International Journal of scientific research and management (IJSRM)
||Volume||2||Issue||4||Pages||732-737||2013||
Website: www.ijsrm.in ISSN (e): 2321-3418

A MODIFIED NICE MECHANISM WITH DISTRIBUTED HOST-BASED IDS (D-HIDS) SYSTEM

J.Ramya[1], G.Sivakumar[2]

[1]PG Student,
Affiliated to Anna University Chennai, Dept. of Computer Science and Engineering
Gnanamani College of Engineering,
Namakkal, India
ramyajaganathan17@gmail.com

[2]Asst. Prof Dept. of CSE
Dept. of Computer Science and Engineering
Gnanamani College of Engineering,
Namakkal, India
sivagce05@gmail.com

Abstract:*In the past few years Cloud security is a challenging issue that has attracted a lot of research and development effort. Particularly, attackers can explore vulnerabilities and compromise virtual machines to deploy further large-scale Distributed Denial-of-Service (DDoS). Within the cloud system, the detection of zombie exploration attacks is extremely difficult. This is because cloud users may install vulnerable applications on their virtual machines. The existing NICE Method can detect and measure the vulnerabilities, and can counter measure the attacks, but the scalability is very less and there is huge performance degradation in intrusion detections at the Host. To overcome the above problems A Modified NICE Mechanism with a Distributed Host-Based Ids (D-Hids) System is proposed. This D-HIDS enhance the overall accuracy of intrusion assessment as well as the ability of detecting new classes of intrusions. In addition to that, this A new decentralized access control scheme is introduced to make our design still stronger to access in a decentralized manner.*

Keywords:cloud computing, security, virtual machine, dos, intrusion detection.

1. INTRODUCTION

A Cloud Security Alliance (CSA) survey shows that among all security issues, abuse and nefarious use of cloud computing is considered as the top security threat, in which attackers can exploit vulnerabilities in clouds and utilize cloud system resources to deploy attacks. In traditional data centers, where system administrators have full control over the host machines, vulnerabilities can be detected and patched by the system administrator in a centralized manner. However, patching known security holes in cloud data centers, where cloud users usually have the privilege to control software installed on their managed VMs, may not work effectively and can violate the service-level agreement (SLA). Furthermore, cloud users can install vulnerable software on their VMs, which essentially contributes to loopholes in cloud security. The challenge is to establish an effective vulnerability/attack detection and response system for accurately identifying attacks and minimizing the impact of security breach to cloud users.

Here addressed that protecting "Business continuity and services availability" from service outages is one of the top concerns in cloud computing systems. In a cloud system, where the infrastructure will be shared by Potentially millions of users, abuse and nefarious use of the shared infrastructure benefits attackers to exploit vulnerabilities of the cloud and use its resource to deploy attacks in more efficient ways. Such attacks are more effective in the cloud environment because cloud users usually share computing resources, e.g., being connected through the same switch, sharing with the same data storage and file systems, even with potential attackers. The similar setup for VMs in the cloud, e.g., virtualization techniques, VM OS, installed vulnerable software, networking, and so on, attracts attackers to compromise multiple VMs.

Computer security aims to protect an organization valuable's resources from unauthorized access, tampering of data, and denial of service. One broad definition of a secure computer system is that "the one that can be depended upon to behave as it is expected to". This concept can be referred to as

trust, a system can be trusted if it can preserve and protect its data.

Nowadays, most intruders have become skilled at determining and exploiting systems' weaknesses to increase privileges of systems' attacks. Damaging intrusions can occur in a matter of seconds by overcoming the password authentication mechanisms designed to protect systems. Moreover, intruders can hide their presence by installing modified versions of system monitoring and administration commands and by erasing their tracks in audit logs. The knowledge required by intruders to launch known methods of attacks is decreasing. Today anyone can attack a network due to the widespread and easy availability of intrusion tools .Hence a High performance Collaborative multiphase detection system is a need for securing the cloud environment as well as the virtual machine environment.

2. Related Works

IDSs in cloud produce alerts for the administrators which are based on true positives or true alarms when actually intrusion takes place and false positive or false alarms in case of a wrong detection by the system. IDSs can detect intrusion patterns by critically inspecting the network packets, applying signatures (pre-defined rules) and generating alarms for system administrators. IDS uses two method of detection i.e. anomaly detection, that works on user behaviour patterns and suspicious behaviour. Other method is misuse detection that can detect through renowned attack patterns and matching a set of defined rules or attack against system vulnerabilities through port scanning. Since Cloud infrastructure has enormous network traffic, the traditional IDSs are not efficient enough to handle such a large data flow. Most known IDSs are single threaded and due to rich dataset flow, there is a need of multi-threaded IDS in Cloud computing environment. In a traditional network, IDS monitors, detects and alert the administrative user for network traffic by deploying IDS on key network choke points on user site. But in Cloud network IDS has to be placed at Cloud server site and entirely administered and managed by the service provider. In this scenario, if an attacker manages to penetrate and damage or steal user's data, the cloud user will not be notified directly. The intrusion data would only be communicated through the service provider and user has to rely on him. The cloud service provider may not like to inform the user about the loss and can hide the information for the sake of his image and repute.

Even though some improvements were made recently in the Cloud Environment, current Intrusion Detection Systems (IDS) propose few response mechanisms in addition to alerts and reports. There is only a small variety of response techniques and the decision criteria that are used to activate the response remain often simplistic. Moreover, in a context of exploitation, security administrators generally balk at using the most interesting responses like automatic reconfiguration of firewalls or routers. This is due to lack of confidence in the capabilities of the IDS to take the right decision. Administrators also fear of not controlling the consequences of the automation of counter-measures. Hence there is a need of higher efficient Collaborative Multi Phase Intrusion detection to maintain the cloud environment secure.

3. NICE Mechanism

A Modified NICE Mechanism with a Distributed Host-Based Ids (D-Hids) System is proposed. With the functions of NICE, This D-HIDS enhance the overall accuracy of intrusion assessment as well as the ability of detecting new classes of intrusions. It selects and maintains a list of collaborators from which they can consult about intrusions. Specifically, evaluates both the false positive (FP) rate and false negative (FN) rate of its neighbouring D-HIDSes opinions about intrusions, and aggregates these opinions using a Bayesian decision model. In addition to that, this A new decentralized access control scheme is introduced to make our design still stronger to access in a decentralized manner.Given Fig.1 shows the NICE Mechanism.

The beneficial in NICE methods are:

- It can detect, Measure the vulnerabilities and can select the counter measures to mitigate the DDOS attacks.
- The D-HIDS enhance the overall accuracy of intrusion assessment as well as the ability of detecting new classes of intrusions in distributed hosts.
- Improved Scalability is achieved.
- This can be accessed with decentralized access control.

3.1 Collabarative Intrusion Detection System Cofiguration

The configurations for Dynamic Host-based Intrusion Detection Systems (HIDSes) identify intrusions by comparing observable intrusion data such as log files &computer activities against suspicious patterns is done. A CIDN acquaintance management is the process of identifying, selecting, and maintaining collaborators for each HIDS a Bayesian learning technique that helps each HIDS to identify dishonest collaborators and remove them from its collaborator list.

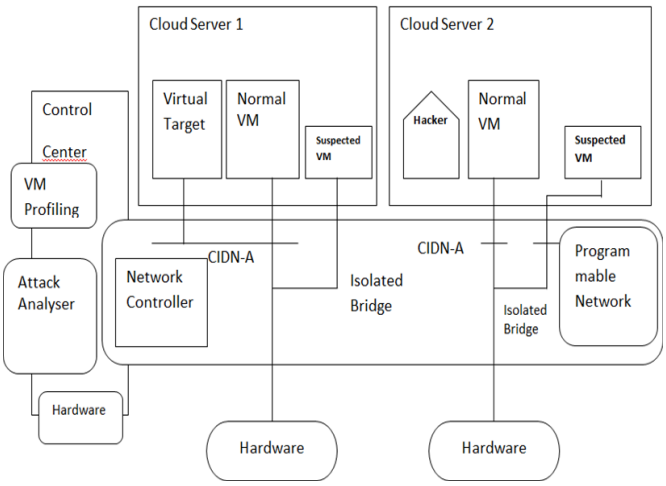

Fig 1: System Architecture

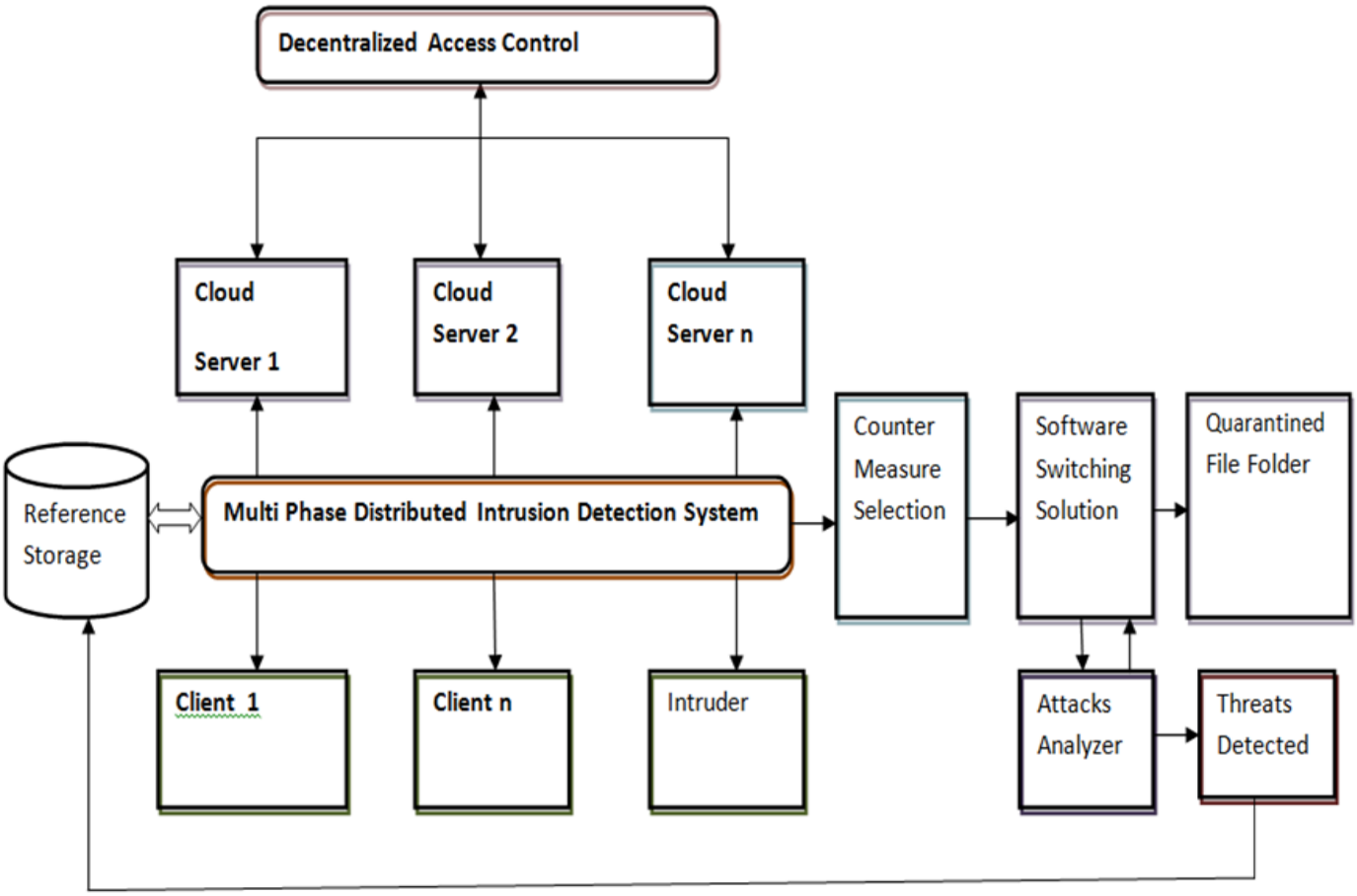

Fig 2: Workflow of Multi phase Distributed Intrusion Detection System

3.2 Multiphase Distributed Network Intrusion Detection

A new multiphase distributed network intrusion detection and prevention framework in a virtual networking environment that captures and inspects suspicious cloud traffic without interrupting users' applications and cloud services.

3.3 Decentralized Access Control

The cloud verifies the authenticity of the user without knowing the user's identity before storing data. Our scheme also has the added feature of access control in which only valid users are able to decrypt the stored information. The scheme prevents replay attacks and supports creation, modification, and reading data stored in the cloud.

3.4 SOFTWARE SWITCHING SOLUTION

This will incorporates a software switching solution to quarantine and inspect suspicious VMs for further investigation and protection. However the programmable network approaches. This method can improve the attack detection probability and improve the resiliency to VM exploitation attack without interrupting existing normal cloud services.

4. System Components

Fig.2 states the Workflow of Multi Phase Distributed Intrusion Detection System. In this section, we explain each component of NICE.

4.1 NICE-A

The NICE-A is a Network-based Intrusion Detection System (NIDS) agent installed in either Dom0 or DomU in each cloud server. It scans the traffic going through Linux bridges that control all the traffic among VMs and in/out from the physical cloud servers. In our experiment, Snort is used to implement NICE-A in Dom0. The traffic generated from the VMs on the mirrored software bridge will be mirrored to a specific port on a specific bridge using SPAN, RSPAN, or ERSPAN methods. The NICE-A sniffing rules have been custom defined to suite our needs. Dom0 in the Xen environment is a privilege domain, that includes a virtual switch for traffic switching among VMs and network drivers for physical network interface of the cloud server. It is more efficient to scan the traffic in Dom0 because all

traffic in the cloud server needs go through it; however, our design is independent to the installed VM. In the performance

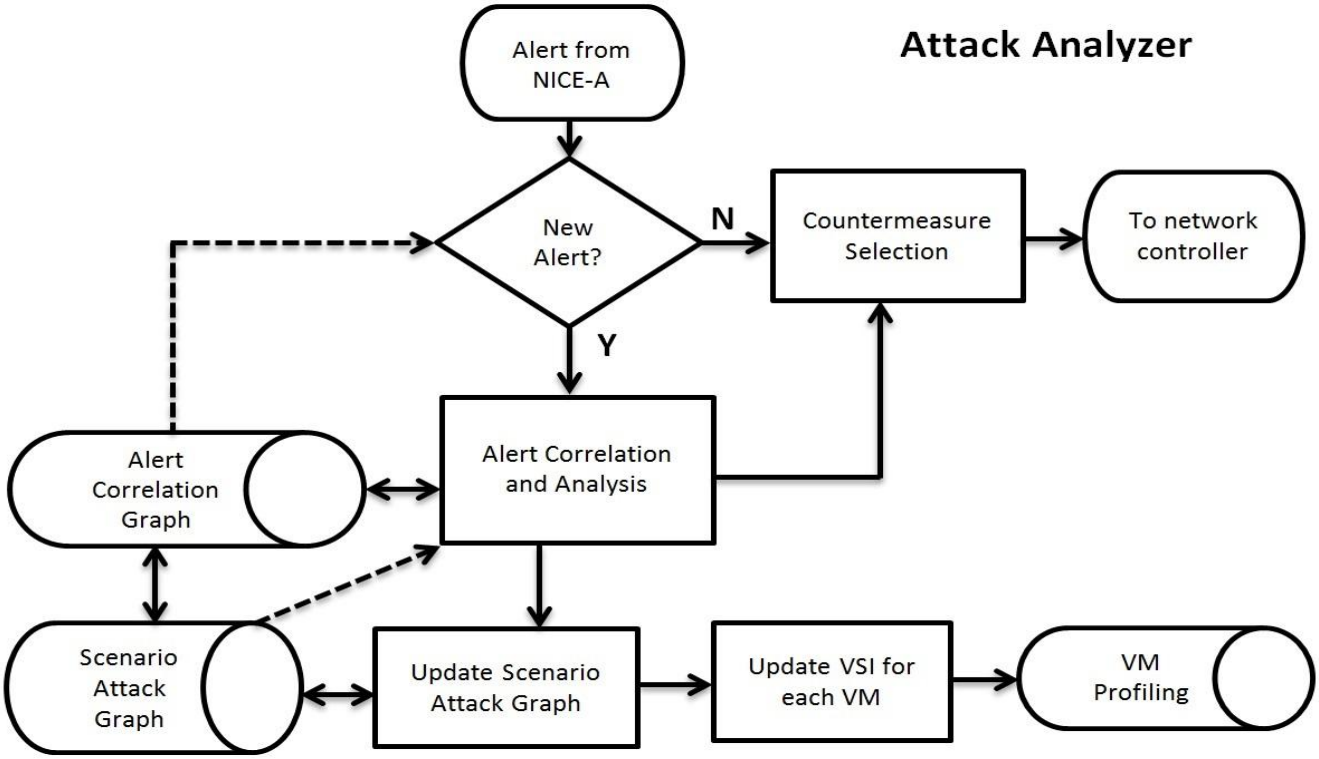

Fig 3: Progress of Attack Analyzer

evaluation section, we will demonstrate the tradeoffs of installing NICE-A in Dom0 and DomU. We must note that the alert detection quality of NICE-A depends on the implementation of NICE-A, which uses Snort. We do not focus on the detection accuracy of Snort in this paper. Thus, the individual alert detection's false alarm rate does not change. However, the false alarm rate could be reduced through our architecture design. We will discuss more about this issue in the later section.

4.2 VM Profiling

Virtual machines in the cloud can be profiled to get precise information about their state, services running, open ports, and so on. One major factor that counts toward a VM profile is its connectivity with other VMs. Any VM that is connected to more number of machines is more crucial than the one connected to fewer VMs because the effect of compromise of a highly connected VM can cause more damage. Also required is the knowledge of services running on a VM so as to verify the authenticity of alerts pertaining to that VM. An attacker can use port-scanning program to perform an intense examination of the network to look for open ports on any VM. So information about any open ports on a VM and the history of opened ports plays a significant role in determining how vulnerable the VM is. All these factors combined will form the VM profile.

VM profiles are maintained in a database and contain comprehensive information about vulnerabilities, alert, and traffic. The data comes from:

- Attack graph generator. While generating the attack graph, every detected vulnerability is added to its corresponding VM entry in the database.
- NICE-A. The alert involving the VM will be recorded in the VM profile database.
- Network controller. The traffic patterns involving the VM are based on five tuples (source MAC address, destination MAC address, source IP address, destination IP address, protocol). We can have traffic pattern, where packets emanate from a single IP and are delivered to multiple destination IP addresses, and vice versa.

4.3 Attack Analyzer

The major functions of NICE system are performed by attack analyzer, which includes procedures such as attack graph construction and update, alert correlation, and countermeasure selection. The process of constructing and utilizing the SAG consists of three phases: Information gathering, attack graph construction, and potential exploit path analysis. With this information, attack paths can be modeled using SAG. Each node in the attack graph represents an exploit by the attacker. Each path from an initial node to a goal node represents a successful attack. Fig.3 shows the overall progress of attack analyzer.

In summary, NICE attack graph is constructed based on the following information:

- Cloud system information is collected from the node controller (i.e., Dom0 in XenServer). The information includes the number of VMs in the cloud server, running services on each VM, and VM's Virtual Interface (VIF) information.

- Virtual network topology and configuration information is collected from the network controller, which includesvirtual network topology, host connectivity, VM connectivity, every VM's IP address, MAC address, port information, and traffic flow information.

- Vulnerability information is generated by both on demand vulnerability scanning (i.e., initiated by the network controller and NICE-A) and regular penetration testing using the well-known vulnerability databases, such as Open Source Vulnerability Database (OSVDB), Common Vulnerabilities and Exposures List (CVE), and NIST National Vulnerability Database (NVD).

The attack analyzer also handles alert correlation and analysis operations. This component has two major functions: 1) constructs ACG, and 2) provides threat information and appropriate countermeasures to network controller for virtual network reconfiguration.

Fig. 3 shows the workflow in the attack analyzer component. After receiving an alert from NICE-A, alert analyzer matches the alert in the ACG. If the alert already exists in the graph and it is a known attack (i.e., matching the attack signature), the attack analyzer performs countermeasure selection procedure according to the algorithm described in Section 5.3. And then notifies network controller immediately to deploy countermeasure or mitigation actions. The higher this value is, more packets this agent can handle. If the alert is new, attack analyzer will perform alert correlation and analysis according to Algorithm 1, and updates ACG and SAG. This algorithm correlates each new alert to a matching alert correlation set (i.e., in the same attack scenario). A selected countermeasure is applied by the network controller based on the severity of evaluation results. If the alert is a new vulnerability and is not present in the NICE attack graph, the attack analyzer adds it to attack graph and then reconstructs it.

4.4 Network Controller

The network controller is a key component to support the programmable networking capability to realize the virtual network reconfiguration feature based on OpenFlow protocol. In NICE, within each cloud server there is a software switch, for example, OVS [5], which is used as the edge switch for VMs to handle traffic in and out from VMs. The communication between cloud servers (i.e., physical servers) is handled by physical OpenFlow-capable Switch (OFS). In NICE, we integrated the control functions for both OVS and OFS into the network controller that allows the cloud system to set security/filtering rules in an integrated and comprehensive manner. The higher this value is, more packets this agent can handle. The network controller is responsible for collecting network information of current OpenFlow network and provides input to the attack analyzer to construct attack graphs. Through the cloud internal discovery modules that use DNS, DHCP, LLDP, and flow initiations, network controller is able to discover the network connectivity information from OVS and OFS. This information includes current data paths on each switch and detailed flow information associated with these paths, such as TCP/IP and MAC header. The network flow and topology change information will be automatically sent to the controller and then delivered to attack analyzer to reconstruct attack graphs.

Another important function of the network controller is to assist the attack analyzer module. According to the OpenFlow protocol, when the controller receives the first packet of a flow, it holds the packet and checks the flow table for complying traffic policies. In NICE, the network control also consults with the attack analyzer for the flow access control by setting up the filtering rules on the corresponding OVS and OFS. For prevent vulnerable VMs from being compromised and to do so in less intrusive and cost effective manner.Once a traffic flow is admitted, the following packets of the flow are not handled by the network controller, but monitored by the NICE-A.

Network controller is also responsible for applying the countermeasure from attack analyzer. Based on VM Security Index (VSI) and severity of an alert, countermeasures are selected by NICE and executed by the network controller. If a severe alert is triggered and identifies some known attacks, or a VM is detected as a zombie, the network controller will block the VM immediately. An alert with medium threat level is triggered by a suspicious compromised VM. The higher this value is, more packets this agent can handle. Countermeasure in such case is to put the suspicious VM with exploited state into quarantine mode and redirect all its flows to NICE-A DPI mode. An alert with a minor threat level can be generated due to the presence of a vulnerable VM. For this case, to intercept the VM's normal traffic, suspicious traffic to/from the VM will be put into inspection mode, in which actions such as restricting its flow bandwidth and changing network configurations will be taken to force the attack exploration behavior to stand out.

5. CONCLUSION

CIDN is proposed to detect and mitigate collaborative attacks in the cloud virtual networking environment. CIDN utilizes the attack graph model to conduct attack detection and prediction. The proposed solution investigates how to use the programmability of software switches based solutions to improve the detection accuracy and defeat victim exploitation phases of collaborative attacks. The system performance evaluation demonstrates the feasibility of CIDN and shows that the proposed solution can significantly reduce the risk of the cloud system from being exploited and abused by internal and external attackers. CIDN only investigates the network IDS approach to counter zombie explorative attacks. In order to improve the detection accuracy, host-based IDS solutions are needed to be incorporated and to cover the whole spectrum of

IDS in the cloud system. This should be investigated in the future work. Additionally, as indicated in the paper, it willsinvestigate the scalability of the proposed CIDN solution by investigating the decentralized network control and attack analysis model based on current study.

6. REFERENCES

[1] S. H. Ahmadinejad, S. Jalili, and M. Abadi, "A hybrid model for correlating alerts of known and unknown attack scenarios and updating attack graphs," Computer Networks, vol. 55, no. 9, pp. 2221–2240, Jun. 2011.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Commun. ACM, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[3] Z. Duan, P. Chen, F. Sanchez, Y. Dong, M. Stephenson, and J. Barker "Open vSwitch project," http://openvswitch.org, May 2012. [6], "Detecting spam zombies by monitoring outgoing messages," Dependable and Secure Computing, IEEE Transactions on, vol. 9, no. 2, pp. 198 –210, Apr. 2012.

[4] S. Govindavajhala, X. Ou, and A. W. Appel, "MulVAL: a logicbased network security analyzer," in Proceedings of the 14th conference on USENIX Security Symposium - Volume 14. Berkeley, CA, USA: USENIX Association, 2005, pp. 8–8.

[5] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "BotHunter: detecting malware infection through IDS-driven dialog correlation," in Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium, ser. SS'07. Berkeley, CA, USA: USENIX Association, 2007, pp. 12:1–12:16.

[6] G. Gu, J. Zhang, and W. Lee, "BotSniffer: detecting botnet command and control channels in network traffic," in Proceedings of 15th Ann. Network and Distributed Sytem Security Symposium, ser. NDSS'08, 2008.

[7] B. Joshi, A. Vijayan, and B. Joshi, "Securing cloud computing environment against DDoS attacks," in Computer Communication and Informatics (ICCCI), 2012 International Conference on, Jan. 2012, pp. 1 –5.

[8] R. Sadoddin and A. Ghorbani, "Alert correlation survey: framework and techniques," in Proceedings of the 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services, ser. PST '06. New York, NY, USA: ACM, 2006, pp. 37:1–37:10.

[9] Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," in 2002 IEEE Symposium on Security and Privacy, 2002. Proceedings.IEEE, 2002, pp. 273– 284. "NuSMV: A new symbolic model checker," http://afrodite.itc.it: 1024/_nusmv.

[10] H. Takabi, J. B. Joshi, and G. Ahn, "Security and privacy challenges in cloud computing environments," IEEE Security & Privacy, vol. 8, no. 6, pp. 24–31, Dec. 2010.

[11] L. Wang, A. Liu, and S. Jajodia, "Using attack graphs for correlating, hypothesizing, and predicting intrusion alerts," Computer Communications, vol. 29, no. 15, pp. 2917–2933, Sep. 2006.