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

Prevention of Vulnerable Virtual Machines against DDOS Attacks in the Cloud

C.Kavitha[1]

[1]M.E, First Year, Department of Computer Science and Engineering, Saveetha Engineering College, Chennai, Tamil Nadu, India

Abstract

Cloud Security is one amongst most significant problems that have attracted plenty of analysis and development effort in past few years. Notably, attackers will explore vulnerabilities of a cloud system and compromise virtual machines to deploy additional large-scale Distributed Denial-of-Service (DDoS). DDoS attacks sometimes involve early stage actions like multi-step exploitation, low frequency vulnerability scanning, and compromising identified vulnerable virtual machines as zombies, and at last DDoS attacks through the compromised zombies. Among the cloud system, particularly the Infrastructure-as-a-Service (IaaS) clouds, the detection of zombie exploration attacks is extremely troublesome. This can be as a result of cloud users could install vulnerable applications on their virtual machines. To stop vulnerable virtual machines from being compromised within the cloud, we tend to propose a multi-phase distributed vulnerability detection, measurement, and countermeasure selection mechanism known as NICE, that is built on attack graph primarily based analytical models and reconfigurable virtual network-based countermeasures.

Keywords- *Cloud Security, Cloud Attacks, Distributed Denial of Service Attack, NICE, Attack graph model*

1. INTRODUCTION

Cloud Computing is a technology that uses the web and central remote servers to keep up information and applications. Cloud computing permits consumers and businesses to use applications without installation and access their personal files at any computer with web access. This technology permits for rather more efficient computing by centralizing information storage, process and bandwidth. Cloud computing is usually used to network-based services, that seem to be provided by real server hardware, and are in fact served up by virtual hardware, simulated

by software package running on one or additional real machines. Such virtual servers don't physically exist and might so be affected around and scaled up or down on the fly without touching the end user. Cloud computing is a network-based environment that focuses on sharing computations or resources.

Actually, clouds are Internet-based and it tries to disguise complexity for clients. Cloud computing refers to both the applications delivered as services over the Internet and the hardware and software in the datacenters that provide those services. Cloud providers use virtualization technologies combined with self-service abilities for computing resources via network infrastructure. In cloud environments, several kinds of virtual machines are hosted on the same physical server as infrastructure.

In 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, 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 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.
Cloud security is an evolving sub-domain of computer security, network security, and, more broadly, information security. It refers to a broad set of policies, technologies, and controls deployed to protect data, applications, and the associated infrastructure of cloud computing. For enterprises the most important problem is also security but with different vision. The cloud is not inherently less safe. There are many forms of cloud attacks. Among them important attacks that exist are *DDoS attacks against Cloud, Cloud against DDoS attacks, Extensible Markup Language (XML) based Denial of Service (X-DoS), Hypertext Transfer Protocol (HTTP) based Denial of Service (H-DoS).*

1) **Denial of service attack against cloud** has become an increasingly prevalent security threat in cloud. The attack intentionally compromises the availability of the virtual machines, and it is typically against the will of affected cloud users.

2) **Distributed denial-of-service attack against cloud** is one in which a multiple compromised systems or compromise multiple virtual machines attack a single target (cloud), thereby causing denial of service for cloud users of the targeted system. A computer under the control of an intruder is called as a zombie or bot. A group of co-opted computers is known as a botnet or a zombie army.

3) **XML based DDOS attack:** XML DoS attacks are extremely asymmetric: to deliver the attack payload, an attacker needs to spend only a fraction of the processing power or bandwidth that the victim needs to spend to handle the payload. Worse still, DoS vulnerabilities in code that processes XML are also extremely widespread.

4) **HTTP based DDOS attack:** When an HTTP client (say, a Web browser) talks to an HTTP server (a Web server), it sends requests which can be of several types, the two main being GET and POST. A GET request is what is used for "normal links", including images; such requests are meant to retrieve a static piece of data, the URL pointing to that piece of data. When you enter a URL in the URL bar, a GET is also done.

Among these different types of attacks, Distributed Denial of Service Attack is more vulnerable to cloud which compromise the virtual machines to explore DDOS attack against cloud. Compromised machines are one of the key security threats on the Internet; they are often used to launch various security attacks such as DDoS, spamming, and identity theft. In this thesis we address this issue by investigating effective solutions to automatically identify compromised machines in a network

In this paper, I propose NICE (**N**etwork **I**ntrusion detection and **C**ountermeasure s**E**lection in virtual net- work 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.

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 (SAGs), 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.

The contributions of NICE are presented as follows:

(1) We devise NICE, a new multi-phase distributed network intrusion detection and prevention framework in a virtual networking environment that captures.

(2) NICE incorporates a software switching solution to quarantine and inspect suspicious VMs for further investigation and protection. NICE can improve the attack detection probability and improve the resiliency to VM exploitation attack without interrupting existing normal cloud services.

(3) NICE employs a novel attack graph approach for attack detection and prevention by correlating attack behavior and also suggests effective countermeasures

(4) 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 solutions.

2. RELATED WORK

K.Santhi [2] propose Service Oriented Trace back Architecture (SOTA) applying framework to OGSA. We further add to our work by introducing a defense filter called XDetector [XML Detector], in which it is distributed throughout the grid, in order to properly defend it. Our system is one of the first defense systems to attempt to defend against these new attacks. DPM methodology is applied to our SOTA framework; by placing the Service-Oriented Traceback Mark (SOTM) within web service messages. If any other web security services (WS-Security for example) are already being employed. Defense filter is used in this paper to detect suspicious messages and attacks. If attack is found, the corresponding request is dropped before forwarding it to server. The request is transferred to the server only when no attack is found and consequent service reply for the request would be obtained.

Peng Chen, et.al [3] proposes effective spam zombie detection system named SPOT by monitoring outgoing messages of a network. SPOT is designed based on a powerful statistical tool called Sequential Probability Ratio Test, which has bounded false positive and false negative error rates. In this paper we address this issue by investigating effective solutions to automatically identify compromised machines in a network. They develop the spam zombie detection system SPOT which utilizes the Sequential Probability Ratio Test (SPRT) presented in the last chapter. As a comparation, it also gives two alternative designs CT and PT.

Nayot Poolsappasit, et.al [4] proposes a risk management framework using Bayesian networks that enable a system administrator to quantify the chances of network compromise at various levels. In this paper, they show how to use this information to develop a security mitigation and management plan. In contrast to other similar models, this risk model lends itself to dynamic analysis during the deployed phase of the network. A multi objective optimization platform provides the administrator with all trade-off information required to make decisions in a resource constrained environment. Further they propose an alternative method of security risk assessment that they call Bayesian Attack Graphs (BAGs). In particular, they adapt the notion of Bayesian belief networks so as to encode the contribution of different security conditions during system compromise. His model incorporates the usual cause consequence relationships between different network states (as in attack graphs and attack trees) and, in addition, takes into account the likelihoods of exploiting such relationships.

Eric Keller, et.al [6] propose NoHype architecture to indicate the removal of the hypervisor, addresses each of the key roles of the virtualization layer: arbitrating access to CPU, memory, and I/O devices, acting as a network device (e.g., Ethernet switch), and managing the starting and stopping of guest virtual machines. Additionally, they show that NoHype architecture may indeed be “no hype”, since nearly all of the needed features to realize the NoHype architecture are currently available as hardware extensions to processors and I/O devices. NoHype architecture removes the virtualization layer yet retains the management capabilities needed by cloud infrastructures. To do this, recall the major functions of the virtualization layer: arbitrating access to memory, CPU, and devices, providing important network functionality, and controlling the execution of virtual machines.

3. SYSTEM ARCHITECTURE

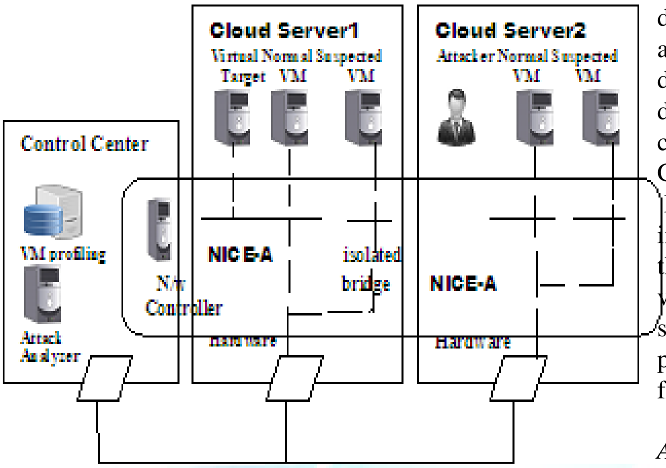

Fig 1: System architecture

The proposed NICE framework is illustrated in figure. 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. 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.

4. SOLUTIONS TO THE PROBLEM

4.1 NICE MODEL

In this section, I describe how to utilize attack graphs to model security threats and vulnerabilities in a virtual networked system, and propose a VM protection model based on virtual network reconfiguration approaches to prevent VMs from being exploited.

Threat model: In this attack model, we assume that an attacker can be located either outside or inside of the virtual networking system. The attacker's primary goal is to exploit vulnerable VMs and compromise them as zombies. Our protection model focuses on virtual-network-based attack detection and reconfiguration solutions to improve the resiliency to zombie explorations. My work does not involve host-based IDS and does not address how to handle encrypted traffic for attack detections. In my proposed solution can be deployed in an Infrastructure-as-a-Service (IaaS) cloud networking sys- tem, and we assume that the Cloud Service Provider (CSP) is begin.

I also assume that cloud service users are free to install whatever operating systems or applications they want, even if such action may intro- duce vulnerabilities to their controlled VMs. Physical security of cloud server is out of scope of this paper. We assume that the hypervisor is secure and free of any vulnerability.

Attack Graph Model: An attack graph is a modeling tool to illustrate all possible multi-stage, multi-host attack paths that are crucial to understand threats and then to decide appropriate countermeasures. In an attack graph, each node represents either precondition or consequence of an exploit. The actions are not necessarily an active attack since normal protocol interactions can also be used for attacks. Attack graph is helpful in identifying potential threats, possible attacks and known vulnerabilities in a cloud system.

Since the attack graph provides details of all known vulnerabilities in the system and the connectivity in- formation, we get a whole picture of current security situation of the system where we can predict the possible threats and attacks by correlating detected events or activities. If an event is recognized as a potential attack, we can apply specific countermeasures to mitigate its impact or take actions to prevent it from contaminating the cloud system.

4.2 SYSTEM COMPONENTS

4.2.1 Nice-A

The NICE-A is a Network-based Intrusion Detection System (NIDS) agent installed in each cloud server. It scans the traffic going through the bridges that control all the traffic among VMs and in/out from the physical cloud servers. It will sniff a mirroring port on each virtual bridge in the Open vSwitch. Each bridge forms an isolated subnet in the virtual network and connects to all related VMs. 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. It's more efficient to scan the traffic in cloud server since all traffic in the cloud server needs go through it; however our design is independent to the installed VM. The false alarm rate could be reduced through our architecture design.

4.2.2 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. 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.

4.2.3 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 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. VSI can be used to measure the security level of each VM in the virtual network in the cloud system

The Attack Analyzer also handles alert correlation and analysis operations. This component has two major functions:

(1) Constructs Alert Correlation Graph (*ACG*)
(2) Provides threat information and appropriate countermeasures to network controller for virtual network reconfiguration. NICE attack graph is constructed based on the following information: *Cloud system information, Virtual network topology and configuration information, Vulnerability information.*

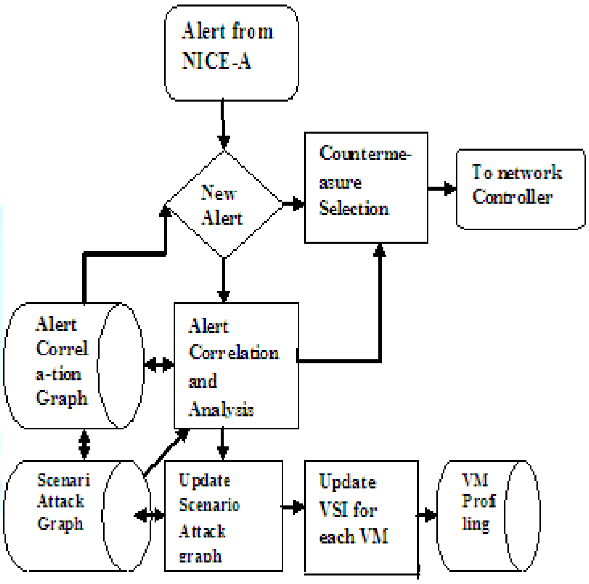
Alert from NICE-A
New Alert
Counterme-asure Selection
To network Controller
Alert Correl a-tion Graph
Alert Correlation and Analysis
Scenari Attack Graph
Update Scenario Attack graph
Update VSI for each VM
VM Prof ling

Fig 2: Workflow of Attack Analyzer

4.2.4 Network Controller

The network controller is a key component to support the programmable networking capability to realize the virtual network reconfiguration. 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 network controller is responsible for collecting network information of current Open Flow network and provides input to the attack analyzer to construct attack graphs.

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. 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.

5. CONCLUSIONS AND FUTURE WORK

In this paper, I 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. 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. This should be investigated in the future work. Additionally, as indicated in the paper, we will investigate the scalability of the proposed NICE solution by investigating the decentralized network control and attack analysis model based on current study.

REFERENCES

[1] Chun-Jen Chung, Pankaj Khatkar, Tianyi Xing, Jeongkeun Lee, and Dijiang Huang, "NICE: Network Intrusion Detection and Countermeasure, Selection in Virtual Network Systems,

[2] K.Santhi, "A Defense Mechanism to Protect Cloud Computing Against Distributed Denial of Service Attacks" International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 5, May 2013

[3] Z. Duan, P. Chen, F. Sanchez, Y. Dong, M. Stephenson, and J. Barker, "Detecting spam zombies by monitoring outgoing messages," *IEEE Trans. Dependable and Secure Computing*, vol. 9, no. 2, pp. 198-210, Apr. 2012.

[4] 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.

[5] E. Keller, J. Szefer, J. Rexford, and R. B. Lee, "NoHype: virtualized cloud infrastructure without the virtualization," *Proc. of the 37th ACM ann. int'l symp. on Computer architecture (ISCA '10)*, pp. 350- 361. Jun. 2010.

[6]P. Mell, K. Scarfone, and S. Romanosky, "Common vulnerability scoring system (CVSS)," http://www.first.org/cvss/cvss-guide. html, May 2010.

[67 N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKe- own, and S. Shenker, "NOX: towards an operating system for networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 3, pp. 105-110, Jul. 2008.

[8] X. Ou and A. Singhal, *Quantitative Security Risk Assessment of Enterprise Networks*. Springer, Nov. 2011.

[9] M. Frigault and L. Wang, "Measuring network security using bayesian Network-Based attack graphs," *Proc. IEEE 32nd ann. int'l conf. on Computer Software and Applications (COMPSAC '08)*, pp. 698-703. Aug. 2008
.

[10]K. Kwon, S. Ahn, and J. Chung, "Network security management using ARP spoofing," *Proc. Int'l Conf. on Computational Science and Its Applications (ICCSA '04)*, LNCS, vol. 3043, pp. 142-149, Springer, 2004.

.