Global Information Assurance Certification Paper

SANS Institute GIAC Certification
GSEC Assignment #1.4

Honey Pots - Strategic Considerations

Abstract

Honey pots can play a key role in a defensive strategy. While there are many studies and sources for information, there is no single source that discusses the major strategic issues concerning honey pots. The main attraction of a honey pot is not limited to what you can learn but how you can learn it.

As a result, a honey pot fits into the defensive plan as a way of:

- Studying blackhat activity,
- Developing a reasoned response to the threat,
- Testing and developing new responses and tactics to a given threat, and
- Honey pots can also slow down an attack thus allowing time to develop a countermeasure.

Honey pots help to develop a reasoned, proactive response to a threat. In addition, they facilitate the building of contingencies thus contributing to the need to 'know what you don't know' in good project management practices. Honey pots are no panacea. There are significant risks and exposures to the organization if objectives are not well defined, do not implement a honey pot just for its own sake.

The primary issues to be addressed fall into two categories: Administrative/Policy and Technical.

The Administrative/Policy category covers legal, liability and misuse of data issues, while the Technical category relates to the choice of building or buying a honey pot, placement of the honey pot and support issues.

We have seen that the use of policy and procedures to manage these issues can go a long way to solve them. Some liability issues such as 'Uplink Liability' can be managed by sound configuration of the technical environment.

When configuring a honey pot, there are three areas of data management to consider: Data Control, Data Capture and, Data Collection.

Consideration must be given to both of these major categories before proceeding to implement a honey pot else; there is a significant risk of exposing the organization to both monetary and legal sanctions. Resolving these issues will build a better honey pot and help in ongoing management.

Introduction

The use of Honey Pots has grown steadily over the past few years. There are now a number of studies covering topics ranging from the building, implementing and evaluation of the honey pot's effectiveness in a defensive strategy. In addition, a number of commercially available honey pot systems are growing in popularity.

On the whole, there has not been, to date, a comprehensive review of the key strategic issues surrounding the decision to implement a honey pot. Nor has there been any work that focuses on the key benefits. This paper will define, review and discuss the major strategic issues to be addressed before making a decision on whether a honey pot should be implemented in your organization.

This paper is based on a review of over 20 studies of honey pot design, implementation and management and lists all of the sources at the end of the discussion.

What is a Honey Pot

A honey pot is a computer system is expressly set up to attract and "trap" individuals who attempt to penetrate other people's computer systems.

Generally, honey pots are seen as a good way to divert intruders away from the main productions system. This diversion results in lengthening the time security personnel has to study the exploit and time to develop countermeasures.

Honey pots are not recommended for the purpose of intrusion prevention. Honey pots do not prevent exploits from happening as their strength lies in the diversion of an exploit.

As a first consideration, this defines the limits to what honey pots can achieve and what role they may play in an overall defensive strategy. The effective use of a honey pot is therefore dependant on:

- A comprehensive policy governing its use along with clear objectives;
- Responsible design of the honey pot system to minimize negative exposures (e.g. uplink liability) and maximize data collection capabilities and,
- Sufficient resources and proper training to provide ongoing care and feeding.

This should be in place before going on to implementing a honey pot. Avoid using honey pots simply for the sake of using them; a specific objective or series of objectives must be well defined before proceeding.

Types of Honey Pots

In general, honey pots can be deployed in two forms: Production and Research[1] and each has a specific purpose:

Production honey pots serve the intrusion detection objective insofar as they simulate 'real' production systems using structures and data that parallel the actual production system.

Research honey pots are deployed to provide information on general blackhat activities. The system structure and data do not necessarily reflect any 'real' production system. Their purpose is to provide general intelligence to the security community. By convention, research honey pots are referred to as 'Honey Nets'.

Return on Investment - Benefits

A Honey Pot's Role in the 'Security Equation'

The Defense in Depth (DiD) strategy is designed to provide overlapping layers of fences, walls, detection devices and policies in order to counter a given exploit. The 'depth' is created so that there is a chance to detect an exploit even if it has succeeded in penetrating a layer in the defense net. A properly designed, deployed and managed Defense in Depth strategy will produce all of the necessary raw data for an administrator at learn about what types of exploits are being tried on the system.

DiD also does one other important thing, it slows down an attack by providing obstacles to the exploit. Effective security results when a hacker is delayed enough so that the time spent in penetrating the secured system (Pt) is greater than the time needed to detect (Dt) and respond (Rt) to the exploit.[2]

Put in an equation format:

Pt >Dt +Rt

Based on this concept, a defensive strategy is designed to slow down an attack long enough so that the exploit can be detected and a response can be developed before the attack succeeds. No good bolting the barn door after the cows have run away!

So slowing down an exploit increases the chance of detection and increases the chance of countering that exploit. Honey pots were conceived as systems or network of systems constructed to lure hackers away from real production systems. Within the honey pot, intrusion detection and logging applications are deployed. The purpose is to watch and listen for intruders and log all of their activities in an effort to discover their methods and to develop countermeasures.

Honey pots are excellent learning tools, they provide an environment where activities can be monitored and logged for study. So what can be learned from honey pots that cannot be learned elsewhere? The short answer is: Nothing. A properly prepared, executed and managed DiD strategy is capable of identifying all of the necessary parameters of a given exploit. The key is not *what* can be learned but *how* one can learn it.

Honey Pots and Proactive Management

Another benefit of honey pots is how it contributes to good project management. The principle of: 'Know what you don't know', underlines the need for contingency planning.[3] Good contingency planning is *proactive*; the only *reacting* that should be done is the deployment of countermeasures identified as part of contingency planning or otherwise known as the search for knowing what you don't know. Leave the firefighting to the local Fire Department.

Benefit Summary

Honey pots provide these essential elements:

- They divert exploits away from the main production system;
- They allow you to gather information;
- They allow you to develop and test responses;
- They are consistent with the DiD goal of slowing down exploits, and
- Because they are non-production systems, they allow you to do all this in a (relatively) safe environment. A test or non-production environment allows us to experiment freely and you can take chances and try out counter exploits with a greater freedom.

Strategic Issues

There are two main categories that strategic issues fall into; Administrative/Policy related issues and Technical issues. These can be summarized as follows:

Category	Issue
Administrative/Policy	Data Selection and (Mis) Use
	Compromised Honey Pots
	Legal Exposures
Technical	Build/Buy
	Placement
	Support

Administrative/Policy Issues

Data Selection and (Mis) use

Most organizations will choose a production type honey pot to implement. This means that choosing the data to be used in the honey pot is important. Simulating a production honey pot requires realistic data to populate the system and using an extract from the actual production system would seem to be a logical choice. Now consider what the implications would be when the honey pot is compromised and the intruder chooses to use the data against the organization.[4] Clearly, this is no trivial matter. Honey pots will be compromised and appropriate contingencies should be in place.

Policy and procedures are one part of the best defense against this situation. Developing a policy on this issue will require a risk analysis to help determine critical exposures the organization may face. In addition, the policy will provide a procedure to follow just in case the false data is published or used in any other negative way.

The risk analysis will help to determine what data to use. Using a completely fictional database or an old or outdated database are two possible choices. Each will have its own risks associated with it. In both cases, the data may give a false impression to shareholders and the public, if released.

After weighing the options, the appropriate policy should be in place as a guide. As an example, the procedure may call for taking the honey pot offline immediately or may decide to risk the release of the data in order to gain more information on the exploit in progress at the discretion of some pre determined authority.

Compromised Honey Pots

Honey pots will be compromised. Of prime concern is minimizing the damage to your system and to other external systems. Once compromised, the honey pot can be used by a hacker to attack other systems and is known as 'uplink liability'[5]. A compromised system could serve as a platform for a distributed Denial of Service (DoS) attack, ICMP attack or having a Trojan added to the honey pot.

So the question becomes: Do you allow the compromised system to continue running or do you pull it off line? This may depend on what your goals are. If you are looking to study/document blackhat activities, you may want to continue operating the honey pot. Consideration should be given to liability concerns should a compromised honey pot be used to attack other systems.[6]

Consider the legal and liability issues before you design and implement the honey pot will save you piles of grief in the long run. The resolution of these issues will determine the type of data stored on the honey pots and determine key policy areas.

There are many ways to limit the activities of intruders by applying control measures thus minimizing uplink liability. A number of these measures will be discussed in the Technical Section below.

Legal Exposures

Entrapment

> "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".[7]

This is or is or is not an issue depending on whether you believe that administrators and IT personnel are not in law enforcement and therefore outside of the definition of entrapment.

Consider the following questions. Are you making the 'pot' too enticing or too easy to penetrate? It could be argued that the unlocked door or open window you left open was an enticement to the curious. Make sure that anyone penetrating your honey pot has to do so by breaking in. As a result, this becomes a liability issue. Consider that you can be judged partially responsible for a break-in if you leave too many open doors. This aspect of partial responsibility becomes even more important if you decide to prosecute an individual for a security violation.

Technical Issues

Build or Buy

There are currently three options available to the potential honey pot owner.

Option One – Purchase a commercial product. There are a number of commercially available turnkey products that can simulate entire network segments, all run on a single host. Applications, such as Recourse Technologies Manhunt[8] or Specter's SPECTER Intrusion Detection System[9], typically require a fully loaded dedicated host with a high processing power and memory requirement. This is also the most costly option.

Option Two – Take advantage of products available for free or for a small fee. There are a number of products like Deception Toolkit[10] and BackOfficer Friendly[11] that simulate servers. These applications listen for traffic on TCP ports

(inbound) for common services like; FTP, telnet, HTTP, Back Orifice, etc. They use scripted responses that simulate the normal responses one would expect from a server. Although the deception is relatively easy to detect, the systems can provide valuable data with minimal demands on resources.

Option Three – Build your own honey pot. Honey pots are basically, a simulation of the production system with enhanced detection and logging capabilities. Start with a computer loaded with your main operating system, network structure and a choice selection of data to entice an intruder. Research has been looking at the many ways operating systems can be used in honey pots. A recent report discussed the use of VMware[12] as a platform for a honey pot.[13]

Your choice of operating system will determine which tools are appropriate. However, there are three critical requirements that are common to building a honey pot.[14]

Honey Pots Common Requirements	
Critical Requirements	**Description**
Data Control	Controls activities of attackers by limiting options
Data Capture	Collecting and recording activities on the honey pot
Data Collection	If more than one honey pot is in operation, then data needs to be collected from these remote sites

Data Control

This area is designed to control the activities of the intruder. The main objective is to minimize the use of the compromised honey pot as a platform to attack other systems. A common practice is to limit the number of outbound connections that can be made from the compromised system. Limiting the number of outbound connections minimizes or prevents the compromised honey pot from being a major staging area for: Distributed DoS attacks (DDoS), SMURF, Ping of Death, or ICMP attacks.

The most common tool to achieve this is to use a firewall capable of limiting the number of outbound connections. A combination of firewall and shell script, to

count the number of connections, can be used. In addition, products such as IPFilter, SWATCH or IPTables can be used to develop this capability.

Many systems use a firewall/router combination to limit activities. The router is configured to pass only local packets on the outbound connections thus limiting DoS or ICMP based attacks. Again this places limits on the use of the compromised system against other/external systems.

Data Capture

This is designed to record all of the activities on the honey pot and contains three layers. The first layer is data from the firewall, which logs all connections and sends an alert when the maximum number of outbound connections is exceeded.

The second layer is built around an Intrusion Detection system (IDS), which is used to capture all network activity by capturing and recording packets. SNORT[15] is one of the most popular packages in this category. In addition, the IDS can be configured to send alerts based on predetermined criteria. A tool such as SWATCH[16] can be used to store and archive this data.

The third layer is data from the monitoring of system and user activity. Both remote and local activities must be monitored to cover off all possibilities. It is also important to note that local activities are as important as remote activities. Monitoring local activities will help to determine internal threats or internally compromised systems. Other methods may involve the capturing of keystrokes for a more detailed investigation. Products like TTY Watcher[17] can be used or you can use a modified bash shell like the one provided by Antonomasia.[18]

Data Collection

This function is necessary if you are operating more than one honey pot. This layer provides the capability to collect data from remote sites and transfer them to one central location. You will be required to create unique identifiers for each system so that the data can be identified when centralized.

There are also a number of considerations that are dependant on the selection of operating system. These requirements vary with the structure and known vulnerabilities of the particular operating system. Tools such as Chkrootkit[19], determine if a rootkit signature has been added to a LINUX or UNIX implementation. Likewise, if a Windows based OS is used, a set of specific tools is needed.

Choosing an Option

The choice of option depends on a number of factors:

- Monetary resources, turnkey solutions are the most expensive and the build your own could end up costing nothing more than time,
- Support resources, in all cases the care and feeding of a honey pot requires knowledgeable, well-trained personnel,
- Support time, resources are required to gather, read and analyze logs, scans and other captures to determine if a system has been compromised. Systems can be compromised relatively quickly, leaving hours of rebuilding time to restore the system.[20]

Given the array of tools available, there is a need to develop clear evaluation criteria to identify the best match for your needs. In this case the '80/20' rule is invaluable.

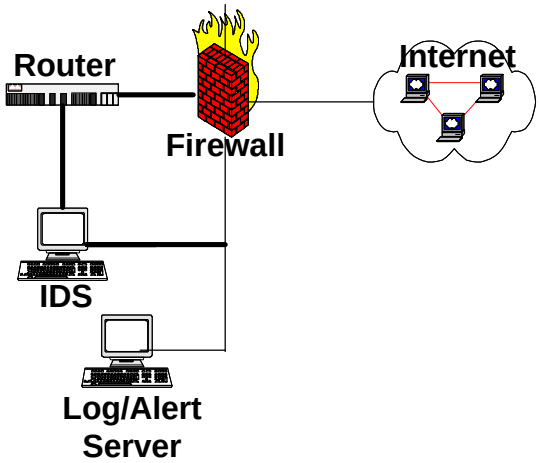
This rule states that any potential tool will perform 80% of the requirement adequately, 10% a superior fashion, and the remaining 10%, in an inferior fashion. The goal is to identify the 10% of your requirement that must be done in a superior fashion. By matching the critical 10% of your need to a specific tool, 90% of the requirement will be met.

Placement

We have already seen that placement of the honey pot is critical in protecting the internal network and in placing limits on your uplink liability by preventing your honey pot from becoming a platform to attack external systems.

There are a number of choices in this but the basic rule is to locate the honey pot where it is relatively isolated from your production system. In addition, you need so way to control and monitor inbound/outbound traffic. Other factors, such as rules for the firewall and router will depend on whether you are monitoring for internal or external intrusions. Two examples are as follows. The first is the one described in the previous section and is based on the Honeynet Project.

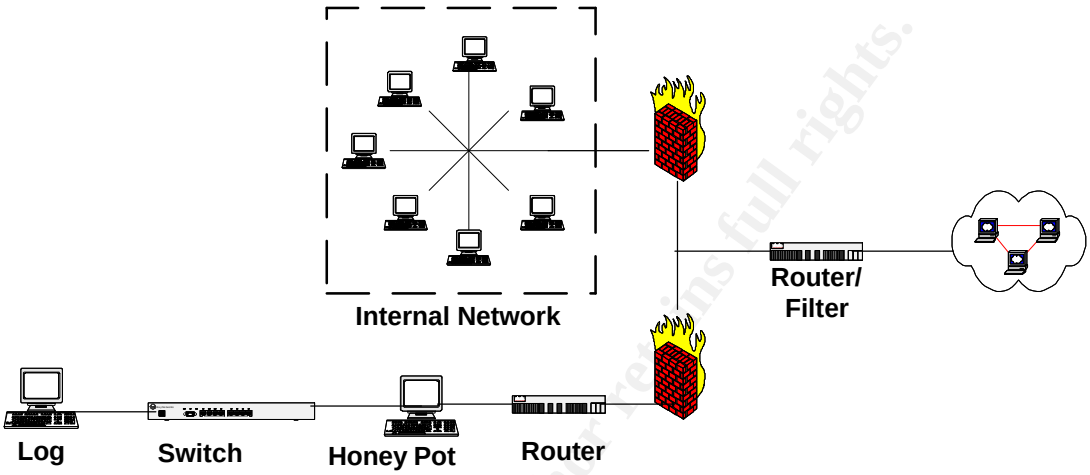
Example 1 Honey Pot Placement[21]



In this example the router and firewall work together to monitor and control inbound and particularly, outbound traffic. Note how the IDS is connected to the router and firewall to cover off all of the monitoring requirements.

Example 2 Honey Pot Placement[22]

In this example, there is a slightly different arrangement of devices:



A DMZ is created for the honey pot to reside in. The router and firewall are still combined to control all traffic in and out of the network.

It is interesting to note that a switch is used to isolate and keep the logging service relatively stealthy.

Support

DiD Meets SiD - a Delicate Balance

There is general agreement that configuring a honey pot is not a trivial matter. In their current state, honey pots are not for the 'faint of heart'. They require a knowledgeable, well-trained support team. Configuring a honey pot requires the careful layering of intrusion detection, monitoring and logging tools.[23]

Once the honey pot is up and running, expect it to generate megabytes of valuable data, all waiting for someone to review and process into useful information. The commitment of resources to the ongoing care and feeding of the honey pot and the ability to keep up with the data that is generated is of prime importance. Otherwise you are finished before you start.

You must have Support in Depth (SiD) to take advantage of the treasures mined by a honey pot.[24] If you do not have the staff in place and you want to proceed, the building of a SiD becomes a strategic objective in your planning process. SiD

is a focal point where strategic and tactical issues come to the foreground. As a result, the support issue becomes a balance between all of the competing demands placed on today's dynamic support team.

Conclusions

We have seen that honey pots can be a valuable addition to a security system. Honey pots serve the community through their ability to collect and record information on blackhat activities. A Honey pot's strength is its ability to divert an attacker from the main production system. Once diverted, the attack can be studied and a countermeasure can be developed. In fact, the honey pot's main advantage is *how* data is collected rather than *what* data is collected. Honey pots do not collect any different data than a well-secured production system; their advantage is in the 'how' data is captured.

There are a number of universal issues surrounding the use of honey pots. Clear objectives must be developed before deciding on implementing a honey pot. If not, the honey pots can cause much trouble.

These issues fall into two categories: Administrative/Policy and Technical. There is some overlap between the categories and in the case of uplink liability, a policy and technical solution is required.

Administrative issues relate mainly to legal and liability concerns, while the Technical issues are concerned with how to: build, place and support the honey pot.

In general, the honey pot design should address three main areas: controlling, capturing and collecting data and are related to how the honey pot system is designed and configured.

Finally, the issue of support should be addressed. Honey pots can be complex and require a well-trained staff. In addition, honey pots require ongoing care and feeding to analyze data and repair damage from attacks.

Organizations need to clearly understand all of the benefits and risks associated with the use of honey pots and by proper planning can maximize the benefits of using them.

Notes

[1] Spitzner, Lance "Honeypots, Definition and Value of Honeypots" http://www.enteract.com/~lspitz May, 2002
[2] Schwartau, W. Time Based Security: Practical and Provable Methods to Protect Enterprise and Infrastructure, Networks, and Nation. Seminole, Florida: Interpact Press, 1999.
[3] Wang, Gene The Programmer's Job Handbook, Osborne-McGraw Hill, ISBN 0078821371, October, 1995
[4] Schwabel, Josephine, et. al, "Lessons Learned from Deploying a Honey Pot". Information Security Bulletin, CHI Publishing, December, 2000 www.chi-publishing.com/isb/
[5] The Honeynet Project, "Know your Enemy: Honeynets", http://project.honeynet.org/ May, 2002
[6] Forristal, Jeff "Luring Killer Bees With Honey" 21 August 2000 http://www.networkcomputing.com/1116/1116ws3.html
[7] Spitzner, Lance "Honeypots, Definition and Value of Honeypots" http://www.enteract.com/~lspitz May, 2002
[8] Recourse Technologies, http://www.recourse.com/product/ManHunt/
[9] http://www.specter.com/default50.htm
[10] http://www.all.net/dtk/
[11] http://www.nfr.com/products/bof/
[12] VMware is essentially a set of software products, the workstation version installs onto Windows or Linux and allows you to run numerous Intel based operating systems on top of it
[13] Seifried, Kurt, Honeypotting with VMware – basics, http://seifried.org/security/
[14] The Honeynet Project, "Know your Enemy: Honeynets", http://project.honeynet.org/ May, 2002
[15] http://www.snort.org
[16] Spitzner, Lance "Watching your Logs", July 19, 2000. http://www.enteract.com/~lspitz/swatch.html
[17] http://project.honeynet.org/papers/honeynet/doord-src.tar.gz
[18] http://www.notatla.demon.co.uk/SOFTWARE/honeypot_code_description.html
[19] : http://www.chkrootkit.org
[20] The Honeynet Project, "Know your Enemy: Honeynets", http://project.honeynet.org/ ,May, 2002
[21] The Honeynet Project, "Know your Enemy: Honeynets", http://project.honeynet.org/ ,May, 2002
[22] Moon, Karen, "Effective Honeypots, GSEC Assignment 1.3, May 2002, http://www.giac.org/practical/Karin_Moon_GSEC.doc
[23] Schwabel, Josephine, et. al, "Lessons Learned from Deploying a Honey Pot". Information Security Bulletin, CHI Publishing, December, 2000 www.chi-publishing.com/isb/
[24] The Honeynet Project, "Know your Enemy: Honeynets", http://project.honeynet.org/ , May 11, 2002

Additional Sources

Even, Loras R**.,** What is a Honeypot? Honey Pot Systems Explained, SANS Reading Room, Intrusion Detection FAQ July 12, 2000 http://www.sans.org/newlook/resources/IDFAQ/honeypot3.htm
Kane, Gregory, "Honey in the Pot or Tar in the Pit", GSEC Assignment 1.4 June 2002, http://www.giac.org/practical/Gregory_Kane_GSEC.doc
Kilpatrick, Ian, TECHS Library of Information and IT Security Papers, http://www.itsecurity.com/papers/honeypot.htm
Klug, David, Honey Pots and Intrusion Detection, SANS Reading Room, http://rr.sans.org/intrusion/honeypots.php
Moran, Douglas, B. Trapping and Tracking Hackers: Collective security for survival in the Internet age, Proceedings of the Third Information Survivability Workshop, Boston, Massachusetts, October 24-26, 2000, http://www.cert.org/research/isw/isw2000/papers/15.pdf
Raikow, David, Building Your Own Honeypot, CNET Australia, http://www.zdnet.com.au/newstech/security/story/0,2000024985,20106785,00.htm

Recourse Technologies, “The Evolution of Deception Technologies as a Means for Network Defense, http://www.recourse.com
Schlotter, Chadd, Anti-Hacking: The Protection of Computers, SANS Reading Room, http://rr.sans.org/attack/antihack.php
Sink, Michael, The Use of Honeypots and Packet Sniffers for Intrusion Detection, SANS Reading Room, http://rr.sans.org/intrusion/honey_pack.php
Stoll, Clifford Stalking The Wily Hacker, Communication Of The ACM, May 1988 vol. 31. No. 5, http://www.acm.org/cacm/toc/0_toc_02.html