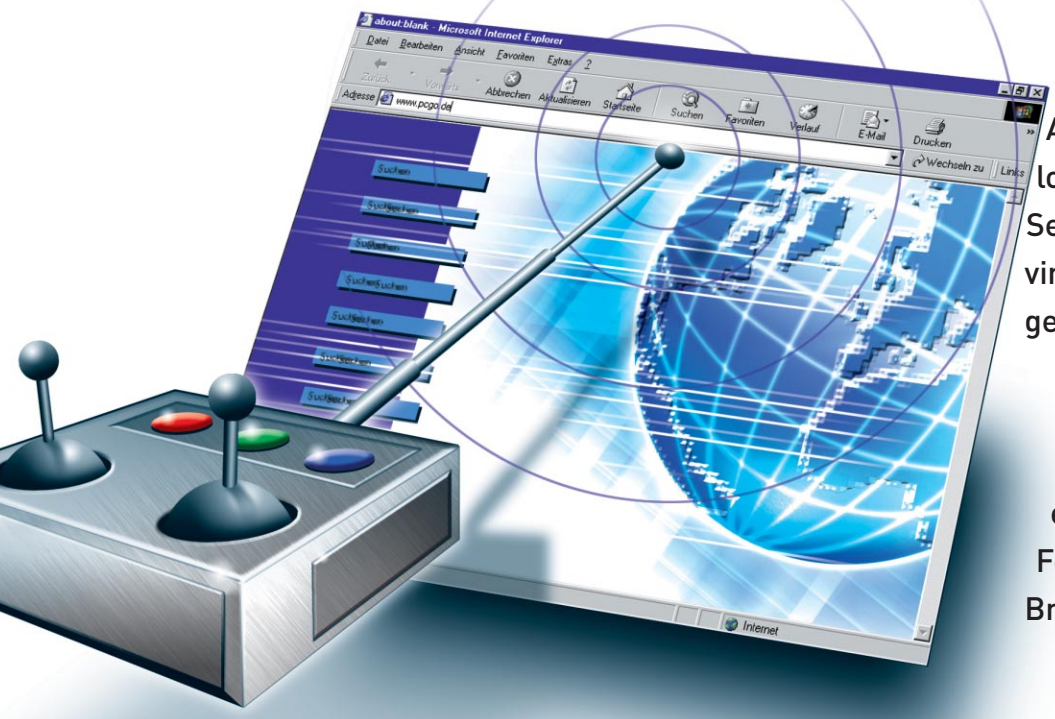
ERSTE SCHRITTE MIT DEM ROOT-SERVER

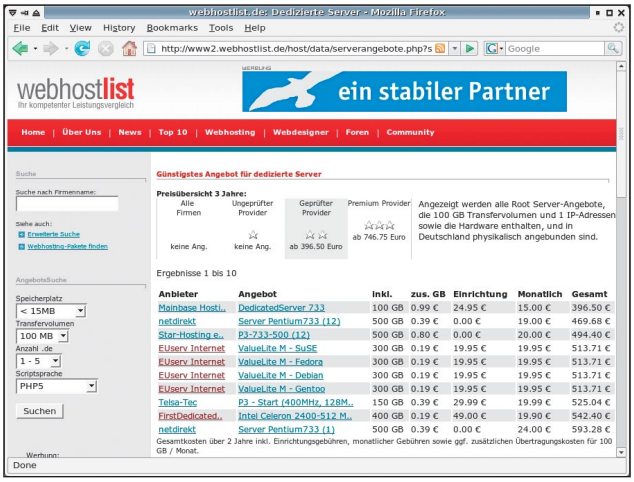
Wurzelbehandlung



Ab 30 Euro pro Monat locken dedizierte Root-Server. Doch der Einzug ins virtuelle Heim will genauso gut geplant sein, wie regelmäßige Sicherheitschecks und Software-Updates. Mit der richtigen Strategie wird der eigene Server zur sicheren Festung gegen die raue Brandung des Internets.

VON **MATTIAS SCHLENKER**

Wer seinen Blog oder ein Bulletin Board eine Zeit lang auf Miet-Webspace betrieben hat, kennt die Einschränkungen derartiger Hosting-Lösungen: Webserver-Konfigurationen, an die der Kunde kaum herankommt, oft suboptimal performante Datenbanken, kaum Eingriffsmöglichkeiten ins Mail-Subsystem und schwer zu realisierende regelmäßige Tasks. Auf dem dedizierten Server ist der Kunde dagegen *root* und kann das System so einrichten, wie es ihm gefällt. Kein amoklaufendes Script anderer Hosting-Kunden zehrt CPU-Leistung auf, und die Bandbreite von 100 MBit/s steht in Stoßzeiten Ihnen ganz alleine zur Verfügung. Vergleicht man die Preise einfacher Root-Server mit denen umfangreicher Hosting-Pakete, scheint die Entscheidung klar: Mehr Rechte für weniger Geld bietet nur der Root-Server. Doch ganz so einfach sollte man sich die Entscheidung für den Root-Server nicht machen.



Verlockend: Auf *webhostlist.de* finden sich Root-Server ab 15 Euro pro Monat.

Der Preis der Freiheit

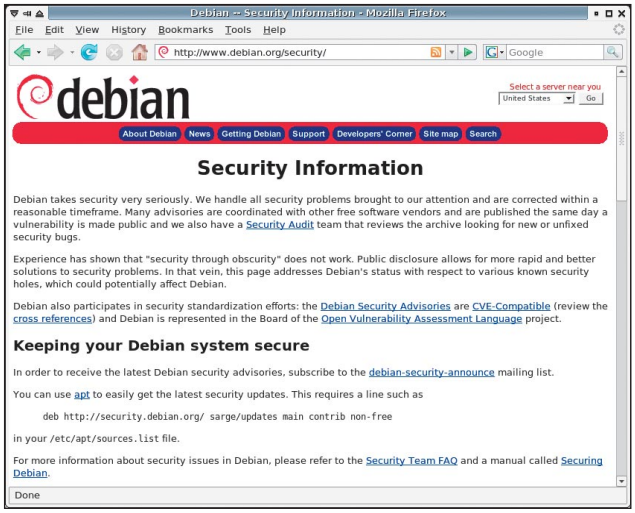
Die Freiheit, den Server ganz nach eigenen Wünschen konfigurieren zu können, birgt die Verantwortung, dies fachkundig und mit Augenmerk auf die Systemsicherheit zu tun. Im Wesentlichen müssen drei Aspekte berücksichtigt werden:

- Jeder unnötige Dienst stellt ein Sicherheitsrisiko dar und sollte deshalb abgeschaltet werden. Administrationswerkzeuge wie Webmin, die zwar einen gewissen Komfort bieten, aber die Angriffsfläche erhöhen, können bei Bedarf aktiviert werden.
- Aktive Dienste müssen abgesichert und oft „verschlankt" werden. So sollten Sie die Konfiguration des SSH-Daemons anpassen und gegebenenfalls nicht benötigte PHP-Module deaktivieren.
- Updates sollten regelmäßig und zeitnah eingespielt werden, veraltete Systeme stellen das größte Risiko dar: Ein lax aufgesetztes Debian 3.1 mit einem veralteten Bulletinboard kann über die Kombination dreier Exploits schrittweise übernommen werden.

Bereits für die zeitnahen Updates sollten pro Woche 30 bis 60 Minuten eingeplant werden. Wird für diese Tätigkeit ein regulärer Stundensatz kalkuliert, übersteigen die Kosten für den Root-Server die von Hosting-Angeboten und „Managed Servern“ schnell. Dazu kommt der rechtliche Aspekt: Zwar entbindet auch das Hosting auf Webspace nicht von regelmäßigen Backups, doch um defekte Hardware oder einen geknackten Rechner und dessen Wiederherstellung muss sich zunächst der Provider kümmern.

Das richtige OS

Ist die Entscheidung für den Root-Server gefallen, steht die Wahl des Betriebssystems an. In einschlägigen Foren wird gerne Debian als „das beste Linux für den Server“ beschrieben – eine Empfehlung, der wir uns nicht anschließen: Debian, Ubuntu, Red Hat, Fedora und openSUSE verfügen alle über ein effizientes Paketmanagement und werden regelmäßig mit Sicherheit-Uupdates versorgt. Sie sollten sich deshalb beim ersten Root-Server für ein System entscheiden, dessen Verwaltungswerkzeuge Sie beherrschen und dessen Software-Angebot die Voraussetzungen der später eingesetzten Scripte erfüllt. Schlimmstenfalls ist ein ungenügend mit Updates versorgtes und mit schlecht eingebundenen Backports verhunztes Debian anfälliger und instabiler als ein einigermaßen ordentlich gewartetes openSUSE, das von Haus aus alle Anforderungen erfüllt.



Admin, sei wachsam: Jeder Distributor unterrichtet auf seiner Seite über aktuelle Patches.

Sehr wichtig bei der Wahl des Betriebssystems ist der verbleibende Support-Zeitraum: Community-Distributionen wie openSUSE, Ubuntu oder Debian werden zwei bis drei Jahre ab Veröffentlichung mit Sicherheits-Updates versorgt. Anbieter von Root-Servern sollten besonders Neukunden deshalb möglichst aktuelle Betriebssysteme vorinstalliert anbieten. Bei unseren Stichproben Anfang Januar stachen zwei Provider heraus: Negativ fiel 1&1 mit dem spätestens Mitte 2007 nicht mehr mit Updates versorgten SuSE 9.3 auf, positiv dagegen Hetzners Angebot, das sowohl openSUSE 10.1, als auch die Preview-Release von Debian 4.0 umfasste.

Immerhin lassen sich auf dem Root-Server – ein halbwegs brauchbares Rettungssystem vorausgesetzt – aktuelle Versionen der Lieblingsdistribution installieren, sodass weder eine ungünstige Partitionierung noch eine veraltete Version der Lieblingsdistribution als Showstopper fungieren. Nach unserer Erfahrung ist die Installation von openSUSE oder Fedora Core etwas einfacher als die Debian basierter Distributionen. Wer jedoch eine gewisse Debian-Erfahrung hat, wird mit den Tools *debootstrap*, *chroot* und einem selbstgebauten Kernel schnell zum Ziel kommen.

Schlüsselübergabe

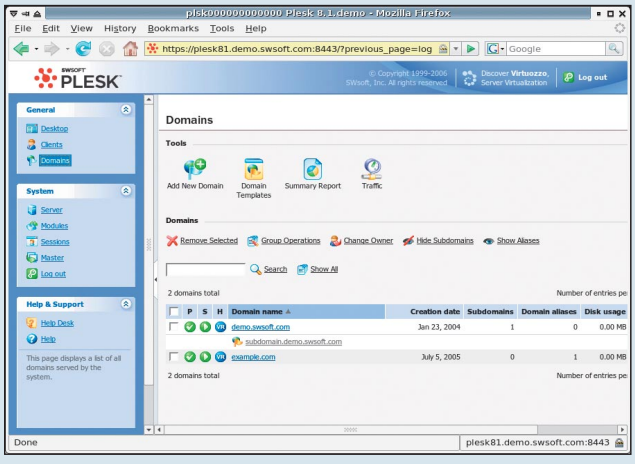
Ist die Wahl für den „richtigen“ Provider gefallen und der Vertrag abgeschlossen, erhalten Sie in der Regel die Zugangsdaten für den Kundenbereich des Providers, wo sie das

Trügerischer Komfort: Administrations-Frontends

Viele Anbieter von Root-Servern offerieren vorinstallierte Webfrontends für Administrationsaufgaben wie *Confixx* oder *Plesk*. Mit diesen Werkzeugen soll das Anlegen von Domains, assoziierten Nutzern und E-Mail-Konten vereinfacht werden. In der Regel erledigen die Tools diese Aufgaben zumindest für nicht allzu exotische Konstellationen gut, weshalb sie gerne eingesetzt werden.

Langfristig droht aber Ungemach durch den Lock-In-Effekt: Die vom Hoster überlassenen Lizenzen beziehen sich meist nur auf eine bestimmte Major-Version des Werkzeuges. Installiert man eine neue Betriebssystemversion, weil die alte nicht mehr mit Sicherheits-Updates versorgt wird, hat man die Wahl zwischen dem Kauf einer neuen Version des Administrationstools und der manuellen Konvertierung aller Einstellungen. Selbst mithilfe einiger Scripte kann diese Aufgabe auf durchschnittlich ausgelasteten Root-Servern mit einem Dutzend Domains und einem Gros E-Mail-Accounts viele Stunden in Anspruch nehmen.

Ein weiteres Problem stellen kollidierende Einstellungen dar: Versierte Webmaster passen Apache-Direktiven gerne von Hand an. Viele Administrationswerkzeuge überschreiben diese aber. Eine Konfiguration per *.htacces* ist jedoch oft aus Performance-Gründen und, weil dies dem Kunden zu viele Optionen eröffnet, nicht vorgesehen. Wird der Server von einem Linux-kundigen Administrator verwaltet, sollten Sie deshalb auf Web-Frontends verzichten. Sind Administrationsfrontends notwendig, weil Kunden eines untervermieteten Servers selbst E-Mail-Accounts anlegen sollen, müssen zumindest potenzielle Probleme und Kosten bei künftigen Updates einkalkuliert werden.



Bequem: *Plesk* und *Confixx* vereinfachen Verwaltungsaufgaben, erschweren aber Betriebssystemwechsel.

Unter Debian erledigt *apt* die Sicherheits-Updates schnell und unkompliziert - zwei Zeilen genügen, und Sie sind auf dem neuesten Stand.

Root-Passwort Ihres Servers abfragen können oder das Root-Passwort mit separater Post. Im Kundenbereich haben Sie die Möglichkeit, den Rechner sanft zu rebooten (*Strg-Alt-Entf*) oder ihn zu resetten sowie ein per PXE aus dem Netzwerk gestartetes Rettungssystem zu aktivieren.

Mit dem zugesandten Passwort steht das erste Login als *root* am neuen Server an. Als Erstes sollten Sie das SSH-Login absichern. Ideal ist es, einen zweiten unpriviligierten Nutzer anzulegen (Befehl *adduser* oder *useradd*), der mit dem Kommando *su* Root-Rechte erlangen kann. Ist diese Möglichkeit getestet, erzeugen Sie auf Ihrem Arbeitsplatz-PC zu Hause mit dem Befehl

```
ssh-keygen -t dsa
```

ein Schlüsselpärchen für das passwortlose SSH-Login. Führen Sie den gleichen Befehl auch als *root* auf Ihrem Server aus. Anschließend erlauben Sie das passwortlose Login, indem Sie den öffentlichen Teil Ihres Schlüssels den erlaubten Schlüsseln hinzufügen:

```
cat .ssh/id_dsa.pub | \
ssh root@123.45.67.89 \
„cat >> /root/.ssh/authorized_keys“
```

Mit den Direktiven

```
PermitRootLogin without-password
PubkeyAuthentication yes
```

sollte – nach einem Restart des SSH-Daemons – das passwortlose Login funktionieren. Wer seine SSH-Logfiles etwas sauberer von automatisierten Login-Versuchen halten möchte und bei möglichen Schwächen im SSH-Daemon nicht zu den allerersten Opfern gehören will, verlegt zudem seinen SSH-Server auf einen anderen Port.

Der nächste Schritt sollte die Aktualisierung der installierten Anwendungs-Software sein. Hierfür müssen Sie zunächst prüfen, ob Server für das Online-Update konfiguriert sind. Unter SUSE erledigt dies das YaST-Modul *YOU Konfiguration*, unter Debian und Ubuntu sollten Sie einen Blick in die */etc/apt/sources.list* werfen und sich davon überzeugen, dass für jeden „Stream“ ein Eintrag für Security-Updates der Form

```
deb http://security.ubuntu.com/ubuntu
➥dapper-security main restricted
```

existiert. Unter Debian/Ubuntu startet die Befehlsfolge

```
apt-get update
apt-get upgrade
```

Root-Server-Alternativen

Noch vor Jahresfrist war es möglich, bei vielen Providern Root-Server zu 15 EUR/Monat zu ordern. Mittlerweile haben sich die Preise bei den großen Anbietern wieder zwischen 30 und 40 EUR/Monat eingependelt. Die Rolle der Einsteiger-Root-Server haben mittlerweile die *vServer* zu Preisen ab 7 EUR/Monat übernommen. Bei ihnen handelt es sich um virtualisierte Systeme: Zwischen vier und vierzig Kundensysteme teilen sich in der Regel einen physikalischen Server. In der Theorie stehen vServer nicht nur aus ökologischen und ökonomischen Gründen gut da: Sie kombinieren die Vorteile von Shared Hosting (gute Ausnutzung vorhandener Hardware) mit den Vorteilen echter Root-Server (fast volle Autonomie über die Konfiguration). In der Praxis sind jedoch auch die Nachteile beider Konzepte spürbar: Als Kunde müssen Sie sich um Updates kümmern und ein amoklaufendes Script in einem anderen vServer kann die Performace des Gesamtsystems spürbar einschränken. vServer eignen sich deshalb grundsätzlich für Aufgaben, die moderat mit Arbeitsspeicher und Rechenleistung umgehen, also Blog, Gallery, PHP-Experimente und das eigene Mailsystem. Für komplexe Foren mit vielen Nutzern sind sie genauso wenig geeignet wie für Mailsysteme mit extrem aufwändiger Spam-Filterung. Da kaum ein Anbieter präzise Angaben über die exakte Zahl der vServer pro physikalischer Maschine macht, sollten Sie beim vServer nur Angebote in Betracht ziehen, die eine sehr kurzfristige Kündigung während der ersten drei Monate zulassen. Nur so können Sie selbst herausfinden, ob Ihre Leistungsansprüche befriedigt werden.

Kompakte Alternative: Bei hohem Upstream und einer moderaten Last kann bereits eine *NSLU2* einen Root-Server überflüssig machen.

Eine weitere Alternative ist „Hosting zu Hause“: Mit Upstreams zwischen 1MBit/s (DSL 16000) und 2,3 MBit/s (VDSL, mittelfristig 6,3MBit/s), kombiniert mit dynamischen DNS-Diensten, können Blogs, einfache Firmenseiten und simple Fotoalben auch zu Hause gehostet werden. Einziger Haken ist die Leistungsaufnahme eines rund um die Uhr laufenden Rechners, die zwischen 30 Watt (optimiertes EPIA-System) und 200 Watt (unoptimierter Gamer-PC) liegen wird. Zumindest letzterer kommt im 24/7-Betrieb bei 20 ct/KWh alleine mit den Stromkosten bedrohlich nahe an die Monatsmiete eines Root-Servers...

Strom lässt sich mit der Um- und Aufrüstung von NAS-Servern wie der NSLU2 (ab ca. 12 Watt) sparen, sogar modifizierte FRITZ!Boxen mit Apache-Servern wurden schon gesichtet. Allen Zu-Hause-Lösungen gemein ist die geringe „Aufstiegsmöglichkeit“, sollte der Anspruch an Rechenleistung oder Bandbreite steigen.

notwendige Aktualisierungen, unter SUSE ist das „YaST Online Update“ zuständig, erst ab openSUSE 10.1 steht mit *rug* ein Kommandozeilentool debianesker Art zur Verfügung:

```
rug update
```

Mit der soeben vorgenommenen Grundsicherung und einem geänderten Root-Passwort haben Sie potentiellen Einbrechern die Arbeit bereits deutlich erschwert.

Schotten dicht

Dennoch sollten Sie das System dahingehend prüfen, dass keine unnötigen Dienste an externen Netzwerkschnittstellen lauschen. Einen Überblick verschafft der Befehl

```
lsof -i
```

der alle bereitgestellten Netzwerkdienste und aktive Verbindungen auflistet. Besonders interessant sind die mit *(*LISTEN) *gekennzeichneten Dienste. Meist können Sie* Webmin (Port *9000*) deaktivieren und nur bei Bedarf temporär einschalten.

Auch der oft aktivierte FTP-Server ist in den meisten Fällen unnötig: Auf dem System angelegte Nutzer können per SCP oder SFTP ihre Daten zum Server transferieren – hierfür muss lediglich der SSH-Daemon aktiviert sein.

Downloads bis zwei Gigabyte Größe können meist per HTTP genauso effizient zur Verfügung gestellt werden. SCP-Clients existieren für alle gängigen Systeme, unter Win-dows ist *WinSCP* sehr beliebt, unter Linux unterstützt der Konqueror *SFTP*.

Wird das auf dem Root-Server installierte Mailsystem zunächst nur verwendet, um Statusmeldungen zu verschicken, sollten Sie Port *25* (und gegebenenfalls *587*) an *localhost* binden.

Gleiches gilt für möglicherweise erreichbare IMAP- und POP-Server.

Bei nicht benötigten Diensten ist die vollständige Deinstallation meist der effizienteste Weg, den Dienst loszuwerden.

Wird ein Dienst gelegentlich gebraucht, reicht die Löschung des korrespondierenden Softlinks im *Init*-Verzeichnis des Default-Runlevels, eine Aufgabe für die SUSE einen Runlevel-Editor bietet.

Zu guter Letzt sollten Sie prüfen, ob der Meta-Daemon *inetd* oder *xinetd* Dienste beim Zugriffsversuch startet.

Liefert der Befehl

```
ps waux | grep inetd
```

einen aktiven Prozess, werfen Sie bitte einen Blick in die Konfigurationsdatei */etc/xinetd.conf* und die in */etc/xinetd.d* eingebundenen Konfigurationsdateien und setzen Sie unbenötigte Dienste auf

```
disable = yes
```

Zugemauert?

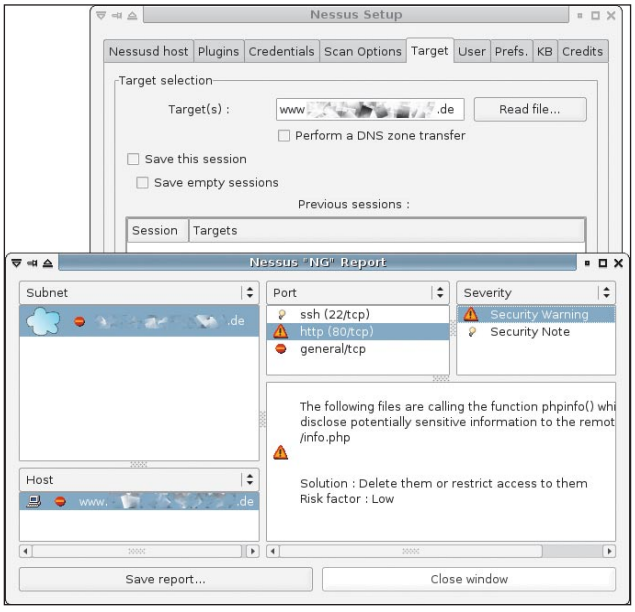
Viele Tutorials empfehlen die Einrichtung einer Firewall auf dem Root-Server.

Diese Maßnahme wird oft überbewertet, da wirksame Angriffe meist über geöffnete Dienste wie den Webserver erfolgen und in der Regel schlecht programmierte Scripte zum Ziel haben. Eine Firewall würde einen derartigen Angriff nicht abwehren können. Sie kann jedoch verhindern, dass ein vom Angreifer geöffneter IRC-Server Befehle entgegennimmt.

Verwenden Sie den Paketfilter des Linux-Kernels, sollten Sie sich jedoch bewusst sein, dass ein Angreifer, der Root-Rechte erlangt hat, auch die Firewall deaktivieren kann. Mehr Schutz bieten deshalb vorgeschaltete Hardware-Firewalls, wie sie bei 1&1 zum Lieferumfang gehören.

Doch auch diese sind nicht perfekt: Mittelmäßig begabte Hacker können eine ausgehende Verbindung als Tunnel für eingehende verwenden – bereits ein Blick in die Manual Page von *ssh* offenbart diese Möglichkeit.

Firewalls können deshalb bestenfalls als kleines Mosaiksteinchen eines umfassenden Sicherheitskonzeptes gelten.



Abgeklopft: Ein Scan mit einem Tool wie Nessus findet unnötige und veraltete Dienste.

Rettungsboot

Trotz aller Vorsicht kann ein Server kompromittiert werden, Hardware geht gelegentlich kaputt, Updates schlagen manchmal fehl oder eine profane Fehlkonfiguration macht ein System unbrauchbar.

Um Reparaturarbeiten vornehmen zu können, sollten Sie sich mit dem von fast allen Providern bereitgestellten Rettungssystem vertraut machen. Starten Sie das Rettungssystem, loggen Sie sich an ihm ein und mounten Sie die Partitionen entsprechend der */etc/fstab* Ihres Servers, beispielsweise

```
mkdir /tmp/server
mount /dev/hda3 /tmp/server
mount /dev/hda1 /tmp/server/boot
mount /dev/hda5 /tmp/server/home
```

Unter /tmp/server ist nun die Verzeichnisstruktur des Root-Servers zu finden.

Sie können diese (mit Root-Rechten) mit dem Tool *rsync* auf Ihrem PC zu Hause sichern:

WEITERFÜHRENDE LINKS

www.webhostlist.de
Bei der Wahl des richtigen Providers hilft Marktübersicht und die Foren von Webhostlist.

www.rootforum.de
Technische Fragen rund um den Root-Server sind im Root-Forum gut aufgehoben. Da viele Forenteilnehmer bei Hostern arbeiten, sind Fragen nach dem „besten“ Provider hier unerwünscht.

http://blog.rootserverexperiment.de/2006/07/26/ubuntu-606-lts-auf-strato-Root-Server
Die Installation von Ubuntu auf einem Root-Server erklärt das Root-Serverexperiment.

```
rsync -avzHP root@server:/tmp/server/ \
/tmp/serverbackup/
```

Um das so angefertigte Backup zurückzuspielen, genügt es, auf dem Root-Server die Partitionen neu zu formatieren, wie oben zu mounten und *rsync* in der Gegenrichtung aufzurufen:

```
rsync -avzHP /tmp/serverbackup/ \
root@server:/tmp/server/
```

Anschließend muss nur noch der Bootloader geschrieben werden, beispielsweise bei GRUB im Rettungssystem mit dem Kommando

```
grub-install –recheck --no-floppy \
--root-directory=/dev/hda3 /dev/hda
```

Ein Vollbackup ist immer nach größeren Änderungen angeraten, die tägliche Sicherung kann ebenfalls mit *rsync* auf den Verzeichnissen */var*, */etc* und */home* durchgeführt werden.

Um keine korrupten Dateien zu erhalten, sollten Sie Web-, Mail- und Datenbankserver während des Laufes von *rsync* abschalten. Da *rsync* nur Änderungen an Dateien überträgt, geht das Backup meist recht schnell.

Ein weiteres sinnvolles Hilfsmittel sind serielle Konsolen.

Mit diesen kann per *SSH* auf den Bootloader und die Bootmeldungen des Kernels zugegriffen werden.

Ein Login und Wartungsarbeiten sind auch möglich, wenn die Initialisierung des Netzwerkes fehlschlägt. Ohne serielle Konsole gleicht der erste Start eines selbstgebauten Kernels einem Blindflug. Stellt der bevorzugte Hoster keine serielle Konsole zur Verfügung, sollten Sie nachfragen, ob bei Problemen schnell und günstig eine KVM-over-IP Box wie LARA bereitgestellt werden kann.

Fazit

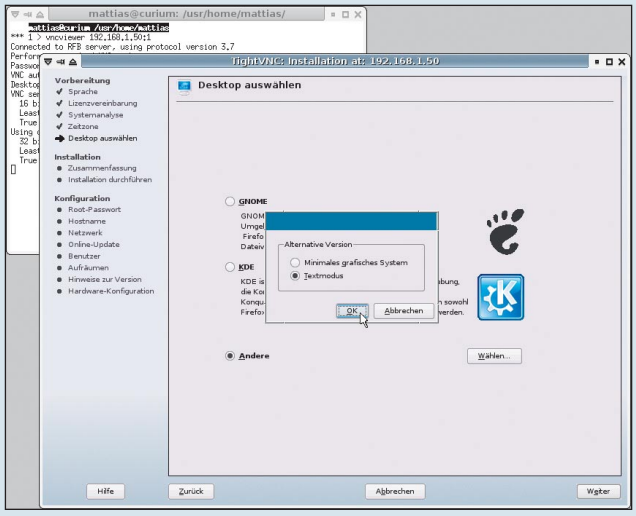
Auch die beste Absicherung stellt keinen Dauerzustand dar. Wer einen komplexen Root-Server betreibt, muss das Betriebssystem stets aktuell halten, nach Fehlern in installierten Webanwendungen Ausschau halten und immer für den Notfall gewappnet sein.

Letztlich gilt es, etwas flinker und etwas gründlicher als die bösen Buben zu sein, die Ihren Server als potentielle Spamschleuder oder als DDoS-Werkzeug betrachten.

Gelingt es, den Vorsprung zu halten, ist der eigene Root-Server die günstigere, flexiblere und schnellere Alternative zum Shared Hosting. **jkn**

Frisch installiert

Jede vorinstallierte Distribution erreicht irgendwann das Ende ihres Lebenszyklus und wird nicht mehr mit Sicherheits-Updates versorgt. Da kaum ein Hosting-Unternehmen aktuellste Distributionen installiert, tickt nach etwa 18 Monaten die Uhr. Mit etwas Glück pflegt der Hoster fertige Images aktueller Betriebssysteme, die per Webinterface auf den Server aufgespielt werden können. Wenn das nicht der Fall ist, können Sie eine aktuelle Distribution wie openSUSE aufspielen. Eine ferngesteuerte Netzwerkinstallation macht es möglich. Voraussetzung auf dem alten System ist lediglich der Bootloader *Grub*. Die Vorgehensweise bei Fedora Core ist prinzipiell gleich, sie verwendet jedoch andere Bootparameter.



Als säße man davor: openSUSEs Netinstall lässt sich per VNC fernsteuern.

- **Laden Sie von einem openSUSE-Mirror, beispielsweise**

```
ftp://ftp.hosteurope.de/mirror/ftp.
opensuse.org/distribution/10.2/repo/
oss/boot/i386/loader
```

den Kernel („linux“) und die initiale Ramdisk („initrd“) herunter und kopieren Sie beide unter eindeutigen Namen in Ihren Boot-Ordner, beispielsweise:

```
/boot/linux102
/boot/initrd102
```

- **Notieren Sie die Netzwerkparamter Ihres Servers: IP-Adresse und Netzmaske zeigt der Befehl**

```
ifconfig -a
```

Den Gateway ermitteln Sie mit

```
netstat -rn
```

und über den primären Nameserver gibt ein Blick in die */etc/resolv.conf* Auskunft.

- **Ergänzen Sie Ihre */boot/grub/menu.lst* um einen Eintrag für das Installationssystem:**

```
title opensuse-install
kernel (hd0,0)/boot/linux102
➥ splash=silent showopts lang=de
➥ vga=normal vnc=1 vncpassword=
➥ geheim78 install=ftp://ftp.host
➥europe.de/mirror/ftp.opensuse.org
➥/distribution/10.2/repo/oss hostip=
➥ ip-adresse/27 gateway=gateway-ip
➥ nameserver=nameserver-ip
initrd (hd0,0)/boot/initrd102
```

Die Zahl hinter dem der IP-Adresse gibt die Netzmaske in Binär-Notation an: 24 Bit bedeutet 255.255.255.0, 27 Bit bedeutet 255.255.255.224, 32 Bit 255.255.255.255. Das VNC-Passwort sollte 8 Zeichen lang sein, und natürlich muss die Bootpartition (hier *(hd0,0)* der tatsächlichen Boot-Partition entsprechen.

- **Ist der Eintrag hinzugefügt, muss der Standardeintrag auf den Installationskernel gesetzt werden, beispielsweise:**

```
default 3
```

Achtung: Grub beginnt bei *0* mit der Zählung!

- **Nun können Sie den Server neu starten. Der braucht einige Minuten, bis das Installationssystem – immerhin 80 Megabyte – geladen ist. Anschließend können Sie sich per VNC mit dem vorher gesetzten Passwort einloggen:**

```
vncviewer 123.45.67.89:1
```

Die Installation verläuft weitgehend so, als säßen Sie direkt am Server. Für den Server genügt ein minimales System ohne X.

- **Auch bei der Installation per VNC ist ein Neustart erforderlich. Die Verbindung zum Server wird abreißen, erst nach etwa zwei Minuten können Sie mit der Einrichtung fortfahren. Bei den Netzwerkeinstellungen sollten Sie *SSH* zulassen. Die Fernwartung (*VNC*) muss wenigstens bis zum Abschluss der Installation ebenfalls offen bleiben.**