

Aufbau und Betrieb von Honeypot-Systemen

Lukas Grunwald
DN-Systems GmbH
Hildesheim, Germany, 31137
l.grunwald@dn-systems.de
<http://www.dn-systems.de>

Hintergrundwissen von DN-Systems

Zusammenfassung

Ziel eines Honeypot-Systems ist es, Tools, Methoden und Motive der Hacker-Community zu lernen und daraus Konsequenzen für effiziente Gegenmaßnahmen zu finden. In diesem Tutorial wird nach einem theoretischen Überblick über die bestehenden Honeypot-Szenarien praktisch gezeigt, wie sich anhand von Beispielen Honeypots aufsetzen, überprüfen und auswerten lassen.

1 Theorie

Als Instrument zur Verteidigung sensibler IT-Systeme erfreuen sich die sogenannten Honeypots zunehmender Beliebtheit. Diese, bewusst unsicher konfigurierten, Rechner sollen zum einen die Bösewichte von den wichtigen Systemen fernhalten. Zum anderen dient die Analyse der auf die "Honigtöpfe" erfolgten Angriffe der Prävention solcher Angriffe.

Bei Netzsicherheitskonzepten wie Firewalls oder Virtual Private Networks (VPNs) geht der Trend seit einiger Zeit dahin, sicherheitsrelevante Ereignisse nicht einfach abzublocken, sondern mit Blick auf die Risikominimierung im produktiven Umfeld exakt nachzuvollziehen und zu analysieren. Dieser Themenkomplex ist unter dem Begriff Intrusion Detection System (IDS) zusammengefasst.

Intrusion Detection kann entweder zeitnah (near realtime) bzw. zeitgleich (realtime) oder im Nachhinein (historic) erfolgen. Dient die Analyse in erster Linie der Einbruchvermeidung, bezeichnet man sie als Intrusion Prevention.

1.1 Aufgaben eines Honeypots

Die Werkzeuge, die zu diesen Zwecken eingesetzt werden, kombinieren verschiedene Funktionen wie die Überwachung von Daten oder Ereignissen (Monitoring), das Protokollieren (Logging), das Filtern von Datenpaketen, (Filtering) sowie Einbruchsalarm (Intrusion Detection), Systemüberprüfung (Audit) und die Veranlas-

sung diverser abgestufter Gegenmaßnahmen (Escalation). Diese Aufgaben sind im Unternehmen organisatorisch in einen Prozess einzubinden, der gewährleistet dass alle anfallenden Daten ausgewertet und den verantwortlichen Personen oder Instanzen zugeführt werden.

Eine Hauptaufgabe der Werkzeuge ist das Sammeln verwertbarer Informationen über alle sicherheitsrelevanten Ereignisse und Aktivitäten, um so eine Erfahrunggrundlage für das Ergreifen geeigneter Gegenmaßnahmen zu erhalten.

In jüngster Zeit setzen Fachleute hier immer häufiger auf so genannte Honeypots. Erstens kann man mit ihnen Ereignisse in allen zeitlichen Relationen – zeitgleich/zeitnah und historisch – nachvollziehen. Zweitens ergänzen sie die Lücken herkömmlicher IDS-Systeme in puncto Vermeidung von Sicherheitsvorfällen.

1.2 Definition Honeypot

Ein Honeypot ist ein fiktives, sicherheitstechnisch verwundbares System, das als Falle für nicht-legitimierte Benutzer und Angreifer fungieren soll.

1.3 Erweiterte Funktionen

Darüber hinaus schützt die Honeypot-Installation sensible IT-Systeme, indem es durch die bewusst unsichere Konfiguration die Aufmerksamkeit auf sich zieht. Von den zusätzlichen Informationen, die man auf diese Weise über die verschiedenen Angriffsarten und -Konstellationen erhält, können die IDS-Systeme in Netzwerken profitieren.

1.4 Ausprägungen

In der täglichen Praxis setzen Sicherheitsadministratoren Honeypots und ihnen zugehörige Techniken ein,

- um sicherheitskritische Angriffe auf sich zu ziehen, (Opferrechner oder ein Dummy)
- um die (in der Regel limitierten) Ressourcen des Angreifenden zu erschöpfen,

- um Hinweise zu noch unbekanntem Angriffsszenarien zu erhalten und die Reaktionsfähigkeit zu verbessern,
- um Beweismittel für den Vorsatz der Tat in juristisch verwertbarer Form zu dokumentieren,
- um die strafrechtliche Identifikation des Angreifers zu ermöglichen,
- um Konflikte mit geltendem Arbeitsrecht und Datenschutz zu vermeiden,
- um als Frühwarnsystem zu agieren.

Weitere Ziele, speziell im Hinblick auf Forschung und Lehre, sind:

- Erforschung der Ziele von Angreifern
- Analyse von neuen Angriffswerkzeugen
- Erforschung neuer Angriffvarianten, -strategien und -philosophien
- Typisierung von Angreifern (Einzeltäter, Gruppe, Unternehmen, staatliche Institutionen, etc.)
- Täter- und Persönlichkeitsanalysen von Angreifern (Profiling) zu ermöglichen oder zu verbessern

Theoretisch lassen sich Honeypots in nahezu alle IT-Systemlandschaften und -Topologien implementieren. In der Praxis ist das allerdings nur in den Bereichen sinnvoll, in denen sensible, exponierte Informationssysteme eines Unternehmens stehen (etwa beim Vorstand oder in der Buchhaltung) beziehungsweise Übergänge zu angreifbaren Bereichen im Netzwerk existieren, wie die DMZ oder Wireless LANs.

Insbesondere für die drahtlosen Netze sind Honeypots eine sinnvolle Ergänzung: Kann man doch mit ihnen die meistens in öffentlichen Bereichen produktiv arbeitenden Infrastrukturen auf ein höheres Sicherheitsniveau bringen – zumindest, wenn es um das Nachvollziehen sicherheitsrelevanter Ereignisse geht.

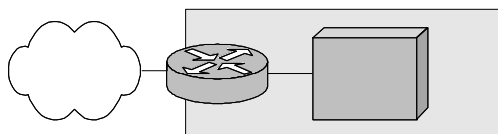
1.5 Honeypot-Typen

Man unterscheidet hostbasierende Honeypot- und netzbasierende HoneyNet-Varianten. Diese lassen sich nach dem derzeitigen Stand der Technik in folgende Untergruppen einteilen:

1.5.1 Typ 1: Physikalisch eigenständiges System

Eigenständiges System, welches via Sicherheits-Hardware angebunden ist (Vollinstallation: speziell notwendig für eine forensische Analyse).

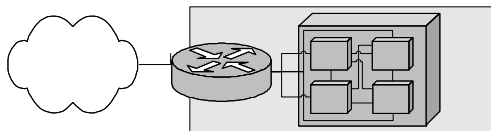
Beispiel: Ein Opferrechner auf der Basis alter i586-Hardware.



1.5.2 Typ 2: Logisch eigenständiges System

Vollinstallation (virtuell; wirtschaftlich sinnvoll, um mehrere Systeme auf einer Hardware zu betreiben), etwa auf der Basis von Virtual Machines (VM).

Beispiel: Usermode Linux, welches als Honeypot einen Server laufen lässt.



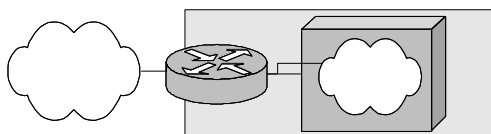
1.5.3 Typ 3: Logische Emulation; One-to-One-Beziehung

(einfache, realitätsnahe Darstellung), beispielsweise shadow, dtk

1.5.4 Typ 4: Logische Emulation; One-to-Many-Beziehung

Hier wird, ohne reelle Hardware zu besitzen, eine komplexe, realitätsnahe Darstellung kompletter IT-Strukturen im Netzwerk mit konnektierten Hosts, Routern und Servern emuliert.

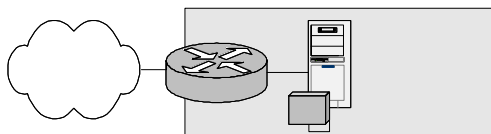
Beispiel: Virtuelles Netzwerk mit Honeyd



1.5.5 Typ 5: Punktuelle Fallen (Traps)

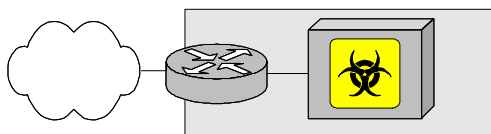
Diese Fallen werden auf Produktiv-Servern an speziellen kritischen Schaltstellen - wie dem Mailserver - installiert. Sobald jemand in diese Falle tappt, wird eine Eskalation durchgeführt.

Beispiel: "netcat -p 23" oder LaBrea, um Code Red und andere Würmer auszubremesen.



1.5.6 Typ 6: Poisoned Honeypot

Die Typen 1-5 werden mit "Malicious Code" (Viren, Würmer, Trojaner) ausgestattet. Dadurch entsteht ein sogenannter poisoned Honeypot (vergifteter Köder). Diese Variante ist manchmal im militärischen Umfeld möglich, da solch ein System aktiv zurück schlägt.



1.6 Täter

Es gibt eine Vielzahl von Angreifern und suspekten Personen, die speziell über das Internet versuchen, in ein HoneyPot-System einzudringen. Als Täter kommen verschiedene Persönlichkeiten in Betracht, welche unterschiedliche Bedrohungsszenarien darstellen.

Folgende sind hier exemplarisch herausgestellt:

1.6.1 Skript-Kinder

Bei den Skript-Kindern (skript kids) handelt es sich um jugendliche Personen, welche sich einen Spaß daraus machen, andere User im Internet zu verärgern. Sie haben einen relativ niedrigen Wissensstand bezüglich der eingesetzten Technologien und Angriffsszenarien. Skript kids benutzen meistens Tools und Werkzeuge, die auf so genannten Hacker-Pages im Netz veröffentlicht werden. Sie agieren mit der Motivation, andere Leute zu verärgern und Schaden anzurichten. Leider treten die Angriffe dieser Gruppe als Massenphänomen auf: Die Gefahr besteht darin, dass sinn-, ziel- und wahllos Rechner angegriffen werden. Die Skript-Kinder gehören auch zu dem Personenkreis, der den meisten für die HoneyPots relevanten Verkehr verursachen, da ihre Angriffe auf Port-Scannern und anderen Tools basieren.

1.6.2 Hacker

Hacker sind Personen, deren Hauptmotivation darin besteht, anderen ihre Kenntnisse zu demonstrieren, indem sie in fremde Systeme einbrechen. Meistens verfügen sie über eine sehr gute Bildung im technischen und Social Engineering-Bereich. Hacker entwickeln in der Regel neue Tools, Techniken und Methoden, welche ausschließlich mit HoneyPot-Systemen zu analysieren sind, da diese Werkzeuge meist nicht veröffentlicht werden. Sie machen sich keine Gedanken hinsichtlich der gesetzlichen Konsequenzen ihres Tuns, arbeiten aber in aller Regel nicht kommerziell.

1.6.3 Cracker

Der Cracker ist die kriminelle Version des Hackers. Im Gegensatz zum Hacker hat er bessere monetäre Ressourcen und einen professionellen bzw. kommerziellen Hintergrund. Der typische Cracker arbeitet im Bereich der Industriespionage und hält hauptsächlich nach Industrie- und Regierungsgeheimnissen Ausschau. Sein Hauptmotiv ist Geld. Der Cracker ist als professioneller Krimineller einzustufen.

1.6.4 Kommerzieller Werbemailversender (Spam-Hacker)

Ein neu auftretendes Phänomen sind die so genannten Spam-Hacker oder Versender von kommerziellen Werbemails. Diese suchen gezielt Rechner mit Schwachstellen, um in diese einzudringen und dann ihre verärgern

Werbebotschaft auf Kosten Dritter - d.h. gänzlich unbeteiligten Firmen im Internet - über diese Schwachstelle zu verbreiten.

1.6.5 Analyse der Herkunft

Eine weitere Aufgabe des HoneyPot-Systems besteht darin festzustellen, aus welchem Netz, aus welcher Region bzw. welchem Erdkreis der Angriff ausgeführt wurde, um mit weiteren Maßnahmen - wie z.B. eine Eskalation an das lokale Fernteam bzw. an den Account des Providers oder aber eine Blockade im eigenen firmeninternen Netz aufzubauen. Bei der Analyse ist zu unterscheiden, ob es sich um einzeln aufgebaute UDP-Verbindungen oder um TCP-Verbindungen handelt. Da nicht alle Netze mit einer Spoof-Protection ausgestattet sind, d.h. einem Schutz gegen das Vortäuschen einer fremden IP-Adresse, ist es notwendig, eine tiefere Analyse des Verkehrs durchzuführen. Auf diese Weise wird sichergestellt, dass diese Pakete nicht in ein fremdes Netz eingespeist und somit die wahre Identität des Angreifers verschleiert wird.

1.6.6 TCP-Verbindung

Bei einer TCP-Verbindung ist die Analyse relativ einfach: Damit eine TCP-Verbindung überhaupt zustande kommt, muss der so genannte 3-Wege-Handshake durchgeführt werden. Dieser stellt sicher, dass von der eingehenden IP-Adresse wieder eine zur ausgehenden aufgebaut wird. Ausnahmeszenario ist ein asymmetrisches Routing, wo der Hacker an zwei Stellen sitzt und asymmetrische Verbindungen in das Netz einspeist. Diese lassen sich aber mit entsprechenden Tools relativieren, wie in dem folgenden Beispiel zu sehen ist.

```
May 4 17:27:36 195.138.147.189:1749 -> 193.1XX.XX1.2:1433 SYN *****S*
May 4 17:27:36 195.138.147.189:1750 -> 193.1XX.XX1.3:1433 SYN *****S*
May 4 17:27:36 195.138.147.189:1753 -> 193.1XX.XX1.6:1433 SYN *****S*
May 4 17:27:36 195.138.147.189:1757 -> 193.1XX.XX1.10:1433 SYN *****S*
May 4 17:27:36 195.138.147.189:1754 -> 193.1XX.XX1.7:1433 SYN *****S*
May 4 17:27:36 195.138.147.189:1755 -> 193.1XX.XX1.8:1433 SYN *****S*
May 4 17:27:37 195.138.147.189:1792 -> 193.1XX.XX1.45:1433 SYN *****S*
May 4 17:27:39 195.138.147.189:1812 -> 193.1XX.XX1.65:1433 SYN *****S*
May 4 18:13:09 193.126.160.199:2477 -> 193.1XX.XX1.2:1243 SYN *****S*
May 4 18:13:10 193.126.160.199:2476 -> 193.1XX.XX1.2:27374 SYN *****S*
May 4 18:13:09 193.126.160.199:2478 -> 193.1XX.XX1.3:27374 SYN *****S*
May 4 18:13:09 193.126.160.199:2479 -> 193.1XX.XX1.3:1243 SYN *****S*
May 4 18:13:09 193.126.160.199:2484 -> 193.1XX.XX1.6:27374 SYN *****S*
May 4 18:13:10 193.126.160.199:2485 -> 193.1XX.XX1.6:1243 SYN *****S*
May 4 18:13:10 193.126.160.199:2487 -> 193.1XX.XX1.7:1243 SYN *****S*
May 4 18:13:10 193.126.160.199:2486 -> 193.1XX.XX1.7:27374 SYN *****S*
May 4 18:13:08 193.126.160.199:2488 -> 193.1XX.XX1.8:27374 SYN *****S*
[...]
```

Nun kann über eine Whois-Abfrage nach der Herkunft des Port-Scans geforscht werden.

```
#whois 193.126.160.199
s is the RIPE Whois server.
% The objects are in RPSL format.
%
% Rights restricted by copyright.
% See http://www.ripe.net/ripenncc/pub-services/db/copyright.html

inetnum:      193.126.128.0 - 193.126.191.255
netname:      KQPT-IOL-DIALUP
descr:        KPN@west Portugal / IOL ISP
country:      PT
admin-c:      IOL5-RIPE
tech-c:        IOL5-RIPE
rev-srv:      ns.EUnet.pt
```

```

rev-srv: ns.EU.net
rev-srv: ns.ripe.net
status: ASSIGNED PA
mnt-by: AS1897-MNT
changed: Jorge.Frazao@KPNQwest.pt 20030217
source: RIPE

route: 193.126.128.0/18
descr: EUnet-PT
origin: AS1897
mnt-by: AS1897-MNT
changed: Jorge.Frazao@KPNQwest.pt 20031008
source: RIPE

role: KPNQwest Portugal IOL ISP Address Space Administrator
address: PORTUGAL
phone: +351 21 794 94 00
fax-no: +351 21 794 94 66
e-mail: abuse-iol@kpnqwest.pt
trouble: -----
trouble: Operational issues: <noc_at_kpnqwest.pt>
trouble: Abuse and SPAM: <abuse-iol@kpnqwest.pt>
trouble: -----
admin-c: JF74
tech-c: JF74
nic-hdl: IOL5-RIPE
mnt-by: AS1897-MNT
changed: Jorge.Frazao@kpnqwest.pt 20020104
source: RIPE

```

1.6.7 UDP-Verbindung

Bei der UDP-Verbindung wird eine zustandslose Netzwerkverbindung mit Datagrammen aufgebaut. Diese Datagramme werden an irgendeiner Stelle in das Netz eingespeist und an einer anderen Stelle vom Netz wieder abgefangen. Auf diese Weise ist es bei einem einzelnen UDP-Paket nicht möglich, die Herkunft dieses Pakets einwandfrei zu identifizieren. (Bei einem einzelnen TCP-Paket ist dieses auch nicht möglich, aber für den Aufbau einer TCP-Verbindung muss eine handshake- und zustandsbasierte Verbindung etabliert werden. Dies entfällt bei der UDP-Verbindung!) Um sich in den durchgeführten Angriff hinein zu versetzen zu können und somit ein tiefgehendes Verständnis für die Werkzeuge und Methoden der Angreifer zu entwickeln, werden hier ein paar tatsächlich von Skript-Kindern und manchen Pseudo-Hackern eingesetzte Tatwerkzeuge vorgestellt.

1.7 Tatwerkzeuge

Diese Tatwerkzeuge lassen sich von einschlägigen Internet-Adressen (siehe Anhang) herunterladen. Sie können am Honeypot-System entsprechend auf ihren Erfolg hin überprüft werden.

1.7.1 Mass-Router

Der sogenannte Mass-Router ist ein Werkzeug, welches zum Ziel hat, möglichst viele Rechner zu übernehmen, d.h. privilegierte Superuser-Rechte (Root-Rechte) auf Systemen zu erlangen, auf denen ein normaler Zugriff von außen nicht gestattet ist bzw. die über eine Schwachstelle verfügen. Diese Schwachstelle wird von dem Mass-Router entdeckt und ausgenutzt. Hierbei gehen die Mass-Routern nach dem "Gießkannen-Prinzip" vor. Es werden zahlreiche Pakete mit dem Ziel gesendet, ein routebares System zu ausfindig zu machen und es entsprechend auszunutzen.

Dabei benutzt der Mass-Router eine Vielzahl von Exploits, um massenhaft Root-Rechte von fremden Rechnern zu übernehmen. Diese werden dann als eine Art Sprungbrett für die Ausführung weiterer illegaler Aktivitäten genutzt.

Linux LPRng, Named & multi FPTD,SSHD,PHP,http RPC and Telnet mass scanner/rooter

```

Project started by daddy_cad from RHC
Greetingz to : Nessuno, Cho-Can, cxe,
amnesiax,buldozer ,tassadar ,mega-,M1
#rohackers, and to all my
friends i missed.
www.aboutthacking.net
05-October-2002

```

```

Vulnerable LPRng:
Red Hat 7.0 Guinness LPRng from RPM

```

```

Vulnerable BIND/named versions:
** 8.2 ** 8.2.1 ** 8.2.2 ** 8.2.2-P3 **
** 8.2.2-P5 ** 8.2.2-P7 ** 4.9.6-REL **

```

```

Vulnerable FTPD versions:
Wu-FTPD prior to 2.6.1 (anonymous login)

```

```

Vulnerable Ssh versions:
SSH-1.5-1.2.27
SSH-1.99-OpenSSH_2.2.0p1
and other 90 targets for SSHD exploit

```

```

Vulnerable RPC OS's:
Redhat 6.2
Redhat 6.1
Redhat 6.0

```

```

Vulnerable Telnet OS's:
FreeBSD 3.1, 4.0-REL, 4.2-REL, 4.3-BETA,
4.3-STABLE, 4.3-RELEASE
NetBSD 1.5
BSDI BSD/OS 4.1
Slackware 8.0
Telnetd 0.17 exploit
Irix
SunOS 5.7, 5.5, 5.5.1 and Solaris 2.6, 2.7 , 2.8

```

```

Vulnerable Pop3 OS's:
Linux x86

```

```

Vulnerable PHP version:
PHP/4.2.2

```

```

Vulnerable http version:
Null httpd 0.5.0 tested on Red Hat 7.3

```

```

* Usage:
* ./r00t <a>[.b][.c] <-d daemon>
* or
* ./r00t random-<class> <-d daemon>
* !!!ATTENTION When u use FTPD daemon don't put DOT between a and b
* use like this ex : ./r00t 80 96 -d 3 NO DOT between a and b!!!
* a,b,c = IP Classes
* class = a,b or c
* daemons :
* 1: bind
* 2: lpd
* 3: ftpd
* 4: ssh
* 5: rpc.*
* 6: telnet
* 7: mail
* 8: php
* 9: http
* 10: All xploits in one

```

1.7.2 Port-Scanner

Ein weiteres Tool, welches in der Skript-Kinder- und Hacker-Szene recht beliebt ist und darüber hinaus auch ebenso zum Auffinden von eigenen internen Schwachstellen eingesetzt werden kann, ist der sogenannte Port-Scanner. Bei dem Port-Scanner werden von außen erreichbare Kommunikationsports von den im Internet befindlichen Rechnern und Firewalls nach dem "Türklingel-Prinzip" abgeklopft: Wird ein solcher Port erkannt, erfolgt eine Meldung an das System. Mittels des Port-Scanners erfolgt jedoch noch kein Einspeisen von maliziösem Code oder ein Ausnutzen identifizierter

Schwachstellen. Ein Port-Scanner kann für eine bessere Vorbereitung eines Angriffs benutzt werden: Potentielle Schwachstellen werden ausfindig gemacht und können dann gezielt ausgenutzt werden.

1.7.3 NMAP

Mit dem freien Port-Scanner NMap lassen sich nicht nur offene Ports erkennen, sondern darüber hinaus mittels TCP-Fingerprinting auch Meta-Informationen vom Host wie beispielsweise Uptime und die Betriebssystem-Version.

Beispiel NMAP:

```
nmap -sT -O 193.xxx.xx1.6

Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2004-02-24 22:21 CET
Interesting ports on callisto.dnx.de (193.xxx.xx1.6):
(The 1648 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
53/tcp    open  domain
79/tcp    open  finger
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
514/tcp   open  shell
515/tcp   open  printer
631/tcp   open  ipp
993/tcp   open  imaps
Device type: general purpose
Running: Linux 2.4.X[2.5.X
OS details: Linux Kernel 2.4.0 - 2.5.20
Uptime 123.284 days (since Fri Oct 24 16:31:33 2003)

Nmap run completed -- 1 IP address (1 host up) scanned in 5.461 seconds
(-suj)-(Tue Feb 24 22:21:11)-<root@callisto>
#nmap -sT -O 193.xxx.xx1.16

Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2004-02-24 22:21 CET
Interesting ports on elara.dnx.de (193.xxx.xx1.16):
(The 1653 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1025/tcp  open  NFS-or-IIS
3389/tcp  open  ms-term-serv
5000/tcp  open  UPnP
Device type: general purpose
Running: Microsoft Windows NT/2K/XP
OS details: Microsoft Windows XP Professional

Nmap run completed -- 1 IP address (1 host up) scanned in 2.083 seconds
```

1.7.4 Schwachstellen-Scanner

Ein Schwachstellen-Scanner ist ein weiter entwickeltes Tool, welches auf einem Port-Scanner basiert. Hiermit werden die gefundenen Ports gezielt abgeklapert. Auf diese Weise sollen Schwachstellen, welche in einer Datenbank hinterlegt sind, mit ausfindig gemacht und anschließend weitergemeldet werden. Der Schwachstellen-Scanner kann auch vom Administrator inhouse für die Suche nach möglichen Schwachstellen verwendet werden. Diese können dann anschließend geschlossen werden, bevor ein Angreifer Gelegenheit hat, die Schwachstellen ausnutzen bzw. sich unberechtigt Zugriff auf das System zu verschaffen.

Beispiel: Nessus

2 Praktischer Aufbau von Systemen

Seit 1999 wurde eine Vielzahl von kommerziellen und nicht-kommerziellen Produkten bzw. Paketen für die Honey-Techniken entwickelt. Für die Erprobung in der

Praxis und die Vorbereitungen im Unternehmen bietet sich zunächst das Testen mit freien Produkten an. Das spart überdies Lizenzkosten.

2.0.5 Aufbau

Eine Voraussetzung für ein HoneyNet/Honeypot-System ist, dass die Verbindungen nicht durch Firewalls gefiltert werden. Ergänzend wurde auf einem zusätzlichen System das netzbasierte Intrusion-Detection-System Snort (www.snort.org) aufgesetzt und an einen weiteren Port des Hubs angebunden. Analog zum Sniffer belauscht Snort den Netzwerk-Traffic und analysiert ihn der Einfachheit halber anhand des Standard-Signatur-Regelsatzes der netzwerkbasieren Intrusion-Detection-Systeme.

Um mehr Kontrolle zu erhalten, bietet es sich an ein GIDS wie Hogwash als Gateway zu betreiben. Somit muss der gesamte Netzwerkverkehr eine zentrale Kontrollstelle passieren.

2.0.6 Timekeeping

Damit die einzelnen Ereignisse korreliert werden können, ist es notwendig für alle Komponenten des Systems, wie z.B. einen Typ-1 oder einen Typ-5-Honeypot ein Timekeeping-System einzuführen.

In der Praxis hat sich ein NTP-System als sinnvoll erwiesen. Dabei ist darauf zu achten, dass die geringeren Stratum-Server unbedingt durch die vom NTP unterstützten ACLs gegen eine Manipulation vom Honeypot-System abgesichert werden.

Beispiel:

```
#
# NTP configuration file for the NTP-Server
#
# Ignore all Requests
#

restrict default ignore

#
# Allow Administration via ntpq
#

restrict 127.0.0.1

#
# Network Clients
#

restrict 192.168.101.0 mask 255.255.255.0 nomodify # honey-net
restrict 192.53.103.103
restrict 192.53.103.104
server 192.53.103.103 #ntp1.ptb.de
server 192.53.103.104 #ntp2.ptb.de

fudge 127.127.1.0 stratum 10 # local clock is unsynchronized restrict
```

2.0.7 Eintrag im DNS und Hostname

Bei dem Eintrag des Netzes bzw. der Host-Adresse in den DNS-Server ist zu vermeiden, dass ein Angreifer auf den "wahren" Zweck dieses Systems schließen kann. In diesem Zusammenhang sollte auf die folgenden Punkte geachtet werden:

- Namen wie A-Records oder MX-Einträge wie honey.myzone.de sind zu vermeiden

- der Hostname darf nicht via SMTP Banner den Zweck des Rechners verraten

Aber auch wenn das System sich unter "honeypot.myzone.com" meldet, hält es die Masse der Angreifer nicht davon ab ihre Kenntnisse zu demonstrieren und uns den notwendigen Hintergrund zu liefern.

2.0.8 Süße Versuchung

Oft ist es sinnvoll, durch einen Portmapper oder einem scheinbar lohnenden Ziel weitere Anreize zu bieten. Gerade im Firmenumfeld können absichtlich platzierte falsche Dokumente auf einem Honeypot-System helfen, einen Innentäter zu ertappen und ihn auf diese Weise zu überführen.

2.0.9 Test-Werkzeuge von Angreifern

Nachdem ein System installiert worden ist, muss kontrolliert werden ob es seinen Zweck erfüllt. Zu diesem Zweck kann mit Angriffstools verifiziert werden ob:

- Informationen übermittelt werden, welche auf den eigentlichen Zweck des Systems schließen lassen
- der angebotene Port auch tatsächlich vorhanden ist
- Alarmierungssysteme und Kontroll- und Beobachtungseinrichtungen funktionieren
- durch eine Not-Aus-Funktion wie eine Firewall-Regel gewährleistet ist, dass bei einem Übernahmeversuch durch den Angreifer der Angriff weitestgehend auf die Honeypot-Zone beschränkt bleibt
- das System gegen die übrigen Produktionssysteme abgeschottet ist

2.0.10 Überwachung

Einen weiteren wichtigen Aspekt beim Aufbau eines Honeypot-Systems stellt die Überwachung im Themenkomplex Kontrolle und Beobachtung dar. Hierbei ist es notwendig, sämtliche Kommunikation und jedes außergewöhnliche Verhalten bei unserem zu kontrollierenden Laborsystem zu überwachen, zu dokumentieren und soweit der Kontrolle des Honeypot-Systems zu entziehen, dass ein Angreifer nicht diese Überwachungsmechanismen aushebeln oder gar umgehen kann. In der Praxis empfiehlt es sich einen Honeypot innerhalb eines UserMode Linux oder innerhalb einer VM, welche auf einem weiteren gehärteten System aufgesetzt ist, zu errichten. Dadurch hat man jederzeit die Möglichkeit den aktuellen Stand des Systems einzufrieren, zu analysieren und die Kommunikation über die Systemgrenzen zu kontrollieren. Bei der Überwachung haben sich die folgenden Tools als geeignet erwiesen:

UserMode Linux: Dieses erlaubt es, einen Linux-Kernel im Benutzerraum eines beschränkten Benutzers innerhalb einer Systemumgebung laufen zu lassen, so

dass der Angriff nur auf dem Unter-Prozess - sprich Tochter-Prozess - läuft, während der Vater-Prozess gesichert ist und die Überwachungsfunktion wahrnimmt. Des Weiteren besteht die Möglichkeit, mit TCPDump die gesamte Kommunikation zwischen dem Honeypot und der real world, also dem Internet, aufzuzeichnen, damit später eine Analyse durchgeführt werden kann. Mittels weiterer Methoden - wie dem Gateway-Intrusion-Detection-System à la Hogwash - ist es weiterhin möglich, einen Ausbruch oder einen maliziösen Code direkt zu filtern und somit den Angreifer mit einer unerwarteten Gegenwehr zu überraschen.

2.0.11 Notbremse

Einen äußerst wichtigen Stellenwert beim Betrieb eines Honeypot-Systems nimmt die Notbremse ein. Diese kann z.B. über ein einfaches Perl-Skript, welche eine IP-Rule etabliert, erfolgen. Sobald der Verkehr auf dem Honeypot-System schlagartig ansteigt oder versucht wird, andere Rechner in der DMZ von Honeypot-Systemen anzugreifen, wird die Notbremse aktiv. Auch wenn sich Checksummen von gewissen Systemfiles verändern oder andere Aktionen durchgeführt werden sollte eine Notbremsung erfolgen. Sie beabsichtigt exorbitante Kosten einzudämmen oder einen unbeabsichtigten Einbruch in die umgebende Systemsicherheit auszubremsen und somit den worst case, d.h. der Hacker erlangt Kontrolle über das Host-System vom Honeypot abzufangen.

2.0.12 Auswertung vom Rechnerinhalt - Festplatte

Nachdem ein Angriff erfolgt bzw. man sich nicht sicher ist ob bereits etwas geschehen ist, empfiehlt es sich eine Auswertung des Rechnerinhalts auf der Festplatte durchzuführen. Diese Auswertung kann nur von einem sogenannten Analyse-System, welches nicht im Honeypot-Umfeld lokalisiert ist, durchgeführt werden. Nachdem das ursprüngliche File-System - insofern es noch verfügbar ist - read-only gemountet worden ist, wird zunächst eine Integritätsüberprüfung der System-Binaries durchgeführt. Erst danach ist es sinnvoll weitere Analyseschritte wie, das Disamblieren vom Angreifer des eingespeisten binaries oder rootkits, oder auch die Signaturüberprüfung durchzuführen.

2.0.13 Auswertung von den Logfiles

Neben dem Auswerten der Daten des Honeypot-Systems empfiehlt es sich auch die Logfiles auszuwerten die z.B. auf einem anderen Kanal auf dem Host-System im uptime-Modus gesichert wurden. Diese Logfiles geben Aufschluss über Kommunikationsbeziehungen der Systeme untereinander sowie versuchter Angriffe.

2.0.14 Auswertung vom GIDS

Das Gateway-Intrusion-Detection-System fungiert als "Pfortner" zwischen unserem Honeypot-System und dem Internet. Auch das GIDS bietet die Möglichkeit Verkehrsbeziehungen und Angriffsversuche zu dokumentieren und auszuwerten. Nachdem die Stufen top-down vom GIDS bis zum Logfile besprochen sind, bietet es sich als weitere Möglichkeit an den vom TCP-Dump mitgeschriebenen "rohen" Netzwerkverkehr nach auffälligen Stellen zu überprüfen, die sich anhand der Logfiles und des Gateway-Intrusion-Detection-Systems auch anschließend tiefer analysieren lassen. Die eigentliche Analyse basiert dann auf dem Extrahieren der ausführbaren Programme bzw. der versuchten Exploits. Diese können auf der einen Seite anhand des mitgeschriebenen Netzwerkverkehrs - z.B. mit isa-rail - extrahiert werden oder auf der anderen Seite als manipulationsfähige Dateien aus dem File-System gefunden werden. Hier schließt sich eine explizite Laboranalyse mit einem Engineering Disexploit an.

2.0.15 Auswertung vom Netzwerkverkehr

Damit der Netzwerkverkehr ausgewertet werden kann ist es sinnvoll diesen im PCAP-Format zu speichern. Eine Analyse kann mit freien Tools wie EtherReal oder TCPDUMP durchgeführt werden.

Verdächtige Stellen im Datenstrom des Netzwerkverkehrs werden festgestellt durch:

- Einträge im Syslog
- Meldungen von der Firewall, Verkehrsänderungen
- durch das GIDS
- durch markiertes Verhalten

Nachdem diese Stellen gefunden sind, kann nun eine Analyse der Pakete z.B. mit TCPDUMP erfolgen.

```
02:38:41.523741 67.75.67.147.5625 > 193.108.181.24.8080: S 674719801:674719801(
03:21:43.720708 67.75.67.147.5625 > 193.108.181.42.8080: S 674719801:674719801(
03:39:30.089344 216.218.158.87.1057 > 193.108.181.8.1434: udp 376
04:16:08.657626 62.135.27.130.1652 > 193.108.181.8.1434: udp 376
04:24:02.581537 209.67.2.126 > 193.108.181.6: icmp: host 165.251.41.31 unreachable
04:24:06.503391 209.67.2.126 > 193.108.181.6: icmp: host 165.251.41.131 unreachable
04:24:10.571422 209.67.2.126 > 193.108.181.6: icmp: host 165.251.41.31 unreachable
04:25:01.337934 209.67.2.126 > 193.108.181.6: icmp: host 165.251.41.131 unreachable
04:25:09.368497 209.67.2.126 > 193.108.181.6: icmp: host 165.251.41.131 unreachable
05:20:17.966979 209.67.2.126 > 193.108.181.6: icmp: host 165.251.41.131 unreachable
05:20:26.006505 209.67.2.126 > 193.108.181.6: icmp: host 165.251.41.131 unreachable
05:37:17.807893 200.203.120.200.3206 > 193.108.181.6.1434: udp 376
05:53:27.096218 68.32.17.41.4302 > 193.108.181.42.1434: udp 376
06:54:20.742452 68.226.158.136.2789 > 193.108.181.6.80: P 3880322941:3880322970
ack 4276001828 win 64240 (DF)
07:02:42.270314 216.218.158.87.1057 > 193.108.181.42.1434: udp 376
07:05:53.159239 66.196.90.216.44526 > 193.108.181.6.80: P 3278756138:3278756346
ack 689530510 win 5840 <nop,nop,timestamp 39383
459 74806124> (DF)
```

2.0.16 Analyse

2.1 SMTP - Honeypot

Aufgabe eines SMTP Honeypots ist es Versuche von Spammern zu erkennen, ihre Werbemails zu relayen bzw. eine Mail über dieses System auszuliefern.

Die Quelle der IP-Adressen von den Spam-Sendern werden in eine DNS-Zone eingetragen, sodass diese

Systeme über eine Realtime-Blockierliste von den eigentlichen Produktiv-MTAs blockiert werden können.

Durch ein gezieltes Platzieren der SMTP-Sensoren ist es möglich, neue SPAM-Sender schnell zu erkennen und auszuschalten.

2.1.1 Systemaufbau

Direkt an das Internet ist ein Debian GNU/Linux System installiert, auf dem der Exim 3.0 MTA als Pseudo-MTA fundiert.

Dabei ist die MTA Konfiguration wie folgt zu ändern:

2.1.2 Open-Mailrelay vortäuschen

Durch diese Option akzeptiert der MTA jeden Relayversuch zu einer gültigen EMail-Adresse. Nicht konnetzte Zonen werden dennoch abgewiesen.

```
# The setting below allows your host to be used as a mail relay only by
# localhost: it locks out the use of your host as a mail relay by any
# other host. See the section of the manual entitled "Control of relaying"
# for more info.
```

```
host_accept_relay = *
```

Dann muss bei den Transports ein neuer Transport installiert werden der dann den Remote-SMTP ersetzen kann.

```
#####
# TRANSPORTS CONFIGURATION
#####
```

```
# This transport delivers all Mail to a specific Maildir, it can also
# write all mail to /dev/null if you don't like to store them.
```

```
forensic_delivery:
driver = appendfile
directory = /var/spool/hsmtp/mail
delivery_date_add
maildir_format
envelope_to_add
return_path_add
user = mail
group = mail
mode = 0660
```

Nun muss der Mail-Router von remote.smtp auf den neuen forensic_delivery Transport umgestellt werden.

```
#####
# ROUTERS CONFIGURATION
# Specifies how remote addresses are handled
#####
# ORDER DOES MATTER
# A remote address is passed to each in turn until it is accepted.
```

```
# Remote addresses are those with a domain that does not match any item
# in the "local_domains" setting above.
```

```
# This router routes to remote hosts over SMTP using a DNS lookup with
# default options.
```

```
lookuphost:
driver = lookuphost
transport = forensic_delivery
```

```
# This router routes to remote hosts over SMTP by explicit IP address,
# given as a "domain literal" in the form [nnn.nnn.nnn.nnn]. The RFCs
# require this facility, which is why it is enabled by default in Exim.
# If you want to lock it out, set forbid_domain_literals in the main
# configuration section above.
```

```
literal:
driver = ipliteral
transport = forensic_delivery
```

```
end
```

Jetzt wird jede EMail nicht von unserem SMTP-Honeygot ausgeliefert, wie es der SPAMMER erwarten würde, sondern wird entsorgt und der Angreifer kann in unsere DNS-Datenbank geschrieben werden.

2.1.3 Live-System

Schon nach wenigen Minuten werden die ersten Systeme versuchen, ihre Werbebotschaften über den Open-Relay zu verteilen. Im Exim-Logfile ist dann so etwas oder etwas ähnliches zu lesen:

```
2004-02-25 14:39:55 1AvzGU-0001jG-00 <= session12002@yahoo.com
H=slip-12-65-114-33.mis.prserv.net (mx2.mail.yahoo.com) [12.65.114.33 ]
P=esmtlp S=831
id=049057051046049048056046049056049046049048050@mx2.mail.yahoo.com
2004-02-25 14:39:55 1AvzGU-0001jG-00 => smtps1@cox.net R=lookuphost
T=Forensic_delivery H=mx.east.cox.net
2004-02-25 14:39:55 1AvzGU-0001jG-00 Completed
[...]
```

3 Wireless Technologie

Im Gegensatz zu den drahtgebundenen Techniken stellt die Wireless-Technologie eine weitere Herausforderung für den Betrieb von Honeygot-Systemen dar. Über einen weiteren physikalischen Layer, den sogenannten Connection-Layer, können sich andere Beteiligte in der Reichweite des Umfeldes mit in das Netz einspeisen, ohne dass sie physikalischen Zugang zu irgendwelchen Kommunikationsleitungen oder Betriebsgeländen haben. Das sogenannte wardriving, d.h. suchen nach Wireless Access Points ohne Verschlüsselung und den Versuch in diese einzubrechen, führt zu einem weiteren Problem im Umfeld der Honeygot's. Gerade hier hat sich gezeigt, dass durch einen gezielten Aufbau von Honeygot-Systemen und Systemen die einen entsprechenden Wardriver irreführen, sich doch die entsprechenden Infrastrukturen auf geeignete Weise schützen lassen.

3.1 Fake-AP

Fake-AP ist nun ein Userspace-Programm welches es ermöglicht, aufgrund eines permanenten Aussendens gefälschter Access-Point-Informationen, eine Fülle von Access-Points vorzutäuschen welche dem WarDriver als reelle Maschinen erscheinen. Versucht er sich nun auf diesen Access-Point zu konnektieren, kann durch einen zweiten installierten Sniffer oder ein wireless Honeygot frühzeitig erkannt werden ob ein Angreifer oder WarDriver versucht oberhalb des Perimeters meines wireless Intrusion-Detection-Systems oder meiner wireless Infrastruktur in interne Systeme einzudringen.

3.2 Honeyd - Ein Honeynetz

Eine weitere Möglichkeit zum Absichern von wireless und drahtgebundenen Netzen besteht mit sogenannten Honeynetzen. Im Gegensatz zu einem Honeygot wird bei einem Honeynetz die gesamte Infrastruktur inklusive Schwachstellen simuliert, wobei auch bei dieser Simulation Auswertungsskripte bzw. Aktionen wie z.B.

das Sperren von Firewall Ports oder das Alarmieren ausgeführt werden kann.

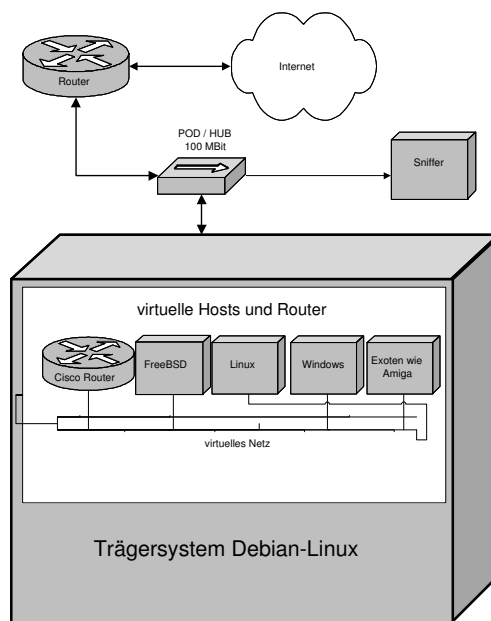
Als abschließendes Beispiel wird daher für den praktischen Einsatz die freie Honeynet-Software Honeyd von Niels Provos gewählt (www.citi.umich.edu/u/provos/honeyd/).

3.2.1 Anforderungen

Minimalanforderungen für ein innerhalb kurzer Zeit einzurichtendes, funktionsfähiges System sind eine PIII-basierte Hardware, (ca 1Ghz), eine Netzwerkkarte, eine Debian GNU/Linux-Distribution, Honeyd, Snort und mindestens zwei öffentlich konnektierte IPs (besser noch ein Subnetz).

3.2.2 Aufbau

Als Trägersystem dient ein Linux-System, das als Minimal-Installation aufgesetzt wurde. Auf Layer-3-Ebene sind vom Trägersystem zum Netzwerk hin keine Ports offen, sodass das System aus externer Sicht ein relativ hohes Sicherheitsniveau zeigt. Es ist über einen 100-MBit-Hub angebunden, zudem lauscht an einem Port ein Sniffer. Damit lässt sich der gesamte Netzwerkverkehr (Internet-Traffic) zwischen Honeynet und dem Internet mitschneiden. Zu dem Debian-Trägersystem wird nur ein 28er Netz geroutet, damit eine Vielzahl von Hosts mit verschiedenen Sicherheitslücken abbildbar sind.



Auf dem Debian-System installiert man die aktuelle Version des Honeygot-Daemon Honeyd. Dieser ist in der Lage, ganze Netze mit verschiedenen Hosts zu emulieren und auf Pings und andere Aktionen zu antworten. (Details hierzu sie-

he: <http://www.citi.umich.edu/u/provos/papers/honeyd-abstract.pdf>)

In dieser Konstellation kann zu jedem TCP- und UDP-Port eines virtuellen Hosts eine Aktion ausgelöst werden, die von einem einfachen RST-Flag zum Beenden einer Verbindung bis hin zum Starten eines Scripts – das eine Applikation mit einer Sicherheitsschwachstelle simuliert und den daraus resultierenden Angriff auf Application-Layer-Ebene mitschneidet – reichen kann.

Damit sowohl Security-Scanner wie Nmap als auch kommerzielle Schwachstellenscanner in die Irre geführt werden, emuliert Honeyd den Stack und das Socket-Verhalten eines bestimmten Betriebssystems was in der Konsequenz dazu führt, dass beispielsweise durch Fingerprinting oder Uptime-Guessing gelieferte Informationen keinen Verdacht bei einem Angreifer (oder Einbruchstester) erregen. Eine interessante Nebenerkenntnis ist, dass Honeyd dieselben Daten wie Nmap benutzt um das Fingerprinting nachzustellen.

Die Applikationen auf dem virtuellen Host werden emuliert. Ferner besteht die Möglichkeit das Routing mit einer gewissen Verzögerung so nachzubilden, dass sich auch komplexe Netzstrukturen nach außen virtuell abbilden lassen.

Hinter den zu beobachtenden Ports liegen Scripte, die im allgemeinen in Perl, Python oder als Shell-Scripte realisiert sind. Alle beim Honeyd ein- und ausgehenden Datenpakete werden vollständig mitgeschnitten um das Verhalten des Angreifers und die Art des durchgeführten (erfolgreichen beziehungsweise erfolglosen) Angriffs aufzuzeichnen und nachzuvollziehen.

Wird der Honeyd während seines Betriebes Ziel einer bis dato unbekanntem technischen Schwachstelle (Exploit), kann man eine Codeanalyse der Datenpakete auf Basis dieser Mitschnitte (Captures) durchführen – was in letzter Konsequenz sogar zur Identifikation des Programmierers führen kann.

3.2.3 Honeyd-Konfiguration

Nach dem Kompilieren muss der Honeyd auf das zu emulierende virtuelle Netz eingestellt werden. Wenn der Honeyd an einem Ethernet-Segment lauscht, wird der Linux-Userspace "arpd" genutzt damit eine Antwort auf Requests durchgeführt wird. Besser ist es eine "Netroute" (statisch) auf das Trägersystem des "Honeyd" zeigen zu lassen.

```
# Beispielkonfiguration des Honeyd-Daemons  
mit Anmerkungen
```

```
# Template für Windows 2K erstellen  
create windows  
set windows personality "Windows Millennium  
Edition (Me), Win 2000, or WinXP"  
add windows tcp port 80 "sh /usr/local/  
scripts/web.sh $ipsrc  
$dport"  
add windows tcp port 6588 "sh /usr/local/  
scripts/web.sh  
$ipsrc $dport"
```

```
set windows default tcp action reset
```

```
# Template für CISCO erstellen  
create cisco  
set cisco personality "Cisco 7206 running IOS 11.1(24)"  
add cisco tcp port 23 "/usr/bin/perl  
/usr/local/scripts/router-telnet.pl"  
set cisco default tcp action reset
```

```
# Template für FreeBSD  
create freebsd  
set freebsd personality "FreeBSD 4.3 -- 4.4-RELEASE"  
add freebsd tcp port 21 "sh /usr/local/scripts/ftp.sh  
$ipsrc $dport"  
add freebsd tcp port 25 "sh /usr/local/scripts/sntp.sh  
$ipsrc $dport"  
add freebsd tcp port 22 "sh  
/usr/local/scripts/scripts/test.sh $ipsrc $dport"  
add freebsd tcp port 110 "sh  
/usr/local/scripts/emulate-pop3.sh $ipsrc $dport"  
add freebsd tcp port 80 "sh /usr/local/scripts/web.sh  
$ipsrc $dport"  
add freebsd tcp port 8080 "sh /usr/local/scripts/web.sh  
$ipsrc $dport"  
add freebsd tcp port 3128 "sh /usr/local/scripts/web.sh  
$ipsrc $dport"  
add freebsd tcp port 6588 "sh /usr/local/scripts/web.sh  
$ipsrc $dport"  
set freebsd default tcp action reset
```

```
# Binden des Templates an die IP-Adresse des virtuellen  
Hosts  
bind 10.1.2.178 openbsd  
bind 10.1.2.181 windows  
bind 10.1.2.182 windows  
bind 10.1.2.183 windows  
bind 10.1.2.184 windows  
bind 10.1.2.185 freebsd  
bind 10.1.2.186 freebsd  
bind 10.1.2.187 freebsd  
bind 10.1.2.188 freebsd  
bind 10.1.2.189 cisco
```

Literatur

- [1] Süße - Falle, Honey-Techniken zur Einbruchsvorsorge – Lukas Grunwald, Jochen Schlichting <http://www.heise.de/ix>, (2003).
- [2] Honeyd - Network Rhapsody for You <http://www.citi.umich.edu/u/provos/honeyd>, (2004).
- [3] Running a wireless Honeyd on CeBIT 2003 <http://www.phreak.de/cebit2003>, (2003).
- [4] Honeyd Project <http://project.honeyd.org>, (2004).
- [5] Packet Storm <http://www.packetstormsecurity.nl>, (2004).
- [6] Exim MTA <http://www.exim.org>, (2004).
- [7] Fake Access Point <http://www.blackalchemy.to/project/fakeap>, (2004).