

# 11 Testumgebungen

Es bietet sich häufig an, bevor ein VPN in einem produktiven Netz genutzt wird, den Einsatz und die Konfiguration in einer Testumgebung zu evaluieren und zu prüfen. Dieses Kapitel stellt verschiedene Möglichkeiten für den Aufbau einer derartigen Testumgebung vor. Hierbei existieren grundsätzlich zwei verschiedene Möglichkeiten. Entweder werden sämtliche benötigten Rechner für den Aufbau des Testfalles in eigenständiger physikalischer Hardware realisiert. Dies benötigt jedoch umfangreiche Hardware Ressourcen, wie Rechner Kabel, Hubs, Switches und so weiter. Von Vorteil ist jedoch, dass direkt die Leistungsfähigkeit der eingesetzten Hardware in der VPN Lösung getestet werden kann. Die Alternative ist eine Virtualisierung der gesamten Testumgebung. Hierbei werden die benötigten Rechner mit Hilfe von Softwareprodukten emuliert. Es werden daher keine zusätzlichen Rechner oder Netzwerkhardware benötigt. Am bekanntesten ist sicherlich die Rechneremulation mit VMware. Dieses Produkt emuliert einen Intel PC und erlaubt die Installation eines beliebigen Betriebssystems. Wenn jedoch nur die Emulation eines Linux Rechners auf der Basis des Linux Betriebssystems gewünscht wird, so genügt in den meisten Fällen User Mode Linux. Diese Linux Variante stellt außerdem geringere Anforderungen an die genutzte Hardware als VMware.

## 11.1 Testumgebungen

Dieser Abschnitt stellt die in diesem Buch verwendete Testumgebung vor und erklärt ihren Aufbau. Diese Umgebungen sind sehr generisch gehalten worden und versuchen die realen Verhältnisse im Internet mit möglichst geringen Mitteln nachzustellen, um möglichst jedes in diesem Buch vorgestellte Szenario nachstellen zu können.

Im Grunde werden zwei verschiedene Testumgebungen benötigt:

- Eine einfache Testumgebung I, bei der zwei Netzwerke über zwei VPN Gateways miteinander kommunizieren. Das Internet wird durch einen zusätzlichen Router zwischen den VPN Gateways emuliert.
- Eine Testumgebung II, bei der einzelne Rechner mit dynamischen IP Adressen über einen Router auf ein VPN Gateway zugreifen.

Wenn die Testumgebungen aufgebaut werden, sollte vor dem Einsatz von FreeS/WAN getestet werden, ob diese Umgebungen funktionieren. Dazu kann mit `ping` die Konnektivität geprüft werden.

### 11.1.1 Testumgebung I

Diese Testumgebung (Abbildung 11.1) stellt den häufigsten Fall einer VPN Lösung in einem Unternehmen dar. Dieses Unternehmen verfügt über zwei lokale Netzwerke. Sie befinden sich in New York und Berlin. Das Netzwerk in New York verwendet die IP Adressen 10.0.1.0/24. Das Netzwerk in Berlin verwendet die IP Adressen 10.0.2.0/24. In beiden Netzen gibt es ein Standard Gateway. Es ist unter der IP Adresse 10.0.1.1 beziehungsweise 10.0.2.1 erreichbar. Beide Gateways verfügen über eine statische IP Adresse im Internet. New York verwendet die IP Adresse 3.0.0.1 mit einer Netzmaske von 255.0.0.0 und einem Standard Gateway von 3.255.255.254. Berlin verwendet die IP Adresse 5.0.0.1/8 mit dem Standard Gateway von 5.255.255.254.

New York und Berlin sind in der Realität über das Internet miteinander verbunden. Hier wird das Internet durch einen Router simuliert, der über zwei Netzwerkkarten mit den IP Adressen 3.255.255.254/8 und 5.255.255.254/8 verfügt.

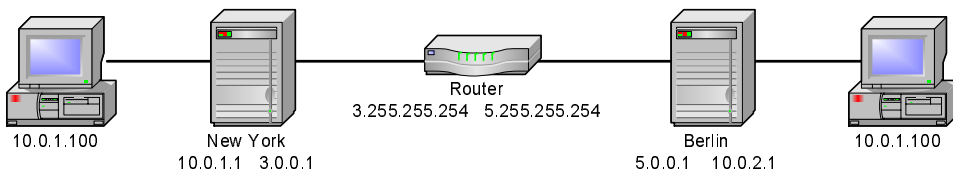


Abbildung 11.1 Testumgebung I

### 11.1.2 Testumgebung II

Diese Testumgebung wird für den Test eines Roadwarrior Szenarios und den Test des NAT Traversal benötigt. Hierbei gleicht die linke Hälfte des Aufbaus der Testumgebung I. Es existiert hier ebenfalls ein Netzwerk New York, das die IP Adressen 10.0.1.0/24 verwendet. Dieses Netzwerk ist über ein Gateway mit dem Internet verbunden. Das Gateway ist dazu mit zwei Netzwerkkarten ausgestattet. Die interne Karte verwendet die IP Adresse 10.0.1.1/24. Die externe Karte verwendet die IP Adresse 3.0.0.1/8 mit einem Standardgateway von 3.255.255.254.

Der Router simuliert erneut das Internet. Über den Router greifen nun Clients mit dynamischen IP Adressen auf New York zu. Hinter den Clients kann sich ein weiteres Netzwerk (Client1, 192.168.3.0/24) befinden. Client1 kann damit auch für den Test des NAT Traversal verwendet werden. Dazu

wird auf Client1 NAT aktiviert. Der VPN Aufbau erfolgt dann von den Rechnern NAT Client1 und NAT Client2 hinter Client1. Client2 ist ein weiterer Client mit dynamischer IP Adresse.

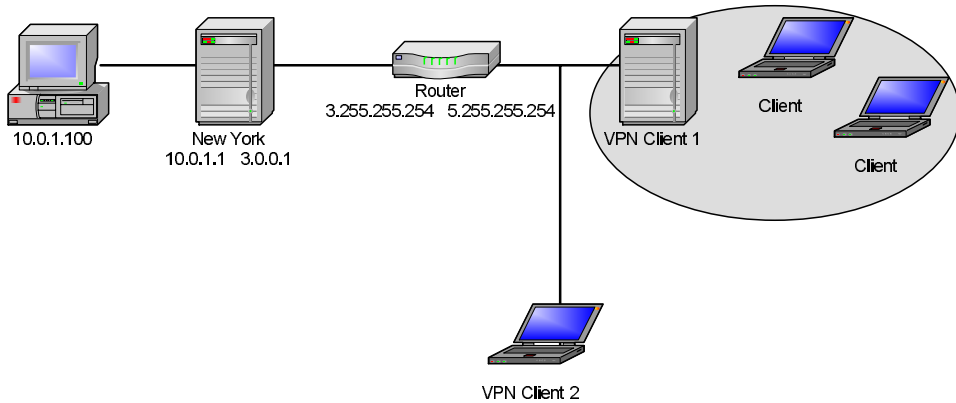


Abbildung 11.2 Testumgebung II

## 11.2 Physikalische Testumgebungen

Es ist möglich diese Testumgebungen physikalisch aufzubauen. Hierzu sind jedoch eine große Anzahl von Rechnern inklusive Hubs, Switches und Kabeln erforderlich. Ein physikalischer Testaufbau erlaubt jedoch eine Abschätzung der tatsächlichen Leistungsfähigkeit der Hardware. So kann geprüft werden, ob die ausgewählte Hardware später im Produktionseinsatz in der Lage ist die gewünschte Leistung zu erbringen.

Für erste Tests ist es jedoch sinnvoller und meist auch schneller und preiswerter, die entsprechenden Aufbauten zu emulieren.

## 11.3 VMware

VMware Workstation ist ein kommerzielles Produkt. Es wird von VMware (<http://www.vmware.com>) hergestellt und stellt das kleinste Produkt aus einer Reihe mit dem VMware GSX Server und dem VMware ESX Server dar. Es erlaubt die Virtualisierung eines Intel PCs auf der Basis von Linux oder Win32 Bit Betriebssystemen. So besteht die Möglichkeit auf diesem virtualisierten PC ein weiteres Intel Betriebssystem (Windows, Linux, \*BSD oder Solaris) zu in-

stallieren. VMware unterstützt dabei eine Vernetzung der virtuellen Rechner.

VMware Workstation ist als Evaluationsversion für 30 Tage erhältlich. Anschließend kostet eine Lizenz 299 Dollar.

Die Installation von Linux oder Windows Betriebssystemen erfordert anschließend keine weiteren Anpassungen. Es wird lediglich empfohlen, die VMware Tools zu installieren. Sie bieten zum Beispiel einen angepassten Grafikkartentreiber, um das Gastbetriebssystem im Vollbild Modus zu betreiben.

Für die Vernetzung bietet VMware die Erzeugung von virtuellen Netzwerken. Dazu werden auf dem Host Betriebssystem virtuelle `vmnetX` Karten erzeugt, die dem Host Zugang zu diesen Netzwerken geben.

Der Aufbau sämtlicher Testumgebungen ist mit VMware Workstation möglich. Für Einzelheiten lesen Sie bitte die VMware Bedienungsanleitung.

## 11.4 User-Mode-Linux

User-Mode-Linux (UML) ist ein Projekt (<http://user-mode-linux.sourceforge.net/>) von Jeff Dike, das den Start eines Linux Betriebssystems auf einem bereits gestartetem Linux System ermöglicht. Auch diese Lösung bietet die Möglichkeit virtuelle Netzwerke aufzubauen. Dabei kann UML auch vollkommen von der Umwelt abgeschnittene Netzwerke aufbauen. Hierzu wird ein Switch Daemon zur Verfügung gestellt.

### 11.4.1 Kernelbau

Damit der Kernel im Usermode funktionieren kann, sind einige Änderungen erforderlich. Ein Großteil des Usermode Codes befindet sich bereits in den aktuellen Kernen. Jedoch ist es für die Funktionalität und Stabilität von Vorteil, den aktuellsten Patch für diesen Zweck zu nutzen. Laden Sie zunächst den aktuellen Linux Kernel von <http://www.kernel.org> und den Usermode Patch von <http://user-mode-linux.sourceforge.net/dl-sf.html>. Hier werden Patches für den Linux Kernel 2.4, 2.5 und 2.6 vorgehalten.

#### TIPP

User Mode Linux unterstützt seit einiger Zeit ein Separate Kernel Address Space (SKAS). Diese Funktion beschleunigt einen User Mode Linux Kernel um 30 bis 50 Prozent. Jedoch ist hierfür ein Patch des Host Kernels erforderlich. Sie können weitere Informationen über SKAS auf <http://user-mode-linux.sourceforge.net/skas.html> nachlesen.

Entpacken Sie den Linux Kernel in einem geeigneten Verzeichnis und patchen Sie ihn mit dem Usermode Patch.

```
# mkdir /usermode
# cd /usermode
# tar -xjf />path</linux-<version>.tar.bz2
# cd linux-2.4.19
# bzipcat />path</uml-patch-<version>.bz2 | patch -p1
# make ARCH=um oldconfig dep
```

Wenn Sie FreeS/WAN benutzen möchten, entpacken Sie nun FreeS/WAN in der gewünschten Version und wenden entsprechende FreeS/WAN Patches an. Anschließend starten Sie die FreeS/WAN Übersetzung. Wenn Sie den Linux Kernel 2.5 oder 2.6 mit dem nativen IPsec Stack nutzen wollen, ist dies nicht nötig.

```
# cd ..
# tar -xzf />path</freeswan-<version>.tar.gz
# cd freeswan-1.99
# make ARCH=um KERNELSRC=../linux-<version> insert mcf confcheck kernel
```

Hierbei kann es zu einem Fehler bei der Übersetzung kommen, da das FreeS/WAN Skript versucht ein bzImage zu generieren. Die Architektur ARCH=um erlaubt aber nur die Erzeugung eines Kernels linux. Um dies anzupassen ist die Datei freeswan-<version>/Makefile.inc zu editieren.

```
KERNEL=$(shell if expr "`uname -m`" : ' i.86' >/dev/null ; \
    then echo linux ; \
    else echo boot ; \
    fi)
```

Geben Sie für die Übersetzung nicht den Befehl `make kinstall` ein. Er wird versuchen den Kernel und die Programme direkt auf ihrem Host System zu installieren.

Wenn das Kernel Konfigurationswerkzeug startet, wählen Sie die Unterstützung für Kernelmodule ab, so das ein monolithischer Kernel erzeugt wird. Dies vereinfacht später die Verwendung des Kernels.

Nun wird der Kernel übersetzt. Dies wird eine Weile dauern. Anschließend sollten Sie den erzeugten Linux Kernel mit einem sinnvollen Namen sichern.

```
# cp ../linux-<version>/linux ../linux-<version>-freeswan-<version>
```

## 11.4.2 UML Installation

Für die Verwendung von User Mode Linux werden einige Werkzeuge auf dem Host benötigt. Dies sind die UML Utilities. Entweder Sie installieren das auf der User Mode Linux Downloadseite verfügbare RPM Paket oder Sie kompilieren die UML Utilities von Hand. Wenn Sie das RPM wählen, sollten Sie daran denken, dass mit diesem RPM auch ein aktueller User Mode Linux Kernel installiert wird. Er befindet sich im Gegensatz zu dem selbst übersetzten Kernel im Suchpfad Ihrer Shell.

Nun müssen Sie noch ein Dateisystem erzeugen. Auf der User Mode Linux Homepage werden einige vorgefertigte Root-Dateisysteme angeboten. Sie können diese aber auch relativ leicht selbst herstellen. Mit `mkrootfs` (<http://www.stearns.org/mkrootfs/>) und UML-Builder (<http://umlbuilder.sourceforge.net/>) stehen Ihnen zwei Anwendungen zur Verfügung, die die Erzeugung stark vereinfachen.

Zusätzlich ist auch eine Swap Partition erforderlich. Sie wird ebenfalls von den Werkzeugen erzeugt, kann aber auch mit dem folgenden Befehl erstellt werden.

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

Auf der CD ist ein kleines einfaches Root Dateisystem enthalten. Es basiert auf der Red Hat Linux 9.0 Distribution.

## 11.4.3 Start von User-Mode-Linux

Nun können Sie User Mode Linux starten. Wählen Sie dazu ein Dateisystem aus. Anschließend können Sie mit dem folgenden Befehl eine User Mode Linux Session starten. Sie sollte dann automatisch über eine Netzwerkverbindung verfügen. Testen Sie sie, indem Sie sich anmelden (*login:root, kennwort:root*) und einen Ping auf ihre Hauptmaschine durchführen. Dort wird automatisch die Netzwerkkarte `tap0` mit der IP Adresse 3.255.255.254 aktiviert.

```
/<path>/linux umid="New-York" root=6200 ubd0=rootfs ubd7=swapfs
eth0=tuntap,,,3.255.255.254
```

## 11.4.4 Aufbau virtueller Netzwerke mit `uml_switch`

User Mode Linux bietet die Möglichkeit komplette eigenständige Netzwerke aufzubauen, die über einen Switch oder ein Hub miteinander verbunden sind. Diese Funktion wird vom Befehl `uml_switch` ausgeübt. Die Kommunikation zwischen den verschiedenen UML Sessions und dem `uml_switch` erfolgt dann über einen Socket. Der Befehl kennt die folgenden Optionen bei seinem Aufruf:

- **-hub** Der Befehl arbeitet als Hub und nicht als Switch.
- **-unix socket** Hiermit können mehrere Switche erzeugt werden. Jeder benötigt einen eigenen Kommunikationssocket. Der kann hier angegeben werden.
- **-t tap-device** Hiermit kann der Switch am Host angeschlossen werden. Dazu ist es erforderlich das Gerät `tapX` zuvor mit dem Befehl `tunctl -u uid` zu erzeugen. Diese Option ist nicht bei allen Versionen von `uml_switch` verfügbar.

Ein typischer Aufruf von `uml_switch` sieht dann wie folgt aus:

```
# uml_switch -unix /tmp/newyorkctl
```

Beim Aufruf von User Mode Linux muss das entsprechende Interface, das an den Switch angebunden werden soll, angegeben werden mit:

```
eth1=daemon,mac-address,,/tmp/newyorkctl,
```

Die `mac-address` muss durch eine eindeutige Adresse ersetzt werden, da ansonsten jeder Rechner, der mit dem Switch verbunden wird die identische Adresse erhält und keine Kommunikation möglich ist.

