call-back mini HOWTO

Pawel Skonecki, stona@kft.umcs.lublin.pl,
Vertaald door: Ellen Bokhorst, bokkie@nl.linux.org

v2.1a, 2001-10-06

In dit document wordt beschreven hoe call-back in te stellen door gebruik te maken van het Linux systeem en een modem. Ik wil hierbij Anna bedanken voor haar geduld.

Inhoudsopgave

1 Introductie

1.1 OPINIE

Ik heb getracht zo volledig mogelijk informatie bijeen te vergaren. Laat het me weten als je fouten aantreft. Ik zal de mensen dankbaar zijn die me enige suggesties of correcties toezenden. Hun bijdragen zullen dit document verbeteren. Ik vind het niet erg je vragen te beantwoorden, maar het is beter als je eerst het gehele artikel leest.

1.2 PUBLICATIE

Dit document kan worden gepubliceerd onder de voorwaarden van het Linux Documentatie Project. Neem contact op met de auteur als je niet aan de licentie kunt komen. Dit document is vrij.

2 Procedure

2.1 DEEL I: Net thuis ?

De meesten van ons gebruiken het Internet op het werk. We hebben het Net echter vaak thuis of buiten het werk nodig. Het kan zijn dat het thuiswerken goedkoper is dan vanuit kantoor. Ik denk dat call-back software op de Linux-server installeren de beste oplossing is. Call-back maakt het mogelijk op kosten van het bedrijf terug te worden gebeld. Ik zal proberen aan te geven hoe het werkt. Een daartoe bevoegd persoon die de modem opbelt, wordt voor de eerste maal op de Linux server geverifieerd. Dan wordt aan de gebruikerskant de "hang up modemïngeschakeld. Tegelijkertijd belt Linux de gebruiker. De gebruiker wordt nogmaals geverifieerd. We hebben verbinding en de server krijgt de rekening. De gebruiker betaalt alleen voor de totstandkoming van de verbinding. De dubbele verificatie en extra opties in het call-back programma deactiveren voor niet daartoe bevoegde personen je een rekening ten laste te brengen. We kunnen de toegang tot de verbinding beperken tot alleen het bedrijfsnetwerk of het Internet. Call-back is zeer flexibel. Hieronder zal ik de configuratie van een call-back server op een Linux systeem aan proberen te geven en zal ik je laten zien hoe je je computer instelt voor het terugbellen van de verbinding. Ik geef geen beschrijving van de configuratie van ISDN call-back omdat ik geen gebruik maak van ISDN in mijn verbinding met het Internet. Als je call-back instelt op ISDN stuur me dan je configuratie. Ik ondervond wat problemen toen ik overging van kernel 2.2.x naar 2.4.x. Ik zal de nieuwe optie voor een nieuwe kernel beschrijven. Denk eraan dat wanneer je overgaat naar een hogere kernelversie, je ook pppd moet upgraden. Ik maak geen nieuwe sectie aan voor een beschrijving van de nieuwe mogelijkheid in kernels 2.4.x, maar zal de nieuwe configuratie in de oude sectie beschrijven. Ik wil me bij deze verontschuldigen bij mensen die me vragen stelden over opties voor nieuwe kernels. Ik had geen tijd een nieuwe versie van deze HOW-TO te schrijven. Ik ben van baan veranderd en verhuisd. Sorry.

2.2 DEEL II: De eerste stappen met de modem.

De beheerders geven de voorkeur aan verschillende modems, maar bij het kopen van een modem zouden we de volgende richtlijnen aan moeten houden:

- Zie de Linmodem-HOWTO als je een USR WinModem hebt.
- De externe modem is het meest flexibel en je hebt meer ruimte in je computer.
- De interne modem met ISA-slot is beter dan die met PCI-slot (je kunt je PCI-slot voor iets anders gebruiken)
- Gebruik geen Plug&Play modem, zie de Plag-and-Play-HOWTO.

Wanneer we eenmaal een geschikt modem hebben, moeten we het instellen op ons systeem. We moeten controleren op welke com poort ons modem is aangesloten. Dan moeten we nog een symbolische link /dev/modem aanmaken naar deze hardware. Als de modem zich bijvoorbeeld bevindt op de tweede com-poort, tikken we in:

```
ln -s /dev/cua1 /dev/modem
```

We controleren dit nog even

```
lrwxrwxrwx 1 root uucp 9 Sep 19 19:10 /dev/modem -> /dev/cua1
```

Als de modem op een andere com-poort is aangesloten, denk dan aan het volgende:

```
/dev/cua0 is com1

/dev/cua1 is com2

/dev/cua2 is com3

/dev/cua3 is com4
```

Voor nieuwere kernels:

```
/dev/ttyS0 is com1
```

```
/dev/ttyS1 is com2
```

```
/dev/ttyS2 is com3
```

```
/dev/ttyS3 is com4
```

Nu controleren we onze configuratie met het programma minicom.

2.3 DEEL III Linux opbellen

De eerste stap die moet worden genomen om call-back onder Linux toegankelijk te maken is het instellen van de van toepassing zijnde parameters in de kernel. Dan controleren we of onze kernel het protocol ppp ondersteunt. Als ppp niet in je kernel is opgenomen of als module is gecompileerd, zul je de kernel moeten compileren en ppp toe moeten voegen. Meer informatie hierover is te vinden in de Kernel-HOWTO. In de kernelserie 2.4.x moet je de volgende opties markeren:

CONFIG_PPP=m # CONFIG_PPP_MULTILINK is not set CONFIG_PPP_ASYNC=m CONFIG_PPP_SYNC_TTY=m CONFIG_PPP_DEFLATE=m CONFIG_PPP_BSDCOMP=m

Na de compilatie moet je een aantal regels toevoegen aan /etc/modules.conf

alias /dev/ppp ppp_generic alias char-major-108 ppp_generic alias tty-ldisc-3 ppp_async alias tty-ldisc-14 ppp_synctty alias ppp-compress-21 bsd_comp alias ppp-compress-24 ppp_deflate alias ppp-compress-26 ppp_deflate

Je mag niet vergeten dat je de nieuwe pppd daemon nodig hebt voor 2.4.x (voor mij was het ppp-2.4.0, deze ppp bevatte een fout, je moet de laatste versie ophalen vanaf ftp.samba.org in /pub/ppp).

OK. We hebben een goede kernel. Nu moeten we de software nog instellen op ons systeem. Het call-back programma maakt onderdeel uit van mgetty-sendfax en ppp. Je zult het in je distributie aantreffen. Omdat het call-back systeem een dubbele verificatie heeft, maken we een gebruiker aan die ppp aan de kant van de server zal draaien. In /etc/passwd maak je een nieuwe gebruiker aan en moet je de shell wijzigen.

```
pppuser:klkIOM89mn65H:230:PPP Dialin:/home/pppuser:/etc/ppp/ppplogin
```

Ik wijzigde de bovenstaande regel voor kernel 2.4.x in de hierna volgende regel in /etc/passwd (Ik maak gebruik van shadow en het wachtwoord zie je dan niet)

```
pppuser:x:6778:44:PPP Dialin:/etc/ppp/:/usr/sbin/pppd
```

Ik maak geen gebruik van een speciaal script voor het starten van pppd, maar voer het direct uit wanneer ingelogd als gebruiker pppuser.

Wijzig dan het wachtwoord. We moeten informatie toevoegen over het wachtwoord in het bestand `/etc/ppp/pap-secrets` (meer in man pppd)

```
pppuser * wachtwoord_voor_pppuser *
```

In de 2.4.x kernels moet je in `/etc/ppp/pap-secrets` schrijven:

```
*                *       ""                    *
```

Deze gebruiker heeft geen gebruikelijke shell, maar een bestand `/etc/ppp/ppplogin`. We moeten het zelf aanmaken. Bijvoorbeeld `vi /etc/ppp/ppplogin` en we tikken in:

```
#!/bin/sh

exec /usr/sbin/pppd -detach 192.168.1.1:192.168.1.2
```

waar het adres 192.168.1.1 het adres is van de server met de modem en het adres 192.168.1.2 wat we hebben toegekend aan ons modem. We stellen uitvoerbare opties in voor dit bestand. Omdat we de ppp daemon gebruiken, moeten we de opties voor deze daemon instellen. We wijzigen het bestand `/etc/ppp/options`:

```
proxyarp

lock

crtscts

modem
```

Met een 2.4.x kernel schrijf je dit in `/etc/ppp/options`

```
-detach
asyncmap 0
modem
crtscts
proxyarp
lock
require-pap
refuse-chap
ms-dns 192.168.1.1
usepeerdns
```

De 3e optie van onder is erg belangrijk. Je gebruikt alleen de PAP-authenticatie en dus require-pap. Gebruik geen chap authenticatie dus refuse-chap. Je kunt ms-dns gebruiken, als je M$ Windows systeem clients hebt, kun je ze informatie toesturen over de DNS-server. Als je de IP van de DNS-server van de Linux/UNIX machine niet wilt versturen, gebruik je de optie usepeerdns, meer hierover in man pppd.

Proxyarp is het belangrijkst van bovenstaande opties, omdat je via de modem op de server het Internet op kunt. De resterende opties worden gebruikt om je modem te besturen. Je kunt alleen op de server werken als je de optie proxyarp verwijdert. Kijk in de PPP-HOWTO en man pppd voor meer informatie. We zullen nu ons modem in gaan stellen. Onze server moet gereed zijn na het starten een verbinding te ontvangen. We wijzigen het bestand `/etc/inittab` en we voegen er de modem op de 2e com-poort aan toe.

```
s1:2345:respawn:/sbin/mgetty ttyS1 -D /dev/ttyS1 vt100
```

of

```
s1:2345:respawn:/sbin/mgetty ttyS1 -s 115200 -D /dev/ttyS1
```

Voor een 1e com-poort ziet het er zo uit:

```
s0:2345:respawn:/sbin/mgetty ttyS1 -D /dev/ttyS1 vt100
```

of

```
s0:2345:respawn:/sbin/mgetty ttyS0 -s 115200 -D /dev/ttyS0
```

We voeren een `init q` uit. Als we geen informatie in logbestanden tot onze beschikking hebben over eventuele opgetreden fouten, gaan we naar de volgende stap. We gaan weer terug naar de directory `/etc/ppp` en maken een `options.ttyS1` (voor modem com1 `options.ttyS0`) aan.

```
IP_local: IP_remote
```

voor ons net zal dit zijn

```
192.168.1.1:192.168.1.2
```

We hebben reeds een boel werk verricht. Nu controleren we het bestand `/etc/mgetty+sendfax/login.config`. De belangrijkste regel is:

```
/AutoPPP/ - a_ppp /usr/sbin/pppd auth -chap +pap login detach 7 debug
```

Als je gebruik maakt van een 2.4.x kernel, moet je het volgende in dit bestand wegschrijven:

```
 /AutoPPP/ -     a_ppp   /usr/sbin/pppd file /etc/ppp/options
```

De resterende regels kunnen worden gemarkeerd met een voorafgaande `#`.

We hebben de ppp daemon setuid ingesteld, omdat pppuser pppd op moet starten en de interface werkend moet krijgen. Ik sta erop een mening van Bill Staehle in te voegen. "NOOT: Een aantal distributies denkt dat ze het beter weet dan jij, en verwijdert deze permissie wanneer hun 'admin' tool (yast in SuSE, linuxconf in vele anderen) wordt gebruikt. Lees de documentatie van de "toolöm te bekijken hoe je deze rommel stopt."

```
chmod u+s /usr/sbin/pppd
```

en het effect is:

```
-rwsr-xr-x 1 root root 106892 Jan 11 1999 /usr/sbin/pppd
```

Ik denk dat het een goed idee is het aan cron toe te voegen omdat ik een probleem constateerde, na een herstart van mijn server wijzigde de preferences van pppd. Dit is de normale situatie. Ik voeg een mening toe aan die van Bill Staehle "OPMERKING: Dit is wat ik bedoelde. Kijk onder Red Hat naar /usr/lib/linuxconf/redhat/perm/ppp (verwijder of hernoem het bestand)."

Onze server zal als router fungeren. We hebben IP forwarding geactiveerd en voegen de volgende regel toe aan het bestand `/etc/rc.d/rc.local` :

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

Je kunt als gebruiker van RedHat in `/etc/sysconfig/network` `FORWARD_IPV4=false` wijzigen in `FORWARD_IPV4=true`.

Ter controle bellen we naar Linux. We gebruiken hier scripts voor. Als we dit onder MS Windows doen, markeren we de opties `"call out a terminal after connection"`

. We loggen in als pppuser met het bijbehorende wachtwoord. Ik hoop dat alles goed gaat.

2.4 DEEL IV Linux belt ons

We kunnen reeds onze Linux bellen. Nu is het tijd dat Linux ons belt. Het is niet zo moeilijk. We hoeven slechts twee bestanden te wijzigen. We maken een bestand met de naam `/etc/mgetty+sendfax/callback.conf` aan het laten dit leeg.

Dan moeten we onze gebruikers vragen om hun telefoonnummer. Het is zover de nummers op te schrijven waarmee we eerder een verbinding maakten. Hiervoor wijzigen we `/etc/mgetty+sendfax/login.conf` en voegen de volgende regel toe:

```
call - - /usr/sbin/callback -S 123456
```

waar call een pseudo-gebruiker is, nodig om de verbinding te beginnen. De regel in `/etc/mgetty+sendfax/login.conf` zet het programma dat het gegeven nummer belt in beweging (in dit geval is dat 123456). Dezelfde procedures kunnen op andere gebruikers worden toegepast. Ik zal proberen uit te leggen hoe het werkt. Wanneer we een server bellen, vraagt het ons om verificatie. We loggen in als pseudo-user, in dit geval roept het aan. Het script op onze computer hangt de modem op. We wachten en de verbinding wordt verbroken. Het call-back programma begint te werken en belt ons terug. We maken onszelf weer bekend als pppuser en geven het bijbehorende wachtwoord op. We combineren de verbinding en interface ppp. Dat is alles. De configuratie van werkstations is zeer simpel. Wanneer je MS Windows hebt, moet je dial-up voor je nummer installeren. Bij de modemeigenschappen vinden we "propriety—>extended—> extra options"waar we intikken:

```
&c0s0=1
```

We sluiten het venster en bellen op. We loggen overeenkomstig de hierboven gegeven beschrijving in. Als we Linux willen gebruiken, moeten we verwijzen naar het script. Het is lastig slechts één goed script voor onder Linux te geven. Een goede configuratie van ppp op het systeem is van primair belang. (Je kunt het als eerste via de scripts als pppuser aanroepen). De scripts hieronder werden geschreven door A. Gozds. Ik raad je aan alles te catalogiseren. Het is alleen maar een suggestie & je hoeft de scripts hier niet op te starten. Gedetailleerde informatie betreffende het schrijven van scripts onder Linux is te vinden in de PPP-HOWTO. Het configuratiebestand van de daemon ppp (een voorbeeld voor een modem aangesloten op com2) **DEZE SCRIPTS WERKEN GOED ONDER LINUX RED HAT 6.x**

- /etc/ppp/options

  ```
  lock

  defaultroute

  noipdefault
  ```

```
modem

115200

crtscts

debug

passive

asyncmap 0
```

- /etc/ppp/pppcallback

```
TIMEOUT 5

ABORT 'ERROR'

ABORT 'BUSY'

ABORT 'NO ANSWER'

ABORT 'NO DIALTONE'

ABORT '\nVOICE\r'

ABORT '\nRINGING\r\n\r\nRINGING\r'

'' AT&FH0 'OK-+++\c-OK' 'AT&C0S0=1'

TIMEOUT 40

OK ATDT5376443 CONNECT ''

ogin:-ogin: ppp-pseudo-user

'\nNO CARRIER\r' ''

TIMEOUT 180

'\nRING\r' AT&C1A

CONNECT ''

TIMEOUT 20

ogin:-ogin: pppuser

sword:-sword password_for_ppuser
```

- /usr/bin/ppp-call

```
#!/bin/bash

teksta="Verbinding mislukt"

tekstb="Waarschijnlijk zal een verbinding worden gemaakt"

# /sbin/setserial /dev/ttyS1 spd_vhi

killall -INT pppd 2>/dev/null

rm -f /var/lock/LCK* /var/run/ppp*.pid

(/usr/sbin/pppd -detach call ppp_call &) || \

(echo $teksta; ls marsss >/dev/null; exit 1)

echo $tekstb

exit 0
```

- Je kunt ppp-call nu uitvoeren. :)

Als je M$ Windows hebt, kun je dit script voor de verbinding gebruiken. Ik heb het niet getest (ik gebruik een terminal), meer informatie kun je vragen aan Adrian Debkowski (*adrian@cr-media.pl*).

```
proc main

delay 1

waitfor "ogin:"

transmit "call^M"

waitfor "RING"

transmit "ATA^M"

waitfor "CONNECT"

waitfor "ogin:"

transmit "pppuser^M"

waitfor "word:"

transmit "ppp^M"

endproc
```

2.5 DEEL V Samenvatting

De call-back configuratie is niet moeilijk. Het belangrijkste is een juiste werking van de ppp-server onder Linux. Ik weet geen betere manier om een toegangsserver op te zetten. De hierboven gepresenteerde configuratie is een resultaat van talrijke pogingen en het kan op verschilende wijzen worden uitgevoerd. Daarom raad ik je de documenten man pppd, de NET-HOWTO, de PPP-HOWTO, de ISP-Setup-RedHat-HOWTO, de Modem-HOWTO aan. Met speciale dank aan Bill Staehle.