



24 Aufbau und Konfiguration eines »echten« Honeypots

Dieses Kapitel beschäftigt sich mit dem Aufbau und der Überwachung eines selbstgebauten Honeypots. Dabei handelt es sich um einen kompletten Rechner, der zur leichten Überwachung und Wartung als virtueller Rechner implementiert wird. Zur Virtualisierung kann sowohl das kommerzielle Produkt *VMware Workstation* oder das Open-Source-Produkt *UserModeLinux* verwendet werden. Die Konfiguration beider Produkte wird kurz vorgestellt. Das Hauptaugenmerk in diesem Kapitel liegt jedoch auf der Überwachung und Sicherung dieser Rechner. Sie dürfen keine zusätzliche Gefahr für das restliche Netzwerk oder andere Benutzer im Internet darstellen.

24.1 Auswahl und Installation des Betriebssystems

Die Auswahl des Betriebssystems ist essenziell für den Honeypot. Schließlich soll ein Angreifer dieses Betriebssystem erkennen, angreifen und einbrechen. Wenn also das Verhalten von Angreifern auf Microsoft Windows 2000 studiert werden soll, macht es selbstverständlich keinen Sinn, SUSE Linux zu installieren. Das Betriebssystem sollte also entsprechend der Zielgruppe der zu analysierenden Angreifer gewählt werden.

Wurde in diesem ersten Schritt die Entscheidung für ein Linux-Betriebssystem getroffen, so sollte im zweiten Schritt die Distribution gewählt werden. Diese Wahl ist meist recht einfach, da das eigene Interesse einer Distribution gilt. Viele Sicherheitslücken existieren jedoch nur in der einen oder anderen Distribution. Dies hängt zum einen mit unterschiedlichen Optionen bei der Übersetzung der Programme, aber auch mit vollkommen unterschiedlich installierten Programmen zusammen. So verwendete SUSE Linux lange Jahre den Proftpd als Standard-FTP-Server, während Red Hat den WU-ftp verwendetete.

Schließlich sollte die Version der Distribution gewählt werden. Hier sollte die gewählte Distribution wieder möglichst den zu studierenden Zielsystemen entsprechen. Jedoch ist zu bedenken, dass bei einem Red Hat 8 unter Umständen direkt nach der Veröffentlichung noch keine Sicherheitslücke bekannt ist. Dieser Rechner eignet sich daher gut für die Sammlung und das Studium neuer Angriffe. Wenn jedoch all-

gemein Angriffe studiert werden sollen, so ist es sicherlich sinnvoller, eine ältere Version zu wählen, bei der bereits mehrere Lücken entdeckt und Werkzeuge entwickelt wurden. Im Falle von Red Hat Linux wird daher gerne die Version 6.2 oder 7.x gewählt.

Red Hat Linux 6.2 wies in der Standardinstallation mehrere verwundbare Netzwerkdienste auf. Hierfür existieren bereits eine ganze Reihe von Werkzeugen, die die Ausnutzung dieser Dienste ermöglichen. Damit bietet sich Red Hat Linux 6.2 als Plattform für einen Honeypot an.

Bei der Installation sollten folgende Punkte beachtet werden, um eine spätere forensische Analyse zu vereinfachen:

- Die verwendeten Festplatten oder Festplattenpartitionen sollten vor der Installation gelöscht werden. Werden bereits gebrauchte Festplatten eingesetzt, so befinden sich auf diesen bereits eine ganze Menge an Informationen. Einerseits dürfen diese Informationen möglicherweise nicht dem Angreifer in die Hände fallen, andererseits besteht die Gefahr, dass bei einer späteren forensischen Analyse diese Daten zu Verwirrungen führen. Es ist ein weit verbreiteter Irrglaube, dass bei der Formatierung von Festplatten diese gelöscht werden. Meist werden nur die Verwaltungsinformationen des Dateisystems auf die Festplatte übertragen. Eine Löschung kann unter Linux mit dem folgenden Befehl erfolgen: `dd if=/dev/zero of=/dev/hdaX`.
- Handelt es sich bei dem Honeypot um eine Linux-Distribution, so sollte diese so konfiguriert werden, dass die Schreiboperationen für die Festplatten synchron erfolgen. So befinden sich immer die aktuellen Daten auf den Festplatten. Ein Abschalten des Rechners führt daher nicht zu Datenverlust durch fehlende Informationen in den Festplattencaches. Hierzu sollte die Datei `/etc/fstab` so modifiziert werden, dass die Mounthoptionen den Eintrag `sync` tragen:

```
/dev/hda5    /home      ext2    rw,suid,dev,exec,auto,nouser,sync 0 2
```

24.2 Verwendung von UserModeLinux oder VMware als virtueller Rechner

Um den Honeypot optimal überwachen zu können, kann am besten der Honeypot als virtuelles System implementiert werden. Das Gastgeber-Betriebssystem kann dann die Netzwerkverbindungen überwachen, kontrollieren und protokollieren. Jedoch sollte sich der Administrator des Honeypots immer bewusst sein, dass der Einbrecher möglicherweise in der Lage ist, aus dem virtuellen Betriebssystem auszubrechen, wenn die Virtualisierung Sicherheitslücken aufweist. Eine derartige Sicherheitslücke existierte bereits bei UML.

Für diese Virtualisierung stehen zurzeit zwei verschiedene Möglichkeiten zur Wahl:

- VMware Workstation 3.1 ist ein kommerzielles Produkt von VMware (<http://www.vmware.com>) und hat einen Preis von etwa € 300,-. Es simuliert einen kompletten Rechner mit BIOS, Arbeitsspeicher, Festplatten, Netzwerkkarten etc. Die Performanz dieses simulierten Rechners ist bei Verzicht auf die grafische Oberfläche sehr gut. Jedoch sollte das Gastgebersystem über genügend Hauptspeicher verfügen. 256 Mbyte Arbeitsspeicher ist ein guter Ausgangswert. Leider ist ein Angreifer in der Lage, nach einem Einbruch recht einfach diese Tatsache festzustellen.
- UserModeLinux ist ein Open-Source-Produkt, verfügbar unter <http://user-mode-linux.sourceforge.net/>. Es handelt sich um einen Linux-Kernel, der so modifiziert wurde, dass er im Userspace booten kann. Sämtliche weiteren benötigten Programme und Dienste befinden sich auf einem Dateisystem, welches in Form einer Datei dem Kernel übergeben wird. Anschließend ist ein vollkommen funktionsfähiges Linux vorhanden. Die benötigten Ressourcen sind wesentlich geringer als bei VMware. Usermode Linux ist in der Lage, auf einem Mobile Pentium I mit 266 MHz und 128 Mbyte Arbeitsspeicher bei akzeptabler Geschwindigkeit zwei virtuelle Linux-Systeme zu simulieren.

24.2.1 Installation und Konfiguration von VMware

Die Installation von VMware erfolgt recht einfach mithilfe eines RPM. Dieses RPM kann von der Homepage <http://www.vmware.com> als Evaluationsversion geladen werden. Anschließend ist ein Evaluationsschlüssel für 30 Tage verfügbar. Nach Ablauf dieses Zeitraums kann dieser einige Male verlängert werden. Schließlich ist jedoch eine Registrierung und ein Kauf der Software erforderlich.

VMware ist eine grafische Software, die vor dem ersten Start zunächst konfiguriert werden muss. Vor dieser Konfiguration sollte bereits grundsätzlich die zu implementierende spätere Netzwerkstruktur, in die der HoneyPot eingebettet werden soll, bekannt sein.

VMware bietet drei verschiedene Netzwerkmodi an:

- *Bridged*. Der virtuelle Rechner nutzt die physikalische Netzwerkkarte des Gastgeber-Rechners. Der Gastgeber-Rechner kann die Netzwerkverbindungen nicht einschränken.
- *NAT*. VMware erzeugt automatisch Network Address Translation-Regeln und ermöglicht damit dem virtuellen Rechner den Zugang auf das Netzwerk.
- *Host-Only*. Der virtuelle Rechner befindet sich in einem eigenen Netzwerk mit dem Gastgeber-(Host-)Rechner. Es existiert keine Verbindung zum physikalischen Netzwerk.

Für die Implementation eines HoneyPots wird idealerweise die letzte Variante gewählt. Hier hat der HoneyPot keine unerwünschte direkte Verbindung zum physika-

lischen Netz. Der Gastgeber-Rechner kann die Netzwerkverbindungen überwachen, kontrollieren und einschränken.

Um nun VMware-Workstation zu installieren, sind die folgenden Befehle erforderlich:

```
# rpm -ivh VMwareWorkstation-<version>.i386.rpm
Preparing...                               ##### [100%]
   1:VMwareWorkstation                     ##### [100%]
[root@redhat root]# vmware-config.pl
Making sure VMware Workstation's services are stopped.
```

```
Stopping VMware services:
  Virtual machine monitor                    [ OK ]
```

You must read and accept the End User License Agreement to continue.
Press enter to display it.

```
END USER LICENSE AGREEMENT
FOR VMWARE(tm) DESKTOP SOFTWARE PRODUCT
```

---Lizenz gekürzt---

Do you accept? (yes/no) **yes**

Thank you.

Trying to find a suitable vmmon module for your running kernel.

None of VMware Workstation's pre-built vmmon modules is suitable for your running kernel. Do you want this script to try to build the vmmon module for your system (you need to have a C compiler installed on your system)? [yes] **yes**

What is the location of the directory of C header files that match your running kernel? [/lib/modules/2.4.<version>/build/include] **<Enter>**

Extracting the sources of the vmmon module.

Building the vmmon module.

---gekürzt---

The module loads perfectly in the running kernel.

Trying to find a suitable vmnet module for your running kernel.

None of VMware Workstation's pre-built vmnet modules is suitable for your running kernel. Do you want this script to try to build the vmnet module for your system (you need to have a C compiler installed on your system)?

↳ [yes] <Enter>

Extracting the sources of the vmnet module.

Building the vmnet module.

---gekürzt---

The module loads perfectly in the running kernel.

Do you want networking for your Virtual Machines? (yes/no/help) [yes] <Enter>

Configuring a bridged network for vmnet0.

Configuring a NAT network for vmnet8.

Do you want this script to probe for an unused private subnet? (yes/no/help) [yes] <Enter>

Probing for an unused private subnet (this can take some time).

The subnet 192.168.188.0/255.255.255.0 appears to be unused.

Press enter to display the DHCP server copyright information.

---gekürzt---

Do you want to be able to use host-only networking in your Virtual Machines? [no] yes

Configuring a host-only network for vmnet1.

Do you want this script to probe for an unused private subnet? (yes/no/help) [yes] no

What will be the IP address of your host on the private network? 10.0.1.1

What will be the netmask of your private network? 255.255.255.0

The following hostonly networks have been defined:

. vmnet1 is a host-only network on subnet 10.0.1.0.

Do you wish to configure another host-only network? (yes/no) [no] no

Do you want this script to automatically configure your system to allow your Virtual Machines to access the host's filesystem? (yes/no/help) [no] no

Starting VMware services:

```
Virtual machine monitor           [ OK ]
Virtual ethernet                  [ OK ]
Bridged networking on /dev/vmnet0 [ OK ]
Host-only networking on /dev/vmnet1 (background) [ OK ]
Host-only networking on /dev/vmnet8 (background) [ OK ]
NAT networking on /dev/vmnet8    [ OK ]
```

The configuration of VMware Workstation <version> for Linux for this running kernel completed successfully.

You can now run VMware Workstation by invoking the following command:
"/usr/bin/vmware".

Enjoy,

--the VMware team

Nun kann VMware gestartet werden. Beim ersten Start ist es erforderlich, eine Lizenznummer einzugeben. Diese kann als Evaluationslizenz bei VMware angefordert werden. Anschließend wird mit dem Wizard die Konfiguration des virtuellen Rechners vorgenommen. Hierbei ist es wichtig, die Option *Host-Only Networking* auszuwählen.

Besonders wichtig ist jedoch die Wahl der Festplattensimulation. Hier sollten für forensische Zwecke physikalische Festplattenpartitionen gewählt werden. Diese lassen sich später wesentlich leichter offline bearbeiten und analysieren. Außerdem können durch Vergrößerung und Verkleinerung der virtuellen VMware-Festplattendatei Daten des Gastgebersystems nicht als Müll auf der virtuellen Festplatte auftauchen. Jedoch können aktuelle Versionen der VMware Workstation die virtuellen Festplatten als Datei direkt komplett anlegen und nutzen. Dann besteht die Möglichkeit, diese Dateien unter Linux zu mounten.



Tipp:

Bevor jedoch die ausgewählte Partition durch VMware genutzt wird, sollte diese gelöscht werden. Das kann sehr komfortabel mit dem Befehl

```
dd if=/dev/zero of=/dev/hdaX
```

erfolgen. Es erleichtert die forensische Analyse nach einem Einbruch ungemein.

Was nun noch bleibt, ist die Installation des Betriebssystems. Dazu wird ein bootfähiger Datenträger (zum Beispiel CD) eingelegt und die virtuelle Maschine durch Betäti-

gen der Taste POWER ON gestartet. Es bootet nun ein kompletter virtueller Rechner und startet die Installation.



Tipp:

VMware fängt die Maus ein. Wenn scheinbar der Mauszeiger verloren gegangen oder nicht aus dem VMware-Bildschirm zu bewegen ist, hat VMware diesen eingefangen. Es gibt ihn wieder frei, wenn STRG-ALT gedrückt wird.

Die Verwendung von VMware bietet zusätzlich die folgenden Vorteile:

- **Zugriff auf die Dateisysteme.** Wenn das VMware Guest-Betriebssystem auf einer eigenen physikalischen Partition installiert wurde, so kann diese Partition gleichzeitig von dem Host-System read-only gemountet werden. Damit kann das Host-System in Echtzeit verfolgen, was auf dem Guest-System passiert. Hierzu können die *e2tools* verwendet werden (http://home.earthlink.net/~k_sheff/sw/e2tools/index.html), wenn es sich um ein *ext2fs*-Dateisystem handelt.

Auf meine Anregung hin hat Keith W. Sheffield die *e2tools* um eine *e2tail*-Variante erweitert. Dieser Befehl versteht auch den Modus `-f`, `--follow`.

Auch wenn dieser Zugriff in Echtzeit nicht gewünscht wird, so kann das Dateisystem des Honeypots vorübergehend read-only gemountet und von Tripwire überprüft werden. Hierbei werden aber Veränderungen des Dateisystems während des read-only mounts nicht erkannt. Leider ist mir keine Möglichkeit bekannt, während des read-only mounts den Lesecache des Linux-Kernels zu deaktivieren.

Listing 24.1 Beispielaufwurf der *e2tools*

```
# ./e2ls -l /dev/sda1:var/log/messages
77024 100600 0 0 226193 6-Aug-2002 01:08 messages
```

Selbst wenn das System sich in einem nicht bootfähigen Zustand befindet, besteht die Möglichkeit, das System zu untersuchen. Wird jedoch die VMware-virtuelle Disk gewählt, so besteht diese Möglichkeit nicht, da VMware ein proprietäres Format für die Speicherung der Festplatten gewählt hat.

- **Suspendmodus.** Der Suspend von VMware speichert den virtuellen Arbeitsspeicher in einer Datei **.vmss* ab. Diese Datei kann mit den Befehlen `strings` und `grep` untersucht werden.

24.2.2 Installation und Konfiguration von UserModelLinux

Die Installation von UserModelLinux (UML) ist eigentlich sehr einfach. Hierzu wird zunächst die Software von ihrer Homepage geladen. Dort ist der gepatchte Linux-Kernel als RPM-Paket verfügbar. Dieses ist jedoch leider häufig veraltet, so dass Sie eine manuelle Installation vorziehen sollten.

Sie benötigen dann zunächst das Paket `uml_utilities_<version>.tar.bz2` von der UML-Homepage (http://prdownloads.sourceforge.net/user-mode-linux/uml_utilities_<version>.-tar.bz2). Dieses sollten Sie herunterladen, auspacken und installieren.

Anschließend wählen Sie einen Kernel von <http://www.kernel.org> und einen UML-Kernel-Patch von <http://prdownloads.sourceforge.net/user-mode-linux/>.

```
# cd /usr/local/src
# tar xzf /pfad/linux-<version>.tar.gz
# cd linux-<version>
# bzcat /download/uml-patch-<version>.bz2 | patch -p1
```

Wurde bereits im Vorfeld ein UML-Kernel installiert, so kann nun ein kleiner Trick angewendet werden, um eine Startkonfiguration zu erhalten. Der folgende Befehl liest die Konfiguration aus dem installierten UML-Kernel aus und schreibt sie in der lokalen Datei `.config`. Diese kann nun angepasst werden.

```
# linux --showconfig > .config
```

Konfigurieren Sie nun den UML-Kernel, wie Sie wünschen. Folgende Parameter sind für den Einsatz als Honeypot interessant:

- *Separate Kernel Address Space* (CONFIG_SKAS). Dies erhöht stark die Sicherheit eines UML-Kernels. Diese Funktion erfordert einen Patch des Host-Kernels.
- *Honeypot Proc Filesystem* (CONFIG_HPPFS). Dies ermöglicht die Steuerung des Proc-Dateisystems des UML-Kernels vom Host-System aus.
- *TTY-Logging* (CONFIG_TTY_LOG). Damit sind Sie in der Lage, alle Sessions, die auf dem UML-Kernel stattfinden, auf dem Hostsystem zusätzlich zu protokollieren. Hiermit können Sie auch SSH-Sitzungen in Klartext überwachen.

Sinnvollerweise schalten Sie die Unterstützung für ladbare Kernel-Module ab. Dies erleichtert die spätere Installation stark. Anschließend rufen Sie bitte folgenden Befehl auf:

```
# make oldconfig ARCH=um
```

Die Angabe von `ARCH=um` ist bei allen `make`-Aufrufen erforderlich. Ansonsten schlägt die Übersetzung fehl. Wenn eine grafisches Werkzeug gewünscht wird, so kann dies aufgerufen werden mit:

```
# make xconfig ARCH=um
oder
# make menuconfig ARCH=um
```

Anschließend kann der UML-Kernel übersetzt werden mit:

```
# make linux ARCH=um
```

Der neue Kernel sollte nun in den Pfad kopiert werden, z.B.:

```
# cp linux /usr/bin
# chmod 755 /usr/bin/linux
```

Da die Module bei der Übersetzung deaktiviert wurden, brauchen sie nicht übersetzt und installiert zu werden.

Nun kann der Linux-Kernel bereits gestartet werden. Jedoch fehlt noch das zu bootende Dateisystem. Hier wird das Linux-Dateisystem benötigt, welches idealerweise in Form einer Datei vorliegt. Auf der Homepage von Usermode Linux werden unterschiedliche Dateisysteme zum Download angeboten. Jedoch sollten die Dateisysteme selbst erzeugt werden.



Tipp: Erzeugung der Dateisysteme

Installieren Sie die gewählte Linux-Distribution ganz regulär auf einem normalen PC. Wählen Sie dabei wie üblich die Partitionen und lassen Sie das Installationsprogramm die Pakete auf die Festplatte spielen. Wenn Sie noch keine Erfahrung mit UML haben, installieren Sie zunächst die Distribution in einer einzigen Partition (z.B. hda1). Anschließend fahren Sie den Rechner herunter und booten ein alternatives Linux-Betriebssystem oder von einer Linux-Bootdiskette. Nun können Sie die Dateisysteme mit `dd` in Dateien sichern:

```
dd if=/dev/hda1 of=/root_fs.img
```

Bitte löschen Sie die Dateisysteme vor der Installation zum Beispiel mit dem Befehl:

```
dd if=/dev/zero of=/dev/hda1
```

Dies stellt bei einer späteren forensischen Analyse sicher, dass Sie nicht alte Dateileichen wieder ausgraben müssen.

Liegt ein Dateisystem vor, so kann der UML Linux-Kernel gestartet werden. Dieser Linux-Kernel unterstützt eine ganze Reihe von Optionen. Werden keine Optionen angegeben, so sucht er sein Dateisystem unter dem Namen `root_fs` im aktuellen Verzeichnis und startet. Die folgenden Optionen verändern das Startverhalten:

- `--showconfig`. Zeigt die Konfiguration, mit der UML erzeugt wurde
- `mem=RAM`. Definiert die Größe des physikalischen Arbeitsspeichers. Weiterer Speicher wird im SWAP alloziert. Beispiel: `mem=64M`
- `iomem=name, file`. Konfiguriert `file` als IO-Speicherregion mit Namen `<name>`
- `debugtrace`. Sinnvoll für das Debugging
- `tty_log_dir=dir`. Wenn Sie das TTY-Logging aktiviert haben, protokolliert der UML-Kernel jede Sitzung in einer eigenen Datei.
- `debug`. Startet den Kernel unter der Kontrolle des `gdb`
- `--version`. Gibt die Versionsnummer aus
- `root=Root-Dateisystem`. Dies wird vom UML-Kernel genauso gehandhabt wie von jedem anderen Kernel. Beispiel: `root=/dev/ubd5`
- `--help`. Gibt Hilfe aus
- `umid=name`. Weist der UML-Maschine einen eindeutigen Namen zu. Dieser Name wird auch für die PID-Datei und den Socket für die Managementkonsole verwendet.
- `uml_dir=directory`. Verzeichnis mit PID-Datei und `umid`-Dateien
- `initrd=nitrd image`. Initrd-Image für den UML-Kernel
- `con[0-9]*=channel description`. Verbindet eine Konsole mit einem Kanal des Gastgeber-Systems
- `ssl[0-9]*=channel description`. Verbindet eine serielle Schnittstelle mit einem Kanal des Gastgeber-Systems
- `eth[0-9]+= transport, options`. Konfiguriert die Netzwerkkarte mit einer der folgenden Möglichkeiten:
 - `eth[0-9]+=ethertap,<device>,<ethernet address>,<tap ip address>`
`eth0=ethertap,tap0,,192.168.0.1`
 - `eth[0-9]+=tuntap,,<ethernet address>,<ip address>`
`eth0=tuntap,,fe:fd:0:0:0:1,192.168.0.1`
 - `eth[0-9]+=daemon,<ethernet address>,<type>,<control socket>,<datasocket>`
`eth0=daemon,unix,/tmp/uml.ctl,/tmp/uml.data`
 - `eth[0-9]+=slip,<slip ip>`
`eth0=slip,192.168.0.1`
 - `eth[0-9]+=mcast,<ethernet address>,<address>,<port>,<ttl>`
`eth0=mcast,,224.2.3.4:5555,3`
- `mconsole=notify: socket`. Benachrichtigung der Managementkonsole
- `fake_ide`. Erzeuge `ide0`-Einträge für die `ubd`-Geräte.
- `ubd<n>= filename`. Dies definiert die Geräte, auf denen sich Dateisysteme befinden. Es können maximal acht (0-7) Geräte definiert werden. Durch das Anhängen von `r` wird dieses Dateisystem als `read-only` angesehen. Beispiel: `ubd1r=./ext_fs`.

- `fakehd`. Benennt intern die `ubd`-Geräte in `hd`-Geräte um. Hiermit ist schwerer zu erkennen, dass es sich um ein `UserModeLinux` handelt.
- `dsp= dsp device`. Erlaubt die Verwendung des Soundtreibers auf dem Gastgeber-System
- `mixer= mixer device`. Erlaubt die Verwendung des Mixers auf dem Gastgeber-System

Ein Aufruf kann daher folgendermaßen erfolgen:

```
# tuncctl -u user
# ifconfig tap0 <IP-Host-Adresse> up
# chmod 666 /dev/net/tun
[<user>]$ linux ubd0=root_fs ubd1=swap_fs mem=64Meth0=tuntap,,,IP-Host-Adresse
```

Die ersten drei Befehle erzeugen das TAP-Gerät auf dem Host-System. Diese Netzwerkkarte wird vom Host-System für die Kommunikation mit dem Guest-UML-System verwendet. UML bietet auch eine automatische Erzeugung dieses Gerätes mit dem Werkzeug `uml_net` an. Dieses erzeugt aber bei der Angabe der Option `honeynet` einen Fehler. Außerdem besteht damit die Möglichkeit, dass die UML-Session auf den `SetUID`-Befehl `uml_net` zugreifen kann.

Das Swap-Dateisystem kann mit den folgenden Befehlen angelegt werden:

```
# dd if=/dev/zero of=swap_fs count=128 bs=1024k
# mkswap swap_fs
```

24.3 Konfiguration der Netzwerkdienste

Damit das System als Honeypot genutzt werden kann, muss es auch gewisse Netzwerkdienste anbieten. Hier kann die gesamte Anzahl der Netzwerkdienste angeboten werden. Jedoch sollten einzelne Dienste auch konfiguriert werden, so dass der Angreifer die Illusion eines schlecht gepflegten, aber genutzten Systems erhält.

In Frage kommende Dienste sind unter Linux sicherlich:

1. Apache-Webserver
2. WU-FTP-Server
3. BSD LPD
4. UCD-SNMP
5. RPC Statd
6. WU Imapd

Wenn eine ältere Distribution eingesetzt wird, so kann mit Sicherheit davon ausgegangen werden, dass der eine oder andere Dienst eine Sicherheitslücke aufweist, die auch von einem Angreifer ausgenutzt wird, wenn sie entdeckt wird.

Diese Systeme sollten nicht nur installiert werden, sondern auch eine gewisse Konfiguration erfahren. Dazu sollten einzelne Benutzer mit willkürlichen Kennwörtern erzeugt werden. Diese Benutzer sollen scheinbare E-Mails in ihren E-Mail-Eingängen erhalten. Der Webserver sollte ein oder zwei unscheinbare Seiten anbieten und auf dem FTP-Server sollten gewisse Pakete oder Informationen zum Download verfügbar sein. Dies können zum Beispiel scheinbare *mp3*-Dateien sein oder Softwarepakete.

Eine scheinbare *mp3*-Datei kann zum Beispiel mit folgenden Befehlen aus einer echten Datei erzeugt werden:

```
# dd if=real.mp3 of=fault.mp3 bs=1 count=10
# dd if=/dev/urandom bs=1 count=3437328 >> fault.mp3
# file fault.mp3
# fault.mp3: MP3, 32 kBits, 22.05 kHz, Mono
```

24.4 Zugang zum Honeypot und Schutz weiterer Systeme vor dem Honeypot

Wenn ein Honeypot eingesetzt wird, ist es äußerst wichtig, dass weitere Rechner vor den Risiken geschützt werden. Es ist ja der Sinn des Honeypots, dass ein Angreifer in ihn eindringt und einbricht. Der Angreifer wird anschließend wahrscheinlich versuchen, von diesem Rechner aus weitere Rechner anzugreifen. Dies sollte unbedingt verhindert werden, um so andere Rechner nicht einer zusätzlichen Gefahr auszusetzen.

Hierzu sollten die Verbindungen, die vom Honeypot ausgehen, überwacht, kontrolliert und eingeschränkt werden.

Dennoch muss sichergestellt werden, dass der Honeypot von außen erreichbar ist. Die ideale Lösung für dieses Problem ist *netfilter* mit dem Kommando *iptables*. Hintergründe und die genaue Funktionsweise werden in dem bereits erwähnten Buch *Linux Firewalls* (siehe hierzu die Literaturhinweise auf Seite 818) erläutert. Hier soll nur so viel gesagt werden: Netfilter erlaubt den Aufbau eines zustandsorientierten Paketfilters.

Es existieren nun drei denkbare Varianten des Netzwerkaufbaus zum Honeypot:

- **Routing.** Der Honeypot bekommt eine eigene offizielle Adresse in einem eigenen Subnetz. Dies kann auch eine Point-to-Point-Verbindung sein. Das Betriebssystem auf dem physikalischen Rechner übernimmt die Rolle des Routers. Diese Anwendung ist insbesondere in internen Netzen interessant, wo relativ unproblematisch weitere IP-Adressbereiche zur Verfügung stehen.

- **NAT.** Der Honeypot bekommt eine private Adresse. Das Betriebssystem auf dem physikalischen Rechner erhält eine offizielle Adresse und nattet alle Anfragen so, dass sie direkt an den Honeypot geliefert werden.
- **Bridge/ProxyARP.** Der Honeypot erhält eine offizielle IP-Adresse aus demselben Netzwerk. Jedoch besitzt lediglich das physikalische Gastgeber-Betriebssystem eine Verbindung mit diesem Netzwerk. Daher muss es Anfragen für den Honeypot annehmen und weiterleiten. Dies kann mit Bridging bzw. mit ProxyARP erreicht werden. Das Honeynet Project hat neue Skripte auf seiner Webpage veröffentlicht, die diesen Aufbau erlauben.

Hier sollen das Routing und das NAT-Szenario genauer besprochen werden.

24.4.1 Routing des Honeypots

In der Abbildung 24.1 ist eine beispielhafte Umgebung skizziert. Dieser Aufbau kann sowohl mit einem virtuellen als auch mit einem physikalisch existenten Honeypot aufgebaut werden. Die Firma *nohup.info* setzt einen Router ein, der die demilitarisierte Zone vom Internet trennt. Dieser Router wird auch für die sich dahinter befindenden Systeme verwendet. Mit einer Netzwerkkarte ist der Honeypot verbunden. Um IP-Adressen zu sparen, wurde hier eine Point-to-Point-Verbindung gewählt. Dies erlaubt die Erzeugung eines Subnetzes mit nur einer einzigen IP-Adresse.

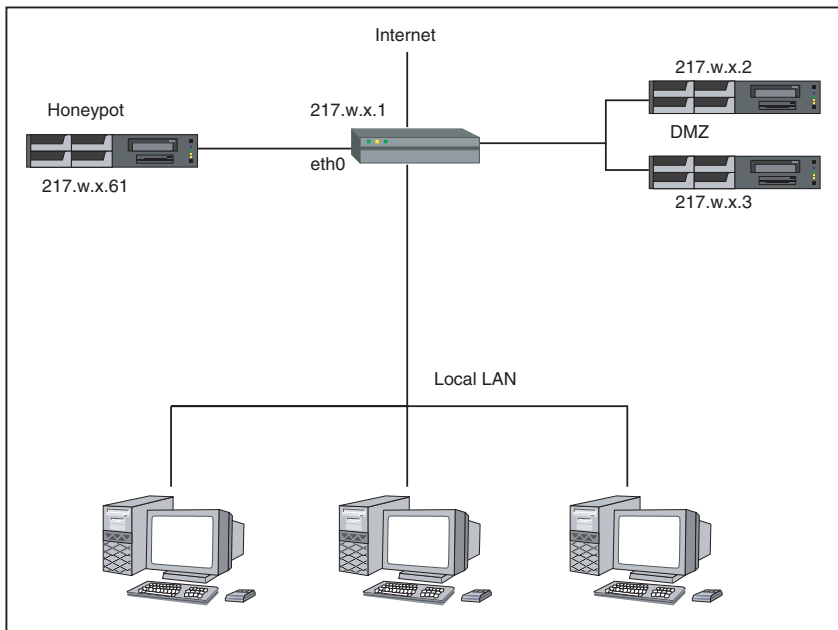


Abbildung 24.1 Implementierung des Honeypots mit Routing

Um diese Netzwerkkonfiguration zu implementieren, ist es erforderlich, die Netzwerkkarten als Point-to-Point-Netzwerke zu konfigurieren. Dies erfolgt unter Linux auf dem Honeypot mit den Befehlen:

```
/sbin/ifconfig eth0 217.w.x.61 netmask 255.255.255.255 pointopoint 217.z.y.1
/sbin/route add default gw 217.z.y.1
```

Auf dem Router ist der entsprechende Befehl mit ausgetauschten IP-Adressen zu verwenden.

```
/sbin/ifconfig eth0 217.z.y.1 netmask 255.255.255.255 pointopoint 217.w.x.61
```

Nun sind einige Paketfilterregeln erforderlich, um sicherzustellen, dass Verbindungen zum Honeypot aufgebaut werden können, aber der Honeypot keine Verbindungen selbst aufbauen kann. Dies erfolgt am einfachsten mit wenigen *iptables*-Aufrufen.

```
#!/bin/bash
#
# Erlaube Zugriff auf den Honeypot

IPCMD=/sbin/iptables
MPCMD=/sbin/modprobe

honeypot=217.w.x.61/32

rest=any/0

hpot_card=eth0

# Lade Modul
$MPCMD ip_tables

# Lösche und leere die Ketten
$IPCMD -F
$IPCMD -X

# Erzeuge eine Kette für den Honeypot
$IPCMD -N hpot_chain

# Sammle alle Pakete, die den Honeypot betreffen in einer Kette
$IPCMD -A FORWARD -s $honeypot -j hpot_chain
$IPCMD -A FORWARD -d $honeypot -j hpot_chain

# Erlaubte Verbindungen dürfen passieren
$IPCMD -A hpot_chain -m state --state RELATED,ESTABLISHED -j ACCEPT

# Erlaube den Verbindungsaufbau zum Honeypot und protokolliere ihn
$IPCMD -A hpot_chain -d $honeypot -m state --state NEW -j LOG \
```

```

--log-prefix "Neue Verbindung: "
$IPCMD -A hpot_chain -d $honeypot -m state --state NEW -j ACCEPT

# Verweigere den Verbindungsaufbau von dem Honeypot
$IPCMD -A hpot_chain -s $honeypot -m state --state NEW -j LOG \
--log-prefix "HONEYPOT Verb.: "
$IPCMD -A hpot_chain -s $honeypot -m state --state NEW -j DROP

# Protokolliere und verweigere alles weitere
$IPCMD -A hpot_chain -j LOG "hpot_chain REST: "
$IPCMD -A hpot_chain -j DROP

sysctl -w net.ipv4.ip_forward=1

```

24.4.2 NAT (Network Address Translation) des Honeypots

In der Abbildung 24.2 ist eine beispielhafte Umgebung skizziert. Dieser Aufbau kann sowohl mit einem virtuellen als auch mit einem physikalisch existenten Honeypot aufgebaut werden. Die Firma *nohup.info* setzt einen Rechner ein, der die offizielle IP-Adresse trägt, die dem Honeypot zugewiesen wurde. Dieser Rechner verfügt über zwei Netzwerkkarten und ist mit einer Netzwerkkarte mit dem Honeypot verbunden. Der Honeypot erhält eine private IP-Adresse in einem eigenen Adressraum.

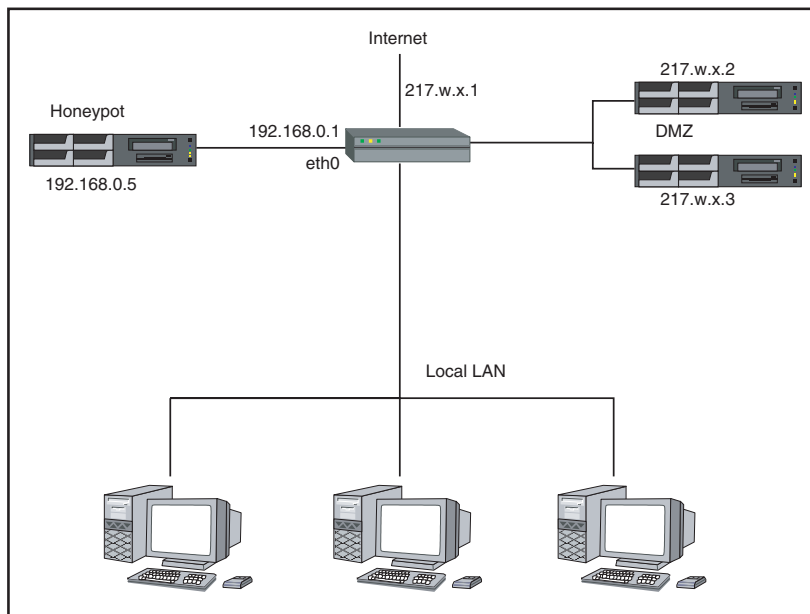


Abbildung 24.2 Implementierung des Honeypots mit NAT

Um diese Netzwerkkonfiguration zu implementieren, können die Netzwerkkarten wie üblich konfiguriert werden. Dies erfolgt unter Linux auf dem Honeypot mit den Befehlen:

```
/sbin/ifconfig eth0 192.168.0.5 netmask 255.255.255.0 broadcast 192.168.0.255
/sbin/route add default gw 192.168.0.1
```

Auf dem Router ist der entsprechende Befehl zu verwenden.

```
/sbin/ifconfig eth0 192.168.0.1 netmask 255.255.255.0 broadcast 192.168.0.255
```

Nun sind einige Paketfilterregeln erforderlich, um sicherzustellen, dass Verbindungen zum Honeypot aufgebaut werden können, aber der Honeypot keine Verbindungen selbst aufbauen kann. Eine derartige Umgebung erfordert zusätzlich Destination NAT. Die Regelerzeugung erfolgt am einfachsten mit wenigen *iptables*-Aufrufen.

```
#!/bin/bash
#
# Erlaube Zugriff auf den Honeypot

IPCMD=/sbin/iptables
MPCMD=/sbin/modprobe

# IP Adresse des Routers
guard=217.w.x.1/24

# IP Adresse des honeypot
honeypot=192.168.0.5
rest=any/0

hpot_card=eth0

# Lade Modul
$MPCMD ip_tables

# Lösche und leere die Ketten
$IPCMD -F
$IPCMD -t nat -F
$IPCMD -X

# Erzeuge eine Kette für den Honeypot
$IPCMD -N hpot_chain

# Erzeuge die Regeln für D-NAT
$IPCMD -A PREROUTING -s $rest -d $guard -j DNAT --to $honeypot

# Sammele alle Pakete, die den Honeypot betreffen in einer Kette
$IPCMD -A FORWARD -s $honeypot -j hpot_chain
$IPCMD -A FORWARD -d $honeypot -j hpot_chain
```

```

# Erlaubte Verbindungen dürfen passieren
$IPCMD -A hpot_chain -m state --state RELATED,ESTABLISHED -j ACCEPT

# Erlaube den Verbindungsaufbau zum Honeypot und protokolliere ihn
$IPCMD -A hpot_chain -d $honeypot -m state --state NEW -j LOG \
    --log-prefix "Neue Verbindung: "
$IPCMD -A hpot_chain -d $honeypot -m state --state NEW -j ACCEPT

# Verweigere den Verbindungsaufbau von dem Honeypot
$IPCMD -A hpot_chain -s $honeypot -m state --state NEW -j LOG \
    --log-prefix "HONEYPOT Verb.: "
$IPCMD -A hpot_chain -s $honeypot -m state --state NEW -j DROP

# Protokolliere und verweigere alles weitere
$IPCMD -A hpot_chain -j LOG "hpot_chain REST: "
$IPCMD -A hpot_chain -j DROP

sysctl -w net.ipv4.ip_forward=1

```

24.4.3 Weiter gehende Firewall-Konfiguration

Die bisher vorgestellte Firewall-Konfiguration erlaubt lediglich Verbindungen von außen auf den Honeypot. Sobald der Honeypot selbst eine Verbindung nach außen aufbaut, wird diese von der Firewall unterbunden. Damit besteht für den Angreifer keine Möglichkeit, weitere Werkzeuge per FTP herunterzuladen, weitere Rechner anzugreifen oder von diesem Rechner aus die umliegenden Rechner zu untersuchen und somit ihn als ein weiteres Sprungbrett zu neuen Angriffen zu nutzen.

Wenn der Honeypot nur als Mittel zur Detektion eingesetzt werden soll, so genügt diese Firewall-Konfiguration bereits. Sobald der Einbruch auf dem Honeypot erfolgte, kann dieser festgestellt werden. Eine weitere Schädigung dritter Personen oder weiterer Rechner ist ausgeschlossen.

Es ist jedoch unter Umständen nicht sinnvoll, den gesamten vom Honeypot ausgehenden Verkehr zu unterbinden. Dies kann die Attraktivität des Honeypots für den Angreifer stark herabsetzen. Unter Umständen führt es dazu, dass sich der Angreifer nicht für den Honeypot interessiert, keine zusätzlichen Werkzeuge installiert und somit diese Aktionen nicht studiert werden können.

Wenn das Studium des Angreifers das Hauptziel des Honeypots darstellt oder der Honeypot als forensisches Hilfswerkzeug bei einem Angriff eingesetzt werden soll, so ist es erforderlich, dass der Angreifer auf dem Rechner verbleibt und dort weitere Aktionen ausführt, die überwacht und analysiert werden können. Dies wird jedoch nur der Fall sein, wenn er auch erfolgreich Verbindungen nach außen aufbauen kann.

Damit aber dennoch keine große Gefahr vom Honeybot für andere Netzwerke und Rechner ausgehen kann, muss der Verkehr überwacht, kontrolliert und möglicherweise eingeschränkt werden. Hierzu gibt es verschiedene Möglichkeiten. Eine Auswahl soll im Folgenden vorgestellt werden.

Die Limitierung der ausgehenden Verbindungen auf eine bestimmte Anzahl pro Zeiteinheit ist eine sinnvolle Einschränkung. Dies erlaubt nicht den Scan oder den automatischen Angriff auf weitere Rechner. Dies kann sehr einfach mit Linux-Netfilter erfolgen.



Tipp:

Die Verwendung von *Snort-Inline* (<http://snort-inline.sourceforge.net>) auf dem Host-System als Firewall kann direkt Angriffe unterbinden, da Snort-Inline die gefährlichen Pakete erkennt, verwirft und nicht durchlässt.

Der Linux-Kernel 2.4 bietet mit Netfilter die Möglichkeit, die Anzahl bestimmter Pakete pro Zeiteinheit zu limitieren. Dies erfolgt mit der Testerweiterung `limit`. Diese erlaubt die Angabe einer Paketrate pro Zeiteinheit und eines Schwellenwertes, ab dem die Paketrate erzwungen werden soll.

Zur Anwendung dieses Befehles sollten zunächst die maximalen Paketraten für bestimmte Protokolle bestimmt werden. Folgende Raten sind wahrscheinlich sinnvoll für ausgehende Verbindungen eines Honeybots:

- **TCP.** Acht Verbindungen/Stunde
- **UDP.** Zwölf Verbindungen/Stunde (Namensauflösung)
- **ICMP.** 20 Verbindungen/Stunde (Ping)

Um derartige Raten zu erzwingen, können mit dem Befehl `iptables` die folgenden Regeln implementiert werden.

```
# iptables -A FORWARD -s $honeypot -m state --state NEW -p tcp \
    -m limit --limit 8/h -j ACCEPT
# iptables -A FORWARD -s $honeypot -m state --state NEW -p udp \
    -m limit --limit 12/h -j ACCEPT
# iptables -A FORWARD -s $honeypot -m state --state NEW -p icmp \
    -m limit --limit 20/h -j ACCEPT
# iptables -A FORWARD -s $honeynet -j DROP
```

Die Regeln sollten außerdem ein Spoofing der Source IP-Adresse des Honeybots nicht erlauben. Als Spoofing bezeichnet man eine Fälschung der Absenderadresse durch den Angreifer. Dies wird in den oben aufgeführten Regeln erreicht, indem nur Verbindungen von der IP-Adresse des Honeybots nach außen erlaubt werden. So

werden weitere Pakete, die möglicherweise vom Angreifer mit gefälschten Absender-IP-Adressen versendet werden, von den Regeln verworfen. Dies bezeichnet man auch als einen Egress-Filter.

Das Honeynet Project hat im Rahmen des *Know your Enemy*-Projektes ein Skript veröffentlicht, das diese Regeln implementieren kann. Eine Betrachtung und möglicherweise ein Einsatz dieses Skripts sollte überlegt werden. Dieses Skript ist frei im Rahmen des Honeynet Projects erhältlich und kann an die eigenen Zwecke angepasst werden. Es ist auf der Homepage des Honeynet Projects und auf verschiedenen Spiegeln verfügbar. Das Honeynet Project hat unter <http://project.honeynet.org/papers/honeynet/rc.firewall> das Skript veröffentlicht.



Achtung:

Das Honeynet Project pflegt auf seiner Homepage weitere Werkzeuge, Firewall-Skripts und Kernel-Patches, um einen Honeypot in einem bridged mode zu betreiben. Es handelt sich um Sebek, welches ein Tastaturlogging durchführt und ein Bridge-Firewallskript für die Einbindung von Snort-Inline. Diese Tools finden Sie unter <http://www.honeynet.org/papers/honeynet/tools/index.html>.

24.5 Überwachung des Honeypots von außen

Damit der Honeypot überhaupt als solcher genutzt werden kann, ist es wichtig, dass er überwacht wird, um die Aktionen des Angreifers zu protokollieren und einen Angriff zunächst überhaupt zu erkennen. Hierzu können unterschiedliche Methoden eingesetzt werden. Eine Besprechung dieser Methoden erfolgt nur oberflächlich, da die entsprechenden Themen entweder bereits ausführlich in anderen Teilen des Buches besprochen wurden oder ihre Konfiguration zu sehr von der speziellen Umgebung abhängt.

- **Netzwerkbasierter IDS.** Dieses NIDS überwacht alle Pakete, die an den Honeypot gesendet und von ihm versendet werden. Hier sollten nun keine weiteren Angaben erforderlich sein. Es sollte nur darauf hingewiesen werden, dass zusätzlich mit *tcpdump* oder *snort* der gesamte Netzwerkverkehr protokolliert werden muss. Dies stellt sicher, dass nach einem Angriff alle relevanten Pakete auch zur Analyse gespeichert wurden. Wenn es sich noch um einen unbekanntes Angriff handelt, hat das NIDS möglicherweise diesen auch nicht erkannt und die entsprechenden Pakete nicht protokolliert. Eine Analyse des Angriffes ist dann kaum möglich. Da der Honeypot unter dauernder Beobachtung stehen sollte, ist es meist ausreichend, diese Daten für wenige Tage vorzuhalten.
- **Hostbasierter IDS.** Dieses HIDS überwacht die Dateien des Honeypots und meldet Veränderungen. Wenn der Honeypot als virtuelles Betriebssystem unter Ver-

wendung entweder von VMware oder Usermode-Linux aufgebaut wurde, so besteht die Möglichkeit, vom Gastgeber-(Host-)System aus stündlich die Dateisysteme des Honeypots read-only zu mounten und mit Tripwire zu analysieren. Des Weiteren können stündlich die Protokolldateien kopiert werden. Damit sämtliche Informationen so gesammelt werden können, ist es erforderlich, dass sowohl das Betriebssystem als auch die Virtualisierungssoftware (VMware) keinen Festplattencache für Schreiboperationen verwendet. In Linux kann dies durch die Mountoption *sync* erreicht werden. VMware bietet die Möglichkeit, dies im Konfigurationseditor zu deaktivieren.

- **Modifizierte Shell.** Eine modifizierte Shell ist in der Lage, sämtliche eingegebenen Befehle auf einen anderen Rechner zu übertragen und so ein Logbuch der Aktionen des Angreifers zu erzeugen. Dies ist selbst dann erfolgreich, wenn der Angreifer die Geschichte der Shell (z.B. `~/.bash_history`) löscht oder mit `/dev/null` verlinkt. Ein derartiger Ansatz wird auch vom Honeynet Project verfolgt und mit Links zu derartigen Patches im Whitepaper *Know your Enemy: Honeynets* erklärt (<http://project.honeynet.org./papers/honeynet/index.html>).
- **Modifiziertes Skript-Kommando.** Ryan Barnett beschreibt in seinem *GCFa Practical* (Eine Prüfung des SANS Institutes zum GIAC Certified Forensic Analyst) die Modifikation des Kommandos `script` und den Einsatz auf einem Honeypot. Dieses Kommando ist nicht nur in der Lage, die vom Angreifer eingegebenen Befehle zu speichern und zu übermitteln, sondern zusätzlich noch die Ausgaben dieser Befehle, wie der Angreifer sie sieht. Diese Informationen helfen insbesondere bei der Rekonstruktion der Ereignisse, denn die analysierende Person kann so häufig erkennen, warum der Angreifer den nächsten Befehl ausgeführt hat. Auch häufige Tippfehler werden protokolliert und erlauben möglicherweise die Identifikation des Angreifers, wenn sich verschiedene Personen auf dem Rechner befinden. Das GIAC Practical mit weiteren Informationen ist unter http://mywebpages.comcast.net/rbarnett45/ryan_barnett_gcfa/ryan_barnett_gcfa_practical.html verfügbar.
- **Kernel-Modul zur Protokollierung der Eingaben.** Das Honeynet Project hat auch den Befehl `ttywatcher` zu einem Kernel-Modul weiterentwickelt, welches die Aufgabe der Protokollierung der Eingaben des Angreifers übernehmen kann. Dieses Modul sendet die Informationen dann über eine TCP-Verbindung. Dieses Kernel-Modul ist jedoch nur für Solaris verfügbar. Ein ähnliches Modul ist `linspy`. Dieses wurde in <http://www.phrack.com/phrack/50/P50-05> besprochen.
- **e2tools.** Die `e2tools` von Keith Sheffield bieten die Möglichkeit, ohne Wissen des Angreifers vorübergehend auf die Dateisysteme eines virtuellen Honeypots zuzugreifen. So können direkt nach dem Angriff bereits die vom Angreifer veränderten Dateien betrachtet werden. Seit der Version 0.0.13 unterstützen die `e2tools` auch den Befehl `e2tail -f`. Dieser Befehl öffnet das Dateisystem read-only, liest die Datei und gibt die letzten Zeilen aus, schließt das Dateisystem, wartet eine Sekunde (einstellbar), öffnet das Dateisystem erneut und liest nur die angehängten Zeilen. Damit ist eine Protokollüberwachung online möglich, ohne eine Netzwerkverbindung aufbauen zu müssen, die von einem Angreifer leicht erkannt

und deaktiviert werden kann. Dies funktioniert jedoch nur bei *ext2fs*-Dateisystemen.

- **Man-in-the-Middle: SSL/SSH.** Ein besonderes Problem bei der Überwachung des Honeybots stellen verschlüsselte Verbindungskanäle des Angreifers dar. Ein NIDS ist nicht in der Lage, bei derartigen Kanälen die vom Angreifer versendeten Informationen lesbar zu protokollieren. Es wurden bereits im Vorfeld Werkzeuge beschrieben, die die Eingaben des Angreifers direkt auf dem Honeybot protokollieren können. Setzt der Angreifer jedoch Werkzeuge ein, die automatisch Daten über verschlüsselte Kanäle versenden, so ist häufig eine Protokollierung der Daten nicht möglich.

Eine in ihrer Konfiguration sehr aufwändige Lösung ist die Installation eines Man-in-the-Middle-Proxys auf dem Überwachungsrechner. Dieser fängt sämtliche Anfragen auf klassischen SSH- und SSL-Ports ab, handelt mit dem Client die Verschlüsselung aus und baut dann selbstständig eine neue verschlüsselte Verbindung zum Server auf. Die klassischen SSH- und SSL-Ports sind:

- 22: ssh
- 443: http/ssl
- 465: smtp/ssl
- 563: nntp/ssl
- 636: ldap/ssl
- 992: telnet/ssl
- 993: imap/ssl
- 994: irc/ssl
- 995: pop3/ssl

Es existieren verschiedene Proxies, die für diese Zwecke verwendet werden können. Die wahrscheinlich ersten verfügbaren Proxies wurden von Dug Song im Rahmen seiner Sniffer-Suite *dsniff* veröffentlicht und heißen: *sshmitm* und *webmitm*. Dug Song hat aufgrund des *Digital Millenium Copyright Act* (DMCA) den Zugriff auf seine Homepage gesperrt. Die Werkzeuge sind aber an verschiedenen Stellen immer noch verfügbar. Des Weiteren wurden verbesserte Proxies inzwischen auch von anderen Autoren zur Verfügung gestellt.

Dieses Kapitel hat kurz die verschiedenen Methoden, die zur Überwachung des Honeybots eingesetzt werden können, skizziert. Die Überwachung ist unumgänglich und sollte nicht auf die leichte Schulter genommen werden. Es besteht ansonsten die Gefahr, dass ein Einbruch nicht erkannt wird und der Honeybot vom Angreifer verwendet wird, um dritte Personen zu schädigen. Dies muss mit allen Mitteln verhindert werden.

24.6 Fazit

Nach der Besprechung der verschiedenen Optionen zur Installation und Konfiguration eines Honeyports sollte der hohe administrative Aufwand deutlich geworden sein. Dies ist ein sehr wichtiger Punkt und darf nie aus den Augen verloren werden. Ein Honeyport stellt in keiner Weise ein Spielzeug dar, welches gekauft wird, mit dem kurz gespielt wird und das, sobald Langeweile aufkommt, in die Ecke gelegt werden kann. Es ist vielmehr mit einem Spielzeug aus einem Horrorfilm vergleichbar, welches nach kurzer Zeit der Missachtung zum Leben erwacht und den Besitzer übel verfolgen kann.

In vielen Fällen kann die Energie und Zeit, die für Installation, Konfiguration, Überwachung und Analyse des Honeyports aufgewendet wird, sinnvoller in die allgemeine Sicherung des Netzwerkes gesteckt werden. Dies mag nicht so reizvoll und spannend erscheinen, ist jedoch häufig sinnvoller.

Die regelmäßige Analyse der Firewall-Protokolle, die Anpassung der Firewallregeln, die Aktualisierung der Virens Scanner, die Überwachung und Pflege der IDS-Systeme inklusive der Reduktion der fehlerhaften Meldungen und der Entwicklung neuer Regeln für neue Angriffe erfordern bereits einen äußerst hohen Administrationsaufwand.

Welchen Honeyport soll ich nun wählen und kann ich ihn mir leisten?

Wird dennoch der Einsatz eines Honeyports gewünscht, so sollte zunächst der Einsatz eines *low involvement*-Honeyports, der lediglich gewisse Netzwerkdienste simuliert, ins Auge gefasst werden. Dieser Honeyport erzeugt nur ein geringes Risiko auf der Seite des Anwenders und benötigt keine umfangreiche Überwachung. Er ist auch in der Lage, neuartige Angriffstrends zu erkennen und eine Aktualisierung der Regeln der Firewall und des IDS anzuregen.

Der Einsatz eines Honeyports hat meist verschiedene Ziele.

Der Honeyport soll mindestens der Detektion von Angriffen dienen, die das IDS nicht alleine erkennen kann. Dies kann daran liegen, dass das IDS falsch konfiguriert wurde oder ein signaturbasiertes IDS diesen Angriff noch nicht kannte.

Zusätzlich wird meist gewünscht, durch eine Analyse des Angriffes die Sicherheitslücken zu erkennen und die Handlungen des Angreifers zu verstehen. Gebündelt mit einer anschließenden forensischen Analyse dient dies der Weiterbildung der Personen, die sich mit dem Honeyport beschäftigen. Weiterbildung ist im Zusammenhang mit Firewalls, IDS und Honeyports essenziell, denn Sicherheit ist kein Produkt, welches man kaufen kann, sondern ein Prozess. Um Sicherheit zu erreichen, ist eine dauernde Anpassung und Verbesserung der vorhandenen Strukturen erforderlich, die nur möglich ist, wenn das entsprechende Wissen vorhanden ist.

Im Rahmen des analysierten Angriffes auf den Honeyport erhält der Analysator sehr viele Informationen. Diese sollen meist in irgendeiner Form auf den Produktions-

rechnern umgesetzt werden, um derartige Angriffe dort zu unterbinden. Dies kann eine Aktualisierung der betroffenen Software bedeuten. Des Weiteren ist eine Anpassung des IDS möglich, so dass der Angriff in Zukunft erkannt werden kann. Ist ein Schließen der Sicherheitslücke nicht möglich, da keine Patches des Herstellers zur Verfügung stehen, so sollte zumindest die Firewall angepasst werden, um den Angriff zu verhindern. In gewissen Fällen mag es auch sinnvoll sein, die im Angriff vom Angreifer verwendeten IP-Adressen auf der Firewall für das Produktionsnetz komplett zu sperren.

Schließlich ist häufig auch eine Verfolgung des Angriffes, eine Warnung anderer beteiligter Personen und eine Meldung an die zuständigen Gremien im Sinn des Einsatzes eines Honeypots. Diese Tätigkeiten wurden bereits in vorangegangenen Kapiteln angesprochen.

