Honeypots
Security on Offense

By

Kareem Sumner

For

Security Architecture
774.716

Instructor
Arthur Friedman

July 10, 2002

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This report presents a very interesting network security model called honeypots. Honeypots are systems or devices that act as bait to divert potential intruders while recording and logging their activities. These systems capture and analyze intruders as they are compromising the honeypot system. This paper discusses honeypots in a general sense of their contribution to network security. It lists the advantages and disadvantages, the many different types and deployment methods, the risk and legal issues involved and the potential future of honeypot systems.

INTRODUCTION

With the increased connectivity of computer systems, the emergence of the Internet and the heightened sense of security, there is no doubt that, the need for security countermeasures is vital for protecting organization's systems and information. For quite some time, network security has been strictly defensive using traditional network devices such as routers, firewalls and Intrusion Detection Systems. Honeypots have taken a different stance on security – a proactive one. Honeypots are systems configured to lure intruders, analyze, and record their every move. This will allow IT staff within organizations to learn the methods of the intruder (or blackhat) and be able to 'harden' their production systems against similar attacks. Honeypots alone cannot solve system security issues; they are just tools that supplement the traditional network devices. The Honeynet Project, a non-profit organization of thirty-security professionals lead by Lance Spitzner, is dedicated to learning the tools, methods and intent of intruders. The group, founded in April of 1999, shares its information with the security community.

WHAT IS SECURITY?

Defining security (System/Network Security) alone can be a project within itself. In a broad sense of the word, security reduces the level of risk. Risk can never be eliminated, but security measures can reduce the risk and vulnerabilities of an organization's assets. In the context of system security, it is the systems ability to protect information and system resources pertaining to confidentiality and integrity. Confidentiality, Integrity and Availability, or CIA, are core concepts synonymous with system security. Confidentiality ensures that only authorized user's access certain information. Integrity ensures that information is consistent, correct, and not tampered with by unauthorized users. Availability ensures that assets are accessible to authorized users. Security is not limited to CIA, but has a functional aspect also.

Prevention

- Preventing intruders from attacking and compromising an organization's resources. This is usually a perimeter defense (first layer defense) keeping intruders out by any means.

Detection

- If prevention fails and intruders manage to penetrate the perimeter, the next crucial step is detecting the intruder quickly.

Reaction (Recovery)

- Once the intruder is detected, there must be a quick and effective response to the breach in security. Policies should be in place to handle such a situation (i.e. backup servers, restore data from backup media, or disaster recovery option).

There are other ideas that are a part of systems security, access control, non-repudiation, authentication, risk avoidance, deterrence etc., but for this paper, the above definition is sufficient.

TRADITIONAL NETWORK SECURITY DEVICES

The use of at least one traditional security device exists today on almost every network. The devices can range from a router, firewall to an IDS. Even on personal home networks many people use software versions of these devices to protect their systems or personal network.

Routers

Reads and filters all data packets passing through it. It determines the packet destination within its own network or passes it further along the Internet.

Sniffers

A sniffer is a program that monitors and analyzes network traffic, locating bottlenecks and other network problems. They are often used on academic networks to prevent bottlenecks caused by file sharing applications such as Napster or MIRC etc.

Firewalls

A Firewall filters all traffic between a protected (internal) network and an unprotected (external) network. The purpose of a firewall is to keep unwanted packets from entering the protected network. Firewalls have policies that can prevent access from the outside, but allow traffic to flow from inside to outside. It can also be more selective with the traffic that it allows i.e. certain places, people, protocols etc. Many firewalls do different things.

Intrusion Detection Systems (IDS)

An IDS checks all network activity leaving or entering the network, and identifies suspicious patterns that can signal a possible intrusion. IDSes can detect misuse by comparing the information gathered to a large signature database. The database however, must always be current and updated. They can also detect anomalies by allowing system administrators to define 'normal' on the network. Any deviances will trigger an alert. There are two types of IDSes, Host-based and Network-based. The host-based runs on each individual system examining it for potential attacks. They are resource 'hogs' and must be installed on all machines present. A network-based IDS checks the entire network for attacks, and requires a dedicated host for each subnet.

All of these traditional security devices do a good job at securing networks. The firewall and router attempt to block hostile activity and IDSes detect attacks as they happen, but they also have limitations. An intruder, with some patience, can bypass many of these devices. Not much can be done once an intruder gets pass a firewall, and an IDS is only able to capture information once the attack is already in progress. There is insufficient time to protect vulnerable systems once the attack has started.

In order to counter-attack these security breaches, one must be able to delay the attack or deceive and contain the intruder. In doing so, one can gather as much information necessary to countermeasure the attack from causing serious damage. Such a 'deception' system can be used to supplement other security devices such as firewalls and IDSes - a deception system known as a honeypot.

WHAT ARE HONEYPOTS?

A honeypot can be defined as any device "designed to attract intruders so that their activities can be monitored without risk to production systems or data" (E. Eugene Shultz, September 22, 2000). The honeypot entices blackhat attackers and examine them as they exploit vulnerabilities within the 'decoy' system. Honeypots do not actually replace any of the other previously mentioned, traditional security devices. They are similar to a standard IDS but with more focus on deception and information gathering. There are a couple of ways to implement honeypot systems. Two of them are mentioned below:

- Installing a system with an old, unpatched version of an operating system (i.e. NT 4.0 or Linux Red Hat 5.0). This allows all of the vulnerabilities to be exploited by the intruder. Add a standard IDS or/and sniffer to log hack attempts, and track the attackers movements once the system is compromised.

- Install special honeypot software for tracking attackers (discussed further in the section, Honeypot Software).

Goals and Objectives

There are two main objectives for using honeypots, research and production.

Research

Research honeypot learn how blackhats attack, penetrate and gain access to a system. Information of all activities is logged and the attack methods used by the intruder. One can use that knowledge to protect the production system. This is primarily to collect information on the blackhat community. These honeypots do not add direct value to a specific organization, but act as 'counter-intelligence'. They try to find out who is the threat, why and how they attack, the tools intruders use, and when they will likely attack again.

Research honeypots provide a platform to learn about intruders as they compromise a system. What makes it even more valuable is learning what occurs after compromising the system. Some intruders communicate with other blackhats or upload tool kits to capture automated attacks such as worms that actually target entire networks.

Generally, research honeypots do not reduce the risk within an organization, but improves prevention, detection or reaction.

Production

Production honeypots collect forensic evidence that can lead to the capture or prosecution of intruders. Its purpose is to help mitigate risk in an organization. If the honeypot is compromised, data is collected and the system is taken offline for a full forensic analysis. The information is then given to law enforcement for prosecution. Based on the information one can only not learn the methods used to attack the system, but what was done while in the system.

Other possible reasons to implement a honeypot can be to setup a training ground for whitehats (persons working for an organization whose purpose is to find vulnerabilities within the honeypot to better protect the production system), and to recruit IT professionals for an organization. A honeypot can be setup with known vulnerabilities and prospective employees must attempt to compromise the system.

The goals of setting up a successful honeypot system are:

- The honeypot must appear as generic as possible without any alterations to the operating system.

- It must be configured in a manner as to not allow intruders to compromise production systems within the network or outside the network.

- Honeypots should contain real and interesting information to attract intruders long enough to track their moves. If not, the blackhat may either become suspicious or avoid the honeypot altogether.

- Honeypots can either be placed in the front of a firewall, in the DMZ (De-militarization Zone), or behind a firewall **(See figure 3 in Appendix, pg. 17)**. In general, however, they are usually setup behind the firewall to appear as a legitimate network to the intruder.

ADVANTAGES AND DISADVANTAGES OF HONEYPOTS

Advantages

- Honeypots can act as deterrence to intruder attacks. Knowing that a system is set up to capture and log all activities may scare away would be intruders.

- It can produce forensic evidence that is admissible in a court of law. Many people think a honeypot is entrapment. As long as it is deployed correctly and is not advertised, it can be used as legal evidence.

- Honeypots usually only accept hostile activity. These systems are normally not accessed so any packets sent to the system are deemed an intrusion. This cuts down the amount of false positives and false negatives associated with IDSes. There are a few exceptions however, whereby persons may incorrectly type a wrong IP address or DNS entry and stumble across a honeypot.

- IT staff can learn about incident response to attacks. System administrators will learn the intruder's methods and will be able to use counter-measures to harden their production system.

- The honeypot can present itself as a banner system accepting 'banner checks' sent by intruders. Some versions of software packages have 'buffer overflow' and allow intruders to take advantage of system vulnerabilities (i.e. POP3 email services on port 110).

- Honeypots divert intruders from the production system making them use all of their efforts in a harmless manner.

- Honeypots can detect inside attacks. Many security concerns stem from employee's within the organization, misusing the system.

Disadvantages

- Intruders can use honeypots to compromise other systems on an organization's network and on the Internet. This can lead to legal implications for the organizations that own the honeypot system.

- Honeypots can add complexity to a network. Depending on how it is deployed, it may increase security complexity and increase exposure to exploits.

- Honeypots are maintained like production systems. This adds to the administrative overhead within the IT department.

- Advertising honeypots of its existence may not deter intruders but actually entice them to try harder to compromise a system. This type of implementation may seem as luring intruders into entrapment. If this is the case, any evidence collected by the honeypot is inadmissible in a court of law.

TYPES OF HONEYPOTS

There are several different types of honeypots, ranging from very simple and low-interaction systems to more complex, medium-high interaction systems. Many honeypots are configured like any one of the following:

Port Monitors

This is the simplest form of honeypot. It listens for traffic on ports usually scanned by blackhats. Its drawback however, alerts the intruder by accepting the connection and then dropping it. This would make the intruder suspicious and abort the attack.

Deception Systems

Deception systems actually do more than listen for traffic on ports, it responds to the intruder like a production server.

Multi-protocol Deception Systems

This deception system is capable of having multi-protocols and banners to emulate software for different Operating Systems (i.e. Specter).

Full Systems

A full system deployed strictly for deception. Many systems of this kind have the capability to send alerts for exceptional conditions. It can include an IDS to supplement internal logging.

Honeynets

A honeynet is similar to a research honeypot. It surveys and gathers information on threats rather than detecting and deceiving intruders. Honeynets are a network of multiple honeypot systems that usually sit behind a firewall or router to control inbound and outbound data **(See figure 1, Appendix, pg. 15)**. The information captured is analyzed to learn the tools, tactics, and motives of the intruders.

Honeynets create a more realistic production network because it uses multiple Operating Systems simultaneously. This allows easy profiling of black hat trends and signatures. The actual systems within the honeynet are real systems using real applications and the systems are no less secure than production systems. The vulnerabilities that exist in a production network also exist in the honeynet. Honeynets are much more risky than a honeypot and have more administrative overhead and work involved.

One of the major problems IT staff face when trying to detect and capture an intruder is the collection of too much information. It is difficult to sort through all of the information to determine valid traffic from malicious traffic. Tools such as IDSes, as mentioned earlier, can distinguish between the two, but at a cost of information overload, false negatives and positives, unknown activity and data pollution. Like the honeypot, honeynets are designed to be compromised therefore any traffic flowing inbound or outbound considered hostile activity.

Data Control and Data Capture are two critical conditions that define each honeynet,

Data Control

This condition actually mitigates risk. Once a honeypot system is compromised within the honeynet, the activity of the intruder must be contained as to not allow any harm be done to the production systems. There must be some controls in place to manage the traffic flowing in and out of the honeynet, without the knowledge of the blackhats, preventing attacks on other production systems.

Data Capture

Secretly captures all of the activity inbound, outbound or within the honeynet without the blackhat knowing they are being watched. If the blackhat uses encryption to disguise the packets, data recording mechanisms are inserted into the kernel as kernel modifications.

If the honeynet is a part of a distributed environment then there is a third condition to consider.

Data Collection

This is an option where there are multiple honeynets connected together and the combination of all of the data is centrally located. This will allow for easy analysis and archiving.

CASE STUDIES

As one can imagine, it is difficult to find production honeypot cases especially in the corporate sector. Many commercial or financial organizations may handle honeypot information internally rather than out in public. Even though honeypots are not production systems, they still unveil potential vulnerabilities that could happen on 'live' systems. This type of information may unnerve customers who do not understand the purpose of the honeypot system. Global Integrity Corporation (an SAIC company) conducted case studies for Recourse Technologies, Incorporated. Four of the case studies (out of eight) were chosen as examples of deploying honeypots in a live environment. Each case was done under anonymity and no changes were made. These are the exact results as were printed in the study.

Case Study 1

In a corporate computing environment with approximately 5000 total computers (servers and workstations), approximately 200 custom honeypots were deployed. Each honeypot appeared to be a "normal" server and was neither specially named nor publicized. Each had substantial kernel modifications, however. The staff that monitored the honeypots discovered a startling number of honeypot hits, approximately 200 per week. The staff found that about 70 percent of the hits were accidental, but about 30 percent were deliberate. Either most of the deliberate hits appeared to be initiated by malcontents looking for critical files, applications, or individuals who wanted to exceed the authority assigned to them. So many hits occurred that the corporation had to set up a central honeypot-monitoring console. Overwhelmed with the number of honeypot hits and without a policy for dealing with unauthorized access to honeypots, the corporation's management directed that the honeypots be shut down and taken off the network.

Case Study 2

In this honeypot deployment "success story," a corporation set up a series of approximately two-dozen custom-built, internal honeypots in a corporate network. Each honeypot was assigned an interesting name (each in accordance with a particular major computing function). One honeypot appeared to be a payroll server. It had a generic, non-password protected account that provided a shell environment, menus, and numerous decoy payroll files. "Payroll lookups" were also possible on this system. One evening late at night, the Chief Operating Officer (COO) of this corporation connected to the honeypot server and tried to "jury rig" a payroll account. He brought up a payroll lookup screen, went to the edit side of the screen, and tried to edit another executive's account.

The honeypot recorded the source IP address and several other pieces of information that conclusively identified the COO. The consultant in charge of the honeypots noticed the COO's activity the next morning and reported it to the Information Security Department. Shortly afterwards the Information Security Manger notified the Head of Human Relations (HR). The HR Head and the consultant arranged for a meeting with the COO the next day. During the meeting, they confronted the COO. At first, the COO admitted only to "looking for his performance review." As the questioning continued, he became agitated and did not participate in the questioning any longer. The HR Head told him that he would have to meet with the Information Security Manager as a follow-up to the initial meeting. During the subsequent meeting with the Information Security Manager, the COO made a full admission of attempted fraud and resigned on the spot. The corporation decided against pressing for legal prosecution for fear of adverse publicity.

Case Study 4

In yet another honeypot deployment scenario, a corporation set up a single, internal honeypot server named "PeopleSoft_Data." This server, reachable only from within this corporation's internal network, was actually a Linux computer that looked like a Windows NT server to all external users. No one promoted this server---it was visible only through the Windows Network Neighborhood and browsing mechanisms. Within one week of its deployment, 30 employees found this server and tried to log on to it. Most of the connections were deemed accidental, but a few appeared to be internal attacks. The corporation would not say what actions were taken with respect to the suspected attackers. The Information Security Manager said, however, that any system that has "SAP" or "PeopleSoft" as part of its name will attract a great deal of attention and will be the target of many suspicious connections.

Case Study 8

Another corporation's information security staff was investigating the posting of confidential or what appeared to be maliciously inaccurate information on a Yahoo stock bulletin board. To identify one individual, they created a web site that offered an online greeting card and sent the Yahoo identity a message telling them to come get their greeting card at the web site. The site was equipped through question-and-answer mechanisms and through technical means to determine more about the individual's identity. The individual did not "bite" and the staff shut down the site after a three-week wait.

These particular case studies were selected because they each present a different honeypot scenario with different outcomes. The case studies showed the different ways honeypots were deployed, the different kinds of intruders that are caught, the reasons why a honeypot can fail its objective(s) and how they were successful.

HONEYPOT SOFTWARE

There are hardware-based honeypots that can be switches, servers or routers that are partially disabled to look like a production system to the intruder. Software-based honeypots imitate applications and Operating Systems and are able to run on low-end PC's. There is a variety of honeypot software available. Some of the more popular ones are mentioned below.

BackOfficer Friendly (NFR Security)

This is a very simple, 'low-interaction', yet effective honeypot. It is an entry-level program that runs on Windows-based systems and emulates a few basic services (ftp, http, mail and backorrifice). BackOfficer can also send out false replies to the intruder.

Specter

This commercial software is also a low-interaction production honeypot. It not only emulates services like BackOfficer, but also a variety of operating systems. Its unique feature is the automatic gathering of information on the intruder.

Honeyd

Honeyd is an Open Source honeypot designed to run on a Unix platform. It can emulate over 400 different operating systems and thousands of different computers, simultaneously. It can also dynamically interact with intruders and detect activity on any port.

Mantrap (Recourse)

Mantrap is a mid-high level interaction honeypot. It is one of the best honeypot software available, but it comes with high level of risk. Rather than emulating services like the other software, it can create up to four sub-systems called 'jails' ('jails' are logically distinct OSes separated from a master OS. Mantrap can be used as a research or production honeypot.

FUTURE OF HONEYPOTS/HONEYNETS

The Honeynet Project, spearheaded by Lance Spitzner, is dedicated to improving the design and deployment of honeynets. The project, now in its second phase, is in the process of making honeynets more simple, flexible, able to adapt to changing environments, and much more difficult to detect. The goal of these generation two (GenII) honeynets is to capture the activities of more sophisticated blackhats. These systems will be able to manage data control, capture and collection on the same system (a layer 2 device). As a layer two device, it will be more stealth in detection and have complete control over inbound and outbound traffic **(See figure 2, Appendix, pg. 16)**. The GenII honeynets will also have increased intelligence that will analyze the intruder's activity, instead of just counting the number of connections

The Honeynet Project is also looking into virtual honeynets. Virtual honeynets will be able to merge all of the essentials (data capture, control and collection) of a honeynet onto a single physical system. It is like taking an entire network of honeypots and implementing it on a single system. Each honeypot acts as a separate OS nothing is emulated. This will obviously cut costs and the amount of resources (systems and devices) needed to deploy a honeynet.

ANALYSIS

As mentioned earlier, honeypots are not a sole solution to shielding networks from attacks. It is a tool that supplements other network security devices. Honeypots simply divert (or deceive) intruders away from production systems and networks.

Before deploying a honeypot system, there must be established policies in place. One must know what information is important for collection and analyzing, and how to respond to attacks. A decision must be made on whether deploying a production or research honeypot.

During research, it has been mentioned that the best way to deploy a honeypot/honeynet is using a layered approach using routers, firewalls and IDSes. The first layer should use a firewall and router to control and capture the data. The firewall can log all connections going to and coming from the honeypot, and the router add more access control. This can prevent many different kinds of attacks such as DoS (Denial of Service) or SMURF attacks.

The second layer uses IDS to capture and record all network activity. It will also alert IT staff to any suspicious activity and give detailed information.

The third layer is the actual honeypot systems. They will log the information both locally and remotely. The purpose for logging remotely is to countermeasure an intruder wiping the disk clean to cover his tracks.

The case studies mentioned earlier were chosen because of their varying results. Case study 1 showed that even though the implementation of the honeypot was correct, without the proper policy in place, the results were not as effective as it could have been. Case study 2 pointed out that attacks can come from high-level corporate executives. Executives may pose a greater risk than regular staff. They may assume they have the authority to access such systems. Case study 4 configured the honeypot system on an internal network and gave it an attractive name (i.e. PeopleSoft_Data). It was visible to users via network neighborhood. This proved that proper naming of the honeypot system is critical in attracting intruders. What is a concern however, is the reason for deploying the internal honeypot. If the IT department is suspicious of employees attacking systems, then using a honeypot in this manner may be appropriate to collect evidence. However, if the deployment of a honeypot were for monitoring who would access it, may actually come close to entrapment. Case study 8 involved a Fortune 10 company. The honeypot system failed to capture the intruder, prompting that honeypots may not work well when trying to attract a particular individual.

LEGAL ISSUES SURROUNDING HONEYPOTS

When one considers the functions of a honeypot system, certain legal issues come to mind. One of the major legal issues involving honeypots is entrapment. Many people have different perspectives on this issue. Two definitions of entrapment are mentioned below.

"To lure into performing a previously or otherwise uncontemplated illegal act", (www.dictionary.com).

"A person is 'entrapped' when he is induced or persuaded by law enforcement officers or their agents to commit a crime that he had no previous intent to commit; and the law as a matter of policy forbids conviction in such a case", (www.lectlaw.com)

First, it appears that entrapment must involve law enforcement officers or their agents. IT staff within organizations or other persons not in law enforcement usually deploy honeypots.

Secondly, honeypots do not encourage criminal activity no more than a production system. The only difference is the level of vulnerability to attacks and recording the activity of the intruder. Even if law officials deployed a honeypot system, and it is not advertised, it is not considered entrapment.

Another legal issue concerning honeypots occurs after the honeypot is compromised. If the intruder is able to use the compromised system to attack other production systems either in the same network or over the Internet, the company that owns the system may be liable for any damage. There are also privacy issues concerning the recording of the intruder's activities. Personally, anyone trespassing on a system, with the intent to cause harm, the owner should have every right to protect the system.

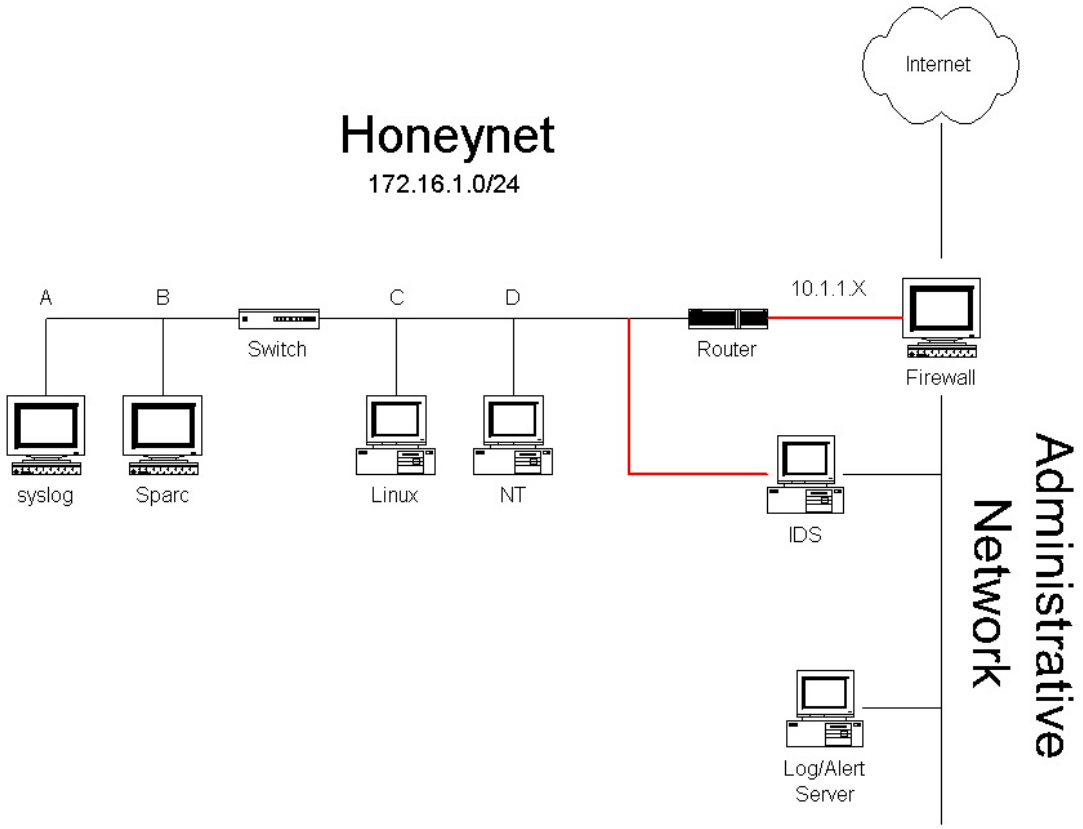
CONCLUSION

Honeypots are extremely useful as countermeasures from intruder attacks on systems. They put network security on the offense and allow IT personnel (and others) to learn the nature of the intruder. One can learn the intent, the tools that are used and the possible vulnerabilities within a production network and possibly learn of other, connected blackhats. The information collected would be substantial in 'hardening' the production systems from similar attacks and as evidence to prosecute attackers.

Deploying honeypots do not contribute directly to the prevention of attacks, simply a diversion from production systems. The more complex the security measure, the more likely there will be vulnerabilities open to exploit. The same applies to honeypots – the concept is simple, but the implementation can become real complex. It is extremely important that policies be in place prior to deploying honeypots (the same goes for any network security devices). The best security measures are those that are well written, defined, and adhered to. It is also important to know the objectives of using a honeypot and not just implement one because one can. Is it for research purposes or production purposes? The amount of risk involved stems from the type of honeypot, its deployment, and its complexity. The level of interaction can dictate the amount of risk involved. The higher the interaction and the more one can learn from it, the more risk may exist. Honeypots alone will not solve a security problem, but it will help in strengthening proper security practices.

APPENDIX

Figure 1 (Know Your Enemy: Honeynets)



In figure 1, the firewall segregates the Honeynet into three networks (Internet, Honeynet and Administrative Network). All inbound and outbound packets must go through the firewall and the router. The firewall is the primary access control point and the router supplements the firewall.

Figure 2 (Know Your Enemy: Honeynets)

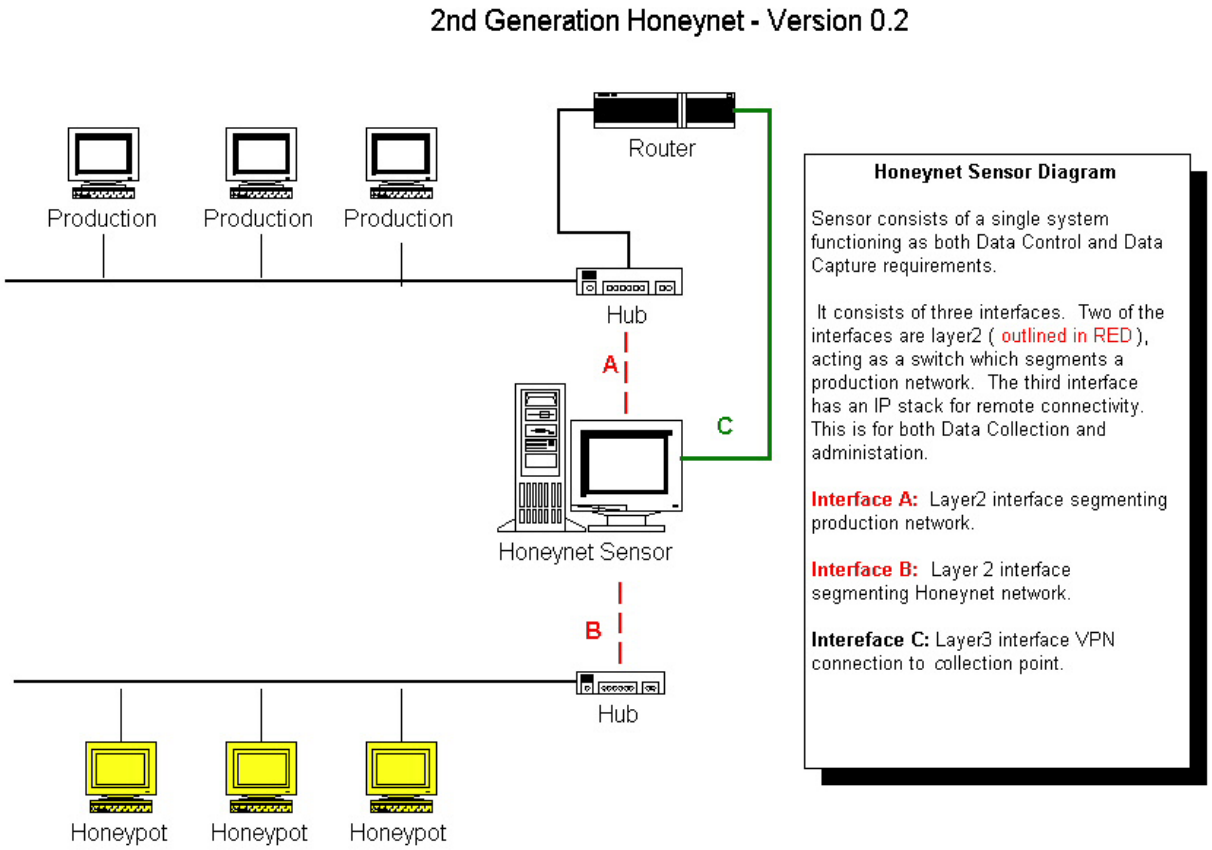
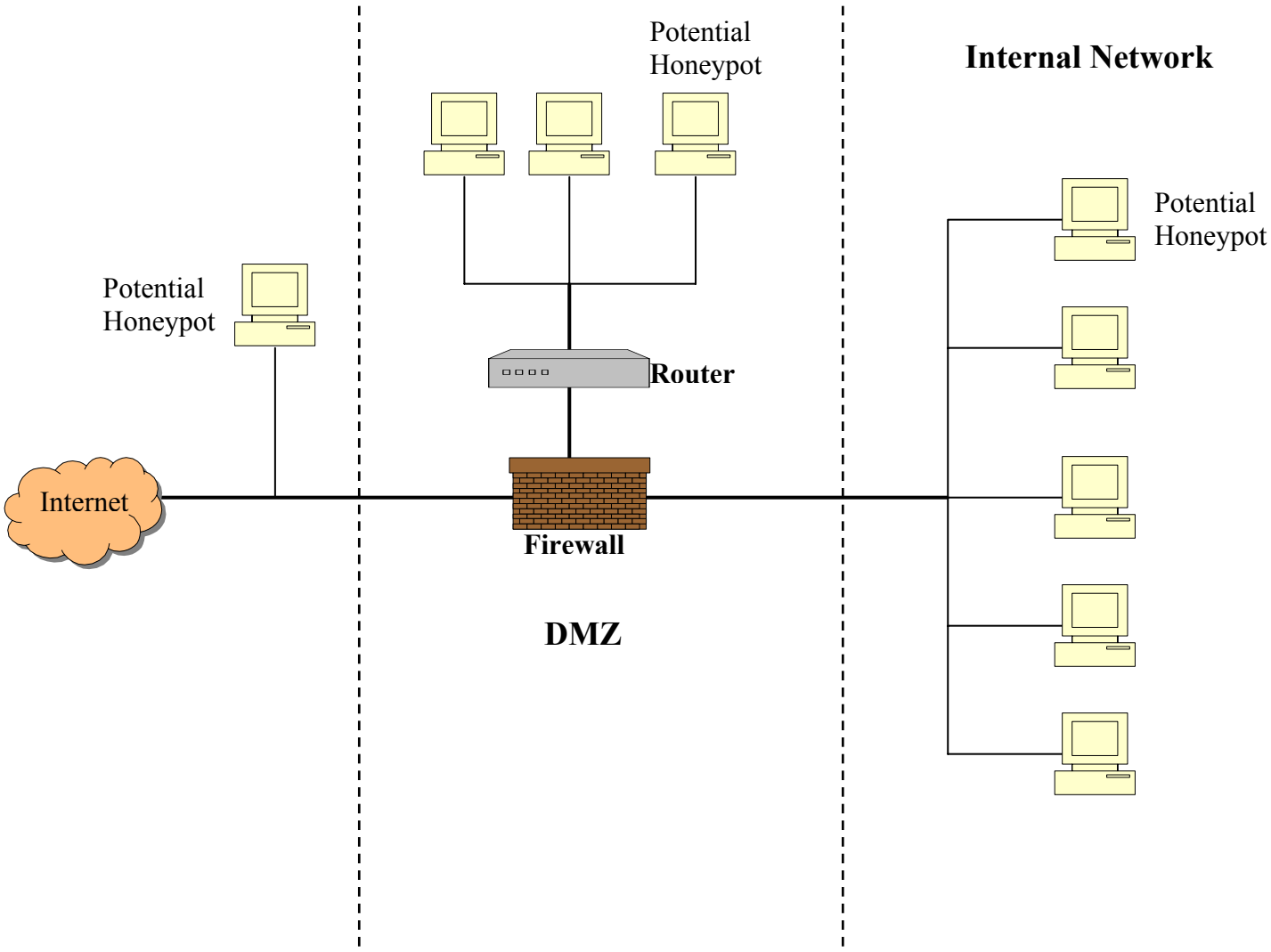


Figure 3 (What is a Honeypot? SANS Institute Resources)



Honeypots can be deployed inside, outside or within the De-militarization zone (DMZ). Usually it is deployed behind a firewall for control reasons. Placing a router between the firewall and the honeypot hides the firewall and makes it appear as a legitimate production system/network. In addition, it acts as a second access control device. To learn about inside attackers, the honeypot is deployed on the internal network. To learn about outside attackers, the honeypot is deployed within the DMZ or outside the DMZ.

BIBLIOGRAPHY

Loras R. Even. What is a Honeypot? SANS Institute. July 12, 2000
URL: http://www.sans.org/newlook/resources/IDFAQ/honrypot3.htm

Michael Sink. The Use of Honeypots and Packet Sniffers for Intrusion Detection. SANS Institute. April 15, 2001.
URL: http://rr.sans.org/intrusion/honey_pack.php

Computer Security: A Practical Definition.
URL: http://www.albion.com/security/intro-4.html

Mark Merkow. CCP, CISSP. Playing With Fire: Not So Sweet Honeypots. January 12, 2001.
URL:
http://ecommerce.internet.com/news/insights/outlook/article/0,,7761_559431,00.html

Mathew Schwartz. Networks use 'honeypots' to catch an online thief. Computerworld. April 4, 2001.
URL: http://www.cs.nmt.edu/~cs491_02/IA/honeypot.htm

Lance Spitzner. Honeypots: Definitions and Value of Honeypots. May 17, 2002.
URL: http://www.eneract.com/~1spitz/honeypot.html

Honeynet Project. Know Your Enemy: Honeynets. May 11, 2002.
URL: http://www.honeynet.org/papers/honeynet/

Douglas B. Moran. Vice President, Research & Development. Recourse Technologies, Inc. Trapping and Tracking Hackers: Collective security for survival in the Internet age.
URL: http://www.recourse.com/

Global Integrity Corporation (an SAIC company). Honeypot Effectiveness Study. Study conducted for Recourse Technologies, Inc. September 22, 2000.

The 'Lectric Law Library's Lexicon On Entrapment.
URL: http://wwwlectlaw.com/def/e024.htm

The Evolution of Deception Technologies as a Means for Network Defense. Recourse Technologies.