HONEYPOTS - THE EASY WAY

Don't you just love Information Security, there's all these cool sounding words and we get to do all this cool stuff. There's all this trying to stay ahead of the game, trying to make sure you're running a tight network. And very few tasks are quite as "James Bond-ish" as a honeypot. The simplest definition of a honeypot is that it is a fake server or network resource which you allow attackers to attack, the honeypot will keep a track of what the attacks looked like and where they came from, but since it is a fake resource, it cannot be compromised. Now there are some very nice tools/projects out there that specialize in this type of thing (the honeynet project being one), but I want to go through a quick and easy way to setup a honeypot on your network.

First Things First
This is aimed at users of linux servers but I am sure that it can be adapted for most systems including windows, as the main tool is available on almost all platforms. The main part of this quick and easy honeypot involves *netcat* - so if you do not have it, you will need to get it before carrying on. Don't worry, it is a small download and can be used for many different things besides a honeypot, so it is well worth the time.

Setting the Pot
Lets setup our server to run this script, we will need a */admin/honeypot* directory first, and inside there create an executable *hpot* script file, and enter the following honeypot code..

```
1   PRT="80"
2   NC=`which netcat`
3   BSE=/admin/honeypot
4   LOG=hpot.log
5   BANNER="MS-IIS WEB SERVER 5.0\r"
6
7   touch /tmp/hpot.hld
8   echo "" > $BSE/$LOG
9
10  while [ -f /tmp/hpot.hld ]
11   do
12    echo -e $BANNER | $NC -l -v -n -p $PRT 1>> $BSE/$LOG 2>> $BSE/$LOG
13    echo "==ATTEMPTED CONNECTION TO PORT $PRT AT `date`==" >> $BSE/$LOG
14    echo "" >> $BSE/$LOG
15    echo "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" >> $BSE/$LOG
16   done
```

Some things to note...

Line Number	Comment
1	This is the variable where you set the port on which you want the honeypot to listen
4	This is where the honeypot writes its logs
5	This is the banner which you want the attackers to see when they connect to the honeypot
7	This is the lock file. Delete it if you want the honeypot to stop running
12	This is where netcat is setup as the honeypot, both stdout and stderr are redirected to the log file

Checking The Pot
Now I ran my honeypot script on port 80, and then ran a MS IIS exploit against the honeypot (simulating an attacker who is about 3 years behind the times). This is what the attacking machine saw..

```
[root@localhost exploit]# ./execiis.sh 192.168.10.13
using netcat:/usr/local/bin/netcat
MS-IIS WEB SERVER 5.0
[root@localhost exploit]#
```

And this is the log created by the honeypot...

```
[root@syplh honeypot]# cat ./hpot.log

Connection from 192.168.10.80:53152
GET /scripts/..%255c..%255c..%255c..%255c..%255c..%255c..%255c
..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\ HTTP/1.0

==ATTEMPTED CONNECTION TO PORT 80 AT Fri Nov 26 19:05:52 SAST 2004==

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

As you can see, we now know where the attacker came from, what exploit method was used and the time the exploit was attempted. Just the thing to show management to prove you're needed.

Why Set a Honeypot
Being totally serious, there is a reason for honeypots besides the coolness factor. You see, there are many exploits out there which are not public and therefore are not in any firewall rules, patches or IDS rules. Knowing how an attacker is trying to exploit your systems can help uncover if it is a new attack, and if it is, it will help you better protect against it as you will know how to fine-tune and update your firewall and IDS.

Some More Tricks
There are some more tricks you can do using this simple script, here are some ideas..

- You may want to run more then one honeypot on a machine, so change the *PRT="80*" line to read *PRT="$1"*. This way the port number the honeypot will fake will be able to be specified when you call the honeypot script, like this: */admin/honeypot/hpot 25*. Just make sure that each honeypot instance uses a different log file, to do this change the *LOG=hpot.log* to read *LOG=hpot.$1.log*.
- You may not want to run the honeypot on your firewall or on the actual port on the honeypot machine. No problem. Lets say you wanted to honeypot port 22, all you do is run the honeypot script on any port - say 333. Then use destination nat to change the port. If you are using iptables, the rule could look something like;
  iptables -t nat -A PREROUTING -d <honeypot ip> -p tcp --dport 22 -j DNAT --to-dest <honeypot ip>:333

Anyway, thats the end of this tutorial, I hope you found it useful and more importantly, I hope you will have fun playing around and learning with the honeypot script.