International Journal of Engineering Research and General Science Volume 3, Issue 1, January-February, 2015
ISSN 2091-2730

Securing Cloud Network Environment against Intrusion using Sequential Algorithm

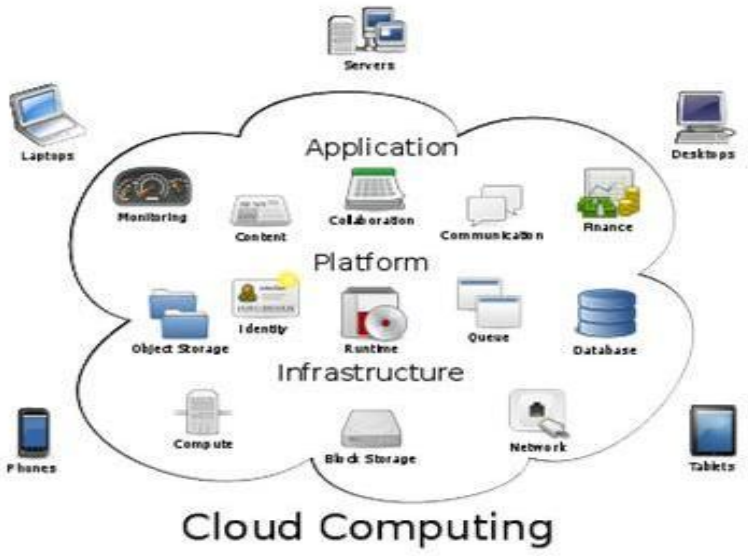
Mr R.Kumar
Assistant Professor, Information Science and Engineering
MVJ College of Engineering, Banglore -67
rkumarmecse@gmail.com
9789234479

Abstract: - Cloud Computing is the newly emerged technology of Distributed Computing System. Cloud Computing user concentrate on API security & provide services to its consumers in multitenant environment into three layers namely, Software as a service, Platform as a service and Infrastructure as a service, with the help of web services. Attackers can explore vulnerabilities of a cloud system and compromise virtual machines to deploy further large-scale Distributed Denial-of-Service (DDoS). Particularly, DDoS attacks usually involve early stage actions such as multistep 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 multiphase distributed vulnerability detection, measurement, and countermeasure selection mechanism called "Securing Cloud Network Environment against Intrusion Using Sequential Algorithm", which is built on attack graph-based analytical models. The proposed framework significantly improves attack detection and mitigate attack consequences.

Keywords: DDoS, Zombies, Attack Graph, Sequential Search, Thresh hold value, Normal Mode, Danger Mode

1. INTRODUCTION

The term cloud is analogical to "Internet". The term cloud computing is based on cloud drawings used in the past to represent telephone networks & later to depict internet in.



Over the years, technology and Internet companies such as Google, Amazon, Microsoft and others, have acquired a considerable expertise in operating large data centers, which are the backbone of their businesses. Their know-how extends beyond physical infrastructure and includes experience with software, e.g., office suites, applications for process management and business intelligence, and best practices in a range of other domains, such as Internet search, maps, email and other communications applications. In cloud computing, these services are hosted in a data center and commercialized, so that a wide range of software applications are offered by the provider as a billable service (Software as a Service, SaaS) and no longer need to be installed on the users PC [5]. 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 [1], in which attackers can exploit vulnerabilities in clouds and utilize cloud systemresources 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 theirVMs, which essentially contributes to loopholes in cloud security. The challenge is to establish an effective vulnerability/attackdetection and response system for accurately identifying attacks and minimizing the impact of security breach to cloud users. In [2] Armbrust et al. Addressed that protecting "Business continuity and services availability" from service outages is one of the top concerns in cloud computing systems. Ina cloud system, where the infrastructure is shared by poten- tially 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 [3]. Such attacks are more effective in the cloud environment because 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 setup for VMs in the cloud, e.g., virtualization techniques, VM OS, installed vulnerable software, networking, and so on, attracts attackers to compromise multiple VMs.

In this paper, we propose Network Intrusion detection and Countermeasure sElection in virtual network systems (NICE) to establish a defense-in-depth intrusion detection framework. For better attack detection, NICE incorporates attack graph analytical procedures into the intrusion detection processes[4]. We must note that the design of NICE does not intend to improve any of the existing intrusion detection algorithms; indeed, NICE employs a reconfigure- able virtual networking approach to detect and counter the attempts to compromise VMs, thus preventing zombie VMs.

In general, NICE includes two main phases:

1) Deploy a lightweight mirroring-based network intrusion detection agent (NICE-A) on eachcloud server to capture and analyze cloud traffic. A NICE-A periodically scans the virtual system vulnerabilities within a cloud server to establish Scenario Attack Graph (SAGs), andthen based on the severity of identified vulnerability toward the colla-borative attack goals, NICE will decide whether or not to put a VM in network inspection state

2) Once a VM enters inspection state, Deep Packet Inspection (DPI) is applied, and/or virtual network reconfigurations can be deployed to the inspecting VM to make the potential attack behaviors prominent.

2 RELATED WORK

2.1 A New Alert Correlation Algorithm Based On Attack Graph :

Intrusion Detection Systems (IDS) are widely deployed in computer networks. As modern attacks are getting more sophisticated and the number of sensors and network nodes grows, the problem of false positives and alert analysis becomes more difficult to solve. 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. For representing the environment information as well as potential exploits, the existing vulnerabilities and their Attack Graph (AG) is used. It is useful for networks to generate an AG and to organize certain vulnerabilities in a reasonable way. In this paper, 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. A formal model of the algorithm is presented and an implementation is tested to analyze the different parameters on a real set of alerts from a local network.

An AG based correlation algorithm is proposed that overcomes the drawbacks of the algorithm described in [9]. It creates only explicit correlations and enables the identification of multiple attack scenarios of the same anatomy. The algorithm consists of a mapping of alerts to AG nodes, the alert aggregation function, a function for building an alert dependency graph, and a function for finding suspicious subsets using the Floyd-Warshall algorithm and the diameter value. In addition to the formal model of the correlation algorithm, multiple possibilities for the node matching and aggregation function analyzed in detail to parameterize the algorithm.

Finally, the capabilities have tested and the influence of the parameters analyzed by using a real data set of alerts generated from network.

2.2 Dynamic Security Risk Management Using Bayesian Attack Graphs

Security risk assessment and mitigation are two vital processes that need to be executed to maintain a productive IT infrastructure. On one hand, models such as attack graphs and attack trees have been proposed to assess the cause-consequence relationships between various network states, while on the other hand, different decision problems have been explored to identify the minimum-cost hardening measures. However, these risk models do not help reason about the causal dependencies between network states. Further, the optimization formulations ignore the issue of resource availability while analyzing a risk model. A risk management framework using Bayesian networks [6] that enable a system administrator to quantify the chances of network
compromise at various levels. It shows how to use this information to develop a security mitigation and management plan. In contrast to other similar models, this risk model lends itself to dynamic analysis during the deployed phase of the network. A multi objective optimization platform provides the administrator with all trade-off information required to make decisions in a resource constrained environment.

The system administrators' dilemma, namely, how to assess the risk in a network system Hence, computing marginal probabilities either for prior or posterior cases, is O(2n) and does not scale very well for a large network. However, there are more efficient algorithms that can give a fair approximation for these probabilities. The administrator should plan to look into how these approximations can be used to speed up the analysis and what the corresponding impact is on the accuracy of the solution. In particular, the evaluation algorithm to include heuristic-based update mechanisms is revised in order to reduce the time required to complete the mitigation analysis, without scarifying the quality of results obtainable. Furthermore, the mitigation process in dynamic situations needs to be improved so that a security administrator can quickly identify the best security response that accounts for all former investments made as part of the static analysis stage.

2.3 Detecting Spam Zombies by Monitoring Outgoing Messages

This scheme focuses on the detection of the compromised machines in a network that are involved in the spamming activities, commonly known as spam zombies. Develop an effective spam zombie detection system named SPOT by monitoring outgoing messages of a network. SPOT is designed based on a powerful statistical tool called Sequential Probability Ratio Test, which has bounded false positive and false negative error rates. The evaluation studies based on a two-month email trace collected in a large U.S. campus network show that SPOT is an effective and efficient system in automatically detecting compromised machines in a network. For example, among the 440 internal IP addresses observed in the email trace, SPOT identifies 132 of them as being associated with compromised machines. Out of the 132 IP addresses identified by SPOT,
126 can be either independently confirmed or highly likely to be compromised. Moreover, only 7 internal IP addresses associated with compromised machines in the trace are missed by SPOT. In addition, it also compare the performance of SPOT with two other spam zombie detection algorithms based on the number and percentage of spam messages originated or forwarded by internal machines, respectively, and show that SPOT outperforms these two detection algorithms.

New scheme developed an effective spam zombie detection system named SPOT by monitoring outgoing messages in a network. SPOT was designed based on a simple and powerful statistical tool named Sequential Probability Ratio Test to detect the compromised machines that are involved in the spamming activities. SPOT has bounded false positive and false negative error rates. It also minimizes the number of required observations to detect a spam zombie. The evaluation studies based on a 2-month email trace collected on the FSU campus network showed that SPOT is an effective and efficient system in automatically detecting compromised machines in a network. In addition, SPOT outperforms two other detection algorithms based on the number and percentage of spam messages sent by an internal machine, respectively.

Intrusion detection system plays an important role in the security and perseverance of active defense system against intruder hostile attacks for any business and IT organization. IDS implementation in cloud computing requires an efficient, scalable and virtualization-based approach. In cloud computing, user data and application is hosted on cloud service provider‟s remote servers and cloud user has a limited control over its data and resources. In such case, the administration of IDS in cloud becomes the responsibility of cloud provider. Although the administrator of cloud IDS should be the user and not the provider of cloud services. In the paper we have proposed an integration solution for central IDS management that can combine and integrate various renowned IDS sensors output reports on a single interface. The intrusion detection message exchange format (IDMEF) standard has been used for communication between different IDS sensors. The authors have suggested the deployment of IDS sensors on separate cloud layers like application layer, system layer and platform layer. Alerts generated are sent to Event Gatherer program. Event

gatherer receives and convert alert messages in IDMEF standard and stores in event data base repository with the help of Sender, Receiver and Handler plug-ins. The analysis component analyzes complex attacks and presents it to user through IDS management system. The authors have proposed an effective cloud IDS management architecture, which could be monitored and administered by the cloud user. They have provided a central IDS management system based on different sensors using IDMEF standard for communication and monitored by cloud user.

3. EXSISTING WORK:

Cloud users can install vulnerable software on their VMs, which essentially contributes to loopholes in cloud security. Application DOS attack[6],[7], which aims at disrupting application service rather than depleting the network resource, has emerged as a larger threat to network services, compared to the classic DOS attack. Owing to its high similarity to legitimate traffic and much lower launching overhead than classic DDOS attack, this new assault type cannot be efficiently detected or prevented by existing detection In a cloud system where the infrastructure 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[4]. 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 similar setup for VMs in the cloud, e.g., virtualization techniques, VM OS, installed vulnerable software, networking, etc., attracts attackers to compromise multiple VMs.

3.1 Drawbacks:

Each request is verified for dos, once it is posted to server. Sometimes continues verification or checking of some request or every request in sequence manner can increase the server work load.

Due to this existing system leads to failure randomly

4. PROPOSED WORK

To identify application IDS attack, proposed a novel group testing (GT)-based approach deployed on back-end servers, which not only offers a theoretical method to obtain short detection delay and low false positive/negative rate, but also provides an Underlying framework against general network attacks. More specifically, first extend classic GT model with size constraints for practice purposes, and then redistribute the client service requests to multiple virtual servers embedded within each back-end server machine, according to specific testing matrices.

Based on this framework, propose a two-mode detection mechanism using some dynamic thresholds to efficiently identify the attackers. The focus of this work lies in the detection algorithms proposed and the corresponding theoretical complexity analysis.

Advantages

- Every request or all the requests to the server are parallel checked for IDS by using GT.
- Due to this server performance is not affected and reduces the workload of Server.

4.1 Architecture Design

:

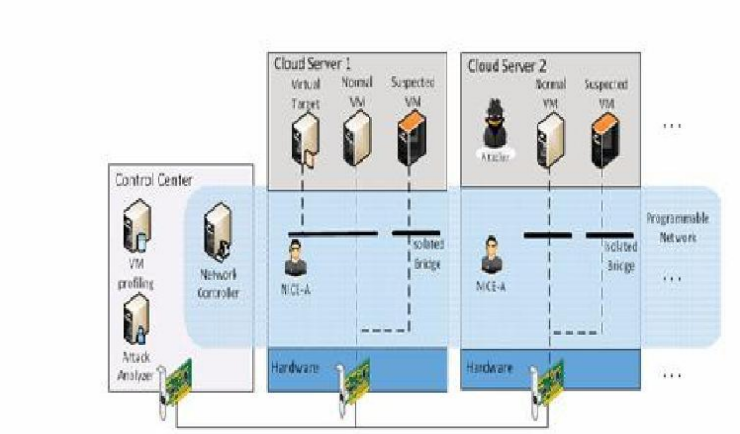
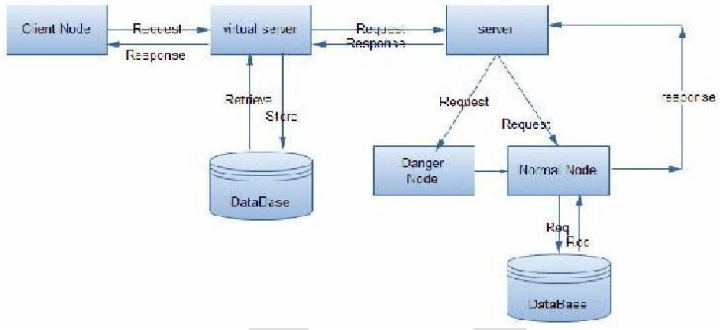


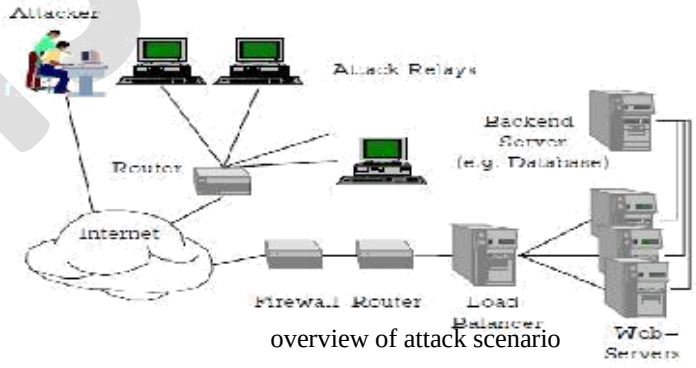
Fig. 1. NICE architecture within one cloud server cluster.



Architectural Design

4.2 Node Analysis:

We consider the case each client provided with a non spoofed ID which is used in identifying client during our detection period. Attackers are assumed to launch the application service request either at high interarriaval rate or high workload or even both. By periodically monitoring the average response time to service requests and comparing them with the specific threshold values fetched from a legitimate profile each virtual server is associated with a negative or positive outcome by this we can identify a attacker from the pool of legitimate users.

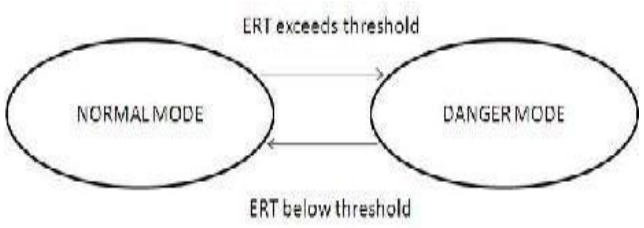


overview of attack scenario

The above figure describes the architeture of web application and its infrastrucure .the requests are generated from users which

consists of both legitamate and attackers are send to the proxy server via router The front end proxy server works as load balancer servers and distributes the requests to the back end servers depending on their usage The backend server cylces between two states which are refered as NORMAL mode and DANGER mode

If the estimated response time (ERT) of any back end server exceeds profile based threshold the system transfer to danger mode



Two state diagram of a system

The ERT value can be calculated using the formulae

$$ERT= (1-\alpha)\ ERT+ \alpha ART$$

If any virtual server has ERT>μ +4σ (μ and σ refer as expected value and standard deviation of the ART distribution). The backend server is probably under attack and it transferred to danger mode for detection. After the detection it is returned to the normal mode.

4.3 Algorithm Description

Sequential algorithm has proposed to block the attackers. It will search sequentially and block the attacker node in transmission path.

Sequential Algorithm

Pseudo-code

```
const
 MAX = ….;
type
 itemType = …;
 listType = array [1..MAX] of integer;
function int position (list: listType; item: itemType)
/* this function searches the array list with the bounds 0 to MAX for the value item. The
function returns the position in list of item, if item is found */
var
 i : integer;
begin
 for I := 0 to MAX do
```

```
 if list[i] = item then

 position := I;

 end for;

end;
```

4.3.1 Explanation

It is the easiest to implement and the most frequently used search algorithm in practice. Unfortunately the sequential search is also the most ineffective searching algorithm. However, it is so commonly used that it is appropriate to consider several ways to optimize it. In general the sequential search, also called linear search, is the method of consecutively check every value in a list until find the desired one.

4.3.2 Implementation

There is no guarantee about the order of elements in the list if insertions have been under a user's control.
Search starts at the first element in the list and continues until either the item is found in the list or entire list is searched .When searching the data set for a particular item, compare the key of the item with the keys of the items in the data set .For each iteration in the loop, the search item is compared with an element in the list, and a few other statements are executed. Loop terminates when search item is found in list. Therefore the execution of the other statement in loop is directly related to the outcome of the key comparisons. If the target is the kth element in the list k comparisons are made.

5. CONCLUSION

A novel technique for detecting application DOS attack by means of a new constraint-based group testing model. Motivated by classic GT methods, three detection algorithms were proposed and a system based on these algorithms was introduced. Theoretical analysis demonstrated the outstanding performance of this system in terms of low detection latency and false positive/negative rate. Focus is to apply group testing principles to application DOS attacks, and provide an underlying framework for the detection against a general case of network assaults, where malicious requests are indistinguishable from normal ones.

For the future work, investigate the potentials of this scheme and improve this proposed system to enhance the detection efficiency. Some possible directions for this can be:

- The sequential algorithm can be adjusted to avoid the requirement of isolating attackers
- Group testing will performed to identify whether the virtual server has affected by the attacker based on the performance of the virtual server.
- The users are identified and placed on appropriate mode by selecting either normal mode dangerous mode.

6. REFERENCES

[1] Coud Sercurity Alliance, "Top Threats to Cloud Computingv1.0,"https://cloudsecurityalliance.org/topthreat s/csathreats. v1.0.pdf, Mar. 2010.

[2] 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.

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

[4] Chun-Jen Chung, , Pankaj Khatkar, Tianyi Xing, Jeongkeun Lee, and Dijiang Huang," NICE: Network Intrusion

Detection and Countermeasure Selection in Virtual Network Systems

[5] Ms. Parag K. Shelke, Ms. Sneha Sontakke, Dr. A. D. Gawande,"Intrusion detection system for cloud computing". [6] P.Ravi kiran verma,D.Sai Krishna,"Application of denial of service Attacks Detection Using Group Testing Based Approach"Vol2(2).

[6]Sikakolanu Hareesh Kumar,U.Nanaji," Detecting application Denial of Service Attacks :A Dynamic Group Testing Based Approach, Vol2(6).

[7] D.Asha,M.Chitra,"Securing cloud from ddos attacks using intrusion detection system in virtual machine".

[8]. M.T. Thai, Y. Xuan, I. Shin, and T. Znati, "On Detection of Malicious Users Using Group Testing Techniques," Proc. Int'l Conf. Distributed Computing Systems (ICDCS), 2008