International Journal of Power Control and Computation(IJPCSC)
Vol 7. No.1 – 2015 Pp. 35-40
©gopalax Journals, Singapore
available at : www.ijcns.com ISSN: 0976-268X

DETECTION OF INTRUSION AND PRESERVING PRIVACY FOR DATA IN CLOUD STORAGE SYSTEM

[1]P. SANTHOSH KUMAR, [2]K.VENNILA and [3]Dr. LATHA PARTHIBAN
[1]Research Scholar, [2]B.E - CSE , [3]Assistant Professor
[1]Sathyabama University, Chennai, India
[2]Sri Aravindar Engineering College, Pondicherry, India
[3]Pondicherry Community College,Pondicherry

ABSTRACT

Cloud Computing plays a vital role in IT field which develop the field of computer in fast manner. Cloud Computing is nothing but sharing of resources to their clients in efficient way. It works under the concept of virtualization and by three different types of service providers such as SaaS, IaaS and PaaS. In cloud computing we come across some problems in security and data stored in the cloud server. 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. Our work does not involve host-based IDS and does not address how to handle encrypted traffic for attack detections. Our proposed solution can be deployed in an Infrastructure-as-a-Service (IaaS) cloud networking system, and we assume that the Cloud Service Provider (CSP) is benign. We also assume that cloud service users are free to install whatever operating systems or applications they want, even if such action may introduce vulnerabilities to their controlled VMs. Physical security of cloud server is out of scope of this paper. We assume that the hypervisor is secure and free of any vulnerability.

Key Terms – Cloud computing, Service providers, attackers and vulnerability

1. INTRODUCTION

In this paper, 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.

2. PROBLEM DEFINITION

In existing, 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. We propose NICE (Network Intrusion detection and Countermeasure selection in virtual network systems) to establish a defense-in-depth intrusion detection framework. For better attack detection, NICE incorporates attack graph analytical procedures into the intrusion detection processes. We must note that the design of NICE does not intend to improve any of the existing intrusion detection algorithms; indeed, NICE employs a reconfigurable virtual networking approach to detect and counter the attempts to compromise VMs, thus preventing zombie VMs. In general, NICE includes two main phases: deploy a lightweight mirroring-based network intrusion detection agent (NICE-A) on each cloud server to capture and analyze cloud traffic.

3. METHODOLOGIES

3.1 Intrusion Detection Model

Intrusion Detection Model plays an important role for the user moves data to cloud. This module has created for the security purpose. In this model we have to enter login user name and password. It will check username and password is match or not (valid username and valid password). If we enter any invalid username or password we can't enter into a cloud to use Cloud data it will shows error message. So we are preventing from unauthorized user entering into the cloud to use the cloud page.

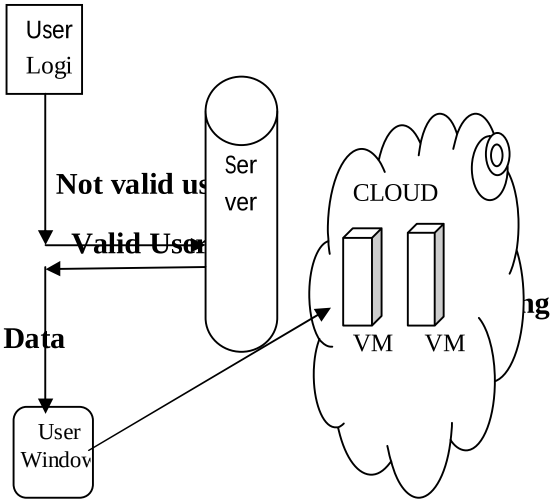


Fig-1: Intrusion Detection Model

3.2 Attack Graph Model

This is the second module of our project in this with the advent of web applications, An attack graph is a modeling tool to illustrate all possible multi-stage, multi-host attack paths that are crucial to understand threats and then to decide appropriate countermeasures . In an attack graph, each node represents either precondition or consequence of an exploit. The actions are not necessarily an active attack since normal protocol interactions can also be used for attacks. Attack graph is helpful in identifying potential threats, possible attacks and known vulnerabilities in a cloud system. Since the attack graph provides details of all known vulnerabilities in the system and the connectivity information, we get a whole picture of current security situation of the system where we can predict the possible threats and attacks by correlating detected events or activities.

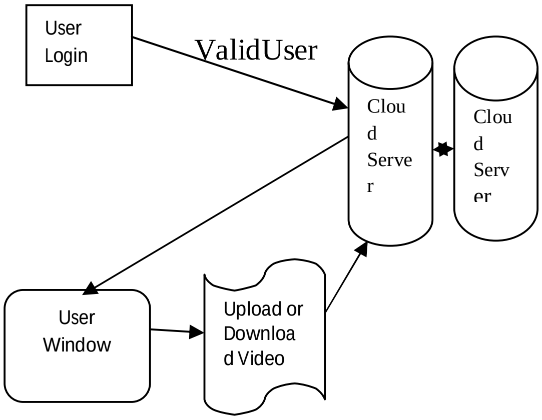


Fig-2: Attack Graph Model

3.3 Implementing Attack Analyzer

Now coming to our third module in this we are going to implementing our techniques to our Attack Analyzer. The major functions of NICE 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. Each node in the attack graph represents an exploit by the attacker. Each path from an initial node to a goal node represents a successful attack.

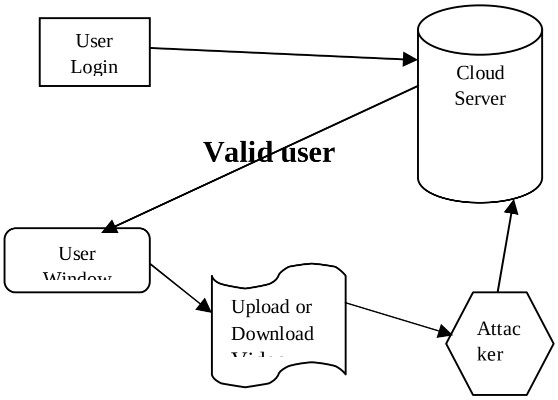


Fig-3: Implementing Attack Analyzer

3.4 Security Measurement Model

When a node is using attack graph as the security metric model for the evaluation of security risks is a good choice. In order to assess the network security risk condition for the current network configuration, security metrics are needed in the attack graph to measure risk likelihood.

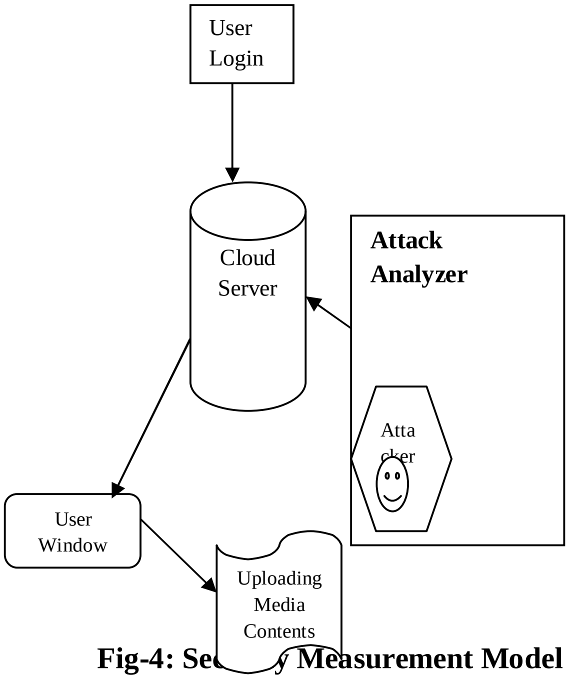


Fig-4: Security Measurement Model

3.5 Counter Measure Selection Model

Algorithm presents how to select the optimal countermeasure for a given attack scenario. Input to the algorithm is an alert, attack graph G, and a pool of countermeasures CM. The algorithm starts by electing the node vAlert that corresponds to the alert generated by a NICE-A. Before selecting the countermeasure, we count the distance of vAlert to the target node. If the distance is greater than a threshold value, we do not perform countermeasure selection but update the ACG to keep track of alerts in the system.

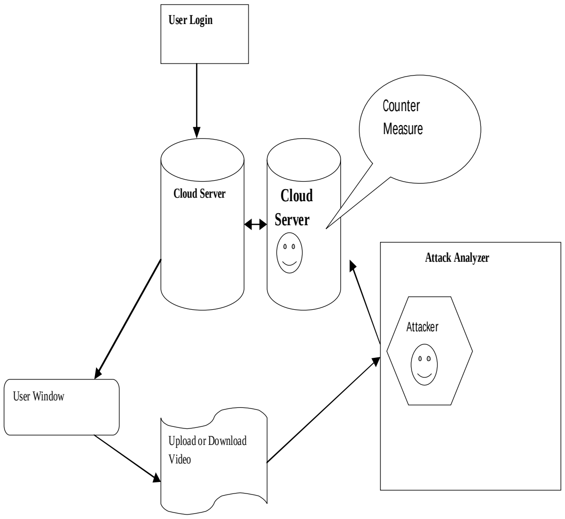


Fig-5: Counter Measure Selection Model

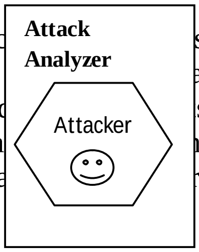
4. ALGORITHM USED

Attack Graph Model (AGM):

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

5. ARCHITECTURE DIAGRAM

The systems architect establishes the basic structure of the system, defining the essential core design features and elements that provide the framework for all that follows, and are the hardest to change later. The systems architect provides the users' view of the users' vision for what the system needs to be and do, and the paths along which it must be able to evolve, and strives to maintain the integrity of that vision as it evolves during detailed design and implementation.



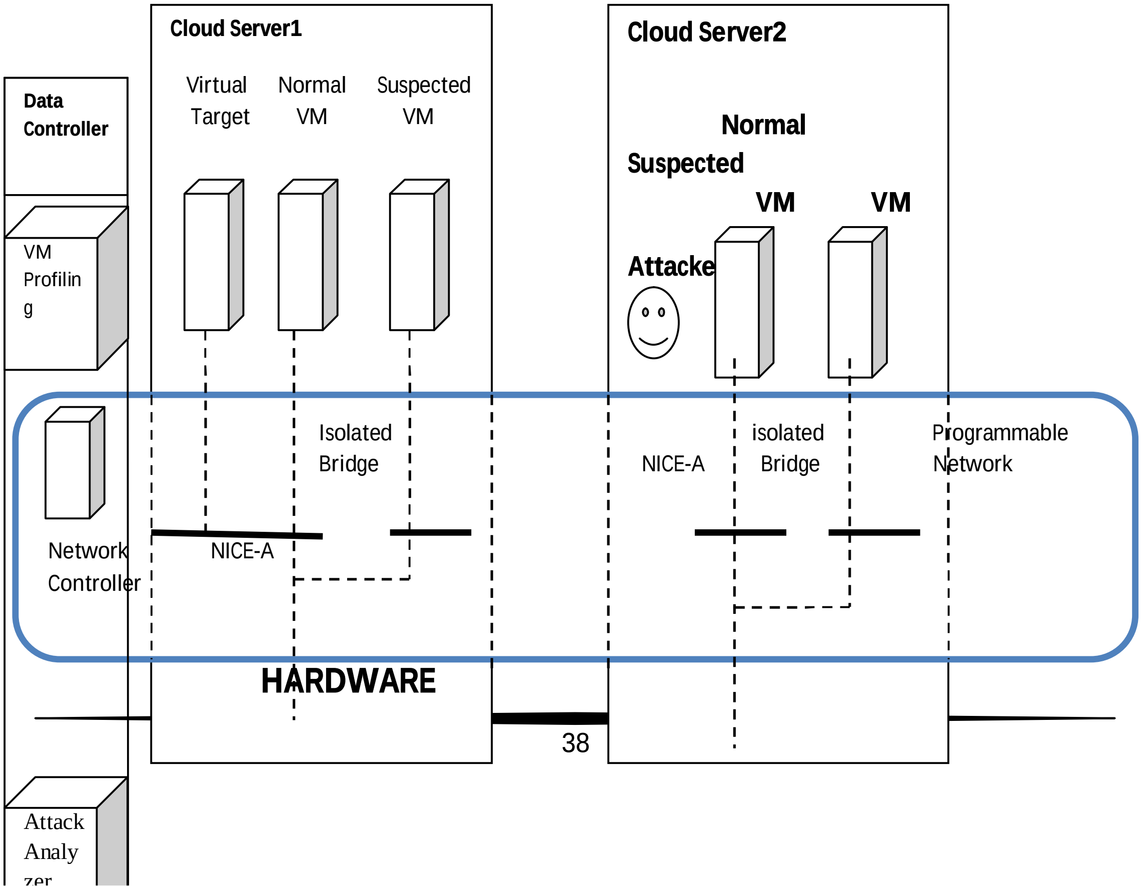


Fig-6: Architecture Diagram

5.1 GANTT CHART

NAME OF THE TASK:

Task1: Paper Analysis & Discussion and Module Separation & GUI Design

Task2: Module Implementation

Task3: 3rd and 4th modules implementation

TASK	AUG-SEP	SEP-NOV	NOV-JAN	FEB-MAR	MAR
1	🟩				
2		▨			
3			🟧		
4				🟧	
5					🟧

Task4: 5th and future enhancement implementation

Task5: Document Preparation and Debugging

Table – 1 : Task Analysis

🟩 → Complete process

▨ → Progressing process

🟧 → Incomplete process

6. CONCLUSION

In this paper, NICE states that, the different attacks can be detected and mitigate in the cloud virtual networking environment. NICE utilizes the attack graph model to conduct attack detection and prediction. The proposed output verifies 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 NICE 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.

6.1 FUTURE ENHANCEMENTS

NICE 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 NICE solution by investigating the decentralized network control and attack analysis model based on current study.

REFERENCES

1. Coud Sercurity Alliance, “Top threats to cloud computing v1.0,” https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf, March 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 Commun., vol. 53, no. 4, pp. 50–58, Apr. 2010.

3. 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.

4. H. Takabi, J. B. Joshi, and G. Ahn, “Security and privacy challenges in cloud computing environments,” IEEE Security & Privacy, vol. 8, no. 6, pp. 24–31, Dec. 2010.

5. 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.

6. G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, “BotHunter: detecting malware infection through IDS-driven dialog correlation,” Proc. of 16th USENIX Security Symp. (SS ’07), pp. 12:1–12:16, Aug. 2007.

7. G. Gu, J. Zhang, and W. Lee, “BotSniffer: detecting botnet command and control channels in network traffic,” Proc. of 15th Ann. Network and Distributed Sytem Security Symp. (NDSS ’08), Feb. 2008.

8. 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.

9. 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.

10. R. Sadoddin and A. Ghorbani, “Alert correlation survey: framework and techniques,” Proc. ACM Int’l Conf. on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services (PST ’06), pp. 37:1–37:10. 2006.

11. L. Wang, A. Liu, and S. Jajodia, “Using attack graphs for correlating, hypothesizing, and predicting intrusion alerts,” Computer Communications, vol. 29, no. 15, pp. 2917–2933, Sep. 2006.

12. S. Roschke, F. Cheng, and C. Meinel, “A new alert correlation algorithm based on attack graph,” Computational Intelligence in Security for Information Systems, LNCS, vol. 6694, pp. 58–67. Springer,2011.