An Efficient Mechanism for Intrusion Detection and Prevention of Zombie Attacks in Cloud

Supritha Gowda K R
M.Tech Scholar
The Oxford College of Engineering
Bangalore, Karnataka, India

Mrs. Malathi Murugesan
Asst. Prof Dept of ISE
The Oxford College of Engineering
Bangalore, Karnataka, India

***Abstract*— Cloud Computing is an emerging technology in recent trends as it provides several services to user. One of the main objectives of cloud is to provide storage capability of the users. Hence security is the main concern with respect to cloud computing as the users are given access to install various applications and store data in the cloud. Among the various attacks on the cloud, Zombie attack and in particular Distributed Denial of Service (DDoS) is considered loosely in the literature survey. DDoS involves the attacker launching an attack at early stage with multistep exploitation, low-vulnerability scanning and compromising the identified vulnerable virtual machine as zombies and later launching a DDoS through the zombie machines. Detecting such an attack is difficult as the cloud user may install vulnerable application in the virtual machine created for the user's purpose. This survey paper aims at compiling the various mechanism developed so far to provide solutions for the various attacks and we conclude by providing an efficient intrusion detection system to identify zombie machine in particular and avoid the attacker from launching an DDoS attack.**

Keywords—Cloud Security, Zombie Machine, Intrusion Detection, Distributed Denial of Service.

I. INTRODUCTION

Cloud computing is receiving a great deal of attention among users. The delivery of computing resources over the Internet is referred to as cloud computing. Various services are provided using cloud concept. Examples of cloud services include social networking sites, online file storage, online business applications and webmail. Cloud services allow individuals and businesses to utilize software and hardware that are managed by third parties at remote locations. The cloud computing model allows access to computer resources and information from anywhere at any time where network connection is available. The cloud removes the need, to be in the same physical location as the hardware that stores the data.

There are different types of clouds that can be subscribed depending on the needs.

1. Private cloud
2. Public cloud
3. Community cloud
4. Hybrid cloud.

Private cloud is cloud infrastructure used especially for an organization or for an individual user. Public cloud is a cloud infrastructure where the services delivered over a network are open for public use. Though technically there is little or no difference between public and private cloud architecture, the security consideration may vary.

Community cloud shares infrastructure among several organizations from a specific community that have similar cloud requirements (like compliance, security, jurisdiction, etc.). A hybrid cloud is essentially a combination of at least two clouds (private, community or public) that remain distinct entities but are bound together, contributing the benefits of multiple deployment models.

There are 3 types of cloud service model as shown in Fig 1

1. Software as a service (SaaS)
2. Platform as a service (PaaS)
3. Infrastructure as a service (IaaS).

Software as a service (SaaS) is a software delivery model in which software and related data are centrally hosted on the cloud by independent software vendors or application service providers.

Platform as a service (PaaS) provides a computing platform and a solution stack as a service. Infrastructure as a Service (IaaS) is a model in which an organization outsources the tools used to support operations, including hardware, storage, servers and networking components. The service provider owns the tools and is in charge for housing, running and maintaining it.

In this paper we consider public cloud in which users are allowed to store data and install applications. Among the various service provided by the cloud we consider Infrastructure as a Service (IaaS) as the user is given storage resources, access to common servers and an individual virtual machine. In public cloud, numerous users are allowed in to access the network and hence security plays a major role.

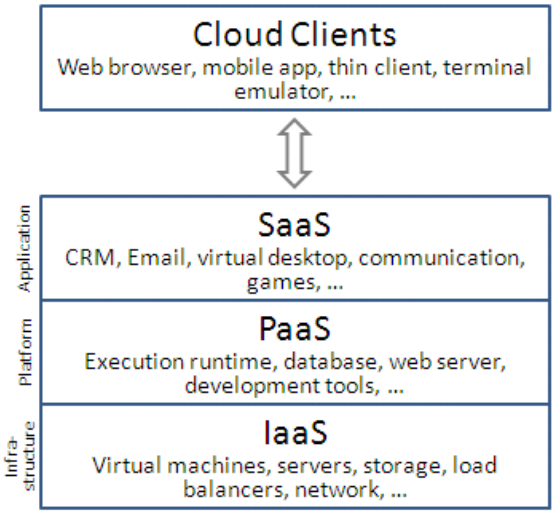

Fig 1. Various Service provided by the cloud

II. RELATED WORK

In this section we briefly talk about the previous work carried out by various researchers on detection of intrusions and various security issues in cloud. A work by Duan et al. [2] projected a mechanism which focuses on detection of zombie machine. It is based on the mechanism called SPOT, which scans the outgoing messages in sequential manner and uses a Sequential Probability Ratio Test [SPRT] to spot the compromised machine. BotHunter [3] is another such mechanism to identify the zombie machine. The working principle is based on the fact that a malware infection process has a well defined number of stages that allow correlating the intrusion. BotSniffer [4] mechanism groups the various flows according to the server connections and searching for similar behavior in the flow.

An attack graph is a representation of series of individual attacks that lead to an undesirable state where an attacker has administrative access to the machine. By constructing such a graph we can determine whether the machine is vulnerable to attacks or is secure. Sheyner et al. [5] proposed a technique that utilizes Binary Decision Diagrams to construct an attack graph. In their scheme the scalability of implementing the same to a bigger network is challenging. Ou et al. [6] came up with a tool called MulVAL to construct an attack graph which adopts a logic programming approach and Datalog language.

Wang et al [7] developed an inmemory structure known as queue graph to keep track of alerts regarding each exploit in the attack graph. In such design it is difficult to make use of correlated alerts for similar attack scenarios. Roschke et al. [8] proposed a modified attack graph based algorithm to create correlations matching alerts to a set of specific nodes in the attack graph. This is known as Dependencies Graph.

After obtaining the possible attack scenarios, it is important to apply countermeasure to avoid further damage to the machine and to the network. Several mechanisms have been proposed to obtain an optimal countermeasure based on cost analysis. Roy et al. [9] considered attacks and their countermeasures together in a tree structure; this particular tree is known as attack countermeasure tree (ACT).

Chun-Jen et al. [1] proposed multiphase distributed vulnerability detection, measurement, and countermeasure selection mechanism called NICE, which is built on attack graph-based analytical models and reconfigurable virtual network-based countermeasures. Host-based IDS solutions are needed to be incorporated so as to improve the detection accuracy and to cover the whole spectrum of IDS in the cloud system.

III. PROPOSED WORK

The Intrusion Detection mechanism [1], investigates the network IDS approach to counter zombie explorative attacks. We, in proposed system incorporate host based intrusion detection solutions (HIDS), so as to improve the detection accuracy and CPU utilization, and to decrease the service delay. Early detection of attacks can be efficiently done by the proposed system as we incorporate host based intrusion detection solutions (HIDS). The main advantage of using HIDS is that we can improve User Security Index (USI). To evaluate security level of a user machine, we define a USI to represent the security level of each user machine in the current network environment.

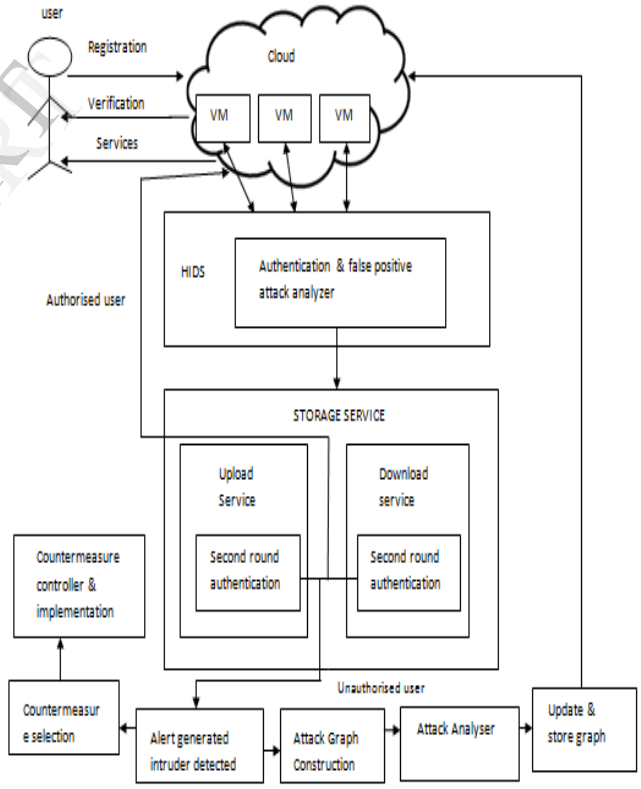

Fig2. Proposed architecture

Our mechanism provides double authentication so that false-positive attack can be efficiently reduced. After first round of authentication and under Deep Packet Inspection the authenticated user will be allowed to avail his service. In second round of verification each user will be checked whether the user has rights to either upload or download files of specific

type (like specific file extensions). The uploaded data of file will be encrypted and stored in cloud. Downloading of specific file is allowed to only those users who have uploaded that specific file, any other user trying to download a file uploaded by other user is considered an attack.

An attack graph will be constructed for each user machine and will be simultaneously analyzed at attack analyzer. The notation of MulVAL logic attack graph is used to construct attack graph. For each new alert, the attack graph will be constructed and updated if necessary. The updated graph will be stored in cloud for further use. The stored updated graph acts as database.

IV. CONCLUSION

One of the main applications of cloud computing is to provide storage service. Public cloud allows users to login and install any vulnerable applications. Hence, security is the most important parameter in cloud. We proposed an efficient enhanced framework to detect and counter the attacks. The proposed framework uses host based intrusion detection solutions, and hence improves the detection accuracy and CPU utilization, and decreases the service delay. The proposed framework also reduces the rate of false-positive attack.

V. ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my guide Mrs Malathi Murugesan, Assistant Professor, The Oxford College of Engineering, Bangalore, for her exemplary guidance, and constant encouragement throughout.

VI. REFERENCES

1. Chun-Jen Chung, Pankaj Khatkar, Tianyi Xing, Jeongkeun Lee, and Dijiang Huang, "NICE: Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems", IEEE Trans. Dependable and Secure Computing, vol. 10, no. 4,pp. 198-211, Jul/Aug 2013.
2. 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.
   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.
3. 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.
4. 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.
5. X. Ou, S. Govindavajhala, and A.W. Appel, "MulVAL: A LogicBased Network Security Analyzer," Proc. 14th USENIX Security Symp., pp. 113-128, 2005.
6. 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.
7. 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.
8. [9] A. Roy, D.S. Kim, and K. Trivedi, "Scalable Optimal Countermeasure Selection Using Implicit Enumeration on Attack Counter- measure Trees," Proc. IEEE Int'l Conf. Dependable Systems Networks (DSN '12), June 2012.