Honeypots in a nutshell -

Tracking hackers for fun and profit

A presentation by Sebastian Wolfgarten (sebastian@wolfgarten.com)

24. November 2005, Dublin City University/Ireland

Duration: approx. 45 minutes

Agenda

1. Introduction to honeypots and honeynets
2. Free and commercial honeypot solutions
3. Installing your own honeypot
4. Honeypot and binary file analysis
5. Case study
6. 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:**

  Identify reasons and strategies why and how you are attacked.

- **Identification and classification:**

  Find out who is attacking you and classify him (her).

Introduction to honeypots and honeynets - Benefits of deploying a honeypot (cont.)

- **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.

- **Time:**

  Operating and analyzing honeypots takes an enormous amount of time ultimately limiting its use.

- **Remaining risk:**

  Just as all security related technologies honeypots have risks associated with them. Depending on the type of honeypot deployed there is the risk of the system 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 an attacker:**
  - **Low-interaction honeypot:** Only certain parts of (vulnerable) applications or operating systems are emulated by software (e.g. honeyd), no real interaction between attacker and honeypot possible.
  - **Medium-interaction honeypot**: A jailed/chrooted or custom-built environment provides a limited system access.
  - **High-interaction honeypot**: An attacker is provided with a complete and fully working operating system enabling him/her to interact in the highest way possible.
- **Obviously several honeypots could be combined to an entire honeynet.**

Introduction to honeypots and honeynets - Low-interaction honeypots in detail

- **Basics:**
  - 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).
- **Pros:**
  - 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
- **Cons:**
  - Pretty boring :-)
  - No real interaction for an attacker possible
  - Very limited logging abilities
  - Easily detectable by a (more or less) skilled attacker

Introduction to honeypots and honeynets - Medium-interaction honeypots in detail

- **Basics:**
  - 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).
- **Pros:**
  - By using medium-interaction honeypots you are able to gather a far greater amount of information.
  - 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).
- **Cons:**
  - 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!).

Introduction to honeypots and honeynets - High-interaction honeypots in detail

- **Basics:**
  - 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).
- **Pros:**
  - 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.
- **Cons:**
  - 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. identify exploits, rootkit, system or configuration modifications etc.).
  - Might lead to difficult legal situations.

Agenda

1. Introduction to honeypots and honeynets
2. Free and commercial honeypot solutions
3. Installing your own honeypot
4. Honeypot and binary file analysis
5. Case study
6. Summary

Free and commercial honeypot solutions - Digest of honeypot products

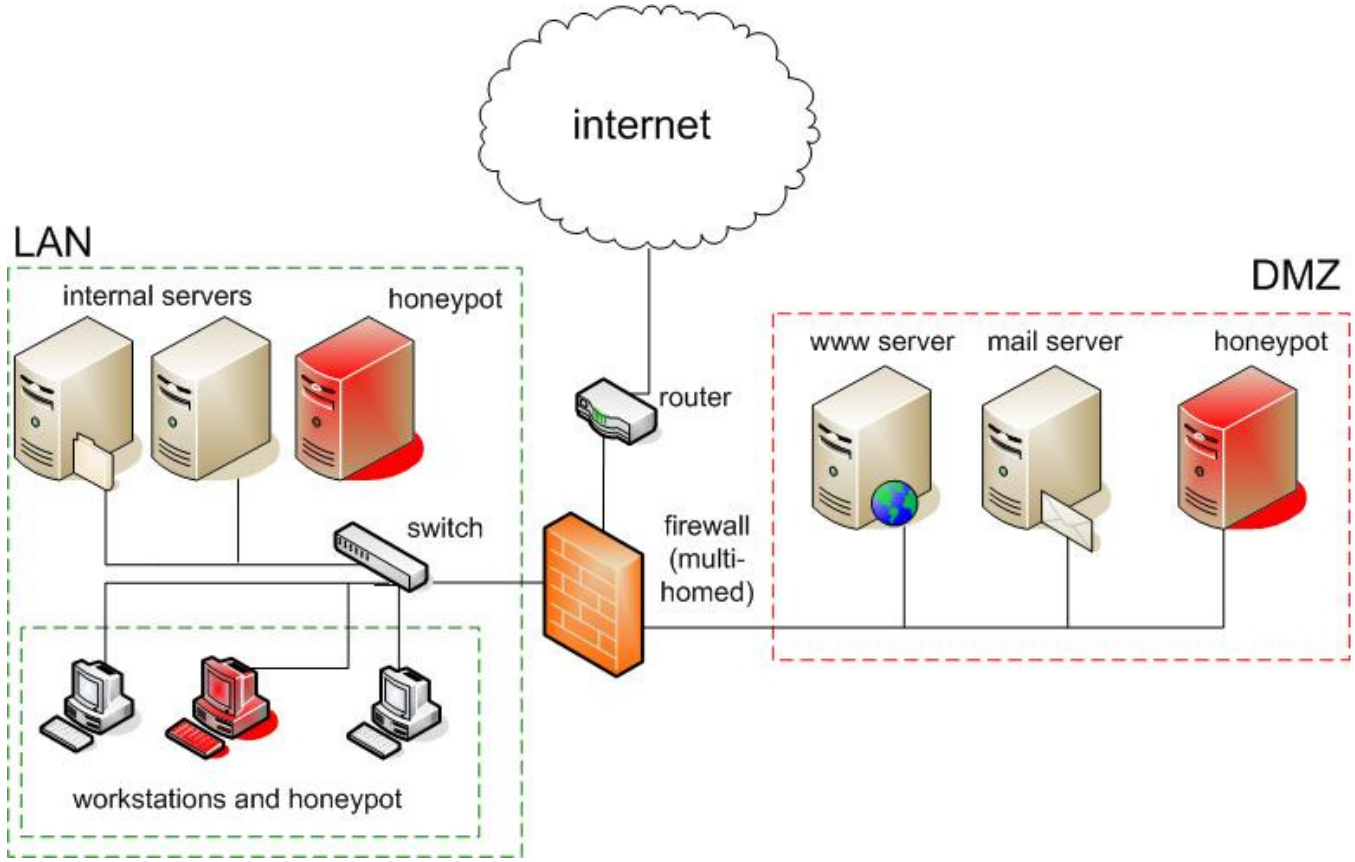
- **Honeyd**: Honeyd is a small daemon that creates virtual hosts on a network. The hosts can be configured to run arbitrary services, and their personality can be adapted so that they appear to be running certain operating systems.
- **Honeycomb**: A system for automated generation of signatures for network intrusion detection systems. The system applies protocol analysis and pattern-detection techniques to traffic captured on a honeypot
- **Honeywall**: The Honeywall CDROM is a bootable CDROM that installs all of the tools and functionality necessary to quickly create, easily maintain, and effectively analyze a third generation honeynet.
- **mwcollect**: A client-side honeypot solution to capture worms and other autonomously spreading malware in a non-native environment like FreeBSD or Linux.
- **See http://www.securitywizardry.com/honeypots.htm for a more complete list of honeypot products available.**

Agenda

1. Introduction to honeypots and honeynets
2. Free and commercial honeypot solutions
3. Installing your own honeypot
4. Honeypot and binary file analysis
5. Case study
6. Summary

Installing your own honeypot -
Positioning a honeypot in a network

- The position of a honeypot within an organization is crucial to its overall success. A sample corporate setup might be (simplified):



Installing your own honeypot -
The do's and dont's of installing a honeypot

- **Don't expect too much**!
  - In the beginning don't push 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).**
- **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 *real* 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

1. Introduction to honeypots and honeynets
2. Free and commercial honeypot solutions
3. Installing your own honeypot
4. Honeypot and binary file analysis
5. Case study
6. Summary

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

- Obviously the correct search expression is very important as imprecise search terms lead to needless or inadequate results.
- **Advanced methods include but are not limited to:**
  - Keyword searches (e.g. suid/sgid, shell, exploit, /bin/sh, shellcode, 0x90 etc.)
  - 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!)
- …

Agenda

1. Introduction to honeypots and honeynets
2. Free and commercial honeypot solutions
3. Installing your own honeypot
4. Honeypot and binary file analysis
5. Case study
6. Summary

Case study -
What happened to good ol' RedHat 7.3?

- One of the first high-interaction honeypots I 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 (willingly increased degree of difficulty).
- Less than 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 local 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 (as no IDS data was available)

Case study -

Files recovered from a 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 the 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
- Scanner software such as chkrootkit or rkhunter did not identify the rootkit partially installed on the system. Manual review is still very important.
- Honeypots are definitely fun and very challenging :-)

Agenda

1. Introduction to honeypots and honeynets
2. Free and commercial honeypot solutions
3. Installing your own honeypot
4. Honeypot and binary file analysis
5. Case study
6. Summary

Summary -
Coming closer to an 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)
- “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)
- “Honeypots for Windows“, Roger A. Grimes, Apress, (ISBN: 1-590-59335-9)

Further information -
Historic reads…

- “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 -
Other ressources

- 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 ressources (digest!)

- 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.org, http://www.honeynet.org
- Google.com :-)
- …

Thank you for your (long) attention.

I am now looking forward to answering your questions.