International Journal of Engineering Research and Development
***e*-ISSN: 2278-067X, *p*-ISSN: 2278-800X, www.ijerd.com**
Volume 2, Issue 10 (August 2012), PP. 23-27

Comparative Study of Different Honeypots System

Ashish Girdhar[1], Sanmeet Kaur[2]
[1]*Student (M.Tech), Patiala Thapar University,*
[2]*Assistant Professor, SMCA Patiala, Thapar University,*

Abstract—A honeypot is a closely monitored network decoy serving several purposes: it can distract adversaries from more valuable machines on a network, provide early warning about new attack and exploitation trends and allow in-depth examination of adversaries during and after exploitation of a honeypot. The concept of honeypots was first proposed in Clifford Stoll's book "The Cuckoo's Egg", and Bill Cheswick's paper "An Evening with Berferd". Honeypots as an easy target for the attackers can simulate many vulnerable hosts in the network and provide us with valuable information of the attackers. Honeypots are not the solution to the network security but they are tools which are implemented for discovering unwanted activities on a network. They are not intrusion detectors, but they teach us how to improve our network security or more importantly, teach us what to look for. Honeypot is a system which is built and set up in order to be hacked. Except for this, honeypot is also a trap system for the attackers which is deployed to counteract the resources of the attacker and slow him down, thus he wastes his time on the honeypot instead of attacking the production systems. This paper discusses honeypots basics, types of honeypots, various honeypots, advantages and disadvantages of honeypots and the last section presents the comparison between different honeypots systems.

Keywords—Honeypots, Honeyd, Specter, Network Security, Honeynet.

I. INTRODUCTION TO HONEYPOTS

In general the term 'honeypot' is usually being used for representing "a container (or pot) of honey". But in the case of computer security, this term is being used to represent a computer security concept that is solely based on deception [2]. Honeypot is a resource to trap the attacker's tools and activities. Lance Spitzner, the founder of The Honeynet Project organization, defines a honeypots as: *"Honeypot is a security resource whose value lies in being probed, attacked or compromised"* [3].

This definition tells the nature of honeypot. It means that if no one attacks honeypot, it is nothing. But honeypot is a valuable security tool if it is being attacked by the attacker. Other security tools such as firewall and IDS are completely passive for that their task is to prevent or detect attacks. Honeypot actively give a way to attacker to gain information about new intrusions. This nature makes honeypot outstanding to aid other security tools. Honeypot differ according to different use. It could be an emulated application, a fully functional operating system with default configuration or an actual net including different OS and applications, even an emulated network on a single machine.

Honeypots are very different, and it is this difference that makes them such a powerful tool. Honeypots do not solve a specific problem. Instead, they are a highly flexible tool that has many applications to security. They can be used to slow down or stop automated attacks, capture new exploits to gather intelligence on emerging threats or to give early warning and prediction. They come in many different shapes and sizes. They can be either a Windows program that emulates common services, such as the Windows honeypot KFSensor3 or entire networks of real computers to be attacked, such as Honeynets.

II. TYPES OF HONEYPOTS

In general honeypots can be divided in to two categories:

- Production honeypots
- Research honeypots[4]

2.1 Production Honeypots

Production honeypots are used to assist an organization in protecting its internal IT infrastructure. They are valuable to the organization especially commercial, as they help to reduce the risk that a specific organization faces. They secure the organization by policing its IT environment to identify attacks. These honeypots are useful in catching hackers with criminal intentions. The implementation and deployment of these honeypots are relatively easier than research honeypots .One of the reasons is that they have less purpose and require fewer functions. As a result, they also provide less evidence about hacker's attack patterns and motives.

2.2 Research Honeypots

Research honeypots are complex. They are designed to collect as much information as possible about the hackers and their activities. They are not specifically valuable to an organization. Their primary mission is to research the threats organization may face, such as who the attackers are, how they are organized, what kind of tools they use to attack other systems, and where they obtained those tools. While production honeypots are like the police, research honeypots act as their

intelligence counterpart and their mission is to collect information about the attacker. The information gathered by research honeypots will help the organization to better understand the hackers attack patterns, motives and how they function. They are also an excellent tool to capture automated attacks such as worms.

III. CLASSIFICATION OF HONEYPOTS

According to the level of involvement between the attacker and the honeypots, the honeypots can be divided into three categories:

- Low-interaction honeypots
- Medium-interaction honeypots
- High-interaction honeypots.

3.1 Low-Interaction Honeypots

Low-interaction honeypots are the easiest to install, configure, deploy, and maintain because of their simple design and basic functionality. Normally these technologies merely emulate a variety of services. The attacker is limited to interacting with these pre designated services. For example, a low-interaction honeypot could emulate a standard Unix server with several running services, such as Telnet and FTP. An attacker could Telnet to the honeypot, get a banner that states the operating system, and perhaps obtain a login prompt. The attacker can then attempt to login by brute force or by guessing the passwords. The honeypot would capture and collect these attempts, but there is no real operating system for the attacker to log on to. The attacker's interaction is limited to login attempts.

Since low-interaction honeypots are simple, they have the lowest level of risk. There is little functionality offered, there is less to go wrong. There is also no operating system for the attacker to interact with, so the honeypot cannot be used to attack or monitor other systems. Low-interaction honeypots are easy to deploy and maintain because they have limited interaction capabilities, which also reduces risk [5].

3.2 Medium-interaction Honeypots

In terms of interaction, medium-interaction honeypots are more advanced than low-interaction honeypots, but less advanced than high interaction honeypots. Medium-Interaction honeypots also do not have a real operating system, but the services provided are more sophisticated technically. Here the levels of honeypots get complicated so the risk also increases especially with regards to vulnerability.

3.3 High-interaction Honeypots

High-interaction honeypots are different; they are a complex solution and involve the deployment of real operating systems and applications. They capture the extensive amounts of information and allowing attackers to interact with real systems where the full extent of their behavior can be studied and recorded. Examples of high-interaction honeypots include Honeynets and Sebek. These kinds of honeypots are really time consuming to design, manage and maintain. Among the three types of honeypots, these honeypots possess a huge risk. But, the information and evidence gathered for analysis is very large. With these types of honeypots we can learn what are the kind of tools hackers use, what kind of exploits they use, what kind of vulnerabilities they normally look for, their knowledge in hacking and surfing their way through operating systems and how or what the hackers interact about[5].

IV. TRADEOFFS BETWEEN HONEYPOT LEVELS OF INTERACTION

Table 1 summarizes the tradeoffs between different levels of interaction in four categories. The first category is installation and configuration effort, which defines the time and effort in installing and configuring the honeypot. In general, if the level of interaction between the user and the honeypot is more then the effort required to install and configure the honeypot is also significant.

The second category is deployment and maintenance. This category defines the time and effort involved in deploying and maintaining the honeypot. Once again, the more functionality provided by the honeypot, the more is the effort required to deploy and maintain the honeypot.

The third category is information gathering which means how much information can the honeypot gain on attackers and their activities? High-interaction honeypots can gather vast amounts of information, whereas low-interaction honeypots are highly limited.

Finally, level of interaction impacts the amount of risk introduced. The greater the level of interaction, the more functionality provided to the attacker and the greater the complexity. Combined, these elements can introduce a great deal of risk. On the other hand, low-interaction honeypots are very simple and offer a little interaction to attackers and thus a very little risk is associated with them.

Table 1: **Tradeoffs between Honeypot Levels of Interaction [1]**

Degree of involvement	Low	Medium	High
Installation and configuration effort	Easy	Medium	Difficult
Deployment and maintenance effort	Easy	Medium	Difficult
Information Gathering	Limited	Medium	Extensive
Level of Risk	Low	Medium	High

V. HONEYPOTS SYSTEMS

Five honeypots are discussed in the following section.

- ManTrap
- Back officer friendly
- Specter
- Honeyd
- Honeynet

5.1 ManTrap

ManTrap is a high-interaction commercial honeypot created, maintained, and sold by Recourse Technologies. ManTrap creates a highly controlled operating environment that an attacker can interact with. It creates a fully functional operating system containing virtual cages rather than a limited operating system. The cages are logically controlled environments from which the attacker is unable to exit and attack the host system. However, instead of creating an empty cage and filling it with certain functionality ManTrap creates cages that are mirror copies of the master operating system. Each cage is a fully functional operating system that has the same capabilities as a production installation.

This approach creates a very powerful and flexible solution. Each cage is its own virtual world with few limitations. An administrator can customize each cage as he would a physically separate system. He can create users, install applications, run processes, and even compile his own binaries. When an intruder attacks and gains access to a cage, to the attacker it looks as if the cage is a truly separate physical system. He is not aware that he is in a caged environment where every action and keystroke is recorded [6].

5.2 BackOfficer Friendly (BOF)

BackOfficer Friendly, or BOF as it is commonly called, is a simple, free honeypot solution developed by Marcus Ranum. It is extremely simple to install, easy to configure, and low maintenance. However, this simplicity comes at a cost. Its capabilities are severely limited. It has a small set of services that simply listen on ports, with notably limited emulation capabilities.

It works by creating port listeners, or open sockets, that bind to a port and detect any connections made to these ports. When a connection is made to the port, the port listeners establish a full TCP connection (if the service is TCP), log the attempt, generate an alert, and then close the connection, depending on how the service is configured. Everything BOF does happen in user space. It does not build or customize any packets when responding to connections. Because of this simple model, BOF can run on any Windows platform, including Windows 95 and Windows 98[1].

5.3 Specter

Specter is a commercially supported honeypot developed and sold by the folks at NetSec. Like

BOF, Specter is a low-interaction honeypot. However, Specter has far greater functionality and capabilities than BOF. Not only can Specter emulate more services, it can emulate different operating systems and vulnerabilities. It also has extensive alerting and logging capabilities. Because Specter only emulates services with limited interaction, it is easy to deploy, simple to maintain, and is low risk. However, compared to medium- and high-interaction honeypots, it is limited in the amount of information it can gather. Specter is primarily a production honeypot. Specter shares the same limitations as BOF. Specifically, it cannot listen on or monitor a port that is already owned by another application. If any service listening on the FTP port (port 21), then Specter is unable to monitor on that port. Specter can only monitor ports that are not owned by any other applications. It also has the capability of emulating different operating systems. This is done by changing the behavior of the services to mimic the selected operating system [6].

5.4 Honeyd

Honeyd is developed and maintained by Niels Provos of the University of Michigan and was first released in April 2002. It is designed as a low-interaction solution; there is no operating system intended for an attacker to gain access to, only emulated services. Honeyd is designed primarily as a production honeypot, used to detect attacks or unauthorized activity [1].

Honeyd works on the principle that when it receives a probe or a connection for a system that does not exist, it assumes that the connection attempt is hostile, most likely a probe, scan, or attack. When Honeyd receives such traffic, it assumes the IP address of the intended destination (making it the victim). It then starts an emulated service for the port that the connection is attempting. Once the emulated service is started, it interacts with the attacker and captures all of his activity. When the attacker is done, the emulated service exits and is no longer running. Honeyd then continues to wait for any more traffic or connection attempts to systems that do not exist. Honeyd assumes an IP address and runs an emulated service only when it receives a connection attempted for a system that does not exist, an extremely efficient method. As Honeyd receives more attacks, it repeats the process of assuming the IP address of the intended victim, starting the respective emulated service under attack, interacting with the attacker, and capturing the attack, and finally exiting. It can emulate multiple IP addresses and interact with different attackers all at the same time.

5.5 Honeynets

Honeynets represent the extreme of high-interaction honeypots. Not only does it provide the attacker with a complete operating system to attack and interact with, it may also provide multiple honeypots. Honeynets are nothing more than a variety of standard systems deployed within a highly controlled network. By their nature, these systems become

honeypots, since their value is in being probed, attacked, or compromised. The controlled network captures all the activity that happens within the Honeynet and decreases the risk by containing the attacker's activity.

Honeynets are a simple mechanism that works on the same principle as a honeypot. You create a resource that has little or no production traffic. Anything sent to the Honeynet is suspect, potentially a probe, scan, or even an attack. Anything sent from a Honeynet implies that it has been compromised— an attacker or tool is launching activity. However, Honeynets take the concept of honeypots one step further: Instead of a single system, a Honeynet is a physical network of multiple systems.

Honeynets are not a product you install or an appliance you drop on your network. Instead, Honeynets are an architecture that builds a highly controlled network, within which you can place any system or application you want [7].

Table 2: **Advantages and disadvantages of various honeypots [1]**

Name of Honeypot	Advantages	Disadvantages
ManTrap	Provides response mechanism based on frequency analysis and shuts down machines by monitoring increased hacker activity. Provides stealth monitoring and thus live attack analysis. Detects both host and network based intrusions	Need highly skilled expertise to maintain and deploy these kinds of honeypots. Even with that, the risk involved for getting compromised remains and if these are connected to the production servers a thorough risk analysis has to be done.
BOF	Easy to install, configure and maintain. Runs on any windows or Unix based platform. Little risk due to simplicity.	Limited to seven ports on which it can detect attacks. Ports cannot be customized, increasing the possibility of fingerprinting. No remote logging, alerting or configuring personality.
Specter	Easy to install, configure and deploy. Extensive service emulation. Monitors twice as many ports as BOF. Outstanding notification capabilities.	Monitors only 14 ports. Preprogrammed emulated services are limited to interacting with known behavior. Limitations on information collected, mainly to transactional information and the attacker's interaction with the seven emulated services.
Honeyd	Can monitor any TCP or UDP port and entire networks. As an open source solution, it is free and will develop quickly with the input and development of others in security community. Resist fingerprinting efforts by emulating operating system at IP stack level.	As a low interaction solution, it cannot provide real operating system for attackers to interact with. As an open source solution, it provides no formal support for maintenance and troubleshooting. No built in mechanism for alerting
Honeynets	Flexibility-any system can be placed in Honeynet environment. Extensive data capture capabilities for both known and unknown tools and tactics. Adaptable to many organizations and environments.	Complexity of deployment and resources required to maintain. High interaction functionality, introduces the risk of attackers using the systems to attack, harm other system. New and immature technologies have a greater risk of breaking and introducing errors.

VI. COMPARISON OF VARIOUS HONEYPOTS:

In this section five honeypots are compared in the tabular form.

- The interaction level between the user and the honeypot is high in case of Mantrap, specter and Honeynet and this level is low in case of BOF and honeyd.
- Honeyd and Honeynet are freely available whereas Mantrap, specter and BOF are not freely available.
- Honeyd and Honeynet are open source whereas Mantrap, specter and BOF are not open source.
- BOF does not support Log file whereas rest of the honeypots support log file.
- BOF does not emulate the operating system whereas rest of the four honeypots can emulate operating system.
- Unlimited services are supported by the ManTrap, Honeyd and Honeynet whereas limited services are supported by the BOF and specter.

Table 3: **Comparison of various honeypots**

	ManTrap	**BOF**	**Specter**	**Honeyd**	**Honeynet**
Interaction Level	High	Low	High	Low	High
Freely Available	No	No	No	Yes	Yes
Open Source	No	No	No	Yes	Yes
Log file Support	Yes	No	Yes	Yes	Yes
OS Emulation	Yes	No	Yes	Yes	Yes
Supported Services	Unrestricted	7	13	Unrestricted	Unrestricted

VII. CONCLUSION

Honeypots are the security resources that can help in achieving network security. Different honeypots systems have been discussed in the paper. An effort has also been made to compare the different systems. Each honeypot has its advantages and disadvantages. Different honeypot system can be deployed under different conditions. An administrator can choose any of the five honeypots discussed in the paper according to his requirements.

REFERENCES

[1]. Spitzner, L.: *Tracking Hackers*. Addison Wesley, September 2002.

[2]. Zanoramy, W., Zakaria, A., et.al,"Deploying Virtual Honeypots on Virtual Machine Monitor".

[3]. Spitzner, L. Honeypot: Definitions and Values. May, 2002. http://www.spitzner.net.

[4]. Levin, J., Labella, R. Henry,: "The Use of Honeynets to Detect Exploited Systems Across Large Enterprise Networks", IEEE Proceedings, June 2003.

[5]. Qassrawi, M., Hongli, Z. "Deception methodology in virtual Honeypots", Second International Conference on Network Security, Wireless Communication and Trusted Computing, 2010.

[6]. Bao, J., Gao, M. "Research on network security of defense based on Honeypot", International Conference on Computer Applications and System Modelling, 2010.

[7]. Levine, J., Grizzard, J. "Using honeynets to protect large enterprise networks," Security & Privacy Magazine, IEEE, vol. 2, pp. 73-75, 2004

[8]. Kuwatly, I., Sraj, M. A Dynamic Honeypot Design for Intrusion Detection . http://webfealb.fea.aub.edu.lb/proceedings/2004/SRC-ECE-04.pdf.

[9]. Provos, N. A Virtual Honeypot Framework,2004 http://www.citi.umich.edu/u/ Provos/papers/honeyd.pdf.

[10]. Lanoy, A., and Romney, G.W.: "A Virtual Honey Net as a Teaching Resource", Information Technology Based Higher Education and Training, 2006. ITHET'06. 7th International Conference on, 2006, pp. 666-669