Configuration d'une connexion VNC over ssh

Author : pierrehirel

VNC est un moyen courant pour accéder à un bureau à distance. Nous verrons ici comment l'utiliser de manière sécurisée à travers un tunnel ssh.

VNC est en effet bien pratique : parmi ses avantages on peut citer son coût en bande passante relativement faible (comparé à l'affichage de fenêtres X via ssh par exemple), sa relative facilité de mise en œuvre, et son interopérabilité puisque serveurs et clients VNC sont disponibles pour Windows, Mac et Linux, ce qui permet de contrôler des machines à distance quels que soient les systèmes d'exploitation du serveur et du client.

Mais un gros point noir de VNC est qu'il fait partie du top 10 des les protocoles les plus fréquemment attaqués. Ainsi, si un port 5901 ouvert sur votre serveur peut vous sembler pratique, il s'agit également d'une vulnérabilité critique. Nous verrons qu'il est possible de conserver les fonctionnalités de VNC, tout en sécurisant la connexion en utilisant un tunnel ssh.

1. Installer le nécessaire sur le PC serveur et le PC client

Installer *vncserver* et *openssh-server* sur le serveur ; et installer un client VNC sur le PC personnel (le « client »). Je ne m'étendrai pas sur cette étape, le nécessaire étant proposé dans les dépôts des distributions.

Il est à noter que seule cette étape requiert les droits d'administrateur. Par la suite, le serveur VNC est exécuté en tant que simple utilisateur ; certains tutos indiquant qu'il faut exécuter le serveur en *root* sont erronés.

2. Lancer le serveur VNC

Exécuter le serveur VNC sur le serveur, en y entrant (en tant qu'utilisateur, et non en tant que root !) la commande :

```
server:~$ vncserver
```

À la première exécution, le serveur VNC vous demandera quel mot de passe utiliser. Comme de coutume, utilisez un mot de passe complexe que vous n'utilisez nulle part ailleurs. Ce mot de passe est sauvegardé (de manière cryptée) dans le fichier *~/.vnc/passwd* du serveur ; si vous l'oubliez, supprimez simplement ce fichier.

Le serveur VNC ainsi créé écoute sur le port 5901 du serveur, auquel le client devra se connecter.

3. Se connecter au serveur

La commande suivante est magique, tapez-la dans une invite de commande sur le client. Elle permet en une ligne de créer un tunnel ssh, et de l'utiliser ensuite pour se connecter au serveur VNC :

```
client:~$ ssh -f -L 25901:127.0.0.1:5901 -p 2222 [user]@[server-address] sleep 10; vncviewer 127.0.0.1:25901
```

Analysons cette commande :

- **-L 25901:127.0.0.1:5901** indique de créer un tunnel ssh entre un port quelconque (ici 25901, peut être remplacé par n'importe quel port disponible) du client, i.e. de la machine locale (localhost ou 127.0.0.1), vers le port VNC (5901) du serveur ;
- **-f** indique à ssh de tourner en tâche de fond, c'est nécessaire car on veut exécuter vncviewer en premier plan ;

- **-p 2222** permet d'indiquer un port alternatif pour la connexion ssh (ici le port 2222 est utilisé) ; cette option dépend de la configuration du serveur (par défaut le port ssh est le 22) ;
- **[user]@[server-address]** sont les informations pour se connecter au serveur, bien sûr **[user]** est à remplacer par votre nom d'utilisateur et **[server-address]** par l'adresse IP du serveur (ou son URL) ;
- **sleep 10** indique au tunnel ssh de s'auto-fermer après 10 secondes en cas d'inactivité ; *vncviewer* utilise ce tunnel, donc tant que vncviewer tourne le tunnel reste ouvert ; à la fermeture de *vncviewer* le tunnel se ferme automatiquement. Cette option est recommandée par sécurité et pour des raisons pratiques (c'est plus simple que de devoir effectuer un brutal *killall ssh* comme le recommandent certains tutos) ;
- **vncviewer 127.0.0.1:25901** lance le client VNC sur la machine locale (127.0.0.1) en lui indiquant de se connecter au port 25901 ; comme celui-ci est lié au serveur par le tunnel ssh, cela va en fait le connecter au port 5901 du serveur.

Pour résumer : le client VNC se connecte au port local 25901, qui est forwardé via le tunnel ssh (et le port ssh du serveur, ici 2222) vers le port 5901 du serveur, que le serveur VNC écoute. Enfantin non ?

À la connexion, deux mots de passe vous sont alors demandés : premièrement le mot de passe pour l'accès ssh au serveur ; et ensuite le mot de passe pour accéder au serveur VNC.

4. Le bug du clavier

Sur certaines configurations il est possible que le clavier ne soit pas correctement reconnu par le serveur distant : toutes les lettres sont mélangées et semblent être aléatoires. Je ne sais pas exactement d'où vient ce bug, mais chez moi la méthode suivante a fonctionné (à répéter pour tous les utilisateurs du serveur, si besoin). Il s'agit d'ajouter, dans le fichier *~/.vnc/xstartup* du serveur, la ligne suivante juste avant la commande de lancement de session (par exemple juste avant */etc/X11/Xsession* ou *exec gnome-session*, selon la config de votre serveur) :

~/.vnc/xstartup

```
export XKL_XMODMAP_DISABLE=1
```

5. Plusieurs sessions VNC

Vous constaterez qu'à l'exécution de *vncserver*, le serveur VNC attribue un numéro à la session : il l'a appelée **:1**. Si vous exécutez *vncserver* une seconde, puis une troisième fois, les numéros attribués seront **:2**, **:3**, et ainsi de suite. Le même utilisateur peut ainsi lancer plusieurs serveurs VNC, et plusieurs utilisateurs différents peuvent lancer leurs propres sessions graphiques sur le serveur.

Chacun de ces serveurs VNC écoute sur un port différent, portant le numéro 5900 + numéro de l'instance. Nous

avons vu plus haut qu'il fallait créer un tunnel vers le port 5901 pour se connecter à la première instance. De manière logique, il faudra employer les ports 5902, 5903, etc. pour les instances suivantes, ce qui donne par exemple :

client:~$ ssh -f -L 26000:127.0.0.1:5902 -p 2222 [user]@[server-address] sleep 10; vncviewer 127.0.0.1:26000

Notez que si on utilisait simplement VNC, il faudrait ouvrir les ports 5901, 5902, etc. pour avoir accès à toutes ces instances. Ici, puisque tout transite par des tunnels ssh, nul besoin d'ouvrir de port supplémentaire, le port ssh suffit. Pour chaque nouvel instance un nouveau tunnel ssh est créé, et il n'y a (virtuellement) pas de limite dans le nombre d'instances VNC possibles.

6. Encore plus de sécurité

Cette méthode est déjà beaucoup plus sécurisée que l'utilisation d'une connexion VNC simple, de plus elle ne requiert pas d'ouvrir de port supplémentaire : tout passe par le port ssh (port 2222 dans l'exemple). Donc si vous aviez ouvert le port 5901 sur votre serveur et/ou routeur, vous pouvez le fermer.

Ensuite, on remarque aussi que la commande utilisée est « vncviewer 127.0.0.1:25901 ». Autrement dit le client VNC ne se connecte pas à une machine distante, mais bien à la machine locale (localhost), la redirection du flux étant assurée par le port. De son côté, le serveur VNC a aussi l'impression que la connexion vient de lui-même, de localhost. Donc pour renforcer la sécurité, il est possible d'ajouter une entrée dans le pare-feu *iptables* pour que les connexions au serveur VNC ne soient possibles que depuis le localhost.

Enfin, laisser tourner le serveur VNC est utile si vous souhaitez que des applications graphiques continuent de fonctionner sur le serveur après que vous vous êtes déconnecté. Mais si ce n'est pas le cas, il est recommandé de fermer le serveur VNC lorsque vous avez fini de l'utiliser, en entrant sur le serveur la commande :

server:~$ vncserver -kill :1

où **:1** est le numéro de session graphique à fermer (remplacez ce numéro si vous avez lancé plusieurs sessions VNC).