A review on Honeypot and it's Tools

Swapnil Saxena[1] , Mr. Sandeep Gonnade[2]
M.Tech. Scholar, Department of Computer Science and Engg. MATS University, Raipur INDIA
Assistant Professor, Department of Computer Science and Engg. MATS University, Raipur INDIA
[1]swapnilsaxena2705@gmail.com
[2]sandeep_gonnade@yahoo.co.in

Abstract—**Information security is growing concern today for organizations and individual alike. This led to growing interest in more aggressive forms of defence to supplement the existing methods. One of these methods involves the use of Honeypots. A Honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource. In this paper we present an overview of Honeypots, its type and various tools.**

Keywords- **Honeypot , Types of** Honeypots**, Tools**..

I. INTRODUCTION

In this day and age, information security is an ever increasing concern . The traditional approach to security has been largely defensive so far, but interest is increasingly paid to more aggressive forms of defence. One of these forms is decoy based intrusion protection through the use of Honeypots or honeynets. A honeypot can be defined as a computing resource that has value in being attacked. A honeypot can be defined as a computing resource that has value in being attacked. A honeypot computer system is a system that has been deployed on a network for the purpose of logging and studying attacks on the honeypot. These systems may be made purposely insecure in order to lure attackers to study their techniques, tools, and motivations. On production systems, it can often be difficult to detect attacks that occur within a large amount of legitimate traffic, so an important property of honeypots is that they should serve no other purpose on the network. Any traffic destined to or originating from a honeypot is suspect

This is a broad definition that can be implemented in many ways. Some honeypot system uses software ,some use actual production machines , and some even use virtual machines such as with VMware. Whichever honeypot design method is chosen, the underlying goal is to create a system that appears to be vulnerable. What makes a honeypot different from from other vulnerable computer systems is its extensive logging capability.
Example of Honeypot can be given as

1. Any system installed on a network with a sole purpose to get exploited or compromised.
2. Installing a buggy operating system. For Example the default installation of WinNT4 with IIS with IIS4.
3. Installation of tools for deception such as Honeyd. Honeypots serves several purposes:
    1. They can distract attackers from more valuable machines on a network.
4. They can provide early warning about new attack and exploitation trends
    2. They can provide early warning about new attack and exploitation trends
    3. They allow in-depth examination of adversaries during and after exploitation of a honeypot.

2. Types of Honeypots

Honeypots can be categorized as:

2.1 Purpose of Honeypots

2.1.1 Research Honeypot

A research honeypot is designed to gain information about the blackhat community and does not add any direct value to a organization . They are used to gather information on the general threats organizations may face, allowing the organization to better secure against those threats. Its main objective is to study the way in which the attackers progress and establish attack, it helps understand their motives, behavior and organization Research honeypots are complex to both deploy and maintain and capture extensive amounts of data. They can be very time consuming.

2.1.2 Production Honeypot

A production honeypot is one use within an organization's environment to protect the organization and help mitigate risk. It is more useful because it provides immediate security to organization resources.

It is easier to deploy and maintain than research honeypot because it requires less functionality. Although they identify attack patterns, they give less information about the attackers as compared to research honeypots.

You may learn from which system attackers are coming from and what exploits are being launched, but maybe not who they are, how they are organized, or what tools they are using.

2.2 Level of Interaction

Honeypots can also be categorized based on the level of interaction as follows:

2.2.1 Low Interaction Honeypot

A low-interaction honeypot simulates only services that cannot be exploited to gain total access to the honeypot. On a lowinteraction honeypot, there is no operating system for the attacker to interact with. Honeyd is a low interaction tool.

2.2.2 Medium Interaction Honeypot

It does not have an operating system installed like low interaction honeypot, but the simulated services are

morecomplicated technically. Probability to provide security increases in it.. Medium-interaction honeypots provide the attacker with a better illusion of an operating system since there is more for the attacker to interact with. With its help more complex attacks can therefore be logged and analyzed.
Eg- mwcollect, honeytrap

2.2.3 High Interaction Honeypot

They are the most complex and time-consuming to design, and involve the highest amount of risk because they involve an actual operating system. The objective of a high-interaction honeypot is to provide the attacker with a real operating system to interact with, where nothing is virtual. With this type of honeypot the possibilities for collecting large amounts of information are therefore greater as all actions can be monitored and examined. It needs to be monitored continuously.
Eg- Honeynet.

II. LITE3. Virtual Honeypots

Virtual Honeypot is a software program that is designed to appear to be real functioning system but is actually network developed to be probed and attacked by unauthorized users. In contrast to a Honeypot, thai is typically a hardware device, a virtual Honeypot uses software to pretend to be some network resource. The difference between real and virtual Honeypot is that a virtual Honeypot uses application software to create a new, separate operating system environment. The virtual honeypot actually uses or shares that same hardware as physical OS does. Many virtual servers can be hosted on single hardware device whereas separate device is required for each host in honeypot.

4 . Virtual Honeynet

Virtual Honeynet combines all the component of a Honeynet onto one hardware device. Not solely ar all 3 necessities of information management, information capture, and information assortment met, however the particular Honeypots themselves run on the only system. Honeypots ar actual software package nothing is emulated. The advantage here is one among price and potency. it\'s a lot of easier to deploy and maintain. One such example is Honeynet exploitation Honeyd daemon, Honeyd are often accustomed produce a virtual honeynet or for general network observance. It supports the creation of a virtual topology together with dedicated routes and routers. The routes are often attributed with latency and packet loss to create the topology appear a lot of realistic

5.Data Capture on Honeypots `

Data capture is the capturing of all blackhat’s activities. These activities are then analysed with the help of tools provided by honeypot. The challenges is to capture as much as data possible, without letting blackhat to know that they are being scrutinized. Data capturing cannot cannot be done on host system as they can be compromised; and in that case crackers may come to know about it and will try to erase it or alter it. Hence information must be logged remotely.

Requirement of the data capture

1. Data should not be captured locally.
2. All activity should be captured i.e. network activity, system activity, Application activity, User activity.
3. The ability to view activity in real time.
4. Log should follow a standard pattern for all Honeypots.
5. Resource to capture data must be secured against compromise.

Data captured by Honeypots primarily have following advantage:

SMALL DATA SET: unlike IDS Honeypot does not logs lots of information as it is a passive device and does not capture every information going through the network. Hence reduces false alarm greatly.

HIGH VALUE INFORMATION: All information through Honeypot is supposed to be malicious traffic as it has no production value and does not supposed to be accessed by anybody.

6. TOOLS TO BE USED

The network setup has been implemented using following tools

- Honeyd(Low interaction Honeypot)
- Nmap v4.76(Fingerprinting tools)
- Xprobe(Fingerprinting tools)
- Arpd(ARP reply daemon)
- Honeysum v 0.3(Honeyd log reader)

6.1 Honeyd

Honeyd may be a tiny daemon that makes virtual hosts on a network. The hosts are often designed to run arbitrary services, and their temperament are often tailored in order that they seem to be running bound operational systems. Honeyd allows one host to say multiple addresses. Honeyd are often designed to some faux virtual system on some unused ip addresses in a network. Honeyd improves cyber security by providing mechanisms for threat detection and assessment. It additionally deters adversaries by concealment real systems within the middle of virtual systems.

Some of its options

• Simulates thousands of virtual hosts at identical time.

• Configuration of arbitrary services via straightforward configuration file:
  o Includes proxy connects.
  o Passive procedure to spot remote hosts.
  o sampling for load scaling.

• Simulates operational systems at TCP/IP stack level:
  o Fools nmap and xprobe.

• Simulation of arbitrary routing topologies:
  o Configurable latency and packet loss. o uneven routing.
  o Integration of physical machines into topology.
  o Distributed Honeyd via GRE tunnelling.

• Subsystem virtualization:
  o Run real UNIX system applications beneath virtual Honeyd ip addresses: web servers, ftp servers, etc...2

Dynamic port binding in virtual address space, background initiation of network connections, etc.

6.2 NMAP

NMAP is a free open supply special softwre for exploring networks and performing security checking. It was designed to rapidly scan huge networks. Nmap uses beginning IP packets to see what hosts are available on the network, what services (application name and version) those hosts are running, what Software pachage (and OS versions) they are running, what kind of packet filtersand firewalls are in use, and dozens of alternative characteristics. Nmap runs on most forms of computers, and each console and graphical versions are obtainable. Nmap is open source software, available with full source code.

Example —

nmap –A –T4 132.140.1.173

Scans given information for IP address for OS details and application it is running.

6.3 Xprobe

Remote OS identification using ICMP packets Xprobe permits you to obtain what software package is running on a foreign host. It sends multiple packets to a host and analyses the returned ICMP packets. Xprobe's functionality is comparable to the OS fingerprinting feature in nmap. It has several advantages over it such as Faster: A maximum of 4 packets are sent to determine the remote OS. It can easily detect whether the host is up, so pinging isn't necessary for next time. Stealthier: ill –formed datagrams do not seem to be sended by it and unambiously can uniquely identify differance between many variants of Microsoft operating systems.

6.4 ARPD

ARPD could be a daemon that listens to ARP requests and answers for IP addresses that are unallocated. Arpd is used in conjunction with Honeyd, to confirm that the Host responds to the *arp* request for the IPs of the Honeypots. Arpd responds with the MAC address of the Honeyd host for any request to an unused IP address.

Example-- #arpd
10.0.0.0/8

"arpd" can currently respond with the MAC address of the

Honeyd host for any request to an unused IP in the 10.x.x.x address space.

6.5 Honeydsum

Honeydsum could be a tool written in Perl designed to produce a summary from Honeyd logs.Different parameters can be used to produce summaries as filters, such as ports, protocols, IP addresses or networks. It displays the top source and port access the number of connections per hour, and helps input from several log files. It allows specifying the Honeypot network address and other network address which will be sanitized with its correspondent fake network. There is validation of networks (address and size) for data's sanitize. It is able to produce an abstract in text or in valid HTML. There is also an chance to create graphics illustrating the information showed by the summary. The abstract can also correlate events from several Honeypots.

7. Approaches to Honeypot Implementation

To implement a Honeypot, some factors you need to consider include [5]:

I. Firstly,, type of data that should be made available with the Honeypot, for the Honeypot to masquerade as an authentic system, realistic data needs to be used. However, there are also the consequences to consider when the Honeypot is compromised and the intruder uses the data against the organization. Measures need to be in place to handle such an occasion when it arises.

II. Secondly, how to prevent uplink liability**,** If a Honeypot is compromised, it could be used by the intruder to attack other systems (this is known as uplink liability). There are liability issues to consider if this happens, and preventative measures to take. Legal issues concerning Honeypots will be covered in more detail in the next section.

III. Building criteria**,** The Honeypot owner also has to system to prevent uplink liability. A lot more information on the considerations involved in Honeypot implementation can be found in.

decide between building a Honeypot and purchasing a commercially available one. Financial resources need to be considered. In addition, maintenance Of The Honeypot Requires knowledgeable personnel, as well as a considerable amount of time to examine the data collected by the Honeypot.

8. Conclusions

From this paper we can conclude that a Honeypot is a valuable resource, especially to collect information about proceedings of attackers. It shows how Honeypots deceptechniques can be used as a trustworthy means to learn about strategy and various tools used by attackers. In future we can use honeypot in intrusion detection system to maintain the security of network.

REFERENCES

[1] Dacier Marc, Pouget Fabien and Debar EurecomHervé, "Honeypots: Practical Means to Validate Malicious Fault Assumptions", In the proceedings of the 10th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'2004), February 2004.

[2] McGrew Robert and B. Vaughn Rayford, "Experiences with Honeypot Systems: Development, Deployment, and Analysis", In the proceedings of the 39th Hawaii International Conference on System Sciences, On page(s): 220a- 220a, January 2006.

[3] Mokube Iyatiti and Adams Michele, "Honeypots: concepts, approaches, and challenges", In the proceedings of the 45th annual southeast regional conference (ACM-SE), New York, USA, On Pages(s): 321 – 326, 2007

[4] Spitzner Lance, "Honeypots, definitions and value of Honeypots" http://www.spitzner.net/Honeypots.html, May

2003.

[5] Sink Michael, "The Use of Honeypots and Packet Sniffers for Intrusion Detection", SANS Institute , As part of GIAC practical repository 2000 - 2002

.

.