Le service de rappel, mini-HOWTO

Pawel Skonecki, *stona@fizyka.umcs.lublin.pl*
Traduction française : Sylvain Amrani *traduc@amrani.com.fr*
v1.1a, Juin 2000

Ce document décrit la manière de mettre en place un service de rappel en utilisant un système Linux et un modem. Je voudrais remercier Anna pour sa patience.

Contents

1 Introduction

1.1 Réactions

Je serai attentif à toutes les réactions concernant ce document. J'ai essayé de rassembler les informations d'une façon aussi complète que possible. Dites-moi si vous trouvez des erreurs. Je serai reconnaissant envers les personnes qui m'enverront des suggestions ou des corrections. Leurs contributions amélioreront ce document. Je ne serai pas ennuyé de répondre à vos questions, mais j'aimerais que vous lisiez ce document dans son intégralité auparavant.

1.2 Publication

Ce document peut être publié sous les conditions du Linux Documentation Project (Projet de Documentation Linux). Prenez contact avec l'auteur si vous ne parvenez pas à vous procurer cette licence. Il s'agit d'un "document libre".

2 Procédure

2.1 Le réseau à la maison ?

La plupart d'entre nous utilisent l'Internet au travail. En revanche, nous avons souvent besoin du réseau à la maison, ou à l'extérieur du bureau. Il est possible que le travail effectué à la maison revienne moins cher que celui fait au bureau. Je pense que la meilleure solution est d'installer un logiciel de rappel sur le serveur Linux. Un service de rappel permet de faire rappeler le serveur pour que la connexion soit à la charge de la société. Je vais essayer de présenter la manière dont cela fonctionne. Une personne, à qui l'on a donné le droit, appelle. L'état du modem du serveur Linux est modifié une première fois. Ensuite, du côté de l'utilisateur, le modem ayant raccroché est ré-activé. Dans le même temps, le serveur Linux appelle l'utilisateur. L'utilisateur s'identifie à nouveau. Nous obtenons alors une connexion à la charge du serveur. L'utilisateur ne paie que l'initialisation de la connexion. La double vérification et des options supplémentaires dans la configuration du logiciel de rappel empêchent que des indésirables ne chargent nos factures. Nous pouvons restreindre l'accès de la connexion au réseau d'entreprise seulement ou à l'Internet. Le service de rappel est très souple. Dans la suite, je vais essayer de vous présenter la configuration d'un service de rappel sur un serveur Linux et de vous montrer comment configurer votre ordinateur pour reprendre la connexion.

2.2 Premiers pas avec les modems.

Les administrateurs procèdent à des choix différents concernant les modems. Mais avant d'en acheter un, vous devrez vous rappeler certaines règles :

- N'achetez pas un Win-Modem (modem spécifique pour Windows). Ils ne fonctionnent pas sous Linux.
- Les modems externes travaillent plus vite que les modems internes.
- Un modem interne sur carte ISA est préférable à un modem sur carte PCI (vous pourrez utiliser votre port PCI pour autre chose).
- N'utilisez pas un modem Plug&Play. Si vous en avez un, désactivez l'option Plug&Play et configurez le modem pour un port série libre (Cf. le Plug&Play-HOWTO).

Maintenant que nous avons le modem adéquat, nous devons le configurer pour notre système. Nous devons vérifier à quel port série il est raccordé. Ensuite, nous devons créer un lien symbolique entre ce port matériel et /dev/modem. Par exemple, si notre modem est connecté au deuxième port série, nous écrivons :

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

Vérifions :

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

Si notre modem est connecté à un port série différent, nous devons nous rappeler que :

```
/dev/cua0 correspond à com1
```

```
/dev/cua1 correspond à com2
```

```
/dev/cua2 correspond à com3
```

```
/dev/cua3 correspond à com4
```

Et pour les nouveaux noyaux :

`/dev/ttyS0 correspond à com1`

`/dev/ttyS1 correspond à com2`

`/dev/ttyS2 correspond à com3`

`/dev/ttyS3 correspond à com4`

Maintenant nous pouvons vérifier notre configuration avec le programme minicom.

2.3 Appeler le serveur Linux

La première chose à faire pour rendre accessible notre service de rappel sous Linux est de configurer les paramètres appropriés dans le noyau. Vérifions que notre noyau inclut le protocole PPP. Si vous n'avez pas le protocole PPP dans votre noyau ou dans un module, vous devrez recompiler votre noyau et y ajouter PPP. Vous trouverez plus d'informations dans le Kernel-HOWTO. Bien. Nous avons maintenant un noyau correct. Nous devons à présent configurer le logiciel pour notre système. Le programme de rappel fait partie de mgetty-sendfax et de PPP. Vous le trouverez tous dans votre distribution. En raison du système de rappel et de la double vérification, nous devons créer un utilisateur qui exécutera PPP côté serveur :

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

Changez son mot de passe. Puis nous devons ajouter son mot de passe dans le fichier `/etc/ppp/pap-secrets` (consultez la page de manuel de pppd) :

```
pppuser * mot_de_passe_pour_pppuser *
```

Cet utilisateur ne dispose pas d'un shell classique, mais d'un fichier `/etc/ppp/ppplogin`. Nous devons le faire nous-même. Par exemple, tapez `vi /etc/ppp/ppplogin` et entrez :

```
#!/bin/sh

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

où 192.168.1.1 est l'adresse de notre serveur et 192.168.1.2 est l'adresse que nous donnons à notre modem. Nous donnons les droits d'exécution à ce fichier. Parce que nous utiliserons le démon PPP, nous devons configurer ses options en éditant le fichier `/etc/ppp/options` :

```
netmask 255.255.255.0

proxyarp

lock

crtscts

modem
```

Proxyarp est l'option la plus importante parce qu'elle permet d'aller sur l'Internet par le modem du serveur. Les autres options sont utilisées pour contrôler votre modem. Votre utilisateur ne pourra travailler que dans le réseau local si vous retirez l'option proxyarp. Référez-vous au PPP-HOWTO et à la page de manuel de pppd pour plus d'informations. Nous allons maintenant configurer notre modem. Notre serveur doit être prêt à recevoir une connexion après son démarrage. Nous éditons le fichier `/etc/inittab` et ajoutons, pour un modem sur le deuxième port série :

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

Avec un modem sur le premier port série, ce serait :

```
s0:2345:respawn:/sbin/mgetty ttyS0 -D /dev/ttyS0 vt100
```

Entrons maintenant `init q`. Si nous n'avons pas trace d'erreur dans les logs, nous pouvons poursuivre. Revenons dans le répertoire `/etc/ppp` et créons `options.ttyS1` (pour un modem sur com1 `options.ttyS0`)

```
IP_locale: IP_distante
```

pour notre réseau, ce sera :

```
192.168.1.1:192.168.1.2
```

Nous avons fait déjà beaucoup de travail. Maintenant, vérifions le fichier `/etc/mgetty+sendfax/login.config`. La ligne la plus importante est :

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

Les autres lignes peuvent être commentées avec des `#`.

Nous devons donner les droits de super-utilisateur à l'exécution (suid) au démon ppp, parce que l'utilisateur pppuser doit lancer pppd et rendre l'interface active.

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

ce qui donne :

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

Je pense que c'est une bonne idée d'ajouter ça aux tâches programmées de cron, parce que j'ai des problèmes de changement des préférences de pppd après un redémarrage de mon serveur. Notre serveur fonctionne comme un routeur. Nous devons activer l'IP forwarding en ajoutant cette ligne au fichier `/etc/rc.d/rc.local` :

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

Si vous avez une distribution RedHat, vous pouvez faire cela avec `/etc/sysconfig/network` en changeant `FORWARD_IPV4=false` en `FORWARD_IPV4=true`.

Pour la vérification quand nous appelons le serveur Linux, nous utilisons un script. Si nous utilisons MS Windows, nous devons choisir l'option `"Appeler terminal après la connexion"`. Nous nous connectons en tant qu'utilisateur pppuser, avec son mot de passe. J'espère que tout est OK.

2.4 Linux nous appelle.

Nous pouvons d'ores et déjà appeler notre serveur Linux. Maintenant, c'est à Linux de nous appeler. Ce n'est pas très compliqué. Nous devons seulement éditer deux fichiers. Nous créons le fichier `/etc/mgetty+sendfax/callback.conf` que nous laissons vide.

Ensuite, nous devons connaître le numéro de téléphone de nos utilisateurs. Nous devons inscrire les numéros qui pourront nous contacter. Nous devons pour cela éditer le fichier `/etc/mgetty+sendfax/login.conf` et ajouter la ligne :

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

où call est un pseudo-utilisateur nécessaire pour lancer la connexion. La ligne dans le fichier `/etc/mgetty+sendfax/login.conf` déclenche le programme devant appeler le numéro fourni (dans cet exemple c'est 123456). La même procédure est à suivre pour les autres utilisateurs. Expliquons comment cela fonctionne. Nous appelons le serveur. Il nous demande de nous authentifier. Si nous nous identifions comme le pseudo-utilisateur, le serveur rappelle. Le script de notre ordinateur fait raccrocher le modem. Nous attendons et la connexion est coupée. Le programme de rappel est alors lancé et nous rappelle. Nous nous authentifions comme l'utilisateur pppuser avec son mot de passe. La connexion est associée à l'interface ppp, et c'est tout. La configuration des stations de travail est très simple. Si vous avez MS Windows, vous devez installer une connexion à distance pour votre numéro. Dans les propriétés du modem, on trouve "Propriétés—>Connexion—>Avancé...—>Paramètres supplémentaires", où nous mettons :

```
&c0s0=1
```

Fermons la fenêtre et appelons. Nous nous authentifions conformément à ce qui a été dit précédemment. Si nous utilisons Linux, nous devons avoir recours aux scripts. Il est difficile de donner un seul bon script pour Linux. Une bonne configuration de PPP est de première importance pour le système. (Vous pouvez tout d'abord appeler sous un profil pppuser avec un script). Les scripts ci-dessous ont été écrits par A. Gozdz. Je suggère de tout cataloguer. C'est juste une suggestion, vous n'êtes pas obligé de démarrer par des scripts à cet endroit. Des informations détaillées concernant les scripts avec Linux peuvent être trouvées dans le PPP-HOWTO.

CES SCRIPTS FONCTIONNENT CORRECTEMENT AVEC UNE SLACKWARE

Fichier de configuration pour le démon PPP (exemple pour un modem relié au deuxième port série) `/etc/options`

```
lock

defaultroute

noipdefault

modem

/dev/cua1

33600

crtscts
```

```
debug

passive

asyncmap 0
```

et les scripts spécifiques :

- le premier appelé /etc/ppp/ppp-call

  ```
  #!/bin/bash

  teksta="La connexion a échoué"

  tekstb="Vous allez probablement être connecté"

  # /sbin/setserial /dev/cua1 spd_vhi

  killall -INT pppd 2>/dev/null

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

  (/usr/sbin/pppd -detach /dev/ttyS1 115200 \

  connect "/usr/sbin/chat -v -f /etc/ppp/pppcallback" &) || \

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

  echo $tekstb

  exit 0
  ```

- le second fichier appelé /etc/ppp/pppcallback

  ```
  TIMEOUT 60

  ABORT ’ERROR’

  ABORT ’BUSY’

  ABORT ’NO ANSWER’

  ABORT ’NO DIALTONE’

  ABORT ’\nVOICE\r’

  ABORT ’\nRINGING\r\n\r\nRINGING\r’

  ’’ AT&FH0 <p>’OK-+++\c-OK’ ’AT&C0S0=1’
  ```

```
TIMEOUT 75

OK ATDT123456

CONNECT ''

ogin:-ogin: ppp_pseudouser

'\nNO CARRIER\r' ''

TIMEOUT 180

'\nRING\r' AT&C1A

CONNECT ''

TIMEOUT 20

ogin:-ogin: pppuser

sword:-sword mot_de_passe_de_pppuser
```

- Vous pouvez maintenant appeler par PPP :)

CES SCRIPTS FONCTIONNENT CORRECTEMENT AVEC UNE RED HAT 6.x

- /etc/ppp/options

```
lock

defaultroute

noipdefault

modem

33600

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 mot_de_passe_de_pppuser
```

- /usr/bin/ppp-call

```
#!/bin/bash

teksta="La connexion a échoué"

tekstb="Vous allez probablement être connecté"

# /sbin/setserial /dev/cua1 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
```

- Vous pouvez maintenant appeler par PPP :)

Si vous avez M$ Windows, vous pouvez utiliser ce script pour vous connecter. Je ne l'ai pas testé (j'utilise un terminal). Vous pouvez en savoir plus auprès d'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 Résumé

La configuration d'un service de rappel n'est pas compliquée. La chose la plus importante est une configuration correcte du serveur PPP sur Linux. Je ne connais pas de meilleur moyen pour configurer un serveur d'accès. La configuration présentée ci-dessus est le résultat de nombreux essais, mais cela peut être fait d'une autre façon. C'est pourquoi je suggère de lire tous les documents concernant ce sujet, la page de manuel de pppd, les NET4-HOWTO et PPP-HOWTO.