

Vorwort

IPCop ist eine speziell für diese Aufgabe als Router angepasste Linux-Distribution, welche sich durch Ihre einfache Handhabung und letztendlich auch durch Ihre Ausgereiftheit geradezu empfiehlt innerhalb einer eigenen virtuellen Maschine die Aufgabe des Routers im Netzwerk zu übernehmen.

Das macht meiner Ansicht nach vor allem Sinn, da IPCop wesentlich mehr Funktionen bietet als gängige Router im SOHO Bereich. Dabei handelt es sich um Funktionen wie VPNs, detaillierter konfigurierte DMZ oder Logdaten.

Die eigentliche Schwierigkeit IPCop in einem Xen Gastsystem zu installieren liegt darin den Xen-Kernel den Bedürfnissen IPCops anzupassen.

Da die Konfiguration IPCops auf verschiedenen Netzwerkinterfaces basiert, sollte man der Eierlegenden Wollmilchsau - falls nicht schon geschehen - weitere Netzwerkkarten spendieren und diese exklusiv dem IPCop-Gastsystem zuweisen.

Beispielsweise habe ich hier 2 physische Netzwerkkarten, welche zu einem als „ROTEN“ Interface für den Anschluss des Modems und zum anderen als „BLAUES“ Interface zum Anschluss eines AccessPoints dienen. Ein virtuelles Interface dient dem Anschluss an das lokale Netzwerk. Somit sind die drei verschiedenen Netzwerksegmente physisch voneinander getrennt.

1. Vorbereitungen

1.1. Download der nötigen Software

Da die c't in Ihrem c't Debian-Server IPCop in einer virtuellen Maschine einer User-Mode-Linux (UML) Umgebung betreibt, bietet es sich an das Image dieser virtuellen Maschine für die Verwendung unter Xen zu verwenden. Somit spart man sich das Erstellen eines eigenen Images.

Zu finden war das Image auf der Heft-CD zu Ausgabe 04/2005.

Wer diese CD nicht besitzt, der kann alternativ auch das ISO-Image des c't Debian-Servers vom ftp-Server des Heise-Verlages herunterladen. Das IPCop-Image muss dann noch entsprechend extrahiert werden.

<http://www.heise.de/ct/ftp/projekte/srv/download.shtml>

Sowohl auf der Heft-CD als auch in dem zum Download angebotenen ISO-Image findet man die .deb-Datei mit den IPCop-Images unter /pool/main/i/ipcop/.

Weiter benötigt werden die aktuellen Sourcen von iptables sowie der Patch-O-Matic Distribution des netfilter-Projektes, welche die Patches für den Linux-Kernel enthält:

<http://www.iptables.org/files/iptables-1.3.1.tar.bz2>

<ftp://ftp.netfilter.org/pub/patch-o-matic-ng/snapshot/>

Ausserdem ist noch die aktuelle 1er-Version von OpenSwan nötig, welches die IPSec-Patches für den Linux-Kernel mitbringt:

<http://www.openswan.org/download/openswan-1.0.9.tar.gz>

1.3. Links zu Dokumentationen

Informationen zur Handhabung von IPCop sowie laufend aktuelle Sicherheitsupdates findet man unter <http://www.ipcop.org>.

2. Vorbereiten des Xen Gast-Kernels

2.1. Kernel-Tree vorbereiten

Für die Arbeit an dem speziellen Kernel für das IPCop-System empfiehlt es sich, eine Kopie des Xen Gast-Kernel-Trees anzulegen, in welcher man die Patches einspielt.

```
# cd /usr/src/xen
# cp -R linux-2.4.29-xenU linux-2.4.29-ipcop
```

Damit der Kernel dann auch den Namenszusatz -ipcop trägt, muss in der Datei /usr/src/xen/linux-2.4.29-ipcop/Makefile in Zeile 4 dieser auch so angegeben werden:

```
EXTRAVERSION = -ipcop
```

2.2. IPsec Patches einspielen

Die Patches für IPsec bringt die OpenSwan-Distribution in der 1er-Version mit. Hierzu muss diese natürlich zuerst entpackt werden:

```
# tar xpvfz openswan-1.0.9.tar.gz
```

Weiterhin ist ein symbolischer Link auf das Verzeichnis mit den zu patchenden Kernel-Quellen anzulegen, da OpenSwan diesen verlangt.

```
# cd /usr/src
# ln -sf xen/linux-2.4.29-ipcop linux
```

Anschliessend kann man OpenSwan den Kernel patchen lassen.

```
# cd openswan-1.0.9
# make menugo
```

OpenSwan startet nun auch automatisch die Kernel „menuconfig“, welche man jedoch gleich verlassen kann ohne die Änderungen an der Konfiguration zu übernehmen. Anschliessend beginnt OpenSwan trotzdem mit dem Kompilieren des Kernels, was man ebenfalls sofort mit Strg+C abbrechen kann.

2.3. netfilter-Patches einspielen

Das Einspielen der netfilter-Patches gestaltet sich ähnlich einfach. Es müssen hierzu zuerst sowohl die Sourcen von iptables sowie die Patch-O-Matic Distribution entpackt werden:

```
# tar xpvfj iptables-1.3.1.tar.bz2
# tar xpvfj patch-o-matic-ng-20050414.tar.bz2
```

Nun wechselt man in das Verzeichnis von Patch-O-Matic und startet die Auswahl der zu installierenden Patches.

```
# cd patch-o-matic-ng-20050414
# KERNEL_DIR=/usr/src/xen/linux-2.4.29-ipcop
IPTABLES_DIR=/usr/src/iptables-1.3.1 ./runme extra
```

Hier müssen dann folgende Patches eingespielt werden, falls diese noch nicht im Kernel-Tree enthalten sind:

```
classify, connmark, same, addrtype, comment, ownercmd, raw, realm,
hoplimit, ipv4optsstrip, netlink, netmap, reject, TTL, conlimit,
fuzzy, iprange, ipv4options, nth, osf, psd, quota, random, set,
time, u32, router account, condition, cuseeme-nat, h323-contrack-
net, mms-contrack-nat, pptp-contrack-nat, quake3-contrack-nat,
rtsp-contrack, string
```

Eventuell kann es sein, dass man bei dem einen oder anderen Patch den „force“ Modus verwenden muss um ihn zu installieren. Dies ist der Fall, wenn das automatische Patchen nicht an die hiesigen Bedingungen anknüpfen sollte; man wird an dieser Stelle dann erneut gefragt was zu tun ist.

2.4. Kernel für IPCop konfigurieren und kompilieren

Eine Beispielkonfiguration könnt Ihr Euch von [hier](#) von mir herunterladen. Diese ist wie sie ist lauffähig, jedoch muss ggf. der Treiber für die verwendeten physischen Netzwerkkarten angepasst werden. In dieser Kernel-Config wird der e100 Treiber verwendet.

Das Kompilieren des Kernels nimmt dann wie gewohnt seinen Lauf:

```
# cd /usr/src/xen/linux-2.4.29-ipcop
# make ARCH=xen clean
# make ARCH=xen dep
# make ARCH=xen bzImage
# make ARCH=xen modules
```

2.5. Kernel und Module installieren

Nachdem der Kernel sowie die Module kompiliert worden sind, müssen diese noch entsprechend installiert werden.

```
# cd /usr/src/xen/linux-2.4.29-ipcop
# make ARCH=xen modules_install

# cp System.map /boot/System.map-2.4.29-ipcop
# cp .config /boot/config-2.4.29-ipcop
# cd arch/xen/boot
# cp bzImage /boot/vmlinuz-2.4.29-ipcop
```

3. Xen konfigurieren

Nachdem nun die Arbeiten am Gast-Kernel abgeschlossen sind, muss das Hostsystem noch entsprechend konfiguriert werden.

3.1. Ausblenden der Netzwerkdevices im Hostsystem

Da wie oben beschrieben physische Netzwerkdevices dem IPCop-Image zugewiesen werden sollten, müssen diese - wie im Xen-Tutorial beschrieben - vor dem Hostsystem und den anderen Gastsystemen ausgeblendet werden.

Die PCI-Adressen der entsprechenden Netzwerkkarten erfährt man mit dem Befehl `lspci` auf der Konsole des Hostsystems. Diese werden benötigt um die auszublendenden Devices in der GRUB-Konfiguration unter `/boot/grub/menu.lst` zu konfigurieren:

```
default=0
timeout=0
#
title Xen 2.0 / XenLinux 2.6.11
    root (hd0,1)
    kernel /boot/xen.gz dom0_mem=131072 \
        physdev_dom0_hide=(00:0a.0)(00:0b.0)
    module /boot/vmlinuz-2.6.11-xen0 root=/dev/hda2 ro \
        console=tty0 max_loop=16
```

Anschliessend muss diese Konfiguration übernommen werden und das Hostsystem neu gestartet werden.

```
# grub-install /dev/hda
```

```
# reboot
```

Die Ausgeblendeten Devices sollten nach dem Neustart nicht mehr von Hostsystems aus zu sehen sein.

3.2. Konfiguration des Gastsystems

Die Konfiguration des Gastsystems findet wie bei Xen üblich im Verzeichnis /etc/xen statt. Hier wird eine neue Datei ipcop angelegt, welche die Konfiguration für das Gastsystem enthält:

```
# -*- mode: python; -*-

# Kernel image file.
kernel = "/boot/vmlinuz-2.4.29-xenU"

# Initial memory allocation (in megabytes) for the new domain.
memory = 32

# A name for your domain. All domains must have different names.
name = "ipcop"

disk = [ 'file:/home/xen/ipcop/ipcop.img,sda1,w', \
         'file:/home/xen/ipcop/ipcoplog.img,sda2,w' ]

nics = 1

# Set root device.
root = "/dev/sda1 ro"

extra = "3"

pci = ["00,0a,0", "00,0b,0"]

#restart = 'onreboot'
```

4. IPCop-Image anpassen

Da das verwendete Image ja eigentlich für den Betrieb unter User-Mode-Linux gedacht war, müssen nun noch ein paar Änderungen an dessen Konfiguration vorgenommen werden.

Im Rahmen dieses Tutorials wird das IPCop-Image unter /home/xen/ipcop/ liegen.

4.1. Image extrahieren

Falls man das IPCop-Image aus der ISO-Datei verwenden möchte, welche man direkt von heises' ftp-Server beziehen kann, so muss aus dieser zuerst das Debian-Paket mit den IPCop-Images extrahiert werden. Hierzu muss diese ISO-Datei gemountet werden. Anschliessend kann das Debian-Paket herauskopiert und die ISO-Datei wieder ausgehängt werden.

```
# mount ctsrv103cd.iso /mnt/loop -o loop
# cp /mnt/loop/pool/main/i/ipcop/ipcop_1.4.2-1_i386.deb \
    /home/xen/ipcop
# umount /mnt/loop
```

Nun muss noch das Debian-Paket ausgepackt werden um die beiden IPCop-Images für das root- und var- Dateisystem zu erhalten:

```
# ar -x ipcop_1.4.2-1_i386.deb data.tar.gz
```

```
# tar xpvfz data.tar.gz ./var/lib/uml/ipcop/umlipcop.tar.bz2
# mv var/lib/uml/ipcop/umlipcop.tar.bz2 ./
# tar xpvfj umlipcop.tar.bz2
```

Nun liegen die beiden IPCop-Imagedateien ipcop.img und ipcoplog.img vor, beide mit einer Größe von je etwa 512MB.

Um nun im folgenden die Änderungen am Image vornehmen zu können, muss das Image mit dem root-Dateisystem (ipcop.img) gemountet werden:

```
# mount ipcop.img /mnt/loop -o loop
```

4.2. Kernel und Module in das Image kopieren

Da IPCop regen Gebrauch von Kernel-Modulen macht ist es nötig diese in das Image zu kopieren:

```
# cp /boot/*-2.4.29-ipcop /mnt/loop/boot/
# cp -R /lib/modules/2.4.29-ipcop /mnt/loop/lib/modules/
# cd /mnt/loop/boot
# ln -sf System.map-2.4.29-ipcop System.map
# ln -sf config-2.4.29-ipcop config
# ln -sf vmlinuz-2.4.29-ipcop vmlinuz
```

4.3. /etc/fstab anpassen

Damit IPCop auch seine „Partitionen“ findet, muss die enthaltene /etc/fstab noch angepasst werden, da Xen ja in der Regel mit virtuellen SCSI-Devices arbeitet:

/dev/sda2	/var/log	ext3	nodev,nosuid,noatime	1 2
/dev/sda1	/	ext3	noatime	1 1
none	/proc	proc	defaults	0 0
none	/dev/pts	devpts	gid=5,mode=620	0 0

4.4. root Passwort setzen

Zum Abschluss sollte man noch ein eigenes root Passwort setzen:

```
# cd /mnt/loop
# chroot ./
# passwd root
...
# exit
```

Somit sind die Vorbereitungen am Image abgeschlossen und es kann wieder ausgehängt werden.

```
# umount /mnt/loop
```

5. IPCop starten

Wenn die Konfiguration korrekt ist, sollte IPCop nun auch starten:

```
# xm create ipcop
```

Da IPCop aber noch kein gültiges Netzwerkinterface hat, muss man die Grundkonfiguration über eine Xen-Konsole machen:

```
# xm console ipcop
```

Nun befindet man sich auf der Konsole des virtuellen IPCop-Systems und kann die Bootmeldungen verfolgen und ggf. so Fehler feststellen. Verläuft der Bootvorgang korrekt kann man das eigentliche Einrichten von IPCop beginnen indem man sich auf dem

System einloggt und das Setup aufruft.

```
# setup
```

Ist das Setup abgeschlossen sollte man IPCop über das Netzwerk erreichen können. Nun kann man sich auch wieder von der Xen-Konsole lösen und die restliche Konfiguration über das komfortable Webinterface erledigen. Das lösen der Xen-Konsole funktioniert mit STRG+“+“.

Das Webinterface ist erreichbar unter <https://ipcop-ip:445>.

6. Nützliche Links

Nützliche Addons für Ipcop, zum Beispiel den Editor joe - welchen ich bevorzuge - oder auch diverse Netzwerkaddons findet man unter

<http://firewalladdons.sourceforge.net/>

Informationen zum Einrichten einer VPN-Verbindung gibt es hier

<http://vpn.ebootis.de/>

Zu guter letzt gibt es noch ein deutsches Diskussionsforum

<http://www.ipcop-forum.de/>