INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 3)
Available online at: www.ijariit.com

Two-layered honeypot system implemented on a cloud server

Abhinandan Shetty
abhishetty76@gmail.com
School of Engineering and Technology Jain University (SET JU), Bengaluru, Karnataka

K Sriram
k.sriram4493@gmail.com
School of Engineering and Technology Jain University (SET JU), Bengaluru, Karnataka

Nandish R
nandish.vijeth@gmail.com
School of Engineering and Technology Jain University (SET JU), Bengaluru, Karnataka

Ruthwik Soudry
soudryruthwik@gmail.com
School of Engineering and Technology Jain University (SET JU), Bengaluru, Karnataka

Madhu B. R
brmadhu@gmail.com
School of Engineering and Technology Jain University (SET JU), Bengaluru, Karnataka

ABSTRACT

This project demonstrates the implementation of a high-interactive Honeypot (Sebek) on a medium-interactive Honeypot (Cowrie) which itself is applied on a Cloud server containing sensitive data or resources. A Honeypot can still have certain weaknesses, which means a hacker can get into the system by detecting the Honeypot, or bypass it completely. This can be detected or even completely avoided if there is a Honeywell logging and protecting the Honeypot itself. It will also underline the importance of having a Honeypot and illustrate the statistical and real data collected by the implemented system.

Keywords: *Honeypot, Honeywell, Cloud Server, Cowrie, Sebek, Logging.*

1. INTRODUCTION

1.1. Cloud Security

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. 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. 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 analyse the different virtual machines running on the same server.

1.2 Honeypots

In the field of computer security, a major question is whether to focus on studying and developing defensive techniques based on pre-existing attacks and known methodologies or to study the approaches used by attackers on live systems via observation and logging. One way to do this is to use a technology called Honeypots, which allows malicious activity to be observed and logged, all without the attacker's knowledge. To the attackers, honeypots appear to be unsecured computers set to be attacked and in actuality serve no other purpose but to be attacked. 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 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 defences and countermeasures against future security threats. Honeynets, which are networks of honeypots, take these concepts one-step further and provide a much more robust environment for an attacker to interact with. This increases both the volume of data that can be gathered, as well as the potential for more complex attacks to be captured. A Honeynet is not only an example of a high-interactive honeypot but also a conceptual network architecture. The gateway device in a Honeynet honeypot is called a Honeywell. It can be considered the main point of entry and exit for all network traffic for a Honeynet honeypot. This allows for complete control and analysis of all network traffic to and from a Honeynet system.

1.3. Types of Honeypots

1) Low interactive: These systems will only emulate some important services like SSH, HTTP, FTP. They will be very easy to be discovered by attackers and they do provide the lowest level of security overall. Ex: Honeyd.

2) Medium-interactive: Kippo Honeypot is a medium-interaction honeypot. This means that it is still a software instance running on an operating system but it will blend so well with the operating system that it will be very hard to be discovered by the attackers. Ex: Kippo, Cowrie.

3) High interactive: The main characteristics of a high interaction honeypot is that it will be using a real operating system and hardware but it will be operated, monitored and analysed as being a honeypot system. Ex: Sebek.

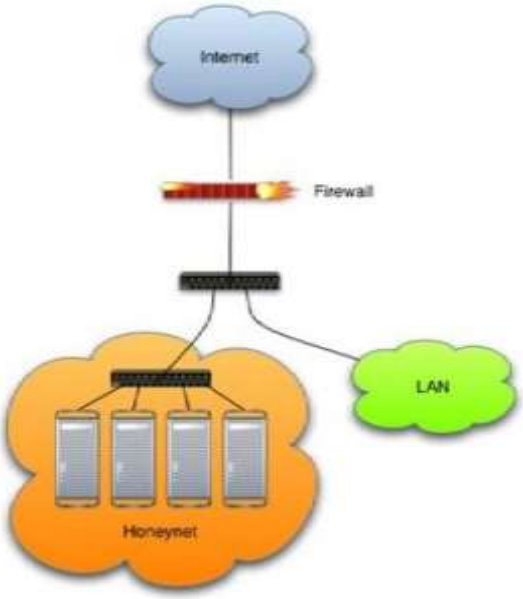

Figure-1: Architecture of Honeypot

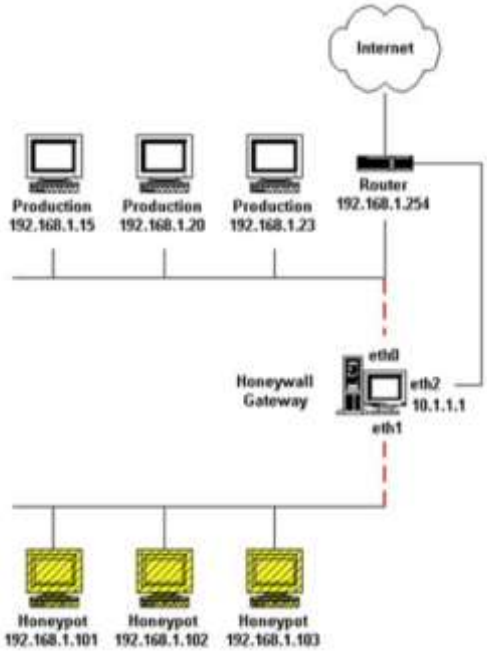

Figure-2: Architecture of Honeynet

2. EXISTING SYSTEMS

2.1. How to build a honeypot System in the cloud

This IEEE paper demonstrates a guide to implementing a cloud-based honeypot using the Kippo honeypot application suite. It also touches upon analyzing the data collected by the Honeypot. Honeypots have been classified based on interaction as discussed in the previous section. Additionally, Honeypots can also be classified based on implementation.

1) Production Honeypots: They are honeypots that are deployed in production having an active part n the overall cyber security defense of an organization. They will be closely monitored and maintained and will have an important part into ISMS (Information Security Management System).

2) Research Honeypots: They are honeypots used to gather statistical data; most of them are used for research only purpose. They are intentionally kept running with the highest level of risk attached to them, in order to expose them to a highest spectrum of attacks and situations.

Honeypots can also be classified based on location:

1) Local Honeypots: Honeypot systems installed in a local domain, being part of a local infrastructure. They are difficult to implement and usually, it takes a medium to the long period from planning to the testing phase.

2) Cloud Honeypots: Honeypots deployed in the cloud with multiple advantages, but also restrictions. They can be deployed quickly, easy to be installed and restored in case of corruption.

The three major phases that a hacker usually follows in a hacking scenario.

• Reconnaissance and Scanning

• Exploiting Phase

• Maintaining access and hiding track

After 8 months of activity, there was a significant amount of data that was captured having a number of almost 6000 unique IPs from where attackers tried to log in. 3 million login attempts were made, 4000 trying to connect using the combination of Admin and 123456 as login credentials.

2.2. Development and Implementation of the Honeynet on a University Owned Subnet

Honeypot/net technology utilizes computers whose sole purpose is to be attacked so the tools, techniques, and modus operandi of attackers can be studied. Their project was to implement a Honeynet with limited resources in such a way as to not endanger the University of Wisconsin - Eau Claire (UWEC) network.

2.3. Cloud Security using Honeypot Systems

In this survey paper, they stress upon Cloud Security using Honeypots –The purpose of this paper is to explain how honeypots are used for securing cloud systems, their advantages and disadvantages, and their value to the security.

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.

How Honeyd works can be shown more appropriately as shown in Figure 1.5 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.

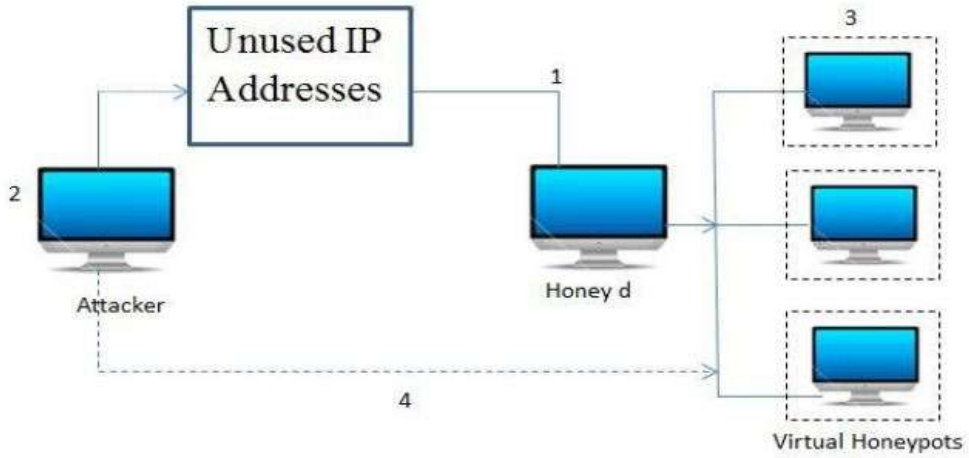

Fig-3: Working of Honeyd - A low interaction honeypot

2.4. A Prevention of DDos Attacks in Cloud Using Honeypot

As Cloud Computing relies on sharing computer resources, it is prone to various security risks. One such security issue is Distributed Denial of Services attack on the cloud. This paper deals with the prevention of DDos attacks and how honeypot approach can be used in cloud computing to counter DDos attacks.

Types of cloud computing models:

1) SAAS: In this model a complete application is offered to the customer as per the demand of the customer on cloud only single instance of service runs. Google, Salesforce and Microsoft offer software as a service. In this cloud provider manages the infrastructure and platform that run the application, which is sometimes referred as "on demand service" and is usually priced on pay per usage.

2) PAAS: in platform as a service an operating system, the provider and the customer provide hardware and network only install or develop its own software and models.

3. IAAS: It is a provision model in which organization outsources the equipment for support in some operations. They outsource storage, hardware, servers and networking components. In IAAS, cloud provider is the owner of all equipment, which is outsourced by the organization, and service provider is responsible for running and maintaining its equipment.

2.4.1. DDos (Distributed denial of services): Distributed Denial of Service (DDos) attack, which means many nodes systems attacking one node all at the same time with a flood of messages. The proposed framework illustrates the implementation of honeypot in the cloud. In this security infrastructure, they introduce a new system: a honeypot that should attract distributed denial- of-service attackers. Web server, mail server, client etc. are forwarded the legitimate destination and honeypot fulfil the task of luring the attacker. Standard mechanisms are used for protection of web and mail servers. Services such as web, mail, ftp services and DNS that should be accessible form the out- side are situated in a demilitarized zone (DMZ). Three major problems must be solved to successfully project this illusion to the attacker: The attack must be detectable. The attack packets must be actively directed to the honeypot. The honeypot must be able to simulate the organization's network infrastructure, at least the parts known to the attacker. The first issue is linked to the solution of the second problem: both should ideally be implemented by a transparent packet forwarder at the border of the corporation's DMZ. Finally, the third problem can be solved by employing a variant of the Honeynet approach.

2.5. Honeypot as a Service in Cloud

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.

3. PROPOSED SYSTEM

3.1. Creating Dynamic website

1) Dynamic website is a collection of dynamic web pages whose content changes dynamically based on the user. It accesses content from a database.

2) A dynamic website is created using HTML CSS JavaScript Bootstrap and PHP.

3) This website has a login page a landing page and a registration page where the users have to enter the information which will be later saved on to the database.

3.2. Hosting the Website on the AWS EC2 Instance

1) The website is hosted on Ec2 in order to have more control over it. It provides higher page ranking on a google page faster access speed and a faster server response.

2) In order to move the website we will the using the putty tool which is a free and open-source terminal emulator, serial console and network file transfer application.

3) It supports network protocols such as SSH which is being used for our ec2 instance and also it can connect to the serial port. Using putty configuration we start the server. Copy the files (the website) onto the server. Import the database into MySQL server.

3.3. Implementing Cowrie on the Cloud

1) Cowrie is a medium-interaction SSH honeypot written in Python. It is used to log brute force attacks and the entire shell interaction performed by an attacker. Cowrie logs everything that has been accessed via port 2222. But most automated tools which are used by hackers, default SSH to port 22. Thus, it would be a good idea to make Cowrie listen to port 22 instead. To do this, we need to change the port which your server uses for SSH.

2) Putty configuration shell is used in order to implement the same. Finally Cowrie is configured and is ready to log attacks.

3) Cowrie allows an attacking entity to attempt a login to the system, believing it is entering into a legitimate SSH session with the server. Upon successful guessing of the password, the attacker is then moved into a fake system with which they can interact.

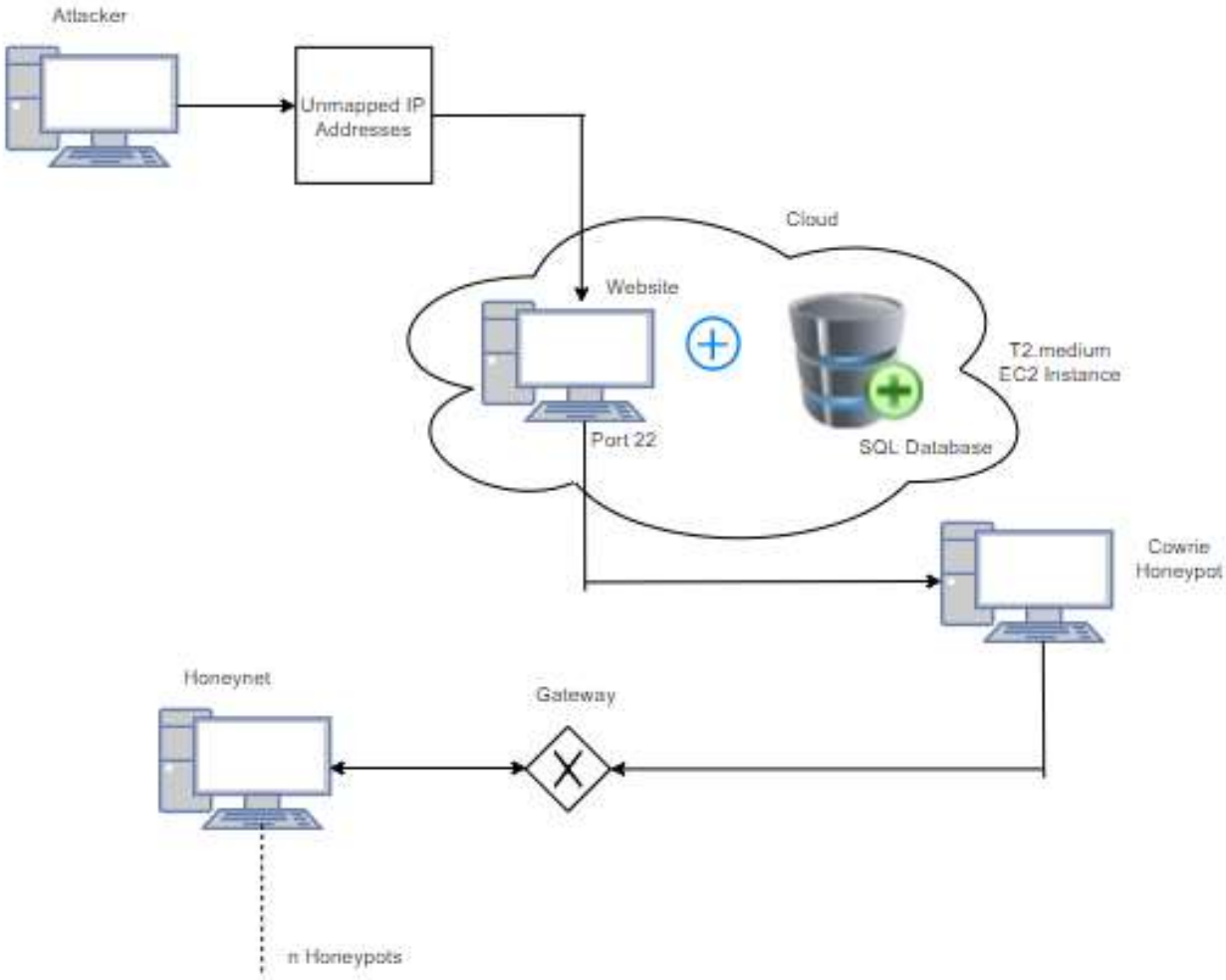

Fig-4: Architecture of Proposed System

3.4. Implementing Honeynet on the Honeypot

1) Cowrie being a medium interactive honeypot can be bypassed by hackers which would ultimately lead the sensitive data on the cloud to be exposed to the hacker.

2) Recognition of Cowrie:- The biggest disadvantage of the honeypot is that it simulates the system, but does not provide the full environment. An example that may overlook the administrator command is ./ where the normal system behaviour looks like this:

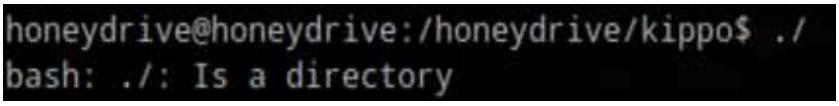

Figure-5: Detection of Honeypot in System

Figure-6: Virtual Shell Behaviour

3) In order to avoid this, we will be using a Honeynet. Honeynets, which is a high interactive honeypot provide a much more robust environment for an attacker to interact with. In order to implement Honeynet on the honeypot, we will be using Sebek.

4) Sebek is a data capture tool designed to capture attacker's activities on a honeypot, without the attacker knowing it. It has two components. The first is a client that runs on the honeypots, its purpose is to capture all of the attacker's activities (keystrokes, file uploads, passwords) then covertly send the data to the server. The second component is the server, which collects the data from the honeypots. The server normally runs on the Honeywell gateway, but can also run independently.

4. CONCLUSION

This can potentially safeguard most cloud-based servers by using a two-layered honeypot system. 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. A honeypot can bring some value to maintain and increase the information security momentum. There are pros and cons to having a Honeypot System installed and used in production. The biggest advantage is that data collected by an owned honeypot system is not just a result of a statistical data but can be a custom set of data, tailored to custom scenarios. But these set of data is just a statistically sorted data and might not apply to your needs. Combating an Advanced Persistent Threat will be easier if you will have an active honeypot system in your ISMS. Another advantage is the list of real usernames and passwords, a custom list that usually is hard to get and trust. In conjunction with a firewall, it will take the security to the next level having real live data it is the real advantage. One of the biggest disadvantages is the fact that it really has to be closely monitored and maintained. Also if it is not properly configured can leave traces it the administration part of it. If a hacker gains access to a honeypot it will be used most of the time as proxies or as a starting point of launching attacks resulting in having a system that does more bad than good. One of the biggest disadvantages is the honeypot itself will be exposed without any security devices IDS or firewall, but we are overcoming this disadvantage by applying a Honeynet over the honeypot itself to safeguard the entire system.

5. REFERENCES

[1] Marius Alin Lihet, Vasile Dadarlat,"How to build a honeypot System in the cloud", 14th RoEduNet International Conference - Networking in Education and Research, pp. 190-194, 2015.
[2] Erin L. Johnson , John M. Koenig , Dr. Paul Wagner (Faculty Mentor), "Development and Implementation of the Honeynet on a University Owned Subnet".
[3] Nithin Chandra S.R, Madhuri T.M, "Cloud Security using Honeypot Systems", International Journal of Scientific & Engineering Research Volume 3, Issue 3, March-2012.
[4] Kumar Shridhar, Nikhil Gautam, "A Prevention of DDoS Attacks in Cloud Using Honeypot", International Journal of Science and Research (IJSR), 2012.
[5] M Balamurugan, B Sri Chitra Poornima, "Honeypot as a Service in Cloud", International Conference on Web Services Computing (ICWSC), 2011.
[6] Brough Davis, "Second Generation Honeynet Honeywell", SANS Institute 2003.

