Coupe-feu pour connexion par modem avec FreeBSD

Marc Silver

marcs@draenor.org

$FreeBSD: head/fr_FR.ISO8859-1/articles/dialup-firewall/article.sgml 38826 2012-05-17 19:12:14Z hrs $

Cet article expose comment mettre en place un coupe-feu utilisant une connexion PPP par modem avec FreeBSD et IPFW, et spécifiquement l'utilisation de coupe-feux pour une connexion par modem avec adresse IP dynamique. Ce document ne couvre pas en premier lieu la configuration de votre connexion PPP.

La redistribution du code source (SGML), modifié ou non, et compilé (HTML, PostScript, etc.) est soumise aux conditions suivantes :

1. Le copyright ci-dessus, la présente liste de conditions et l'avertissement qui la suit doivent figurer dans le code source.
2. Le code source distribué sous forme compilée doit faire apparaître le copyright ci-dessus, la présente liste de conditions et l'avertissement qui la suit.

CE DOCUMENT EST FOURNI "TEL QU'EN L'ÉTAT" PAR LE PROJET DE DOCUMENTATION FRANÇAISE DE FreeBSD ET IL N'EST DONNÉ AUCUNE GARANTIE, IMPLICITE OU EXPLICITE, QUANT À SON UTILISATION COMMERCIALE, PROFESSIONNELLE OU AUTRE. LES COLLABORATEURS DU PROJET DE DOCUMENTATION FRANÇAISE DE FreeBSD NE PEUVENT EN AUCUN CAS ÊTRE TENUS POUR RESPONSABLES DE QUELQUE DOMMAGE OU PRÉJUDICE DIRECT, INDIRECT, SECONDAIRE OU ACCESSOIRE (Y COMPRIS LES PERTES FINANCIèRES DUES AU MANQUE À GAGNER, À L'INTERRUPTION D'ACTIVITÉS, OU LA PERTE D'INFORMATIONS ET AUTRES) DÉCOULANT DE L'UTILISATION DE LA DOCUMENTATION OU DE L'IMPOSSIBILITÉ D'UTILISER CELLE-CI, ET DONT L'UTILISATEUR ACCEPTE L'ENTIÈRE RESPONSABILITÉ.

Version française de Marc Fonvieille <`blackend@FreeBSD.org`>.

1. Préface

Coupe-feu pour connexion par modem avec FreeBSD

Ce document couvre le processus requis pour configurer un coupe-feu avec FreeBSD quand votre fournisseur d'accès vous assigne une adresse IP dynamique. Alors que de nombreux efforts ont été faits afin de rendre ce

document aussi instructif et correct que possible, vous êtes encouragés à envoyer vos commentaires/suggestions à l'adresse <marcs@draenor.org>.

2. Options du noyau

La première chose dont vous aurez besoin est de recompiler votre noyau. Si vous avez besoin de plus d'informations sur comment recompiler un noyau, alors le meilleur endroit pour commencer est la section de configuration du noyau du manuel (../../books/handbook/kernelconfig.html). Vous devez rajouter les options suivantes dans le fichier de configuration de votre noyau:

`options IPFIREWALL`

Intègre au noyau le code de filtrage de paquets.

`options IPFIREWALL_VERBOSE`

Envoie les paquets tracés au système de traces.

`options IPFIREWALL_VERBOSE_LIMIT=`*`100`*

Limite le nombre de paquets similaires tracés. Cela évite que votre fichier de traces soit submergé de nombreuses entrées répétées. *`100`* est une valeur raisonnable, mais vous pouvez l'ajuster en fonction de vos besoins.

`options IPDIVERT`

Autorise le *détournement* des sockets, cela sera explicité plus tard.

Il y a d'autres éléments *optionnels* que vous pouvez rajouter dans le noyau pour plus de sécurité. Ils ne sont pas requis pour avoir un filtrage de paquets qui fonctionne, mais il se peut que quelques utilisateurs un peu plus paranoïaques désirent les utiliser.

`options TCP_DROP_SYNFIN`

Cette option ignore les paquets TCP avec les indicateurs SYN et FIN activés. Cela empêche certains utilitaires tel que nmap etc. d'identifier la pile TCP/IP de la machine, mais cela rompt le support des extensions RFC1644. Cela *n'est pas* recommandé si la machine héberge un serveur web.

Ne pas redémarrer une fois que vous avez recompilé le noyau. Avec un peu de chance, nous n'aurons besoin de redémarrer qu'une fois pour achever l'installation du coupe-feu.

3. Modifier `/etc/rc.conf` pour charger le coupe-feu

Nous avons maintenant besoin de quelques modifications de `/etc/rc.conf` afin de signaler notre coupe-feu. Ajoutez simplement les lignes suivantes:

```
firewall_enable="YES"
firewall_script="/etc/firewall/fwrules"
natd_enable="YES"
natd_interface="tun0"
natd_flags="-dynamic"
```

Pour plus d'informations sur la fonction de ces éléments jetez un coup d'oeil à `/etc/defaults/rc.conf` et lisez la page de manuel rc.conf(5).

4. Désactiver la traduction d'adresse réseau de PPP

Il se peut que vous utilisiez déjà la traduction d'adresse réseau (NAT) intégrée à PPP. Si c'est le cas alors vous aurez à la désactiver, étant donné que nos exemples utilisent natd(8) pour faire la même chose.

Si vous avez déjà un ensemble d'options pour démarrer automatiquement PPP, cela doit probablement ressembler à ceci:

```
ppp_enable="YES"
ppp_mode="auto"
ppp_nat="YES"
ppp_profile="profile"
```

Si c'est le cas, vous devrez spécifiquement désactiver `ppp_nat` en vous assurant que vous avez bien la ligne `ppp_nat="NO"` dans `/etc/rc.conf`. Vous devrez également retirer les lignes `nat enable yes` ou `alias enable yes` de `/etc/ppp/ppp.conf`.

5. Le jeu de règles pour le coupe-feu

Nous avons presque terminé. Tout ce qu'il reste à faire est de définir les règles du coupe-feu et alors nous pourrons redémarrer, et notre coupe-feu devrait fonctionner. Je me suis rendu compte que chacun désirera quelque chose de légèrement différent quand il est question de son ensemble de règles. Ce que j'ai essayé de faire est d'écrire un ensemble de règles qui conviendra à la plupart des utilisateurs de modems. Vous pouvez bien évidement le modifier selon vos besoins en utilisant les règles suivantes comme fondation pour votre propre ensemble de règles. Tout d'abord commençons avec les bases du filtrage fermé. Ce que vous voulez faire est de refuser tout par défaut et ensuite n'autoriser que les choses dont vous avez vraiment besoin. Les règles devraient être ordonnées de façon à autoriser tout d'abord puis ensuite refuser. Le principe est que vous ajoutiez les règles pour vos autorisations, et ensuite tout est refusé. :)

Maintenant, créons le répertoire `/etc/firewall`. Allez dans ce répertoire et éditez le fichier `fwrules` comme nous l'avons spécifié dans `rc.conf`. S'il vous plaît, notez que vous pouvez changer le nom de ce fichier pour celui que vous désirez. Ce guide donne juste un exemple de nom de fichier.

Maintenant, jettons un coup d'oeil à cet exemple de fichier de coupe-feu, qui est minutieusement commenté.

```
# Règles du coupe-feu
# Ecrit par Marc Silver (marcs@draenor.org)
# http://draenor.org/ipfw
# Librement distribuable


# Définie la commande du coupe-feu (comme dans /etc/rc.firewall)
# pour une référence aisée. Facilite la lecture.
fwcmd="/sbin/ipfw"

# Vide les règles actuelles avant rechargement.
$fwcmd -f flush
```

```
# Détourne tous les paquets à travers l'interface tunnel
$fwcmd add divert natd all from any to any via tun0

# Autorise toutes les données de ma carte réseau et de l'hôte local.
# Soyez sûr de changer votre carte réseau (la mienne était fxp0) avant
# de redémarrer.  :)
$fwcmd add allow ip from any to any via lo0
$fwcmd add allow ip from any to any via fxp0

# Autorise toute les connexions dont je suis l'initiateur.
$fwcmd add allow tcp from any to any out xmit tun0 setup

# Une fois les connexions établies, les autorise à rester ouvertes.
$fwcmd add allow tcp from any to any via tun0 established

# Tous le monde sur internet est autorisé à se connecter aux services
# suivants sur la machine.  Cet exemple autorise spécifiquement les
# connexions à ssh et apache.
$fwcmd add allow tcp from any to any 80 setup
$fwcmd add allow tcp from any to any 22 setup

# Ceci envoie un RESET à tous les paquets ident.
$fwcmd add reset log tcp from any to any 113 in recv tun0

# Autorise les requettes DNS sortantes SEULEMENT vers les serveurs
# spécifiés.
$fwcmd add allow udp from any to x.x.x.x 53 out xmit tun0

# Autorise leur retour avec les réponses...  :)
$fwcmd add allow udp from x.x.x.x 53 to any in recv tun0

# Autorise l'ICMP (pour permettre à ping et traceroute de fonctionner).
# Vous pouvez peut-être désirer désactiver ceci, mais je pense que cela
# répond à mes besoins de les conserver ainsi.
$fwcmd add allow icmp from any to any

# Bloque tout le reste.
$fwcmd add deny log ip from any to any
```

Vous disposez désormais d'un coupe-feu tout à fait fonctionnel qui autorisera les connexions sur les ports 80 et 22 et tracera tout autre tentative de connexion. Maintenant, vous devriez être en mesure de redémarrer sans risques et votre coupe-feu devrait se lancer sans problèmes. Si vous trouvez une quelconque erreur ou expérimentez des problèmes, ou que vous avez des suggestions pour améliorer ce document, s'il vous plaît écrivez-moi.

6. Questions

1. Pourquoi utilisez-vous natd(8) et ipfw(8) alors que vous pourriez utiliser les filtres intégrés à ppp(8)?

Je serais honnête et dirais qu'il n'y a aucune raison définitive pour que j'utilise `ipfw` et `natd` plutôt que les filtres intégrés à `ppp`. D'après les discussions que j'ai eu avec de nombreuses personnes le consensus semble être qu'`ipfw`

est certainement plus puissant et configurable que les filtres `ppp`, mais ce qu'il apporte dans la fonctionnalité il le perd en facilité d'utilisation. Une des raisons de mon utilisation est que je préfère que le filtrage de paquets soit fait au niveau du noyau plutôt que par un programme utilisateur.

2. J'obtiens des messages du type `limit 100 reached on entry 2800` et après cela je ne vois plus jamais de refus dans mes traces. Mon coupe-feu fonctionne-t-il toujours?

Cela signifie simplement que le nombre maximal de traces pour la règle a été atteint. La règle fonctionne toujours, mais elle n'enregistrera plus de trace jusqu'au moment ou vous réinitialiserez les compteurs de traces. Vous pouvez réinitialiser les compteurs de traces avec la commande `ipfw resetlog`. Alternativement, vous pouvez augmenter la limite de trace dans la configuration de votre noyau avec l'option `IPFIREWALL_VERBOSE_LIMIT` comme décrit précédemment. Vous pouvez également changer cette limite (sans recompiler votre noyau ou avoir à redémarrer) en utilisant la valeur sysctl(8) net.inet.ip.fw.verbose_limit.

3. Si j'utilise des adresses privées en interne, comme dans la plage 192.168.0.0, pourrais-je ajouter une commande comme `$fwcmd add deny all from any to 192.168.0.0:255.255.0.0 via tun0` aux règles du coupe-feu pour prévenir les tentatives externes de connexions vers les machines internes?

Une réponse simple est non. La raison de cela est que `natd` effectue la traduction d'adresse pour *tout* ce qui sera détourné à travers le périphérique `tun0`. En ce qui concerne les paquets entrant, ces derniers ne parleront qu'à l'adresse IP dynamiquement assignée et *non pas* au réseau interne. Notez que cependant vous pouvez ajouter une règle comme `$fwcmd add deny all from 192.168.0.4:255.255.0.0 to any via tun0` qui limiterait un hôte de votre réseau interne de sortir via le coupe-feu.

4. Il doit y avoir quelque chose d'erroné. J'ai suivi vos instructions à la lettre et maintenant tout est bloqué.

Ce guide suppose que vous utilisez *userland-ppp* aussi le jeu de règle donné intervient sur l'interface `tun0`, qui correspond à la première connexion établie avec ppp(8) (a.k.a. *user-ppp*). Les connexions supplémentaires utiliseront `tun1`, `tun2` et ainsi de suite.

Vous devriez également noter que pppd(8) utilise à la place l'interface `ppp0`, donc si vous établissez une connexion avec pppd(8) vous devez remplacer `tun0` par `ppp0`. Une façon rapide d'éditer les règles du coupe-feu pour refléter ce changement est présentée ci-dessous. Le jeu de règles original est sauvegardé sous `fwrules_tun0`.

```
% cd /etc/firewall
/etc/firewall% su
Password:
/etc/firewall# mv fwrules fwrules_tun0
/etc/firewall# cat fwrules_tun0 | sed s/tun0/ppp0/g > fwrules
```

Pour savoir si vous utilisez actuellement ppp(8) ou pppd(8) vous pouvez examiner la sortie d'ifconfig(8) une fois que la connexion est établie. E.g., pour une connexion faite par pppd(8) vous verriez quelque chose comme ceci (on ne montre que les lignes importantes):

```
% ifconfig
(skipped...)
ppp0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1524
        inet xxx.xxx.xxx.xxx --> xxx.xxx.xxx.xxx netmask 0xff000000
(skipped...)
```

D'autre part, pour une connexion faite avec ppp(8) (*user-ppp*) vous devriez voir quelque chose de similaire à:

```
% ifconfig
(skipped...)
ppp0: flags=8010<POINTOPOINT,MULTICAST> mtu 1500
(skipped...)
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1524
        (IPv6 stuff skipped...)
        inet xxx.xxx.xxx.xxx --> xxx.xxx.xxx.xxx netmask 0xffffff00
        Opened by PID xxxxx
(skipped...)
```