CheatSheet

Faire du Port Forwarding avec SSH

Les variables de cette page :

- **$ip_ecoute** : IP d'écoute pour le port (falcutatif, par défaut localhost)
- **$port_entrer** : Port d'entré pour joindre le $port_local
- **$host** : host que l'émetteur du port va joindre
- **$port_local** : Port d'écoute du port à joindre

En gros, une fois le tunnel mis en place, l'host ayant accès au port devra joindre **$ip_ecoute** sur le **$port_entre** afin de joindre **$host** sur le **$port_local**

Forward un port local vers un hôte distant

Théorie

Ce mode permet de donner accès à un host distant l'accès a un port d'un host qu'il a accès

```
# ssh -R $ip_ecoute:$port_entrer:$host:$port_local root@hostname
```

Pratique

Je veux que quelqu'un puisse se connecter à ma machine par SSH mais je ne veux/peux pas modifier mon firewall pour faire du NAT sur mon port 22 alors que lui par contre l'a fait.

Je vais donc forward mon port SSH (22) dans un tunnel SSH grâce a la commande :

```
# ssh -R 2222:localhost:22 root@XXX.XXX.XXX.XXX
```

Il devra donc se connecter sur le port 2222 pour joindre ma machine sur le port 22

Forward d'un port distant sur la machine local

Théorie

Cela permet d'accéder à un port distant via la machine distante

```
# ssh -L $port_entrer:$host:$port_local root@hostname
```

Pratique

Je souhaite accéder a un serveur par RDP mais pas de NAT n'est fait vers lui, mais il y a du NAT vers le SSH d'un serveur linux qui est dans le même réseau (10.10.0.0/24), je vais donc faire du RDP vers le serveur Windows en passant dans un tunnel SSH grâce à la commande :

```
# ssh -L 3389:10.10.0.2:3389 root@XXX.XXX.XXX.XXX
```

Il faudra que je me connecte à localhost pour me connecter au serveur Windows

10.10.0.2 étant l'IP du serveur Windows

Forward dynamique via proxy SOCKS

Théorie

Cela permet de permettre de faire passer l'intégralité du trafic d'une app ou de la machine via un proxy SOCKS.

```
# ssh -D $host:$port_local root@hostname
```

Pratique

Je souhaite que tout le trafic de mon firefox passe par cette session SSH.
Je lance donc la session :

```
# ssh -D 5999 root@XXX.XXX.XXX.XXX
```

Puis de lancer mon Chrome avec la configuration adéquate :

```
# google-chrome --proxy-server="socks5://127.0.0.1:5999"
```

 Le DNS ne passe pas a travers le proxy

Notes Importantes

Certains port nécessite que l'utilisateur distant soit root pour pouvoir créer le port forwarding

L' argument **-N** permet d'indiquer qu'aucune commande ne sera exécuter sur l'host distant et **-f** met la session SSH en arrière plan. Ces arguments sont utiles si vous voulais lancer ces commandes en arrière plan (au démarrage par exemple)

Manipulation de la session SSH en cours

Il est parfois nécessaire ou plus simple de manipuler la session SSH active, pour cela il est possible d'envoyer une combinaison de touche afin d'effectuer une action. Elle doit toujours être précédé d'un caractère d'échapement (après un appui sur la touche entrée).

Voici les combinaisons possibles :

Commande	Description
~?	Commande d'aide, donc toutes ces commandes
~.	Kill la session de manière forcé
~B	Envoi un signal BREAK au système distant
~R	Renégociation des clés (uniquement sur SSH protocol 2)
~^Z	Suspendre la session
~#	Liste les sessions forwardés active
~&	background la session
~~	Envoi le carectère ~
~C	Ouvre un interface de commande pour SSH

Les commandes disponibles sont :

Commande	Description
-L $ip_ecoute:$port_entre:$host:$port_local	Request local forward
-R $ip_ecoute:$port_entrer:$host:$port_local	Request remote forward
-D $ip_ecoute:$port_entrer	Request dynamic forward
-KL $ip_ecoute:$port_entrer	Cancel local forward
-KR $ip_ecoute:$port_entrer	Cancel remote forward
-KD $ip_ecoute:$port_entrer	Cancel dynamic forward