IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

Distributed Vulnerability Detection, Menstruation and Measure Choice Mechanism for Zombie Detection in Cloud Computing Environment

A. Srinivas[1], A. Sravanthi[2], Shazia[3]

Assistant Professor, Department of CSE, Sri Indu College of Engineering and Technology, Telangana[1]

PG Scholar, Department of CSE, Sri Indu College of Engineering and Technology, Telangana[2,3]

Abstract: Cloud security is one in every of most significant problems that has attracted lots of analysis and development effort in past few years. Notably, attackers will explore vulnerabilities of a cloud system and compromise virtual machines to deploy any large-scale Distributed Denial-of- Service (DDoS). DDoS attacks typically involve early stage actions like multistep exploitation, low-frequency vulnerability scanning, and compromising known vulnerable virtual machines as zombies, and eventually DDoS attacks through the compromised zombies. Among the cloud system, particularly the Infrastructure-as-a-Service (IaaS) clouds, the detection of zombie exploration attacks is extraordinarily troublesome. This is often as a result of cloud users might install vulnerable applications on their virtual machines. To stop vulnerable virtual machines from being compromised within the cloud, I tend to propose a point in time distributed vulnerability detection, menstruation, and measure choice mechanism known as NICE, that is constructed on attack graph-based analytical models and reconfigurable virtual network-based countermeasures. The planned framework leverages Open Flow schedule Apis to make a monitor and management plane over distributed programmable virtual switches to considerably improve attack detection and mitigate attack consequences. The system and Security evaluations demonstrate the potency and effectiveness of the planned answer.

Keywords: Attack Graph, Intrusion Detection, Cloud Computing, Network Security, Zombie Exploration Attacks.

I. INTRODUCTION

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 addressed that protecting "Business continuity and services availability" from service outages is one of the top concerns in cloud computing systems.

A. Motivation

NICE significantly advances the current network IDS/IPS solutions by employing programmable virtual networking approach that allows the system to construct a dynamic reconfigurable IDS system. By using software switching techniques, NICE constructs a mirroring-based traffic capturing framework to minimize the interference on users' traffic compared to traditional bump-in-the-wire (i.e., proxy-based) IDS/IPS. The programmable virtual networking architecture of NICE enables the cloud to establish inspection and quarantine modes for suspicious VMs according to their current vulnerability state in the current SAG. Based on the collective behaviour of VMs in the SAG,NICE can decide appropriate actions, for example, DPI or traffic filtering, on the suspicious VMs. Using this approach, NICE does not need to block traffic flows of a suspicious VM in its early attack stage.

IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

II. RELATED WORK

In this section, we have a tendency to gift literatures of many extremely related analysis areas to NICE, including: Zombie detection and interference, attack graph construction and security analysis, and package outlined networks for attack countermeasures. The area of police work malicious behavior has been well explored. The work by Duan et al. [6] focuses on the detection of compromised machines that are recruited to function spam zombies. Their approach, SPOT, is based on consecutive scanning outgoing messages whereas employing a statistical procedure consecutive likelihood magnitude relation Test (SPRT), to quickly confirm whether or not a number has been compromised. BotHunter [7] detects compromised machines based on the very fact that a radical malware infection process contains a variety of well-defined stages that permit correlating the intrusion alarms triggered by incoming traffic with ensuing outgoing communication patterns. BotSniffer [8] exploits uniform spatial-temporal behavior characteristics of compromised machines to find zombies by grouping flows in line with server connections and searching for similar behavior within the flow. An attack graph is in a position to represent a series of exploits, called atomic attacks, that cause associate degree undesirable state, for example, a state wherever associate degree wrongdoer has obtained body access to a machine. There are units several automation tools to construct attack graph.

Sheyner et al. [9] planned a technique supported a changed symbolic model checking NuSMV [10] and Binary call Diagrams (BDDs) to construct attack graph. Their model will generate all possible attack methods, however, the quantifiability could be a massive issue for this resolution. P. capital of Jordan et al. [11] introduced the assumption of monotonicity, that states that the precondition of a given exploit is rarely nullified by the successful application of another exploit. In alternative words, attackers ne'er have to be compelled to get back. With this assumption, they can get a laconic, ascendable graph illustration for encoding attack tree. Ou et al. [12] planned associate degree attack graph tool referred to as MulVAL, that adopts a logic programming approach and uses Datalog language to model and analyze network system. The attack graph within the MulVAL is constructed by accumulating true facts of the monitored network system. The attack graph construction method can terminate expeditiously as a result of the amount of facts is polynomial in system. to produce the protection assessment and alert correlation options, during this paper, we have a tendency to changed and extended MulVAL's attack graph structure. Intrusion Detection System (IDS) and firewall area unit wide used to monitor and find suspicious events within the network. However, the false alarms and also the giant volume of raw alerts from IDS area unit 2 major issues for any IDS implementations.

To spot the supply or target of the intrusion within the network, particularly to find multistep attack, the alert correction could be a must-have tool. the first goal of alert correlation is to produce system support for a global and condensed read of network attacks by analyzing raw alerts [13]. Many attack graph-based alert correlation techniques have been planned recently. Wang et al. [14] devised associate degree inmemory structure, referred to as queue graph (QG), to trace alerts matching every exploit within the attack graph. However, the implicit correlations during this style create it troublesome to use the correlate alerts within the graph for analysis of comparable attack situations. Roschke et al. [15] planned a changed attack-graph-based correlation rule to form specific correlations solely by matching alerts to specific exploitation nodes within the attack graph with multiple mapping functions, and devised associate degree alert dependencies graph (DG) to clusterelated alerts with multiple regression criteria. every path in DG represents a set of alerts that may be a part of associate degree attack situation. However, their rule concerned all pairs shortest path looking and sorting in decigram that consumes considerable computing power. After knowing the doable attack situations, applying countermeasure is that the next necessary task. Many solutions have been planned to pick out optimum countermeasures based on the chance of the attack path and price benefit analysis.

Roy et al. [16] planned associate degree attack measure tree (ACT) to contemplate attacks and countermeasures together in associate degree attack tree structure. They devised many objective functions supported greedy and branch and sure techniques to reduce the amount of measure, reduce investment price, and maximize the like implementing an explicit measure set. In their style, each measure improvement drawback may well be solved with and while not likelihood assignments to the model.

However, their resolution focuses on a static attack scenario and predefined measure for every attack. Pools appasit et al. [17] planned a theorem attack graph (BAG) to handle dynamic security risk management problem and applied a genetic rule to unravel measure optimization drawback. Our resolution utilizes a brand new network management approach called SDN [18], wherever networking functions are often programmed through package switch and Open Flow protocol [19], plays a significant role during this analysis. Flow based switches, like OVS [5] and Open Flow Switch (OFS) [19], support fine-grained and flow-level management for packet change [20]. With the assistance of the central controller, all Open Flow- based switches are often monitored and organized. We have a tendency to profit of flow-based change (OVS) and network controller to use the chosen network countermeasures in our resolution.

IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

III. MODULES

A. Nice Model

It shows the NICE framework within one cloud server cluster. Major components in this framework are distributed and light-weighted NICE-A on each physical cloud server, a network controller, a VM profiling server, and an attack analyzer. The latter three components are located in a centralized control center connected to software switches on each cloud server (i.e., virtual switches built on one or multiple Linux software bridges). NICE-A is a software agent implemented in each cloud server connected to the control center through a dedicated and isolated secure channel, which is separated from the normal data packets using Open Flow tunneling or VLAN approaches. The network controller is responsible for deploying attack countermeasures based on decisions made by the attack analyzer.

B. Threat Model

In our attack model, we assume that an attacker can be located either outside or inside of the virtual networking system. The attacker's primary goal is to exploit vulnerable VMs and compromise them as zombies. Our protection model focuses on virtual-network-based attack detection and reconfiguration solutions to improve the resiliency to zombie explorations as shown in Fig.1. Our work does not involve host-based IDS and does not address how to handle encrypted traffic for attack detections.

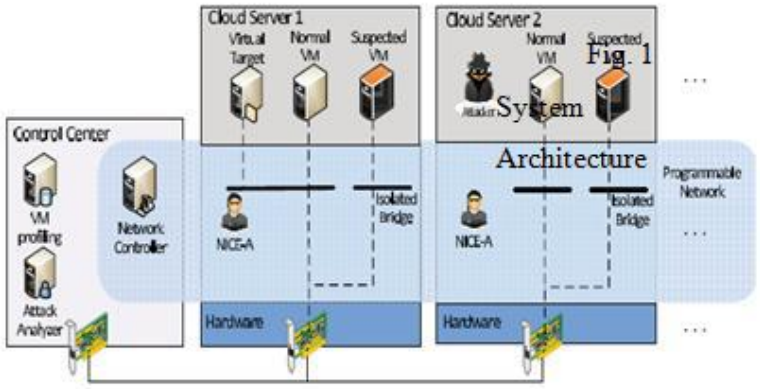

Fig. 1. System Architecture

C. Attack Graph Model

An attack graph could be a modeling tool as an instance all doable multistage, multihost attack methods that area unit crucial to understand threats then to make a decision acceptable countermeasures [22]. In associate degree attack graph, every node represents either precondition or consequence of associate degree exploit. The actions aren't essentially a lively attack as a result of traditional protocol interactions can even be used for attacks. Attack graph is useful in distinguishing potential threats, possible attacks, and notable vulnerabilities during a cloud system. Since the attack graph provides details of all notable vulnerabilities within the system and also the property info, we get a full image of current security scenario of the system, wherever we will predict the doable threats and attacks By correlating detected events or activities. If an event is recognized as a possible attack, we will apply specific countermeasures to mitigate its impact or take actions to forestall it from contaminating the cloud system. To represent the attack and also the results of such actions, we extend the notation of MulVAL logic attack graph as presented by Ou et al. [12] and outline as Scenario Attack Graph (SAG).

Def: 1 (SAG): An SAG is a tuple SAG = (V, E), where
V = NC U NRU ND denotes a set of vertices thatinclude three types namely conjunction node NC to represent exploit,disjunction node ND to denote resultof exploit, and root node NR for showing initial step ofan attack scenario.

- E = EpreUEpost denotes the set of directed edges.

Def: 2 (ACG): An ACG is a three tuple ACG = (A, E, P), where

- A is a set of aggregated alerts.
- E is a set of directed edges representing correlation between two alerts.
- P is a set of paths in ACG

I make a case for a technique for utilizing SAG and ACG along so on predict associate degree attacker's behavior. Correlation_Alert algorithm is followed for each alert detected and returns one or additional methods Si. For each alert alc that's received from the IDS, it's additional to ACG if it doesn't exist. For this new alert alc, the corresponding vertex within the SAG is found by using function map(alc) (line 3). For this vertex in SAG, alert related to its parent

IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

vertex of sort American state is then correlate with the current alert ac (line 5). This creates a brand new set of alerts that belong to a path Si in ACG (line 8) or splits out a brand new path Si+1 from Si with set of Si before the alert a and appends alc to Si+1 (line 10). Within the finish of this rule, the ID of alcis going to be additional to alert attribute of the vertex in SAG. Algorithm one returns a collection of attack methods S in ACG.

Algorithm 1: Correlation_Alert Require: alert alc, SAG, ACG

- if (alc is a new alert) then
- create node alc in ACG
- map(alc)
- for all n2 parent(n1) do
- create edge (n2.alert,alc)
- for all Si containing a do
- if a is the last element in Si then
- append alc to Si
- else
- create path Si+1={subset(Si,a),ac}
- end if
- end for
- add alc to n1.alert
- end for
- end if
- return S

C. VM Protection Model

The VM protection model of NICE consists of a VM profiler, a security indexer, and a state monitor. We specify security index for all the VMs in the network depending upon various factors like connectivity, the number of vulnerabilities present and their impact scores. The impact score of vulnerability, as defined by the CVSS guide, helps to judge the confidentiality, integrity, and availability impact of the vulnerability being exploited. Connectivity metric of a VM is decided by evaluating incoming and outgoing connections. Based on the information gathered from the network controller, VM states can be defined as following:

- **Stable:** There does not exist any known vulnerability on the VM.
- **Vulnerable:** Presence of one or more vulnerabilities on a VM, which remains unexploited.
- **Exploited:** At least one vulnerability has been exploited and the VM is compromised.
- **Zombie**: VM is under control of attacker

IV. NICE SECURITY MEASURE ATTACK MITIGATION AND COUNTER MEASURES

In this section, we have a tendency to gift the ways for choosing the countermeasures for a given attack situation. When vulnerabilities area unit discovered or some VMs area unit known as suspicious, many countermeasures are often taken to restrict attackers' capabilities and it's necessary to differentiate between compromised and suspicious VMs. The measure serves the aim of:

- protective the target VMs from being compromised, and
- Creating attack behavior stand distinguished so the attackers' actions are often known.

A. Security Measure Metrics

The issue of security metrics has attracted a lot of attention and there has been vital effort within the development of quantitative security metrics in recent years. Among different approaches, victimization attack graph because the security metric model for the analysis of security risks [8] could be a good alternative. To assess the network security risk condition for this network configuration, security metrics area unit needed within the attack graph to live risk chance. After an attack graph is built, vulnerability info is included within the graph. For the initial node or external node (i.e., the foundation of the graph, NRND), the priori likelihood is assigned on the chance of a threat supply changing into active and also the problem of the vulnerability to be exploited. We use GV to denote the priori risk likelihood for the foundation node of the graph and frequently the worth of GV is appointed to a high likelihood, e.g., from 0.7 to 1. For the inner exploitation node, every attack-step node two American state can have a likelihood of vulnerability exploitation denoted as GM½e_. GM½e_ is appointed in line with the bottom Score (BS) from Common Vulnerability classification system (CVSS). The BS as shown in (1)[4] is calculated by the impact and exploitability issue of the vulnerability. BS can be directly obtained from National Vulnerability info[6] by checking out the vulnerability CVE id.

IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

IV = 10.41X(1 – (1- C) X(1 - I) X (1 - A), E = 20 X AC X AU X AV ; and f(IV) = 0 if IV = 0, 1:176 otherwise.
The impact worth (IV) is computed from 3 basic parameters of security specifically confidentiality (C), integrity (I), and accessibility (A). The exploitability (E) score consists of access vector (AV), access complexness (AC), and authentication instances (AU). The worth of BS ranges from 0 to 10. In our attack graph, we have a tendency to assign every internal node with its BS worth divided by ten, as shown in The impact worth (IV) is computed from 3 basic parameters of security specifically confidentiality (C), integrity (I), and accessibility (A). The exploitability (E) score consists of access vector (AV), access complexness (AC), and authentication instances (AU). The worth of BS ranges from 0 to 10. In our attack graph, we have a tendency to assign every internal node with its BS worth divided byten. In the attack graph, the relations between exploits will be divisional or conjunctive in line with however they're related through their dependency conditions [29]. Such relationships are often portrayed as probability, where the danger likelihood of current node is set by the connection with its predecessors and their risk probabilities. For any privilege node n two ND with immediate predecessors set W = parent (n).

B. Mitigation Ways

Based on the protection metrics outlined within the previous subsection, NICE is in a position to construct the mitigation strategies in response to detected alerts. First, we define the term measure pool as follows:

Def: 4 (Countermeasure Pool): A measure pool CM = {cm1, cm2, . . . ,cmn} could be a set of countermeasures. Each cm two CM could be a tuple cm = (cost, officiousness, condition, effectiveness), where In general, there area unit several countermeasures which will be applied to the cloud virtual networking system relying on offered measure techniques which will be applied. whilenot losing the generality, many common virtual-networking- based countermeasures area unit listed in Table 1. The optimum measure choice could be a multiobjective optimization drawback, to calculate MIN (impact, cost) and MAX (benefit). In NICE, the network reconfiguration ways in the main involve 2 levels of action: Layer-2 and layer-3. At layer-2, virtual bridges (including tunnels which will be established between 2 bridges) and VLANs area unit main element in cloud's virtual networking system to attach 2 VMs directly. A virtual bridge is associate degree entity that attaches VIFs. Virtual machines on totally different bridges area unit isolated at layer two. VIFs on constant virtual bridge however with totally different VLAN tags cannot communicate to every alternative directly. Based on this layer-2 isolation, NICE will deploy layer-2 network

Sr. No	Countermeasure	Intrusive ness	Cost
1	Traffic isolation	4	2
2	Traffic redirection	3	3
3	Deep Packet Inspection	3	3
4	MAC address change	2	1
5	Creating filtering rules	1	2
6	IP address change	2	1
7	Block Port	4	1
8	Quarantine	5	2
9	Software Patch	5	4
10	Network topology change	0	5
11	Network reconfiguration	0	5

TABLE I: Possible Counter Measure Types Table

In NICE, the network reconfiguration ways in the main involve 2 levels of action: Layer-2 and layer-3. At layer-2, virtual bridges (including tunnels which will be established between 2 bridges) and VLANs area unit main element in cloud's virtual networking system to attach 2 VMs directly. A virtual bridge is associate degree entity that attaches

IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

VIFs. Virtual machines on totally different bridges area unit isolated at layer two. VIFs on constant virtual bridge however with totally different VLAN tags cannot communicate to every alternative directly. Based on this layer-2 isolation, NICE will deploy layer-2 network reconfiguration to isolate suspicious VMs. for instance, vulnerabilities owing to Arp spoofing [30] attacks aren't possible once the suspicious VM is isolated to a special bridge. As a result, this measure disconnects associate degree attack path within the attack graph inflicting the wrongdoer to explore associate degree alternate attack path. Layer-3 reconfiguration is another way to disconnect associate degree attack path.

Through the network controller, the flow table on every OVS or OFS will be changed to vary the topology. We should note that victimization the virtual network reconfiguration approach at lower layer has the advantage therein upper layer applications can expertise least impact. Especially, this approach is just doable once victimization software-switching approach to alter the reconfiguration in a extremely dynamic networking setting. Countermeasures such as traffic isolation are often enforced by utilizing the traffic engineering capabilities of OVS and OFS to restrict the capability and reconfigure the virtual network for a suspicious flow. Once a suspicious activity like network and port scanning is detected within the cloud system, it is necessary to see whether or not the detected activity is indeed malicious or not. For instance, attackers will purposely hide their scanning behavior to forestall the NIDS from distinguishing their actions. In such scenario, changing the network configuration can force the wrongdoer to perform additional explorations, and successively can create their attacking behavior stand out.

B. Measure Choice

Algorithm two presents the way to choose the optimum measure for a given attack situation. Input to the rule is an alert, attack graph G, and a pool of countermeasures CM. The rule starts by choosing the node vAlert that corresponds to the alert generated by a NICE-A. Before selecting the measure, we have a tendency to count the space of vAlert to the target node. If the space is larger than a threshold value, we have a tendency to don't perform measure choice however update the ACG to stay track of alerts within the system (line 3). For the supply node vAlert, all the accessible nodes (including the supply node) area unit collected into a collection T (line 6). as a result of the alert is generated solely once the wrongdoer has performed the action, we have a tendency to set the likelihood of vAlert to one and calculate the new possibilities for all of its kid (downstream) nodes in the set T (lines seven and 8). Now, for all t two T the applicable countermeasures in CM area unit designated and new possibilities are calculated in line with the effectiveness of the chosen countermeasures (lines thirteen and 14). The amendment in likelihood of target node provides the profit for the applied measure using (7).

Algorithm 2: Measure_Choice

Require: Alert; G(E, V), CM

- Let vAlert= Source node of the Alert
- if Distance_to_Targ(vAlert)> threshold then
- ACG_Update
- return
- end if
- Let T =Descendant(vAlert)UvAlert
- □□Set Pr(vAlert) = 1
- □ Calculate_Risk_Prob(T)
- □ Let benefit[|T|,|CM|] = 0 ; 10: for each t T do
- □ for each cm CM do
- □ if cm.condition(t) then
- □ Pr(t)=Pr(t)*(1-cm.effectiveness)
- □ Calculate_Risk_Prob(Descendant(t))
- □ benefit[t, cm] =Pr(target_node).
- □ end if
- □ end for
- □ end for
- □ Let ROI[|T|,|CM|]=0
- □ for each t T do
- □ for each cm CM do
- □ ROI[t,cm]=benefit[t,cm]/cost.cm+intrusiveness+c
- □ end for
- □ end for
- □ SAG_Update and ACG_Update 26: return Optimal_Select_CM(ROI)

IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

IV. PERFORMANCE ANALYSIS

- In this section, we have a tendency to gift the performance analysis of NICE. Our analysis is conducted in 2 directions: The security performance, and also the system computing and network reconfiguration overhead owing to introduced security mechanism as shown in

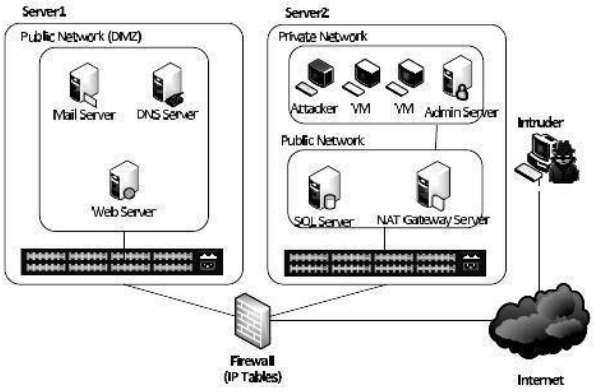

Fig.2. Virtual network topology for security evaluation

A. Security Performance Analysis

To demonstrate the protection performance of NICE, we created a virtual network testing setting consisting of all the conferred elements of NICE.

Setting and Configuration:

To evaluate the protection performance, a demonstrative virtual cloud system consisting of public (public virtual servers) and personal (VMs) virtual domains is established as shown in Fig. 3. Cloud Servers one and a couple of area unit connected to Internet through the external firewall. Within the Demilitarized Zone (DMZ) on Server one, there's one Mail server, one DNS server and one internet server. Public
network on Server two houses SQL server and NAT entree Server. Remote access to VMs within the non-public network is controlled through SSHD (i.e., SSH Daemon) from the NAT entree Server. Table two shows the vulnerabilities gift during this network and Table three shows the corresponding network property that can be explored supported the known vulnerabilities.

Attack Graph and Alert Correlation:

The attack graphs are often generated by utilizing network topology and also the vulnerability info, and it is shown in Fig. 4. Because the attack progresses, the system generates varied alerts which will be associated with the nodes in the attack graph. Creating associate degree attack graph needs information of network connectivity, running services, and their vulnerability information. This info is provided to the attack graph generator because the input. Whenever a brand new vulnerability is discovered or there are a unit changes within the network property and services running through them, the updated information is provided to attack graph generator and previous attack graph is updated to a brand new one. SAG provides information regarding the doable methods that associate degree wrongdoer will follow. ACG serves the aim of confirming attackers' behavior, and helps in determinant false positive and false negative. ACG can even be useful in predicting attackers' next steps.

Measure Choice: To illustrate however NICE works, allow us to contemplate, for instance, an alert is generated for node sixteen (vAlert ¼ 16) once the system detects LICQ Buffer overflow. Once the alert is generated, the accumulative likelihood of node sixteen becomes one because that wrongdoer has already compromised that node. This triggers a amendment in accumulative possibilities of kid nodes of node sixteen. Now, ensuing step is to pick out the countermeasures from the pool of countermeasures CM. If the measure CM4: produce filtering rules is applied to node five and that we assume that this measure has effectiveness of eighty five p.c, the likelihood of node five can change to zero.1164, that causes amendment in likelihood values of all kid nodes of node five thereby accumulating to a decrease of twenty eight.5 p.c for the target node one. Following the same approach for all doable countermeasures which will be applied, the share amendment within the accumulative likelihood of node 1, i.e., profit computed victimization (7) area unit.Apart from shrewd the profit measurements, we also present the analysis supported ROI victimization (8) and represent a comprehensive analysis considering profit, cost, and intrusiveness of measure. Fig. half dozen shows the ROI evaluations for gifted countermeasures. Results show that countermeasures CM2 and CM8 on node five have the maximum profit evaluation; but, their price and intrusiveness scores indicate that they may not be smart candidates for

IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

the optimum measure and ROI evaluation results make sure this. The ROI evaluations demonstrate that CM4 on node five is that the optimum resolution.

Experiment Privately Cloud Setting:

Previously, we have a tendency to conferred associate degree example whereverthe attackers target is VM within the non-public network. For performance analysis and capability take a look at, we have a tendency to extended the configuration in Table three to form another take a look at setting, which incorporates 14 VMs across 3 cloud servers and organized every VM as a target node to form a zealous SAG for every VM. These VMs incorporates Windows (W) and UNIX system (L) machines in the non-public network 172.16.11.0/24, and contains additional number of vulnerabilities associated with their OSes and applications. we have a tendency to created penetration testing scripts with Meta split framework [31] and Armitage [32] as attackers in our take a look at setting. These scripts emulate attackers from different places within the internal and external sources, and launch diversity of attacks supported the vulnerabilities in each VM.

Sr.No.	Protocol	From	To
1	SSHD,I MAP, HTTP, SMTP	Internet	NAT Gateway Server Mail Server Web Server
2.	SQL	Web Server	SQL Server
3.	Basic Network Protocols	NAT Gateway Server	Admin Server VM Group

TABLE II: Virtual Network Connectivity

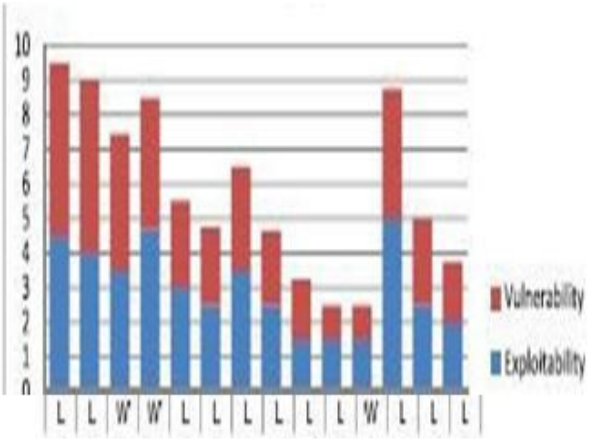

Fig.3. VM Security Index

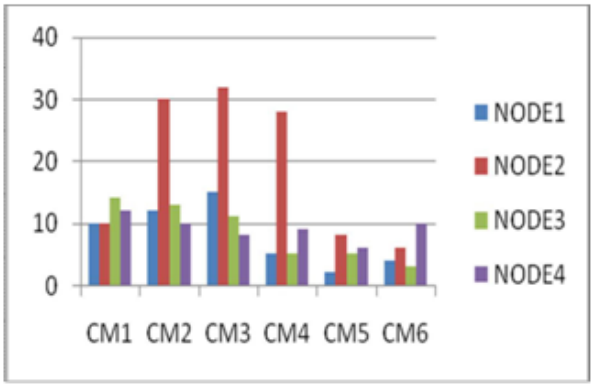

Fig.4.Benefit Evaluation Chart

IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

To evaluate security level of a VM, we have a tendency to outline a VSI to represent the protection level of every VM within the current virtual network setting. This VSI refers to the VEA-bility metric [33] and utilizes 2 parameters that embrace Vulnerability and Exploitability as security metrics for a VM. The VSI worth ranges from zero to ten, wherever lower worth means higher security.
Fig5 compares VSI values before and once applying the countermeasure CM4, i.e., making filtering rules. It shows the percentage amendment in VSI once applying measure on all of the VMs. Applying CM4 avoids vulnerabilities and causes VSI to drop while not interference normal services and ports.

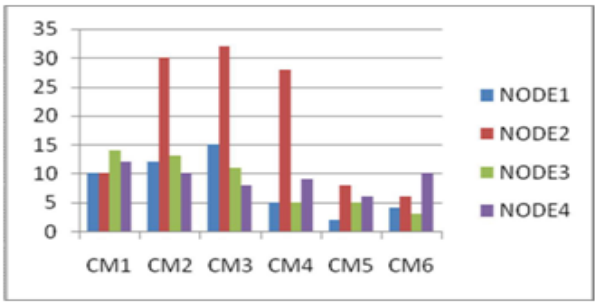

Fig.5. ROI Evaluation Chart

False Alarms:
A cloud system with many nodes can have vast amount of alerts raised by Snort. Not all of those alerts will be relied upon, and an efficient mechanism is required to verify if such alerts have to be compelled to be addressed. Since Snort are often programmed to get alerts with CVE id, one approach that our work provides is to match if the alert is truly related to some vulnerability being exploited. If so, the existence of that vulnerability in SAG implies that the alert is more possible to be a true attack. Thus, the false positive rate will be the chance of the correlate alerts, which will not increase the false positive rate compared to every individual false positive rate. Moreover, we have a tendency to can't keep aside the case of zero-day attack, wherever the vulnerability is discovered by the wrongdoer but isn't detected by vulnerability scanner. In such case, the alert being real are going to be thought to be false, provided thatthere does not exist corresponding node in SAG. Thus, current research doesn't address the way to scale back the false negative rate. it's necessary to notice that vulnerability scanner should be ready to find most up-to-date vulnerabilities and adjust with the newest vulnerability info to scale back the prospect of Zero-day attacks.

B. NICE System Performance
We value system performance to produce steering on how much traffic NICE will handle for one cloud server and use the analysis metric to rescale to an oversized cloud system. During a real cloud system, traffic coming up with is required to run NICE, that is on the far side the scope of this paper. Due to the area limitation, we'll investigate the analysis involving multiple cloud clusters within the future. To demonstrate the feasibleness of our resolution, comparative studies were conducted on many virtualization approaches. we have a tendency to evaluated NICE supported Dom0 and DomU implementations with mirroring-based and proxy based attack detection agents (i.e., NICE-A). In mirror based IDS situation, we have a tendency to established 2 virtual networks in each cloud server: traditional network and watching network. NICE-A is connected to the watching network as shown in Fig.6. Traffic on the conventional network is reflected to the watching network victimization Switched Port instrument (SPAN) approach.

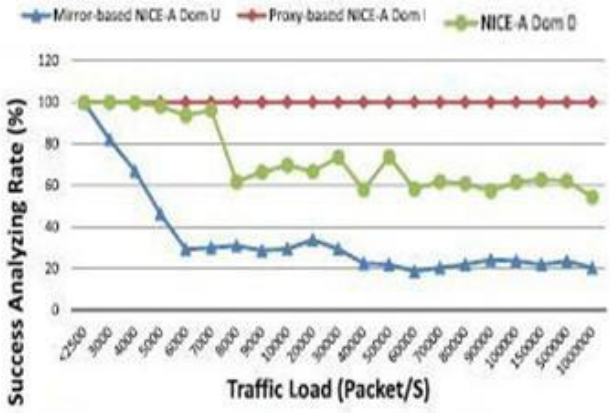

Fig.6. NICE-A success analyzing rate

DOI10.17148/IJARCCE.2017.6639

Within the proxy- based IDS resolution, NICE-A interfaces 2 VMs and also the traffic goes through NICE-A. Additionally, we've got deployed the NICE-A in Dom0 and it removes the traffic duplication operate in mirroring and proxy-based solutions.

NICE-A running in Dom0 is additional economical as a result of it can sniff the traffic directly on the virtual bridge. However, in DomU, the traffic have to be compelled to be duplicated on the VM's VIF, inflicting overhead. once the IDS is running in Intrusion interference System (IPS) mode, it has to intercept all the traffic and perform packet checking, which consumes additional system resources as compared to IDS mode. To demonstrate performance evaluations, weNICE-A running in Dom0 is additional economical as a result of it can sniff the traffic directly on the virtual bridge. However, in DomU, the traffic have to be compelled to be duplicated on the VM's VIF, inflicting overhead. once the IDS is running in Intrusion interference System (IPS) mode, it has to intercept all the traffic and perform packet checking, which consumes additional system resources as compared to IDS mode. To demonstrate performance evaluations.

V. CONCLUSION

In this paper, we have a tendency to confer NICE that is planned to detect and mitigate cooperative attacks within the cloud virtual networking setting. NICE utilizes the attack graph model to conduct attack detection and prediction. The proposed resolution investigates the way to use the programmability of package switches- based solutions to enhance the detection accuracy and defeat victim exploitation phases of cooperative attacks. The system performance analysis demonstrates the feasibleness of NICE and shows that the proposed resolution will considerably scale back the danger of the cloud system from being exploited and abused by internal and external attackers. NICE solely investigates the network IDS approach to counter zombie exploratory attacks. to enhance the detection accuracy, host-based IDS solutions area unit required to be incorporated and to hide the complete spectrum of IDS within the cloud system. This could be investigated within the future work. to boot, as indicated within the paper, we will investigate the quantifiability of the planned NICE resolution by investigating the localized network management and attack analysis model supported current study.

REFERENCES

[1] CoudSercurity Alliance, "Top Threats to Cloud Computingv1.0,"https://cloudsecurityalliance.orgtopthreats/ csathre ats. v1.0.pdf, Mar. 2010.
[2] K. Muralidhar and Dr. N. Geethanjali, "A Comparative Study of Security Concerns in Public and Private Clouds", in Proceedings ofthe NationalConference on High Performance Computing & Simulation (NCHPCS-2013), pp. 66-69
[3] Joshi, A. Vijayan, and B. Joshi, "Securing Cloud Computing Environment against DDoS Attacks,"Proc. IEEEInt'lConf. Computer Comm. and Informatics (ICCCI '12), Jan. 2012
[4] 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.
[5] "OpenvSwitch Project,"http://openvswitch.org,May 2012.
[6] 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.
[7] 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.
[8] 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.
[9] 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,
[10] "NuSMV: A New Symbolic Model Checker," http://afrodite.itc. it:1024/nusmv. Aug. 2012.
[11] P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, graphbasednetworkvulnerability analysis," Proc. 9th ACM Conf. Computer and Comm. Security (CCS '02), pp. 217-224, 2002.
[12] X. Ou, S. Govindavajhala, and A.W. Appel, "MulVAL:A Logic-Based Network Security Analyzer," Proc. 14th USENIX Security Symp.,pp. 113-128, 2005.
[13] R. Sadoddin and A. Ghorbani, "Alert Correlation Survey: Framework and Techniques," Proc. ACM Int'l Conf. Privacy, Security and Trust: Bridge the Gap between PST Technologies and Business Services (PST '06), pp. 37:1-37:10, 2006.
[14] L. Wang, A. Liu, and S. Jajodia, "Using Attack Graphs for Correlating, Hypothesizing, and Predicting Intrusion Alerts," Computer Comm., vol. 29, no. 15, pp. 2917-2933, Sept. 2006.
[15] 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.
[16] Roy, D.S. Kim, and K. Trivedi, "Scalable Optimal Countermeasure Selection Using Implicit Enumeration on Attack Countermeasure
[17] Trees," Proc. IEEE Int'l Conf. Dependable Systems Networks (DSN '12), June 2012.
[18] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic Security Risk Management Using Bayesian Attack Graphs," IEEE Trans. Dependable and Secure Computing, vol. 9, no. 1, pp. 61-74, Feb. 2012.
[19] Open Networking Fundation,"Software- Defined Networking: TheNew Normfor Networks," ONF White Paper, Apr. 2012.
[20] "Openflow," http://www.openflow.org/wp/learnmore/, 2012.
[21] N. McKeown, T. Anderson, H. Balakrishnan,G.Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," SIGCOMM Computer Comm. Rev., vol. 38, no. 2, pp. 69-74, Mar. 2008.
[22] Keller, J. Szefer, J. Rexford, and R.B. Lee, "NoHype: Virtualized Cloud Infrastructure without the Virtualization," Proc. 37th ACM Ann. Int'l Symp. Computer Architecture (ISCA '10), pp. 350-361, June 2010.