hosts.allow, hosts.deny

Ces deux fichiers dans **/etc** sont le lieu commun pour stocker les règles qui permettront d'autoriser des hôtes à se connecter à des services sur votre machine.

Alors qu'un pare-feu peut être considéré comme cachant une porte, ces fichiers contrôlent qui est autorisé à ouvrir la porte.

Utilisé en combinaison, ces deux fichiers peuvent être utilisés pour créer soit:

- une ouverture par défaut avec des exclusions (liste d'interdiction)
- une fermeture par défaut avec des exclusions (liste d'invités)

Par défaut Slackware est livré avec ces deux fichiers vide, cela signifie que la porte est déverrouillée et que personne n'est interdit.

Ce document vous aidera à changer cet état par défaut par une configuration plus sécurisée.

Contenu

1. Faites en sorte d'avoir une clé avec vous
2. Vérouiller la porte
3. Ecrire la liste des invités
    1. Ajouter un deuxième hôte
    2. Ajouter plusieurs hôtes
    3. Ajouter d'autres services
    4. Parlez vous!
4. Notes
5. Voir aussi

Faites en sorte d'avoir une clé avec vous

Si vous vous connectez à la machine par ssh, vous aurez envie de vous assurer que les connexions suivantes sont autorisées. Si la machine devant laquelle vous êtes assis est 192.168.0.10, éditer **/etc/hosts.allow** et ajoutez:

```
sshd:   192.168.0.10
```

Si vous utilisez un DNS, vous pouvez également référer à votre Machine par son nom, par exemple:

```
sshd:   wibble.mynet.invalid
```

Verrouiller la porte

Cela se fait simplement en éditant **/etc/hosts.deny** et en ajoutant la ligne:

```
All:    All
```

Les connexions qui sont en cours d'utilisation seront toujours utilisables, seuls les nouvelles connexions via SSH depuis 192.168.0.10 seront autorisés.

Ecrire la liste des invités

Ajouter un deuxième hôte

Nous avons déjà permis les connexions au serveur sshd uniquement depuis 192.168.0.10, si nous voulons permettre à un second hôte de se connecter, c'est aussi simple que d'ajouter :

```
sshd:   192.168.0.10 192.168.0.11
```

ou

```
sshd:   wibble.mynet.invalid wobble.mynet.invalid
```

Vous pouvez avoir un espace entre eux ou ajouter une virgule pour plus de clarté.

Ajouter plusieurs hôtes

Il est possible de permettre à des plages d'adresses de se connecter soit en raccourcissant l'adresse, soit en utilisant un masque réseau.

```
sshd:   192.168.0.
```

```
sshd:   192.168.0.0/255.255.255.0
```

Les deux auront le même effet.

Vous pouvez permettre à tous ceux d'un nom de domaine de se connecter, par exemple:

```
sshd:   .mynet.invalid
```

Ajouter d'autres services

Dans l'ensemble, le nom du service **VERS** lequel vous vous connectez comme; par exemple sshd, in.telnetd, vstfpd, proftpd; doit être placé dans hosts.allow, mais comme avec toutes choses il y a des exceptions... NFS, avec NFS nous faisons des règles pour les services **DEPUIS** lesquels nous autorisons les connexions.

Si, par exemple, la machine nous souhaitons sécuriser est un serveur NFS et que vous voulez le monter sur 192.168.0.10, nous mettrions dans **/etc/hosts.allow**

```
portmap:    192.168.0.10
mountd:     192.168.0.10
```

De même, de façon similaire, si vous voulez qu'il monte un export NFS, nous devrions mettre dans l'adresse du nfsd que nous voulons monter:

```
portmap:    192.168.0.10
nfsd:       192.168.0.10
```

Parlez vous!

Parfois, ce n'est pas une mauvaise idée, par exemple pour le processus rndc pour le rechargement de *bind* pourrait être sur la même machine exécutant *named*; dans ce cas, nous voulons autoriser les connexions à partir de cette même machine.

```
rndc:   127.0.0.1
```

Encore une fois, il est à noter que c'est le nom du proccess avec lequel nous voulons communiquer, pas le nom du processus que nous écoutons.

Notes

Cela ne couvre pas toutes les variations grammaticales de ces deux fichiers ni n'évoque tous les services qui ouvrent des ports mais devrait, espérons-le, vous donner un avant-goût de ce qui peut être fait.

Voir aussi

man (5) hosts_access

Sources

- Traduit de l'anglais par — *Cedric M. 2015/09/09 13:00*

howtos, security, slackware allversions, inetd, translator cedric