SECURE INTRUSION DETECTION AND COUNTERMEASURE SELECTION IN VIRTUAL SYSTEM USING IDM

Sinju N S[1]

[1]*PG Scholar, Dept of CSE, Paavai Engineering College, Namakkal, Tamil Nadu, India*

Abstract

In a virtual networked system where infrastructure is shared by more than one user security is a serious concern. Good network security protects a network in a manner that is consistent with its purpose and precautions must be taken when choosing a network provider for an organization. An attacker can enter into the system and compromise virtual machines and then exploit vulnerabilities. During the packet flow through the network experiences some delay. The existing intrusion detection processes incorrectly identify this delay as intruder's attack. This work is intended to efficiently identify the intruders in the case of congestion and find possible countermeasure using Intrusion detection manager (IDM) in virtual networked system. The Intruder Detector and Manager (IDM) algorithm detects and prevents the intruder entering into the network by maintaining tables and reconfigurable virtual network-based countermeasures in order to avoid the masquerading DDoS attacks. When this procedure is followed, IDM increases the throughput by preventing intruders and maintains the history of intruders.

Keywords: *Network security, Intrusion detection, compromised machine, cloud computing, countermeasure selection.*

----------***----------

1. INTRODUCTION

As the computer and networked system increases in the world of today major security challenge on the internet is the existence of the large number of compromised machine. This compromised machine used to launch various security attacks including spamming and spreading malware, DDoS, and identify theft. Nowadays all users are moving to a new technology called cloud computing [1], thus security in virtual networked system also an important aspect. The cloud model provides three types of services Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS).

In SaaS the capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email). In PaaS the capability provided to the consumer is to deploy onto the cloud infrastructure his own applications without installing any platform or tools on their local machines. PaaS refers to providing platform layer resources, including operating system support and software development frameworks that can be used to build higher-level services. In IaaS the capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications.

IaaS providers offer their customers the illusion of unlimited compute, network, and storage capacity — often coupled with a 'frictionless' registration process where anyone with a valid credit card can register and immediately begin using cloud services then exploit vulnerabilities and deploy attacks by utilizing the cloud system resources[2]. Since cloud users share the computing resources like file systems, data storage and connected through same switch attacks are more effective in cloud environment [3].

In this paper Secure Intrusion detection and countermeasure selection in virtual networked system incorporate IDM (Intrusion Detection Manager) Algorithm into intrusion detection process. Here the intrusion detection process is carried out in two phases. In first phase the IDM agent on each cloud server monitor and analyze the cloud traffic and in second phase the proposed system decides whether or not the VM put in inspection state. Based on this result Deep Packet inspection or traffic filtering is applied. This approach inspects all suspicious cloud traffic without affecting user's application.

2. RELATED WORK

Network attacks have been discovered to be as varied as the system that they attempt to penetrate. Attacks are known to either be intentional or unintentional and technically competent intruders have been interested in targeting the protocols used for secure communication between networking devices. But as the intruders increase, the network experts are deriving many techniques in preventing attackers from accessing company networks.

SPOT system developed by Duan et al detects compromised machine by sequentially scanning outgoing messages. SPOT use sequential probability ratio test (SPRT) [4], it minimizes the expected number of observation to reach decision. SPRT is a statistical method; both the false positive and false negative probabilities can be bounded by user defined threshold. S.Roschke et al developed a new alert correlation algorithm based on attack graph [5]. Alert correlation was proposed to analyze alerts and to decrease false positives. Knowledge about the target system or environment is usually necessary for efficient alert correlation [6]. For representing the environment information as well as potential exploits, the existing vulnerabilities and their Attack Graph (AG) is used. Design a correlation algorithm based on AGs that is capable of detecting multiple attack scenarios for forensic analysis. It can be parameterized to adjust the robustness and accuracy [7].
P.Ammann et al proposed a scalable graph based vulnerability analysis; they represent more information explicitly than is necessary for the analyst. Although it is possible to produce attack trees from graph representation. This reduces the complexity of the analysis problem from exponential to polynomial, thereby bringing even very large networks within reach of analysis [8]. MulVAL an attack graph tool developed by X.Ou et al which uses Datalog as its modeling language. One of the important features of MulVAL is the ability to reason about multistage attacks. After an exploit is successfully applied, the reasoning engine must discover how the attacker can further compromise a system. The drawback of this approach is that when a new advisory comes, the scanning will have to be repeated on each host [9]. Lee W. and Stolfo S. and Mok K, a data mining framework for adaptively building Intrusion Detection (ID) models was developed. The central idea of this work is

To utilize auditing programs to extract an extensive set of features that describes each network.

BotHunter which detects compromised machine through malware infection process by correlating the IDS dialog trace in a network. BotHunter [10] tracks the two-way communication flows between internal assets and external entities, developing an evidence trail of data exchanges that match a state-based infection sequence model. BotSniffer , can capture the spatial-temporal correlation in network traffic and utilize statistical algorithms to detect botnets with theoretical bounds on the false positive and false negative rates. In botsniffer flaws are classified into groups based on the server connection. If the flaws within a group reveal behavioral similarity, the corresponding hosts involved are detected as being compromised [11].

Quite different from these methods this paper proposed IDM algorithm for network intrusion detection in virtual networked system. When this procedure is followed, IDM increases the throughput by preventing intruders and maintains the history of intruders and employ reconfigurable virtual network-based countermeasures in order to avoid the masquerading DDoS attacks.

3. SYSTEM DESIGN

3.1 System Design Overview

The proposed framework is shown in fig- 1.IDM-Agent, Network controller ,Attack analyzer ,and Virtual machine profiler are the main component of the proposed system., Network controller ,Attack analyzer ,Virtual machine profiler are located in centralized manner .IDM-Agent is implemented using snort and is connected to the cloud server using an isolated secure channel.

IDM-Agent monitors all the cloud traffic and sent intrusion detection alerts to control center when suspicious traffic is detected. The severity of the alert is analyzed by attack analyzer and select possible countermeasure .The selected countermeasures are initiated through network controller by reconfiguring virtual or physical or virtual OFS [12].

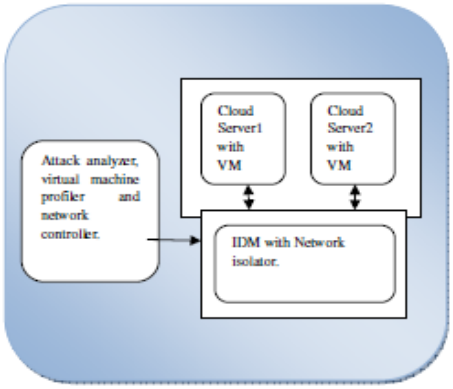

Fig-1.System architecture

3.2 System Components

IDM-Agent

The IDM-Agent scans all the cloud traffic and detects intruders using Intrusion Detection Manager Algorithm .It is implemented using snort for real-time traffic analysis and packet logging on internet protocol (IP).

Algorithm: 1- IDM
Start
Event_type (login, logout)
If (event_Request = login) then
int_mac_a = get_Mac_Address()
If (int_mac_a is in T2) then /*Check Intruders'List*/
(Ignore the request)
else
if (int_mac_a is in T3) then /*Check Authenticated Clients' List*/

(Ignore login request) and
(store int_mac_a in T2)
else
if (int_mac_a is in T5) then /*Check CurrentClient's List*/
(Ignore the request)
else
(Accept the login request) and
(Start communication)
end if
end if
end if
end if
Stop

IDM is intended to detect the alerts in an infrastructure network by maintaining five tables and a timer. The tables are named as account (T1), intruder (T2), authenticated client (T3), unauthenticated client (T4) and client table (T5). The descriptions of the tables are as follows: T1 is for checking the client identity based on their Medium Access Control (MAC) address. T2 contains the MAC address of all the intruders which was detected and spoofed by IDM. T3 consists of MAC addresses of (working) clients who are in the communication process, login time of the client and logout time. Table T4 records the MAC address, login and logout time of client. Table T5, the client table, consists of MAC address and login time of all the clients.

The IDM-Agent then passes all the detected alerts to attack analyzer for further processing.

3.3 VM Profiling

Collect information about VM state, open ports, services running.The information is collected from IDM-Agent, network controller and attack analyzer and is maintained in a VM profile Database and contain complete information about vulnerabilities, alert and traffic.

3.4 Attack Analyzer

Attack analyzer performs alert correlation; calculate severity of alert and countermeasure selection. The optimal countermeasures are selected from countermeasure pool using the countermeasure selection algorithm based on Return of Investment (ROI).

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

The countermeasure which gives least value of ROI is selected as optimal countermeasure.

3.5 Network Controller

Attack analyzer selects optimal countermeasure based on ROI value. The network controller gets this information from network controller and then executes the countermeasures. It will block the VM immediately when the virtual machine is detected as Zombie or the alert is severe. Otherwise put the suspected virtual machine into quarantine mode if the threat level is medium and if threat level is low put it into inspection state.

4. RESULT AND ANALYSIS

Proposed IDM is better in detecting and preventing DDoS Attacks in cloud IaaS when compared with the existing Attack gragh based alert correlation method. It also maintains duplicate IDM, which takes over the network, in case of IDM's failure. IDM updates the duplicate IDM often. When an intruder enters into the network with legitimate client's MAC address, the communication between client and cloud server isdisconnected. So the throughput of the system is dropped during the period of attack. The Fig- 2 clearly shows the rapid fall of throughput during the attack.

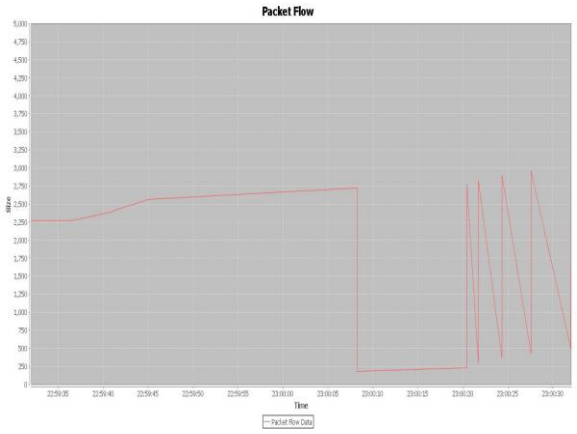

Fig-2 Throughput measurements during DDoS attack

After implementing IDM throughput does not decrease during the period of attack. The performance of the system is increased by maintaining the throughput.

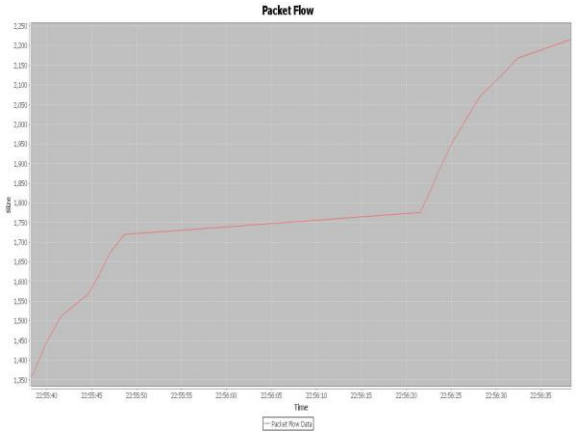

Fig-3 Throughput Measurements after implementing IDM

5. CONCLUSIONS

In this paper, secure intrusion and countermeasure selection in virtual system detects and prevents attacks in cloud IaaS using IDM .It employs a reconfigurable virtual networking approach for attack mitigation. This approach has significant improvement in attack detection during congestion. IDM increases the throughput by preventing intruders and maintains the history of intruders.

REFERENCES

[1]. 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.

[2]. Cloud Security Alliance, "Top Threats to Cloud Computing v1.0," https://cloudsecurityalliance.org/topthreats/csathrats. v1.0.pdf, Mar. 2010.

[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]. 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.

[5]. O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J.M. Wing,"Automated Generation and Analysis of Attack Graphs," Proc. IEEE Symp. Security and Privacy, pp. 273-284, 2002,

[6]. S. Roschke, F. Cheng, and C. Meinel, "A New Alert Correlation Algorithm Based on Attack Graph," Proc. Fourth Int'l Conf. Computational Intelligence in Security for Information Systems, pp. 58-67, 2011.

[7]. X. Ou, W.F. Boyer, and M.A. McQueen, " A Scalable Approach to Attack Graph Generation," Proc. 13th ACM Conf. Computer and Comm. Security (CCS '06), pp. 336-345, 2006.

[8]. P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, graphbased network vulnerability analysis," Proc. 9th ACM Conf. Computer and Comm. Security (CCS '02), pp. 217-224, 2002.

[9]. X. Ou, S. Govindavajhala, and A.W. Appel, "MulVAL: A Logic- Based Network Security Analyzer," Proc. 14th USENIX Security Symp., pp. 113-128, 2005.

[10]. G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "BotHunter: Detecting Malware Infection through IDS-driven Dialog Correlation," Proc. 16th USENIX Security Symp. (SS '07), pp. 12:1-12:16, Aug. 2007.

[11]. G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic," Proc. 15th Ann. Network and Distributed Sytem Security Symp. (NDSS '08), Feb. 2008.

[12]. "Openflow,"http://www.openflow.org/wp/learnmore/, 2012.