IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 2, Apr-May, 2014
ISSN: 2320 – 8791 (Impact Factor: 1.479)
www.ijreat.org

Secure Attack Measure Selection and Intrusion Detection in Virtual Cloud Networks

Kruthika S G[1], VenkataRavana Nayak[2], Sunanda Allur[3]

[1, 2, 3]**Department of Computer Science, Visvesvaraya Technological University, GSS Institute of Technology, Bangalore, Karnataka**

Abstract

Cloud security is one of most important issues that has attracted a lot of research and development effort in past few years. Particularly, attackers can explore vulnerabilities of a cloud system and compromise virtual machines to deploy further large-scale Distributed Denial-of-Service (DDoS).DDoS attacks usually involve early stage actions such as multi-step exploitation, low frequency vulnerability scanning, and compromising identified vulnerable virtual machines as zombies, and finally DDoS attacks through the compromised zombies. Within the cloud system, especially the Infrastructure-as-a-Service (IaaS) clouds, the detection of zombie exploration attacks is extremely difficult. This is because cloud users may install vulnerable applications on their virtual machines. To prevent vulnerable virtual machines from being compromised in the cloud, we propose a multi-phase distributed vulnerability detection, measurement, and countermeasure selection mechanism called NICE, which is built on attack graph based analytical models and reconfigurable virtual network-based countermeasures.
.

Keywords: *Network Security, Cloud Computing, Intrusion Detection, Zombie detection.*

1. Introduction

RECENT studies have shown that users migrating to the cloud consider security as the most important factor. A recent 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 [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 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. In, M. Armbrust et al. 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 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. Such attacks are more effective in the cloud environment since 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, etc., attracts attackers to compromise multiple VMs.

In this article, we propose NICE (Network Intrusion detection and Countermeasure selection in virtual network systems) to establish a defense-in-depth intrusion detection framework. For better attack detection, NICE incorporates attack graph analytical procedures into the intrusion detection processes. We must note that the design of NICE does not intend to improve any of the existing intrusion detection algorithms; indeed, NICE employs a reconfigurable virtual networking approach to detect and counter the attempts to compromise VMs, thus preventing zombie VMs. In general, nice includes two main phases: (1) deploy a lightweight mirroring-based network intrusion detection agent (nice-a) on each cloud server to capture and analyze cloud traffic. A nice-a periodically scans the

virtual system vulnerabilities within a cloud server to establish scenario attack graph(sag) and then based on the severity of identified vulnerability towards the collaborative attack goals, NICE will decide whether or not to put a VM in network inspection state. (2) 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.

NICE significantly advances the current network IDS/IPS solutions by employing programmable virtual networking approach that allows the system to construct a dynamic reconfigurable IDS system. By using software switching techniques [5], NICE constructs a mirroring-based traffic capturing framework to minimize the interference on users' traffic compared to traditional bump-in-the-wire (i.e., proxy-based) IDS/IPS. The programmable virtual networking architecture of NICE enables the cloud to establish inspection and quarantine modes for suspicious VMs according to their current vulnerability state in the current SAG. Based on the collective behavior of VMs in the SAG, NICE can decide appropriate actions, for example DPI or traffic filtering, on the suspicious VMs. Using this approach, NICE does not need to block traffic flows of a suspicious VM in its early attack stage. The contributions of NICE are presented as follows:

• NICE device, a new multi-phase 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.

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

• NICE employs a novel attack graph approach for attack detection and prevention by correlating attack behavior and also suggests effective countermeasures.

• NICE optimizes the implementation on cloud servers to minimize resource consumption. Our study shows that NICE consumes less computational overhead compared to proxy-based network intrusion detection solution.

2. Literature Survey

2.1 Overview of DDOS attacks.

Title: Securing cloud computing environment against DDoS attacks.Cloud computing is becoming one of the next IT industry buzz word. However, as cloud computing is still in its infancy, current adoption is associated with numerous challenges like security, performance, availability, etc. In cloud computing where infrastructure is shared by potentially millions of users, Distributed Denial of Service (DDoS) attacks have the potential to have much greater impact than against single tenanted architectures. This paper tested the efficiency of a cloud trace back model in dealing with DDoS attacks using back propagation neural network and finds that the model is useful in tackling Distributed Denial of Service attacks. Cloud Computing is the newly emerged technology of Distributed Computing System. Cloud Computing user concentrate on API security & provide services to its consumers in multitenant environment into three layers namely, Software as a service, Platform as a service and Infrastructure as a service, with the help of web services. It provides service facilities to its consumers on demand. These service provided can easily invites attacker to attack by Saas, Paas, and Iaas. Since the resources are gathered at one place in data centers in cloud computing, the DDOS attacks such as HTTP & XML in this environment is dangerous & provides harmful effects and also all consumers will be affected at the same time. These attacks can be resolved & detected by a proposed methodology. In this methodology, this problem can be overcome by using proposed system. The different kinds of vulnerabilities are detected in proposed system. The SOAP request makes the communication between the client and the service provider. Through the Service Oriented Trace back Architecture the SOAP request is send to the cloud. In this architecture service oriented trace back mark is present which contain proxy within it. The proxy that marks the incoming packets with source message identification to identify the real client. Then the SOAP message is travelled via XDetector. The XDetectors used to monitors and filters the DDoS attacks such as HTTP and XML DDoS attack. Finally the filtered real client message is transferred to the cloud service provider and the corresponding services are given to the client in secured manner.

Authors' Names and Addresses:
Bansidhar Joshi, A.Santhana Vijayan.
Department of computer Science
NIT, Tiruchirappalli, India. & Bineet Kumar Joshi
ICFAI University Dehradun, India.

2.2 Overview of DDOS attacks

Title: Security and privacy challenges in cloud computing environments.

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three delivery models, and four deployment models.

Applications running on or being developed for cloud computing platforms pose various security and privacy challenges depending on the underlying delivery and deployment models. The three key cloud delivery models are software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). In IaaS, the cloud provider supplies a set of virtualized infrastructural components such as virtual machines (VMs) and storage on which customers can build and run applications. The three key cloud delivery models are software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS).

Authors' Names and Addresses:
Hassan Takabi and James B.D .Joshi. University of Pittsburgh.Gail-Joon Ahn Arizona State University.

3. Proposed Method

3.1 NICE SYSTEM DESIGN

In this section, we first present the system design overview of NICE and then detailed descriptions of its components.

3.1.1 System design overview

The proposed NICE framework is illustrated in Section 4.1; figure 1. It shows the NICE framework within one cloud server cluster. Major components in this framework are distributed and light-weighted NICE-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). NICE- 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 OpenFlow tunneling or VLAN approaches. The network controller is responsible for deploying attack countermeasures based on decisions made by the attack analyzer.

3.1.2 VM Profiling

Virtual machines in the cloud can be profiled to get precise information about their state, services running, open ports, etc. One major factor that counts towards 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 5 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.

3.1.4 Attack Analyzer

The major functions of NICE system are performed by attack analyzer, which is shown in section 4.1; figure 2 that include procedures such as attack graph construction and update, alert correlation and countermeasure selection. The process of constructing and utilizing the Scenario Attack Graph (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.

3.1.5 Network Controller

The network controller is a key component to support the programmable networking capability to realize the virtual network reconfiguration feature based on Open Flow protocol. In NICE, within each cloud server there is a software switch, for example, Open vSwitch (OVS) [5], which is used as the edge switch for VMs to handle traffic in & 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.

Network controller is also responsible for applying the countermeasure from attack analyzer. Based on VM Security Index 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. Countermeasure in such case is to put the suspicious VM with exploited state into quarantine mode and redirect all its flows to NICEA Deep Packet Inspection (DPI) mode. An alert with a minor threat level can be generated due to the presence of a vulnerable VM. For this case, in order 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.

Countermeasure selection:
Algorithm presents how to select the optimal countermeasure for a given attack scenario. Input to the algorithm is an alert, attack graph G, and a pool of countermeasures CM. The algorithm starts by selecting the node v that corresponds to the alert generated by a NICE-A. Before selecting the countermeasure, we to the target node .If the distance is greater than a threshold value, we do not perform countermeasure selection but update the ACG to keep track of alerts in the system (line 3). For the source node v count the distance of v Alert, all the reachable nodes (including the source node) are collected into a set T (line 6). Because the alert is generated only after the attacker has performed the action, we set the probability of v to1 and calculate the new probabilities for its entire child (downstream) nodes in the set T (line 7 & 8). Now for all t ∈ T the applicable countermeasures in CM are selected and new probabilities are calculated according to the effectiveness of the selected countermeasures (line 13 & 14). The change in probability of target node gives the benefit for the applied countermeasure using (7). In the next double for-loop, we compute the Return of Investment (ROI) for each benefit of the applied countermeasure based on (8). The countermeasure which when applied on a node gives the least value of ROI, is regarded as the optimal countermeasure. Finally, SAG and ACG are also updated before terminating the algorithm. The complexity of Algorithm is O (|V |×|CM|) where |V |is the number of vulnerabilities and |CM| represents the number of countermeasures.

Algorithm: Countermeasure Selection
Require: Alert, G (E, V), CM
1: Let vAlert= Source node of the Alert
2: if Distance to Target (valert)>threshold then
3: Update _ACG
4: return
5: end if
6: Let T = Descendant (valert) U vAlert
7: Set Pr (vAlert) =1
8: Calculate Risk Prob (T)
9: Let benefit [|T|, |CM|] =Ø
10: for each t∈T do
11: for each cm ∈CM do
12: if cm.condition (t) then
13: Pr (t) =Pr (t) * (1 - cm.effectiveness)
14: Calculate_ Risk_ Prob (Descendant (t))
15: benefit [t, cm] =ΔPr (target_ node).
16: end if
17: end for
18: end for
19: Let ROI [|T|, |CM|] =Ø
20: for each t∈T do
21: for each cm ∈CM do

$$\text{ROI}[t, cm] = \frac{\text{Benefit}[t, cm]}{\text{cost.cm} + \text{intrusiveness.cm}}$$

23: end for
24: end for
25: Update _SAG and Update_ ACG
26: return Select_ Optimal CM (ROI)

4. Figures

4.1. Figures

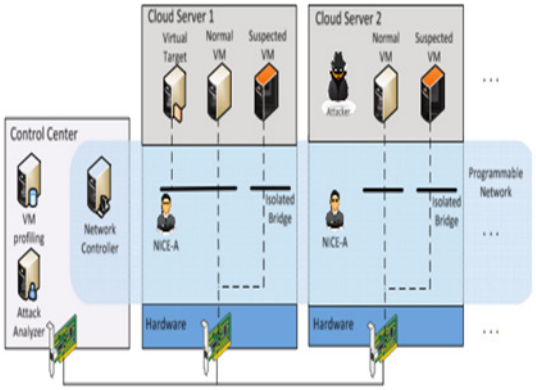

Fig.1. Proposed NICE framework within one cloud server

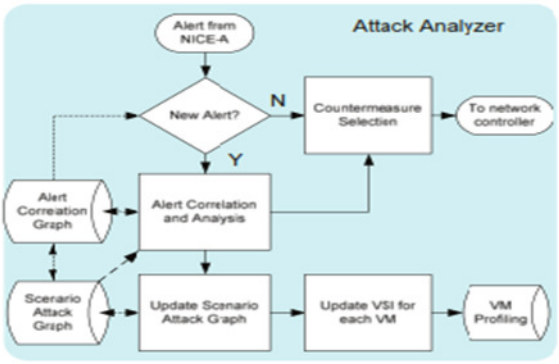

Fig.2. Workflow of Attack Analyzer

5. Conclusions

In this paper, we presented NICE, which is proposed to detect and mitigate collaborative attacks in the cloud virtual networking environment. NICE 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 NICE 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.

NICE 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

References

[1] Cloud Security Alliance, "Top threats to cloud computingv1.0,"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.