IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

The complex and choosing countermeasure in virtual Network Systems

Vijay Kumar[1], Sata Nand[2], Lekh Raj[3]

Assistant Professor, Department of Computer Science & Application, Ch. Devi Lal University Sirsa Haryana[1]

M.Tech Scholar, Department of Computer Science & Application, Ch. Devi Lal University Sirsa Haryana[2,3]

Abstract: Network Intrusion Detection System (NIDS) is an intrusion detection system that attempts to discover unauthorized access to a computer network by analyzing traffic on the network for signs of malicious activity. In traditional data centers, where system administrators have full control over the host machines, vulnerabilities can be detected and informed by the system administrator in a centralized manner. But in case of cloud data centers, where the cloud users are having the independence to install and control the desired software, they can install vulnerable software's on their managed VM's and contravene the Service Level Agreement (SLA) and as a result loophole in cloud security are created.

Keyword: NIDS, signs, vulnerabilities, cloud data centers, Service Level Agreement.

I. INTRODUCTION

The effect of security breach in cloud computing environment we have to develop an efficient vulnerability/attack detection and response system for precisely recognizing attacks. As in cloud system the infrastructure is communal to millions of users, hence mistreat and despicable use of the shared infrastructure benefits attackers to take advantage of vulnerabilities of the cloud and use its resources 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.

A technique is proposed to detect and select Countermeasure on virtual Machines: NICE (Network Intrusion detection and Countermeasures Election in virtual network systems) to establish a defense-in-depth invasion detection framework. Existing intrusion detection systems are based on the assumption that an intruder will behave differently from the legitimate user and hence can be easily identified. It also assumes that nearly all the unauthorized actions are noticeable. For better attack detection, NICE incorporates attack graph analytical procedures into the intrusion detection processes. We must note that the design of NICE utilizes a reconfigurable virtual networking approach to detect and oppose the attempts to concess the VM's, thus averting zombie VMs.

II. METHODOLOGY

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. 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.

Disadvantages of existing system:
1) No detection and prevention framework in a virtual networking environment.
2) Not accuracy in the attack detection from attackers.

Proposed System:
Here 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.

Advantages of the proposed system:
The contributions of NICE are presented as follows:

DOI 10.17148/IJARCCE.2016.54217

- We devise NICE, 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 solutions.

NICE (Network Intrusion detection and Countermeasure Selection in the virtual network system)

Here, 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.

The contributions of NICE are presented as follows:

- We devise NICE, 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.

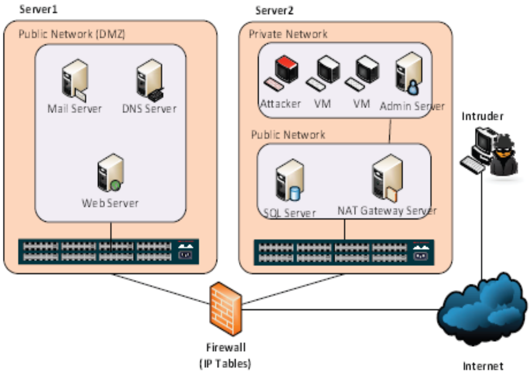
NICE optimizes the implementation on cloud servers to minimize resource consumption.

Our study shows that NICE consumes less computational overhead compared to proxy-based network intrusion detection solutions.

III. MODULES

There are four modules to define my presented technique and its implementation. So the followings are module:

1. Nice-A Module
2. VM Profiling Module
3. Attack Analyzer Module
4. Network Controller Module



Virtual network topology for security evaluation

Modules Description:

Nice-A Module

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 v Switch. 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.

VM Profiling Module

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.

Attack Analyzer Module

The major functions of NICE system are performed by attack analyzer, which includes procedures such as attack

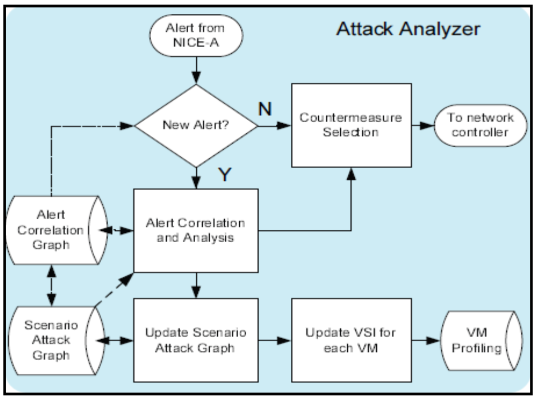
DOI 10.17148/IJARCCE.2016.54217

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. 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.



Workflow of Attack Analyzer

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 Algorithm 2, and then notifies network controller immediately to deploy countermeasure or mitigation actions.

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.

Future Scope:

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. An attack graph is able to represent a series of exploits, called atomic attacks, that lead to an undesirable state, for example a state where an attacker has obtained administrative access to a machine. There are many automation tools to construct attack graph. Intrusion Detection System (IDS) and firewall are widely used to monitor and detect suspicious events in the network. However, the false alarms and the large volume of raw alerts from IDS are two major problems for any IDS implementations. Additionally, we can investigate the scalability of the proposed NICE solution by investigating the decentralized network control and attack analysis model based on current study.

IV. CONCLUSION

Here a technique is proposed to detect and mitigate collaborative attacks in the cloud virtual networking environment. The proposed technique is an NIDS. A "network intrusion detection system (**NIDS**)" monitors traffic on a network looking for suspicious activity, which could be an attack or unauthorized activity. For this purpose a large NIDS server can be set up on a backbone network in order to monitor all traffic; or smaller systems can be set up to monitor traffic for a particular server, switch, gateway, or router. NIDS server does not replace primary security such as firewalls, encryption and other authentication methods. Existing intrusion detection systems are based on the assumption that an intruder will behave differently from the legitimate user and hence can be easily identified. It also assumes that nearly all the unauthorized actions are noticeable.

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.

REFERENCES

[1]. Denning, D. E. and Neumann, P. G. “Requirements and Model for IDES -- a Real-Time Intrusion Detection System”, Tech. report, Computer Science Lab, SRI International, 1985.

[2]. K. Scarfone, P. Mell, “Guide to Intrusion Detection and Prevention Systems (IDPS)”. Computer Security Resource Center (National Institute of Standards and Technology).

[3]. Ptacek, Thomas H. & Newsham, Timothy N. (January 1998); "Insertion, Evasion, and Denial of Service: Eluding Network Instrusion Detection".

[4]. Lunt, Teresa F., "Detecting Intruders in Computer Systems," 1993 Conference on Auditing and Computer Technology, SRI International.

[5]. Sebring, Michael M., and Whitehurst, R. Alan.,"Expert Systems in Intrusion Detection: A Case Study," The 11th National Computer Security Conference, October, 1988.

[6]. Cloud Security Alliance (CSA), “Top threats to cloud computing v1.0,”https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf, March 2010.

[7]. 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.

[8]. 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.

DOI 10.17148/IJARCCE.2016.54217

[9]. 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.
[10]. “Open vSwitch project,” http://openvswitch.org, May 2012. Z. Duan, P. Chen, F. Sanchez, Y. Dong, M. Stephenson, and J. Barker, “Detecting spam zombies by monitoring outgoing messages,”
[11]. G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, “BotHunter: detecting malware infection through IDS-driven dialog correlation,” Proc. of 16th USENIX Security Symp. (SS ’07), pp. 12:1–12:16, Aug. 2007.
[12]. G. Gu, J. Zhang, and W. Lee, “BotSniffer: detecting botnet command and control channels in network traffic,” Proc. of 15th Ann. Network and Distributed Sytem Security Symp. (NDSS ’08), Feb. 2008.
[13]. O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing,“Automated generation and analysis of attack graphs,” Proc. IEEE Symp. on Security and Privacy, 2002, pp. 273–284.
[14]. “NuSMV: A new symbolic model checker,” http://afrodite.itc.it: 1024/~nusmv. Aug. 2012.
[15]. 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.
[16]. X. Ou, S. Govindavajhala, and A. W. Appel, “MulVAL: a logicbased network security analyzer,” Proc. of 14th USENIX Security Symp., pp. 113–128. 2005.
[17]. R. Sadoddin and A. Ghorbani, “Alert correlation survey: framework and techniques,” Proc. ACM Int’l Conf. on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services (PST ’06), pp. 37:1–37:10. 2006.
[18]. 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.
[19]. S. Roschke, F. Cheng, and C. Meinel, “A new alert correlation algorithm based on attack graph,” Computational Intelligence in Security for Information Systems, LNCS, vol. 6694, pp. 58–67. Springer, 2011.
[20]. A. Roy, D. S. Kim, and K. Trivedi, “Scalable optimal countermeasure selection using implicit enumeration on attack countermeasure trees,” Proc. IEEE Int’l Conf. on Dependable Systems Networks (DSN ’12), Jun. 2012.

DOI 10.17148/IJARCCE.2016.54217