS confirms the legitimacy of the agent, the license ticket containing the session key to agent whose responsibility is decrypting the new public key certificate with own private key to get the public key and session key of TGS.

5) Agent encrypts the AC by use of the session key, then re-encrypts with private key. At last, after agent using the server's public key to encrypt its own public key and ticket, it is sent to TGS server. So here, we achieve a two-factor authentication, that is to say, Kerberos's session key certification and CA certificate authentication.

6) After the TGS server decrypts the got information with private key to obtain the agent's public key, other information is also decrypted to verify the agent's identity. What's more, the TGS server return a ticket and time stamp encrypted with private key to agents. So that after the more secure recognition based on the two-factor authentication, the two sides can communicate through the secure channel.

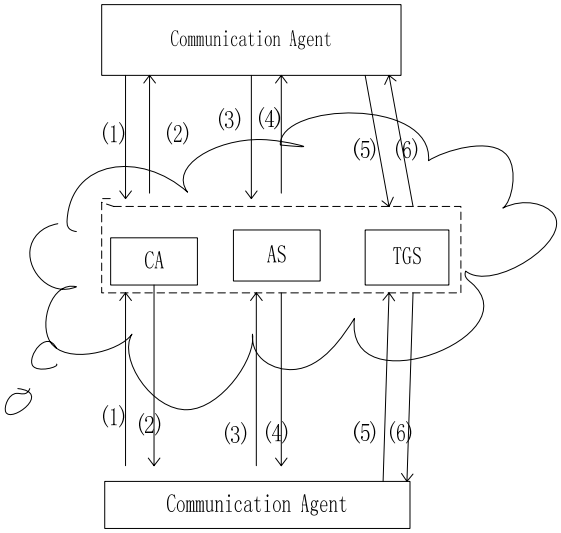


Figure 3. Certificate module

E. Communication Agent

In order to avoid the single point attack of failure, alert coming from the threshold component is passed to another cloud region's communication agent by the Internet. Then the received alert is further judged the reliability of alerts as the following formula:

$$\frac{\#\text{number of IDSs sends the same alert}}{\#\text{number of IDSs in the cloud}} > 0.5 . \qquad (2)$$

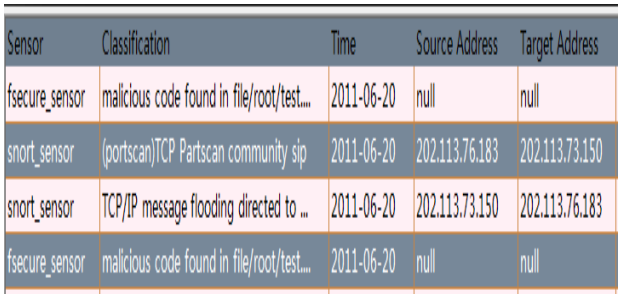
If the result is larger than 0.5, the packet is considered to be anomaly. Meanwhile its rule set is sent to database and the alert is passed to other cloud region communication agent.

So through the component we know that if the packet is passed the component frequently, the attack will be found and then this type of packet will be dropped. And the single point attack of failure will not be occurred.

IV. SIMULATION

To test the feasibility of the above mentioned architecture, we simulate the experiment. It consists of two servers I and II to simulate the two different cloud regions, at the same time, VM-based intrusion detection system is set up. Server-I with the IP address 202.113.73.150, executes two VMs including F-Secure sensor [25] and snort sensor and server-II with the IP address 202.113.73.183, executes three VMs including two Samhain sensors [26] and one snort sensor. The five different IDS sensors are separately used to supervise the local malicious or filter the network traffic. The basic configuration of each server is 2GB memory, 500GB hard disk and 2GHZ CPU and on-chip TPM v1.2. The third server as an attacker firstly launches attacks like TCP/IP packets, SYN flooding [27] to the servers in the format Network Mapper (NMAP) to scan the ports. We also compare with the NIDS system deployed in the cloud. Our measuring framework will incur additional, un-avoidable computing overhead. In cloud computing,

many applications run concurrently. Generally, the measuring framework would affect parallel programs more than sequential programs, so we tested parallel programs. The experiment results are as follows.



Sensor	Classification	Time	Source Address	Target Address
fsecure_sensor	malicious code found in file/root/test....	2011-06-20	null	null
snort_sensor	(portscan)TCP Partscan community sip	2011-06-20	202.113.76.183	202.113.73.150
snort_sensor	TCP/IP message flooding directed to ...	2011-06-20	202.113.73.150	202.113.76.183
fsecure_sensor	malicious code found in file/root/test....	2011-06-20	null	null

Figure 4. The simulation result

As shown in Fig. 4, the working state of the whole architecture can be viewed by user. The end user can also inspect each IDMEF alert directly for the detailed information of the alert. Further, the IDS and VM on this platform can also be detected by the end user.

TABLE I
THE DETECTION CONDITION OF SYN FLOODING

Simulation systems	Detection rate	False-positive rate	Negative rate
NIDS system	82%	0.8%	1.12%
Propose system	94.27%	0.55%	0.52%

The architecture's simulation results prove that it has a high detection rate, little false-positive rate and negative rate. It resists the failure of single point attack effectively.

V. FUTURE WORK

The realization of this proof-of-concept is only just beginning in the domain of the VM-based IDS system in the complicated cloud computing environment. First of all, the virtualization for alerts can be further regulated to support manual analysis. What is more, the next step needs to be researched is that the collect component needs to be upgraded to enable different approaches for analyze & compare component, such as labeling, filtering, classification. The other interesting research point is the possibility of implementing counter measures by use of VM management unit. It is possible that new correlation methods will also be needed, by taking the special allocation and related components of the cloud computing framework into consideration. Moreover, in order to truly realize the allocation, enforceability and scalability problems need further to be considered. So far, this frame-work has only been implemented in several physical machines. Because clouds involve multiple computers (servers) cooperating with each other, further research should focus on a measuring framework applicable to multiple physical machines followed by a number of unexpected problems.

VI. CONCLUSION

Considering the complexity of the cloud security architecture, a extensible VM-based multiple IDSs are deployed in each layer to monitor specific virtual component and the core management unit are constructed by multiple plugs satisfying the IDMEF standard to realize the ideas of virtualization and cloud alliance which is mainly used for avoiding the single point attack of failure. That is to say, the large scale attacks to several users may be detected easily by related and mutual alerts passed by communication agents. To further ensure the alerts could be sent to communication agent of another cloud region, the certificate module plays an important role here.

ACKNOWLEDGMENT

This work was supported by the Foundation of Tianjin for Science and Technology Innovation (No.10FDZDGX004 00&11ZCKFGX00900), Education Science and Technology Foundation of Tianjin (No.SB20080053 & SB20080055).

REFERENCE

[1] Google Apps, http://www.google.com/a.
[2] WindowsAzurePlatform,http://www.microsoft.com/azure/default.mspx.
[3] Amazon Elastic Compute Cloud and Simple Storage Service, http://aws.amazon.com.
[4] M.M. Yasin, A.A. Awan, "A study of host-based IDS using system calls," *Lahore, Pakistan*, vol. 3, pp. 36-41, June 2004
[5] A.K. Ganame, J. Bourgeois, R. Bidou, F. Spies, "A global security architecture for intrusion detection on computer networks," Montbeliard, vol. 27, pp.30-47, March 2008,
[6] Xiang Yang; Ke Li; Wanlei Zhou, " Low-rate DDoS attacks detection and traceback by using new information metrics," vol. 6 (2), pp.426-437,Jun 2011.
[7] K. Sunil, L. Jun-yong, "A system architecture for high-speed deep packet inspection in signature-based network intrusion prevention," vol. 53, pp. 310-320, May-June 2007,
[8] Q. Yan, W. Xie, B. Yan, et al. "An anomaly intrusion detection method based on HMM," vol. 38 (13), pp. 663-664, May 2006.
[9] Xiantao. Zhang, Qi. Li; Sihan. Qing, "Building an IDS architecture using VMM-based non-intrusive approach," *Adelaide, SA, Australia* , pp. 594-600, Jan. 2008.
[10] http://radlab.cd.berkley.edu/,UC Berkley Reliable Adaptive Distributed Systems Laboratory (accessed on Feb. 2009).
[11] P. Willmann, J. Shafer, D. Carr, A. Menon, S. Rixner, A. L. Cox, et al, "Concurrent direct network access for virtual machine monitors," *Lausann. Switzerland*, vol. 10-14, pp. 306-317, February 2007
[12] http://code.google.com/appengine/,Google (accessed on Oct 2009).
[13] G.W Dunlap, S.T King, S.Cinar, M.Basrai, " Enabling intrusion analysis through virtual-machine logging and replay," *Boston, MA, USA*, pp. 211-240, December 2002.
[14] F. Cheng, S. Roschke, Ch. Meinel, "Implementing IDS management Lock-Keeper (ISPEC'09)," *Xi'an, China*, pp. 360-371, April 2009

[15] S. Ranjan, R. Swaminathan, M. Uysal, et al., "DDoS-shield: DDoS-resilient scheduling to counter application layer atttacks," vol. 17 (1) , pp. 26-39, Feb. 2009,
[16] M. Jelena, R. Peter, "A taxonomy of DDoS attack and DDoS defense mechanisms," vol. 34(2), pp. 39-53, 2004
[17] A. Hussain, J. Heidemann, C. Papadopoulos, "A framework for classifying denial of service attacks," vol.33 (4), pp. 99-110, 2003
[18] Alex Wun, Alex Cheung, Hans-Arno Jacobsen, "A taxonomy for denial of service attacks in content-based publish/subscribe systems," *Toronto, Ontario, Canada*, pp. 116-127, 2007
[19] T. Usman, H. ManPyo, and L. Kyung-suk, "A comprehensive categorization of DDoS attack and DDoS defense techniques," *Berlin*, vol.4093, pp. 1025-1036, 2006
[20] G. Michael, "A summary of DoS/DDoS prevention, monitoring and mitigation techniques in a service provider environment," 2003.
[21] Qian. Liu , Chuliang. Weng , Minglu. Li , Yuan. Luo, "An In-VM measuring framework for increasing virtual machine security in clouds," vol. 8 , pp. 56 - 62 , Nov.-Dec, 2010
[22] H. Debar, D. Curry, B. Feinstein, "The intrusion detection message exchange format," *Korea*, pp. 193-199, July 2004.
[23] M. Kudo, S. Hada. "XML document security based on provisional authorization," vol. 33(4), pp. 379-389, 2004.
[24] J.T. Kohl. "The evolution of the Kerberos authentication service," *Tromso, Norway*, pp. 295-313, 2009.
[25] M. Komssi, M. Kauppinen, J. Heiskari, M. Ropponen, "Transforming a software product company into a service business: Case study at F-Secure issue," vol. 1, pp. 61 - 66 , July 2009
[26] http://www.f-secure.com/linux-weblog/ F-Secure Corporation (accessed Oct 2009).
[27] D. Moore, C. Shannon, D.J. Brown, and et al. "Inferring internet denial-of-service activity," *California, San Diego*, vol. 24(2), pp.115-139, 2006.