International Journal of Multidisciplinary Research and Development

Volume: 2, Issue: 4, 172-176
April 2015
www.allsubjectjournal.com
e-ISSN: 2349-4182
p-ISSN: 2349-5979
Impact Factor: 3.762

Ramya. R
M. Phil Research Scholar, Department of Computer Science Vivekanandha College of Arts and Sciences for Women, Namakkal, Tamil Nadu, India.

Correspondence:
Ramya. R
M. Phil Research Scholar, Department of Computer Science Vivekanandha College of Arts and Sciences for Women, Namakkal, Tamil Nadu, India.

Securing the system using honeypot in cloud computing environment

Ramya. R

Abstract
Cloud Computing means accessing the data from their own datacenters such that the chances of eavesdropping have been reduced and storage cost is reduced. A honey pot is a computer system on the Internet that is expressly set up to attract and "trap" people who attempt to penetrate other people's computer systems. Honeypots are systems used to trap, monitor, and identify erroneous requests within a network. Honeypots using various cloud computing platforms (such as Amazon EC2, Windows Azure etc.) with the objective of learning more about what kind of packets they receive. Honeypots are not always designed to identify hackers. Honeypot developers are often more interested in getting into the minds of hackers, which then permits them to design more secure systems, as well as to educate other professionals about the lessons learned through their efforts. Honeypots are designed to purposely engage and deceive hackers and identify malicious activities performed over the Internet. Honeypots are considered an effective method to track hacker behavior and heighten the effectiveness of computer security tools.

Keywords: Honeypot, Cloud Computing, Honeyd, Honeynets, Honeywall, Cloud IDS.

1. Introduction
Now-a-days, more and more people are using internet all over the world. Most of the collected data are valuable since any traffic come to the honeypot/net is suspicious. The honeypots varies in the interaction they provide to the attackers, from the low interaction to medium and high, each type has its advantages and disadvantages. The aim of the honeypot is analyzing, understanding, watching and tracking hacker's behaviors in order to create more secure systems. Honeypot is enormous method to get better network security administrators' knowledge and learn how to get information from a victim system using forensic tools. "A honeypot is a closely monitored computing resource that we want to be probed, attacked, or compromised. More precisely, a honeypot is "an information system resource whose value lies in unauthorized or illicit use of that resource".

2. Honeypot in Cloud
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. Honeypot is a detection and response tool, rather than prevention which it has a little value in. Because honeypots cannot prevent a particular intrusion or spread of virus or worm, it merely collects information and detects attack patterns. After doing so, the defenders can respond to this evidence by building better defenses and countermeasures against future security threats.
A honeypot is a tool to collect evidence or information, and to gain as much knowledge as possible especially on the attack patterns, hacker‟s purpose and motivations and the commonly used programs launched by them. From all the information received, we can even learn more about the hacker‟s ability especially their technical knowledge. Honeypots can also be used to catch hackers while they are in the network and to redirect hackers from the actual production systems to the honeypot system. The best personnel to manage the honeypot is one with extensive knowledge in three critical areas – Security, Systems, and Networks.

3. Types of Honeypots
Honeypots come in many shapes and sizes. Honeypots are divided into low-interaction and high-interaction honeypots. Low-interaction honeypots (Production honeypots) have limited

interaction they normally work by emulating services and operating systems. Attacker activity is limited to the level of emulation by the honeypot. For example, an emulated FTP service listening on port 21 may just emulate a FTP login, or it may support a variety of additional FTP commands. The advantages of a low-interaction honeypot are their simplicity. These honeypots tend to be easier to deploy and maintain, with minimal risk. Usually they involve installing software, selecting the operating systems and services you want to emulate and monitor, and letting the honeypot go from there. This plug and play approach makes deploying them very easy for most organizations. Also, the emulated services mitigate risk by containing the attacker's activity, the attacker never has access to an operating system to attack or harm others. The main disadvantages with low interaction honeypots is that they log only limited information and are designed to capture known activity. The emulated services can only do so much. Also, its easier for an attacker to detect a low-interaction honeypot, no matter how good the emulation is, skilled attacker can eventually detect their presence. Examples of low-interaction honeypot include Specter, Honeyd, and KFSensor.

High-interaction honeypots (Research honeypots) are usually complex solutions as they involve real operating systems and applications. Nothing is emulated, we give attackers the real thing. If you want a Linux honeypot running an FTP server, you build a real Linux system running a real FTP server. The advantages with such a solution are two fold. First, you can capture extensive amounts of information. By giving attackers real systems to interact with, you can learn the full extent of their behavior, everything from new rootkits to international IRC sessions. The second advantage is high-interaction honeypots make no assumptions on how an attacker will behave. Instead, they provide an open environment that captures all activity. This allows high-interaction solutions to learn behavior we would not expect. An excellent example of this is how a Honeynet captured encoded back door commands on a non-standard IP protocol (specifically IP protocol 11, Network Voice Protocol). However, this also increases the risk of the honeypot as attackers can use these real operating system to attack non-honeypot systems. As result, additional technologies have to be implement that prevent the attacker from harming other non-honeypot systems. In general, high-interaction honeypots can do everything low-interaction honeypots can do and much more. Examples of high-interaction honeypots include Symantec Decoy Server and Honeynets.

4. Honeyd: Low-Interaction Honeypot

Developed by NielsProvos, Honeyd is Open Source and designed to run primarily on UNIX systems (though it has been ported to Windows). Honeyd works on the concept of monitoring unused IP space. Anytime it sees a connection attempt to an unused IP, it intercepts the connection and then interacts with the attacker, pretending to be the victim. By default, Honeyd detects and logs any connection to any UDP or TCP port. In addition, you can configure emulated services to monitor specific ports, such as an emulated FTP server monitoring TCP port 21. When an attacker connects to the emulated service, the honeypot detects and logs the activity and also captures all of the attacker's interaction with the emulated service. In the case of the emulated FTP server, we can potentially capture the attacker's login and password, the commands they issue, and perhaps even learn what they are looking for or their identity. Most emulated services work the same way. They expect a specific type of behavior, and then are programmed to react in a predetermined way. If attack A does this, then react this way. If attack B does this, then respond this way. The limitation is, if the attacker does something that the emulation does not expect, and then it does not know how to respond. Most low-interaction honeypots simply generate an error message. You can see what commands the emulated FTP server for Honeyd supports by review the source code.

Some honeypots, such as Honeyd can emulate services and actual operating systems. In other words, Honeyd can appear to the attacker to be a Cisco router, WinXP web server, or Linux DNS server. There are several advantages to emulating different operating systems. First, the honeypot can better blend in with existing networks if the honeypot has the same appearance and behavior of production systems. Second, you can target specific attackers by providing systems and services they often target, or systems and services you want to learn about. When an attacker connects to an emulated service, you can have that service behave like and appear to be a specific OS.

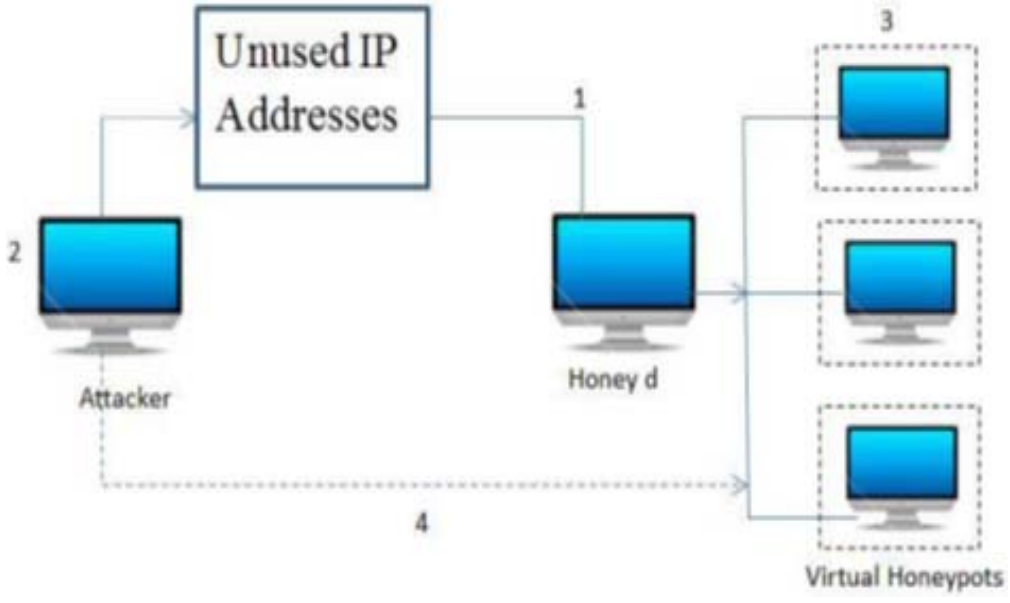

Fig 1: Working of Honeyd

Honeyd monitors unused IP space (1). When an attacker(2) probes an unused IP, Honeyd detects the probe, takes over that IP via ARP spoofing, then creates a virtual honeypot(3) for the attacker to interact with (Honeyd can create multiple virtual honeypots to fool attackers on all unused addresses). The attacker is fooled into thinking he is interacting with a successful hacked system (4). In addition, Honeyd automatically updates its list of unused IPs as systems are added or removed from the network. Honeyd is primarily used for detecting attacks. It works by monitoring IP addresses that are unused, that have no system assigned to them. Whenever an attacker attempts to probe or attack a non-existent system, Honeyd, through Arp spoofing, assumes the IP address of the victim and then interacts with the attacker through emulated services. These emulates services are nothing more than scripts that react to predetermined actions. For example, a script can be developed to behave like a Telnet service for a Cisco router, with the Cisco IOS login interface. Honeyd's emulated services are also Open Source, so anyone can develop and use their own. The scripts can be written in almost any language, such as shell or Perl. Once connected, the attacker believes they are interacting with a real system. Not only can Honeyd dynamically interact with attackers, but it can detect activity on any port. Most low interaction honeypots are limited to detecting attacks only on the ports that have emulated services listening on. Honeyd is different, it detects and logs connections made to any port, regardless if there is a service listening. The combined capabilities of assuming the identity of non-existant systems, and the ability to detect activity on any port, give Honeyd incredible value as a tool to detect unauthorized activity.

5. Honeynets: High Interaction Honeypots

Honeynets are architecture, an entire network of computers designed to attack. The idea is to have an architecture that creates a highly controlled network, one where all activity is controlled and captured. Within this network we place our intended victims, real computers running real applications. The bad guys find, attack, and break into these systems on their own initiative. When they do, they do not realize they are within a Honeynet. All of their activity, from encrypted SSH sessions to emails and files uploads, are captured without them knowing it. This is done by inserting kernel modules on the victim systems that capture all of the attacker's actions. At the same time, the Honeynet controls the attacker's activity. Honeynets do this using a Honeywall gateway (Figure 2). This gateway allows inbound traffic to the victim systems, but controls the outbound traffic using intrusion prevention technologies. This gives the attacker the flexibility to interact with the victim systems, but prevents the attacker from harming other non-Honeynet computers.

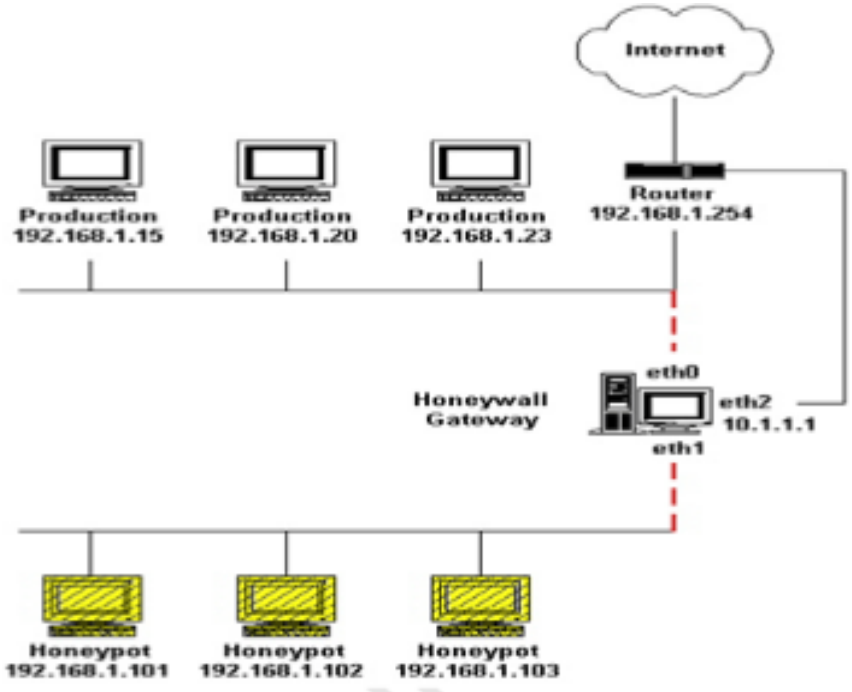

Fig 2: Honeynet

6. Cloud Ids with Honeypot

There are many ways for attacks to attack the target system and then acquiring advantage of the known vulnerabilities of computer systems. In fact, attack leads to loss and disclosure of sensitive information and data stored in the computer. Signature matching is used in the integrated model with normal traffic profiling to improve attack detection. Moreover, the system deploy IDS in the virtual machine itself as well as the virtual network in order to monitor the activities of the system in addition of monitoring the packet traffic in the network to filter the malicious packets coming from suspected sources (see in Figure 3). The system need to configuration at firewall setting because all data will transfer the path of Honeypot server. Honeypot perform as a surveillance and early warning tool. It is a computer or a network site that appears to be the isolated part of the network. It contains the information that is very valuable to the hackers and attackers. The information contains in the Honeypot is very valuable to the hackers and attackers. This paper proposes a new way of protecting data and resources in the Cloud computing environment with Honeypot and it is based on the rational implementation of intrusion detection system (IDS)over the Clout computing infrastructure. The system focus on the Infrastructure as a Service (IaaS) which is a one layer of Cloud computing.

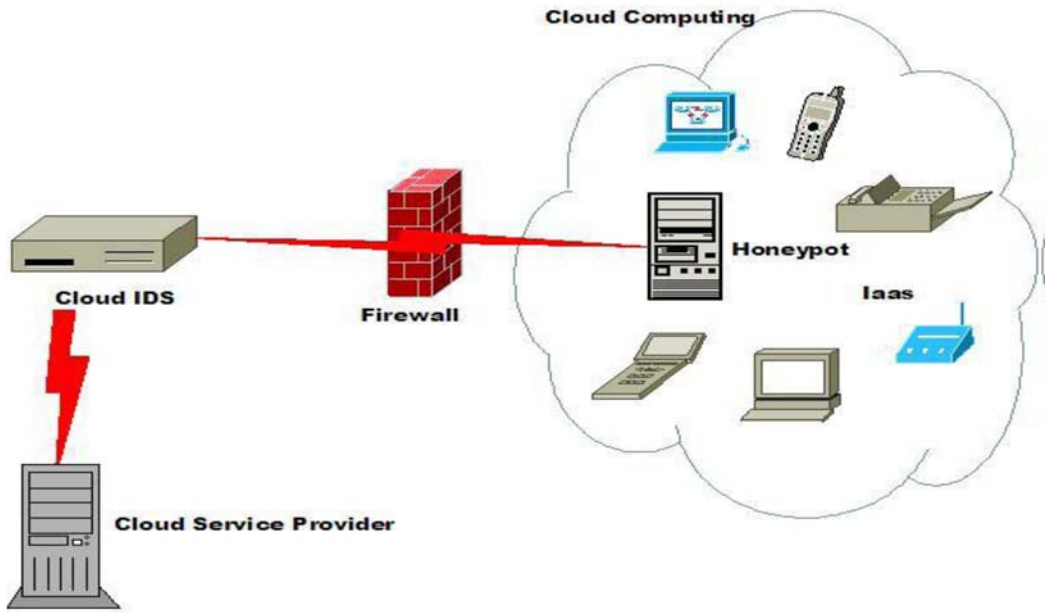

Fig 3: The proposed Cloud IDS with Honeypot

Honeypot are deployed at the Intrusion detection and prevention system (IDPS) which is an integrated model that consists of two techniques (AD) and (SD). When Honeypot are placed behind a firewall, it can introduce new security risks to the internal network, especially if the internal network is not secured against the Honeypot through additional firewalls. There is important to distinguish between a setup where the firewall enables access to the Honeypot or where access from the Internet is denied. A Honeypot does provide a lot of services and also most of them are not used as exported services to the Internet. They are not forwarded to the Honeypot by the firewall. By placing the Honeypot behind a firewall, it is inevitable to adjust the firewall rules if access from the Internet should be permitted. The proposed integrated system detected any of the attacks and compare it with the know threats (signature) and produce an alarm in the case of matching according to Signature Based Detection technique.

7. Advantages of Honeypots

i) Small data sets: Honeypots only collect attack or unauthorized activity and also dramatically reducing the amount of data they can collect. Many organizations can log the thousands of alerts a day with Honeypots. Honeypots can collect the data to easily manage and analyze.
ii) Reduced False Positives: Honeypots may be able to reduce false alerts when they capture unauthorized activity.
iii) Catching False Negatives: Honeypots can easily identify and collect new attacks never seen before.
iv) Minimal Resources: Even on the large dataset, Honeypots require minimal resources. It may case cost effective solution.
v) Encryption: Encrypted attacks can be captured by Honeypots.

8. Legal Issues and Challenges

There are potential legal pitfalls that may turn your honeypot in to a liability. There are many factors which determine whether or not the use of a honeypot is legal. We provide a brief overview of some of the issues. If deploying a honeypot in the United States, then there are at least three legal issues that you must consider,

- Entrapment - Attackers may argue entrapment.
- Privacy – Laws exist that might restrict your right to monitor users on your system.
- Liability - Realize that attackers may misuse your honeypot to harm others.

9. Conclusion

Honeypots provide an updated source of information about attacks methodologies and tools in addition to several log files. It can imitate the original victim, to repeat the analysis process. It also provides a cost-effective approach; instead of sniffing the whole network traffic, honeypots receives only malicious traffic. Honeypots bring to intrusion-protection solutions are hard to ignore, especially now as production honeypots are beginning to be deployed. In time, as deployments proliferate, honeypots could become an essential ingredient in an enterprise-level security operation. Honeypots can be used for production purposes by preventing, detecting, or responding to attacks. Honeypots can also be used for research, gathering information on threats so we can better understand and defend against them.

Reference:

1. D.Esesvehttp://www.oocities.org/dresesve/honeypots.pdf
2. Http://en.wikipedia.org/wiki/Cloud_computing
3. J.B.RavenAlder,AdamDoxtater, James Foster, Toby Kohlenberg, &Micheal Rash, “Snort2.1 Intrusion Detection, “2nd ed. Roackland, MA: Syngress (Distributed by O’Reilly and Associates), 2004.
4. Lanc.Spitzner, http://www.tracking-hackers.com
5. Lance Spitzner, “Honeypots Definition and Value of Honeypots”, 17 May, 2002, URL:http://www.enteract.com/~lspitz/honeypot.html
6. RetoBaumann and Christian Plattner,“White Paper: Honeypots”, 26 February 2002 URL: http://www.inf.ethz.ch/~plattner/pdf/whitepaper.pdf
7. SR Nithin Chandra and TM Madhuri. Cloud security using honeypot systems.
8. M. Balamurugan and B.S.C. Poornima. Honeypot as a service in cloud

9. Y.K.Jain, S. Singh "Honeypot based Secure Network System" in IJCSE.Vol 3.No.2 Feb 2011.
10. Honeypots: Tracking Hackers-http://www.tracking-hackers.com.
11. Provos, N. Developments of the Honeyd Virtual Honeypot, http://www.honeyd.org.
12. Martin, W.W. Honeypots and Honeynets – Security through Deception. http://www.sans.org/reading_room/whitepapers/attacking/41.php , SANS Institute, 2001, As Part of the Information Security Reading Room.
13. Mokube,I.& Adams M.,2007. Honeypots: Concepts, Approaches, and Challenges. ACMSE 2007, March 23-24, 2007, Winston-Salem, North Carolina, USA ,pp.321-325.