Dynamic IP Hacks Mini-HowTo

Michael Driscoll *fenris@lightspeed.net*
Traduction française Mathieu Arnold *arn_mat@club-internet.fr* V 2.1.3, 25 Mai 1997

Comment s'en sortir avec une IP dynamique.

Table des matières

1 Copyright et Définitions...

par Michael Driscoll *fenris@lightspeed.net*

> [15 Janvier 1998 - Ce Mini-HowTo n'est plus maintenu par son auteur. Si vous êtes intéressés par la maintenance de ce Mini-HowTo, prenez contact avec moi à *gregh@sunsite.unc.edu.*]

Avec les Hacks de :

- Michael Driscoll *fenris@lightspeed.net*
- Ryan R. Klems *rklems@primenet.com*
- Matthew Driver *mdriver@cfmeu.asn.au*
- Matthew Nuckolls *mnuck@umr.edu*
- Justin Cragin *beyond@lightspeed.net*
- Brad Baker *bpb@mlb.cca.rockwell.com*
- Justin (Gus) Hurwitz *ghurwitz@dyndns.com*
- Jeremy D. Impson *jdimpson@camelot.syr.edu*
- Steve Przepiora *gearhead@dreamscape.com*
- Artur Skawina *skawina@usa.net*
- Dale Jolliff *taftbbs@e-tex.com*

1.1 Ou trouver la dernière version de ce Mini-HowTo

La toute dernière version peut être trouvée à *frob.base.org/howto.txt.*

L'adresse ou vous pourrez toujours trouver ce document est *sunsite.unc.edu/pub/Linux/docs/howto/mini/Dynamic-IP-Hacks*, toutefois, vous devriez utiliser un site miroir du LDP si vous en connaissez un.

1.2 Commentaires

Tous les commentaires à propos de ce document devraient être adressés à Michael Driscoll *fenris@lightspeed.net.*

1.3 Pourquoi ce Mini-HowTo ?

Le problème est que Linux ne s'accommode pas aussi bien qu'il le pourrait de l'adressage dynamique des IP. Cet adressage dynamique pourrait être l'équivalent Unix au fait de se réveiller tous les matins et constater que votre adresse postale a changé et que toutes vos cartes de visites sont dépassées. Malheureusement, les FAI[1] tendent de plus en plus à utiliser ce type d'adressage de nos jours, et tous ceux qui ne peuvent pas se permettre de dépenser un peut plus d'argent pour obtenir une IP fixe sont coincées avec ça. Le but de ce document est donc de rendre votre Linux aussi à l'aise avec l'adressage dynamique d'IP que possible, il vous rendra plus heureux et sera plus pratique à utiliser.

[1]NdT : Fournisseur d'Accès Internet

1.4 Qu'est ce que l'adressage dynamique des IP, et pourquoi ai-je à m'en préoccuper ?

Une adresse IP est une suite de quatre chiffres, chacun ayant une valeur comprise entre 0 et 255, et ces 4 chiffres sont séparés par un point. Par exemple : `198.41.0.8`. Chaque ordinateur sur Internet à une unique adresse IP. Les adresses que vous utilisez, telles `sunsite.unc.edu` et `bac2.lightspeed.net`, sont des alias qui sont traduits en adresses IP par un serveur de noms, le DNS, chaque fois que vous y tentez d'y accéder.

L'adressage dynamique des IP est quelque-chose utilisé par les FAI pour réduire le nombre d'adresses dont ils ont besoin. Chaque fois que vous vous connectez à votre FAI, ils vous donnent juste la dernière adresse IP dans la liste de celles disponibles. Ainsi, ils n'ont pas besoin d'une adresse par client, mais seulement pour les clients qui sont connectées à un instant donné. Habituellement, les FAI possèdent une adresse pour dix clients, bien que ce ratio varie.

Pourquoi font-ils ça ? Et bien, la façon la plus simple de l'expliquer est de dire qu'il n'y a pas suffisamment d'adresses IP avec le protocole actuel. Une façon plus complexe est de dire que nous avons suffisamment d'adresses, mais qu'il y en a de plus en plus d'utilisées, les tables de routage utilisées par les routeurs des dorsales commencent à ressembler à des disque MS-DOS vieux de 10 ans et qui n'a jamais été défragmenté :-) En utilisant tant d'adresses, il y a une perte d'efficacité dans les recherches. Le système actuel est censé exploser entre 2000 et 2010. Heureusement, nous utiliserons tous le protocole IPv6 à ce moment là, et nous aurons tous notre propre IP ce qui rendra ce Mini-HowTo inutile :-)

1.5 Pouvez-vous m'aider a configurer {pppd,telnetd,sendmail...} ?

Non, Débrouillez vous tout seul. Ça forme le caractère :-)

En fait, il me serait plutôt inutile de réinventer la roue en vous aidant à les configurer, puisque beaucoup de documents existent déjà pour vous permettre de vous en sortir. Commencez par essayer les suivants :

- La page man de `pppd`
- La page man de `chat`
- La page man de `in.telnetd`
- La page man de `inetd`
- le PPP-HOWTO
- le NET-2-HOWTO
  (Disponibles sur des sites tels : *sunsite.unc.edu/pub/Linux/docs/howto*)
- le NAG
  (Network Administrator's Guide par Olaf Kirch, disponible sur les sites du LDP, comme : *sunsite.unc.edu/pub/Linux/docs/LDP*)

1.6 Hé, et le SLIP ?

Pour vous dire la vérité, si je n'écris rien à ce sujet, c'est parce que je ne sais rien à ce sujet.

Heureusement, cela ne devrait pas être un trop gros problème, puisque PPP semble être le standard actuel, spécialement en ce qui concerne l'adressage IP.

Je suis désolé pour vous si vous faites partie des 0.3% de ceux qui ont leur adressage IP avec SLIP, mais je pense que vous pourrez peut-être utiliser ce HowTo et débrouillez-vous pour que ça marche (si vous le faites, lisez l'appendice A pour voir votre hack ajoute à ce HowTo !)

1.7 Copyright

1.8 Avertissement

Cela va sans dire, mais je le dis quand même au cas où.

Je ne suis pas sur que tout cela est fait de la meilleure façon, et si quelque chose que j'ai fait est trop bizarre, dites moi comment le rendre plus clair. Il se peut qu'il y ait des risques de sécurité dans tout ça, mais je ne les connais pas. Tout ceci marche bien sur ma machine, et je ne pense pas que cela risque de trop casser la votre :-) Par conséquent, vous devriez quand même faire attention et être sur de bien comprendre au moins un petit peu ce que vous faites, ainsi, vous pourrez le réparer si ça ne marche plus.

Comme son nom l'indique, ce document parle de **hacks**. Les hacks, par définitions sont des façons plutôt bizarres de configurer un système, bien que parfois, ils soient la seule façon. En plusieurs endroits, ces hacks pourraient être faits de façons plus élégantes, mais au coût de la simplicité. Je ne l'ai pas fait car il faut que la personne qui met en place les hacks les comprenne, pour qu'elle puisse les refaire en fonction de ses besoins.

De plus, en lisant ce HowTo, gardez à l'esprit cette phrase venant du DNS-HOWTO de Nicolai Langfeldt *janl@math.uio.nl* :

Dans ce document, j'énonce deux ou trois choses qui ne sont pas tout à fait vraies (mais qui le sont toujours au moins à moitié). Si je le fais, c'est toujours dans le but de rendre les choses plus simples. Tout marchera (probablement ;-) très bien si vous croyez ce que je vous dis.

2 Introduction aux hacks 1 au 3

par Michael Driscoll *fenris@lightspeed.net*

Les trois premiers hacks de ce HowTo servent à maintenir une connection PPP constante en redémarrant pppd quand la liaison est perdue. J'ai pensé au deuxième de ces hacks quand j'ai démarré Linux, mais depuis, j'ai changé pour le troisième qui est quand même plus efficace (et malheureusement plus difficile à mettre en place). Le premier hack n'a pas été fait par moi mais par Steve Przepiora *gearhead@dreamscape.com*, et qui est en fait meilleur et plus efficace que le second hack (celui que j'ai inventé), et est aussi facile à mettre en oeuvre...

Ces trois hacks fonctionnent très bien avec le hack 4, qui va mettre automatiquement votre IP dans un endroit accessible (votre compte e-mail sur le serveur POP3 de votre FAI) à chaque fois que votre connection est de nouveau établie. Avec ces deux hacks marchant ensemble (reconnection automatique + publication de l'IP par e-mail) vous pourrez facilement trouver l'IP dynamique de votre machine n'importe ou sur le réseau et être capable de faire du telnet, ftp, etc. vers elle.

Les méthodes 1 et 2 sont celles qui sont faciles à implémenter, puisqu'elles sont fournies avec une recette et tout ce que vous avez à faire c'est de suivre les instructions et remplir les blancs. La méthode 3 n'est pas fournie avec une recette car elle utilise les `runlevels` et l'`init` choses qui varient beaucoup d'une distribution à l'autre, et peuvent laisser votre machine dans un état bizarre s'ils sont mal configurés.

Alors, quelle méthode utiliser ? Je suggère la méthode 1, elle est facile et plutôt efficace, ou si vous avez un peu plus de connaissances, jetez un coup d'oeil à la méthode 3 pour voir si vous pouvez réussir à implémenter cela, puisque c'est la plus efficace. La méthode 2 est là parce que c'est la première que j'ai écrite, et parce que c'est la seule de ce howto que j'ai testé (les autres sont de récentes additions et devraient rapidement remplacer la 2 de toute façon).

Si aucune de ces méthodes ne vous plaît, écrivez la votre, et envoyez-la-moi, je l'inclurais probablement si c'est utilisable par d'autres.

Bon, maintenant, on passe aux choses sérieuses :

3 Hack 1 : Garder votre système connecté et accessible 24h/24 et 7j/7 (méthode 1)

par Steve Przepiora *gearhead@dreamscape.com*

3.1 Ce dont vous avez besoin

Une configuration PPP qui marche.

3.2 ip-up, ip-down, et d'autres scripts assortis

Bon, alors voici comment je l'ai fait :

1. Lancez `vi`[2] et créez un fichier appelé `/etc/ppp/ppp-check.dat`. N'y mettez rien, sauvez le juste.
2. Mettez au début du fichier `/usr/sbin/ppp-on` :
   `echo on > /etc/ppp/ppp-check.dat`
   Ceci mettra le mot 'on' dans `/etc/ppp/ppp-check.dat`
3. Mettez ceci dans `/usr/sbin/ppp-off` :
   `echo off > /etc/ppp/ppp-check.dat`
4. Changez le nom de votre `/etc/ppp/ip-down` en `/etc/ppp/ppp-off-final`
5. Puis sauvez le script suivant dans `/etc/ppp/ip-down` :

```
#!/bin/sh

export LOGGING="yes" ## Changez ça si vous ne voulez pas l'enregistrer.
export LOG_STRING=""
export PPP_CHECK=""

## Configuration de base
PPP_CHECK=`cat /etc/ppp/ppp-check.dat`
```

[2] NdT : l'utilisation de `vi` me paraît inutile, un "`touch /etc/ppp/ppp-check.dat`"étant suffisant

```
## PPP a été déconnecté -> on l'enregistre
if [ "$LOGGING" = "yes" ]
 then
   logger -i -p local0.notice -t ppp-check  PPP- Disconnected
fi

## Maintenant, on regarde comment le script a été appelé
if  [ "$PPP_CHECK" = "off" ]
  then
   if [ "$LOGGING" = "yes" ]
     then
     logger -i -p local0.notice -t ppp-check  PPP- Legal disconnect
     logger -i -p local0.notice -t ppp-check  PPP- Turning off NET \
subsystem
     /etc/ppp/ppp-off-final
   fi
   exit
fi

## Si on a été déconnecté -> on recompose le numéro
if [ "$LOGGING" = "yes" ]
  then
   logger -i -p local0.notice -t ppp-check  PPP- Illegal disconnect
   logger -i -p local0.notice -t ppp-check  PPP- Attempting to reconnect
fi
/etc/ppp/ppp-off-final
while test -e /var/run/ppp0.pid && test -e /var/lock/LCK..ttyS3
 do
  sleep 1
done
  /usr/sbin/ppp-on
```

Les étapes 2 et 3 permettent au script de savoir comment la connection a été rompue

4 Hack 2 : Garder votre système connecté et accessible 24h/24 et 7j/7 (méthode 2)

par Michael Driscoll *fenris@lightspeed.net*

Tout d'abord lisez les notes à propos de ce hack dans le chapitre 2.

4.1 Ce dont vous avez besoin

Pas grand chose. Une configuration PPP qui marche. Un `cron` qui marche.

4.2 Mise en place du script `keepalive.sh`

4.2.1 En utilisant `crontab`

(note : Cette section sera probablement mieux avec `diald`, voir la section suivante)

`cron` est un daemon qui lance des programmes à intervalles réguliers. Vous pouvez regarder votre crontab en exécutant "`crontab -e`". Nous utiliserons crontab pour lancer le programme shell qui gardera la connection PPP.

Lancez "`crontab -e`" en tant que root et ajoutez la ligne suivante :

```
*/2 * * * * /etc/ppp/keepalive.sh
```

Ce que cela fait est de lancer le script `/etc/ppp/keepalive.sh` toutes les deux minutes. (Le # met l'entrée en commentaire pour qu'elle ne soit pas exécutée tant que le script n'est pas prêt).

Puis mettez le script suivant appelé `keepalive.sh` dans le répertoire `/etc/ppp` :

```
#!/bin/sh
if [ -f /var/run/ppp0.pid ] ; then
 ping -c8 -l3 <your nameserver> 2>&1 | grep "0 packets" > /dev/null && \
 { /usr/sbin/ppp-off > /dev/null 2>&1 ; sleep 2 ; /usr/sbin/pppd }
else
 /usr/sbin/pppd
fi
```

Maintenant tapez `chmod 700 /etc/ppp/keepalive.sh` en tant que root pour qu'il soit exécutable.

Conseil 1 :
Vérifiez les chemins ! Pour `pppd` et `ppp-off` insérez ce que vous utilisez pour lancer et arrêter les connections.

Conseil 2 :
Utilisez le serveur de DNS de votre FAI car s'il est en panne, votre connection à Internet est hors service de toute façon :-)

Conseil 3 :
Utilisez l'adresse IP numérique de votre serveur de DNS, sinon, `ping` renvoie un message diffèrent et le `grep` ne marchera pas.

Maintenant, quand vous voulez rester connecté, vous avez juste à lancer `crontab -e` et enlever le "#" pour décommenter l'entrée... et quand vous voulez rester déconnecter, relancez `crontab -e` et remettez le "#", puis arrêtez votre connection avec `ppp-off` ou ce que vous utilisez.

4.2.2 Utilisation `diald` à la place du cron

par Divya Mahajan *vmahajan@giasdl01.vsnl.net.in*

L'entrée crontab que nous venons de faire peut aussi être faite (et probablement plus élégamment) par d'autres programmes, tel `diald`. Cette section explique comment utiliser `diald` pour cette utilisation, vous devriez peut être décider d'utiliser cette manière. J'ai laissé la section avec le crontab pour référence et parce qu'il ne nécessite aucun package, mais vous voudrez peut-être utiliser diald à la place.

1. Récupérer la dernière version de diald (cela ne devrais pas être loin de *sunsite.unc.edu/pub/Linux/system/Network/serial/*)

2. Compilez le tout. (Vous trouverez peut être une version compilée de `diald` sur un site de la Slackware)
3. Après avoir installé `diald`, vous aurez à modifier `/etc/diald.conf`, ajoutez les lignes suivantes à la fin du fichier :

```
> restrict 06:00 19:00 * * *
> up
   (Ceci forcera à rester connecté entre 6 heures et 19 heures tous les jours.
    Si vous voulez rester 24h/24 et 7j/7, supprimez le restrict.)
> device /dev/modem
   (utilisez le bon périphérique...)
> dynamic
> reroute
> connect-timeout 120
   (Modifiez cela si votre composeur prends plus ou moins de temps pour se
    connecter à votre FAI.)
> redial-timeout 10
   (Intervalle entre 2 recomposition du numéro)
> defaultroute
> accounting-log /var/adm/diald.log
   (Garde une trace de la durée de vos connections)
>connect /path_to_myscript
   (Quand vous utilisez un script pour vous connecter mettez le sur la ligne
    ci dessus. Quand diald appelle ce script, les entrées et sorties standard
    sont redirigées vers le périphérique (le modem) donc, vérifiez que le
    script n'imprime pas de bêtises. J'utilise personnellement "/usr/sbin/dip
    mydipfile.dip >> /var/adm/dip.log 2> /var/adm/dip.err" qui enregistre les
    tentatives de connections. Au départ, vous exécuterez peut-être dip avec
    l'option "-v" pour mettre au point le fichier dip, par exemple : #dip -v
    mydipfile.dip. N'oubliez pas de mettre la ligne "mode ppp" après que vous
    soyez connectés et que vous ayez lancé les services PPP de votre FAI
    (utilisez dip-3.3.7n-uri). D'autre part, enregistrez tous les paramètres
    PPP dans /etc/ppp/options plutôt que de faire confiance à la ligne de
    commande. Dès que votre fichier dip est au point est prêt, il est temps de
    l'utiliser. Lancez diald /usr/sbin/diald, si tout s'est bien passé, il
    devrait commencer à se connecter à votre FAI. Quand la liaison PPP est
    close à cause d'un problème de modem, diald réessayera automatiquement.
    Dès que vous pensez que c'est au point, ajoutez /usr/sbin/diald dans votre
    /etc/rc.d/rc.local)
```

Ça y est, vous avez une connection permanente.

`diald` est aussi très bon pour bien d'autres choses, allez jeter un coup d'oeil sur la page de diald *www.dna.lth.se/˜erics/diald.html* pour quelques exemples.

5 Hack 3 : Garder votre système connecté et accessible 24h/24 et 7j/7 (méthode 2)

par Michael Driscoll *fenris@lightspeed.net*

Tout d'abord, allez voir les notes à propos de ce hack au chapitre 2.

Voici la troisième méthode pour rester connecté, c'est aussi celle que j'utilise. Je ne conseille à personne de l'utiliser à moins de bien en comprendre le fonctionnement. S'il y a des éléments que vous ne comprenez pas, je vous suggère d'aller lire :

- `man init`
- `man inittab`
- Essential System Administration, par AEleen Frisch
  (Pas vraiment nécessaire, mais ça aide à comprendre des trucs, comme les runlevels. Je pense que n'importe quel texte parlant des runlevels sera suffisant.)

5.1 Ce dont vous avez besoin

Une configuration PPP qui marche. Un `init` qui marche (Je pense que vous l'avez puisque c'est le programme qui lance tous les process sous Unix).

5.2 Jouons avec les runlevels

Voici comment j'ai mis ça en marche. Tout d'abord, j'ai change mon inittab et mes scripts rc, chose que je n'avais jamais fait depuis l'installation de ma Slackware 3.0, donc, ils ressemblaient à ça :

```
0) Halt
1) Un setup mono-utilisateur pas franchement opérationnel
2) Vide
3) Vide
4) xdm (X)
5) Multi-utilisateur normal (le runlevel par défaut)
6) Reboot
```

J'ai nettoyé ça et je les ai refait, et ils ressemblaient à ça après :

```
0) Halt
1) Mode mono-utilisateur
2) Multi-utilisateur normal  # Je vais expliquer ces deux la plus tard
3) Multi-utilisateur normal  (le runlevel par défaut)
4) Vide
5) Vide
6) Reboot
```

Puis, j'ai mis la ligne suivante dans `/etc/inittab` :

```
pu:3:respawn:/usr/sbin/pppd -detach
```

Ce que cela fait, c'est de garder `pppd` en exécution quand mon système est dans le runlevel 3, et donc, me reconnecter à chaque déconnection, et tuer ma connection quand je passe au runlevel 2. Quand je veux ma connection, je repasse dans le runlevel 3, et c'est reparti.

Le `-detach` sur la ligne respawn empêche `pppd` de se mettre en tache de fond, et donc, faire croire à init qu'il est mort (sinon, `init` va redémarrer `pppd` une douzaine de fois etc...).

Pour ceux qui sont intéressés par ma configuration exacte, j'ai fait un `tar` de mon `/etc/inittab` et de `/etc/rc.d/*` et je les ai mis sur mon site à *frob.base.org/rc.tar.gz*.

6 Hack 4 : Envoi de l'IP par e-mail

par Michael Driscoll *fenris@lightspeed.net*

Ce hack marche très bien avec les scripts précédents, car il envoie automatiquement votre IP quand elle change, donc, vous pouvez facilement trouver la nouvelle adresse de votre machine depuis n'importe ou sur le réseau après une reconnection.

Les autres qui font la même chose (envoi automatique de l'IP) en utilisant d'autres moyens sont les 6, 8, 9, 10, 11, 12, et 14 (presque la moitié, si on y pense :-).

6.1 Ce dont vous avez besoin

Juste une configuration PPP qui marche, juste ça. Ce hack marche très bien avec les hacks précédemment décris, car il vous permet de retrouver facilement votre machine après qu'elle se soit reconnecté et que son adresse ai changé.

6.2 Mise en place de vos `ip-up` and `ip-down`

Ce hack rends votre nouvelle IP accessible depuis à peu près n'importe quelle machine sur Internet. Pour réussir cet exploit :-) on va utiliser deux scripts `/etc/ppp/ip-up` et `/etc/ppp/ip-down`, qui sont automatiquement appelés par `pppd` quand vous vous connectez/déconnectez.

Créez un fichier (s'il n'existe pas déjà) en tant que root appelé `/etc/ppp/ip-up` avec dedans :

```
#!/bin/sh
# $4 Est notre nouvelle IP passée par pppd
# /var/run/add gardera notre adresse
echo $4 > /var/run/add
# Envoie notre adresse au serveur mail de notre FAI. avec comme sujet :
# "Nouvelle adresse  IP"
mail -s "Nouvelle adresse  IP" votre_nom@votre_adresse < /var/run/add
```

Puis créez `/etc/ppp/ip-down` et mettez-y :

```
#!/bin/sh
rm /var/run/add
```

Puis faites un `"chmod 700 /etc/ppp/ip-up /etc/ppp/ip-down"` pour les rendre exécutables.

Voilà[3] ! Maintenant, à chaque fois que `pppd` démarre, votre adresse sera envoyée au serveur de mail de votre FAI. On le récupérera dans la section suivante.

6.3 Comment récupérer votre nouvelle adresse IP depuis un autre ordinateur ?

Et bien, cela va demander un peu de réflexion de votre part. Je ne puis vraiment pas vous aider à vous en sortir, car cela dépends depuis quel type de machine vous de la récupérer. Depuis Windows, vous pouvez essayer de configurer un programme de mail pour récupérer les mails depuis le serveur de votre FAI, et si vous êtes sous Unix, allez voir s'il n'y aurais pas un programme appelé `"popclient"`, ou `"fetchmail"`, ou quelque chose de ce style. Si tout cela ne marche pas, vous pouvez toujours essayer le `telnet` (allez jeter un coup d'oeil dans la section qui suit).

[3] En français dans le texte original :-)

6.3.1 Faire un telnet sur un serveur POP3

Voilà comment je récupère mon adresse IP, car je suis rarement chez moi quand je fais tout ça, et en plus, ça impressionne toujours les gens à qui je le montre :-) Heureusement, votre FAI utilise un serveur POP3 pour le mail (j'espère), sinon, vous allez devoir vous débrouiller par vous même en parcourant les RFC pour le protocole dont vous avez besoin et en trouvant comment le faire par telnet.

Bon, tout d'abord, vous devez faire un telnet sur le port 110 du serveur de mail de votre FAI. Sous Unix, on fait ça avec `"telnet your.mail.server.net 110"`, sur un VAX/VMS vous devrez faire `"telnet your.mail.server.net/port=110"`, et depuis un navigateur Web, vous devrez utiliser `"telnet ://your.mail.server.net :110"`. Débrouillez-vous :-)

Dès que vous êtes connecté, vous devriez voir un truc ressemblant à :

```
>Connected to new-ls.lightspeed.net.
>Escape character is '^]'.

>+OK QUALCOMM Pop server derived from UCB (version 2.1.4-R3) at
>new-ls.lightspeed.net starting.

Tapez "user votre_nom" pour vous loger.

>+OK Password required for fenris.

puis "pass votre_password"

>+OK fenris has 2 message(s) (3030 octets).

Tapez "list" pour regarder la liste des messages.

>+OK 2 messages (3030 octets)
>1 2400
>2 630
>.
```

Vous voyez le message avec comme taille 630 ? C'est mon adresse IP ! Comment le sais-je ? Parce que c'est toujours de cette taille :-)

Tapez "retr numéro_du_message" pour récupérer le message que vous voulez.

```
>+OK 630 octets
>Received: from ulfheim.lightspeed.net (avatar@bak2-pp-ls.lightspeed.net
>[204.216.66.74]) by new-ls.lightspeed.net (8.6.12/8.6.12) with ESMTP id
>TAA12048 for <fenris@lightspeed.net>; Mon, 22 Apr 1996 19:15:37 -0700
>Received: (from avatar@localhost) by ulfheim.lightspeed.net (8.7/8.6.9)
>TAA00594 for fenris@lightspeed.net; Mon, 22 Apr 1996 19:15:29
>Date: Mon, 22 Apr 1996 19:15:29 -0700
>From: Deus In Machina <avatar@ulfheim.lightspeed.net>
>Message-Id: <199604230215.TAA00594@ulfheim.lightspeed.net>
>To: fenris@lightspeed.net
>Subject: Nouvelle adresse IP
>
>204.216.66.74
```

```
>
>.
```

Et voilà ! Faites `"dele numéro_du_message"` pour vous en débarrasser ou juste un `"quit"` pour le laisser et quitter. Maintenant, faites juste un telnet sur votre machine à cette adresse et amusez-vous bien !

Si tout ça ne marche pas, allez jeter un coup d'oeil dans la RFC 1225, qui décris le protocole POP3 en détail.

Par la même occasion, si ce hack ne marche pas car vous avez `cron` qui récupère votre mail automatiquement, alors, allez voir le hack 10 qui utilise `procmail` pour envoyer votre nouvelle IP à la demande.

7 Hack 5 : Changement dynamique de votre `/etc/hosts`

par Michael Driscoll *fenris@lightspeed.net*

Comme je l'ai dit plus tôt, Linux n'est pas vraiment à l'aise avec les changements d'IP. Par exemple, il arrive que votre `talkd` ne marche pas avec quand il y a des choses fantaisistes dans votre `/etc/hosts`. Néanmoins, le hack qui suit s'arrange pour changer votre `/etc/hosts` avec l'adresse IP qu'on reçoit à la connection.

Pour moi, ça a règlé les problèmes avec `"hostname"` et `"ntalk"`.

7.1 Ce dont vous avez besoin.

Juste une configuration PPP qui marche.

Ce hack est vraiment facile, tout ce que vous avez à faire c'est de suivre ce qui est écrit et remplir les blancs.

7.2 Création des patrons de hosts

NOTE : D'abord, faite une sauvegarde de votre `/etc/hosts`, juste au cas ou plus rien ne marcherais (on ne sait jamais :-). Faites juste `"cp /etc/hosts /etc/hosts.backup"`

D'abord, nous allons faire nos patrons de fichiers hosts. Le premier va s'appeler `/etc/hosts-down` et va contenir les lignes suivantes :

```
127.0.0.1 myhostname.mydomain.net localhost myhostname
```

Substituer **vos** hostname et nom de domaine dans ces valeurs. Ce fichier est celui qui sera utilisé quand vous serez déconnecté.

Le second fichier appelé `/etc/hosts-up` contiendra au moins les lignes suivantes :

```
127.0.0.1 localhost
--IP-- myhostname.mydomain.net myhostname
```

Comme précédemment, faites les substitutions de vos hostnames et noms de domaine, mais **ne touchez pas** au `"-IP-"`, car, il servira plus tard d'ancre pour faire un remplacement. Si vous ne comprenez pas, ce n'est pas grave, sachez simplement que vous **êtes** censés ne pas toucher au `"-IP-"` là dedans.

Vous pouvez faire un `/etc/hosts-up` plus gros si vous le voulez, il peut contenir des adresses de machines que vous visitez souvent ou des surnoms que vous voulez leur donner. par exemple, mon `/etc/hosts-up` ressemble à ça :

```
127.0.0.1 localhost
--IP-- ulfheim.lightspeed.net ulfheim
136.168.201.9 ultrix6.cs.csubak.edu ultrix ultrix6
128.214.48.39 linux.cs.helsinki.fi linux
152.2.254.81 sunsite.unc.edu sunsite
136.168.1.4 academic.csubak.edu academic
128.214.248.6 nic.funet.fi ftp.funet.fi funet
```

7.3 Et on retripote `ip-up` et `ip-down`

Ajoutez la ligne suivante à `/etc/ppp/ip-up` :

```
cat /etc/hosts-up | sed -e s/--IP--/$4/g > /etc/hosts
```

Ceci met en place le fichier `hosts-up` grâce à un script `sed` qui substitue le mot `"-IP-"` par `$4`, la variable qui contient notre nouvelle adresse.

Puis ajoutez la ligne suivante à `/etc/ppp/ip-down` :

```
cp /etc/hosts-down /etc/hosts
```

Cela copie le patron `hosts-down` vers `/etc/hosts`.

8 Hacks 6 à 8 : Divers hacks

Introduction par Michael Driscoll *fenris@lightspeed.net*
Code directement de Ryan R. Klems *klems@primenet.com*

Voici un peu de `C` envoyé par Ryan R. Klems *rklems@primenet.com* `<mailto:rklems@primenet.com>`.

Il y a trois programmes, le premier `mail.c`, est un `CGI` qui parcourt votre fichier de mail pour trouver votre nouvelle adresse IP donnée par le hack 1. Ensuite, il utilise l'adresse IP pour mettre en place une page qui contient un lien vers cette adresse IP.

Le second, `pppdm.c`, peut probablement prendre la place du hack 1, car il cherche une connection PPP, redémarre `pppd` s'il est mort, et envoie votre nouvelle adresse au serveur de mail de votre FAI.

Le troisième, `portmsg.c`, attend une connection telnet sur un port précis. Dès qu'une connection s'établie, il va récupérer votre adresse dans votre fichier de mail, et enverra un message la contenant.

Ces sources auront besoin d'un peu de personnalisation, alors, vous feriez mieux de ne pas y toucher si vous ne savez pas ce que vous faites :-)

Oh, oui, et Ryan m'a écrit pour me dire que ça ne le dérangerais pas de vous aider à mettre en place ce code chez vous si vous lui demandez gentiment :-)

8.1 mail.c

```
/*
 * mail.c written by Ryan R. Klems (rklems@primenet.com)
 * Copyright 1996, Author releases this source freely, allowing
 * copying and modification, so long as the original copyright notice
 * is maintained.
 *
```

```
 * I request that if you use this file you mail me... Thats all I ask =)
 *
 * A CGI for reading through your mailfile and finding an IP
 * address that you had your computer mail to you.
 *
 * Compiling:
 *    gcc mail.c -o mail.cgi
 *
 * Make sure to 'chmod +s mail.cgi' afterwards...must run with set uid
 * bit on to be able to open the mail file.
 */

#include <stdio.h>
#include <string.h>
#define MAILFILE "/var/mail/rklems" /* your mailfile */

main(void)
{
  FILE *mail;   /* file pointer for mail file */
  char bob[80], location[80];

  printf("Content-type: text/html\n\n");
  printf("<HTML><HEAD><TITLE>IP Address</TITLE></HEAD>\n");
  printf("<BODY><BASEFONT SIZE=4>\n");
  printf("<H1>IP Address</H1>\n");
  strcpy(bob, "42.**");
  if((mail = fopen(MAILFILE, "r")) == NULL)
    printf("Mail file is empty or does not exist.\n");
  else
  {
    /*
     * loop continues till end of file because you want most recent IP
     * 198.68. is the domain of my ISP, change to yours...
     */
    while(!feof(mail)) /* until reaching EOF, do this */
    {
      fgets(location, 80, mail); /* Grab a line, from mail */
      sscanf(location, "198.68.%s", bob); /* look for domain */
    }
    strcpy(location, "198.68.");
    strcat(location, bob);
    printf("The IP Address of your computer is: %s\n", location);
  }
  printf("</BODY></HTML>\n");
}
```

8.2 pppdm.c

```
/*
 * pppdm.c created by Ryan R. Klems (rklems@primenet.com)
 *   Released freely by the author to use/modify/copy/reditribute
```

```
 *   My only request is that if you use it...mail me and let me know =)
 *
 * This program keeps your link dialed up to an ISP and mails you
 * the newest IP address.  Useful for people with Dynamically allocated
 * IP addresses
 * uses the following files...
 * /root/ip        :   Outputs the IP to this file
 * /root/log       :   If logging is defined
 * /root/pppchat   :   The chat file set up for chat.
 *     My chat file looks like:
 *       ""  ATDT7917777 CONNECT "" "ogin:" "rklems" "assword:" "<password>"
 *  <password> is YOUR password of course (like I'm gonna give you mine ;)
 *  *NOTE* for silent dialing do ATMDT
 *
 *  Compiling...
 *    gcc pppdm.c -o pdm
 *  *NOTE* Don't call it anything like pppdm b/c it looks for pppd
 *  might accidentally kill itself off ;)
 */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <signal.h>
#include <time.h>
#define DOLOG

void main(void)
{
  FILE *fin,            /* multiuse file pointer */
       *popen();        /* proto of popen() */
#ifdef DOLOG
  FILE *log;            /* log file pointer */
#endif
  char line[80],        /* a line of a file */
       bah[80],         /* just stuff, also used for holding IP addr */
       crap[80],        /* just stuff */
       bob;             /* single char placeholder, not used for anything */
  int j,                /* flag for if a link was found */
      k,                /* flag for if this is a new link */
      pid;              /* pid of pppd process to kill off */
#ifdef DOLOG
  time_t now;           /* thing for time logging */
#endif

  for(;;) /* Loop forever... */
  {
    /* look at ifconfig for IP addr */
    if ((fin = popen("ifconfig", "r")) != NULL)
      while(fgets(line, 80, fin) != NULL)
```

```
        if(sscanf(line, "ppp0 %s", bah))
        {
          fgets(line, 80, fin);
          sscanf(line, "          inet addr:%15s", bah);
          j=1;
        }
    fclose(fin);
    if (!j) /* no link */
    {
      if((fin = popen("ps -a -x", "r")) == NULL)
      {
        fprintf(stderr, "PPPdm error: cannot open file.\n");
        exit(1);
      }
      /* scan through processes & kill off any zombie pppd processes */
      while(fgets(line, 80, fin) != NULL)
        if (sscanf(line, "%d  ?  %c     %4s pppd%s", &pid, &bob, crap, bah) == 4)
          kill(pid, SIGKILL);
      fclose(fin);
      k=0; /* new dial attempt */
      system("pppd connect 'chat -v -f /root/pppchat'"); /* try again */
#ifdef DOLOG
      now = time(NULL);
      if ((log = fopen("/root/log", "a")) == NULL)
      {
        fprintf(stderr, "Error in opening log file.\n");
        exit(1);
      }
      fprintf(log, "Initiating ppp-link. %s\n", ctime(&now));
      fclose(log);
#endif
      sleep(60); /* wait 1 min and check again */
    }
    if(j && !k) /* first time with new address */
    {
      if ((fin = fopen("/root/ip", "w")) == NULL)
      {
        fprintf(stderr, "Error in opening output file.\n");
        exit(1);
      }
      fprintf(fin, "%s\n", bah); /* write out addr */
      fclose(fin);
      /* mail it to yourself */
      system("mail -s IP joker@your.moma.com < /root/ip");
      k=1;
    }
    else /* take a nap and check again when we wake up */
    {
      j = 0;
      sleep(300); /* wait 5 minutes to check again */
```

```
    }
  }
}
```

8.3 portmsg.c

```
/*
 * Portmsg.c written by Ryan R. Klems (rklems@primenet.com)
 * Copyright 1996, Author releases this source freely, allowing
 * copying and modification, so long as the original copyright notice
 * is maintained.
 *
 * I request that if you use this program that you mail me.  Thats
 * all I ask.
 *
 * This program sets up a port on a server to accept telnets.  Upon
 * accepting a telnet, the program outputs a message, and then closes
 * the connection.
 *
 * address of message would be xxx.xxx.xxx.xxx yyyy where the x's
 * s the IP number or IP name, and yyyy is the port number set up
 * within this program.
 *
 * Compiling instructions:
 *      Linux : gcc portmsg_gen.c -o <your_file_name>
 *      SunOS : gcc portmsg_gen.c -lsocket -lnsl -o <your_file_name>
 * ***Note***
 *  I don't have access to any other operating systems, so if you
 *  compile this program on an OS I don't have listed, and use
 *  compiler options I didn't mention...please email me =)
 */

#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/uio.h>
#include <errno.h>
#include <netinet/in.h>
#include <strings.h>
#include <netdb.h>
#include <unistd.h>
#define HOSTNAME "ares" /* hostname of computer */
#define PORT 3000       /* tcp port to bind to */
/* #define GETHOSTNAME */ /* uncomment if your server has gethostname() */

void get_location(void);        /* proto of my mail reading function */
char location[80];              /* The IP address */

void main(void)
{
```

```
  struct in_addr host_ip_number;
  struct sockaddr_in host_ip_addr;
  struct sockaddr_in addr;
  char host_name[100];
  struct hostent *hp;
  int s, new_sock;
  int tmp, length;

  /*
   * The server I wrote this for doesn't have gethostname()
   * so, I put in a little fix...
   */
#ifdef GETHOSTNAME
  gethostname(host_name, sizeof(host_name));
#else
  strcpy(host_name, HOSTNAME);
#endif
  hp = gethostbyname(host_name);
  bzero((char *)&host_ip_addr, sizeof(host_ip_addr));
  memcpy((char *)&host_ip_addr.sin_addr, hp->h_addr, hp->h_length);
  host_ip_addr.sin_family = hp->h_addrtype;
  host_ip_number = host_ip_addr.sin_addr;
  host_ip_addr.sin_port = htons(PORT);
  host_ip_addr.sin_addr.s_addr = INADDR_ANY;
  /* open a socket s */
  s = socket(host_ip_addr.sin_family, SOCK_STREAM, 0);
  if ((int)s==-1)
  {
    fprintf(stderr, "Error in opening socket.\n");
    exit(1);
  }
  tmp = 1;
  if(setsockopt(s, SOL_SOCKET, SO_REUSEADDR, (char *)&tmp, sizeof(tmp))<0)
  {
    fprintf(stderr, "Error in setsockopt.\n");
    exit(1);
  }
  /* bind the socket to the server */
  if (bind(s, (struct sockaddr *)&host_ip_addr, sizeof(host_ip_addr))  == -1)
  {
    if(errno == EADDRINUSE)
    {
      fprintf(stderr, "Socket already bound!\n");
      exit(1);
    }
    else
    {
      fprintf(stderr, "Other error binding socket.\n");
      exit(1);
    }
```

```
  }
  /* tell the server to listen to the port */
  if (listen(s, 1) == -1)
/* 1 is the maximum size of the connection queue */
  {
    fprintf(stderr, "Error in listen.\n");
    exit(1);
  }
 while(1) /* just keep looping */
 {
   length = sizeof(addr);
   /*
    * port has been opened with socket(), bound with bind(), and set
    * active with listen(), now accept() watches the port for
    * connections, it will wait here until it has one...
    * new_sock is the file descriptor for the new socket
    */
   new_sock = accept(s, (struct sockaddr *)&addr, &length);
/*
 * The function get_location() and the send()'s are what I did to
 * suit my particular needs.  Put your own messages in here...
 */
   get_location();
   /* send just sends a string foo of length strlen(foo) with flags */
   send(new_sock, "Location:\n", 11, 0);
   send(new_sock, location, strlen(location), 0);
   close(new_sock); /* Close connection after message printed */
 }
}

void get_location(void)
{
  FILE *mail;   /* file pointer for mail file */
  char a[80];   /* char array for holding ip addr */

/*
 * FYI, this just opens my mail file, looks for a line with
 * 198.68.(the domain of my ISP), takes the last part, puts
 * the 198.68. in location, then cats the rest on the end
 */
  strcpy(a, "42.**");
  /* open /var/mail/ryan for read, and check to see there is a file */
  if((mail = fopen("/var/mail/ryan", "r")) == NULL)
  {
    strcpy(location, "Error in obtaining information.\n");
    return;
  }
  else
  {
    while(!feof(mail)) /* until reaching EOF, do this */
```

```
    {
      fgets(location, 80, mail); /* Grab a line, from mail */
      sscanf(location, "198.68.%s", a); /* look for domain */
    }
    strcpy(location, "198.68.");
    strcat(location, a);
    /* loops continues till end of file because I want most recent IP */
  }
}
```

9 Entrées DNS dynamiques

Si ça vous intéresse de mettre en place un hack mettant en jeu des entrées DNS dynamiques (Entrées DNS qui changent pour pointer sur votre ordinateur quand votre IP change) alors, vous aimerez ces URLs :

1. *www.cfmeu.asn.au/matthew/virtualip.html*
   Allez y faire un tour, Matthew a réussi à le faire marcher et demande à de généreux volontaires de le tester, pour le perfectionner. Ce hack demande d'avoir déjà les connaissances d'un DNS sur une IP fixe.
2. *ietf.org/internet-drafts/draft-ietf-dnsind-dynDNS-11.txt*
   A cette adresse, l'IETF[4] donne le papier sur les DNS dynamiques sur lesquels ils travaillent et devrais être lu par tous ceux qui veulent faire un pas dans cette direction.

Pour ceux qui ne {veulent, peuvent} pas s'embarrasser avec des détails, mais qui veulent juste une adresse dynamique mise en place sans problèmes, allez voir chez `DynDNS.com`. J'ai entendu dire pas mal de bien à propos de ce service, alors, il se pourrait qu'il vous intéresse. Leur site est à *www.dyndns.com*, et vous pouvez joindre Gus Hurwitz *info@dyndns.com* pour plus de détails. Un autre service viens d'apparaître, il s'agit de `dynip.com`, que vous pouvez aller visiter à *www.dynip.com*.

[De Artur Skawina *skawina@usa.net*] :

Un autre projet de DNS dynamique existe à *www.ml.org*. Il est opérationnel depuis janvier 1997 et marche très bien.

Il est très facile à utiliser : vous avez juste à enregistrer votre hostname une fois, ensuite, vous n'avez qu'à lancer un "client" à chaque fois que vous changez d'IP (Il existe déjà plusieurs client pour Linux, par exemple : *skawina.home.ml.org/mlddc.html*). Pour plus de détails sur ce service, allez visiter *www.ml.org/dyndns/*.

> [Note de l'éditeur : Si vous utilisez les services de `ml.org`, Pensez à leur faire une donation pour les remercier d'avoir travaille sans relâche sur ce projet. Apres tout, ils ne sont encore qu'un groupe de copains qui n'ont pas beaucoup de moyens et qui font des trucs que l'InterNIC vous ferais payer 100 $ pièce – MCD]

10 Hack 9 : Mettre à jour votre `.plan`

par Matthew Nuckolls *skawina@usa.net*

Ces deux scripts vous permettent de mettre votre IP dans votre `.plan` sur un serveur distant. Je l'utilise, comme ça, mes amis et ma famille peuvent voir quand je suis connecté, et où envoyer les `ytalk`.

[4] Internet Engineering Task Force, des gens qui mettent en place les futurs protocoles qui seront utilisés sur Internet

Mettez quelque chose ressemblant à

```
/etc/ppp/putip "Y'en a pas, la connection est morte "
```

Sur la première ligne de votre script `ppp-off`.

Vous aurez besoin d'un `$HOME/.netrc` en état de marche pour que putip marche. Le mien ressemble à ça :

```
machine rocket
login mnuck
password <mypassword>
```

et est `chmod 600`

`ip-up` :

```
#!/bin/sh
#
# Assurez-vous que ce fichier est chmod 711
# puisque le password apparaît en clair dedans

PLANLOC = /home/mnuck/.plan
REMOTE_USER_NAME = mnuck
REMOTE_PASSWORD =
REMOTE_PLANLOC = /afs/umr.edu/users/mnuck/pub/.plan
REMOTE_SERVER = rocket

echo "Mon IP dynamique est : " $4 > /tmp/ip.myip

cat $PLANLOC /tmp/ip.myip > /tmp/plan

echo $REMOTE_USER_NAME > /tmp/ip.script
echo $REMOTE_PASSWORD >> /tmp/ip.script
echo "put /tmp/plan" $REMOTE_PLANLOC >> /tmp/ip.script
echo "quit" >> /tmp/ip.script

ftp $REMOTE_SERVER < /tmp/ip.script &> /dev/null

rm -f /tmp/ip.myip
rm -f /tmp/ip.script
rm -f /tmp/plan
```

`putip` :

```
#!/bin/sh
#
# This script relies on a vaild .netrc file
#  -see ftp man page for details

PLANLOC = /home/mnuck/.plan
REMOTE_PLANLOC = /afs/umr.edu/users/mnuck/pub/.plan
REMOTE_SERVER = rocket
```

```
echo "Mon IP dynamique est : " $1 > /tmp/ip.myip

cat $PLANLOC /tmp/ip.myip > /tmp/plan

echo "put /tmp/plan" $REMOTE_PLANLOC > /tmp/ip.script
echo "quit" >> /tmp/ip.script

ftp $REMOTE_SERVER < /tmp/ip.script &> /dev/null

rm -f /tmp/ip.myip
rm -f /tmp/ip.script
rm -f /tmp/plan
```

11 Hack 10 : Une recette simple avec `procmail` pour retrouver votre nouvelle IP

Par Justin Cragin *beyond@lightspeed.net*
Ecrit par Michael Driscoll *fenris@lightspeed.net*

En voilà un facile, en supposant que vous avez déjà le `procmail` de configuré. Ce hack a besoin que vous ayez une adresse e-mail dédiée à ça sur internet, au cas ou vous auriez un job cron qui rapatrierais votre mail automatiquement.

`host.bogus.net` sera le nom de la machine ou vous avez le compte mail ou votre IP sera envoyée.

Dans votre `.procmailrc`, ajoutez la recette suivante :

```
:0:
* ^Subject:.*envoiemonip
|mail -s "Your new IP" other.mail.address@host.bogus.net < /var/run/add
```

Cette adresse e-mail ne doit pas être récupérée automatiquement par un job cron sur votre machine.

`/var/run/add` est bien sur le fichier qui contient votre adresse IP, souvenez vous du hack 1, si vous n'avez pas utilisé ce hack, mettez cette ligne dans votre `/etc/ppp/ip-up` :

```
echo $4 > /var/run/add
```

Pour utiliser ça, envoyer vous un mail depuis n'importe où sur Internet avec comme sujet : "envoiemonip", puis récupérez le mail sur le compte ou sera envoyé le mail, et attendez que le job cron chez vous récupère le mail et que le `procmail` y réponde.

C'était trop dur ? Dites-le-moi, j'essayerais d'éclaircir tout ça un peu.

12 Hack 11 : Une home page dynamique via ftp

par Michael Driscoll *fenris@lightspeed.net*

Voici un hack très simple qui permet de mettre sa nouvelle IP sur une page sur le serveur http de votre FAI. Il nécessite :

- `ncftp` (A moins que vous réussissiez à écrire un script qui utilise le ftp normal)
- Un accès `ftp` à votre FAI.

Vous pouvez voir ce hack en marche à *frob.base.org/*.

(J'ai aussi reçu des scripts de *shadow@indirect.com* pour un autre hack qui fait pratiquement la même chose, bien qu'il utilise le ftp normal et sont plus complexes. Vous pouvez les trouver à l'adresse *frob.base.org/contrib.tar.gz*).

> **Note** : une alternative à ce hack (utilisant les CGI) est présente dans le hack 14. C'est pratique pour ceux qui n'ont pas d'accès ftp à leur FAI.

12.1 Pourquoi j'utiliserais ça ?

1. Parce que c'est un hack qui est simple et propre.
2. Comme ça, vos amis et votre famille pourront facilement trouver votre IP (Ok, ce n'est peut être pas une bonne idée :-)
3. Comme ça vous pouvez lancer votre propre `httpd`, avec vos propres CGI (mettre un pointeur `http ://` sur votre nouvelle IP pour accéder à votre propre `httpd`
4. Depuis quand on a besoin d'avoir une raison pour faire quelque chose ?

12.2 Mise en place de la page

Tout d'abord, vous allez devoir écrire un peu de HTML pour aller sur la page du pointeur. Ce n'est pas trop l'objectif de ce HowTo de vous apprendre le HTML, alors, vous allez devoir l'apprendre par vous même. Allez jeter un oeil sur *w3.org* et *hoohoo.ncsa.uiuc.edu*, ils ont de bons manuels de base en ligne. Bon, quand vous écrirez la page, à la place de l'adresse de votre machine, mettez une ancre reconnaissable (je vais utiliser le mot `"-IP-"` dans cet exemple) comme ça, on va pouvoir utiliser un script qui remplacera l'ancre par l'adresse IP. Par exemple, une page pas mal serait :

```
<HTML>
<HEAD>
<TITLE>Ceci est la page de l'IP dynamique de Mike Driscoll</TITLE>
</HEAD>

<BODY>
<P>La derni&egrave;re adresse IP connue de ulfheim.lightspeed.net est :
<P><a href="telnet://--IP--">--IP--</a>
</BODY>
</HTML>
```

Remarquez que le `href` et le nom du lien utilisé `"-IP-"`, que le script va remplacer par votre IP avant l'envoi des pages sur le serveur.

Pour cet exemple, je supposerais que vous avez nommé ce fichier `/etc/ppp/index.html`.

12.2.1 Une note sur les redirections

par Artur Skawina *skawina@usa.net*

Au lieu de juste uploader la page avec votre IP courante, il est possible d'envoyer une page qui redirigera tout le monde sur votre machine, par exemple, un tag comme ça dans la section `HEAD` d'une de vos pages :
`<META HTTP-EQUIV="refresh" CONTENT="3 ; URL=http ://-IP-/page">`

12.3 Utilisation de `ncftp` pour la mise à jour automatique des pages.

Avant de faire cette partie, on va d'abord regarder quelle version de `ncftp` vous avez. Pour le savoir, exécutez `ncftp`

Voici deux exemples :

```
fenris@ulfheim:~$ ncftp
1.9.5 (October 29, 1995)
ncftp>_

mdriscol@ultrix6:/usr/stu/mdriscol$ ncftp
NcFTP 2.0.5 (May 1, 1995), by Mike Gleason, NCEMRSoft.
NcFTP> _
```

Suivant la version de `ncftp`, on va faire tout d'une façon différente. Si la version est 1.x.x (la mienne est 1.9.5), utilisez la première méthode, si la version est 2.x.x (celle de mon compte à l'école est 2.0.5) alors, utilisez la seconde méthode.

12.3.1 ncftp 1.x.x

Maintenant, on va ajouter quelques trucs au `.ncftprc` de root. Dans le mien, j'ai ce qu'il suit :

```
#set auto-binary on
#set recent-list off

machine Le.serveur.de.votre.FAI
        user Votre_nom_d'utilisateur
        password Votre_mot_de_passe
        macdef init
                cd /vers/votre/repertoire
                put /tmp/index.html index.html
                quit
```

Le auto-binary s'assure qu'on sera en mode binaire, le recent-list empêche le fichier `.ncrecent` d'être crée (je ne l'aime pas, c'est tout, vous pouvez en avoir un si vous voulez), et la rubrique machine définit le nom d'utilisateur, le mot de passe, et les actions à faire (vous n'aurez peut être pas besoin du `cd`, moi, je n'en ai pas besoin, puisque le serveur de mon FAI me met automatiquement dans le bon répertoire).

12.3.2 ncftp 2.x.x

par Tomas Jamate *tvj@miser.umass.edu*

> [**Note** : J'ai modifié le script de Tomas un peu avant de l'inclure dans le HowTo, si je l'ai cassé, dites-le-moi ! – MCD]

Voici les fichiers dont je me suis servi pour faire marcher `ncftp` v2.x.x avec le hack 8. Notez que `ncftp` v2.x.x met tous ses fichiers de configuration dans `~/.ncftp`. Je ne garantis pas que cela marchera pour vous, mais ça marche pour moi. Lisez les pages de man si vous voulez connaître les autres options de `ncftp`.

Premièrement, connectez-vous manuellement à votre FAI avec `ncftp`.

```
$ ncftp -u service.provider.com
```

Le `-u` oblige à demander le nom d'utilisateur et le mot de passe. Une fois connecte, faites un `cd` dans le répertoire ou sont vos pages. Cela va créer une entrée pour `service.provider.com` dans le bookmark de `ncftp`. (Je n'aime pas trop les bookmarks, mais hé, ça marche). Quittez `ncftp`.

Editez `~/.ncftp/bookmarks`. Vous allez voir une entrée avec votre nom d'utilisateur, vous devrez ajouter le mot de passe juste après le nom d'utilisateur. ça devrait ressembler à :

```
NcFTP bookmark-file version: 6
Number of entries: 1
service,service.provider.com,Utilisateur,Motdepasse,,/repertoire/de/votre/page,[etc]
```

Maintenant créez et éditer `~/.ncftp/macros`, il devra ressembler à ça :

```
macro .open.example
put /tmp/index.html index.html
exit
end
```

Maintenant, vous pouvez envoyer automatiquement votre page avec :

`"ncftp -L example"`, le `-L` force le programme en mode ligne de commande
ou
`"ncftp -Lf example"`, le `-f` force à recouvrir le fichier existant, pour le paranoïaque ;-)

12.4 Une routine de recherche/remplace simple avec `sed`

Maintenant, on va écrire la routine `sed`. Mettez les lignes qui suivent dans `/etc/ppp/ip-up` :

```
# On commence le code de l'IP dynamique
if [ -e /tmp/index.html ]; then
        mv /tmp/index.html /tmp/index.html-`date +%s`
fi
cat /etc/ppp/index.html | sed -e s/--IP--/$4/g > /tmp/index.html
ncftp le.serveur.de.votre.fai > /dev/null 2>&1
```

D'abord, on vérifie la préexistence de `/tmp/index.html`. Un cracker pourrait essayer d'exploiter notre utilisation de ça pour sortir n'importe quel fichier du disque en le liant à `/tmp/index.html` et en attendant que le script soit lancé. Donc, si le fichier existe déjà, on le sauvegarde gentiment pour que vous puissiez y jeter un coup d'oeil plus tard, empêchant toutes manipulations frauduleuses...

Puis, on lance le script `sed`. Il substitue toutes les occurrences de notre ancre `"-IP-"` avec notre nouvelle adresse IP et envoie le tout dans le nouveau `index.html` dans `/tmp`. `ncftp` est alors lance pour envoyer notre nouveau `/tmp/index.html` qui à l'IP actuelle en lui.

Félicitations, vous l'avez fait, vous avez maintenant une home page dynamique que vous pouvez utiliser pour pointer sur votre machine ! Amusez vous bien en le personnalisant ! Éditez juste `/etc/ppp/index.html`, et il sera mis à jour la prochaine fois que vous vous connectez à votre FAI.

13 Hack 12 : Envoyez un message sur votre pager avec votre nouvelle adresse IP

par Michael Driscoll *fenris@lightspeed.net*

Voici un joli hack qui ne sera probablement utilisé par personne. Il requiert d'utiliser un programme pour faire composer à votre modem le numéro de votre pager et lui envoie votre nouvelle IP. Il requiert :

- un modem (La vitesse n'est pas importante puisqu'on ne va faire que composer un numéro [Vous n'êtes pas super content d'avoir garde votre modem à 300bps ? :-)])
- Une ligne téléphonique supplémentaire (elle n'a pas besoin d'être dédiée, et heureusement, vous avez un modem qui abandonnera s'il y a déjà une conversation sur la ligne (Je pense que c'est quelque chose qu'il faudra que vous vérifiiez, quelques modems sont un peu brutaux avec ça[5] .
- Un pager ordinaire
- Un programme pour composer le numéro

13.1 Le programme qui compose le numéro

Pour ce hack, vous allez avoir besoin d'un programme qui sait composer depuis la ligne de commande. Pour l'exemple suivant, j'utilise `"modem-stats"` par Kenneth J. Hendrickson *kjh@usc.edu*, qui peut être trouvé à *sunsite.unc.edu/pub/Linux/apps/comm/modem-stats.tar.gz* ou pas loin.

J'imagine que d'autres programmes seraient tout à fait capables de faire la même chose.

13.2 Appel depuis `ip-up`

On va mettre les lignes suivantes dans `/etc/ppp/ip-up` :

```
#On sépare $4 (l"adresse IP) en quatre morceaux
part1=‘echo $4 | cut -f1 -d.‘
part2=‘echo $4 | cut -f2 -d.‘
part3=‘echo $4 | cut -f3 -d.‘
part4=‘echo $4 | cut -f4 -d.‘
#On lance le programme
/usr/sbin/modem-stats -c "atz" /dev/cua2
/usr/sbin/modem-stats -c "ats7=15" /dev/cua2
/usr/sbin/modem-stats \
 -c "atdt6384658,,,"${part1}"*"${part2}"*"${part3}"*"${part4} /dev/cua2
```

Explications :

Le `atz` initialise le modem, le `ats7=15` règle le temps d'attente pour la porteuse, les `','`s sont des pauses, les `'*'`s sont utilisées pour remplacer les - (c'est ce que ça fait sur mon pager en tout cas, espérons qu'il en sera de même pour le votre).

Oh, oui, vous aurez sûrement de meilleurs résultats si vous mettez **votre** numéro de pager à la place de `638-4658`, car, c'est **mon** numéro de pager :-), de plus, remplacez `/dev/cua2` par le nom de votre modem.

Félicitations, vous avez réussi !

14 Hack 13 : logins `xterm` à travers un firewall

par Brad Baker *bpb@mlb.cca.rockwell.com*

[5]NdT : Je confirme, mon modem Olitec si il ne trouve pas la tonalité raccroche la ligne, redécroche, et ce tant qu'il n'a pas la tonalité :-(

> [Note de l'éditeur : Comme l'auteur de ce hack le dit judicieusement, ce hack peut avoir des problèmes de sécurité, comme la possibilité d'un vol de mot de passe ou d'ouvrir une brèche dans la sécurité du firewall à cause d'un login à l'extérieur de votre réseau local. Ce hack est inclus plus pour sa valeur que pour son utilité. Je ne vous conseille pas de l'utiliser à moins que vous soyez l'administrateur système de ce réseau ou que vous soyez en très bons termes avec lui ou elle :-) – MCD]

Voici un autre hack que vous pouvez essayer. Il marche, et je ne suis pas sûr des risques de sécurité, mais il est très amusant.

Cela serait sympa de pouvoir accéder à ma machine de travail (une station Sun) depuis chez moi, et réciproquement, le problème est que le telnet est bloqué par le firewall au bureau. Voici une façon de le contourner.

Pour cet exemple, je vais donner la façon d'accéder à ma station Sun (au travail) depuis chez moi avec Linux, avec une connection PPP à mon FAI et une IP dynamique.

>Chez moi, quand je veux accéder à ma machine de travail, je me connecte et je lance X, je tape `"xhost +"`, je trouve mon IP dynamique et je l'envoie à mon travail dans un e-mail avec un format particulier. Sur ma machine de travail, j'ai un script `procmail` qui lis les mails dont le sujet est spécial, disons `"X-W"`. Si le texte du message à un format défini, alors, l'IP est extraite du message, et il lance un `xterm` qui s'affiche directement chez moi comme ça :

```
xterm -display l.ip.que.j.ai.envoye:0.0 -e login
```

Voila ! En à peu près 30 secondes à une minute, j'ai un écran de login qui apparaît chez moi ! Je n'ai pas essayé d'aller dans l'autre direction car l'ordinateur de la maison n'est pas tout le temps connecté, mais il devrait être possible de faire la même chose.

Je suis assez préoccupé par les problèmes de sécurité, et je l'ai mis hors service pour l'instant jusqu'à ce que j'en apprenne plus. Quand le `xterm` apparaît, il donne un prompt de login, et je ne sais pas comment le Secure Keyboard[6] marche dans un `xterm`. De plus, je ne suis pas sur des sécurités fournies par `X`. Je suis plus préoccupé par le `xterm` apparaissant à une mauvaise IP ou sur une IP que je viens de quitter dû à une déconnection, mais bon, la majorité des utilisateurs utilisent Windows, et l'accès X sera simplement ignoré.

D'autres sécurités pourraient être ajoutées en encryptant l'IP dans le mail avec PGP ou une encryption classique et de restreindre l'IP à un certain nombre, mais mon FAI refuse de me donner ce nombre.

Voici le script `.procmailrc`, le script et un exemple de message mail que j'ai utilisé pour tester (disons que `111.222.333.444` est l'IP dynamique).

Le morceau de `~/.procmailrc`

```
:
^Subject: X-W
| cat | $HOME/scripts/send_xterm
```

`~/scripts/send_xterm`

```
#!/bin/sh
XX=`cat $1 | grep "^IP: .*" | sed "s/IP: \(.*\)/\1/"`
XX=`echo $XX | perl -e '$_ = <>; s/[^0123456789.]//g; print'`
xterm -display $XX -e login &
```

Exemple de message

---

[6] Clavier sécurisé, il permet d'empêcher la récupération de ce qui est tapé

```
To: bpb@my.machine.com
Subject: X-W

IP: 111.222.333.444:0.0
```

Vous pouvez essayer sur votre propre machine en vous envoyant un mail à vous-même (après avoir installe `send_xterm` et la règle `.procmailrc` en interne bien sur).

15 Hack 14 : Home page dynamique via un CGI

par Michael Driscoll *fenris@lightspeed.net*

Ce hack est similaire au hack 11, mais il diffère en la méthode employée. Alors que le hack 11 implique l'utilisation d'un accès ftp vers le serveur http de votre FAI pour mettre à jour la page, celui ci accède à un CGI sur le serveur pour mettre en place la page elle-même.

Pour utiliser ce script, vous allez devoir convaincre votre l'administrateur de votre FAI de mettre ce CGI sur le serveur. Il est tout à fait possible qu'ils vous demandent de l'argent pour vérifier la sécurité de ce script, puisque cela va leur demander du temps, et c'est pour eux l'assurance que le script ne présentera pas de risque une fois mis en place (je vais faire de mon mieux pour qu'il soit le plus "sûr" possible). Par exemple, mon FAI (lightspeed.net) me demande $50 par script pour le vérifier.

Attention : Non seulement ce script est totalement nouveau et (largement) non testé, mais en plus, je l'écris à 3 heures du matin car je n'ai pas envie de faire ma philosophie. Si vous ne connaissez rien aux CGI ni comment les rendre sûrs, alors, je vous suggère d'attendre que les quelques bugs soient retirés avant de l'envoyer à votre pauvre administrateur système.

Note : Bon, cela fait maintenant deux mois que j'ai écrit ce message, et je n'ai rien entendu à propos de la sécurité de ce CGI, ni pour, ni contre. Alors, soit il **est** sûr et personne ne me l'a fait savoir, soit personne ne l'utilise :-) Par conséquent, je vous conseille de faire attention.

15.1 Voilà ce que l'on va faire.

A la base, tout ce que vous avez à faire, c'est de prendre le script et de remplir quelques endroits.

Dans le script, on va essayer de :

- Récupérer un paramètre appelé newip=, en s'assurant que c'est une IP qui convient.
- Récupérer un paramètre appelé password= ce qui est une **très** mauvaise sécurité pour s'assurer que personne ne pourra utiliser le script pour mettre sa propre IP (Note : je sais que ce n'est pas 100% efficace, mais bon, ça ne devrais pas être **trop** grave, et cela ne pourra rien détruire, et puis il y aura les logs du serveur pour savoir qui a accédé à la page...)
- Confronter les $REMOTE_HOST et $REMOTE_ADDR du CGI avec votre nom de domaine pour être sur que c'est une requête locale.
- Et enfin, si tout marche, mettre en place la page...

15.2 Le script (dynip.pl)

```
#!/usr/bin/perl
# Ecrit par Michael Driscoll <fenris@lightspeed.net>
# Les suggestions sont les bienvenues

require 5; # Requiert perl5
```

```
use CGI; # Ceci est le module CGI.pm, disponible sur tous les sites CPAN tels
         # ftp://ftp.cdrom.com/pub/perl/CPAN/modules/by-module/CGI

# Remplir les trucs qui suivent
$givendomain = "domain.net";              # Mettez le domaine de votre FAI
                                          # (Pour l'instant, ce sont les deux
                                          # premières parties, mais cela peut
                                          # être change en dessous avec le
                                          # split())
$givenpassword = "potrzebie";             # Mettez le mot de passe à utiliser
$filename = "/some/path/to/my/index.html"; # Demandez à votre admin de mettre
                                          # ici la page à mettre à jour
                                          # Note à l'administrateur :
                                          # le fichier devra avoir les
                                          # droits en écriture pour l'uid
                                          # de httpd, à moins que vous
                                          # utilisiez la méthode suid
                                          # d'Apache ou un autre truc
                                          # (mais c'est pas recommandé)

# Ce qui suit joue avec les IP en utilisant la forme aaa.bbb.ccc.ddd
$higha = "255"; # Mettez ici le plus grand numéro de la partie aaa
                # de votre FAI.
$lowa = "0";    # De même pour le plus petit numéro de la partie aaa.
                # probablement la même que $higha vu que la majorité des
                # FAI ne prennent pas plus qu'un domaine de classe A :-)
$highb = "255"; # Mettez la borne supérieure pour bbb.
$lowb = "0";    # La borne inférieure de bbb.
$highc = "255"; # Vous avez compris.
$lowc = "0";    #
$highd = "255"; #
$lowd = "0";    #

# on fait un lien entre $query et le module cgi.pm
$query = new CGI;

# $newip est la nouvelle IP du paramètre newip
$newip = $query->param("newip");

# $trypassword est la tentative de mot de passe à confronter avec $givenpassword
$trypassword = $query->param("password");

# $hostname est le nom d'hote à confronter avec $givendomain
$hostname = $query->remote_host;

# Verifier le mot de passe avant tout
unless ("$trypassword" eq "$givenpassword") {
        print "Content-type: text/plain\n\n";
        print "Sorry, wrong password\n";
        exit;
```

```
}

# On découpe l'IP dans @IP
# On vire d'abord les caractères non numériques sauf le '.'
# on le splice() pour supprimer les trucs après les quatre premier points
# (Comment c'est arrivé là ??  De toute façon, j'essaye de le rendre aussi
# peu craquable que possible).
$newip =~ s/[^\d\.]//g;
@IP = split /\./, $newip;
splice(@IP, 4);

# On vérifie que l'IP est dans les bornes
unless (($lowa <= $IP[0]) &&
        ($lowb <= $IP[1]) &&
        ($lowc <= $IP[2]) &&
        ($lowd <= $IP[3]) &&
        ($higha >= $IP[0]) &&
        ($highb >= $IP[1]) &&
        ($highc >= $IP[2]) &&
        ($highd >= $IP[3])) {
        print "Content-type: text/plain\n\n";
        print "Désolé, cette adresse IP ne fait pas partie des adresses admises\n";
        exit;
}

# Maintenant, on vérifie le nom de l'hôte
# On le découpe dans @hostnamearray
@hostnamearray = split /\./, $hostname;
$dompart2 = pop @hostnamearray;
$dompart1 = pop @hostnamearray;

# On le vérifie
unless (("$dompart1"."\.$dompart2" eq "$givendomain") ||
        ("$hostname" eq "localhost")) {
# On affiche une erreur
        print "Content-type: text/plain\n\n";
        print "Désolé, vous n'avez pas le bon nom de domaine\n";
        exit;
}

# Tout à l'air d'être OK, on va donc mettre en place la page
# Prenez garde aux trucs dans le style de #, \, $, @, %, et '

open(FILE, ">$filename") ||
        die "dynip.pl cannot open $filename to write: $!";
flock(FILE, 2);

# Ici commence le HTML. $newip est notre nouvelle adresse IP.
print FILE "<html><head><title>Title etc.</title></head>\n\n";
```

```
print FILE "<body><h1>body here</h1><hr>\n";
print FILE "<p>blah blah blah\n";

print FILE "<p>And <a href=telnet://$newip>here</a> est un lien vers mon ";
print FILE "addresse IP actuelle.\n";

print FILE "</body></html>\n";
flock(FILE, 8);
close(FILE);

# On dit que tout s'est bien passé
print "Content-type: text/plain\n\n";

print "Installation réussie\n";

# Fin
```

Accéder au CGI quand la connection s'établie

Maintenant, tout ce que nous avons à faire est d'y accéder avec lynx dans notre `/etc/ppp/ip-up`. La ligne suivante devrait le faire, utilisant le $4 de `/etc/ppp/ip-up` comme nouvelle IP :

```
lynx -dump \
"http://www.votre-FAI.net/cgi-bin/dynip.pl?newip=$4&password=potrzebie" \
>> /etc/ppp/lynxlog
```

Mettez le mot de passe que vous avez choisis à la place de `"potrzebie"`, bien sur. Et assurez-vous de bien faire `chmod 700 /etc/ppp/ip-up` quand c'est Ok, car il y a des informations sensibles dedans.

(Au fait, les \ sont juste utilisés pour que la commande tienne en 80 colonnes. Ils ne font qu'escaper les retours à la ligne pour que ce soit considéré comme une seule commande)

De plus, videz périodiquement le `/etc/ppp/lynxlog` sinon, il va être très gros après quelque temps :-)

Voilà... C'est fait !

16 Hack 15 : Suggestion pour les rc.*

par Jeremy D. Impson *jdimpson@camelot.syr.edu*
Écris par Michael Driscoll *fenris@lightspeed.net*

Jusqu'ici, ce HowTo avais pleins de morceaux d'instructions dispersées dans plein de fichiers avec des morceaux dans les `/etc/ppp/ip-down` ou les `rc.*`, qui devaient être exécutés quand le système plantait et que la connection PPP étais toujours là. Jeremy m'a suggéré de lancer `/etc/ppp/ip-down` au démarrage plutôt que de le recopier petit à petit dans les `rc.*`. Bien sur, il faut s'assurer que les commandes dans `/etc/ppp/ip-down` ont un sens au démarrage, et ne s'appuient pas sur des paramètres passés par `pppd` (si c'est le cas, vous pouvez toujours écrire une fausse ligne de commande), il n'y a qu'à ajouter la ligne suivante dans le `rc` de votre choix :

```
# Exécute /etc/ppp/ip-down si pppd n'a pas été arrête proprement
if [ -f /var/run/ppp?.pid ]; then
        /etc/ppp/ip-down
fi
```

Un mot de Jeremy : Ce petit truc ne sera pas utile dans chaque situation, mais il l'est pour moi, et il le sera peut être pour vous

Merci Jeremy !

17 Hack 16 : Empêcher les time-out imposés par votre FAI avec un ping

par Artur Skawina *skawina@usa.net*

> [Note de l'éditeur : Comme Arthur le dit, celui ci peut aller trop loin, et peut réellement avoir un impact sur votre FAI s'ils sont un peu justes dans le nombre de leurs lignes. Si vous faites ceci et que votre FAI vous demande de l'enlever, vous devriez sérieusement penser à dépenser un peu plus d'argent pour acheter un modem à votre FAI, ce qui vous fournira certainement une IP fixe, rendant ce HowTo inutile :-) – MCD]

Pour garder une connection RTC, il est parfois nécessaire de s'assurer que la connection n'est pas inactive pendant de longues périodes. Certains FAI vous déconnectent automatiquement s'il n'y a pas de trafic, et certains modems ont aussi la possibilité de mettre fin à une connection inactive. Alors qu'il est simple de reconfigurer votre modem, cela risque d'être dur, voir impossible de convaincre votre FAI de faire la même chose (Ils n'aiment pas les gens utilisent leurs ressources 24h par jours).

La solution ? Ajoutez la ligne suivante à votre script `ip-up` :

```
ping <une_machine> -i 180 &
```

Ou <`une_machine`> peut être le serveur DNS principal de votre FAI ou autre chose...

Pour changer la déconnection automatique des modems, il va falloir regarder dans la documentation du modem. Par exemple, sur les modems à base de CirrusLogic, c'est `ATS90=0`

18 Hack 17 : Utilisation des SSI pour retrouver son IP

par Dale Jolliff *taftbbs@e-tex.com*

Ceci n'est pas spécialement pour Linux, mais pour n'importe quoi, et vous n'avez pas à faire des modifications. Il requiert néanmoins que votre FAI autorise les SSI[7] dans vos pages.

Il y a un exemple à *www.e-tex.com/personal/taftbbs/onlinecheck.shtml.* Allez y jeter un coup d'oeil... ;>

Vous allez avoir à connaître certaines choses à propos de votre FAI :

- Le nom (ou l'IP) du routeur sur lequel vous vous loggez.
- La "gateway" et votre propre nom d'utilisateur, vous pouvez les récupérer avec la commande "`ifconfig`", c'est l'IP qui n'est pas votre machine... pour les utilisateurs de Windows, Trumpet Winsock a un "default gateway", et pour les utilisateurs de Win95, c'est dans le coin de la configuration TCP/IP.

La partie la plus compliquée est que votre FAI vous laisse avoir votre "home page" – C'est ce que font tous les FAI de nos jours. S'ils ne sont pas assez pointus pour limiter l'exécution des CGI et des SSI, et y'a même mieux... Mon FAI autorise les SSI, mais pas l'exécution des CGI (bien sur, si vous pouvez faire des SSI, vous pouvez faire des CGI... il faut juste les appeler autrement).

[7] Server Side Includes : directives que le serveur http remplace par leur résultat

J'ai enlevé tous les trucs inutiles, laissant juste la partie critique pour que ça marche...

Ajoutez ces lignes sur une page SSI (d'habitude, l'extension est `".shtml"` pour la majorité des serveurs) :

`david5.e-tex.com` est le routeur sur lequel je me logge quand je me connecte à mon FAI...
`taftbbs` est mon nom d'utilisateur sur leur système
Voici donc une ligne à mettre dans une page...

```
<pre><!--#exec cmd="finger @david5.e-tex.com | grep taftbbs"--></pre>
```

```
<a href="http://<!--#exec cmd="finger @david5.e-tex.com | grep taftbbs |
cut -c '66-'"-->">Click here to see if you can see anything!</a>
```

La ligne ci dessus va faire un lien qui s'affichera sur la page. Faites un finger sur votre routeur, et regardez la sortie, le `"cut -c '66-'"` sera différent suivant le routeur. Si vous avez des comptes sur plusieurs FAI, et comme il semble que tous aient des routeurs différents, et chacun aura un format différent quand vous aller faire un finger dessus.

Une petite explication de ce qui se passe :

<a `href="http ://` <–Une ancre HTML standard

< `!-#exec cmd="` <– Voici la commande SSI. Cela signifie que ce sera lu et exécuté par le serveur avant de vous être envoyé. Donc, même si vous regardez la source de la page depuis votre navigateur, vous ne verrez pas ça, mais juste le résultat de la commande.

```
finger @david5.e-tex.com | grep taftbbs | cut -c '66-'
```

Voilà ce qui est exécuté. On fait un finger sur le routeur, on ne garde que le bon utilisateur, et on coupe tout sauf l'IP, ce qui sur le routeur que j'utilise commence à la colonne 66, jusqu'à la fin de la ligne... Cela peut être totalement différent pour les autres.

```
"-->
```

Et puis on ferme la commande SSI proprement, sinon, ça ne marchera pas... Faites attention à la façon d'utiliser les apostrophes et les guillemets, il est très facile de faire une erreur, jouez un peu avec, vous verrez.

```
">Click here to see if you can see anything!</a>
```

Et bien sur, on ferme le lien HTML, et on met un texte cliquable.

A On a besoin d'autre hacks !

Si vous avez un hack concernant les IP dynamiques et que vous voudriez contribuer à ce HowTo, contactez moi. Attention, si vous faites ça, il aura à être couvert par le copyright de la section 1.7, pour des raisons de publications...

Si vous {avez aimé ce HowTo, détesté ce HowTo, pensé que ce HowTo étais trop confus, trouvé quelque chose ici qui ne marche pas ou qui est totalement faux, voulez m'envoyer un mail, ne voulez pas m'envoyer un mail, trouvé un problème de sécurité, etc} alors, s'il vous plaît, envoyez un moi un e-mail à *fenris@lightspeed.net*. Je le pense, j'aimerais avoir **n'importe quel** retour sur ce HowTo, même pour me dire que vous l'avez juste lu !

Vraiment ! Envoyez-moi un e-mail pour n'importe quoi ! J'adore **réellement** recevoir des e-mails !

Les remarques négatives **n'iront pas** vers `/dev/null`. Car je pense que si vous en êtes à ce point, je ferais mieux de vous écouter.

Si vous écrivez un chapitre pour ce HowTo, non seulement vous aiderez la communauté Linux, mais vous recevrez tous les honneurs à la fin de ce HowTo, vous serez remercié comme contributeur au début de ce HowTo, vos idées seront archivées sur presque toutes les distributions de Linux à travers le monde, vous serez enviés de vos pairs, etc.

B CRÉDITS

(Par ordre d'apparition)

Michael Driscoll ***fenris@lightspeed.net***
Est le mainteneur de ce hack et aussi l'auteur de certains d'entre eux.

Christian G. Warden ***cwarden@loop.com***
M'a aidé à débogger le script `keepalive.sh` utilisé dans le hack 2, maintenant, les choses devrais bien se passer dans tous les cas, incluant le cas ou la connection est rompue et ou le `pppd` ne meurt pas.

Justin Cragin ***beyond@lightspeed.net***
M'a donné l'idée des hacks 2 et 4, et a faillit devenir fou quand il a découvert que je les lui avais volé. Je lui ai aussi piqué le message de son répondeur, je pense donc que je lui dois bien ça. Il a aussi récemment découvert un hack qui est devenu le hack 10, alors il est maintenant trois fois remercié. Et puis il a aussi aidé à faire germer l'idée du hack 12, je l'en remercie encore une fois. Bon, je pense qu'il sera le nouveau mainteneur quand je déménagerais cet été au CSM et que j'aurais mon IP stable.

shadow@indirect.com
M'a envoyé un superbe hack de sa création, allez voir 12.0 pour plus de détails. Il semble que j'ai perdu son nom, Heureusement, il va bien me le renvoyer :-)

Ajit Deshpande ***adeshpan@ddt.eng.uc.edu***
Voulais avoir son nom ici.

Divya Mahajan ***vmahajan@giasdl01.vsnl.net.in***
M'a envoyé les informations à propos de `diald`, il remplacera certainement le `crontab` dans le hack 2.

Ryan Klems ***rklems@primenet.com***
M'a envoyé plein de ses hacks, allez voir 8.x pour plus de détails.

Matthew Driver ***mdriver@cfmeu.asn.au***
M'a donné l'adresse de la page sur les DNS dynamiques, allez voir le chapitre 9.0 pour l'URL.

Matthew Nuckolls ***mnuck@umr.edu***
M'a fait le hack 9, qui met à jour le .plan contenant votre nouvelle adresse IP sur un serveur distant via ftp.

Scott Johnston ***sj@odin.iac.net***
M'a montré comment rendre le hack 2 plus efficace (enlever une entrée crontab immense et la mettre dans un script appelé par cron à la place), et m'a donne quelques autres adresses.

Brad Baker ***bpb@mlb.cca.rockwell.com***
A écrit le superbe hack 13.

Christian Hardmeier ***101502.1521@compuserve.com***
M'a motivé pour écrire le hack 14.

Justin (Gus) Hurwitz ***hurwitz@dyndns.com***
M'a fait connaître son service de DNS dynamique pour une machine avec une IP dynamique. Allez voir l'adresse au chapitre 9.0.

Tomas Jamate ***tvj@miser.umass.edu***
M'a fait me souvenir de `ncftp 2.x.x` et a même été jusqu'à m'envoyer ses propres scripts pour que je les mette dans ce HowTo (merci Tomas !). Il va à umass, mais malheureusement, il n'écoute pas les Pixies. Tant pis.

Jeremy D. Impson ***jdimpson@camelot.syr.edu***
M'a envoyé une suggestion qui est devenu le hack 15.

Steve Przepiora ***gearhead@dreamscape.com***
A écrit le hack 1, et m'a aide à réorganiser le HowTo pour qu'il devienne la version 2.0.0.

Whit Blauvelt ***whit@transpect.com***
A tous mes remerciements pour m'avoir beaucoup écrit à propos de ce HowTo, et surtout pour avoir réparé le hack 1.

Paul C. Richard ***pcricha@cs.concordia.ca***
A aussi mes remerciements pour avoir aide à faire marcher le hack 1.

Per Sjoholm ***Soile.Kaasila@sth.frontec.se***
M'a donné la nouvelle adresse de `diald`.

Bill Duncan ***bduncan@beachnet.org***
M'a aussi donné la nouvelle adresse de `diald`.

Artur Skawina ***skawina@usa.net***
En a beaucoup fait, comme le hack 16, une addition au chapitre 9, et la nouvelle section 12.2.1.

Scott Dier - DiEMaN ***sdier@isd.net***
M'a fait connaître le projet ML.org.

Dale Jolliff ***taftbbs@e-tex.com***
M'a envoyé le hack 17.

Bryan Rittmeyer ***bryanr@flash.net***
A debuggé le hack 2 et m'a parlé de ML.org.

Marek Kubita ***kubitovi@mbox.lantanet.cz***
A supprimé un trou de sécurité dans le hack 13.