HOWTO Réseaux Privés Virtuels

Arpad Magosanyi <mag@bunuel.tii.matav.hu> v0.2,7 Aug1997 Traduction par Nicolas Prigent <prigentn@cicrp.jussieu.fr> v0.2, 7 August 1997

1 Modifications

Le problème de 'pas de contrôle du tty' -> -o 'BatchMode yes' par Zot O'Connor <zot@crl.com>

Avertissements au sujet du kernel 2.0.30 par mag

2 Burps

Vous avez sous les yeux le howto Réseaux Privés Virtuels, un rassemblement d'informations concernant la manière de monter un Réseau Privé Virtuel sous Linux (et les autres Unix en général).

2.1 Copyright

2.2 Mise en garde

Comme d'habitude : L'auteur n'est en aucun cas responsable de tout dommage occasionné. Pour la formulation exacte, se référer à la partie correspondante de la GNU GPL 0.1.1

2.3 Mise en garde

Il est question de sécurité : vous n'êtes pas en sécurité si vous n'instaurez pas une politique de sécurité efficace, et autres choses ennuyeuses de ce genre.

2.4 Remerciements

Merci à tous ceux qui ont écrit les outils utilisés.

Merci à Zot O'Connor <zot@crl.com> pour m'avoir montré le problème de "no controlling tty", et sa solution.

Le traducteur voudrait remercier Aude Hurtrel pour son aide précieuse.

2.5 Etat de ce document.

Voici les préliminaires. Vous devez avoir une connaissance générale de l'administration IP, au moins quelques connaissances sur les firewalls, ppp et ssh. Vous êtes de toute facon censé les connaitre si vous voulez monter un VPN. J'ai simplement décidé d'écrire mes expériences afin de ne pas les oublier. En fait, Il se peut qu'il y ait des trous de sécurité. Pour être honnête, j'ai réalisé mes essais sur des machines configurées comme des routeurs, et pas des firewalls, c'est plus simple.

2.6 Documentation relative au sujet

– Le Firewall-HOWTO /usr/doc/HOWTO/Firewall-HOWTO
– Le PPP-HOWTO /usr/doc/HOWTO/PPP-HOWTO.gz
– La documentation ssh /usr/doc/ssh/*
– The Linux Network Admins' Guide
– NIST Computer Security Special Publications http ://csrc.ncsl.nist.gov/nistpubs/
– Firewall list (majordomo@greatcircle.com)

3 Introduction

Les firewalls étant de plus en plus largement utilisés dans les systèmes de sécurité internet et intranet, la capacité de réaliser de bons VPNs est de plus en plus importante. Voici le relevé de mes expériences. Les commentaires sont les bienvenus.

3.1 Conventions de dénomination

J'utiliserais les termes de "firewall-maître" et "firewall-esclave", bien que réaliser un VPN n'ait rien à voir avec l'architecture client-serveur. Je me réfère simplement à eux en tant que participant actif et passif de la mise en place de la connection. On utilisera la dénomination de "maître" pour l'hôte qui initie la connection, et celle d'esclave pour le participant passif.

4 Le faire

4.1 Préparation

Avant de commencer à mettre en place votre système, vous devriez connaître les détails concernant le réseau. Je considère que vous avez deux firewalls, chacun protégeant un intranet, et qu'il sont tous deux connectés à l'internet. De fait, vous devriez avoir deux interfaces (au moins) par firewall. Prenez une feuille de papier et écrivez leurs adresses IP et masques de réseau. Vous aurez besoin d'une adresse IP suplémentaire par firewall pour le VPN que vous voulez mettre en place. Ces adresses devraient être extérieures à vos sous-réseaux existants. Je vous suggère d'utiliser des adresses de l'espace d'adressage "privé". Les voici :
– 10.0.0.0 - 10.255.255.255
– 172.16.0.0 - 172.31.255.255
– 192.168.0.0 - 192.168.255.255

Pour les besoins de l'exemple, voici une configuration : les deux bastions s'appellent fellini et polanski. Ils ont une interface vers l'internet (-out), une pour l'intranet (-in), et une pour le VPN (-vpn).

- fellini-out : 193.6.34.12 255.255.255.0
- fellini-in : 193.6.35.12 255.255.255.0
- fellini-vpn : 192.168.0.1 point-a-point
- polanski-out : 193.6.36.12 255.255.255.0
- polanski-in : 193.6.37.12 255.255.255.0
- polanski-vpn : 192.168.0.2 point-a-point

Voici pour les préparatifs.

4.2 Rassembler les outils

Vous aurez besoin :

- d'un firewall pour Linux
- d'un kernel
- d'une configuration minimale
- d'ipfwadm
- de fwtk
- des Outils pour le VPN
- ssh
- pppd
- sudo
- pty-redir

Version actuelles (NDT :au moment de la rédaction de cet HOWTO)

- kernel : 2.0.29 Utilisez un kernel stable, qui doit être plus récent que la version 2.0.20, à cause du bug du "ping de la mort". Au moment de la rédaction, la version 2.0.30 est le dernier kernel stable, mais il contient des bugs. Si vous voulez bénéficier du super code efficace et rapide qu'il contient, essayez un prépatch. Je trouve que le 3ème fonctionne plutôt bien.
- Un système de base : Je préfère Debian. Vous n'avez pas du tout envie d'utiliser de gros logiciels et vous n'avez jamais eu l'intention d'utiliser Sendmail (comme souvent dans le cas d'autres hôtes unix). Vous ne voulez donc absolument pas permettre l'utilisation de telnet, ftp et des commandes "r".
- ipfwadm : j'ai utilisé le 2.3.0
- fwtk : j'ai utilisé le 1.3
- ssh : >= 1.2.20. Il y a des problèmes avec le protocole sous-jacent dans les versions plus anciennes.
- pppd : j'ai utilisé la version 2.2.0 por les tests, mais je ne suis pas sûr qu'elle soit sécurisée, c'est pourquoi j'ai placé le bit setuid à 0, et utilisé sudo pour le lancer.
- sudo : 1.5.2 est la dernière version dont je sois au courant.
- pty-redir : que j'ai écrit. Essayez ftp ://ftp.vein.hu/ssa/contrib/mag/pty-redir-0.1.tar.gz. On en est à la version 0.1 mantenant. Dites-moi si vous rencontrez un quelconque problème en l'utilisant.

4.3 Compiler et installer

Compilez ou installez les outils que vous venez de rassembler. Consultez attentivement leur documentation (et le firewall-howto) pour de plus amples informations. Mantenant, nous disposons des outils.

4.4 Configurer les autres sous-systèmes

Configurez correctement les paramètres des firewalls. Vous devez autoriser les communications ssh entre les deux hôtes disposant de firewalls. Cela signifie qu'il doit exister une connexion sur le port 22 du maître vers l'esclave. Lancez sshd sur l'esclave et vérifiez que vous pouvez vous connecter. Je n'ai pas vérifié cette étape, n'hésitez pas à me communiquer les résultats que vous avez obtenus.

4.5 Configurer les comptes pour le VPN

Créez un compte sur le firewall esclave en utilisant vos outils favoris (par exemple vi, mkdir, chown, chmod). Vous pouvez aussi créer un compte sur le maître, mais je pense que vous souhaitez que la connexion se fasse au démarrage, nous nous servirons donc de votre compte root habituel. Est-ce que quelqu'un pourrait me signaler les risques qu'il y a à utiliser le compte root sur le maître ?

4.6 Générer une clé ssh pour le compte du maître

Utilisez le programme de génération de clé de ssh. Donnez un mot de passe vide pour la clé privée si vous voulez réaliser une configuration automatique du VPN.

4.7 Configurer une connexion ssh automatique pour le compte esclave

Copiez la clé publique fraîchement générée dans le compte esclave dans le fichier .ssh/authorized_keys, et configurez les droits d'accès comme indiqué ci dessous :

```
drwx------ 2 esclave esclave 1024 Apr 7 23:49 ./
drwx------ 4 esclave esclave 1024 Apr 24 14:05 ../
-rwx------ 1 esclave esclave 328 Apr 7 03:04 authorized_keys
-rw------- 1 esclave esclave 660 Apr 14 15:23 known_hosts
-rw------- 1 esclave esclave 512 Apr 21 10:03 random_seed
```

la première ligne étant ~esclave/.ssh, la seconde ~esclave.

4.8 Resserrer la sécurité ssh sur les bastions.

Ce qui se traduit par la configuration suivante dans sshd_conf :

```
PermitRootLogin no
IgnoreRhosts yes
StrictModes yes
QuietMode no
FascistLogging yes
KeepAlive yes
RhostsAuthentication no
RhostsRSAAuthentication no
RSAAuthentication yes
PasswordAuthentication no
PermitEmptyPasswords no
```

L'authentification par mot de passe étant désactivée, la connexion n'est possible qu'avec les clés autorisées. (Vous aurez bien entendu désactivé telnet et la commande 'r').

4.9 Permettre l'exécution de ppp et route sur les deux comptes.

Comme le compte maître est aussi le compte root en ce qui me concerne, Il n'y a rien eu à faire. Pour le compte esclave, les lignes suivantes apparaissent dans /etc/sudoers :

```
Cmnd_Alias VPN=/usr/sbin/pppd,/usr/local/vpn/route
esclave ALL=NOPASSWD: VPN
```

Comme vous pouvez le voir, j'utilise des scripts pour mettre en place ppp et les tables de routage sur l'hôte esclave.

4.10 Faire les scripts

Sur l'hôte maître, j'utilise un full-blown script :

```
#! /bin/sh
# skeleton      example file to build /etc/init.d/ scripts.
#               This file should be used to construct scripts for /etc/init.d.
#
#               Written by Miquel van Smoorenburg <miquels@cistron.nl>.
#               Modified for Debian GNU/Linux
#               by Ian Murdock <imurdock@gnu.ai.mit.edu>.
#
# Version:      @(#)skeleton  1.6  11-Nov-1996  miquels@cistron.nl
#

PATH=/usr/local/sbin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/bin/X11/:
PPPAPP=/home/esclave/ppp
ROUTEAPP=/home/esclave/route
PPPD=/usr/sbin/pppd
NAME=VPN
REDIR=/usr/local/bin/pty-redir
SSH=/usr/bin/ssh
MYPPPIP=192.168.0.1
TARGETIP=192.168.0.2
TARGETNET=193.6.37.0
MYNET=193.6.35.0
ESCLAVEWALL=polanski-out
ESCLAVEACC=esclave

test -f $PPPD || exit 0

set -e

case "$1" in
  start)
        echo setting up vpn
        $REDIR $SSH -o 'Batchmode yes' -t -l $ESCLAVEACC $ESCLAVEWALL sudo $PPPAPP >/tmp/device
        TTYNAME=`cat /tmp/device`
echo tty is $TTYNAME
        sleep 10s
        if [ ! -z $TTYNAME ]
```

```
        then
        $PPPD $TTYNAME ${MYPPPIP}:${TARGETIP}
        else
                echo FAILED!
                logger "vpn setup failed"
        fi
        sleep 5s
        route add -net $TARGETNET gw $TARGETIP
        $SSH -o 'Batchmode yes' -l $ESCLAVEACC $ESCLAVEWALL sudo $ROUTEAPP
    ;;
  stop)
        ps -ax | grep "ssh -t -l $ESCLAVEACC " | grep -v grep | awk '{print $1}' | xargs kill
    ;;
  *)
    # echo "Usage: /etc/init.d/$NAME {start|stop|reload}"
    echo "Usage: /etc/init.d/$NAME {start|stop}"
    exit 1
    ;;
esac

exit 0
```

L'esclave utilise un script pour la préparation du routage (/usr/local/vpn/route) :

```
#!/bin/bash
/sbin/route add -net 193.6.35.0 gw 192.168.0.1
```

et son .ppprc est tel qu'indiqué ci-dessous :

```
passive
```

5 Regardons ce qui se passe :

Le maître se connecte à l'esclave, commence pppd, et redirige tout vers un terminal pty local. Ce qui consiste en l'enchaînement suivant :
- allouer un nouveau pty
- ssh'er dans l'esclave
- lancer pppd sur l'esclave
- le maître lance pppd sur son pty local
- et met en place la table de routage sur le client.

Le temps entre en considération (ne serait-ce qu'un peu), c'est pour cela que l'on a ajouté 'sleep 10'.

6 Le faire à la main.

6.1 Se connecter

Vous avez déjà essayé de voir si ssh marche bien, n'est-ce pas ? Si l'esclave refuse de vous laisser vous connecter, lisez les fichiers logs. Peut-être y a-t-il des problèmes d'autorisation sur certains fichiers, ou avec la configuration de sshd.

6.2 Faire chauffer ppp

Connectez-vous à l'esclave, et tapez :

```
sudo /usr/sbin/pppd passive
```

Vous devriez voir les ennuis arriver à partir de ce moment. Si ca marche, c'est bien ; sinon, c'est qu'il y a des problèmes soit avec sudo, soit avec pppd. Regardez ce que les commandes ont dit, ainsi que les fichiers */etc/ppp/options* et *.ppprc* . Si tout fonctionne, ajoutez 'passive' dans .ppprc, et essayez de nouveau. Pour vous débarrasser des problèmes et continuer à travailler, appuyez sur Enter, '~' et '^Z'. Vous devriez alors avoir l'invite du maître, et faire kill %1. Regardez la partie concernant les réglages si vous voulez en savoir plus sur les caractères d'échappement.

6.3 Réunir les deux.

Bon, alors

```
ssh -l esclave polanski sudo /usr/sbin/pppd
```

devrait aussi marcher, et vous renvoyer son blabla en pleine tête.

6.4 Redirection du terminal

Essayez de tout rediriger cette fois-ci :

```
/usr/local/bin/pty-redir /usr/bin/ssh -l esclave polanski sudo /usr/sbin/pppd
```

Longue phrase, hein ? Vous êtes supposé utiliser le chemin d'accès complet dans l'exécutable ssh, du fait que le programme de redirection du pty n'autorise que cette forme pour des raisons de sécurité. Maintenant, vous disposez d'un nom de fichier spécial pour le programme. Disons que c'est */dev/ttyp0* . Vous pouvez utiliser la commande ps pour regarder ce qui s'est passé. Regardons 'p0'.

6.5 Y a-t-il quelque chose sur le dispositif ?

Essayez

```
/usr/sbin/pppd /dev/ttyp0 local 192.168.0.1:192.168.0.2
```

pour établir la connexion. Regardez la sortie de la commande ifconfig pour voir si le dispositif s'est installé, et utilisez ping pour vérifier votre réseau virtuel.

6.6 Mettre en place le routage.

Configurez les routes sur l'hôte maître, ainsi que sur l'esclave. Vous devriez maintenant être capable de lancer un ping sur un hôte d'un intranet depuis un hôte sur l'autre intranet. Mettez en place des règles additionnelles de firewall. Maintenant que vous avez le VPN, vous pouvez mettre en place les règles concernant l'interconnexion des deux intranets.

7 Réglages

7.1 Réglages de la configuration

Comme je l'ai déjà dit, cet HOWTO est avant tout un mémo rapide sur la manière dont j'ai monté un VPN. Il y a des choses dans la configuration que je n'ai pas encore essayées. Ces choses rejoindront leur place quand je les aurais essayées, ou que quelqu'un m'aura dit : "C'est comme ça que ça marche". La chose la plus importante est que la connexion qu'utilise ppp n'est pas en 8 bits. Je crois que l'on peut faire quelque chose à ce sujet avec la configuration de ssh ou celle de pty. Dans cette configuration, ssh utilise le caractère tilde (˜) comme un caractère d'échappement. Cela pourait stopper ou ralentir la communication si une séquence retour-à-la-ligne/tilde conduisait ssh à retourner une invite. Selon la documentation de ssh : <sur la plupart des systèmes, donner au caractère d'échappement la valeur "none" rendra de la session transparente, même si un tty est utilisé.> Le drapeau correspondant à cela pour ssh est '-*e*', et vous pouvez aussi le placer dans le fichier de configuration.

7.2 Bande pasante contre cycles d'horloge

Créer quelque chose, aussi virtuel soit-il, entraîne l'utilisation de ressources du monde réel. Un VPN utilise de la bande passante et des ressources de calcul. Le but étant de trouver un équilibre entre les deux. Vous pouvez faire des réglages avec le drapeau '-C' ou l'option 'CompressionLevel'. Vous pourriez essayer de trouver un autre algorithme de chiffrement, mais je ne vous le recommande pas. Notez aussi que le temps de transmission peut être allongé si vous utilisez un meilleur taux de compression. Toutes vos expériences sont les bienvenues.

8 Analyse de vulnérabilité

J'essaie de couvrir ici les trous de sécurité naissant de cette mise en oeuvre en particulier, et des VPNs en général. Tous les commentaires seront vivement appreciés.

- sudo : en fait, j'utilise sudo de manière excessive. Je crois que c'est toujours plus sûr que d'utiliser les bits setuid. C'est encore un inconvénient de Linux de n'avoir pas un contrôle d'accès plus rigoureux. On attend la compatibilité avec POSIX.6 <http ://www.xarius.demon.co.uk/software/posix6/>. Ce qui est pire, c'est qu'il y a des scripts shell qui vont être lancés avec sudo. Plutôt mauvais. Quelqu'un a une idee ?
- pppd : lui aussi lance suid root. Il peut être configuré par le .ppprc de l'utilisateur. Il se pourrait qu'il y ait de beaux dépassements de la mémoire tampon. Ligne de défense : sécurisez votre compte esclave autant que possible.
- ssh : faites attention au fait que les versions de ssh antérieures à la 1.2.20 contiennent des trous de sécurité. Pire, nous avons établi une configuration telle que lorsque le compte maître a été compromis, le compte esclave l'est lui aussi, et est grand ouvert aux attaques utilisant des programmes lancés avec sudo. C'est parce que j'ai choisi de ne pas avoir de mot de passe sur la clé secrète du maître pour permettre la configuration automatique du VPN.
- firewall : avec des règles de firewall incorrectes sur un des bastions, vous ouvrez les deux intranets. Je recommande d'utiliser le camouflage d'adresse IP (car l'installation de routes incorrectes est un peu moins évidente), et faire des contrôles très sérieux sur les interfaces VPN.