IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 3, Ver. IX (May-Jun. 2014), PP 41-47
www.iosrjournals.org

Network Intrusion Detection in Virtual Network Systems and Countermeasure Selection (NIDCS)

[1]Mallamma C G, [2]Meera J, [3]Sujata Ramesh P
[1,2,3]*Asst Prof, Dept of CSE, Sambhram Institute of Tech,Bangalore,India*

***Abstract*:** *Cloud computing provides shared resources to various cloud users. All the users share various computing resources e.g., being connected through the same switch, sharing the same data storage and the file systems. Hence the cloud security is the major concern in the cloud computing and has attracted lot of research activities. The most common issue with the cloud computing is the Distributed-Denial-of-Service (DDoS) attacks. The DDoS attacks involve actions such as multistep exploitation and compromising identified vulnerable virtual machines as zombies. In the Infrastructure-as-a-service (IaaS) model, the detection of zombie virtual machines is difficult as the user may install vulnerable applications in their own Virtual machines. To detect and prevent the virtual machines from becoming zombie, we propose Defense-in-depth intrusion detection framework called NIDCS, which is an attack graph based analytical model.*

Index Terms: *Network security, cloud computing, attack graph, intrusion detection, zombie detection.*

I. Introduction

Cloud computing presents a new way to supplement the current consumption and delivery model for IT services based on the Internet, by providing for dynamically scalable and often virtualized resources as a service over the Internet. The end user of a service running "in the cloud" is unaware of how the infrastructure is architected-it just works. The provider of that service is able to dynamically provision infrastructure to meet the current demand by leasing resources from a hosting company. The cloud provider can leverage economies of scale to provide dynamic, on-demand, infrastructure at a favorable cost.To date, there are a number of notable commercial and individual cloud computing service providers, including Amazon, Google, Microsoft, Yahoo, and Sale force. Examples of cloud services include online file storage, social networking sites, webmail, and online business applications. The cloud computing scenario is as shown in the Figure 1.

Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services. The cloud also focuses on maximizing the effectiveness of the shared resources. Cloud

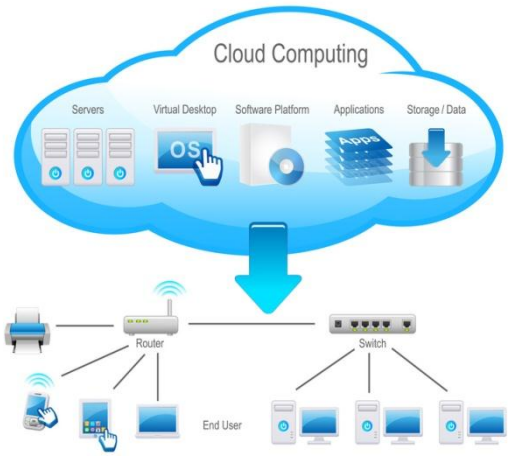

Figure 1: Cloud Computing

resources are usually not only shared by multiple users but are also dynamically reallocated per demand. This can work for allocating resources to users.

The survey conducted by Cloud Security Alliance (CSA) survey shows that among all security issues, abuse and hazardous use of cloud computing is considered as the top most security threat [1] 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 virtual machines may not work effectively and can violate the service level agreement (SLA).

II. Problem Statement

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. In a cloud system where the infrastructure is 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 [2]. Such attacks are more effective in the cloud environment since cloud users usually share computing resources, e.g., beingconnected through the same switch, sharing with the samedata storage and file systems, even with potential attackers [3].The similar setup for VMs in the cloud, e.g., virtualizationtechniques, VM OS, installed vulnerable software,networking, etc., attracts attackers to compromise multipleVMs.

Hear, we propose NIDCS (Network Intrusion detection and Countermeasure selection in virtual network systems) to establish a defense-in-depth intrusion detection framework. For better attack detection, NIDCS incorporates attack graph analytical procedures into the intrusion detection processes. We must note that the design of NIDCS does not intend to improve any of the existing intrusion detection algorithms; indeed, NIDCS employs are configurable virtual networking approach to detect and counter the attempts to compromise VMs, thus preventing zombie VMs.

III. System Architecture

In general, NIDCS includes two main phases:

- Deploy a lightweight mirroring-based network intrusion detection agent (NIDCS-A) on each cloud server to capture and analyze cloud traffic. A NIDCS-A periodically scans the virtual system vulnerabilities within a cloud server to establish Scenario Attack Graph (SAGs), and then based on the severity of identified vulnerability toward the collaborative attack goals, NIDCS will decide whether or not to put a VM in network inspection state.
- Once a VM enters inspection state, Deep Packet Inspection (DPI) is applied, and/or virtual network reconfigurations can be deployed to the inspecting VM to make the potential attack behaviors prominent.

The proposed NIDCS framework is illustrated in Figure 2. It shows the NIDCS framework with in one cloud server cluster. Major components in this framework are distributed and light- weighted NIDCS-A on each physical cloud server, a network controller, a VM profiling server,and an attack analyzer. The latter three components are located in a centralized control center connected to software switches on each cloud server (i.e., virtual switches built on one or multiple Linux software bridges).NIDCS-A is a software agent implemented in each cloud server connected to the control center through a dedicated and isolated secure channel, which is separated from the normal data packets using Open Flow tunneling or VLAN approaches. The network controller is responsible for deploying attack counter measures based on decisions made by the attack analyzer.
NIDCS-A is a network intrusion detection engine that can be installed in either Dom0 or DomU of a XEN cloud server to capture and filter malicious traffic. Intrusion detection alerts are sent to control center when suspicious or anomalous traffic is detected. After receiving an alert, attack analyzer evaluates the severity of the alert based on the attack graph, decides what counter measure strategies to take, and then initiates it through the network controller. An attack graph is established according to the vulnerability information derived from both offline and real-time vulnerability scans.

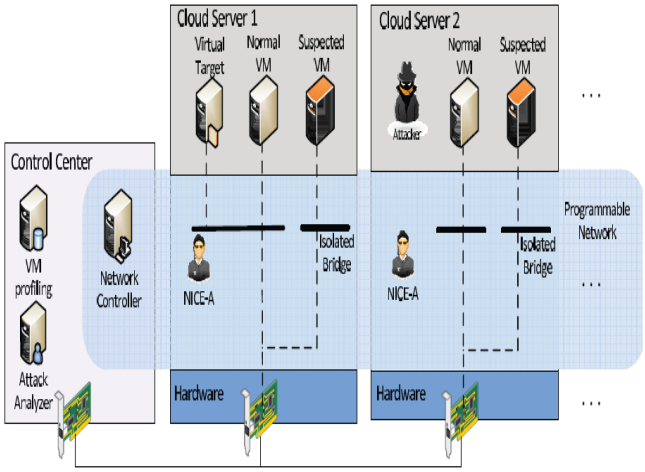

Figure 2: NIDCS architecture within one cloud server cluster.

The flow of the complete system is illustrated through the data flow diagram in Figure 3.

1) Create cloud server and allow virtual machine to join

- Create a cloud server.
- After creating cloud server add virtual machine to cloud server.
- A virtual machine is an independent operating environment which uses virtual resources.
- The virtual machine will fetch the details like free physical memory, cpu load, total physical memory, free swap space, & cpu time.
- After fetching the details the virtual machine will send them to cloud server using IP address and port number of server.
- It will wait for server response.
- After the server response, it will add the virtual machine into it.

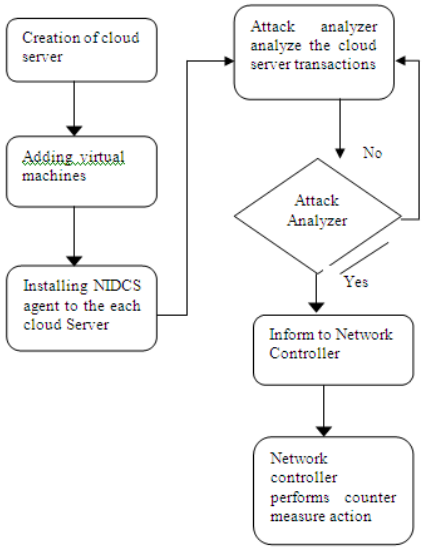

Figure 3: Flow Diagram

2) **Deploy an agent(NIDCS-A) on each cloud server**

- The NIDCS-A is a network based intrusion detection system(NIDS) agent installed in each cloud server.
- In order to scan and examine the traffic among virtual machines, introduce Network Controller.
- Network Controller will capture the packets.
- Stores the packets in the database.

3) Analyzing and description VM in the inspection state.

- In this stage we will inspect IP's based on the packet size and duration.
- Packet size is considered as one of the important parameter during communication.

- Packet size is calculated using formula

$$\textbf{Packet threshold size} = \frac{\textbf{total packets in all transfers}}{\textbf{number of transfers}}$$

- The duration also plays a role to inspect IP address. Which is calculated as,
-
- $\textbf{Duration threshold value} = \frac{\textbf{sum of all duration in all transfers}}{\textbf{number of transfers}}$

- The packets whose size is greater than **packet threshold size** and packets whose duration is greater
- than **duration threshold value** is fetched from the database. Suspect the fetched packet to be vulnerable.

4) Scanning the virtual system vulnerabilities with in a cloud server

- In this stage we construct an attack graph using the packets that we have captured using the network controller.
- We first fetch the captured packets that are stored into the database.
- In our attack graph the source and destination IPs are represented as the nodes and the edges represent the communication between the source and the destination IP. The attack graph is as shown in Figure 4.
- Then we fetch distinct source IPs and display them using the drawOval( ) specified under the graphics class.
- Then we fetch the Destination IPs and display them using the DrawOval( ) method which is specified in the graphics class.
- Then we draw a path between the source IP and the destination IP between which communications have happened.
- We use the DrawLine( ) method to draw an edge between the source IP and the destination IP.
- We then determine for each pair of source and destination IP, the number of communications that have happened and put it in the counter.
- We then examine which pair of source and Destination IP had maximum transfer by analyzing its counter value and represent that transfer between the IPs using a RED line.
- We then create a new test scenario where we generate a random number of requests and display the random communications from the captured packet based on random selection from the database.

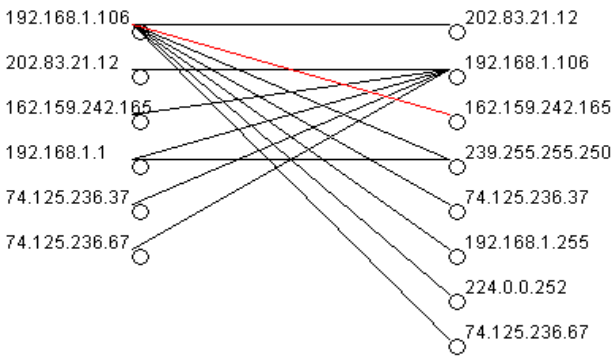

Figure 4: Attack graph

- Then again analyze the captured packets by creating another graph for all those packets whose packet size is greater than the packet size threshold value computed during the phase 3.
- We examine the IPs from this new graph and see whether they were alerts which were notified in the old graph or are a new alert which are discovered .in this graph.
- Then we perform metric measurement in the final phase.

5) Performing the counter measure actions to protect the cloud server

- In this phase we will determine the severity of the alert by classifying them into stable, vulnerable and exploited.
- In order to classify them into the following categories we compute the average value for each packet using the formula

$$packet\ average\ value = \frac{the\ individual\ packet\ size}{total\ packet\ size} * 100$$

- We compare this packet average value of each communication with the threshold value.
- We have two threshold values. One is the lower bound threshold value which is 0.7 and the other threshold value is the upper bound threshold value which is 3.0 .

- For all the communication,
- Where packet average value < 0.7, we classify those communications as stable.
- If (0.7 ≤ packet average value ≤ 3.0) we classify those communications as vulnerable.
- If the packet average value > 3.0 we classify those communications as exploited.
- Then all the captured communications are displayed along with their classifications.
- Then we compute the risk probability value and the ROI value.
- The Risk probability is calculated using the below formula,

  Risk probability= 1 – packet average value for each communication.
- Lesser the value of the risk probability lesser will be the vulnerability for that particular communication.
- We then compute the ROI (Return of Investment) value for each communication.
- The ROI is calculated using the formula,

$$ROI = \frac{Benefit}{cost + Intrusiveness}$$

- We require the overall average value for entire communication. That is given by the formula,

Overall average value= sum of all the packets/(2* no of transfers)

- Then we determine the benefit using the condition

If(overall average value > packet average value)
Then,
benefit = packet avg value - overall avg value;
Else
Benefit = packet avg value + overall avg value;

- based on the classification we determine the cost.

If the classification of a particular communication is **"vulnerable"** then its cost value is, **Cost=0.3.**

- If the classification of a particular communication is "stable" then we set the cost as 0.1
- The intrusiveness is the packet average value of each communication.
- Then using the benefit, cost and intrusiveness we compute the ROI value for all the communications captured.
- More the ROI value for a particular communication, the corresponding communication will be a better candidate for countermeasure selection.
- The risk probability, ROI and the severity of alert for all the captured communications will be given to the network administrator.
- The network admin can perform countermeasure to avoid the creation of zombie virtual machines using the above values.
- The countermeasure to be performed by the network administrator can be the future work for our model.

IV. Results

The output of the system is shown in the following figures. Vulnerability classification like the file is exploited, stable or vulnerable**,** has been done based on the threshold values.

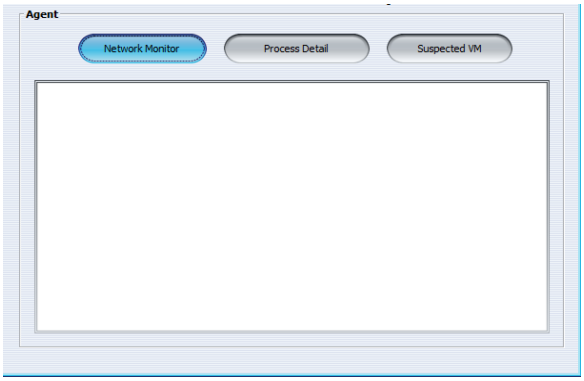

Figure 5: NIDCS-A

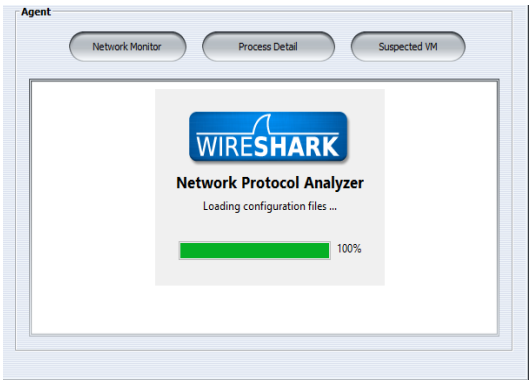

Figure 6: Opening Network Monitor.

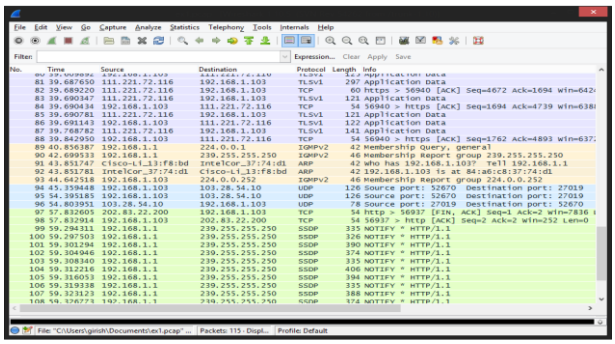

Figure 7: Capturing the packets using Wireshark (network monitor).

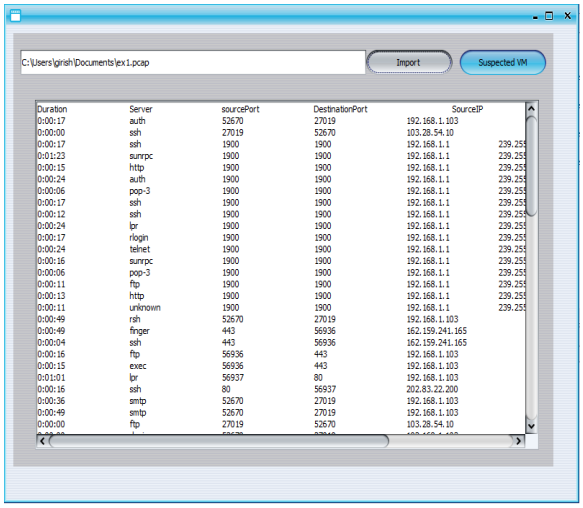

Figure 8: Fetching data from the database.

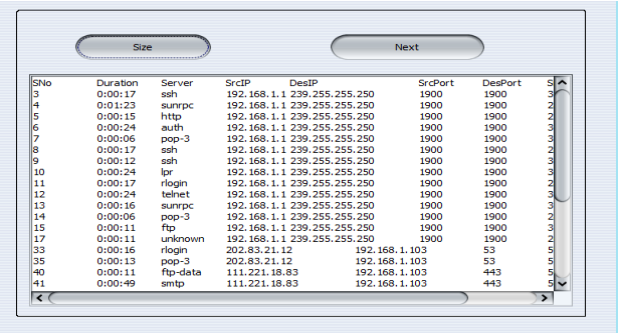

Figure 9: List of suspected IPs based on Packet size

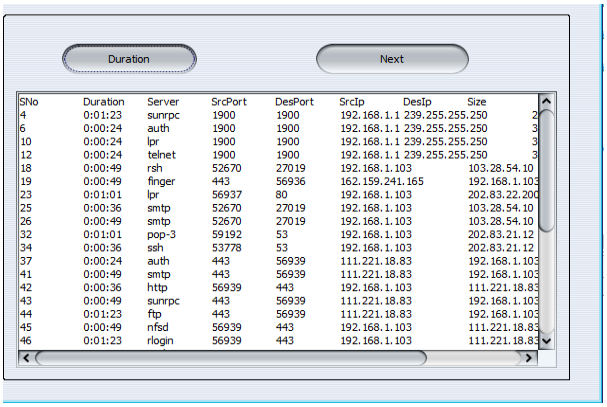

Figure 10: List of suspected IPs based on Duration.

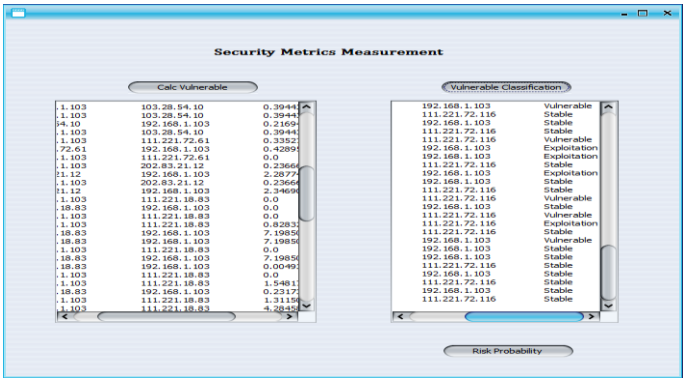

Figure 11: Vulnerability classification based on threshold value.

V. Conclusion And Future Enhancement

The system is presented to detect and mitigate collaborative attacks in the cloud virtual networking environment. It 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 NIDCS 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.

NIDCS 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, we will investigate the scalability of the proposed NIDCS solution by investigating the decentralized network control and attack analysis model based on current study. In future it can be applied to real cloud.

References

[1]. Coud Sercurity Alliance, "Top Threats to Cloud Computing v1.0,"https://cloudsecurityalliance.org/topthreats/csathreats. v1.0.pdf, Mar. 2010.

[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," ACM Comm., vol. 53,no. 4, pp. 50-58, Apr. 2010.I.

[3]. H. Takabi, J.B. Joshi, and G. Ahn, "Security and Privacy Challenges in Cloud Computing Environments," IEEE Security and Privacy, vol. 8, no. 6, pp. 24-31, Dec. 2010.

[4]. L. Wang, A. Liu, and S. Jajodia, "Using Attack Graphs for Correlating, hypothesizing, and Predicting Intrusion Alerts," Computer Comm., vol. 29, no. 15, pp. 2917-2933, Sept. 2006.

[5]. Roy, D.S. Kim, and K. Trivedi, "Scalable Optimal Countermeasure Selection Using Implicit Enumeration on Attack Countermeasure Trees," Proc. IEEE Int'l Conf. Dependable Systems Networks (DSN '12), June 2012.

[6]. N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic Security Risk Management Using Bayesian Attack Graphs," IEEE Trans. Dependable and Secure Computing, vol. 9, no. 1, pp. 61-74, Feb. 2012.

[7]. M. Frigault and L. Wang, "Measuring Network Security Using Bayesian Network-Based Attack Graphs," Proc. IEEE 32nd Ann. Int'l Conf. Computer Software and Applications (COMPSAC '08), pp. 698-703, Aug. 2008.