Multiplexage sous OpenSSH

Présentation pour le groupe SUR (Sécurité Unix et Réseaux)
10/01/2006

Saâd Kadhi <saad.kadhi@hapsis.fr>

Agenda

- Qu'est-ce que c'est ?
- Pourquoi est-ce important ?
- Comment l'utiliser ?
- Conclusions, Références, Remerciements

Qu'est-ce que c'est ?

(à part une bonne résolution 2006)

Multiplexage de sessions

- Fonctionnalité “relativement” nouvelle, introduite par OpenSSH 3.9 (p), version disponible depuis le 17 Août 2004
- Stabilisée dans les versions 4.0(p) et 4.1(p)
- Nettement améliorée dans la version 4.2(p)

En quoi ça consiste ?

- Etablir une session SSH qui va véhiculer toutes les suivantes
  - session dite “maître”
- Les autres sessions utilisent une socket ouverte par la session “maître”
  - sessions interactives, SCP, SFTP, X11 forwarding, Agent Forwarding

Le client à l'honneur

- Tout se passe au niveau du client SSH
  - il faut donc un client OpenSSH $\geqslant$ 3.9 (p), de préférence 4.2(p)
- Aucune configuration sur le serveur

Pourquoi est-ce important ?

Authentification

- L'utilisateur s'authentifie une seule fois, lors de l'établissement de la session "maître"
- Toutes les sessions suivantes bénéficiant de la session "maître" se font sans authentification
  - d'où un gain en temps et en simplicité
- Attention ! Le serveur ne voit qu'une seule authentification

Performances

- L'établissement des clés symétriques de sessions via les échanges à clés publiques est consommateur en ressources
- ...surtout depuis le doublement de la taille par défaut des clés RSA/DSA introduit dans la version 4.2(p) : de 1024 à 2048 bits
- Avec le multiplexage de sessions, l'échange à clés publiques n'est effectué qu'une seule fois

... Et pour les geeks

- Qui dit augmentation de performances dit établissement de sessions beaucoup plus rapide
- L'auto-completion proposée par certains shells (bash, zsh,...) est plus rapide

Sécurité

- La session “maître” ouvre une socket locale protégée par les permissions du système de fichiers
- La session “maître” dure généralement plus longtemps qu’une session “normale”
- Un pirate a plus de temps pour essayer d’en percer le secret
- La sécurité de toutes les sessions multiplexées reposent sur la sécurité de la session “maître”

Limitations

- Le multiplexage de sessions n'est possible qu'avec le protocole SSHv2
- 9 sessions maximum par socket (session "maître" non incluse)

Comment l'utiliser ?

3 étapes

- Configurer le client SSH
- Créer la session “maître”
- Etablir les sessions multiplexées

Configuration du client

- Il existe plusieurs façons de configurer le client
  - selon si on veut utiliser systèmatiquement ou non le multiplexage
- La façon la plus simple reste le mode “opportunistic” introduit par la version 4.2(p)
  - multiplexage automatique

ControlMaster, ControlPath

- Tout se passe au niveau de ssh_config (ou ~/.ssh/config)
- Deux directives : ControlMaster et ControlPath
  - ControlMaster active le multiplexage
  - ControlPath indique où créer/trouver la socket de multiplexage

Exemple

Pour le mode “opportunistic”, il suffit de rajouter au fichier de configuration les trois lignes suivantes :

```
Host *
   ControlPath ~/.ssh/ctl-%r-%h-%p
   ControlMaster auto
```

Explications

- La valeur de ControlPath permet de créer des sockets comportant le nom de l'utilisateur distant (%r), le nom de la machine distante (%h), et le nom du port d'écoute du serveur (%p)
- La valeur de ControlMaster indique au client d'utiliser une session "maître" si elle existe, sinon en créer une

Session "maître" - 1

- Pour créer une session "maître", il suffit d'établir une session comme d'habitude si on utilise le mode "opportunistic" :

```
$ ssh <remotehost>
```
(session interactive)

```
$ ssh -Nnf <remotehost>
```
(session en arrière-plan)

- Sinon, il faut utiliser l'option M (Master mode) :

```
$ ssh -M <remotehost>
```

Session "maître" - 2

Une fois la session "maître" créée :

```
$ ls -l ~/.ssh/ctl-*
srw-------   1 saad  saad  0 Jan  9 08:21 /Users/saad/.ssh/
ctl-saad-orphu.test.net-22

srw-------   1 saad  saad  0 Jan  9 08:38 /Users/saad/.ssh/
ctl-saad-mahnmut.abc.org-22
```

Pour vérifier son état ou la terminer :

```
$ ssh -O [check|exit] <remotehost>
```

Sessions multiplexées

- Les sessions multiplexées s'établissent de la même manière qu'une session "normale"
- Coté serveur, on voit :

```
5333 ?        S      0:00 sshd: saad@pts/2,pts/13
```

Conclusion, Références,

Remerciements

Conclusion

- Le multiplexage de sessions est une raison suffisante pour mettre à jour les clients en 4.2(p)
- Simplification et performances
- Mais attention aux problèmes potentiels de sécurité (pure spéculation pour le moment ?)

Références

- http://www.openssh.com/fr/
- http://www.openssh.com/txt/release-4.2
- http://www.undeadly.org/cgi?action=article&sid=20051222152522
- “Les chantiers OpenBSD” par Guillaume Arcas et Saâd Kadhi, MISC 21.

Remerciements

- HAPSIS, mon employeur
  - http://www.hapsis.fr/
- Jack Johnson et N'Gou Bagayoko
  - pour la musique ;-)
- Et vous tous
  - pour m'avoir écouté !