Track Detection and Countermeasure in Cloud Networks (TDCCN)

[1] **Navya Y.U**, [2] **E G Satish** , [3] **Thara D.K**

[1] Channabasaveshwara Institute of Technology(CIT), Gubbi, Karnataka, India

[2] Nitte Meenakshi Institute of Technology(NMIT), Bangalore, Karnataka, India

[3] Channabasaveshwara Institute of Technology(CIT), Gubbi, Karnataka, India

Abstract- Cloud security is one of most important issues that have 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 TDCCN, which is built on attack graph based analytical models and reconfigurable virtual network-based countermeasures. The proposed framework leverages Open Flow network programming APIs to build a monitor and control plane over distributed programmable virtual switches in order to significantly improve attack detection and mitigate attack consequences. The system and security evaluations demonstrate the efficiency and effectiveness of the proposed solution.
.

***Keywords-* Distributed Denial-of-Service (DDoS), Infrastructure-as-a-Service (IaaS) clouds, network programming APIs.**

1. Introduction

Computer security (Also known as cyber security or IT Security) is information security as applied to computers and networks. The field covers all the processes and mechanisms by which computer-based equipment, information and services are protected from unintended or unauthorized access, change or destruction. Computer security also includes protection from unplanned events and natural disasters. Otherwise, in the computer industry, the term security -- or the phrase computer security -- refers to techniques for ensuring that data stored in a computer cannot be read or compromised by any individuals without authorization. Most computer security measures involve data encryption and passwords. Data encryption is the translation of data into a form that is unintelligible without a deciphering mechanism. A password is a secret word or phrase that gives a user access to a particular program or system.

2. Motivation and Proposed Work

In this article, we propose TDCCN (Intrusion detection and Countermeasure in Cloud networks) to establish a defense-in-depth intrusion detection framework. For better attack detection, TDCCN incorporates attack graph analytical procedures into the intrusion detection processes. We must note that the design of TDCCN does not intend to improve any of the existing intrusion detection algorithms; indeed, TDCCN employs a reconfigurable virtual networking approach to detect and counter the attempts to compromise VMs, thus preventing zombie VMs.

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 infra-structure 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 main aim of this project is to prevent the vulnerable virtual machines from being compromised in the cloud server using multi-phase distributed vulnerability detection, measurement, and countermeasure selection mechanism called TDCCN.

3. Input and Output Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

3.1 Objectives

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

3.2 Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

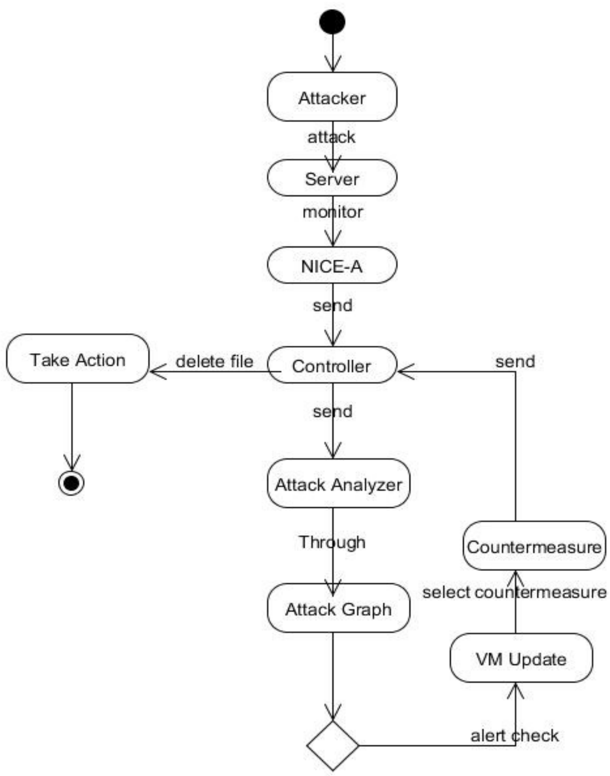

Fig.1 - Flowchart

MODULES:

- TDCCN-A
- VM Profiling
- Attack Analyzer
- Network Controller

TDCCN-A

The TDCCN-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.

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.

Attack Analyzer

The major functions of TDCCN 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. 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.

TDCCN attack graph is constructed based on the following information: *Cloud system information, Virtual network topology and configuration information, Vulnerability information*

Network Controller

The network controller is a key component to support the programmable networking capability to realize the virtual network reconfiguration. In TDCCN, 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 TDCCN, 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 TDCCN and executed by the network controller.

4. Conclusions

In this paper, we presented TDCCN, which is proposed to detect and mitigate collaborative attacks in the cloud virtual networking environment. TDCCN 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 TDCCN 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. TDCCN 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 TDCCN solution by investigating the decentralized network control and attack analysis model based on current study.

References

[1] 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.

[2] 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.

[3] 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

[4] 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.

[5] 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.

[6] 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.

[7] "NuSMV: A new symbolic model checker," http://afrodite.itc.it: 1024/∼nusmv. Aug. 2012.

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

First Author Navya Y.U is a Computer Science Engineer, presently Pursuing M.Tech in the Department of Software Engineering of Channabasaveswara Institute of Technology (CIT Gubbi.She pursued her B.E from Shreedevi Institute of Technology,Tumkur. Currently she is trying to utilize her abilities and Knowledge in the field of wireless sensor networks and Bussiness Intelligence.

Second Author E G Satish is a Computer Science Engineer,Presently working as an Assistant Professor in the Department of Computer Science & Engineering of Nitte Menakshi Institute of Technology.

Third Author Thara D.K is a computer Science Engineer, presently working as an assistant Professor in the department of ISE of Channabasaveswara Institute of Technology (CIT Gubbi)