Honeypot as a Service in Cloud

M Balamurugan
Sri Krishna College of
Engineering and Technology
Coimbatore
Tamil Nadu, India.

B Sri Chitra Poornima
Sri Krishna College of
Engineering and Technology
Coimbatore
Tamil Nadu, India.

ABSTRACT

Today's world is increasingly relying on cloud computing. It is the technology and business of the next generation. The increase in the use of cloud environment is followed by a rising volume of security problems. Most cloud providers support classical security systems to avoid illegal access to the resources. But compromising a resource in a cloud environment is easier for the hackers since all the resources are connected together and monitored using centralized controllers. If a single instance of a resource is vulnerable, compromising it might expose other instances or resources in the cloud. Also, it is impossible to have a completely secure coding. Rather than giving much care to make the code bug-free, it is efficient to implement fake services using Honeypots and set a trap for the hackers. The Honeypot gives us valuable information regarding the intruders. In a cloud environment, running a Honeypot helps to track hackers and secure the real instances of the resources. Also it can be provided as a Service which is an additional business profit in cloud. This architecture is based on a decision and a redirection that automatically filters attacks and saves the details about the attackers. The information about the attackers will help to block them and can be used to take legal actions against them. With Honeypots, we can get statistical information about the attackers and the types of attacks. All the above information about the attackers can be given to those who purchase this service (customers) and allowing them to decide on further actions needed to be taken. Thus providing Honeypot as a Service not only gives cloud providers a better security and an additional business profit but it also secures the customers relying on the cloud providers.

General Terms

Cloud Security, Virtualization, Hacking, Port Scanning, Honeypot, Intrusion Detection, Virtual Machine Manager (VMM), Hypervisor.

Keywords

Honeypot as a Service (Haas), Honeybug, FRE (Filter and Redirection Engine), Honey Controller.

1. INTRODUCTION

Most of the people switching to cloud environment always face an issue related to security. Major security issue will be remote hacking of an instance running in the cloud. In classical networking, a Honeypot is a trap set for the attackers making them believe that the system is vulnerable. More likely, in a cloud environment, an advanced Honeypot can be implemented and rendered as a Service.

2. CLOUD SECURITY

Cloud security is an emerging sub-domain of computer security. It refers to a broad set of policies, technologies, and controls deployed to protect data, applications, and the associated infrastructure of cloud computing [1]. With cloud computing techniques that used Virtual Machine Manager (hypervisors) to create and control virtual processors, networks, and disk drives, many of which may operate on the same physical servers was publicized. This makes the cloud environment vulnerable because attackers can steal data by using eavesdropping programs to analyze the different virtual machines running on the same server [2].

2.1 Issues faced by cloud providers

The provider must ensure that their infrastructure is secure and that their clients' data and applications are protected. The cloud provider should ensure that the data from one customer is properly segregated from that of another; it must be stored securely when "at rest" and it must be able to move securely from one location to another. Cloud providers must have systems in place to prevent data leaks or access by third parties. Cloud providers should also ensure that the applications available as a service via the cloud are secure by implementing testing and acceptance procedures for outsourced or packaged application code. It also requires application security measures (application-level firewalls) to be in place in the production environment.

2.2 Issues faced by customers

The customers who purchase the services from the cloud providers are abstracted from the architecture of the cloud. Hence if they face any kind of intrusion or data loss, they cannot identify the source of the attack easily. Also it is very difficult for them to monitor the instances. The cloud providers do not give way much information regarding the access to their instances of the resources. Moreover, the customers may not know where their data is hosted exactly. The privacy of their data may not be ensured always when the service providers are subject to external audits and scrutiny. The customers should also verify whether they can get back their data in proper format in case the service providers go out of business [3].

3. HONEYPOT

A Honeypot is a security resource whose value lies in being probed, attacked, or compromised. This means that whatever we designate as a Honeypot, it is our expectation and goal to have the system probed, attacked, and potentially exploited. Honeypots cannot prevent a particular intrusion or spread of

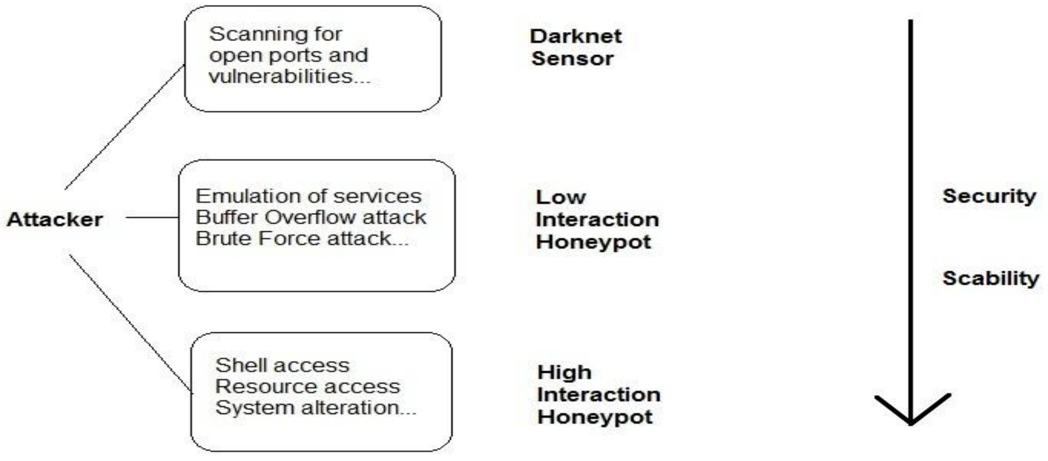
Attacker
Scanning for open ports and vulnerabilities...
Emulation of services Buffer Overflow attack Brute Force attack...
Shell access Resource access System alteration...
Darknet Sensor
Low Interaction Honeypot
High Interaction Honeypot
Security
Scability

Fig 1: Different Honeypot architectures to collect different phases of network attacks

virus or worm; it merely simulates fake services similar to the real one's running on the server, so that hackers attack the wrong targets and eventually get trapped in the Honeypot thereby collecting information about the hacker and detects attack patters. After doing so, the defenders can respond to this evidence by building better defenses and countermeasures against future security threats [4]. So a Honeypot is a tool to collect evidence or information about the attackers, and to gain as much knowledge as possible especially on the attack patterns, hackers' purpose and motivations and commonly used programs to launch attacks**.**

4. HONEYPOT IMPLEMENTATION

When it comes to cloud, the implementation of Honeypot is entirely different since the environment is different. Honeypot can create a population of virtual instances on a cloud network using unassigned IP addresses. They can be used to simulate any TCP or UDP and other commonly used services. Each instance can be configured with a set of emulated services and a specific Operating System behavior.

4.1 Components

The following are the components that are used to implement Honeypot in a Cloud Environment.

4.1.1 Cloud Controller

The cloud controller is the centralized controlling unit for a cloud environment. It is used to manage and expose the virtualized resources (machines, network and storage) via user-facing APIs typically a web browser [5].

4.1.2 Cluster Controller

Multiple clusters are present in a cloud environment each managed by a separate Cluster Controller. It acts like a Virtual Machine Manager (VMM) controlling the execution of various Virtual Machines running on the nodes and their interaction with the users. The clusters also contain a Storage Controller which is used to provide block level storage of data [5] .Under each Cluster Controller runs a group of Node Controllers which are used for managing the startup, execution and termination of services provided by the Virtual Machine instances.

4.1.3 Honey Controller

A virtual Honeypot is a virtualized instance running under a Node Controller. This Node Controller is separated from other Node Controllers which provide real services by providing a separate Cluster Controller for the Honeypot instances. This separation is highly essential because compromising the Honeypot instances should not affect other controllers and instances running in the cloud. This Cluster Controller under which the Honeypots are running is called as Honey Controller. There can be many simulated instances matching the instances running in the cloud to provide a high interaction Honeypot [6].

4.1.4 Filter and Redirection Engine (FRE)

This is an engine which is running on the central Cloud Controller. All the network traffic that is coming into the cloud environment will pass through this engine. The

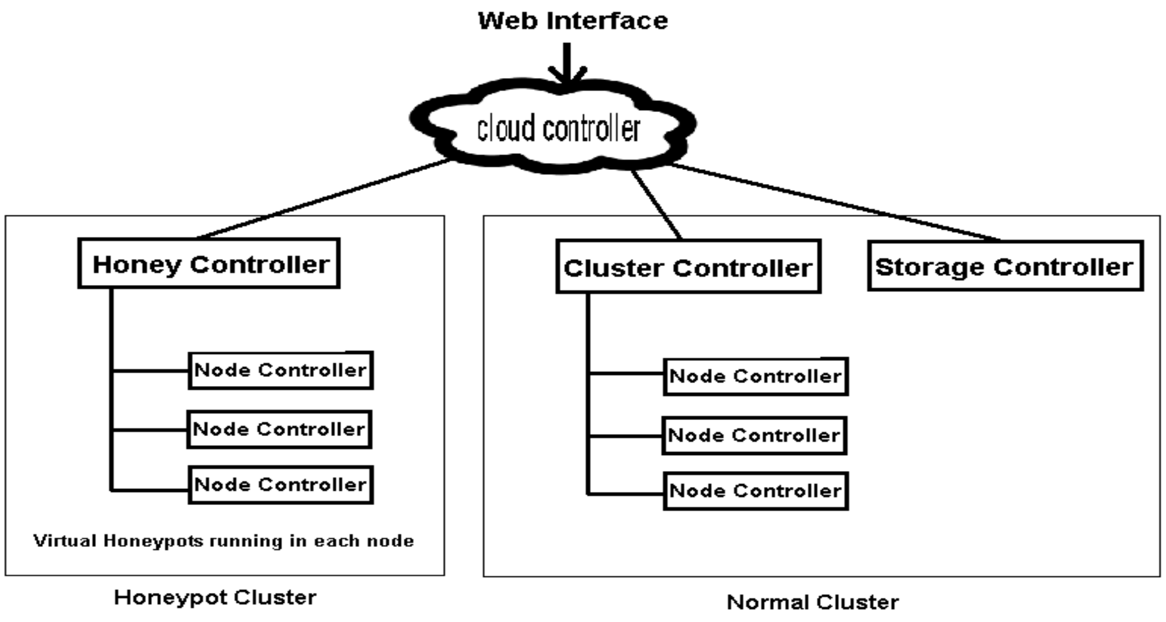

Fig 2: Honeypot implemented in a cloud environment

purpose of this engine is to separate the traffic of malicious users from legitimate users and to redirect them to the Honey Controller. The traffic is analyzed based on some flagged signatures and the fake ports running in the Honeypot.

4.1.5 Log Storage System

All the information about the attackers is logged in the FRE before the traffic is redirected to Honey Controller. These logs are stored in a centralized Log Storage System in Cloud Controller for convenient analysis and logs from the Honey Controller are accessed using remote syslogd [7]. Also it is necessary to provide a web interface to the customers for viewing these logs. The logs typically contain information regarding IP address of attacker, type, time, date and frequency of attack.

5. HONEY CONTROLLER

The Honey Controller is the centralized controller for the Honeypot instances running in the cloud. Under this controller, there are many Node Controllers. Each Node Controller runs a user specific instance which is already running in the cloud. This instance is made vulnerable so that the attacker thinks that the actual instance is vulnerable. The attacker continues attacking this instance while the normal instance will be running without any load. The Honey Controller also logs all the traffic information which is coming through it. This log information serves not only as a backup but it can also be shown as fake logs to the hackers who are compromising the Honeypot instances. It should be noted that one instance of Honeypot is running for many similar real instances for simulating the services. The Honey Controller maintains a detailed list of bugs which are simulated in virtual machines under each node controller so that the traffic can be efficiently handled.

5.1 Node controller

The Node Controller which is running under the honey controller is capable of running many virtualized instances of Honeypots. Each of these is more or less similar to the instance which is already running in the cloud. The information regarding each instance is stored and is sent to the cloud controller to insert the values into the FRE.

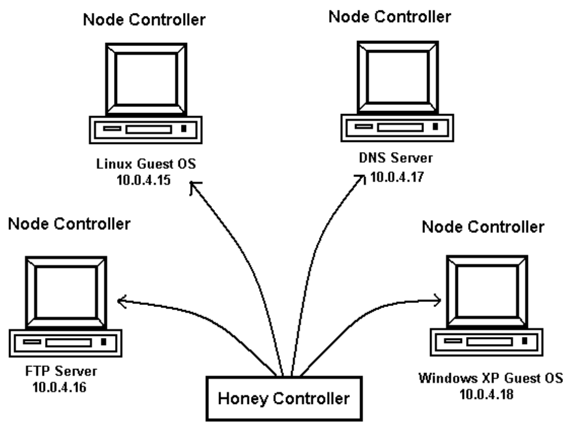

Fig 3: Honey Controller with Virtual Instances

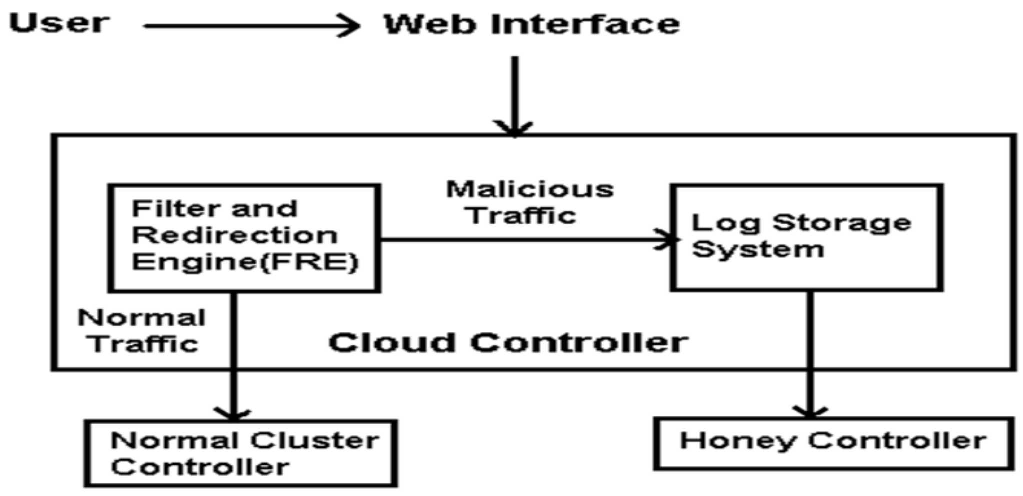

Fig 4: Cloud Controller with FRE and Log Storage System

5.2 Virtual Honeypots

Under each Node Controller, a virtualization platform is setup [8]. Each instance should be run under one virtual machine. The instances must be similar to already running instances and also it should appear vulnerable. This can be achieved by opening fake ports and running Honeybug (a fake exploitation). Also a fake database reply daemon could be run to prevent database attacks like SQL injection. The Honeypot instances can also be made to simulate many well-known exploits. We can use Virtual Machines to host multiple Honeypots on the instances using the concept of nested virtualization so that even if one Honeypot is compromised, the remaining can be used to trap the blackhat and the system can be quickly recovered. Virtual Honeypots are cheaper and easier to deploy than real Honeypots but the drawback is that setting up and managing many virtual instances increases the load on the Honey Controller and a virtual honeypot can gather only limited information unlike a real honeypot. In real Honeypots where nested virtualization is not used to set up many Honeypots, each Honeypot is implemented and maintained in a separate system which makes it expensive and difficult for maintenance. Another disadvantage of real Honeypots is that once it is compromised, it can be used to attack or harm other systems but this problem is easily overcome when using virtual Honeypots as the virtual instances are separated from one another.

6. FILTER AND REDIRECTION ENGINE (FRE)

This is the major process that is running in the Cloud Controller. The function of this engine is to redirect all the malicious traffic destined for a particular real instance to the Honey Controller which has fake instances running in the cloud [9]. It also logs all the information about the traffic source when a redirection is taking place. This engine should be implemented with efficient algorithms since all the traffic into the cloud should be analyzed by this engine. It functions similar to a Network Intrusion Detection System (NIDS) scanning each packet for patterns known as signatures which may indicate the type of attack as most attacks today have some distinct signatures [10]. For example, a person trying to launch a Denial of Service (DoS) attack can be detected by efficiently matching the signatures of DDoS (Distributed Denial of Service) packets and by filtering known attacks having large probability. However, this method, although being very efficient, has the drawback that new attacks are not detected until our system knows the signature for which we have to maintain a database of signatures and keep updating it to include signatures of new attacks [11]. In conventional networks the Honeypot is used to monitor the system for changes. A legitimate user does not interact with the Honeypot and hence any interaction with the Honeypot is considered malicious and the hacker is trapped. But in this proposed architecture, the FRE uses signatures to scan the incoming traffic and redirects it to Honeypot upon finding it to be malicious. This prevents the Honeypot from being compromised and ensures security of the cloud. One of the other methods to detect malicious traffic is by using inverse distribution of packet contents [12]. In a more sophisticated environment, this engine could be run on the particular instance for which the Honeypot is purchased as a service by the customer. If the detectors implemented in FRE are unable to find an attacker then resources can be attacked and hence the value of the system lies in the efficiency of the Intrusion Detection Systems (IDS) implemented in the FRE.

7. HONEYPOT AS A SERVICE (HAAS)

The above part shows how to implement Honeypot in a cloud environment. Next is to make business profit out of it. To do this, the Honeypot should not be activated normally unless the customer buys one for his instance. The customer should be provided with an option to buy Honeypot when purchasing the instances as Honeypots are mostly needed for large networks where security and data reliability is of great concern [13]. Upon checking out the Honeypot, an entry should be added to the Filter and Redirection Engine and hence all the malicious traffic to the customers' instance gets redirected to the Honeypot. Also the customer must be given the logs containing information about the attackers and also the statistics about the attacks. This will help them to safeguard their resources against future attacks. The customer then will take necessary actions against the attackers.

8. SUMMARY

This paper demonstrates the implementation of Honeypot in a cloud environment for providing security and business profit. Thus, if the Honeypot architecture is deployed while setting up a cloud, it results in enhancement of security as well as profit to the cloud providers. For efficient operation of such a Honeypot in a cloud, it should be independent from the software used to setup the cloud environment. Hence, a cloud platform independent Honeypot is the best way of improving the security in the cloud.

9. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of this idea. This includes the staff members of our college and our friends.

10. REFERENCES

[1] Major security issues in cloud URL: "http://en.wikipedia.org/wiki/Cloud_computing_security" Retrieved: August 2011

[2] Security in the Ether, Jan/Feb 2010, Technology Review India, Published by MIT URL: "http://www.technologyreview.in/web/24166/"

[3] Gartner: Seven cloud-computing security risks, July 02, 2008 URL: "http://www.infoworld.com/d/security-central/gartner-seven-cloud-computing-security-risks-853?page=0,0"

[4] Berthier,Robin G. 2009 "Advanced Honeypot Architecture for Network Threats Quantification" University of Maryland, College Park

[5] Eucalyptus Documentation Version 2.0,Eucalyptus Systems URL:"http://open.eucalyptus.com/wiki/IntroducingEucalyptus_v2.0" Retrieved: August 2011

[6] Mohamed Noordin Yusuff, "Honeypots Revealed" URL: "http://www.infosecwriters.com/text_resources/pdf/Honeypots.pdf" Retrieved: September 2011

[7] Remote Host Logging with syslogd,FreeBSD Handbook URL: "http://www.freebsd.org/doc/handbook/network-syslogd.html" Retrieved: September 2011

[8] Running OpenStack under VirtualBox – A Complete Guide , February 17,2011,System Administration and ArchitectureBlog URL: "http://uksysadmin.wordpress.com/2011/02/17/running-openstack-under-virtualbox-a-complete-guide/"

[9] Todd Lammle , "CCNA-Cisco Certified Network Associate Study Guide" ,Second Edition

[10] Loic Etienne/EPFL-SSC "Malicious Traffic Detection in Local Networks with Snort" URL: "http://infoscience.epfl.ch/record/141022/files/pdm.pdf"

[11] Nathalie Weiler,2002 "Honeypots for Distributed Denial of Service Attacks", Proceedings of Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'O2) 1080-1383/02

[12] Vijay Karamcheti, Davi geiger, Zvi Kedem, S.Muthukrishnan "Detecting Malicious Network Traffic Using Inverse Distributions of Packet Contents"

[13] "Software as a Service Inflection Point: Using Cloud Computing to Achieve Business Agility" Melvin B. Greer Jr ,Published by iUniverse(May13,2009), ISBN-10: 1440141967 ,ISBN-13: 978-1440141966