Honeypot forensics -

No stone unturned or logs, what logs?

Krisztian Piller
krisztianp2@yahoo.com
Sebastian Wolfgarten
sebastian.wolfgarten@de.ey.com

21C3, December 2004

Agenda

- Preface
- Introduction to honeypots and honeynets
- Free and commercial honeypot solutions
- Installing your own honeypot
- Introduction to forensics
- Honeypot and binary file analysis
- Case study
- How to be court proof
- Legal aspects of operating honeypots
- Detection of honeypots
- Future of honeypot technologies
- Summary

Preface

Hey, who are you?

- Krisztian Piller (28):
  - IT security expert at European Central bank, Frankfurt
  - Responsible for security-conscious planning, development and implementation of IT related projects at ECB
  - Focus on penetration testing activities
  - Former Ernst & Young employee
  - Speaker at various IT security-related conferences all over Europe

Preface

Hey, who are you? (cont.)

- Sebastian Wolfgarten (23):
  - Student of business & computer science at the University of Cooperative Education in Stuttgart/ Germany
  - Working with Ernst & Young's Risk Advisory Services (RAS) group for more than 2 years
  - Specialized in network security, pen-testing and IT forensics
  - Author of more than a dozen articles for various German IT magazines as well as three books (e.g. "Apache Webserver 2") for the Addison & Wesley publishing house
  - Reviewer for Addison & Wesley and O'Reilly US

Agenda

- Preface
- Introduction to honeypots and honeynets
- Free and commercial honeypot solutions
- Installing your own honeypot
- Introduction to forensics
- Honeypot and binary file analysis
- Case study
- How to be court proof
- Legal aspects of operating honeypots
- Detection of honeypots
- Future of honeypot technologies
- Summary

Introduction to honeypots and honeynets

What is a honeypot?

- Abstract definition:
  "A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource." (Lance Spitzner)
- Concrete definition:
  "A honeypot is a fictitious vulnerable IT system used for the purpose of being attacked, probed, exploited and compromised."

Introduction to honeypots and honeynets

Benefits of deploying a honeypot

- Risk mitigation:
  - A honeypot deployed in a productive environment may lure an attacker away from the real production systems („easy target“).
- IDS-like functionality:
  - Since no legitimate traffic should take place to or from the honeypot, any traffic appearing is evil and can initiate further actions.
- Attack strategies:
  - Find out reasons and strategies why and how you are attacked.

Introduction to honeypots and honeynets

Benefits of deploying a honeypot (cont.)

- Identification and classification:
  - Find out who is attacking you and classify him (her).
- Evidence:
  - Once the attacker is identified all data captured may be used in a legal procedure.
- Increased knowledge:
  - By knowing how you are attacked you are able to enlarge your ability to respond in an appropriate way and to prevent future attacks.
- Research:
  - Operating and monitoring a honeypot can reveal most up-to-date techniques/exploits and tools used as well as internal communications of the hackers or infection or spreading techniques of worms or viruses.

Introduction to honeypots and honeynets

Downside of deploying a honeypot

- Limited view:
  - Honeypots can only track and capture activity that directly interacts with them. Therefore honeypots will not capture attacks against other systems.
- Additional risk:
  - Deploying a honeypot could create an additional risk and eventually put a whole organizations' IT security at risk.
- Remaining risk:
  - Just as all security related technologies honeypots have risk. Depending on the type of honeypot deployed there is the risk the system is being taken over by a bad guy and being used to harm other systems. This could lead to serious legal consequences.

Introduction to honeypots and honeynets

How to classify a honeypot?

- Honeypots are classified by the level of interaction they provide to the attacker:
  - ✓ Low-interaction honeypot: Only parts of (vulnerable) applications or operating systems are emulated by software (e.g. honeyd), no real interaction
  - ✓ Medium-interaction honeypot: A jailed or custom-built environment provides a limited system access.
  - ✓ High-interaction honeypot: An attacker is provided with a full and working operating system enabling him/her to interact in the highest way possible.
- Several honeypots could be combined to an entire honeynet.

Introduction to honeypots and honeynets

Low-interaction honeypots in detail

- Low-interaction honeypots are typically the easiest honeypots to install, configure, deploy and maintain.
- They partially emulate a service (e.g. Unix telnet server or Microsoft's IIS) or operating system and limit the attacker's activities to the level of emulation provided by the software.
- Most importantly there is no interaction with the underlying operating system (at least there shouldn't be).

Introduction to honeypots and honeynets
Advantages of low-interaction honeypots

- Good starting point
- Easy to install, configure, deploy and maintain
- Introduce a low or at least limited risk
- Many ready-to-use products are available
- Logging and analyzing is simple
  - only transactional information are available, no information about the attacks themselves, e.g. time and date of an attack, protocol, source and destination IP as well as port)
- Did we mention simplicity yet?

Introduction to honeypots and honeynets
Disadvantages of low-interaction honeypots

- Pretty boring :-)
- No real interaction for an attacker possible
- Very limited logging abilities
- Can only capture known attacks
- Easily detectable by a skilled attacker

Introduction to honeypots and honeynets

Medium-interaction honeypots in detail

- Medium-interaction honeypots generally offer more ability to interact than a low interaction honeypot but less functionality than high-interaction solutions.
- A typical approach would be a honeypot designed to capture a worm or worm-related activity. Therefore it must interact with the worm more intensively.
- Another example would be the use of UML or a jailed or chrooted environment on a Unix/ Linux system (homemade).

Introduction to honeypots and honeynets
Advantages of medium-interaction honeypots

- By using medium-interaction honeypots you are able to gather a far greater amount of information.
- Unlike low-interaction honeypots you are able to capture worm payloads or real attacker activity.
- Additionally you are able to control attackers (“poisoned honeypot”) and learn what happens after they gain access and how they elevate privileges (e.g. capture their toolkit/ rootkit).

Introduction to honeypots and honeynets
Disadvantages of medium-interaction honeypots

- Medium-interaction honeypots involve a high level of development and customization. Jailed or chrooted environments must be manually created, deployed and maintained.
- As attackers have greater interaction you must deploy this interaction in a secure manner.
- An attacker *might* be able to access the underlying operating system (dangerous!).
- Logging, monitoring and analyzing can be very complex.

Introduction to honeypots and honeynets

High-interaction honeypots in detail

- High-interaction honeypots are the extreme of honeypot technologies.
- Provide an attacker with a real operating system where nothing is emulated or restricted.
- Ideally you are rewarded with a vast amount of information about attackers, their motivation, actions, tools, behaviour, level of knowledge, origin, identity etc.
- Try to control an attacker at the network level or poison the honeypot itself (e.g. with sebek).

Introduction to honeypots and honeynets
Advantages of high-interaction honeypots

- This is where the fun part starts :-)
- You will face real-life data and attacks so the activities captured are most valuable.
- Learn as much as possible about the attacker, the attack itself and especially the methodology as well as tools used.
- High-interaction honeypots could help you to prevent future attacks and get a certain understanding of possible threats.

Introduction to honeypots and honeynets
Disadvantages of high-interaction honeypots

- Building, configuring, deploying and maintaining a high-interaction honeypot is very time consuming as it involves a variety of different technologies (e.g. IDS, firewall etc.) that has to be customized.
- Analyzing a compromised honeypot is extremely time consuming (40 hours for every 30 minutes an attacker spend on a system!) and difficult (e.g. identity exploits, rootkit, system or configuration modifications etc.).
- A high-interaction honeypot introduces a high level of risk and - if there are no additional precautions in place - might put an organizations overall IT security at stake.
- Might lead to difficult legal situations.

Agenda

- Preface
- Introduction to honeypots and honeynets
- Free and commercial honeypot solutions
- Installing your own honeypot
- Introduction to forensics
- Honeypot and binary file analysis
- Case study
- How to be court proof
- Legal aspects of operating honeypots
- Detection of honeypots
- Future of honeypot technologies
- Summary

Free and commercial honeypot solutions

Digest of honeypot products

- BackOfficer Friendly:
  - A free win32 based honeypot solution by NFR Security (a separate Unix port is available but has restricted functionality). It is able to emulate single services such as telnet, ftp, smtp and to rudimentary log connection attempts (http://www.nfr.com/resource/backOfficer.php).
- Deception toolkit (DTK):
  - A free and programmable solution intending to make it appear to attackers as if the system running DTK has a large number of widely known vulnerabilities (http://www.all.net/dtk/dtk.html).
- HOACD:
  - This is a ready-to-run honeyd+OpenBSD+arpd on a bootable CD (http://www.honeynet.org.br/tools/)

Free and commercial honeypot solutions

Digest of honeypot products (cont.)

- !HYW – Honeyweb
  - An in-depth simulation of an IIS 6.0 webserver that enables you to use your web content (perfect choice for capturing worms).
- Mantrap / Decoy Server (commercial)
  - Symantec Decoy Server sensors deliver holistic detection and response as well as provide detailed information through its system of data collection modules.
- Specter
  - SPECTER offers common Internet services such as SMTP, FTP, POP3, HTTP and TELNET. They appear to be normal to the attackers but are in fact traps for them to mess around and leave traces without even knowing they are connected to a decoy system. It does none of the things it appears to but instead logs everything and notifies the appropriate people.
- See http://www.securitywizardry.com/honeypots.htm

Agenda

- Preface
- Introduction to honeypots and honeynets
- Free and commercial honeypot solutions
- Installing your own honeypot
- Introduction to forensics
- Honeypot and binary file analysis
- Case study
- How to be court proof
- Legal aspects of operating honeypots
- Detection of honeypots
- Future of honeypot technologies
- Summary

Installing your own honeypot
How to prepare the installation of a honeypot

- Depending on the type of technology used there are different things to consider when installing and deploying a honeypot.
- Low-interaction honeypot:
  - Make sure an attacker can't access the underlying operating system (especially when using plugins!), just KEEP IT SIMPLE!.
  - If possible make use of the honeypot's features to emulate a more realistic environment (e.g. traffic shaping).
  - Make sure to use the latest versions available.

Installing your own honeypot
How to prepare the installation of a honeypot (cont.)

- Medium-interaction honeypot:
  - Make sure an attacker can't escape the jailed or chrooted environment. Be aware of SUID or SGID files.
- High-interaction honeypot:
  - Use advanced network techniques to control the honeypot (e.g. firewalls, intrusion detection systems) and make sure it can't be used to harm third parties (e.g. legal issues of an open relay)
  - If possible, poison the honeypot (could lead to detection of the poison or the honeypot itself).
  - Use software that actually has vulnerabilities or your honeypot *might* never be exploited successfully.
  - Use tripwire or AIDE to get a snapshot of the system.
  - …

Installing your own honeypot
The do's and don'ts of installing a honeypot

- Don't expect too much!
  - In the beginning don't force yourself too much. You will probably want to catch 0-day exploits but that is a *long* way to go! Start with something simple.
- Wipe the hard drive before using it in a honeypot
  - When recovering files of a compromised honeypot a "dirty" hard disk might confuse you as there is probably old and non-honeypot related data on it which might also be recovered.
- Copy the evidence before analyzing it (e.g. with dd).

Installing your own honeypot

The do's and don'ts of installing a honeypot (cont.)

- Give the honeypot enough time to work.
  - An attacker needs time to compromise a system and work with it. Just give him or her enough time to play (e.g. two weeks).
- Don't put any production data on the honeypot.
  - It's a good idea to place pseudo-interesting data on a honeypot but just don't put any real production data on it!
- Never ever connect to your honeypot while it is in the wild!
  - You will modify the evidence when you connect to your own honeypot while it is active. Just don't do it.

Agenda

- Preface
- Introduction to honeypots and honeynets
- Free and commercial honeypot solutions
- Installing your own honeypot
- Introduction to forensics
- Honeypot and binary file analysis
- Case study
- How to be court proof
- Legal aspects of operating honeypots
- Detection of honeypots
- Future of honeypot technologies
- Summary

Introduction to forensics

No stone unturned

- Computer forensics involves the court-proof preservation, identification, extraction, documentation and interpretation of computer data.
- It is often more of an art than a science making it probably the most complicated part of honeypot research.
- Bear in mind laws and legal regulations when installing, operating or analyzing a honeypot as this might lead to quite difficult legal situations…
  - Monitoring/surveillance without permission
  - Assisting crime
  - Violation of privacy and data protections laws
  - …

Introduction to forensics

No stone unturned (cont.)

- During a forensic investigation follow a clear and well-defined methodology:
  - Acquire the evidence without modifying or damaging the original (and eventually without leaving any traces of your actions behind!)
  - Check integrity of recovered data and verify recovered data and original is identical
  - Analyze the data without modifying it
- The key to any investigation is documentation. Use any documentation alternative (e.g. photos) available to document the investigation process.

Introduction to forensics

Volatile vs. non-volatile information

- Volatile information: Information stored in RAM (e.g. list of running processes, memory contents, open files, network connections, passwords etc.) will be lost when the machine is turned off.
- Non-volatile information: Information is preserved even when the power is switched off (e.g. files stored on a hard drive).
- The important question is: What about volatile information in a forensic analysis?

Introduction to forensics

Volatile information

- Volatile information will be destroyed when the system is switched off however collecting those information on a running system is modifying the evidence.
- No ultimate solution, however experts say: Simply power off Microsoft Windows (e.g. 2000, XP or 2003) systems and fully shutdown Unix/Linux computers.
- We say: Choose your poison :-) Power off a system to start an analysis from the very first. Be aware that as part of a forensic analysis volatile information can be extremely important (e.g. rootkits, backdoors etc.), especially in an incident response.

Introduction to forensics

Tools/commands for obtaining volatile information

- Use safe, statically-linked and non-modified tools (e.g. insert a CD like Helix, see http://www.e-fense.com/helix/) to collect volatile information as binaries on target system might have been modified
- Unix/Linux:
  - ps, netstat, ifconfig, date, grep, last, cat, ls, lsof, mount, dd, fdisk, …
- Microsoft Windows:
  - netstat, ipconfig, VICE, diskmon, filemon, handle, listdlls, process explorer, pstools, regmon, tcpview, tdimon, tokenmon, livekd, dir, vision, dumpacl, fport, loggedon, nbtstat, sfind, etc.…
- Do not store information obtained on local system but transfer them to a third party (e.g. using netcat or ssh).

Introduction to forensics

Safety first!

- After eventually obtaining volatile information, forensically (=bit by bit) copy the entire system in question to another hard drive:
  - Boot the system with Knoppix or Helix and use dd over SSH or netcat/cryptcat (automated tools like AIR/ Automated Image and Restore could help)
  - Alternatively use ghost or dd for Windows as well as hardware write-blockers (e.g. fastbloc)
  - After finishing the imaging, create and store MD5 hashes
- Now, it's time to get yourself a strong coffee and to analyze the data…

Agenda

- Preface
- Introduction to honeypots and honeynets
- Free and commercial honeypot solutions
- Installing your own honeypot
- Introduction to forensics
- Honeypot and binary file analysis
- Case study
- How to be court proof
- Legal aspects of operating honeypots
- Detection of honeypots
- Future of honeypot technologies
- Summary

Honeypot and binary file analysis

Forensic tools

- To support a forensic analysis a variety of tools (http://www.l0t3k.org/security/tools/forensic/) is available including both commercial as well as Open Source products such as
  - EnCase (quoting Encase.com): “As the standard in computer forensics, EnCase Forensic Edition delivers the most advanced features for computer forensics and investigations. With an intuitive, yet flexible GUI and unmatched performance, EnCase software provides investigators with the tools to conduct complex investigations with accuracy and efficiency.”
  - Yes, Encase is good and well accepted (used by some law enforcement agencies across Europe) but pricy

Honeypot and binary file analysis

Forensic tools (cont.)

- Of course there is an Open Source alternative
  - Sleuthkit: The Sleuth Kit (TSK), previously called TASK, is a collection of command line tools based on The Coroner's Toolkit (TCT). Autopsy provides a graphical interface to the command line tools provided by TSK.
  - Both are open source digital forensics tools from Brian Carrier that run on Unix systems (such as Linux, OS X, FreeBSD, OpenBSD, and Solaris) and analyze NTFS, FAT, Ext2, Ext3, UFS1, and UFS2 file systems (see http://www.sleuthkit.org).
  - Sleuthkit is not as professional and convenient as Encase but it is definitely an alternative for performing forensic investigations (not only because it’s free!).

Honeypot and binary file analysis

Forensic analysis – Basic methods

- Manual searching: Manually browsing through the file system of the target helps you in gaining a certain understanding of the system.
- Automated searching: The tools available may assist in searching for valuable data including:
  - Deleted files or data stored in the slack space (e.g. logs, history files, downloaded/installed files)
  - Hidden data in (multi-media) files etc.
  - All files created/modified after a specific date
  - Timeline of activities (MACtimes!)
  - Strings in SWAP etc.
  - …

Honeypot and binary file analysis

Forensic analysis – Advanced methods

- Keyword searches (e.g. suid/sgid, shell, exploit, /bin/sh, shellcode, 0x90 etc.)
  - The correct search expression is very important as imprecise search terms lead to needless or inadequate results
- Use hash sets and tools (e.g. rkhunter, chkrootkit) to identify well-known or modified files (e.g. rootkits, exploits, replaced system binaries)
- If available use the log files of additional network components (e.g. firewalls, intrusion detection systems) to reconstruct the attack
- Also use scripts available (e.g. EnCase.com) to search for malicious data
- Perform a binary file analysis of any data found on target system

Honeypot and binary file analysis

Binary file analysis in a nutshell

- Firstly set up a secure test environment for the analysis, as part of the analysis try to avoid running the program in question, if necessary execute in an isolated but monitored network segment
- Create MD5 sums of the files found
- Scan a suspicious file with an up to date virus scanner (e.g. Symantec AntiVirus)
- Analyze the file and its header (hex editor!) and use the Unix command “file” to (hopefully) identify the true file type
- Extract file properties from an executable (Windows only), try to identify additional programs used (e.g. UPX using PEid)
- Use the “strings” command to extract all strings from the file in question (ensure to get both 7-bit ASCII and 16 bit Unicode strings from a binary!)
- Attempt to reverse-engineer the file(s) found (quite difficult!), if necessary run the file (monitor EVERYTHING!)
- …

Honeypot and binary file analysis

Tools for binary file analysis and RCE (digest)

- Windows:
  - BinText, OllyDbg, dumbug, filemon, regmon, TDIMon, RegShot, ultraedit, IDA Pro, SoftICE, ProcDump, strings.exe, InstallControl, PEid, eXeScope, md5sum, LordPE…
- Unix/Linux:
  - strace/ltrace (if file is executed), gdb, biew, nm, objdump, file, strings, lsof, dd, od, hexdump, elfgrep, ar, md5sum, truss, ldd, …
- Beware of the fact that if run in a virtual environment (e.g. VMware) programs might behave differently (e.g. not malicious) than they would in a non-virtual environment

Honeypot and binary file analysis

A sample binary file analysis on Linux (simplified)

- Malicious file (unknown.bin) was found on October 2nd 2004 on a web server.
- The “file” command identified unknown.bin as “data”.
- Using the “strings” command, the exe packer UPX was easily identified:

```
UPX!u
j!Xj
/tmp/upxAAAAAAAAAAA
[m{r
nux.so.2
6*+7
t?>09
```

- After unpacking the file it was found to be a ELF 32-bit LSB executable, dynamically linked (uses shared libs)

Honeypot and binary file analysis
A sample binary file analysis on Linux (cont.)

- Now the “strings” command extracted more valuable information:

```
/lib/ld-linux.so.2
__gmon_start__
libc.so.6
[ ... ]
210.169.91.66
j010333
65000
httpd
/usr/bin/ping
```

- In VMware the file was found to be an IRC bot that uses vulnerable PHP scripts to gain access to vulnerable system. DFN-Cert Germany published a warning about this bot on October 5th 2004.

Honeypot and binary file analysis

A sample binary file analysis on Windows (simplified)

- RaDa.zip, a malicious binary file, was the challenge of Scan of the Month #32 and was provided by honeynet.org (credits to Chris Eagle for this analysis)
- This file will be analyzed using both Unix/Linux and Microsoft Windows
- Therefore firstly use the Unix command “file” to identify the true file type:

```
$ file RaDa.zip
RaDa.zip: Zip archive data, at least v2.0 to extract
$ unzip RaDa.zip
Archive:  RaDa.zip
  inflating: RaDa.exe
$ file RaDa.exe
RaDa.exe: MS-DOS executable (EXE), OS/2 or MS Windows
```

Honeypot and binary file analysis

A sample binary file analysis on Windows (cont.)

- The “strings” command enables you to obtain a list of all strings a file contains:
- strings -a RaDa.exe

```
!This program is the binary of SotM 32..
[ … ]
rsr%
KERNEL32.DLL
MSVBVM60.DLL
LoadLibraryA
GetProcAddress
ExitProcess
```

- Based on its use of MSVBVM60.DLL (instead of MSVCRT0.DLL, which is the standard C library) the program was probably developed using Visual Basic

Honeypot and binary file analysis

A sample binary file analysis on Windows (cont. 2)

- With “strings” you can also extract the file properties from a given Windows-compatible file on Unix/Linux:

```
$ strings -e l RaDa.exe
VS_VERSION_INFO
StringFileInfo
040904B0
CompanyName
Malware
ProductName
RaDa
FileVersion
1.00
ProductVersion
1.00
InternalName
RaDa
OriginalFilename
RaDa
VarFileInfo
```

Honeypot and binary file analysis

A sample binary file analysis on Windows (cont. 3)

- When starting to analyse a file with Windows make sure to rename it (e.g. to RaDa.bin) in order to prevent the file from accidentally being executed!
- As the limited amount of strings in RaDa.exe indicates, the file as been obfuscated in some way.
- PEid identifies the obfuscator used as the UPX exe packer (upx.sourceforge.net).
- However UPX refuses to unpack the executable as it has been tampered with.

Honeypot and binary file analysis

A sample binary file analysis on Windows (cont. 4)

- Nevertheless using external plugins, PEid (or ollydbg) allows you to unpack RaDa.exe. However be aware of the fact that the file might be executed!
- After unpacking the file all strings can finally be extracted:
  http://10.10.10.10/RaDa
  RaDa_commands.html
  download.cgi
  upload.cgi
  C:\RaDa\tmp
  HKLM\Software\Microsoft\Windows\CurrentVersion\Run\
  C:\RaDa\bin
  RaDa.exe
  HKLM\Software\VMware, Inc.\VMware Tools\InstallPath
  [...]
  - --verbose
  - --visible

Honeypot and binary file analysis
A sample binary file analysis on Windows (cont. 5)

- RaDa.exe seems to add itself to the registry in order to be executed during the system start.
- The file might check for the existence of VMware preventing people from analyzing the program in a virtual environment.
- The program seems to support quite a number of command-line switches (--gui, --verbose, --visible, --install, --server etc.) to (remotely) control the application.
- It is able to download files from a remote server using a non-visible instance of Internet Explorer and therewith to execute given commands locally.

Agenda

- Preface
- Introduction to honeypots and honeynets
- Free and commercial honeypot solutions
- Installing your own honeypot
- Introduction to forensics
- Honeypot and binary file analysis
- Case study
- How to be court proof
- Legal aspects of operating honeypots
- Detection of honeypots
- Future of honeypot technologies
- Summary

Case study

What happened to good old RedHat 7.3?

- One of our honeypots deployed was a high-interaction honeypot based on RedHat 7.3 which was deployed in Frankfurt at the Telehouse data center.
- The honeypot was available for two weeks and wasn't supported by an IDS or a firewall (increased degree of difficulty).
- Three hours after connecting the system to the Internet it was compromised with an Apache exploit.
- The attacker was then able to access a shell on the server and upload data to the home directory of the user running Apache.

Case study

id? uid=0(root) gid=0(root) groups=0(root)!

- By using a kernel exploit the attacker become root.
- Afterwards he (or she?) installed an IRC bouncer allowing him/her to connect anonymously to IRC-based chat networks.
- The attacker downloaded a rootkit and used parts of it to erase his traces.
- Attacker hacked other systems in Tokyo/Japan
- Attack could NOT be fully reconstructed

Case study

Files recovered from this RedHat 7.3 honeypot

- The files were found in a hidden directory on the honeypot (digest):
  - "j" was identified as "sense", a program to sort the output from LinSniffer, part of the Devil rootkit
  - ".all" was identified as Wojciech Purczynski's Linux kernel ptrace/ kmod local root exploit
  - ".kde" was identified as LinSniffer, a powerful Linux ethernet sniffer
  - "logcleaner" was identified as "S.A.R.T. log cleaner“
  - "p" was identified as other local root exploit called ptrace24.c which is an exploit for execve/ptrace race condition in Linux
  - "sslport" was identified as a program to modify the httpd.conf to change the default SSL port (443) to something else (114). Then it restarts the apache server.
  - "sslstop" modifies the httpd.conf to disable the SSL support
  - "wipe" was identified as a modified version of vanish.c, an old program to clean WTMP, UTMP, lastlog, messages, secure, xferlog, maillog, warn, mail, httpd.access_log and httpd.error_log

Case study
So what?

- Lessons learned:
  - It really takes an *enormous* amount of time to analyze a compromised honeypot
  - A honeypot is more valuable when using in combination with other security techniques (e.g. firewalls, intrusion detection systems etc.) to simply the post-mortem analysis
  - Neither chkrootkit nor rkhunter did identify the rootkit partially installed on our system. Manual review is still very important
  - Honeypots are definitely fun and very challenging :-)

Agenda

- Preface
- Introduction to honeypots and honeynets
- Free and commercial honeypot solutions
- Installing your own honeypot
- Introduction to forensics
- Honeypot and binary file analysis
- Case study
- How to be court proof
- Legal aspects of operating honeypots
- Detection of honeypots
- Future of honeypot technologies
- Summary

Introduction to forensics

How to be court proof?

- Most importantly: The chain of custody must be kept at all time!!!
  - Chain of custody is a concept in jurisprudence which applies to the handling of evidence and its integrity.
- So how to deal with it? Documentation, checksums, timestamps, questions (digest):
  - Who had access to the evidence?
  - What procedures did we follow in working with the evidence?
  - How to proof that our analysis is based on copies that are 100% identical to the original evidence?

Introduction to forensics

Chain of custody – the definition

- An identifiable person must always have the physical custody of a piece of evidence.
- All transactions, and every succeeding transaction between the collection of the evidence and its appearance in court, should be completely documented chronologically in order to withstand legal challenges to the authenticity of the evidence.
- Documentation should include the conditions under which the evidence is gathered, the identity of evidence handlers, duration of evidence custody, security conditions while handling or storing the evidence, and how evidence is transferred to subsequent custodians of the evidence for each link in the chain.

Introduction to forensics

Chain of custody – what does it mean for us?

- Chain of custody also refers to the document or paper trail, showing the seizure, custody, control, transfer, analysis, and disposition of physical and electronic evidence.
- Because evidence can be used in court to convict persons of crimes, it must be handled in a scrupulously careful manner to avoid later allegations of tampering or misconduct which can compromise the case of the prosecution toward acquittal or to overturning a guilty verdict upon appeal.

Introduction to forensics

Chain of custody – what does it mean for us? (cont.)

- A testimony (a detailed report) of each step during the analysis must be prepared:
  - Preparation and environmental description
  - Activities in operation
  - Switching off the system
  - Removing the evidence
  - Creating the exact copy of the evidence
  - Findings and how they were found
  - Storage of the evidence and the duplicate
- All step must include the date/time, reason for that step and the name of the person(s) who conducted the investigation.
- Yes, it is awful lot of paperwork.

Introduction to forensics

Some notes from us

- Create photos
  - You can save a lot of time on documentation by attaching photos to the case (operational environment, storage, etc.)
- You cannot decide to create a chain-of-custody if you are already performed any of the steps.
  - Think before you act
- If you are really serious ask for an attorney to help you
- Always describe every possible detail in the reports
  - You never know what will be important later

Agenda

- Preface
- Introduction to honeypots and honeynets
- Free and commercial honeypot solutions
- Installing your own honeypot
- Introduction to forensics
- Honeypot and binary file analysis
- Case study
- How to be court proof
- Legal aspects of operating honeypots
- Detection of honeypots
- Future of honeypot technologies
- Summary

Legal aspects of operating honeypots

Legal aspects in Germany

- First of all: We are no lawyers! If you have any questions or doubts contact your lawyer BEFORE deploying your own honeypot.
- The installation and deployment of a honeypot tends to be legally allowed. However monitoring and identifying an attacker is critical as it may be subject to civil, penal and data protection regulations.
- Installing a honeypot is NOT aiding and abetting an offence.

Legal aspects of operating honeypots

Legal aspects in Germany (cont.)

- Re-attacking an attacker after he or she broke into a honeypot is NOT permitted.
- If an attacker starts to hack other systems, you may have to face legal charges as you have provided him with the inherently insecure honeypot system.
- Generally speaking the punishability of hacking a honeypot is debatable, however once in court all evidence available (e.g. logs, files etc.) has to be accepted by the judge.

Agenda

- Preface
- Introduction to honeypots and honeynets
- Free and commercial honeypot solutions
- Installing your own honeypot
- Introduction to forensics
- Honeypot and binary file analysis
- Case study
- How to be court proof
- Legal aspects of operating honeypots
- Detection of honeypots
- Future of honeypot technologies
- Summary

Detection of honeypots

Techniques of local detection

- Technical properties of the honeypot
  - Respond times, banners, registry entries, inconsistent parameters
- “Social” properties of the system, user interaction
  - No typical usage (e.g. no new files created or accessed on a server for more than a week…)
- Network sniffing
  - Packets going to/from the system (sniffing may be done from an different system on the network if possible)
- Search for traces of Vmware
  - Vmware is a popular platform for honeypots, but it can be detected locally

Honeypot Detection

Techniques of local detection (cont.)

- Search for traces of honeypot tools
  - Temp folders, kernel dumps, backdoors (sebek etc.)
- Search for the history files/logs and other configuration errors
  - Not only bad guys make mistakes :-)
- Vulnerabilities/exploits for the honeypot product itself (low- or medium-interaction honeypots only)
- Just be creative :-)

Honeypot Detection

Remote detection techniques

- This one is much harder: Inconsistency is your best friend (only applies to low-interaction honeypots!)...
- Technical properties of the honeypot
  - Respond times, banners, registry entries, inconsistent responses or parameters
- Vulnerabilities/exploits for the honeypot
  - Could lead to the detection of the honeypot (still waiting for the first honeypots scanners…)

Honeypot Detection

Examples of honeypot detection

- Remotely fingerprinting honeyd:
  - Honeyd <0.8 is detectable by sending an invalid TCP packet (SYN+RST flag) to a target system as answers those types of requests (which it shouldn't)
- Spotting sebek:
  - The presence of sebek is usually not visible although some hidden kernels modules are in use. Nevertheless there are ways to detect the presence of those modules by spotting system anomalies, see http://www.security.org.sg/vuln/sebek215.html and http://www.phrack.org/unoffical/p62/p62-0x07.txt (as well as last DefCon!)

Honeypot Detection

Examples of honeypot detection (cont.)

- Inconsistencies in TCP/IP stack (remotely detectable):
  - Tools like hping can be used to detect incorrect TCP/IP stack emulations indicating the use of a low-interaction honeypot (nmap doesn't recognize the difference yet!):
    1) Normal RH9: TTL=64, window=0, id=0, DF
    2) RH9 on vmware: TTL=64, window=0, id=0, DF
    3) RH9 on honeyd: TTL=64, window=1460, id=0, DF
  - This method works even better on Unix systems emulating Windows and vice versa:
    1) Normal Win2k SP4: TTL=128, window=0, id=+, DF
    2) honeyd emulating Win2k SP4: TTL=64, window=1460, id=0, DF
- The interesting elements of a packet are: Time to live, window size, IPID and Don't Fragmentation-Bit

Honeypot Detection

Overview of different TCP/IP stacks

- A list of properties of different TCP/IP stacks could easily be build (e.g. with hping):

OS	Platform	Vendor	Device/System	Default TTL	WINDOW SIZE	ID	DF bit
AIX 4.2.1	R6000	IBM	n/a	60	16384	+	Y
AIX 5.2	R6000	IBM	n/a	60	16384	+	N
FreeBSD 4.7	Intel	FreeBSD	n/a	64	57344	+	Y
Linux 2.4.20	Intel	Gentoo	n/a	64	32767	0	Y
Linux 2.4.20	Intel	Debian	n/a	64	5840	0	Y
Linux 2.4.21	Intel	SuSE	n/a	64	0	+	Y
Linux 2.4.21	Intel	RedHat	n/a	64	5840	+	Y
OS/400 5.1	Intel	?	n/a	64	8192	+	Y
Solaris 2.5.1	Sparc	Sun	n/a	255	9112	+	Y
Solaris 2.6	Sparc	Sun	n/a	255	9112	+	Y
Solaris 2.7	Sparc	Sun	n/a	255	9112	+	Y
Solaris 2.7	Sparc	Sun	n/a	255	9112	+	Y
Solaris 2.8	Sparc	Sun	n/a	255	24656	+	Y
Solaris 2.9	Intel	Sun	n/a	60	65392	+	Y
Windows 2000 Professional SP3	Intel	Microsoft	n/a	128	64512	+	Y
Windows 2000 Professional SP3	Intel	Microsoft	n/a	128	64240	+	Y
Windows 2000 Server SP4	Intel	Microsoft	n/a	128	65535	+	Y
Windows 2003 Server Standard	Intel	Microsoft	n/a	128	16616	+	Y
PIX 6.2.2	?	Cisco	n/a	257	4096	+	N
FreeBSD 4.9	Intel	FreeBSD	n/a	64	57344	+	Y
D-Link DWL-900+ Wireless AP	?	D-Link	Wireless AP	127	8192	+	N
Linux 2.4.24	Intel	Kernel.org	n/a	64	5840	0	Y
Solaris 2.8	Intel	Sun	n/a	60	65392	+	Y
Fiberline Broadband Router	?	Fiberline	Broadband Router	60	4096	+	N

Honeypot Detection

Demonstration
honeyd detection

Honeypot Detection

VMware detection

- VMware detection is only possible locally as the attacker deals with the same OS than without VMware.
- However there are at least some ways:
  - Detection of the BIOS version used (e.g. UNICORE Bios Wizard)
  - Detect installed VMware-tools
  - Detect VMware magic value (0x564D5868)
    - This is a special I/O Port used by the VMware-tools to communicate between the Host system and the virtual system. Can be used for funny tricks, too (move mouse, set clipboard, pop-up dialogs, …).
  - VMware fingerprinting checks for standard virtual VMware devices (e.g. processor, ioport, scsi, …)
  - Anomalies in VMware configuration (Intel Pentium4 2,6GH with only 128M RAM??? or an unusual amount of system memory such as 96MB or 224MB)

Agenda

- Preface
- Introduction to honeypots and honeynets
- Free and commercial honeypot solutions
- Installing your own honeypot
- Introduction to forensics
- Honeypot and binary file analysis
- Case study
- How to be court proof
- Legal aspects of operating honeypots
- Detection of honeypots
- Future of honeypot technologies
- Summary

Future of honeypot technologies

Future on the good side…

- Honeytokens
- Wireless honeypots
- SPAM honeypots
- Honeypot farms
- Search-engine honeypots

Future of honeypot technologies

Honeytokens

- The concept of honeytokens is not new. This concept is as old as security itself. For example, map-making companies often inserting bogus cities or roads into their maps to determine if competitors are selling copied versions of their own maps.
- Generally a honeytoken could be a bogus record in a database which is not needed by any application. If someone tries to access this an alarm can be indicated (honeypot inside an application).
- Example: Patient record John F. Kennedy in a hospital's patient database. There is no such patient in the hospital.
  - The monitoring can be made in the database or on the wire (e.g. Snort) looking for the signature "John F. Kennedy"

Future of honeypot technologies

Wireless honeypots

- Usage of honeypot technology to detect intruders of wireless networks.
- Unlike Internet-based honeypots, anyone detected on a wireless network will be located within a few blocks of the trap, perhaps parked in a car or sitting on a bus bench. Therefore you may plan to deploy video cameras on the street, or to physically confront hackers.
- Other wireless technologies, like Bluetooth could be also considered.

Future of honeypot technologies

Spam honeypots

- Simply put a honeypot with a SMTP service running in your own IP range. Everyone accessing this service can be added on your black-list of spammers. This list can be used by your real mail gateway not to accept mails from these addresses (email and ip).
- If you don't want to build your own open relay honeypot, you can simply download a complete package like Jackpot, which is a ready-to-run Simple Mail Transport Protocol (SMTP) relay honeypot called Bubblegum Proxypot.
- Spam honeypots could also be used for statistic spam analysis (e.g. where are the spammers coming from, how many messages are they sending etc?).

Future of honeypot technologies

Honeypot farms

- Farming is a solution to simplify large honeynet deployments
- Instead of deploying large numbers of honeypots, or honeypots on every network, you simply deploy your honeypots in a single, consolidated location. This single network of honeypots becomes your honeypot farm, a dedicated security resource (“honeypot outsourcing”).
- Attackers are then redirected to the farm, regardless of what network they are on or probing.
- Administration efforts and inherit risks can be decreased enormously.
- Even more future: Dynamic appliance of honeypots…

Future of honeypot technologies

Search-engine honeypot

- A web server build to catch attackers using a search engine (mostly Google) as an attacking tool.
  - A site describing Google hacking: http://johnny.ihackstuff.com
  - A working search engine honeypot: http://gray-world.net/etc/passwd/
- This idea could be developed further to create specific honeypots against specific hacking techniques.

Future of honeypot technologies

Future on the evil side…

- New honeypot detection technologies
- Automated honeypot scanners and “confusers” – Anti Honeypot Technologies
- Honeypot exploits

Future of honeypot technologies

Honeypot detection technologies

- Finding honeypots is a difficult process
- As discussed before attackers look for differences between a real system and a honeypot representation of a system. Examples of techniques under development:
  - Connection Limiting
    - Honeypot will count the outbound connections within a period of time.
    - Once the threshold is reached the new outbound connections are denied
    - One of the most easiest characteristics to detect
    - Simply open up 10-20 websites and see if the connection is blocked
  - Outbound packet alteration
    - Modifies packets that are believed to be of an exploitive nature
    - Honeypots compute a hash of portions of the packet
    - Returns a response based on the hash
    - Attacker expects to receive a known response but instead receives a modified response from the honeypot

Future of honeypot technologies

Anti honeypot technology

- If a honeypot is detected, users can attempt to bypass detection or destruct the honeypot
- Honeypot can be attacked if detected
- The honeypot could be used to attack other systems
- Prevents honeypots from collecting valuable information
- Honeypot itself loses effectiveness of being a covert system once compromised !!!!!!!!

Future of honeypot technologies

Anti honeypot technology

- Send-Safe's proxy scanner searches for multiple open proxy servers for obscuring a spammers identity.
  http://www.send-safe.com/honeypot-hunter.php
- "Send-Safe Honeypot Hunter is a tool designed for checking lists of HTTPS and SOCKS proxies for so called "honey pots". "Honey pots" are fake proxies run by the people who are attempting to frame bulkers by using those fake proxies for logging traffic through them and then send complaints to ones' ISPs." … ☺
- It creates his own mail server and tries to send test emails to himself.

Agenda

- Preface
- Introduction to honeypots and honeynets
- Free and commercial honeypot solutions
- Installing your own honeypot
- Introduction to forensics
- Honeypot and binary file analysis
- Case study
- How to be court proof
- Legal aspects of operating honeypots
- Detection of honeypots
- Future of honeypot technologies
- Summary

Summary

Coming closer to the end…

- Honeypots are a quite new field of research, lot's of work has still to be done (so start your own now!)
- Try your first own forensic investigation by analyzing the files provided by honeynet.org :-)
- Analyzing compromised honeypots supports you in getting a certain understanding of tools, methodologies and avenues used by attackers in the wild (may improve your own hacking skills as well as defence strategies!)

Further information

Good reads offline

- “Computer Forensics”, Warren G. Kruse II et. al, Addison & Wesley Professional, 1st edition 2002 (ISBN: 0-201-70719-5)
- “Honeypots”, Lance Spitzner, Addison & Wesley Professional, 2002 (ISBN: 0-321-10895-7)
- “Computer Forensik”, Alexander Geschonneck, dpunkt-Verlag, 2004 (ISBN: 3-898-64253-4)
- “Süße Falle”, Lukas Grunwald et. al, iX 6/2003

Further information

Good reads offline (cont.)

- “Windows Forensics and Incident Recovery”, Harlan Carvey, Addison & Wesley Professional, 1st edition 2004 (ISBN: 0-321-20098-5)
- “Incident Response”, Kevin Mandia et. al, Osborne/McGraw-Hill, 1st edition 2001 (ISBN: 0-072-13182-9)
- “Security Warrior”, Cyrus Peikari et. al, O’Reilly, 1st edition 2004 (ISBN: 0-596-00545-8)

Further information

Historic reads (offline)

- “The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage”, Clifford Stoll, 1990 (!)
- “An Evening with Berferd In Which a Cracker is Lured, Endured, and Studied”, Bill Cheswick, 1991 (!)

Further information

Online resources

- Honeynet Project, http://www.honeynet.org
- Lance Spitzner, “Tracking hackers”, http://www.tracking-hackers.com
- Lance Spitzner, “Honeypot Farms”, http://www.securityfocus.com/infocus/1720
- Lance Spitzner, “Honeytokens”, http://www.securityfocus.com/infocus/1713
- Distributed Honeypot Project, http://www.lucidic.net
- Niels Provos, honeyd, http://www.honeyd.org

Further information

Online resources (cont.)

- Jacco Tunnissen, “Honeypots, Intrusion Detection, Incident Response”, http://www.honeypots.net
- Phrack magazine, http://www.phrack.org
- Lance Spitzner, “Fighting Relay Spam the Honeypot Way”, http://www.tracking-hackers.com/solutions/sendmail.html
- Honeynet Germany, “IT-Sicherheit in Deutschland”, http://www.honeynet.de
- Google.com :-)

Become involved…

Honeynet Germany is looking for members!

Honeynet Germany is looking for new members! Please take a look at the projects' website which is http://www.honeynet.de. If you are passionate about honeypots and/or IT security in general, feel free to join us.

Honeypot Forensics

The end.

Thanks for your (long) patience
and attention!

We would now like to
answer your questions.

This presentation is available online at http://www.wolfgarten.com/ccc.