Honeypots and Spam

Ahmed Obied
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada, T2N 1N4
obieda@cpsc.ucalgary.ca

Abstract

Honeypots are closely monitored computing resources that can provide early warning about new vulnerabilities and exploitation techniques, distract attackers from valuable computer systems, or allow in-depth examination of attackers during and after exploitation of a honeypot. Extensive research into honeypot technologies has been done in the past several years to provide better countermeasures against malicious attacks and track attackers. This paper describes honeypots in-depth and discusses how honeypots can be used to fight spam and spammers effectively.

1 Introduction

The use of computer systems increased tremendously in the last few years and millions of users joined this technological revolution due to the creation of the Internet that made the world look so small and at our disposal. The widespread use of the Internet caused the number of warnings being made about the dark side of our technological revolution to increase and we are becoming uniquely vulnerable to many mysterious and malicious threats. Malicious attacks on computer systems are used to spread mayhem, enact political revenge on a corporate target, steal data, increase access to a network resource, hijack networks, deny companies use of their networks, or sometimes simply gain bragging rights. Malicious attacks are getting smarter, more widespread and increasingly difficult to detect, and dozens more are added to the menagerie each day.

Identifying and classifying the type of a malicious attack is a crucial step in developing strategies to defend against it. However, the wide range of computer hardware, the complexity of operating systems, the variety of potential vulnerabilities, and the skill of many attackers combine to create a problem that is extremely difficult to address. As a result, exploitation of newly discovered vulnerabilities often catches us by surprise [18]. Exploit automation and massive global scanning for vulnerabilities enable attackers to compromise computer systems shortly after vulnerabilities become known [18].

To stay one step ahead and get early warnings of new vulnerabilities and exploits, one can use honeypots. Honeypots are a powerful, new technology with incredible potential [23]. Honeypots can do everything from detecting new attacks never seen in the wild before, to tracking botnets, automated credit card fraud, and spam.

In this paper, we present a survey on honeypots. We discuss their history, types, purpose, and value. We also present an in-depth discussion of how honeypots can be used to fight spam and spammers.

2 Definition of Honeypots

Many definitions for a honeypot exist. The most accurate definition is the one used by Lance Spitzner [23]. Spitzner defines a honeypot as an information system resource whose value lies in unauthorized or illicit use of that resource. The information system resource does not have any production value and should see no traffic because it has no legitimate activity [22]. The real value of a honeypot is determined by the information we can obtain from it. If the attacker does not interact with or use the honeypot, then it has little or no value. This is very different from most security mechanisms such as firewalls, IDS, PKI certificate authority since the last thing you would want an attacker to do is interact with such mechanisms [23]. All the activities on a honeypot, and the traffic that enters and leaves it is closely monitored. Since a honeypot does not have any production value, all incoming and outgoing traffic is considered suspicious.

A honeypot lures attackers by pretending to be an important host hidden in the network topology that contains interesting and valuable information or services. For example, an interesting system name, large number of user accounts, huge number of data, vulnerable services, etc [11]. Honeypots help security professionals and researchers learn the techniques used by attackers to compromise computer systems. Honeypots can do

everything from detecting new attacks never seen in the wild before such as zero-day exploits, to tracking automated credit card fraud and identify theft [22].

3 History of Honeypots

The first article that described a honeypot approach in luring and capturing an attacker was published in 1988 by Clifford Stoll [24]. Markus Hess, a West German citizen, was a computer prodigy and particularly effective cracker who was recruited by the KGB to be an international spy with the objective of securing United States military information for the Soviets. In 1986, Hess attacked the Lawrence Berkeley Laboratory (LBL). Stoll, who was working as a systems administrator of the computer centre of the LBL in California, discovered that someone had obtained root privileges on one of the LBL systems. Instead of trying to keep Hess out, Stoll took a novel approach of allowing him access while he printed out his activities and traced him, with the help of local authorities, to his source [24].

Bill Cheswick [9], who was working at AT&T Bell Laboratories in 1991, discovered that an attacker was trying to exploit the famous *sendmail* DEBUG security hole to gain access to the Internet gateway of AT&T Bell Laboratories. Cheswick lured the attacker into believing that he exploited the security hole, and used the UNIX *chroot* and *jail* tools to monitor the attacker's keystrokes and study his techniques.

Steven Bellovin published a paper [7] in 1992 that described his experience with honeypots. Bellovin replaced most of the standard servers at AT&T Bell Laboratories with a variety of trap programs that look for attacks. Using this approach, Bellovin detected a wide variety of pokes ranging from simple doorknob-twisting such as simple attempts to log in as guest to determined assaults such as forged NFS packets [7].

Between the years 1997 and 2006, a number of honeypot solutions have been released. Fred Cohen released the *Deception Toolkit* in 1997 that emulates a variety of known vulnerabilities with a collection of PERL scripts. The Deception Toolkit is known to be one of the original and landmark honeypots [2]. In 1999, the *CypberCop sting* was released by Network Associates which can simulate a network containing different types of network devices [2]. *NetFacade* was released in the same year CypberCop sting was released, and which can be used to simulate a network of hosts and IP addresses. The first Windows honeypot, *Back Officer Friendly*, was released in 1999.

Due to the rising interest in honeypots, Lance Spitzner and a group of people decided in 1999 to form the *honeynet project* [4] which is a non-profit group dedicated to researching attackers and sharing their work with others. In 2002, a number of groups around the world interested in honeypots joined the honeynet project and formed what is known know by the *Honeynet Research Alliance*. [2].

4 Types of Honeypots

Honeypots can be classified into three categories based on the interaction level they provide to the attacker. The more a honeypot can do and the more an attacker can do to a honeypot, the greater the information that can be derived from it. However, the more an attacker can do to a honeypot, the more potential damage an attacker can do [22]. The three levels of interaction are described in detail in this section.

4.1 Low interaction

A low interaction honeypot provides, as the term describes, limited interaction between the attacker and the honeypot [11]. A low interaction honeypot's primary goal is to detect and log unauthorized connection attempts. Low interaction honeypots are the easiest type of honeypots to design, develop and deploy. This is due to the fact that they are merely programs that emulate services. A connection attempt to an emulated service on a low interaction honeypot is logged and closed after presenting some banner. Although a low interaction honeypot has a low risk level, the information it collects is very limited. Low interaction honeypots are not able to log more than:

- The date and time of the connection.
- The destination port number, source IP address, and source port number.

4.2 Medium interaction

A medium interaction honeypot offers the attacker more ability to interact than a low interaction honeypot but less functionality than a high interaction honeypot [20]. When the attacker attempts to connect to a specific service on a medium interaction honeypot, the honeypot may respond to commands sent by the attacker with some bogus information. This is different from low interaction honeypots where only the banner is sent back to the attacker and the connection is closed afterwards. On a medium interaction honeypot, the attacker can only use emulated services as in the low interaction honeypot. However, the use of UNIX functions such as *jail* and *chroot* which allow the system administrator to create some virtual operating system inside a real one can be used [11]. Although the attacker connects to an environment that behaves like a real operating system, ev-

Table 1: Tradeoffs of honeypot level of interaction [22]

Interaction	Installation/configuration	Deployment/Maintenance	Information gathering	Risk
Low	Easy	Easy	Limited	Low
Medium	Involved	Involved	Variable	Medium
High	Difficult	Difficult	Extensive	High

erything is controlled and heavily monitored by the underlying operating system [11].

4.3 High interaction

High interaction honeypots are actual systems with full-blown operating systems and applications that an attacker can interact with. Attackers who break into high interaction honeypots operate on real systems [11]. High-interaction honeypots capture network traffic, gather extensive information, and can establish elements of the attacker's skill level and psychology. Although high interaction honeypots provide vast amounts of information about attackers and their techniques, they are mostly used for research purposes and are placed in controlled environments such as behind a firewall. This is due to the fact that high interaction honeypots can be used by an attacker to attack or compromise other systems on the same network or on other networks.

5 Purpose of honeypots

Honeypots can be divided into two categories based on their purpose. These two categories are described below.

5.1 Production honeypots

Production honeypots are systems that help mitigate risk in a network environment. Production honeypots are mostly low interaction honeypots and sometimes medium interaction honeypots that help slow down attacks. This is done by deceiving the attacker into interacting with the honeypot and distracting him from attacking valuable computer systems on the network. While the attacker wastes time interacting with the honeypots, the honeypot administrators can examine the attacker's techniques and harden the rest of the systems on the network [22].

5.2 Research honeypots

Research honeypots help security researchers learn about the techniques used by attackers to attack systems and networks. Research honeypots are high interaction honeypots that capture extensive information. They are different from production honeypots as they are not necessarily deployed to mitigate risk in a network environment. Their primary purpose is to capture extensive information that can be analyzed and used in devising effective countermeasures in the future.

6 Value of Honeypots

Honeypots do not provide a solution to a specific problem in security. They are tools that can help improve the overall security architecture. The value of honeypots and the problems they solve depend on how they are built, deployed, and used [22]. In this section, we describe the advantages and disadvantages of honeypots that affect their value.

6.1 Advantages

6.1.1 Data value

Millions of packets are sent from and to any organization's network. Although organizations can monitor and log large amount of traffic every day using firewalls and Intrusion Detection Systems, such traffic becomes extremely difficult to analyze. This is due to the fact that not every logged packet is suspicious. Hence, deriving any value from the captured traffic can be overwhelming. Honeypots, on the other hand, collect very little data, but what they do collect is normally of very high value. This is because honeypots are isolated systems that must not see any legitimate traffic. All traffic captured by a honeypot is considered suspicious.

6.1.2 Minimal Resources

One of the challenges most security mechanisms face these days is resource limitations, or even resource exhaustion [23]. Resource exhaustion is when a security resource can no longer continue to function because its resources are overwhelmed [22]. Firewalls and Intrusion Detection Systems, for instance, may fail any time due to the large amount of traffic they have to capture and process. Honeypots, on the other hand, typically do not have problems of resource exhaustion [22] because they capture and process little activity.

6.1.3 Simplicity

Honeypots do not require developing complex algorithms or setting up large signature databases to operate. All what you have to do is set up a honeypot somewhere in an organization's network, and wait for suspicious traffic. Although research honeypots are more complex than production honeypots, they all operate on the same premise: If someone connects to the honeypot, check it out [22]. The simplicity of the honeypot concept is the primary reason for its reliability [11].

6.1.4 Encryption

It does not matter if an attack or a malicious activity is encrypted, the honeypot will capture the activity [23]. Since encrypted attacks (e.g., SSH burteforcing) interact with the honeypot as an end point, such malicious activities are decrypted by the honeypot.

6.1.5 Reducing false positives

One of the challenges with most traditional detection systems is the generation of false positives. For example, an Intrusion Detection System may be triggered to fire an alert after processing innocent traffic that looks somewhat similar to a signature stored in the database. Honeypots dramatically reduce false positives since all activity with honeypots is by definition unauthorized, making it extremely efficient at detecting attacks [23].

6.1.6 Catching false negatives

Traditional detection systems fail to detect unknown attacks such as zero-day exploits because they rely upon known signatures or upon statistical detection. Honeypots, on the other hand, can capture new attacks since any activity with them is an anomaly, making new or unseen attacks easily stand out [23]. Catching false negatives is a critical difference between honeypots and traditional computer security technologies.

6.1.7 Insider threats

An organization cannot be attacked only by an outsider but by an insider as well. Honeypots can be used effectively to trap and catch insider threats. Any connection from computer systems inside an organization's network to a honeypot is very suspicious and might be an evidence of a regular user who exceeds his privileges [11].

6.2 Disadvantages

6.2.1 Narrow field of view

The greatest disadvantage of honeypots is their limited field of view. Honeypots only see activities mounted against them. If an attacker breaks into an organization network, evades the honeypot, and attacks a variety of production systems then the honeypot will be unaware of the activity. As mentioned earlier, honeypots have a microscope effect on the value of the data they capture and collect, enabling you to focus closely on valuable data. However, like a microscope, the honeypot's very limited field of view can exclude activities happening all around it [22].

6.2.2 Fingerprinting

Honeypot fingerprinting is when an attacker can identify the true identity of a honeypot because it has certain expected characteristics or behaviours [11]. For example, if a honeypot is implemented to emulate SMTP then the attacker must be able to send commands to it and get back responses as defined in the RFC documents. If a honeypot is implemented incorrectly and responds to a command sent by the attacker incorrectly (e.g., sends the attacker an "okay" message instead of "OK") then the attacker may figure out that he is interacting with a honeypot. Once an attacker identifies the true identify of a honeypot then he can do the following:

- Spoof the identity of other production systems on the same network and attack the honeypot. The honeypot would detect these spoofed attacks, and falsely alert the honeypot's administrators that a production system was attacking it, sending the organization on a wild goose chase [22].
- Post the IP address of the honeypot on the Internet so other attackers can take caution. A list of IP addresses of well known honeypots set up by government agencies such as the FBI, CIA, NSA, etc which have been identified can be found on the Internet.
- Feed bogus information to the honeypot as opposed to avoiding detection. This bogus information would then lead the security community to make incorrect conclusions about attackers [22] and their techniques.

6.2.3 Risk

The use of honeypots introduces risk. By risk, we mean that a honeypot, once attacked, can be used to attack, infiltrate, or harm other computer systems or networks [23]. As mentioned earlier, the more an attacker can do to the honeypot, the more potential damage an attacker can do. Recently, the concept of honeywalls has been introduced to reduce the risk involved deploying high interaction honeypots. A honeywall is a system that sits between a honeypot and an external network. It is a system that works like a firewall but only incoming traffic is allowed to pass through. If the attacker tries to launch an attack from the honeypot to another system then the honeywall blocks it.

7 Honeynets

The concept of a honeypot was further developed into the idea of a honeynet. Levine [14] defines a honeynet as a network placed behind a reverse firewall that captures all inbound and outbound traffic. Honeynets are more complicated arrangement of a honeypot, using one or more honeypots within an entire network that is set up for the sole purpose of monitoring an attacker's activities [12]. This network is then protected by a honeywall, which as described earlier, protects the outside world from attacks originating from within the

honeynet or honeypot. Honeynets are complex in that they are entire networks of computers to be attacked and nothing in the network is emulated [23].

The honeypots used within honeynets are high interaction honeypots that capture extensive information on threats, both internal and external. Honeynets are flexible because they are not a standardized solution. You can add any operating systems or run any services. For example, you can set up a honeynet that has a Solaris system, a Linux system running a MySQL database server, and a Windows system running MS SQL. The Honeynet project [4] is an example of a honeynet that contains many computers running different operating systems and services constructed using User Mode Linux (UML) or VMware.

8 Honeytokens

Honeytokens represent one of the most interesting implementations of a honeypot. The term honeytoken was first coined by Augusto Paes de Barros in 2003 on the honeypots mailing list. A honeytoken is like a honeypot, you set it up somewhere and no one should interact with it. Any interaction with a honeytoken most likely represents unauthorized or malicious activity [21]. Honeytokens are not systems; instead they are digital entities. For example, a word document, database record, a UNIX password file, etc. To use a honeytoken, all what you have to do is decide what your honeytoken is, set it up, and monitor it. If someone accesses it then they most likely have violated the system's usage policy [21]. Due to their simplicity, honeytokens can be very effective in detecting unauthorized access by outsiders by insiders or outsiders.

9 Honeyclients

Honeyclients represent one of the newest implementations derived from the idea of honeypots. In traditional honeypots, you set up a honeypot and wait for it to be probed, attacked, or compromised. A honeyclient, on the other hand, actively crawls the Web seeking Web sites that try to exploit it. Honeyclients mimic, either manually or automatically, the normal series of steps a regular user would make when visiting various Web sites. Although Microsoft was far from being the first to explore the idea of honeyclients, its Strider HoneyMonkey project [26] was one of the first honeyclient implementations to get widespread attention due to its success.

Microsoft's *Strider HoneyMonkey Exploit Detection System* consists of a pipeline of monkey programs running possibly vulnerable browsers on virtual machines with different patch levels and patrolling the Web to seek out and classify Web sites that exploit browser vulnerabilities [26]. Within the first month of utilizing Strider HoneyMonkeys, 752 unique URLs hosted on 288 Web sites attempted to exploit unpatched Windows XP machines when the monkeys crawled the URLs. One out of the 288 Web sites was operating behind 25 exploit-URLs and was performing zero-day exploits of the javaprxy.dll vulnerability.

10 Application to Spam

In previous sections, we presented a survey on honeypots. Honeypots are a powerful technology that can be used to detect known or unknown attacks and track attackers back to their source. In this section, we describe how honeypots can be used to fight spam and spammers. Spam is defined as unsolicited email sent by a third party. In today's highly technical world and our computer-connected society, spam has become a serious problem that affects every Internet user. Spam has also become a security concern as it can be used to deliver malware, spyware, phishing attempts, and cause denial of service attacks [25]. According to Symantec [25], between January 1st 2006 and June 30th 2006, 54% of email traffic was classified as spam. Spam consumes computer and network resources, and wastes human time and money. Billions of dollars are spent every year to counter spam. This includes lost in productivity and the additional equipment, software, and manpower needed to combat the problem.

A number of anti-spam techniques have been proposed, developed, and deployed to counter spam from different perspectives. One of the techniques to counter spam is using honeypots. Open mail relays and open proxies such as off-the-shelf SOCKS and HTTP proxies play an important role in the spam epidemic [27]. Spammers continually scan the Internet for open mail relays and open proxies to abuse them. By using open mail relays and open proxies, spammers can obscure their originating IP address and remain anonymous [13]. Lets not also forget about the role of botnets in the spam epidemic. Spammers use an army of zombies to send spam, obscure their originating IP address, and sometimes act as reverse proxies for the spammer's Website to hide the IP location of the spammer's dedicated servers [5].

Security professionals and researchers started designing and deploying open mail relay, open proxy, and zombie honeypots to counter spam, and collect valuable information about spammers and spamming techniques. In this section, we present an in-depth discussion of spammers activities, and based on these activities we describe how honeypots can be used effectively to

counter spam and track spammers. We also describe the latest techniques used by spammers to detect spam honeypots.

10.1 Spammer activities

10.1.1 Email addresses

To send large volumes of spam, spammers need large lists of email addresses. Spammers can get email addresses using any of the following methods:

- Break into an organization's database and retrieve a list of the organization's email addresses.
- Buy a list of email addresses from another spammer or from an organization specialized in selling such lists.
- Install spyware on computer systems that can search for email addresses stored on local disk, or extract email addresses from email messages stored locally. The spyware can also be used to steal the username and password of a user's account on known Web-based email systems (e.g, hotmail, gmail, etc), and use the username and password to connect to the email server, download the user's email messages via POP or IMAP, and extract email addresses from the downloaded email messages.
- Exploit poorly configured mailing lists that give out the list of its subscribers [16].
- Crawl the Web and extract email addresses from Web pages. This method is known as *email address harvesting* and the automated software used to harvest email addresses from a Web page is called a *spambot*.

10.1.2 Operating anonymously

Spamming activities are illegal in many (but not every) countries, thus anonymity is one of the most important goals pursued by spammers [6]. Furthermore, the main objective of spammers is to send out spam to a large number of email addresses without getting blocked very easily.

Whenever an IP addresses is the source of large volumes of spam, that IP address is added to a blacklist and many Internet Service Providers (ISPs) and email systems block any further email messages sent from it. As a result, spammers are highly motivated to send spam that is difficult to trace back to a particular IP address. Spammers can send spam and remain anonymous using the following methods:

Open mail relays. Email messages are hardly ever sent directly from the sender's email server to the recipient's email server [8]. Instead, email messages pass through a number of gateways called *mail relays* [8]. Open mail relays are Mail Transfer Agents (MTAs) that allow unauthenticated Internet systems to connect and forward email messages through them. Originally intended for user convenience (e.g., to let users send mail from a particular relay while they are travelling or otherwise in a different network), open mail relays have been exploited by spammers due to the anonymity and amplification offered by the extra level of indirection [19].

Whenever an email message passes through an open mail relay, the relay inserts a *Received* header at the front of the message that shows the IP address of the computer that connected to the open mail relay and relayed an email message through it. By the time an email message reaches its recipient, it contains a number of *Received* headers: one for every open mail relay through which the email message has passed [8].

When a mail relay is properly configured, it only allows certain Internet systems that successfully authenticate to it to connect and relay email messages through it. However, when it is poorly configured, which is the case in many mail relays these days, any Internet system can connect and relay email messages through it. When the spam travels from the spammer to the open mail relay and then to the recipient, the spam appears to come from the open mail relay, not the spammer.

Open mail relays do not conceal the spammer's identity as well as open proxies or botnets since the IP address of the spammer's computer system appears in one of the *Received* headers in the email. Nevertheless, most bulk email tools such as Send-Safe [1] add fake *Received* headers to email so that the recipient cannot tell which of the *Received* headers in the email message contains the IP address of the spammer's computer system [8].

Using open mail relays becomes effective when the spam is routed first through an open proxy and then through an open mail relay.

Open proxies. A proxy server is a computer system that helps two computer systems communicate with one another by forwarding traffic back and forth between the two systems. An open proxy is a proxy that allows an unauthorized Internet systems to connect through it to other systems on the Internet. Similar to open mail relays, spammers abuse open proxies due to the anonymity offered by the extra level of indirection. When a spammer sends spam through an open proxy, the spam is forwarded from the proxy to the spam

recipient. From the email recipient's point of view, the spam is coming from the proxy, not the spammer's system [8].

To remain untraceable and have a very high level of anonymity, spammers use a chain of open proxies located in different countries. The longer the chain, the stealthier spammers become [16]. Different countries have different spam laws and some countries do not even have any laws against spam. This makes tracking the spammer down difficult if not impossible.

Botnets. The majority of spam sent these days is sent via botnets. Botnets are collections of compromised systems known as *zombies* infected with a software called a *bot* that communicates under one centralized controller known as the *bot controller* or the *command and control (C&C) server*. Botnets are a very real and quickly evolving problem that is still not well understood and studied [10]. Installing bots can be done using a variety of ways (e.g., viruses, worms, spyware, exploitation techniques, social engineering, etc). For example, the W32/Bobax worm exploited the DCOM and LSASS vulnerabilities on Windows systems, and allowed infected systems to be used as an open mail relay [19].

Once a bot is installed on a victim's computer system, the bot can receive commands from the bot controller to send spam. Illegal spam sent by zombies has increased dramatically in recent years. In addition, computer criminals use zombie computers to launch phishing attacks that try to steal personal information, such as Social Security and credit-card numbers [15], launch Distributed Denial of Service attack (DDoS), etc. Although the originators of botnets, known as *bot herders*, are not necessary the spammers, bot herders can be paid by spammers to send spam via their botnets.

To send spam via a botnet, a spammer instructs the bots under his control to send spam to email addresses on his list. Even a relatively small network of 10,000 zombies can generate spam at an incredible aggregate rate [8]. To the recipients, the spam messages sent by the zombies in a botnet appear to come form legitimate home or corporate users [8].

10.2 Spam Honeypots

In the previous section, we described various activities performed by spammers to send spam anonymously. Based on such activities, one can design and deploy honeypots that can lure spammers and attempt to expose their identities, and capture the spam they send. For example, a honeypot can be used to trap email harvesters, act as an open mail relay or proxy, or turned into a zombie that can join a botnet. In this section, we describe how such honeypots can be used to provide better countermeasures against spam.

10.2.1 Harvesting

Spambots crawl the Web very often to build lists of email addresses. One way to trap spambots is by creating links in Web pages that are invisible for a human reader but visible for a spambot. The links can point to Web pages that automatically generate hundreds or thousands of fake email addresses to trap the spambot into an endless loop. Another technique would be to point to Web pages that feed the spambot monitored email addresses (honeytokens). If the spammer tries to send spam to any of the monitored email addresses then the IP address of the computer system used by the spammer to send spam can be logged and used to track him down. Furthermore, since we know that all the email messages sent to any of the monitored email addresses are spam messages, one can use such information in filtering similar email messages with a spam filter. For example, Microsoft maintains more than 130,000 *MSN Hotmail* trap email addresses (email harvester honeypots) to investigate patterns within spam [15] and build better spam filters.

Another example of an email harvester honeypot is *Project Honeypot* [3] created by Unspam Technologies Inc. The Project Honeypot system is a distributed system designed to identify spammers and the spambots they use. The system installs email addresses that are custom-tagged to the time and IP address of a visitor to any Web page. If one of these addresses begins receiving email messages then such messages must be spam. Thus, the exact moment when the email address was harvested and the IP address of the spambot can be identified. Project Honeypot's Web site provides statistics about spambots. For example, the time from harvest to first spam, harvester traffic, spam messages sent, active harvesters, top-10 countries for harvesting, etc.

Although the above techniques might trap naive spammers and spambots, it is not the case with skilled spammers. Skilled spammers use sophisticated spambots and open proxies to crawl the net. Thus the monitored email addresses will just help with finding the IP addresses of the open proxies and the spammer will keep his anonymity [16].

10.2.2 Open proxies and open mail relays

As mentioned earlier, spammers rely heavily on open proxies and open mail relays to remain untraceable. Set-

ting up open proxies or open mail relays as honeypots can be very effective in capturing spam. An open mail relay honeypot can be used to emulate SMTP on port 25 and an open proxy honeypot can be used to emulate SOCKS4 or SOCKS5 on port 1080.

Low interaction open proxy or open mail relay honeypots might not be able to log more than the IP address of the computer system that attempts to forward traffic via the proxy or using the mail relay. However, high interaction open proxy or open mail relay honeypots can be used to capture extensive information. For example, if a spammer discovers that a system (the high interaction open proxy honeypot) is running SOCKS4 then he will try to reach an open mail relay or a usual MTA by bouncing through the open proxy [17]. The high interaction honeypot can not only log the IP address of the system connecting to the honeypot but can capture all the spam sent by the spammer. Interesting information can be extracted from the spam headers and body, and submitted to a blacklist or used by a spam filter.

Honeyd [18] is a honeypot that can be used emulate open mail relays and open proxies. Honeyd is a framework for virtual honeypots that simulates virtual computer systems at the network level and which runs on unallocated network addresses. When a spammer attempts to send spam via an open proxy or an open mail relay emulated with honeyd, honeyd redirects the spam to a spam trap. The spam trap then submits the collected spam to a collaborative spam filter [18]. Honeyd has support for passive fingerprinting to identify the operating system that opens a connection to the honeypot. According to [18], most machines that submit spam are running or compromising either Linux or Solaris.

Recently, spammers started to develop and use strategies to counter open mail relays and open proxy honeypots. A popular spamming software called *Send-Safe* [1] sends a test email message using an open mail relay or through an open proxy before using it. If the test email message is not delivered then Send-Safe will not use the mail relay or proxy. Although open mail relay and open proxy honeypots are not supposed to deliver any spam, some of these honeypots deliver only the first email message to make the honeypots look realistic and fool the spamming software.

10.2.3 Zombies

In 2005, Microsoft took a novel approach [15] in fighting spam and spammers based on the idea of honeypots. A team at Microsoft infected a Windows system with a bot (turned it into a zombie). The zombie system was quarantined to prevent it from sending any spam onto the public Internet if instructed to do so. In less than three weeks, the Microsoft lab's zombie computer received more than 5 million requests to send 18 million spam emails [15]. According to Microsoft, these requests contained advertisements for more than 13,000 unique Web sites. After the exercise, Microsoft analyzed the traffic sent to the zombie system and the spam it was meant to send out. It compared those with other spam messages captured in Hotmail accounts. This allowed Microsoft to uncover the IP addresses of the computer systems that were sending spamming requests to the quarantined zombie, along with the addresses of the Web sites advertised in the spam [15]. The evidence gathered contributed to a lawsuit in which Microsoft has identified 13 different spamming operations.

The approach used by Microsoft seems interesting since spammers usually control thousands of bots so it is almost impossible for them to figure out that one of their bots is a honeypot. To counter this issue, sophisticated spammers started using twisted ways to evade honeypot detection. Usually spammers post instructions to bots through a command and control (C&C) server. To counter the risk of that server being detected, spammers post new instructions to bots by using a path through multiple computer systems, often including computer systems located outside the United States [8]. In such instances, the information obtained from the zombie honeypot is of little use in identifying the spammer's true IP address [8].

Another technique used by spammers to evade zombie honeypots is by designing botnets in a form of a peer-to-peer network so the C&C server with which individual bots communicate is not fixed. For example, bots can receive instructions from other peers instead of receiving instructions directly from a C&C server. In this case, if a zombie honeypot joins such botnet then it will only communicate with a few other bots. Thus, its view of the botnet is local and limited, and it would not have access to the IP address of the C&C server [8].

11 Conclusion

Honeypots are a powerful and interesting technology with extensive potential. They help improve the overall security architecture by providing early warning about new attacks and attacking techniques, distracting attackers from more valuable systems, and allowing us to monitor attackers as they exploit systems. Honeypots capture data of high value, reduce false positives, and catch false negatives. They are simple and require minimal resources to set up.

In this paper, we presented a survey on honeypots. We defined honeypots and discussed their history. We described the different types of honeypots based on their interaction level with the attacker and based on their purpose. We described the advantages and disadvantages of honeypots that affect their value. The different implementations of honeypots and terminologies used such as honeytokens, honeyclients, and honeynets have been discussed.

We also presented an in-depth discussion of the activities performed by spammers to send large volumes of spam anonymously, and discussed how honeypots can be used to lure spammers, capture their spam messages, and attempt to track them down.

References

[1] Bulk Email Software, http://www.send-safe.com.

[2] Honeypots 101: A Brief History of Honeypots http://www.philippinehoneynet.org/docs/Honeypot101_history.pdf.

[3] Project Honeypot, http://www.projecthoneypot.org/.

[4] The Honeynet Project, http://www.honeynet.org.

[5] The Spamhaus Project, http://www.spamhaus.org.

[6] M. Andreolini, A. Bulgarelli, M. Colajanni, and F. Mazzoni. HoneySpam: Honeypots Fighting Spam at the Source. In *Proceedings of USENIX SRUTI*, pages 77 – 83, 2005.

[7] S. Bellovin. There Be Dragons. In *Proceedings of the Third USENIX Security Symposium*, pages 1 – 16, 1992.

[8] D. Boneh. The Difficulties of Tracing Spam Email, FTC Expert Report, http://www.ftc.gov/reports/rewardsys/expertrpt_boneh.pdf. 2004.

[9] B. Cheswick. An Evening with Berferd in which a cracker is Lured, Endured, and Studied. In *Proceedings of USENIX*, 1990.

[10] E. Cooke, F. Jahanian, and D. McPherson. The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets. In *USENIX SRUTI Workship*, 2005.

[11] D. Joho. Active Honeypots, M.Sc. Thesis, Department of Information Technology, University of Zurich, Switzerland, http://www.ifi.unizh.ch/archive/mastertheses/DA_Arbeiten_2004/Joho_Dieter.pdf. 2004.

[12] J. Jones and G. Romney. Honeynets: An Educational Resource for IT Security. In *Proceedings of the 5th conference on Information technology education*, pages 24 – 28, 2004.

[13] N. Krawetz. Anti-Honeypot Technology. In *Proceedings of IEEE Security and Privacy*, volume 2, pages 76 – 79, 2004.

[14] J. Levine, J. Grizzard, and H. Owen. The Use of Honeynets to Detect Exploited Systems Across Large Enterprise Networks. In *Proceedings of the 2003 IEEE Workshop on Information Assurance*, pages 92 – 99, 2003.

[15] Microsoft. Stopping Zombies Before They Attack, http://www.microsoft.com/presspass/features/2005/oct05/10-27Zombie.mspx.

[16] L. Outdot. Fighting Spammers With Honeypots: Part 1, http://www.securityfocus.com/infocus/1747.

[17] L. Outdot. Fighting Spammers With Honeypots: Part 2, http://www.securityfocus.com/infocus/1748.

[18] N. Provos. A Virtual Honeypot Framework. In *Proceedings of the 13th USENIX Security Symposium*, 2004.

[19] A. Ramachandran and N. Feamster. Understanding the Network-level Behavior of Spammers. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 291 – 320, 2006.

[20] K. Sadasivam, B. Samudrala, and T. Yang. Design of Network Security Projects using Honeypots. In *Journal of Computing Sciences in Colleges*, volume 20, pages 282 – 293, 2005.

[21] L. Spitzner. Honeytokens: The Other Honeypot http://www.securityfocus.com/infocus/1713.

[22] L. Spitzner. *Honeypots: Tracking Hackers*. Pearson Education Inc, 2002.

[23] L. Spitzner. Honeypots: Catching the Insider Threat. In *Proceedings of the 19th Annual Computer Security Applications Conference*, 2003.

[24] C. Stoll. Stalking the Wily Hacker. In *Communications of the ACM*, volume 31, pages 484 – 497, 1988.

[25] Symantec. Symantec Internet Security Threat Report, Trends for January 06 - June 06,

http://www.symantec.com/specprog/threatreport/ent-whitepaper_symantec_internet_security_threat_report_x_09_2006.en-us.pdf. 2006.

[26] Y. Wang, D. Beck, X. Jiang, and R. Roussev. Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites That Exploit Browser Vulnerabilities. In *Proceedings of the 14th USENIX Security Symposium*, 2005.

[27] M. Xie, H. Yin, and H. Wang. An Effective Defense Against Email Spam Laundering. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 179 – 190, 2006.

[28] J. Zdziarski. *Ending Spam*. No Starch Press, 2005.