Honeypots

Sweet and sticky for the cyber "bad guys"

Nancy Rader

GIAC Security Essentials Certification (GSEC)

Practical Assignment

Version 1.4b

Option 1

12/06/2003

Abstract

This paper attempts to give both the author and the reader a broad-based understanding of honeypots, one of the many techniques used to help secure network connected computer systems. In addition to a general description and example of a typical deployment, the paper gives a bit of information about the honeypot's "formative years" and its recognized creator, Lance Spitzner. To explain the honeypot concept in greater detail, the classifications of honeypots are described, with examples of each along with suggestions of how to use them for the most value. Legal concerns of honeypot usage are addressed before closing with some of the advantages and disadvantages of the honeypot technique and my concluding qualified endorsement.

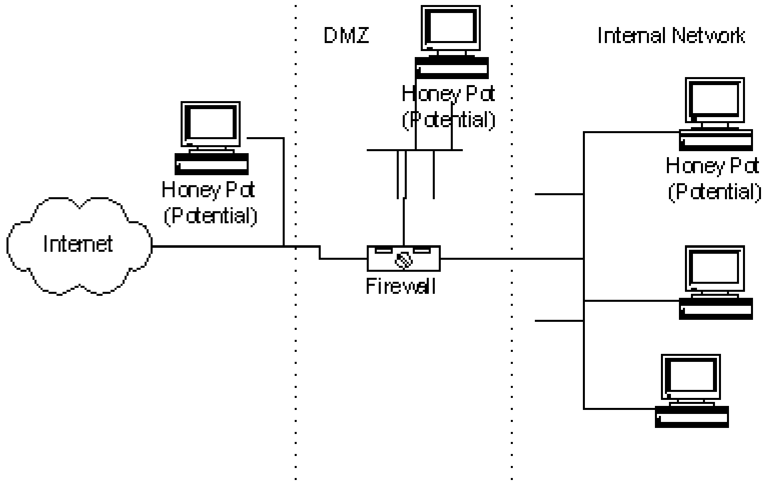
Introduction

As unethical hackers (commonly referred to as blackhats) become increasingly inventive and proficient in their efforts to gain unauthorized access to computers, novel intrusion detection and defense strategies are being developed to counter their activities. Routers, firewalls, and Intrusion Detection Systems (IDS) are traditionally thought to describe defense in depth on Internet attached networks. However, one other layer of defense is fast emerging…the **honeypot**. Based on the principle that to successfully defend yourself against the enemy, you must first know who the enemy is, the honeypot "takes the appearance of an attractive service, set of services, an entire operating system, or even an entire network, but is in reality a tightly sealed compartment built to lure and contain an attacker" (Scottberg, Yurik, Doss, 2002) for the primary purpose of gathering information about how they gain access to a computer. From a record of intruders' activities, insight can be gained into attack methodologies: that is, how they choose their targets, the tools they use, and how they cover their tracks. Since a honeypot is not part of a production system, there is no "legitimate" use for it. Therefore, any interaction with a honeypo**t** should be considered suspect and carefully logged and studied. Information gathered by the honeypot can be used to develop better system security to more effectively protect critical information against unauthorized access.

George Bakos, senior security expert at the institute for Security Technology Studies at Dartmouth College in Hanover, N. H. says this about honeypots…"It's all about appearing to be something you're not to get the baddies to show their hand. The information we glean from it is fantastic. You can observe details of the compromise—what technology they use, their intent, motivations and the resources they went after…They give us a leading indicator of things to come." (Gaudin, 2002)

One of the earliest and possibly the most famous description of the use of honeypots is found in Clifford Stoll's best selling novel, *The Cuckoo's Egg: Tracking a Spy through the Maze of Computer Espionage.* Written in 1989**,** Stoll describes how he sets up a fake government project, complete with simulated files from the Strategic Defense Initiative, to attract intruders. While the intruders spent extended periods of time downloading and analyzing, Stoll was able to monitor all their activities and trace them back to their source in Hanover, Germany**.**

"Honeypots can be setup inside, outside, or in the DMZ of a firewall design or even in all of these locations, although they are most often deployed inside of a firewall for control purposes." (Even, 2000) Loras Even includes the following example of a honeypot system installed in a traditional Internet security design in his article "What is a Honeypot"



Lance Spitzner

According to Keith Johnson, technical writer with the Wall Street Journal, the honeypot's creator is the self-described computer geek, Lance Spitzner. Spitzner, recognized by many as the leading authority and spokesperson on honeypots, is currently a security consultant with Sun Microsystems, Inc. in Chicago and lead analyst on the Honeynet Project, an all volunteer, not-for-profit security research organization. Spitzner says he is applying tactics and techniques he learned as a tank commander in the U.S. Army to the cloak-and-dagger world of Internet security. Spitzner states "I used to have to crawl around inside Soviet T-72 tanks to get an idea what the enemy was doing, what they had to work with. Now I'm doing the same thing, just with different tools." (Johnson, 2000)

Spitzner's military experience also taught him the value of having reliable information on the enemy. When he joined Sun in 1998 as a consultant advising corporate clients on security issues, he found "There was little information out there on just who these hackers were, on what motivated them, on how they operated." (Johnson, 2000) To gain this information for himself, Spitzner built his first honeypot in early 1999.

Spitzner's goal in building a honeypot was not to capture attackers, but to learn as much as possible about their attack tools to better understand what vulnerabilities and threats existed for his production network. Spitzner built his first honeypot using a standard Linux box that mirrored his production system. He did not do anything special to the box, just built it like any other thinking this simple system would be less likely to make attackers suspect a deception. He then put the box on the Internet and waited for the blackhats. Within 15 minutes, a hacker scanned his box and erased its hard drive when he became aware he was being watched.

But Spitzner realized attracting the hackers to his box was only the first step. He also had to be alerted when an attack occurred, track the attackers' every move, and prevent the attacker from compromising production systems. To accomplish this, Spitzner put the honeypot on its own network behind a firewall. This, according to Spitzner, was the answer to many of his needs:

- First, most firewalls log all traffic going through it. This becomes the first layer of tracking the blackhat's moves. By reviewing the firewall logs, we can begin to determine how blackhats probe our honeypot and what they are looking for.

- Second, most firewalls have some alerting capability. You can build simple alerts whenever someone probes your network. Since no one should be connecting to your honeypot, any packets sent to it are most likely a blackhat probing the system. If there is any traffic coming FROM the honeypot out to the Internet, then the honeypot was most likely compromised.

- Third, the firewall can control what traffic comes in and what traffic goes out. In this case, the firewall lets everything from the Internet in, but only limited traffic out. This way the blackhats can find, probe, and exploit our honeypot but they cannot compromise other systems. (Spitzner 1999)

In addition to the firewall logs, Spitzner uses other tracking methods to create a multi-layered tracking system. By using more than one tracking method, it is less likely that a hacker will be able to alter or destroy all the tracked information before he is discovered and booted off the system. Also, tracking by different sources provides different information. Spitzner' tracking layers include:

- System logs on the honeypot to help identify changes or attempted changes;

- Sniffers to see the packets going between the firewall and the honeypot**;**

- Modifications to the shell to capture keystrokes to syslog;

- Tripwires on the honeypot to indicate what binaries have been altered on a compromised system.

Hardware or Software

Spitzner's technique of a standard hardware based honeypot made up of servers, switches, and routers set up to mimic an actual production network is not the only honeypot design concept. There is also a form of "virtual" honeypot consisting of software emulation designed to look like a real working network. Several commercial products are available which offer high-powered honeypot software packages capable of simulating entire network segments on one machine. One such product is titled CyberCop Sting built by Network Associates. CyberCop Sting is sophisticated enough to imitate multiple operating systems simultaneously, including Linux, Solaris, Cisco IOS, and Windows.

ManTrap, newly named Decoy Server, from Symantec, is another excellent example of honeypot software. Mantrap is made up of complete and realistic operating systems for attackers to interact with and has outstanding data collection capabilities. It provides early detection of internal, external, and unknown attacks and unauthorized use of passwords and server access. Advanced filtering capabilities allow Mantrap to discard insignificant events, and focus only on the data requiring a response. Lance Spitzner, who has said he is not a big fan of honeypots that emulate known services because it is hard to outwit real hackers with pseudo software, endorses Mantrap because it runs on real, semi disabled hardware and looks real to hackers. According to Spitzner "What's cool about Mantrap is it doesn't emulate anything." Mantrap sounds a silent alarm when traffic is detected then precedes to log all intruder activity, including keystrokes, for forensic use. (Schwartz, 2001)

Honeypot simulation software from sNet Systems Corporation may be attractive to corporations with a need to safeguard business secrets because it adds IP logging to a simulated Windows NT environment. Because Windows NT does not log IP addresses, only computer names, becoming aware of inside efforts, like someone attempting to access secured files in Network Neighborhood, is difficult without additional tools like those offered by sNet. According to Barry Schlossberg, security advisor at sNet …the theory is that if someone has clicked, say, 10 folder levels down in Network Neighborhood into areas they are not authorized to see, they might have malicious intent. "There's nothing wrong with casual attempts to rattle the door, such as looking around here and there, but how can an 'innocent' person use 50 different log-ins, then 'borrow' the customer database?"

For someone wishing to build their own honeypot, Fred Cohen's Deception Tool Kit includes complete honeypot instructions. DTK, first released in 1997, is the original OpenSourse honeypot. It is a compilation of Perl scripts and C source code that emulates a variety of listening services with the primary purpose of deceiving human attackers. It works by creating the appearance of a highly vulnerable system and providing known responses to make it seem as though intruders have attacked a real system.

Whether the honeypot basis is hardware or software, it may be that the most important attribute of the honeypot is psychological. It has to look attractive, challenging, be easy to break into but not too easy or hackers will easily identify the honeypot and go after other servers on the same network. To intimidate, a honeypot may advertise a default banner such as "Honeypot in Use" on an unauthorized deception port to psychologically increase an intruder's level of uncertainty in a manner similar to posting a security alarm or "Beware of Dog" sign in the physical world.

Low Interaction and High Interaction

In addition to their type, honeypots can be categorized by class, i.e., low interaction or high interaction. Interaction refers to the extent of attacker activity the honeypot allows. A low interaction honeypot is comprised of a limited number of fake services such as HTTP (Hyper Text Transfer Protocol), and SMTP (Simple Mail Transfer Protocol). They are relatively easy and affordable to deploy and maintain and offer little risk because an attacker never has access to a real operating system from which to launch an attack or do harm to others. Setting up a low interaction honeypot can be as simple as installing a software package on a PC and selecting an operating system and services to emulate and monitor. Although limited, they can provide useful information at a high level about network probes, worm activities, and spammers.

One example of a low interaction honeypot is Honeyd. Developed by Niels Provos, it is OpenSource and can be built to run on both Unix and Windows systems. Honeyd works by monitoring unassigned IP space. When Honeyd detects an attempt to connect to an unused IP, it intercepts the connection then interacts with the attacker, simulating responses as the victim while capturing the attack activity.

While Honeyd is generally built to detect activity at any UDP and TCP port, Honeyd can be configured to emulate services to monitor specific ports, such as an emulated FTP server monitoring TCP on port 21. When Honeyd detects an intruder connection on the emulated FTP service, it can do more than log the activity; it can capture all of the attacker's interaction with the emulated service. In the case of the emulated FTP server, Honeyd may be able to capture the attacker's login and password, the commands they issue, and even learn what they are looking for or their identity, depending on the level of emulation of the system. However, like most other low interaction honeypots, Honeyd, has limitations. It is programmed to identify specified attacker behaviors and to give predetermined responses. If an attacker does something Honeyd can not recognize, it will simply respond with an error message. (Spitzner 2003)

In contrast to low interaction honeypots, high interaction honeypots are quite complex, involving real systems and applications designed to be compromised by intruders in order to collect realistic data. High interaction systems are not limited by a specified set of attacker behaviors. They provide an open environment that can capture a large volume and variety of activity. This characteristic makes it important that these systems be tightly controlled so they do not become a host to an attack on another system. The additional technologies required to prevent an attacker from harming other non-honeypot systems makes high interaction systems more demanding and costly to deploy and maintain.

Honeynets are an excellent example of a high interaction honeypot. Honeynets are much more than a single product or a software package ready to be installed on a computer. Honeynets are comprised of an entire network of computers like Cisco switches and routers and Windows, Linux, and Solaris boxes, all partially disabled but convincingly realistic to a potential attacker. The goal of the Honeynet is to create a tightly managed network where all hacker activity can be controlled and captured without any awareness on the part of the attacker. The Honeynet controls attacker activity with a gateway that allows inbound traffic to the victim systems but disallows outbound traffic with prevention technologies. With this system, the attacker has the ability to interact with the victim systems, but can not launch an attack from them or perpetrate harm to any non-Honeynet computers.

Research and Production

Honeypots can also be categorized by their value. Marty Roesch, creator of Snort, divides them into two broad categories: production and research. In general, low interaction honeypots are most often used for production purposes and high interaction honeypots are used for research purposes.

Production honeypots are used primarily by large commercial organizations to help protect their networks. Research honeypots do not add direct value for a commercial organization. Instead, research honeypots are most often used by education, government, military, or security research organizations to study the threats organizations face and learn how to protect against those threats. (Spitzner, 2002)

Production honeypots are intended to protect an organization by "preventing, detecting, or helping companies respond to an attack". (Spitzner, 2003) When used as a guinea pig or "sacrifice box", the honeypot can draw an attacker away from other boxes in a network. An attractive and vulnerable honeypot may keep an intruder occupied for hours, even days, using his resources to attack a honeypot and giving a security team time to recognize and close holes in the production system to prevent the attacker from gaining access. Sacrifice boxes, unlike commercial honeypots, have minimal hardware requirements and use standard operating systems and software. This makes them relatively inexpensive, easy to implement, and difficult for an attacker to distinguish from production boxes.

Honeypots called sticky honeypots or tar pits defend against automated attacks such as worms or auto-rooters. These attacks use tools that randomly scan entire networks looking for vulnerable systems. If vulnerable systems are found, the tools will attack and take over the systems, and in the case of worms, self replicate, copying themselves to the victim. Sticky Honeypots defend against these attacks by slowing down or even stopping the attacker's scanning using various features of TCP/IP such as the "receive window" feature, i.e., information a networked system can include in a SYN/ACK packet to tell how busy it is in terms of bytes of data it is prepared to accept. When a system is heavily saturated, it may tell a remote peer that it has "zero space" for new data forcing the client's TCP connection into a wait loop as it keeps checking back with the tar pit to see if it has space to receive data. (Ranum, 2002)

LaBrea, developed by Tom Liston in 2001, is one example of a tar pit that takes advantage of TCP tricks to slow network attacks. Independent security consultant and author, Marcus Ranum, tested LaBrea with a couple of scans on his network and found it slowed scans to the point where they did not complete at all. Ranum cleverly explains how LaBrea works to protect an organization by using a telemarketing call as an analogy. He says… "LaBrea is the networking equivalent of answering the phone, hearing that it's a telemarketer, and saying, 'Hang on, you need to talk to my daddy!' and then putting down the phone with no intention of actually speaking to the caller. Just when you think the telemarketer will hang up, you yell, 'Hang on another minute, daddy is on his way!'"

Some subject matter experts, including Lance Spitzner, feel honeypots add little value to prevention but do add extensive value to detection. Organizations may be so overwhelmed with activity on their production systems that they find it extremely difficult to detect attacks. Traditional Intrusion Detections Systems generate a huge volume of alerts, including a high percentage of false positives. The volume of this reporting takes extensive amounts of time, resources, and funds to review and analyze. The number of false positives leads administrators to lose confidence in the IDS and begin to ignore its warnings. In contrast, honeypots collect data only when direct interaction occurs. And since a honeypot should not see any traffic because it has no legitimate purpose, interaction with a honeypot is very likely from an unauthorized or ill-intended source. As such, honeypots collect only small data sets and nearly eliminate false positives. This makes it easier and less expensive to analyze honeypot collected data and derive value from it.

Honeypots can also provide value in incidence response. Commonly, organizations that experience a compromised system are unable to stop production activity after the compromise occurs. Continued activity combines normal day-to-day activity with the malicious activity making evidence gathering far more difficult. It may be impossible to take compromised systems off-line because the services they offer cannot be eliminated. As a result, an incident response team is unable to conduct a complete forensic analysis. Honeypots can reduce or eliminate these problems. Because they have very limited data collection, they drastically reduce the need to sort through large quantities of data; and because they are not part of the production system, they can be readily taken off-line to allow for a full forensic analysis. Based on this analysis, it may be possible to learn who the attacker is, how the attacker got in, and what he did once he had access. The newest high interaction honeypots are built with even stronger threat response mechanisms. Some are able to shut down systems when attacker activity is detected. Others have frequency-based policies that allow security administrators to control the actions of the attacker within the honeypot.

Research honeypots help to resolve one of the biggest challenges the security community faces: the lack of information on threats. They collect as much information as possible to learn from it, information few other solutions are capable of gathering. They also provide a platform to study the threat, to record, step-by-step, the activities of attackers as they attempt to compromise a system and what they do after an attack, such as communicating with other attackers or uploading a new tool kit. As stated by Lance Spitzner, "How can we defend against an enemy when we don't even know who that enemy is? For centuries military organizations have depended on information to better understand who their enemy is and how to defend against them. Why should information security be any different?" (Spitzner, 2003)

One of the most widely known examples of using honeypots for research is the work done by the Honeynet Project. Founded in April, 1999, the Honeynet Project is a security research organization whose goal is to learn the tools, tactics, and motives of the blackhat community and share the lessons learned with the security community. Honeynet has no products, services, or employees; all of its research is done on a volunteer basis. The Honeynet Project aims to raise awareness and educate other professionals about security risks through published papers detailing their findings and the group's book "Know Your Enemy." Stephen Northcutt, of the SANS Institute, endorses Honeynet when he states "The work they have done putting systems up and seeing how long they last is quite a valuable service. You can point out to people who don't want to be concerned about security and show them in gory detail that you have to be concerned." (Fordahl, 2001)

Honeypots may also be useful as a teaching tool, and are currently included in some commercial training programs, like the Ultimate Hacking: Hands on course by Foundstone. At the Vermont National Guard, Honeypots are used to train students in the Computer Emergency Response teams, the group that teaches network security to military IT workers in all 50 states. They run an experimental network, gathering attack information to show their students what to look for and what to do when it happens. Retired Sgt. Bill Scherr, a senior instructor with the Guard's Electronic Warfare Associates team, says they've harvested information about attackers from all over the world that has offered valuable lessons to students who may be defending military networks from hacker attacks. (Gaudin, 2002) In the corporate world, honeypots can also be a valuable testing environment for prospective employees. Setting up a honeypot with known sets of vulnerabilities then giving each candidate the opportunity to find one or more of the weaknesses and take control of the system, may help identify those who do not have the ability to ward off attacks to a production system.

Legal Issues

Various legal issues have been connected to the use of honeypot systems. One of those is entrapment. The issue of entrapment could be relevant if the attacker is intentionally lured by law enforcement agencies to a honeypot for the sole purpose of identifying him for prosecution. To mitigate this possibility, there should be no implied permission to access the honeypot system. Banners should be carefully stated and identical on both production and honeypot systems. Without these measures, an entrapment legal defense may nullify the prosecution of attackers.

Privacy has also been raised as a possible legal issue with honeypots. Some say intercepting communication and viewing files on a honeypot or relayed by a honeypot is a violation of right to privacy laws. Case law exists pertaining to a loss of the right of privacy for files stored on a stolen or compromised computer without the owner's permission, but there is little to no case law on the loss of privacy when communications are intercepted and relayed through a compromised computer.

Though seemingly farfetched, organizations using honeypots can find themselves suffering the unintended consequence of loss of reputation, falling stock prices, or both if one of their compromised honeypot systems contains false data to lure attackers and that data is made publicly available by the attacker at cracker websites or chat rooms. Additionally, a honeypot system that allows outbound traffic that is used to attack other systems, may make the owner of the honeypot liable for lacking due diligence in protecting corporate assets or even gross negligence because of a hazard that was deliberately set up and not properly supervised. (Scottberg, Yurcik, Doss, 2002)

The creators of software used on a honeypot are also not immune from the possibility of being in legal "hot water" for the actions of their products. One example is the creator of LaBrea tar pit, Tom Liston. Tom Liston first developed his honeypot product in 2001 and placed it free for the taking on his Hackbusters.net web site under the General Public License (GPL). In April of 2003, he first became aware that distributing LaBrea from his site may place him in violation of the Illinois Criminal Code. This code "makes it illegal to create a device capable of disrupting a communication service without the express consent or express authorization of the communication service provider". It further "makes it a crime to conceal the existence, origin, or destination of any communication from a service provider or any lawful party". Since LaBrea both disrupts communications and conceals the true origin of communication, Liston has halted the distribution of LaBrea from his web site rather than risk prosecution. (Hulme, 2003)

Honeypots are not restricted to use on the Internet. Some honeypots are designed for use inside a network to handle attacks by disgruntled employees and other malicious users with legitimate network access. These honeypots are intended to do more than track attacker methodologies. They also provide the forensic evidence needed by law enforcement officials to apprehend and prosecute intruders. “Honeypot logs can provide good forensic evidence for prosecution, but the logs should be recorded to a nonmodifiable media such as CD-Recording to prove that the evidence was not tampered with. Also, a cryptographic seal around the logged files makes for a stringer forensic trail.” (Schwartau, 2000) It may be prudent to consult with the human resources department and legal counsel with cyber-knowledge before using honeypots internally and also for advice on methods of collecting forensic information that is admissible in a court of law.

Advantages and Disadvantages

The advantages of the honeypot as a security tool are many. The data they collect is normally of high value. Because they are not a production system that sends or receives production data, the data they do collect is likely specific to a probe, scan, or even an attack. Compared to other IDS, the amount of data they collect is small making it easy and cost effective to identify and act on unauthorized activity. Limited traffic also reduces the incidence of missed attacks that can occur when an IDS is unable to keep up with network traffic, dropping packets and potentially dropping indicators of an attack.

Honeypots can function with only minimal hardware and management resources even on large networks. According to Lance Spitzner, a single Pentium computer with 128MB of RAM can be used to monitor millions of IP addresses. This feature alone gives honeypots “more bang for the buck” than any other IDS.

Just as more and more organizations are encrypting their data because of security and privacy issues, more hackers are also using encryption. Unlike other IDS, honeypots function well in encrypted environments. For example, they can log the key strokes of an interactive session even if encryption is used to protect the network traffic.

Honeypots are superior to other IDS in their ability to detect vulnerabilities that are not yet known or understood. Other IDS use patterns or signatures of known attacks to detect malicious traffic. Because of this, they often fail to detect compromises that were unknown at the time they were deployed or raise large numbers of false alarms when unable to distinguish between legitimate and malicious traffic.

While proponents of the honeypot system laud their successes, it is important to recognize its potential shortcomings. First of all, many hackers are technology experts who constantly develop cutting edge tools to allow them to stay one step ahead of their pursuers. An advanced hacker could compromise a honeypot leaving it open to attack or allowing it to be used to attack another system. Awareness by a hacker of an organization's efforts to set honeypot traps may rally the “hacker community” against the organization making them a popular hacker target.

Honeypots have a very restricted field of view. They are only aware of activity directed to them. If a hacker attacks a production server, a honeypot will not sound an alert or capture any of the activity. Therefore, honeypots should only augment, not replace other IDS watching production systems.

Additionally**,** honeypots, to be used effectively, may require more skill, administrative effort, and funding than many organizations have or choose to spend to secure their systems. According to Elias Levy, chief technology officer at Securityfocus.com, “They won't fulfill their promise unless you have the time to administer them correctly”. Levy believes that companies concerned about security threats are “better off using an intrusion-detection system” if they don't have a dedicated team of highly trained administrators. (Johnson, 2000)

Conclusion

In the world of technology, not all the players are “good guys”. Today, efforts to identify the “bad guys” are moving away from the old-fashioned static intrusion detection systems and toward the study of the dynamic behavior of electronic visitors to separate the good guys from the bad. According to Lance Spitzner, “Everything in security has been about prevention, protection and reaction. The bad guys have the initiative. We want to take the initiative and gain intelligence in the bad guys and counter before they attack.” (Fordahl, 2001) While not a total solution, honeypots are an effective tool in this effort. How this tool is implemented is up to the user and should depend on the intended objective.

To effectively protect a computer against unauthorized access, it is important for a computer owner and company IT professionals charged with securing company assets to be armed with as many defense mechanisms as possible. Although honeypots are not intended to replace other traditional security systems, they seem to have proven to add value to the security efforts of an organization and, therefore, are an important part of information security.   However, as with any other technology, system administrators are well advised to do their homework before bringing anything into their environment that has the potential to do as much harm as good.

List of References

Choudhury, Tareque. “Honeypots: the trap is Set” 15 April, 2003.
http://www.itsecurity.com/papers/cyberguard1.htm (22 November, 2003).

Even, Loras. "What is a Honeypot?" 12 July 2000.
http://www.sans.org/resources/idfaq/honeypot3.php (26 November, 2003).

Fordahl, Matthew. “Honeynet Project: Security gurus study hackers” 29 July, 2001
http://www.newstribune.com/stories/072901/wor_0729010050.asp (2 December, 2003).

Gaudin, Sharon. “Honeypots Turn the Tables on Hackers” 30 July, 2002
http://itmanagement.earthweb.com/secu/article.php/1436291 (22 November, 2003).

Hulme, George V. “Security Developer snared in Legal tar Pit” 23 April, 2003.
http://cert.uni-stuttgart.de/archive/isn/2003/04/msg00102.html (28 November, 2003).

Johnson, Keith. "Hackers caught in security 'honeypot'" 19 December 2000 :
http://www.zdnet.com/zdnn/stories/news/0,4586,2666273,00.html (22 November, 2003).

Liston, Tom. “Hack Busters” 16 April 2003
http://www.hackbusters.net/ (28 November, 2003).

Merkow, Mark. “Playing with Fire: Not So Sweet Honeypots” 12 January, 2001. L:
http://ecommerce.internet.com/news/insights/outlook/article/0,,7761_559431,00.html
(22 November, 2003).

Messmer, Ellen. “‘Decoy nets' gain backers in battle against hackers" 3 May 2001.
http://www.nwfusion.com/news/2001/0305honeypot.html (26 November, 2003).

Provos, Niels. “A Virtual Honeypot Framework”
http://www.citi.umich.edu/techreports/reports/citi-tr-03-1.pdf (28 November, 2003).

Raikow, David. "Building your own honeypot” 22 November, 2000.
http://www.zdnetindia.com/techzone/resources/security/stories/7602.html
(2 December, 2003).

Ranum, Marcus J. “Hacker Tar Pit” September 2002.
http://infosecuritymag.techtarget.com/2002/sep/cooltools.shtml (2 December, 2003).

Schwartau, Winn. "Lying to hackers is okay by me" 7 June 1999. :
http://www.nwfusion.com/newsletters/sec/0705sec2.html?nf (2 December, 2003).

Schwartau, Winn. "Honeypots wreak sweet revenge against cyber intruders"
4 December 2000.
http://www.nwfusion.com/columnists/2000/00173866.html (22 November, 2003).

Schwartz, Mathew. “Networks use ‘honeypots’ to catch online thief” 4 April 2001.
http://www.cnn.com/2001/TECH/internet/04/04/trap.a.thief.idg/ (22 November, 2003).

Scottberg, Yurcik, Doss. “Internet Honeypots: Protection or Entrapment?” June 2002.
http://www.sosresearch.org/publications/ISTAS02honeypots.PDF (2 December, 2003).

Symantec Decoy Server
http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=157&EID=0
(22 November, 2003).

Symantec “Honeypots Have Eyes on the Enterprise” 29 July, 2003.
http://enterprisesecurity.symsntec.com/article.cfm?articleid=2371&EID=0
(22 November, 2003).

Spitzner, Lance. "To build a Honeypot" 4 August, 1999.
http://www.sdconsult.no/linux/info/honeypot/honeypot.html (28 November, 2003).

Spitzner, Lance. “Honeypots – Definitions and Value of Honeypots” 10 December 2002.
http://secinf.net/honeypots/Honeypots_Definitions_and_Value_of_Honeypots.html
(28November, 2003).

Spitzner, Lance. “Honeypots Definitions and Value of Honeypots” 29 May 2003. http://www.tracking-hackers.com/papers/honeypots.html (28 November, 2003).