NICE: A System to Detecting A DoS in Virtual Machine

Kakade Rohini.B[1] Jadhav Asmita K[2] Bhise Varsha D[3] Prof.C.S Aryan[4]
[1,2,3]B.E. Student [4]Project Guide
[1,2,3,4]Department of Computer Engineering
[1,2,3,4]Jaihind College of Engineering Kuran Pune, India

Abstract— Cloud computing is getting popular now a days. Use of cloud is increase daily. As in the cloud environment resources such as OS virtual machines, software is shared by billions of users of the cloud. The virtual machines resides on the cloud are more vulnerable to the denial of service attack. If these machines are connected to more achiness then it becomes more dangerous as it harms all cloud networks. In the cloud especially infrastructure as a service the detection of denial of service attack is more challenging task. This is due to cloud users can install vulnerable software on the virtual machines. In this paper we have proposed multiphase vulnerability detection in the cloud environment. We have proposed an open flow network program NICE to detect and mitigate the attacks on the virtual machines. NICE is built on the scenario attack graph based model. We have proposed a novel approach to mitigate the attacks in the cloud environment by selecting different countermeasure depending upon the percentages of vulnerability of the virtual machines. Now a day IDS is used to detect the attack in the network by many organizations. In the proposed system we focus on the distributed denial of service attack in the cloud.

Key words: Cloud Computing, Scenario Attack Graph, Correlation, Network Analyzer, Intrusion, Zombies

I. Introduction

Large amount of research and studies have shown that cloud computing is vulnerable to the attacks. Cloud consist of large number of resources by millions of users. Now a days cloud computing is at the top of the security thread. Cloud have services like infrastructure as a service, platform as a service where user can deploy their software on the cloud virtually. We can see that the number of are moving towards a cloud but the main problem they are facing is of security. The cloud resources are used by attacker to deploy the vulnerable attacks on the shared virtual machines. In the existing system data is stored on the central server and server admin have full control over the data management. In the cloud resources are shared by millions of the user and user can install any software on the shared virtual machines and this leads to the violation of the cloud security. The main challenging issue in the cloud is to identify the cloud attack and find the solution to mitigate this attack. Attackers are using large shared infrastructure to deploy the attack. The cloud users are sharing the resources such as hub, switches, operating systems, virtual machines. All of this shared things are attract to the attackers to compromise the virtual machines. In [2] author addressed the business continuity and service availability from service outage is also main concern in the cloud computing. In the [1] authors have explains the cloud computing systems and its architecture. The cloud resources are transfer to the economical mode and its nothing but economical mode of denial of service. This system identifies that the request is generated by normal user or it get generated by bot itself.

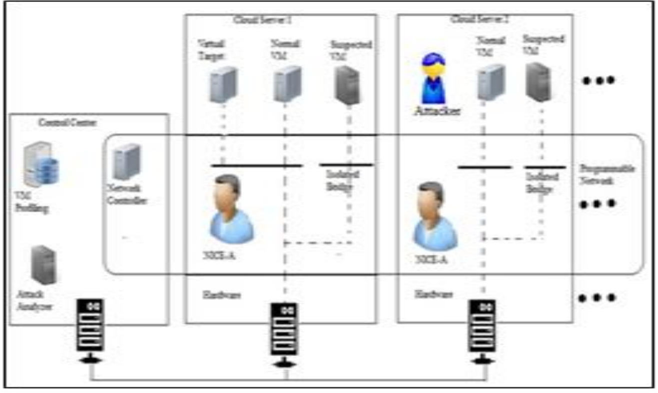

Fig. 1: NICE System architecture.

In this paper we have proposed a novel network detection and countermeasure selection procedure. Figure 1. Shows the NICE architecture. The NICE includes an attack graph correlation intrusion detection system. NICE introduces the attack graph analytical process to incorporate the intrusion detection process. Generally NICE include two main phases one is install light weight mirroring based intrusion detection agent at the virtual machines to scans the traffic to the virtual machines and provide the information to the attack analyser. Then according to the severity of the attack NICE decide whether to put the virtual machine in inspection or not. Once the virtual machines put in the inspection phase deep packet is applied and network reconfiguration is done. NICE improve the current intrusion detection or prevention method by introducing programmable or reconfigurable intrusion detection system by using software switching system [5]. The all information about VM is stored in the scenario attack graph (SAG) and according to all information in the SAG nice decide the appropriate action about the VM. NICE not needed to block the traffic of the virtual machine which is in suspicious mode. NICE incorporates the software switching technique for the virtual machine in the suspicious stage.

The rest of paper is arranged as follows. Section II present the existing work done for the network intrusion detection in the cloud environment. In Section III we have described our proposed system, the proposed security measurement, mitigation, and countermeasures. Section IV explains NICE in terms of network performance and security. Finally, Section V concludes this paper.

II. Literature Survey

In this section we will see the literature of the highly related to NICE existing methods. In [6] author explains the detection of the spam in the network. His work SPOT is based on the sequentially scanning of outgoing messages while deploying statistical method sequential probability ratio test method. This method determines whether host machine is compromised or not. Botsniffer [7] define the malware in the system according to the several stages which follows the correlation in the alarm triggered by the inbound traffic. The attack graph is maintained in this system shows that serious of the exploits. There are number of automation tools to construct the attack graph. [9] Proposed a modified symbolic model checking system and binary diagram to construct the attack graph. This module can generate the graph for all attack but the scalability is the main issue in this method. P. Ammann [11] introduced the monotonicity assumptions, which states that the precondition of a given exploit is never invalidated by the successful application of another exploit. [12] Introduces the logic programming approach and issues the datalog language to model and analyze the network attack detection system. Attack graph can be generated by accumulating true facts of the network. The attack graph generation process terminate because number of facts are polynomial in the network. To monitor and improve the attack graph system we have introduces attack correlation graph. The IDS system and firewall are widely used in the network intrusion detection system but the main problem with this is raw generation of false alarm. The main task in the attack correlation system is detection of the raw alert. Many attack correlation graphs are proposed by different authors recently. In [14] author proposed in memory graph called queue graph (QH) to detect the attack on each matching of the exploit. But it is difficult to detect the correlated alert in the attack graph for analysis of similar attack. [15] Proposed a modified attack correlation method for the attack graph mapping. He have proposed a new function for mapping multiple function to the map. He have proposed depended attack graph to group the related alert with multiple correlation criteria. Each edge of the DG represent the subset of the alert which might be part of attack scenario. After considering attack graph the next important task is apply the countermeasure. Several method to select the optimal correlation such as attack path and attack correlation. The paper [16] proposed a attack countermeasure tree (ACT). To consider attack countermeasure and countermeasure both in one. They depicts some branch and bound technique to minimize the countermeasure. Each optimized problem can be solved by using minimized probability in the tree. [17] Proposed (BAG) baysian attack graph which address dynamic security risk management. And applied genetic algorithm to mitigate the countermeasure selection. The another method to detect the network attack is bothunter. The following figure shows the working of bothunter.

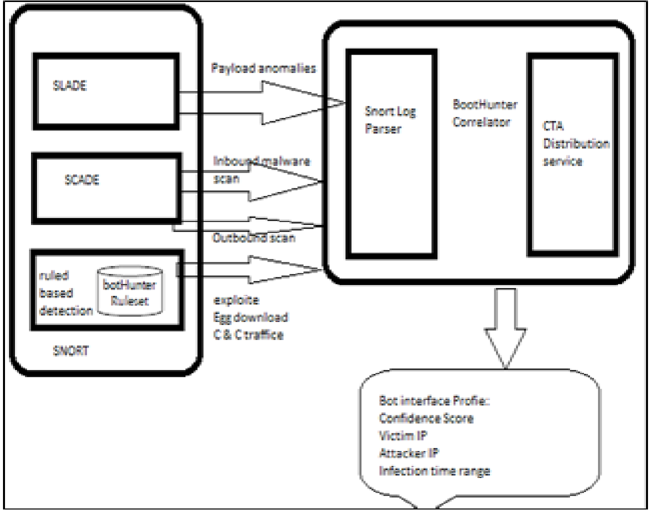

Fig. 2: Working of Bothunter

Our proposed solution is based on the network control approach SDN where networking function can be programmed through software switching methods. We have taken advantages of open flow network and software switching method to select and minimize the

III. PROPOSED SYSTEM

In the proposed system we have utilize the scenario attack graph to model the thread and vulnerability detection in the virtual network. NICE is based on the reconfigurable network to minimize the vulnerability in the virtual machines.

A. *Threat Model:*

In our proposed method we have consider attacker may be located inside or outside of the network. In the proposed system the main aim of proposed NICE is find out vulnerable virtual machine and compromise that machine as a zombie. We have introduces new software model which can resilient the zombie attack. In our method we are using clouds infrastructure as a service to deploy the nice agent. The proposed system predict the attacks on the virtual machines and mitigate the attack independently on the operating system. We have assumption that user can install any of the operating system he wants.

B. *Attack Graph Model:*

Attack graph is a tool to detect the all possible multithread multi-host attacks. In the attack graph each node explains precondition or consequence. As the attack graph provide all detailed information about the exploited vulnerabilities due to this we can get whole picture of the security threads of the system. The attack graph helps to take the appropriate decision for the countermeasure selection according to the current network security and can be mitigate the attack. According to the attack graph we can take appropriate decision about the vulnerable virtual machine.

1) *Definition 1: Scenario Attack Graph:*

Scenario attack graph is a tuple S= (V, E) where

V= union of set of vertices Nc, Nd, Nr. ,where Nc is exploit node, Nd is result of the exploit, and Nr is initial step of the attack.

E=E is union of Epre and Epost this are the directed edges.

Algorithm 1 (Alert Correlation):

Require: alert ac, SAG, ACG

```
if (ac is a new alert) then
create node ac
$n1 \leftarrow vc \in map(ac)$
for all $n2 \in parent(n1)$ do
create edge (n2.alert, ac)
for all si having a do
if a is last element In si then
append ac to the si
else
create path Si+1 = {subset(Si, a), ac}
end if
end for
add ac to n1.alert
end for
end if
return S
```

C. VM Profiliing:

The VM profiling model of NICE consist of all detailed information about all virtual machines, incoming traffic toward the virtual machine, we can make analysis of the vulnerability in the virtual machine. According to the vulnerabilities in the virtual machines we have three states in the virtual machines.

1) Stable: The VM will be in stable state if and only if there is not present any vulnerability on the virtual machine.
2) Vulnerable: It is the state of vulnerable machine which may have one or more vulnerability on it but not get exploited.
3) Exploited: It is the state of virtual machine which at least one vulnerabilities is get exploited and machine is get compromised.
4) Zombie: VM totally under control of Zombie.

IV. NICE SYSTEM DESIGN

NICE system consist of VM profiling, network analyser, network controller, NICE agent Reconfigurable network.

The figure 1 shows the system architecture of the proposed NICE model. This figure shows the nice system in cloud cluster. In the proposed system we have installed NICE at the host machine.

A. Network Analyzer:

The following figure shows the working of NA. The main function of the network analyser is to collect the all information from the NICE agent, virtual machine profiling, and maintain the scenario attack graph and attack correlation graph. According to the vulnerability of the virtual machine network analyser decides or select the countermeasure and forward this message to the network controller. Network analyser will decide is alert send by the nice agent is new or old if the alert is old then it make new entries in the virtual machine table and if the alert is old then network analyser update attack correlation graph and scenario attack graph.

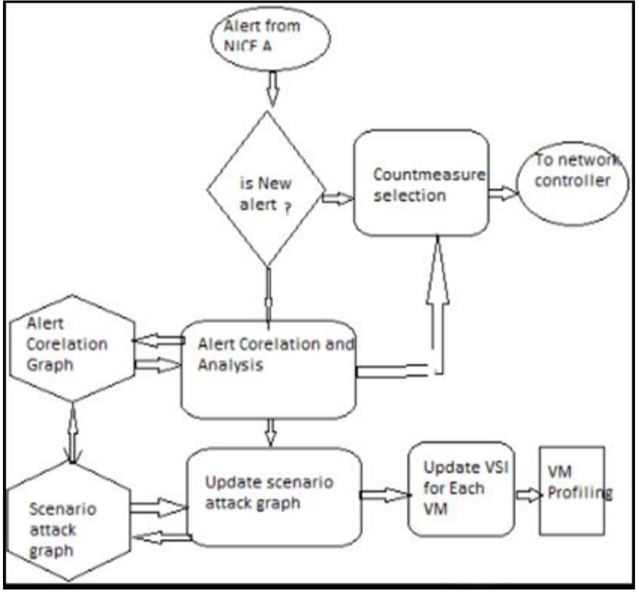

Fig. 3: Working of Network Analyser

The following figure shows the working of attack analyser. The network analyser collects the information from different parts of the system and maintains two graph and it will operate the network controller. The following figure shows the working of network analyser. It is present at the network side. The use of attack correlation is to detect the denial of service attack on the virtual machine. After selecting appropriate countermeasure it will forward the message to the network controller.

The attack analyser maintain the graph by using

B. Network Controller:

The second main part of the proposed system is network controller. The network controller acts as an assistant for network analyser. Network controller performs the countermeasure selected by the network analyser. The main functions of the network controller are reconfiguring the network and manage the processes running on the virtual machine.

V. EXPERIMENTAL SETUP

For the system set up we have consider two clouds one is private cloud and another one is private cloud. The cloud server1 and cloud server 2 are connected to each other by external firewall. If any vulnerability detect in the virtual machine then nice agent send alert message to the analyzer. The efficiency and correction in the attack detection and mitigation in the NICE is more as compare to the existing system.

VI. CONCLUSION

In the proposed system we have proposed a novel method to detect and mitigate the attacks in the cloud virtual environments. NICE uses the attack graph model to find and mitigate the attacks on the virtual machine. The NICE also introduces a programmable network model which helps to mitigate the attacks on the virtual machines. We have used host based approach to mitigate and to provide security to the whole cloud system. The proposed system can mitigate the attacks on the virtual machines. NICE investigate the counter zombie attack in the network ids. The proposed system perform better because we have deploy the NICE on the host Machine.

ACKNOWLEDGMENT

Author would like to take this opportunity to express our profound gratitude and deep regard to my guide prof. Saba Siraj for his exemplary guidance, valuable feedback and constant encouragement throughout the duration of the project. His valuable suggestions were of immense help throughout my project work. His perceptive criticism kept me working to make this project in a much better way. Working under him was an extremely knowledgeable experience for me.

REFERENCES

[1] Coud Sercurity Alliance, "Top threats to cloud computing v1.0," https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf, March 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 Commun., vol. 53, no. 4, pp. 50–58, Apr. 2010.

[3] B. Joshi, A. Vijayan, and B. Joshi, “Securing cloud computing environment against DDoS attacks,” IEEE Int’l Conf. Computer Communication and Informatics (ICCCI ’12), Jan. 2012.

[4] 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.

[5] “Open vSwitch project,” http://openvswitch.org, May 2012.

[6] Z. Duan, P. Chen, F. Sanchez, Y. Dong, M. Stephenson, and

[7] J. Barker, “Detecting spam zombies by monitoring outgoing messages,” IEEE Trans. Dependable and Secure Computing, vol. 9, no. 2, pp. 198–210, Apr. 2012..

[8] J. Li, X. Chen, M. Li, J. Li, P. Lee, andW. Lou. Secure deduplication with efficient and reliable convergent key management. In IEEE Transactions on Parallel and Distributed Systems, 2013.

[9] C.-K Huang, L.-F Chien, and Y.-J Oyang, “Relevant Term Suggestion in Interactive Web Search Based on Contextual Information in Query Session Logs,” J. Am. Soc. for Information science and Technology, vol. 54, no. 7, pp. 638-649, 2003.

[10] S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. Twin clouds: An architecture for secure cloud computing. In Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.

[11] W. K. Ng, Y. Wen, and H. Zhu. Private data deduplication protocols in cloud storage. In S. Ossowski and P. Lecca, editors, Proceedings of the 27th Annual ACM Symposium on Applied Computing, pages 441–446. ACM, 2012.

[12] R. D. Pietro and A. Sorniotti. Boosting efficiency and security in proof of ownership for deduplication. In H. Y. Youm and Y. Won, editors, ACM Symposium on Information, Computer and communications Security, pages 81–82. ACM.

[13] S. Quinlan and S. Dorward. Venti: a new approach to archival storage. In Proc. USENIX FAST, Jan 2002.

[14] A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui. A secure cloud backup system with assured deletion and version control. In 3rd International Workshop on Security in Cloud Computing, 2011.

[15] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. IEEE Computer, 29:38–47, Feb 1996.

[16] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl. A secure data deduplication scheme for cloud storage. In Technical Report, 2013.

[17] C. Ng and P. Lee. Revdedup: A reverse deduplication storage system optimized for reads to latest backups. In Proc. of APSYS, Apr 2013.