International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

General Article

Decision Tree based counter measures for Host-based IDS in Encrypted Network: A General Survey

Swapnali G. Game[Á*], Sachin S. Taware[Ḃ] and M. C. Kshirsagar[Á]

[Á]Department of Computer Engineering, VACOE,Ahmednagar , University of Pune, Maharashtra, India
[Ḃ]Department of E&TC Engineering, PREC, Loni, University of Pune, Maharashtra, India

Accepted 04 January 2014, Available online 01 February 2014, **Vol.4, No.1 (February 2014)**

Abstract

Providing Cloud security is one of challenging issues that has attracted a lot of research and development effort in last few years. In general, in very first step attackers tries to explore vulnerabilities of a cloud system and forces virtual machines to deploy further Distributed Denial-of-Service (DDoS) at a large-scale. DDoS attacks i.e variation of DOS attack usually involve early stage actions such as scanning low-frequency vulnerabilities, multistep exploitation and compromising identified vulnerable virtual machines as zombies and then at last achieving DDoS attacks through the compromised zombies. Within the cloud system, mainly in Infrastructure as a Service (IaaS) clouds, detecting zombie exploration attacks is extremely difficult. The main reason behind this may be that the cloud users may install vulnerable applications on their virtual machines. So that the security of the virtual machines get compromised. To avoid vulnerable virtual machines from being compromised in the cloud there is need of a multiphase distributed vulnerability detection, measurement, countermeasure selection mechanism. The security mechanism can be built based on attack graph-based analytical models and reconfigurable virtual network-based counter measures.

Keywords: *Network Security, cloud computing, zombie detection, intrusion detection*

1. Introduction

A recent Cloud Security Alliance (CSA) survey concludes that among all security issues, Data Breaches is considered as a top security threat, in which data loss and data leakage are serious threats to cloud computing. But the main reason behind this is losing control over the system, in which attackers tries to exploit vulnerabilities in clouds and uses cloud system resources to deploy attacks. Cloud users usually have the privilege to control software installed on their virtual machines, that installed software may affect the system security and can violate the service level agreement (SLA). Cloud users can install vulnerable software on their VMs unknowingly, which essentially contributes to create loopholes in cloud security. The challenge before cloud service provider (CSP) is to establish an effective vulnerability or attack detection and response system for accurately identifying attacks and minimizing the impact of security breach to cloud users.

In a cloud system especially in IAAS, the infrastructure is shared by millions of users, misuse and wicked use of the shared infrastructure benefits attackers to exploit vulnerabilities of the cloud and use its resource to deploy attacks and use compromised part of the system to penetrate whole system. Such attacks are more effective in the cloud environment because cloud users usually share computing resources, e.g., many resources are being connected through the same switch, sharing file systems along with the same data storage, even with potential attackers. The similar setup for VMs in the cloud, e.g., VM OS, virtualization techniques, various installed software which may be vulnerable, networking, and so on, attracts attackers to compromise multiple VMs.

It is usual practice to implement a firewall or a security policy, but experience has shown these to be dramatically insufficient. Simply, denial-of-service attacks are attacks to prevent users of a cloud service from being able to access their data or their applications. By forcing the zombie VMs to consume inordinate amounts of finite system resources such as processor power, memory, disk space or network bandwidth, the attacker or attackers, as is the case in distributed denial-of-service (DDoS) attacks causes an intolerable system slowdown and leaves all of the legitimate service users confused and angry as to why the service isn't responding. While DDoS attacks take advantage of vulnerabilities in web servers, databases, or other cloud resources which allows a malicious individual to take out an application.

2. Existing Methods and Tools

In this section, we discuss literatures of several highly related research areas to IDS, Zombie detection and prevention techniques, their drawbacks.

A considerable amount of research has been done towards detecting malicious behavior. Detecting malicious

*Corresponding author: **Swapnali G. Game**

behavior has been well explored by Duan discusses detection of compromised machines that have been chosen to serve as spam zombies. Their approach called as SPOT, is based on sequentially scanning outgoing messages while employing a statistical method Sequential Probability Ratio Test (SPRT), to quickly determine whether a host has been compromised or not. Bianchi detects compromised machines by comparing images of physical memory taken from similar machines to identify differences associated with rootkit infections. Porras detects compromised machines through malware infection process which has a number of well-defined stages that allow correlating the intrusion alarms triggered by inbound traffic with resulting outgoing communication patterns. Zhang first exploits uniform spatial-temporal behavior characteristics of compromised machines and then detect zombies by grouping flows by considering server connections and searching for similar behavior in the flow. Each path in an attack graph is a series of exploits or actions that leads to an undesirable state. We can say an undesirable state is a state where the intruder has obtained administrative access to a critical host. Firstly, scanning tools determine vulnerabilities of individual hosts. Then the analyst produces an attack graph using local vulnerability information along with other information about the network like connectivity between hosts [9]. There are many automation tools are available to construct attack graph. Sheyner constructs attack graph using a technique based on Binary Decision Diagrams (BDDs) and modified symbolic model checking NuSMV. Although their model can generate all possible attack paths, the scalability is a big issue for their solution. Amman considered the assumption of monotonicity, which explains that the precondition of a given exploit is never invalidated by the successful application of another exploit. It means attackers never need to backtrack. With this assumption, it becomes possible to obtain a scalable and concise graph representation for encoding attack tree. Ou presents MulVAL (Multihost, multistage Vulnerability Analysis), a framework for modelling the interaction of software bugs with system and network configurations. MulVAL uses Datalog as its modelling language. MulVAL comprises a scanner which tests a machine for vulnerable software. MulVAL aims at detecting potential attack paths before an attack happens. But to provide the security assessment and alert correlation features we can modify and extend MulVAL's attack graph structure. Ingols describes substantial enhancements to the NetSPA attack graph system required to model additional present-day threats and countermeasures like host-based vulnerability scans, intrusion prevention systems, proxy firewalls and personal firewalls.

The aim of Intrusion Detection is Detecting and reacting to an attack. But the current solution by IDS as well as firewall does not work very well in real life. For any IDS implementations the large volume of raw alerts from IDS and false alarms are two major problems. For signature-based IDSs (one of the method of IDS) there will be gag between a new threat discovery and its signature being used by the IDS. But in meanwhile the IDS will be unable to identify the threat whose signature is not available with current IDS. Alert correction tool plays an important role in identify the source or target of the intrusion in the network and also specially to detect multistep attack.

Many attack graph-based alert correlation techniques have been proposed recently. Roschke proposed an AG based correlation algorithm that overcomes the limitations in applying the nested loop-based correlation methods and proposed a QG called queue graph approach to remove this limitation. The algorithm is able to identify multiple attack scenarios of the same anatomy by using an attack graph. Once any exploit is examined QG is used to trace alerts matching each exploit in the attack graph. But the algorithm needs some computing power to consume and algorithm needs to be tested using larger data sets. Wang extend the basic QG approach to a unified method to hypothesize missing alerts and to predict future alerts and propose a compact representation for the result of alert correlation. But the limitations of this method are overcome in.

Once we know the possible attack scenarios, selecting and then applying countermeasure is the next important step. Selecting optimal countermeasures depends on attack path and cost benefit analysis so that final solution cost can be optimal as much as possible. Poolsappasit proposed a Bayesian attack graph (BAG) model of the network which enables to better understand the causal relationships between pre-conditions, vulnerability exploitations, and post-conditions. He proposed a genetic algorithm capable of performing both single and multi-objective optimization of the system administrator's objectives. Using a BAG, the system administrator performs risk assessment and risk mitigation and uses genetic algorithm for giving solution to the countermeasure optimization problem. Roy et al. proposed an attack countermeasure tree (ACT) which is considering both attacks and its countermeasures. He used greedy and branch and bound techniques to minimizing the number of countermeasures. This approach aims for minimizing security investment cost and maximizing the benefit from implementing a certain countermeasure set in the ACT.

3. Guidelines and Recommendations

While implementing the protection model, we are having both the options host-based IDS or network-based IDS. HIDS is appropriate for protecting an individual computer systems and the information it contains as the name itself indicates. However it doesn't provide data security on the network as a whole. Also the security systems take on considerable processing resource of the host like RAM, CPU and storage. NIDS monitor and analyze network traffics on a designated network segment. It can be categorized as knowledge or behavior based. But for knowledge based NIDS, system can generate few false positives, good packets are labelled as bad packets and transmission could be interrupted due to poorly defined signature one more important thing to consider is NIDS is unable to stop encrypted packets of system attack from intruders [19]. To overcome the limitations of HIDS or NIDS we can combine the strength of both systems by

forming hybrid systems that is HIDS.But as per the protection model above discussed we need to work for reducing false alarms and the large volume of raw alerts generated by IDS system. So, as per need we can go for HIDS, NIDS or hybrid IDS.

Multiple tools are available in the market for handling functions of constructing attack graphs, updating attack graphs, selecting optimal cost countermeasures then finally applying selected countermeasures successfully by reducing overall solution cost.

There are plenty of IDS tools are available in the market, for example SNORT & BRO are popular IDS systems available in the market. But Bro is the suitable IDS system for users those are UNIX experts. It plays supporting role to the main IDS system. Bro is a good choice if user wants to customize IDS according to his/her network. As compared to Snort Bro is more effective for Gbps networks. For high speed networks snort is a good choice. Snort mainly focuses on simplicity and performance accuracy. This is the main plus point which makes the snort best choice to run on any operating system. But main parameter of comparison is false alarm rate which significantly affect the overall system performance.

Conclusions

This paper surveys comparative analysis of different components needed to provide cloud security through intrusion detection system and mainly focuses on DDos attacks. By considering the pros and cons of different tools for implementing cloud protection model we can choose the effective one, mainly for attack graph construction, countermeasure selection, and alert correlation models. Existing IDS system does not perform well for encrypted traffic analysis, so we need a better solution for overcoming this drawback.

References

NICE (July/August 2013), Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems, *IEEE transactions on dependable and secure computing*, vol. 10, no. 4.

Cloud Security Alliance (February 2013), The Notorious Nine Cloud Computing Top Threats in 2013, https://downloads.cloudsecurityalliance.org/initiatives/top_threats/The_Notorious_Nine_Cloud_Computing_Top_Threats_in_2013.pdf.

B. Joshi, A. Vijayan, and B. Joshi (), Jan. 2012), Securing Cloud Computing Environment Against DDoS Attacks,Proc. *IEEE Int'l Conf. Computer Comm. and Informatics* (ICCCI '12).

H. Takabi, J.B. Joshi, and G. Ahn (Dec. 2010), Security and Privacy Challenges in Cloud Computing Environments, *IEEE Security and Privacy*, vol. 8, no. 6, pp. 24-31.

Z. Duan, P. Chen, F. Sanchez, Y. Dong, M. Stephenson, and J. Barker (Apr. 2012.), Detecting Spam Zombies by Monitoring Outgoing Messages, *IEEE Trans. Dependable and Secure Computing*, vol. 9, no. 2, pp. 198-210

Antonio Bianchi, Yan Shoshitaishvili, Christopher Kruegel, Giovanni Vigna, Blacksheep (Oct 2012), Detecting Compromised Hosts 16–1,8, Raleigh, North Carolina, USA.

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

G. Gu, J. Zhang, and W. Lee,BotSni_er (Feb. 2008), Detecting Botnet Command and Control Channels in Network Tra_c,Proc. 15th *Ann. Network and Distributed System Security Symp*. (NDSS '08),.

Oleg Sheyner, JeannetteWing, F.S. de Boer et al. (Eds.) (2004), FMCO 2003, LNCS 3188, pp. 344–371.

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

NuSMV (Aug. 2012), A New Symbolic Model Checker,http://afrodite.itc. it:1024/nusmv.

P. Ammann, D. Wijesekera, and S. Kaushik, Scalable (2002), Graph based network vulnerability analysis, Proc. 9th *ACM Conf. Computer and Comm. Security* (CCS '02), pp. 217-224.

X. Ou, S. Govindavajhala, and A.W. Appel, MulVAL (2005), A Logic- Based Network Security Analyzer, *Proc. 14th USENIX Security Symp*, pp. 113-128.

Kyle Ingols, Matthew Chu, Richard Lippmann, Seth Webster, Stephen Boyer, Modeling Modern Network Attacks and Countermeasures Using Attack Graphs

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

L. Wang, A. Liu, and S. Jajodia (Sept. 2006), Using Attack Graphs for Correlating, Hypothesizing, and Predicting Intrusion Alerts, *Computer Comm*., vol. 29, no. 15, pp. 2917-2933,.

N. Poolsappasit, R. Dewri, and I. Ray (Feb. 2012), Dynamic Security Risk Management Using Bayesian Attack Graphs, *IEEE Trans. Dependable and Secure Computing*, vol. 9, no. 1, pp. 61-74,.

A. Roy, D.S. Kim, and K. Trivedi (June 2012),Scalable Optimal Countermeasure Selection Using Implicit Enumeration on Attack Countermeasure Trees,Proc. *IEEE Int'l Conf. Dependable Systems Networks* (DSN '12).

SANS Institute InfoSec Reading Room, Intrusion Detection - Systems for today and tomorrow, Ho, Swee Yenn (George), Version 1.2e.

Pritika Mehra (August 2012), A brief study and comparison of Snort and Bro Open Source Network Intrusion Detection Systems, IJARCCE, Vol. 1, Issue 6..