IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET)
ISSN(E): 2321-8843; ISSN(P): 2347-4599
Vol. 2, Issue 5, May 2014, 33-42

HONEYPOTS FOR NETWORK SURVEILLANCE

NIHARIKA[1] & RANJEET KAUR[2]

[1]Research Scholar, Department of Computer Science & Engineering, DAV University, Jalandhar, Punjab, India

[2]Assistant Professor, Department of Computer Science & Engineering, DAV University, Jalandhar, Punjab, India

ABSTRACT

A honeypot is a non-production system which offers sweet bait to the intruders, blackhat community [1] that can enhance the ability of system administrators to identify system vulnerabilities. This paper presents a survey on recent advances in honeypot research from a review of 20+ papers on honeypots and related topics. A recent technology in the area of intrusion detection is honeypot technology that unlike common IDSs tends to provide the attacker with all the necessary resources needed for a successful attack. Honeypots provide a platform to study the approaches and tools used by the intruders, thus acquiring their value from the unauthorized use of their resources.

KEYWORDS: Honeypots, Intrusion Detection System, Security, Legal Issues

INTRODUCTION

The underlying goal of computer security is to defend computers against attacks launched by malicious users. There are a numerous ways in which researchers and developers can work to protect the software that they write. Some are proactive, like code reviews and regression testing, while others are reactive, like the pwn2own contest where new vulnerabilities are used to exploit browsers. One class of tools that can take on aspects of both is **honeypots**. The term honeypot or honey trap was used during the cold war as a name for employing ensnarement to gain information from an enemy. In computer terminology, a **honeypot** is a trap set to detect, deflect, or, in some manner, counteract attempts at unauthorized use of information systems. From few research papers, we come to know about, the Cuckoo's Egg where Cliff Stoll's hunt for a hacker using honeypot like methods are used. He posted fake data he knew the hacker would find interesting to keep the hacker occupied in his system while he was tracing him. Thanks to these medications which gave accurate information about various types of attacks which can be recorded. The term honeypot was first presented by Lance Spitzner in 1999 [2] in a paper titled "To Build a Honeypot".

The idea behind these systems is to provide systems or services that deceive the intruder. Honeypots can be used as tools to gather information which can be used to enforce and strengthen existing intrusion detection tools or network firewalls. Honeypots should not be viewed as a solution to network security; they should be seen as an aid to it.

INTRUDERS

A person who intrudes i.e. who puts or forces in inappropriately, especially without invitation or permission. Thus an intruder can be defined as somebody attempting to break into an existing computer. An outsider attack is an attack from a person who is not a member of the organization. Usually the intruder is a hacker whose intensions are to cause harm or mischief.

Intruders are classified into two types [3]:

- One who has something to gain by the intrusion and
- A curious person trying to probe the security of the system.

The first type is popularly termed as a "**cracker**". Crac34kers attack web-sites or database servers in an attempt to gain critical information such as credit card or social security information. Some try to deface government websites or deny normal service and may be backed by political motive. The second type is the "**hacker**" who can be further broken down into two types:

- An extremely intelligent computer knowledgeable person.
- Script kiddie.

An **intelligent hacker** is one who studies protocols and algorithms and tries to detect vulnerabilities in them. There is nothing malicious about this type although his curiosity and intent is often criticized by many security analysts as irresponsible behavior. A script kiddie is often, but not always, a juvenile hacker; an attacker who uses scripts or programs developed by more sophisticated hackers or crackers. Oftentimes the underlying motivation for a script kiddies attack is simply to garner the attention of peers. Script Kiddies are generally looked down upon by the hacking community for their lack of self-taught skills and reliance upon premade exploit programs and files. Script kiddies cut and paste code written by others without having or desiring an understanding of how the code actually works and generally only care what the code can do for them.

INTRUSION DETECTION SYSTEMS

An intrusion detection system (IDS) is a device or software application that inspects all incoming and outgoing network activity and identifies skeptical figures that may indicate a network or system attack from someone attempting to break into or compromise a system. This identity is popularly termed as a hacker, blackhat or cracker. IDS come in a variety of "flavors" and approach the goal of detecting suspicious traffic in different ways. There are network based (NIDS), Network Node Intrusion detection system (NNIDS) and host based (HIDS) intrusion detection systems [4]. Some systems may attempt to stop an intrusion attempt but this is neither required nor expected of a monitoring system (Figure 1).

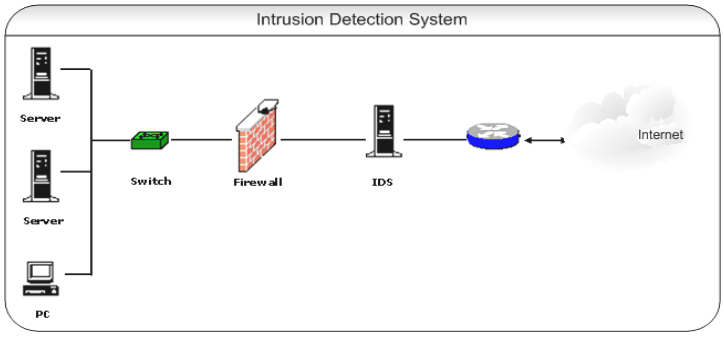

Figure 1: Intrusion Detection System

Intrusion detection and prevention systems (IDPS) are primarily focused on identifying possible incidents, logging information about them, and reporting attempts. In addition, organizations use IDPSes for other purposes,

such as identifying problems with security policies, and deterring individuals from violating security policies. IDPSes have become a necessary addition to the security infrastructure of nearly every organization. IDPSes typically record information related to observed events notify security administrators of important observed events and produce reports. Many IDPSes can also respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which involve the IDPS stopping the attack itself, changing the security environment (e.g. reconfiguring a firewall) or changing the attack's content.

TYPES OF HONEYPOTS

CLASSIFICATION BASED ON THEIR INTERACTION WITH INTRUDER

Bailey [5] integrated low and high-interaction honeypots to solve the trade-off problem between width of honeypot coverage (the advantage of low-interaction honeypots) and behavioral fidelity (the advantage of high-interaction honeypots). These categories help to understand what type of honeypot one is dealing with, its strengths, and weaknesses. Interaction defines the level of activity a honeypot allows to an attacker.

Low-Interaction Honeypots

Low-interaction honeypots are the easy to install, configure, deploy, and maintain with minimal risk because of their simple design and basic functionality. Low-interaction 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. 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 record only limited information and are designed to capture known activity. For an attacker, a low interaction honeypot is easily detected. Examples of low-interaction honeypots include Specter and Honeyd. Also, HosTaGe [6] is a Low-Interaction honeypot for mobile devices.

High-Interaction Honeypots

High-Interaction honeypots are a far more complex solution and typically involve the deployment of real operating system and applications. The major advantages associated with a high- interaction honeypot is the capture of extensive amounts of information [7]. By allowing the attackers to interact with real systems, the full extent of their behavior can be studied and recorded. Examples of high-interaction honeypots include Mantrap and Honeynets.

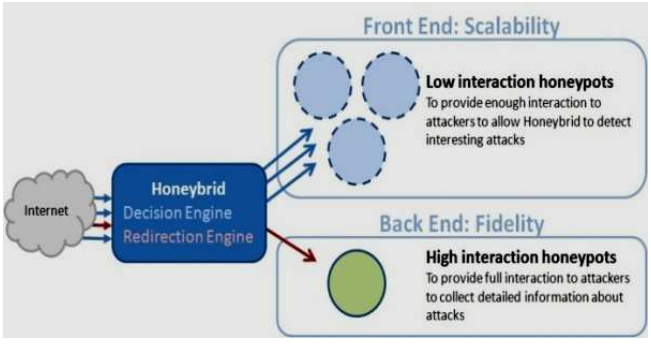

Figure 2: Classification Based on Interaction with Intruder

Honeybrid: Honeybrid is a network application used to provide the hybrid functionality of combining low and high interaction honeypots.

Table 1: Comparison between Low and High-Interaction

Low-Interaction	High-Interaction
Solution emulates operating systems and services	No emulation, real OS and services are provided.
Captures limited amounts of information.	Can capture far more information
Minimal risk, as the emulated services controls attackers.	Increased risk, as attackers are provided real OS to interact with.
Easy to install and deploy.	Can be complex to install or

CLASSIFICATION BASED ON THEIR DEPLOYMENT

This classification illustrated by Sadasivam [8] is as under:

- Production honeypots
- Research honeypots

Production Honeypots

According to Verma [9], the concept of production honeypots is to emulate real production systems and have attackers spend time and resource attacking them as opposed to the production or critical systems and to learn the way they exploit vulnerabilities in production environment. Production honeypots are easy to use, capture only limited information, and are used primarily by companies or corporations. Production honeypots are placed inside the production network with other production server by organization to improve their overall state of security. Normally, production honeypots are low- interaction honeypots, which are easy to deploy. They give less information about the attacks or attackers than research honeypots do. The purpose of a production honeypot is to help mitigate risks in an organization. The honeypot adds value to the security measures of an organization.

Research Honeypots

Research honeypots are run by a volunteer, non-profit research organization or an educational institution to gather information about the motives and tactics of the Blackhat community targeting different networks. These honeypots do not add direct value to a specific organization. Instead they are used to research the threats organization face, and to learn how to protect better against those threats. Research honeypots are complex to deploy and maintain [10], capture extensive information and are used primarily by research, military or government organization.

Table 2: Comparison between Production Honeypots and Research Honeypots

Production Honeypots	Research Honeypots
Captures only limited information	Collects as much information as possible about the hackers and their activities
Primarily used by companies or corporations	Run by a volunteer, non-profit research organization or an educational institution
Acts as police i.e. Useful in catching hackers with criminal intentions	Acts as intelligence counterpart and their mission is to collect information about the attacker
Easy to use and easy to deploy	Comparatively complex

CLASSIFICATION BASED ON THEIR PHYSICAL PRESENCE IN THE NETWORK

By Jiang and Wang [11], honeypots can be classified as:

- Hardware based honeypots
- Software based honeypots

Hardware Based Honeypots

Hardware-based honeypots are servers, switches or routers that have been partially disabled and made attractive with commonly known misconfigurations. They sit on the internal network, serving no purpose but to look real to outsiders. The operating system of each box however has been subtly disabled with tweaks that prevent hackers from really taking it over or using it to launch newattacks on other servers.

Software Emulation Honeypots

Software emulation honeypots are elaborate deception programs thatmimic real Linux or other servers and can run on machines as low-power as a 233-MHz PC. Since an intruder is just dancing with a software decoy, at no time does he come close to actually seizing control of the hardware, no matter what the fake prompts seem to indicate.

Even if the hacker figures out that it's a software honeypot, the box on which it's running should be so secure or isolated that he couldn't do anything but leave anyway. Software emulation might be more useful for corporate environments where business secrets are being safeguarded.

Table 3: Comparison between Hardware Based and Software Emulation Honeypots

Hardware Based Honeypots	Software Emulation Honeypots
They sit on the internal network, serving no purpose but to look real to outsiders	They mimic real servers
Servers, switches, routers can be used	Can run on machines as low-power as a 233-MHz PC
Prevent hackers from really taking it over or using it to launch new attacks on other servers	Hacker if trapped under this couldn't do anything but leave

APPROACHES TO HONEYPOT IMPLEMENTATION

To implement a honeypot, Mokube and Adams [12] has given following factors which shall be included:

- **Availability of Data Kind through the Honeypot**

For the honeypot to masquerade as an authentic system, real data is 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.

- **Prevention of 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.

- **Building of Honeypot**

The honeypot owner also has to 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 considerable amount of time to examine the data collected by the honeypot.

- **Best Location for Your Honeypot**

According to experts, isolating the honeypot from your production system would prevent uplink liability.

LEGAL ISSUES

In the past there has been some confusion on what are the legal issues with honeypots. There are several **reasons** for this.

- First, honeypots is relatively a new concept.
- Second, honeypots come in many different shapes and sizes which accomplish different goals. Hence, different legal issues are applied based on the different uses of honeypots.
- Last, there are no precedents for honeypots. There are no legal cases recorded on the issues. The law is developed through cases. Without cases directly on point, we are left trying to predict, based on cases in other contexts, how courts will treat honeypots. Until a judge gives a court order, we will really never know.

Using Honeypots: Illegal or Not

This question cannot be answered in a single document; there are far too many variables. For example, the country you reside in determines what legal statutes, regulations, or case laws apply to you. The legal processes, procedural rules, and substantive law in each country can differ significantly as they relate to information security, information collection, and specifically to application of honeypot technologies. The legality of you honeypot depends upon the type information you are collecting and its intended use [13].

In my review there are at least three legal issues that you must consider:

- **Entrapment:** This issue is simplest of all other issues. Honeypots are not a form of entrapment. Entrapment, by definition is "a law-enforcement officers or government agent's inducement of a person to commit a crime, by means of fraud or undue persuasion, in an attempt to later bring a criminal prosecution against that person."
- **Privacy:** Moving on from the simplest (entrapment) to the most complex, privacy. Honeypots can capture huge amount of information about attackers, which can potentially violate their privacy. The risk is more with high interaction honeypots. As an example, IDS sensor that is used for detection and capturing network activity is doing so as to detect (and thus enable organizations to respond to) unauthorized activity. Such a technology is most likely not considered a violation of privacy.
- **Liability:** The third issue is liability. This realizes that attackers may misuse your honeypot to harm others. Simply, liability means you could be sued if your honeypot is implemented to harm others in anyway.

One thing to keep in mind, for years legal experts have been discussing possible liability for an organization that

has been compromised and in turn was used to attack, compromise, or harm another system or organization. To date, we have seen no published decision addressing whether the operator of an insecure system can be liable to other operators for the misuse of the system by a hacker. So while liability is an issue, it may be an overblown one, as there is no recorded case of it happening with compromised systems.

EXAMPLES OF HONEYPOT SYSTEMS

Examples of honeypots include:

- **Back Officer Friendly (BOF):** It is developed by Marcus Ranum and crew at NFR. This is one of the simplest honeypots to use. Functionally it is easy to understand and configure. Anyone can use virtual BOF [14].
- **Deception Toolkit:** DTK was the first Open Source honeypot released in 1997. It is a collection of Perl scripts and C source code that emulates a variety of listening services. Its foremost purpose is to deceive human attackers [15].
- **LaBrea:** This is designed to slow down or stop attacks by acting as a sticky honeypot to detect and trap worms and other malicious codes. It can run on Windows or UNIX [15, 16].
- **Honeywall CDROM:** The Honeywall CDROM is a bootable CD with a collection of open source software. It makes honey net deployments simple and effective by automating the process of deploying a honey net gateway known as a Honeywall. It can capture, control and analyses all inbound and outbound honey net activity [17].
- **Honeyd:** Yet low-interaction Open Source honeypot, this is a powerful, and can be run on both UNIX-like and Windows platforms. It can monitor unused IPs, simulate operating systems at the TCP/IP stack level, simulate thousands of virtual hosts at the same time, and monitor all UDP and TCP based ports[18].
- **Honeytrap:** This is a low-interactive honeypot developed to observe attacks against network services. It helps administrators to collect information regarding known or unknown network-based attacks [19].
- **Honey C:** This is an example of a client honeypot that initiates connections to a server, aiming to find malicious servers on a network. It aims to identify malicious web servers by using emulated clients that are able to solicit the type of response from a server that is necessary for analysis of malicious content [20].
- **Honey Mole:** This is a tool for the deployment of honeypot farms, or distributed honeypots, and transport network traffic to a central honeypot point where data collection and analysis can be undertaken [21].
- **Specter:** This is produced commercially whose value lies in detection. It is created and supported by NetSec, a network security company based in Switzerland. Conceptually, it resembles BOF where attackers have no operating system to access [22].

APPLICATIONS

Honeypots can be used in different areas of system security which include network allurement and detecting and countering of worms.

Network Allurement

The traditional role of a honeypot is that of a network decoy. The framework can be used to instrument the unallocated addresses of a production network with virtual honeypots. Adversaries that scan the production network can potentially be confused and deterred by the virtual honeypots.

Detecting and Countering Worms

Honeypots are ideally suited to intercept traffic from adversaries that randomly scan the network. This is especially true for Internet worms that use some form of random scanning for new targets, e.g. Blaster, Code Red, Nimda, Slammer, etc. A virtual honeypot deployment can be used to detect new worms and how to launch active counter measures against infected machines once a worm has been identified.

To intercept probes from worms, virtual honeypots are instrumented on unallocated network addresses.

The worm propagation chance depends on the worm propagation algorithm, the number of vulnerable hosts and the size of the address space.

In general, the larger the honeypot deployment the earlier one of the honeypots receives a worm probe. To detect new worms, Honeyd framework can be used in two different ways. A large number of virtual honeypots have to be deployed as gateways in front of a smaller number of high-interaction honeypots.

CONCLUSIONS

In this growing Information Technology arena, there is a need to strengthen its security. Honeypots are thus the security resources that can help to achieve 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 own advantages and disadvantages. Different honeypot system can be deployed by the administrator under different conditions according to his requirements.

REFERENCES

1. Yaser Alosefer, Omer Rana, Honeyware: a web-based low interaction client honeypot, 2010.
2. Lance Spitzner, Honeypots: Catching the Insider Threat, 1999.
3. Yogendra Kumar Jain, Surabhi Singh, Honeypot based Secure Network System, 2011.
4. Scarfone and P. Mell, Guide to Intrusion Detection and Prevention Systems (IDPS), 2007.
5. Michael D. Bailey, “Data Reduction for the Scalable Automated Analysis of Distributed Darknet Traffic,” 2005
6. Emmanouil Vasilomanolakis, Shankar Karuppayah, Mathias Fischer, Max Mühlhäuser HosTaGe - a Low-Interaction Honeypot for Mobile Devices, 2013.
7. Abdulrazaq Almutairi, Survey of High Interaction Honeypot Tools: Merits and Shortcomings, 2012.
8. K. Sadasivam, et al., "Design of network security projects using honeypots," Journal of Computing Sciences in Colleges, vol. 20, pp. 282-293, 2005.
9. Abhilash Verma, Production Honeypots: An Organization’s view, 2003

10. Simon Mavsar, *Research honeypot - a prototype*, 2008.
11. Xuxian Jiang, Dongyan Xu, Yi-Min Wang, Collapsar: A VM-Based Honeyfarm and Reverse Honeyfarm Architecture for Network Attack Capture and Detention, 2006.
12. Iyatiti Mokube, Michele Adams, Honeypots: Concepts, Approaches, and Challenges, 2006.
13. Bradley J. S. Chaufenbuel, The Legality of Honeypots, 2008.
14. Marcus hum. Backofficer Friendly (BOF), 2008, http://www.nfr.net/products/.
15. Lance Spitzner, Honeypots: *Definitions and Value of Honeypots*, 2003
16. http://labrea.sourceforge.net/labrea-info.html
17. http://www.honeynet.org/tools/cdrom/
18. Niels Provos. A virtual honeypot framework. In Proceedings of the 13th conference on USENIX Security Symposium - Volume 13, SSYM'04, pages, Berkeley, CA, USA, 2004.
19. Alec Yasinsac and Yanet Manzano, Honeytraps, A Network Forensic Tool.
20. http://www.symantec.com/business/support/documentation.jsp?language=english&view=manuals &pid=51899
21. http://www.honeynet.org.pt/index.php/HoneyMole
22. Spector. http://www.specter.com/defauItSO.htm
23. Wikipedia. http://en.wikipedia.org/wiki/Honeypot_ (computing).
24. Know Your Enemy: Honeynets. http://www.honeynet.org/papers/kye.html.