A Whirlwind Introduction to Honeypots

Marcus J. Ranum
<*fishlips@ranum.com*>

What is a honeypot?

- A security resource thats value lies in being attacked, probed, or compromised
  - A honeypot is more a *state of mind* than a specific implementation
  - You can set up a production system to be a honeypot or you can laboriously construct your own dedicated honeypot system

Cool Record:

- Shortest time in which a honeypot was attacked and compromised:
  - 17 seconds from connection to the Internet (by a worm)

Two Kinds of Honeypots

- Research
- Production

Research Honeypots

- Research honeypots are for the hardcore hacker hunters
  - Learn what the bad guys are doing
  - Study their methods
  - Capture their keystrokes
  - Capture their tools
  - Monitor their conversations
- Takes a ***lot*** of work!!

Production Honeypots

- Production honeypots are more in line with conventional intrusion detection
  - Identify hostile activity
  - Generate an alert
  - Capture a minimum of data
- Takes a ***minimum*** amount of work!

Interaction

- Low Interaction
- Middle Interaction (chroot/jail)
- High Interaction

Low Interaction Honeypots

- Present the Bad Guy with emulators of vulnerable programs
  - Summarize or capture limited interactions
  - Simpler to deploy (no system administration)
  - Less likely to be penetrated
  - More likely to be detected by the Bad Guy
- Tend to be production honeypots

Middle Interaction Honeypots

- Tend to be application-centric chroot/jail systems
  - It is hard to set up a reliable chroot/jail
  - Limited usefulness: no chroot for Windows
  - Requires exhaustive specialized knowledge (chroot semantics vary between versions of UNIX)
  - Really more of an operational / application security process than a honeypot
- Too much work for most people

High Interaction Honeypots

- Present the Bad Guy with a complete operational environment that you assume will be penetrated completely
  - Monitor everything they do
  - Install data control to reduce likelihood of outgoing attacks
  - Collect their tools and keystrokes
- Tend to be research honeypots

Risk

- What are we afraid of?
  - Primary fear - someone uses our honeypot as a jumping-off point for an attack against someone and does them harm
    - E.g.: a distributed denial of service attack against cnn.com that came from us!
  - Secondary fear - someone uses our honeypot to attack our own systems

Mitigating Risk

- Risk is greater with high interaction honeypots
  - Must use traffic control (e.g.: firewall filtering etc - see the honeynet rules on their site) to prevent jump-off attacks
- Risk still somewhat present but largely eliminated in low interaction honeypots

Fingerprinting

- Fingerprinting occurs when a Bad Guy realizes that he's on a honeypot
  - Very rare in research honeypots
  - More common with low interaction honeypots
- May trigger destructive attacks
- May trigger Bad Guy simply vanishing
- There will almost *always* be a *potential* fingerprint on any honeypot!

Minimizing Fingerprinting

- First: ***decide if you care!***
  - Production honeypots have succeeded in their mission by the time they are fingerprinted!
  - Research honeypots care more
- Provide the best possible emulation of a real vulnerable system
  - Usually, that's simply a matter of providing a real vulnerable system!

Legal Issues

- Entrapment
- Privacy
- Attacks against 3rd parties

Entrapment

- Entrapment is a *defense* **not** an *offense*!
  - Nobody can prosecute you for “entrapment”
  - Defendant’s lawyer might try to get their client off because the plaintiff “entrapped” the defendant
  - Plaintiff must be law enforcement / Gov’t
  - Plaintiff must have modified the defendant’s behavior
    - So don’t promote your honeypot; they will come

Privacy

- This is actually the one you have to worry about most!
  - Lots of legal issues regarding privacy
    - Somewhat contradictory
    - Common carrier versus network administrator
    - What about conversations with multiple parties?
    - ECPA (electronic communications privacy act)
  - No case law regarding honeypots

Attacks on 3rd Parties

- Potential for liability if a 3rd party is injured by an attacker that used your honeypot as a jump-off point
  - Attempt to show diligence in protecting 3rd parties using data control
  - Nobody (that we've heard of yet) has gone after someone for being a jump-off point, whether their system was a honeypot or just a normal, insecure machine

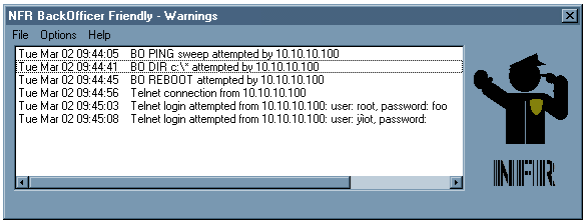
Techniques and Tools

- These are just a few examples
- There are loads of tools and techniques for building honeypots
  - Remember - they should all be different enough that they are harder to fingerprint
  - It is the unknown defense that may block the unknown attack (no, Sun Tzu didn't say that...)

NFR BOF (backofficer friendly)

- Low-interaction ***free*** honeypot for Windows (9*,2*) that installs in seconds
  - Emulates a machine infected with BackOrifice 1.0
  - Emulates other services with fake replies:
    - http
    - telnet
    - ftp
    - pop / imap

BOF Alerts



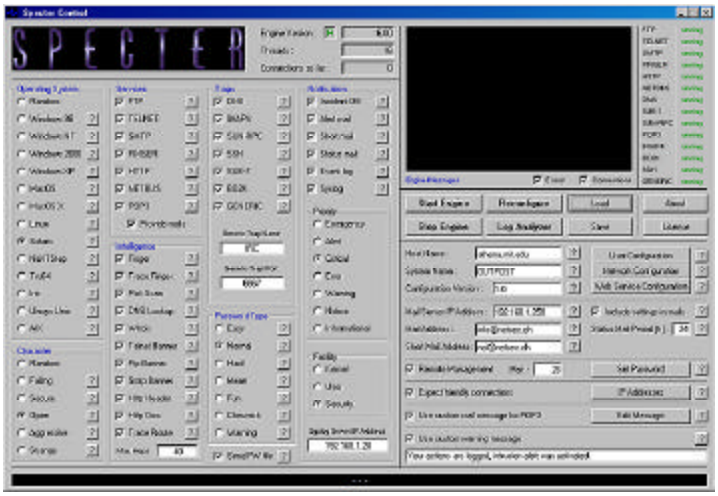
BOF From the Other Side

- BO>**host 10.10.10.1**
  New host: 10.10.10.1:31337
  BO:10.10.10.1>**dir**
  ------- Packet received from 10.10.10.1 port 31337 -------
  Error 53:The network path was not found opening file c:*
  ------- End of data -------
  BO:10.10.10.1>**reboot**
  ------- Packet received from 10.10.10.1 port 31337 -------
  Naughty, naughty. Bad hacker! No donut!
  ------- End of data -------
  BO:10.10.10.1>**quit**

Spectre

- Low-interaction commercial honeypot for windows NT systems
  - Emulates a variety of operating systems at the application level
    - Does not perform IP-level emulation
  - Includes centralized management and reporting
  - Includes session capture and logging

Spectre U/I



Mantrap

- High-interaction commercial honeypot
- Builds multiple virtual installations of Solaris
  - Each looks like a complete system
  - All interactions with the “jail” systems are recorded
  - Remotely manageable (via ssh into the host system)

Port Suckers

- Netcat is a great tool for collecting data on a port
  - `nc -l -p 80 > capture.txt`
  - This will capture all traffic coming in on port 80 to the output file
  - Can easily be harnessed into a .BAT file

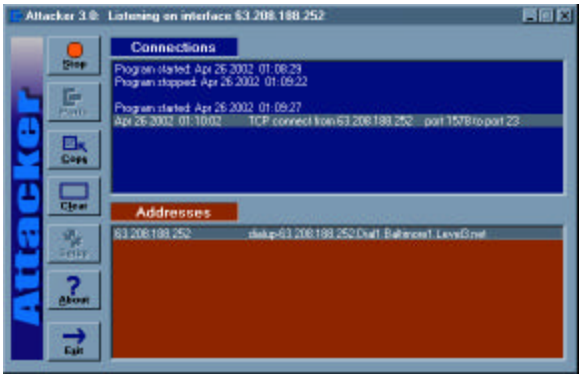
Netcat Port Sucker

```
@echo off
echo port sucker off and running
echo > capture.txt
:top
  nc -l -p 80 >> capture.txt
  netstat -a | find "_wait" >> capture.txt
goto top
```

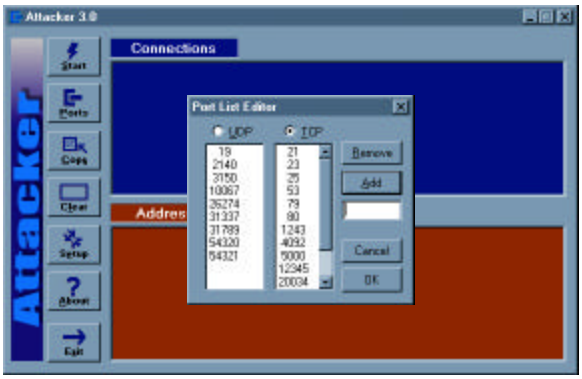

Attacker

- What is it?
  - Free port sucker from foundstone
  - Win-32 application, very simple to install
    - Can listen to a large number of UDP and TCP ports
    - Does not emulate services
    - Does not capture traffic - just beeps
  - Listens on external interface (not loopback) - very non-intrusive!

Attacker In Action



Attacker Port Configuration



Jailing Applications

- Use `chroot` to "jail" applications that are frequently attacked
  - Look for unusual activity within the jail area - such as files being created, or executed
  - Make sure the jail has a writeable /tmp directory! Lots of hack-tools wind up in /tmp
    - Consider setting /tmp as no-exec

Trapping Jailbreaks

- Lots of hacker scripts rely not only on /tmp, they rely on the `rm` command - here are some fun ideas: (you can get fancier!)
  - Put a version of `/bin/false` in the jail area called `/bin/rm` - when the hackers' scripts delete the files it'll leave them there intact
  - Put a version of something called `/bin/sh` that simply records its parameters and inputs then sounds an alarm

SMTP Spam Honeypot

- Instead of running sendmail in queue-processing mode, use:
  sendmail -bd
- Turn on open relaying:
  `FEATURE(`promiscuous_relay')dnl`
- Change delivery mode to queue instead of performing automatic delivery

Queue Options

- Change:

```
# default delivery mode
O DeliveryMode=background
```

- To:

```
# default delivery mode
# Mail never gets delivered if sendmail
is run without
# a "-q" option
O DeliveryMode=queue
```


Now Get Spammed

- You now have an open relay that accepts anything but doesn't re-send it
- Fun things to try:
  - Write scripts to watch the queue
  - Send spammers' test messages on their way but don't deliver the actual messages
  - Contact spammers' base of operations ISPs
- This idea brought to you by *Brad Spencer*

References

- Lots of stuff on:

http://www.tracking-hackers.com

http://www.honeynet.org

- Great free tools on:

http://www.foundstone.com/knowledge/forensics.html

- Mailing list:

honeypots-subscribe@securityfocus.com

