Honeypots And Honeynets Are Not Just For Bears
GSEC Practical Version 1.4
Lawrence Lauer
July 8, 2002

Introduction

With the ever increasing threat of system compromise and blackhat activity wouldn't it be nice to know what the blackhat community is doing to our systems? This paper will try and give the reader a broad overview about honeypots and honeynets.

Honeynets can be an invaluable source of information for the whitehats arsenal against the blackhat community on the information battlefield where hackers, system crackers and the like are waging war on our systems and networks. Let's find out how they are doing it.

A little history lesson

Honeypots are not a new technology they have been around for quite some time. They may not have been called honeypots at first. The term honeypot came along later. There are a couple of really good papers on the subject when honeypots where a new concept: Cliff Stoll's "Cuckoo's Egg"[1], and Steve Bellovin and Bill Cheswick's "An evening with Berferd."[2] One of the best sources of information I have found on the subject of Honeynets is the "Knowing Your Enemy" whitepapers, and the book "Knowing your Enemy"[3] by The Honeynet Project.

The Honeynet Project[4] was founded by Lance Spitzner around April 1999. The Honeynet Project consists of 30 security professionals all specializing in different areas. All of these professionals volunteer their time to better the project. You can read more about the research of these members at http://project.honeynet.org.

What are Honeypots and Honeynets?

A Honeypot in general is a single system plugged into a network that is not fully secured. Many times this will be a default installation of an operating system with no extra efforts to secure it. This system may emulate or actually run specific services or applications that are insecure.

[1] Stoll, Cliff "The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage" (New York: Pocket books, 1990)
[2] Bellovin, Steve and Cheswick, Bill "An Evening with Berferd"
URL: http://www.securityfocus.com/data/library/berferd.ps
[3] Honeynet Project, The "Know Your Enemy : Revealing the security tools, Tactics, and motives of the blackhat community" Addison-Wesley 2002
[4] Honeynet Project, The URL: http://www.honeynet.org

There are a few different options available for software based honeypots one choice would be the Deception Tool Kit (DTK)[5]. The DTK is a collection of scripts that emulate known vulnerabilities. The DTK is a great way to learn about current system vulnerabilities but, if a new attack comes along you may not catch it since you are limited to your current scripts. However new scripts are being released.

There are also a few commercial honeypot products out there. One option would be “Mantrap” by Recourse Technologies which can provide various types of systems and hardware in a jail type environment. This means that you could possibly install Linux, Windows 2000 and Cisco IOS images on the honeypot to allow an intruder to interact with a real operating system or simulated piece of hardware.

The value of a honeypot lies in being probed, prodded, scanned, exploited and eventually compromised. Once an intruder has compromised the honeypot it can be used for learning what risks there might be to your environment. You can learn a lot from watching an intruder chatting with others in IRC (Internet Relay Chat) and capturing keystrokes on your honeypots.

The learning begins when a potential intruder first contacts your system. There is a lot to be learned from watching what types of attacks are coming in from the outside and watching what types of probes are being conducted against your systems.

The honeypot is used to mitigate the risk to your networks by providing you with information about what the black hats are doing to your network. You can see what exploits are being executed, and what types of commands a blackhat might run after he controls the system. The most valuable information comes after the intruder has compromised the system. In essence it gives you a head start to harden your production network before an attack happens on your production network. Remember that it’s not if an attack will happen but when. Besides learning about the risks you also learn about what techniques are being used and how skilled the attackers are that are trying to penetrate your defenses.

Honeynets are different from honeypots in that they are not single system but, networks comprised of multiple honeypots. Honeynets are generally used for research in gathering intelligence on what the black hats are doing and how they are doing it. Most of the time research honeynets are implemented by research firms and colleges.

[5] DTK “The Deception Toolkit” URL: http://www.all.net/dtk/

Honeynet Setup

The basic setup of a Honeynet consists of a Firewall, Router, Intrusion Detection System (IDS), Syslog server, log/alert server and the honeypots. Your results may vary. Depending on your network design this is only limited by one's own imagination. There are many different options for network configuration this is just one.

The Firewall is your primary device for data control and first line of defense. Data Control is essential to not allow compromised systems on the Honeynet to be launch pads for future attacks on other networks. The basic firewall setup would be to allow all inbound traffic from the untrusted network, the internet, to allow connections to the Honeynet but not to the administrative network, this is accomplished by firewall rules.

We also want to limit connections outbound from the Honeynet to a fairly low number to allow an intruder to ftp (File Transfer Protocol) a root kit or initiate Internet Relay Chat (IRC) sessions. The higher the number of outbound connections allowed increases the risk substantially that the Honeynet will be used to attack another site. (The rule setup will vary depending on the type of firewall you are using.) You can look at a sample iptables firewall setup at: http://project.honeynet.org/papers/honeynet/rc.firewall

Iptables is the Linux firewall subsystem. It provides packet filtering, many different kinds of NAT (Network Address Translation) and packet mangling. You can learn more at the netfilter website http://netfilter.samba.org/.

The Router is placed in between the Honeynet and the firewall to mask the identity of the firewall. The reason behind this is that once a system intruder has compromised one of the honeypots they will see the router instead of the firewall and not alert the intruder that they had come in through a firewall and prevent the intruder from attacking the firewall from within. The router also provides logs of connections that can be used for analysis.

The Intrusion Detection System (IDS)[6] is used for data capture and alerting if something suspicious is happening. The IDS however is not fool proof there are a few instances where an IDS system will not detect an attack but since it is capturing all of the data passed on the network we will get a detailed log of what is happening.

The IDS that I have used is snort www.snort.org. Snort is a lightweight full blown intrusion detection system capable of performing real-time traffic analysis and packet logging on IP networks.

[6] Snort a free open source IDS runs on Win32 and various flavors of Unix/Linux.
URL: http://www.snort.org

The following is a sample configuration file. (Note that all the comments have been removed for the sake of space.) You can learn more at www.snort.org.

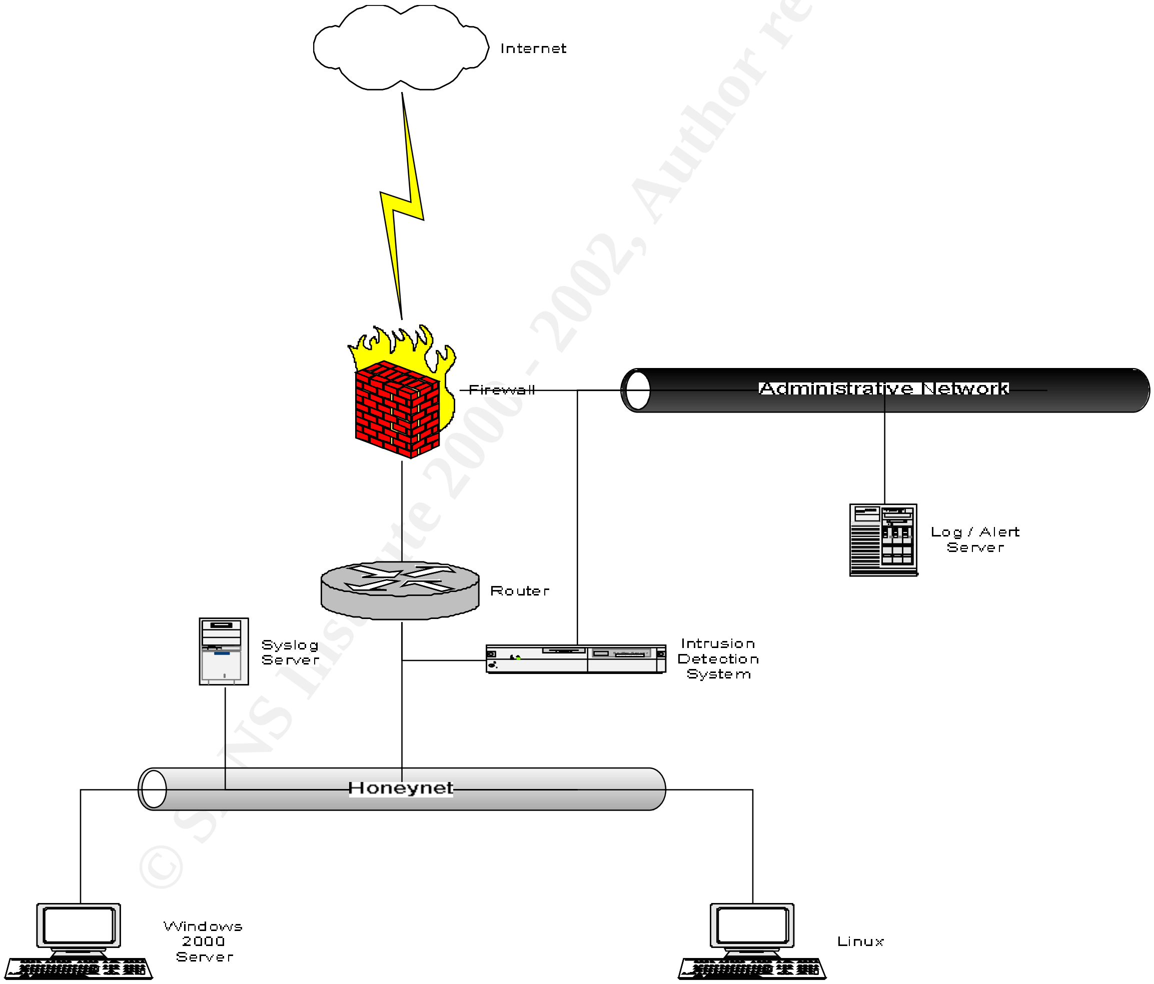
```
#--------------------------------------------------
#   http://www.snort.org     Snort 1.8.1 Ruleset
#     Contact: snort-sigs@lists.sourceforge.net
#--------------------------------------------------
# NOTE:This ruleset only works for 1.8.0 and later
#--------------------------------------------------
# $Id: snort.conf,v 1.77.2.7 2002/03/02 05:33:01 cazz Exp $
###################################################
var HOME_NET 192.168.1.0/24
var EXTERNAL_NET any
var SMTP $HOME_NET
var HTTP_SERVERS $HOME_NET
var SQL_SERVERS $HOME_NET
var DNS_SERVERS $HOME_NET
var RULE_PATH /root/snort/snort-1.8.4/
preprocessor http_decode: 80 -unicode -cginull
preprocessor bo: -nobrute
preprocessor telnet_decode
preprocessor portscan: $HOME_NET 4 3 portscan.log
include $RULE_PATH/classification.config
include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/smtp.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/tftp.rules
include $RULE_PATH/web-cgi.rules
include $RULE_PATH/web-coldfusion.rules
include $RULE_PATH/web-iis.rules
include $RULE_PATH/web-frontpage.rules
include $RULE_PATH/web-misc.rules
include $RULE_PATH/web-attacks.rules
include $RULE_PATH/sql.rules
include $RULE_PATH/x11.rules
include $RULE_PATH/icmp.rules
include $RULE_PATH/netbios.rules
include $RULE_PATH/misc.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/shellcode.rules
include $RULE_PATH/policy.rules
include $RULE_PATH/porn.rules
include $RULE_PATH/info.rules
include $RULE_PATH/icmp-info.rules
include $RULE_PATH/virus.rules
#include $RULE_PATH/experimental.rules
include $RULE_PATH/local.rules
```

The syslog server is a system on the Honeynet that accepts all logging from all machines and devices on the Honeynet. This device should be the most secured system on the Honeynet. As stated in the book, ”Know Your Enemy” by Lance Spitzner, Page 39 “The syslog server is also a sophisticated honeypot and as such should be one of the most heavily secured systems on the Honeynet.”

The log/alert server is a machine that resides on the administrative network to accept logs and alerts from the IDS and Firewall.

Below is an example of a basic honeynet design. The design of the honeynet can be as simple as this one or as complicated as one sees fit. For this paper the design below is basic but very functional for learning about attacks and what black hats are doing. This is also a good starting network to build upon. This network consists of all the necessary components to make up a honeynet. One of the reasons I like this design is that it is small, can be setup with minimal hardware requirements, can be rebuilt fairly quickly in the event of a malicious hacker destroying your honeynet and it is expandable. The honeynet example network below is similar in design to the one described in the book "Know Your Enemy" by the Honeynet Project figure 2-2 page 13.

.



Knock knock who's there

Now that we have laid out how our network is setup we should spend a little time on whom and what the threat is. Once you have your honeypots or honeynets setup you will hopefully see many of the risks that are at your production network door everyday.

It seems that there are two major factions within the blackhat community, script kiddies and more advanced hackers. Of course there are many others in other parts of the spectrum that may not fit these two definitions.

The script kiddies are defined by webopedia as:

> "A person, normally someone who is not technologically sophisticated, who randomly seeks out a specific weakness over the Internet in order to gain root access to a system without really understanding what it is s/he is exploiting because the weakness was discovered by someone else. A script kiddie is not looking to target specific information or a specific company but rather uses knowledge of a vulnerability to scan the entire Internet for a victim that possesses that vulnerability."

There are tools and scripts popping up on a daily basis to make a black hats job a whole lot easier. Some tools are as easy as clicking go and will begin to probe a network for listening systems to develop a database of IP addresses. Once, a list of IP addresses has been created a system can be searched for with a known bug that can be easily exploited. In some cases this can all be done while the blackhat sleeps and in the morning they have a list of machines that they currently own. (Have root/super user access to.)

You can find out more information about script kiddies in the "Know Your Enemy" whitepapers on http://www.honeynet.org.

There are of course other more advanced hackers out there that will install trojans and backdoors to allow undetected future access. Since we are watching all network activity there will be no undetected access.

Once an intruder has compromised a system it is not uncommon for them to not revisit the system for a couple of days or maybe even weeks. It could be because they are busy taking over other systems or they are just letting the heat die down before they come back to wreak havoc but, we will be waiting and ready when they do.

Analysis

An average day of logs on a production network could take a system administrator all day or longer (a month of Sundays comes to mind) to sift through and find out what is suspicious. On the Honeynet it is easy to find out what are suspicions since all data going to and from the Honeynet is suspicious in nature since there is no production work being done on this network.

It is very important that you have logs that you can trust that is why you should keep logs on the local system and in a remote secured location. Once a blackhat has compromised the system one of the first things they will do is try and hide all traces that they have been there. In the configuration we have talked about we have logs in three spots:

1. The local system
2. Remote syslog
3. The IDS

These are capturing all log data. So even if the blackhat wipes out the system we will have captured the logs in other locations.

A good tool to watch your log files is swatch[7i] it will watch your logs for anything abnormal and can send the system administrator an email. I found this tool useful on a number of Linux systems. Swatch is a very simple tool that can be setup to alert you in a number of different ways from email to pager if an instance matching your rules happens.

Now that you have secured your logs and configured swatch you may be wondering what to do about all the captured data that you are getting. Since this could be a whole paper on its own I will not try to reinvent the wheel, I will direct you to some good places to get you going in the right direction.

The Honeynet project[8ii] whitepapers contain good information on passive fingerprinting, data analysis and forensic data analysis. Passive fingerprinting is the art of learning more about an attacker without being detected in the process. Passive fingerprinting uses sniffer traces gathered by a remote system. Fingerprinting is used to find out what operating system, software that is being used and services being used on a remote system.

Another type of fingerprinting is active fingerprinting using tools such as Fyodor's Nmap the stealth port scanner. More information regarding Nmap can be found at http://www.insecure.org/nmap/index.html. One more thing about tools such as Nmap is that since you are using a tool to learn about an attacker's machine they may notice that they are being scanned and become spooked. If this

[7] Swatch The simple watcher E. Atkins, Todd URL: http://www.oit.ucsb.edu/~eta/swatch/

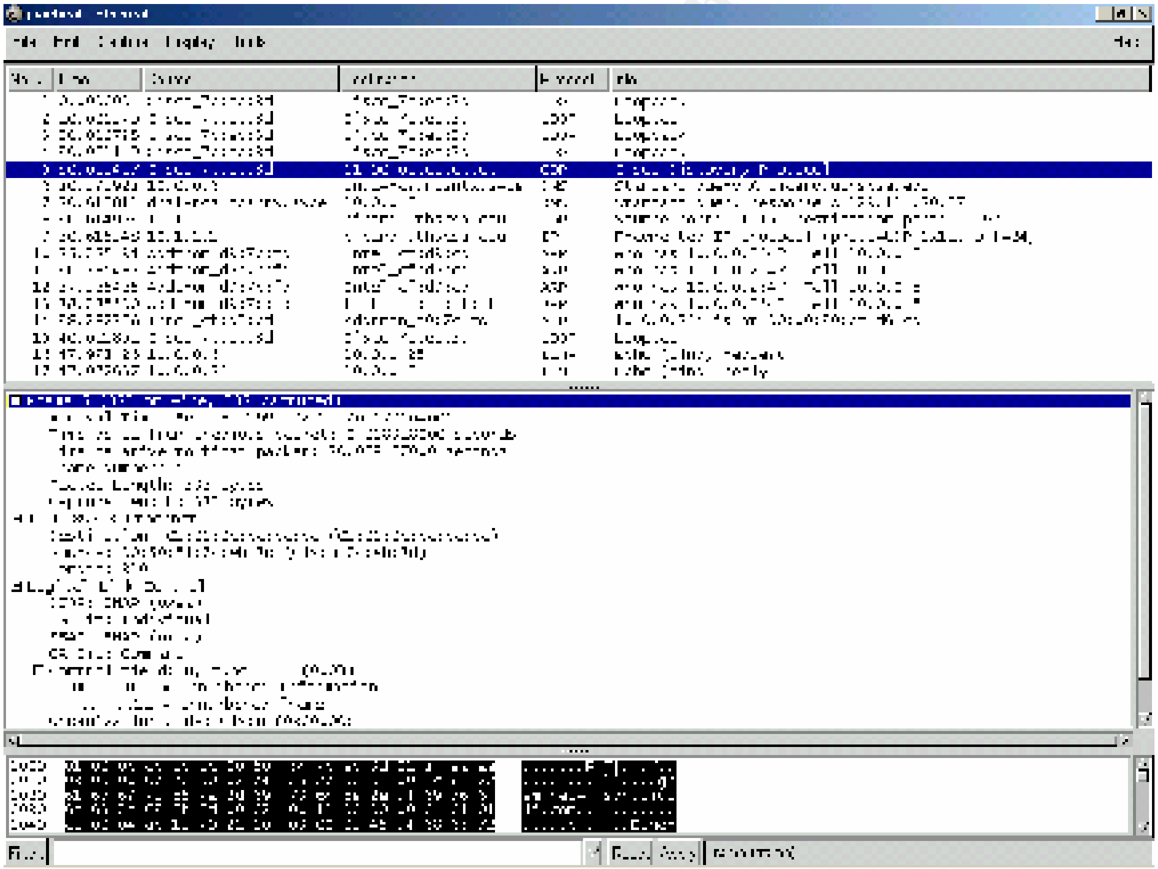
[8] Honeynet Project, The "Security Papers" URL: http://project.honeynet.org/papers/

happens you may clue an attacker in that he/she has been found out and either do something drastic like delete your system or possibly just leave and not come back. If they leave and do not return the value of learning about black hats has been lost and your Honeynet is compromised.

Forensics is the process of figuring out what has happened to a system after it has been compromised. There is so much information available on forensics. One source of information is by Dave Dittrich and can be found at http://staff.washington.edu/dittrich/misc/forensics/.

A good knowledge of TCP/IP would be very beneficial to understand what the packet captures are telling you. Deciphering such captures is critical to find out what types of attacks are being launched.

Ethereal[9] is a great free packet sniffer that rivals many commercial products. The following is a sample packet capture with ethereal.



There are many sites out on the internet that will give you a good understanding of TCP/IP one site that has some good links and book references is: http://networking.ittoolbox.com/nav/t.asp?t=444&p=454&h1=444&h2=454 .

[9] Ethereal URL: http://www.ethereal.com/

A tool that I have found useful in identifying what has been changed on a system has been Tripwire[10][iii] to figure out what has changed on a Linux system. Tripwire basically takes a snapshot of your system in its current state and saves the information to a 1028-bit encrypted algorithm protected log file. Tripwire also comes in NT, 2000 and XP flavors.

For NT you can use sysdiff[11][iv] to take an initial snapshot of the system and then once you think something has changed you can run the sysdiff command with the /diff to help you figure out what has changed.

Be sure to secure your image files since if you leave them available for the blackhat, you may not have anything to come back to or it may not be something that you want to come back to. One possibility would be to copy the image files off to the log/alert server on the administrative network for future use.

There are many IDS systems available and more are coming out all of the time and the types of alerts and logs they produce will vary slightly. The IDS system that has seemed to produce very good results has been Snort[12][v] by Marty Roesch. There is a wealth of information on setup and configuration of Snort on the snort website www.snort.org.

There are also many different plug-ins that is useful with Snort one is ACID (Analysis Console for Intrusion Databases)[13][vi] that will display a nice PHP webpage for you to view what snort is squealing about.

A few common snort alerts are:

```
[**] IDS198/SYN FIN Scan [**]
07/4-01:50:45.254726 192.168.1.8:53 -> 192.168.0.1:53
TCP TTL:23 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x4D622A79  Ack: 0x7EEF29AF  Win: 0x404  TcpLen: 20

07/5-19:36:23.468056
[**] spp_portscan: PORTSCAN DETECTED from 192.168.2.8 (THRESHOLD 3
connections exceeded in 4 seconds) [**]

07/5-19:36:39.561360
```

[10] Tripwire is available in different flavors at URL: http://www.tripwire.com also another version is the open source version of Tripwire that ships with some of the different distributions of Linux is available at URL: http://sourceforge.net/projects/tripwire

[11] Sysdiff Grigsby, Lee "Advanced Sysdiff"
URL Wraps:
http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/prodtechnol/winntas/deploy/advsysdf.asp

[12] Snort "The Open Source Intrusion Detection System" URL: http://www.snort.org

[13] Acid "Analysis Console for Intrusion Databases" URL: http://acidlab.sourceforge.net/

```
[**] spp_portscan: portscan status from 192.168.1.11: 5 connections
across 1
hosts: TCP(0), UDP(5) [**]
```

Also while working with Snort there is another tool that you will find invaluable it is arachNIDS by Max Vision (**a**dvanced **r**eference **a**rchive of **c**urrent **h**euristics for **n**etwork **i**ntrusion **d**etection **s**ystems)[14][vii] this is a database that contains attack signatures that snort will produce IDS#'s, event numbers that you can reference in arachNIDS to get more information on what is happening on your Honeynet. An example of an IDS alert might look like IDS126/X11_OUTGOING_XTERM[15].

Some other tools that may be of help while working with analyzing a compromised system is TCT (The Coroners Toolkit) it is available at http://www.porcupine.org/forensics/tct.html . These tools are mainly used on compromised Unix systems. One notable tool in TCT is grave-robber it is used to collect data from a compromised system like MD5 (Message Digest number 5) checksums, critical log files and MAC times. MAC times stands for Modify/Access/Change. If you are using tripwire, it creates a MD5 database so if you use grave-robber to collect data from a compromised system you can use the tripwire image against the grave-robber collected MD5 checksums to find out what files have changed. You can also gain a lot of information from the collected log files.

By reviewing MAC times it would be possible to nail down when critical systems files where changed.

Legal Information

There are many different opinions out there on what the legal ramifications would be of setting up a honeypot/honeynet. A major issue that everyone seems to be talking about is entrapment. One definition of entrapment as www.thelawyerpages.com states:

> "A defense in a criminal action asserting that the defendant was tricked or coerced to perform a criminal act by law enforcement officers, and that but for such trickery or coercion, the defendant would not have committed the act."

In our situation with the Honeynet it seems to me that it would be terribly difficult for a blackhat to claim entrapment since no one told them to scan for our systems and launch an attack. At most they may have been talked into doing something illegal by one of their so called friends. I suppose that it could be

[14] arachNIDS "**a**dvanced **r**eference **a**rchive of **c**urrent **h**euristics for **n**etwork **i**ntrusion **d**etection **s**ystems" URL: http://www.whitehats.com/ids/

[15] More information on this alert can be found at the URL: http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids126&view=event

entrapment by your peers. Or it could be the black hats curiosity got the best of him and had to check it out. We all know that curiosity killed the cat.

There is always the possibility that someone is forcing someone to do something they do not want to do, then that would be something else, and I will not get into that here.

The other issue that has been brought up quite often is the invasion of privacy. The Supreme Court has defined privacy as the right of the individual to control the dissemination of information about oneself. This could get kind of tricky since we are capturing everything from IRC sessions, FTP sessions, email and any other type of communication coming in or leaving the Honeynet.

There are two federal statutes that pertain to privacy. The Electronic Communication Privacy Act (18 U.S.C 2701)[16][viii] and the Interception and Disclosure of Wire, Oral, or Electronic Communications (18 U.S.C. 2511)[17][ix] the locations of the definitions are located in the endnotes.

The Honeynet project is currently working on trying to find out the legal issues involved with honeynets but at the time of this paper I was unable to find any finished documents. To be safe and cover your self be sure to consult your own legal counsel prior to moving forward with any actions against any intruders.

Recommendations

I would suggest if you are interested in learning more about Honeypots and Honeynets install a basic Linux system which can run on just about anything that you may have lying around holding the door open and take a look at honeyd[18][x] a really powerful open source Honeypot. Or if you don't have that extra system laying around and still would like to start checking things out take a look at BOF (Back Officer Friendly) you can download a free copy for personal use at http://www.nfr.com/products/bof/. This program will give you a basic idea behind honeypots and you can see what kind of things you can do with a honeypot.

I hope that there was some useful information is this paper and that it will excite some people to go out and take a look at honeypots and honeynets. The more people learn about what the capabilities of the black hats are, the more secure our systems will be. We want to make black hats think very carefully before they make a move to attack a system because we are watching.

[i] Swatch The simple watcher E. Atkins, Todd URL: http://www.oit.ucsb.edu/~eta/swatch/

[16] 18 U.S.C. 2701 URL http://www.cybercrime.gov/usc2701.htm

[17] 18 U.S.C. 2511 URL http://www.cybercrime.gov/usc2511.htm

[18] Honeyd URL: http://citi.umich.edu/u/provos/honeyd

[ii] Honeynet Project, The "Security Papers" URL: http://project.honeynet.org/papers/

[iii] Tripwire is available in different flavors at URL: http://www.tripwire.com also another version is the open source version of Tripwire that ships with some of the different distributions of Linux is available at URL: http://sourceforge.net/projects/tripwire

[iv] Sysdiff Grigsby, Lee "Advanced Sysdiff"
URL Wraps:
http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/prodtechnol/winntas/deploy/advsysdf.asp

[v] Snort "The Open Source Intrusion Detection System" URL: http://www.snort.org

[vi] Acid "Analysis Console for Intrusion Databases" URL: http://acidlab.sourceforge.net/

[vii] arachNIDS "**a**dvanced **r**eference **a**rchive of **c**urrent **h**euristics for **n**etwork **i**ntrusion **d**etection **s**ystems" URL: http://www.whitehats.com/ids/

[viii] 18 U.S.C. 2701 URL http://www.cybercrime.gov/usc2701.htm

[ix] 18 U.S.C. 2511 URL http://www.cybercrime.gov/usc2511.htm

[x] Honeyd URL: http://citi.umich.edu/u/provos/honeyd