Zugriff von „außen“ auf eine DSL-Verbindung

Christoph Stockmayer, Ingenieurbüro Stockmayer, Schwaig

DSL-Verbindungen stellen praktisch eine permanente Verbindung zum Internet dar. Problematisch ist, daß die IP-Adresse dynamisch vom Provider zugewiesen wird und bei vielen Providern auch einmal am Tag geändert wird. Dadurch ist es schwierig, von „außen“ sich einzuwählen, da die IP-Adresse unbekannt ist. Nun gibt es die Möglichkeit, einem Namensserver die gerade aktuelle IP-Adresse mitzuteilen, der sie in seiner Datenbank updatet. Dies ist aber idR nur gegen ein Entgeld möglich und oft ist der Internetname dann auch eingeschränkt.

Hat man jedoch bereits eine www-Adresse und einen schreibenden Zugriff darauf, funktioniert ein anderer Weg – und das mit den Bordmitteln von UNIX/Linux.
Grob funktioniert das so:

- Das UNIX/Linux-System schaut sich zyklisch die IP-Adresse an (mit `cron`)
- hat sich die IP-Adresse geändert, wird eine HTML-Seite aufgebaut und dem eigenen Web-Server zugesandt
- der Firewall auf dem DSL-Rechner wird z.B. für `ssh` oder für `www` aufgemacht
- `ssh` kann so konfiguriert werden, daß weder Passwort noch eine Host-Bestätigung notwendig sind
- soll eine Verbindung aufgebaut werden, schaut man zunächst auf die HTML-Seite, bekommt die aktuelle IP-Adresse und kann damit die Verbindung aufbauen

Nun im Detail:

1. IP-Adresse des DSL-Anschlusses holen

Die IP-Adresse des DSL-Anschlusses bekommt man am einfachsten aus dem Kommando `ifconfig`, das alle Netzwerkschnittstellen beschreibt. Dazu filtert man den Eintrag, der mit `ppp0` beginnt (idR der DSL-Anschluß) und entnimmt ihm die IP-Adresse. Folgendes Shell-Skript führt dies durch:

```
#!/bin/sh

TMP=/tmp/ip.html               # HTML-Datei
HOME=/home/sto
PATH=$HOME/bin:$PATH

IP=`/sbin/ifconfig |
           awk '
              BEGIN             { flag = 0 }
              /^ppp/            { line = NR; flag = 1 }
              line+1 == NR &&
              flag == 1         { flag = 0; print; exit }
               ' |
           awk '{ print $2 }' |
           awk -F: '{ print $2 }'`
```

Da die IP-Adresse in der nächsten Zeile steht, muß etwas getrickst werden mit `awk`. Dann wird aus der Zeile die Adresse extrahiert (per Kommandosubstitution) und der Variablen IP zugewiesen.

2. HTML-Seite mit IP-Adresse aufbauen und absenden

Nun kann die HTML-Seite aufgebaut werden (mit einem `here`-Dokument der Fortsetzung des Shell-Skripts):

```
cat > $TMP <<!
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
<HEAD>
    <META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html; charset=iso-8859-15">
    <TITLE></TITLE>
    <META NAME="GENERATOR" CONTENT="OpenOffice.org 1.0.1  (Linux)">
    <META NAME="AUTHOR" CONTENT="Christoph Stockmayer">
    <META NAME="CREATED" CONTENT="20021218;21340200">
    <META NAME="CHANGEDBY" CONTENT="Christoph Stockmayer">
    <META NAME="CHANGED" CONTENT="20021218;21352600">
</HEAD>
<BODY LANG="de-DE">
    <P STYLE="text-decoration: none">Die momentane Internetadresse
     lautet: <A HREF="http://${IP}/">${IP}</A></P><br>
     Datum: `date`
</BODY>
</HTML>
!
```

In den Text des `here`-Dokuments wird die Shell-Variable IP eingefügt (`${IP}`, und auch noch das Datum, wann diese Änderung gemacht wurde – zur Info) – und gespeichert in einer temporären Datei. Damit der Update auf dem Web-Server nicht unnötig oft geschieht, wird die IP-Adresse auch in einer Datei gespeichert (`$HOME/lib/ip`) und verglichen: erst wenn sich eine Änderung ergeben hat, erfolgt ein Web-Zugriff.

```
if [ $IP ]
then
    if [ "$IP" != "`cat $HOME/lib/ip`" ]
    then
        ftp -A -u ftp://user:passwd@ip-des-webservers/htdocs/ip/index.html \
          /tmp/ip.html
        echo $IP > $HOME/lib/ip
    fi
fi

rm -f $TMP
```

Wird also eine Änderung festgestellt, schreibt `ftp` die Seite an den Web-Server (bei `ftp`-Zugriff). Wird der ftp von Luke Mewburn (lukem@netbsd.org) vom NetBSD Projekt verwendet (bis SuSE7.3 Standard), können Username und Passwort auf der Kommandozeile mitgegeben werden; sonst muß eine `~/.netrc`-Datei diese Daten liefern:

```
machine ip-des-webservers login user password passwd
macdef init
epsv4
cd htdocs/ip
put /tmp/ip.html index.html
quit
```

(Man beachte die abschließende Leerzeile!)

Dieses Shellskript wird nun zyklisch von `cron` ausgeführt (und vielleicht sofort nach einem Neustart der DSL-Verbindung):

```
30 6-20 * * * /home/sto/bin/ip_update
```

Einzutragen ist diese Zeile im Kommando `crontab -e`.

3. Firewall-Einstellungen

Natürlich sollte die DSL-Verbindung mit einem Firewall abgesichert werden (z.B. SuSE-Firewall2). Will man sich aber von außen einwählen (sicherheitshalber nur mit `ssh`), muß für diesen Dienst der Firewall geöffnet werden. Dies wird (im Beispiel des SuSE-Firewalls2) in der Konfigurationsdatei `/etc/rc.config.d/firewall2.rc.config` (bis SuSE7.3) bzw. `/etc/sysconfig/SuSEfirewall2` (ab SuSE8.0) erreicht durch:

```
...
# 9.)
# Which services ON THE FIREWALL should be accessible from either the internet
FW_SERVICES_EXT_TCP="ssh"
...
```

Oder möchte man auch Zugriff auf den eigenen Web-Server gestatten:

```
FW_SERVICES_EXT_TCP="ssh www"
```

Dann sollte allerdings auch noch ein automatisches Weiterleiten in der oben erzeugten HTML-Seite realisiert werden.

4. `ssh`-Konfiguration

`ssh` ist in den meisten Linuxen bereits standardmäßig eingerichtet, sodaß hier sofort losgelet werden kann. Im Falle des Einwählens mit der Secure-Shell `ssh` kann man das Login ohne Passwortabfrage aufbauen. Dazu ist zunächst ein Schlüsselpaar zu erzeugen mit

`ssh-keygen` (bis SuSE7.3) oder
`ssh-keygen -t rsa1`
`ssh-keygen -t rsa`
`ssh-keygen -t dsa` (ab SuSE8.0)

Der öffentliche Schlüssel davon (aus den Dateien `~/.ssh/id_dsa.pub` `~/.ssh/id_rsa.pub` `~/.ssh/identity.pub`) muß auf sicherem Weg zum einzuwählenden Rechner transportiert und dort in der Datei `~/.ssh/authorized_keys` abgelegt werden. Nun ist eine sichere (verschlüsselte) Verbindung direkt möglich.

Auch die Abfrage nach dem Merken des einzuwählenden Rechners kann umgangen werden (da sich die IP-Adresse ja ändert ist das für `ssh` immer wieder ein neuer Rechner), indem man dem ssh-Server dies in der Konfigurationsdatei `/etc/ssh/sshd_config` mitteilt:

```
IgnoreUserKnownHosts yes
```

Nicht vergessen, nach einer Änderung der Konfig-Datei dies dem Server mitzuteilen:

`rcsshd reload` oder
`rcsshd restart`

Beziehungsweise kann im Client dies in der Konfigurationsdatei `/etc/ssh/ssh_config` erreicht werden durch:

```
CheckHostIP no
StrictHostKeyChecking no
```

Ein Neustart des Servers ist dabei nicht notwendig.

5. Aufbau einer Verbindung

Vor einem Verbindungsaufbau steht nun das Holen der momentanen IP-Adresse. Dies kann durch einen Web-Browser gemacht werden `http://www.eigener-server.de/ip`, oder wieder durch ein kleines Skript:

```
#!/bin/sh
# holt IP-Adresse: Name 'ip'

cd /tmp
rm -f index.htm*
wget http://www.eigener-server.de/ip/index.html > /dev/null 2>&1
grep "HREF=" index.html | awk -F"[<>]" '{ print $3 }'
```

Danach steht dem Verbindungsaufbau nichts mehr im Weg:

`ssh -2 ip-adresse` oder
`ssh -2 \`ip\`` bzw. für das Protokoll 1
`ssh -1 \`ip\``

`\`ip\`` ist eine Kommandosubstitution, die zuerst das Skript `ip` ausführt (s.o.) und mit dem Ergebnis in den ssh-Befehl reingeht.

Selbst ein graphisches Programm funktioniert (gleich mit Angabe des Benutzernamens):

```
ssh -2 `ip` -l name -X
kcalc
```

Dipl. Ing. Christoph Stockmayer
Ingenieurbüro Stockmayer GmbH
Dreihöhenstr. 1
90571 Schwaig
Tel.: 0911/505241
Fax: 0911/5009584
eMail: sto@stockmayer.de
Web: http://www.stockmayer.de