ABSTRACT:
Attackers can look at vulnerabilities of a cloud system and compromise virtual machines to organize additional large-scale Distributed Denial-of-Service (DDoS). DDoS attacks generally occupy early stage actions such as multi-step exploitation, low frequency vulnerability scanning and compromising identified vulnerable virtual machines as zombies and lastly DDoS attacks through the compromised zombies. Within the cloud system particularly the Infrastructure-as-a-Service (IaaS) clouds the detection of zombie study attacks is tremendously hard. This is for the reason that cloud users may set up susceptible applications on their virtual machines. To prevent susceptible virtual machines from being compromised in the cloud we suggest a multi-phase distributed vulnerability detection, measurement and countermeasure selection method called NICE which is built on attack graph based on analytical models and reconfigurable virtual network-based countermeasures. The proposed framework leverages Open Flow network programming APIs to assemble a monitor and control plane over dispersed programmable virtual switches in order to considerably pick up attack detection and mitigate attack consequences.


INTRODUCTION:
The objective is to avoid the susceptible virtual machines from being compromised in the cloud server using multi-phase disseminated vulnerability detection, measurement and countermeasure selection mechanism called NICE. The latest Cloud Security Alliance (CSA) study shows that amid all security issues, abuse and describable use of cloud computing is measured as the top security hazard in which attackers can use vulnerabilities in clouds and make use of cloud system possessions to deploy attacks. Conversely patching known security holes in cloud data centers where cloud users typically have the freedom to control software installed on their managed VMs, may not work efficiently and can infringe the Service Level Agreement (SLA). In addition cloud users can install vulnerable software on their VMs which fundamentally contributes to loophole in cloud security. The challenge is to create an effectual vulnerability/attack detection and response system for precisely recognizing attacks and reducing the impact of security breach to cloud users. In a cloud system where the infra-structure is communal 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 efficient in the cloud environment since cloud users generally share computing resources.

RELATED WORK:
An attack graph is capable to symbolize a sequence of exploits called atomic attacks that direct to an undesirable state. There are several automation tools to build attack graph. A proposed method based on a customized symbolic replica checking and Binary Decision Diagrams (BDDs) to build attack graph. Their model can produce all possible attack paths. On the other hand the scalability is a huge problem for this solution. P. Ammann introduced the assumption of monotonicity which declares that the precondition of a given exploit is never invalidating by the thriving application of another exploit. In other words attackers never need to back down. Intrusion Detection System (IDS) and firewall are broadly used to check and notice distrustful events in the network. Conversely the false alarms and the large volume of raw alerts from IDS are two major troubles for any IDS implementations. In order to recognize the source or target of the interruption in the network especially to perceive multi-step attack the alert correlation is a must-have tool. The primary goal of alert correlation is to afford system support for a global and compacted view of network attacks by analyzing raw alerts.

EXISTING METHOD:
Cloud users can install vulnerable software on their VMs which basically supply to loopholes in cloud security. The challenge is to set up a successful vulnerability/attack detection and response system for perfectly 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 immoral use of the shared infrastructure benefits attackers to exploit vulnerabilities of the cloud and use its reserve to deploy attacks in more capable ways. Such attacks are more valuable 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 system for VMs in the cloud e.g., virtualization techniques VM OS installed vulnerable software, networking, etc. attracts attackers to compromise multiple VMs.

DISADVANTAGES:

No detection and prevention framework in a virtual networking environment. Doesn’t maintain accuracy in the attack detection from attackers.

PROPOSED METHOD:

Proposed Network Intrusion detection and Countermeasure (NICE) selection in virtual network systems to found a defense-in-depth intrusion detection framework. For better attack detection NICE integrate attack graph analytical measures into the intrusion detection processes. We must note that the plan of NICE does not aim to progress any of the existing intrusion detection algorithms certainly NICE employs a reconfigurable virtual networking approach to detect and oppose the attempts to compromise VMs thus preventing zombie VMs.

ADVANTAGES:

NICE is a latest multi-phase distributed network intrusion detection and deterrence framework in a virtual networking environment that confines and examines apprehensive cloud traffic without disrupting users’ applications and cloud services. NICE integrates a software switching solution to quarantine and scrutinize suspicious VMs for further examination and protection. Through programmable network approaches NICE can progress the attack detection possibility and develop the resiliency to VM utilization attack without interrupting existing normal cloud services. NICE employs a novel attack graph approach for attack detection and avoidance by correlating attack behavior and also proposes effective countermeasures. NICE optimizes the implementation on cloud servers to diminish resource consumption.

SYSTEM ARCHITECTURE:

NICE-A:

The NICE-A is a Network-based Intrusion Detection System (NIDS) agent installed in each cloud server. It examines the traffic going through the bridges that manage all the traffic among VMs and in/out from the physical cloud servers. It will inhale a mirroring port on each virtual bridge in the Open v Switch. Each bridge forms an inaccessible subnet in the virtual network and joins 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 competent to scan the traffic in cloud server because all traffic in the cloud server desires go through it. However our design is autonomous to the installed VM. The false alarm rate could be condensed through our architecture design.

VM PROFILING:

Virtual machines in the cloud can be outlined to get accurate 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 acquaintance of services running on a VM so as to validate the authenticity of alerts pertaining to that VM. An attacker can use port scanning program to carry out a strong assessment 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 important role in influencing how vulnerable the VM is. All these factors mutually will form the VM profile. VM profiles are sustaining in a database and contain all-inclusive information about vulnerabilities, alert and traffic.

**ATTACK ANALYZER:**

The main functions of NICE system are performed by attack analyzer which contains procedures such as attack graph construction and update, alert correlation and countermeasure selection. The procedure 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 holds alert correlation and analysis operations. This component has two major functions which construct Alert Correlation Graph (ACG), offers threat information and suitable countermeasures to network controller for virtual network reconfiguration. NICE attack graph is created based on the Cloud system information, Virtual network topology and configuration information, Vulnerability information.

**ALGORITHM USED:**

**Algorithm 1 Alert Correlation**

Require: alert \(a\), SAG, ACG
1. if \(a\) is a new alert then
2. create node \(a\) in ACG
3. \(n_1 \leftarrow n_2 \in \text{map}(a)\)
4. for all \(n_2 \in \text{parent}(n_1)\) do
5. create edge \((n_2, \text{alert}, a)\)
6. for all \(S_i\) containing \(a\) do
7. if \(a\) is the last element in \(S_i\) then
8. append \(a\) to \(S_i\)
9. else
10. create path \(S_i+1 = \text{subvec}(S_i,0,a)\)
11. end if
12. end for
13. add \(a\) to \(n_1\_\text{alert}\)
14. end for
15. end if
16. return \(S\)

**HOST BASED INTRSUION EVALUATION**

1) Log monitor
Monitoring the log file, once the log change, log monitor will send events to the log analyzer immediately.

Generally, we need to monitor three kinds of event logs: application log, security log and system log. We can add three XML nodes in the following configuration file.
The node “localfile” represents the local file when system initialization. The node “location” represents file path in the disk. The node

---

**NETWORK CONTROLLER:**

The network controller is a key constituent to hold up the programmable networking competence to comprehend the virtual network reconfiguration. In NICE we integrated the control functions for both OVS and OFS into the network controller that permits the cloud system to set security/filtering rules in an integrated and inclusive manner. The network controller is answerable for collecting network information of current Open Flow network and affords input to the attack analyzer to build 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 pertaining the countermeasure from attack analyzer. Based on VM Security Index and severity of alert countermeasures are selected by NICE and carried out by the network controller.
“log_format” represents what type of the log. Log type includes event log, firewall log, SQL log. When initialize the HIDS, it will automatically load the above log files that need to be monitored. When finished the initialization work, the HIDS will open a demon, and the demon will check every log files to find whether there is changes in the log file. If there really exits a change, then the demon will report to the log analyzer.

2) System resources monitor
Monitoring the use of system resources, and sends the status of the system resources utilization to the system resources analyzer at regular time.

3) Connector
The connector is responsible for receiving messages from log monitor and system resources monitor, and sending these messages to log analyzer and system resources analyzer.

4) Log analyzer
Receiving events from the log monitor, match with the rule base to determine whether there is invasion, if there is invasion occurrence, report to the active response unit.

5) System resources analyzer
Receiving events form the system resources monitor, to calculate whether the abnormal state of current resources use and thus to determine whether the status is invaded, if it find there is invasion, report to the active response unit.

6) Active response unit
Receiving events from the log analyzer and system resources analyzer, decided to perform what kind of operation. Usually, the normal operations include notifying users, auditing, disconnecting from network and so on.

7) Audit database
Recording the entire process of intrusion detection, and the attack situation, prepare for use when necessary.

EXPERIMENTAL RESULTS:

The presentation of NICE-A in terms of percentage of productively analyzed packets i.e. the number of the analyzed packets separated by the total number of packets received. The higher this value is more packets this agent can handle. It can be observed from the result that IPS agent exhibits 100% performance because every packet captured by the IPS is cached in the detection agent buffer. We generated 100 consecutive normal packets with the speed of 1 packet per second to test the end-to-end delay of two VMs compared by using NICE-A running in mirroring and proxy modes in DomU and NICE running in Dom0. We record the minimal, average and maximum communication delay in the comparative study. Results show that the delay of proxy-based NICE-A is the highest because every packet has to pass through it.

CONCLUSION:

The proposed solution scrutinizes how to use the programmability of software switches based solutions to progress the detection accurateness and overcome victim utilization phases of collaborative attacks. The system performance assessment express the possibility of NICE and shows that the proposed solution can considerably reduce the risk of the cloud system from being exploited and abused by internal and external attackers. NICE only investigates the network IDS approach to counter zombie explorative attacks. In order to get better detection accuracy, host-based IDS solutions are needed to be incorporated and to cover the whole spectrum of IDS in the cloud system. Additionally will investigate the scalability of the proposed DNIS solution by investigating the decentralized network control and attack analysis model based on the study. 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.

REFERENCES:


Mr.B.Hemanth is a student of Pydah College of Engineering, Patavala. Presently he is pursuing his M.Tech [Computer Science and Engineering] from this college and he received his B.Tech from Adarsh College of Engineering, Affiliated to J N T University Kakinada in the year 2012. His area of interest includes Computer Networks, System security, and Object oriented Programming languages, all current trends and techniques in Computer Science.

Mr.U.Vinod Kumar, well known Author and excellent teacher Received M.Tech (CSE) from Jawaharlal Nehru Technological University, Kakinada, AMIE (Computer Engineering) from The Institute of Engineers and M.Sc (Computer Science) from Andhra University. He is working as Assistant Professor in Pydah College Of Engineering. He is an active member of ISTE and CSI. He has 10 years of teaching experience in various engineering colleges. To his credit couple of publications both national and international conferences/journals. His area of Interest includes Data Warehouse and Data Mining, Mobile Computing, Information security, Computer Networks, Computer Applications.