

# Rapport de Mini-Projet Linux

ING2 2013-2014

Atallah Sami – Plouvier Julien

27 avril 2014

## Résumé

Le présent rapport vise à rendre compte de nos travaux de documentation et d'approfondissement personnel quant à la découverte des fonctionnalités de Linux. Nous nous sommes intéressés au contrôle d'un ordinateur à distance, à l'aide du logiciel **TightVNC**.

## Table des matières

<b>1</b>	<b>Objectifs des travaux pratiques</b>	<b>2</b>
1.1	Choix stratégiques . . . . .	2
<b>2</b>	<b>Recherches et documentation associées</b>	<b>2</b>
2.1	SSH, <i>Secure Shell</i> . . . . .	2
2.2	VNC, <i>Virtual Network Computing</i> . . . . .	3
2.3	Le logiciel TightVNC . . . . .	3
<b>3</b>	<b>Manipulation et paramétrages</b>	<b>4</b>
3.1	Création d'un serveur . . . . .	4
3.2	Chiffrement des données . . . . .	5
3.3	Connexion à distance . . . . .	5
<b>4</b>	<b>Sources de documentation</b>	<b>6</b>

# 1 Objectifs des travaux pratiques

Nous avons cherché à réaliser une manipulation utile, offrant diverses perspectives dans le cadre de nos réalisations futures ou d'un usage personnel. Nous nous sommes alors intéressés au **contrôle d'un ordinateur à distance**, évoqué en amphithéâtre par Clément Duhart lorsqu'il nous a présenté les différents bureaux accessibles depuis sa machine.

Outre le fait de pouvoir accéder depuis n'importe quel endroit à la bonne vieille unité centrale familiale, ce projet offre des possibilités de création d'un **multi-desktop**, un environnement multi-utilisateurs relié à une seule machine sans qu'elle ne soit reliée à un écran.

Cela peut s'avérer particulièrement pratique pour utiliser à distance les capacités d'une machine puissante à partir d'un petit appareil.

## 1.1 Choix stratégiques

Nos recherches ont révélé qu'il y avait une multitude de manières différentes de lancer un bureau à distance à partir de Linux. Le **X Window System** par exemple, environnement graphique fenêtré gérant l'interaction homme-machine via l'écran, la souris et le clavier de certains ordinateurs en réseau, a lui-même été conçu pour fonctionner à distance. Mais son protocole se révèle particulièrement inefficace à moins d'être connecté au même réseau que le serveur.

Une alternative intéressante est **FreeNX**, permettant d'obtenir une réponse beaucoup plus rapide du bureau contrôlé à distance. Mais son installation est particulièrement compliquée, et la technique d'installation est différente selon les distributions.

À mi-chemin entre les deux méthodes précédentes, nous avons **VNC**, alias *Virtual Network Computing*, dont le fonctionnement général consiste à découper le bureau en petites entités graphiques, comprimées et envoyées en masse. Il nécessite des techniques d'encodage pour réduire la largeur de bande-passante utilisée. Néanmoins, son protocole efficace et simple d'utilisation nous a conduits à l'employer.

# 2 Recherches et documentation associées

## 2.1 SSH, *Secure Shell*

Le **Secure Shell** est un protocole permettant d'établir des connexions sécurisées (chiffrées) entre un serveur et un client SSH. L'installation d'un serveur SSH offre aux utilisateurs la possibilité d'accéder à distance à une machine, en renseignant simplement leur login et leur mot de passe (ou avec un mécanisme de clefs). Un pirate peut donc tout à fait essayer d'avoir un compte sur le système (pour accéder à des fichiers sur le système ou pour utiliser le système comme une passerelle pour attaquer d'autres systèmes), en essayant une multitude de mots de passes différents pour un même login (il peut le faire de manière automatique en s'aidant d'un dictionnaire électronique). On appelle ça une attaque en *force brute*.

Pour garder un système sécurisé après l'installation d'un serveur SSH, il faut alors :

- avoir un **serveur SSH à jour** au niveau de la sécurité ;
- choisir pour chaque utilisateur un mot de passe suffisamment compliqué pour résister à une attaque en force brute ;
- surveiller l'historique des connexions en lisant régulièrement le fichier de log `/var/log/auth.log`.

Dans l'usage le plus classique, SSH permet :

- de réaliser un accès au **shell** à distance, ce qui permet de réaliser des opérations sur un appareil à distance ;
- de visualiser graphiquement le bureau de la machine distante ;
- de transférer des fichiers en ligne de commande.

## 2.2 VNC, *Virtual Network Computing*

Virtual Network Computing, ou *Informatique Virtuelle en Réseau*, est un système de visualisation et de contrôle de l'environnement de bureau d'un ordinateur distant. Il transmet à l'ordinateur distant les informations de saisie du clavier et de la souris à partir de l'ordinateur actuellement utilisé. L'ordinateur à distance doit pour ce faire posséder un logiciel serveur VNC à travers un réseau informatique. VNC utilise le protocole Remote Frame Buffer, passant par défaut par le port 5900 UDP/TCP.

Dans le protocole RFB, l'application qui s'exécute sur la machine devant laquelle se trouve l'utilisateur (contenant l'affichage, le clavier et le pointeur) est appelée **client**. L'application qui tourne sur la machine où se trouve le tampon de trame (qui exécute le système de fenêtrage et les applications que l'utilisateur commande à distance) est appelée **serveur**. Ainsi, KRDC est le client de KDE pour le protocole RFB.

*"VNC est indépendant du système d'exploitation ; un client VNC installé sur n'importe quel système d'exploitation peut se connecter à un serveur VNC installé sur un système d'exploitation différent. Il existe des clients et des serveurs VNC pour la plupart des systèmes d'exploitation. Plusieurs clients peuvent se connecter en même temps à un unique serveur VNC."*

Ceci est particulièrement pratique quand on sait que la plupart des téléphones portables, tablettes et netbooks possèdent un client VNC.

VNC possède de vastes domaines d'utilisation, parmi lesquels l'administration et la maintenance de systèmes ou logiciels ne permettant que des contrôles graphiques et demandant l'utilisation de la souris ou bien encore la visualisation distante d'applications diverses et variées.

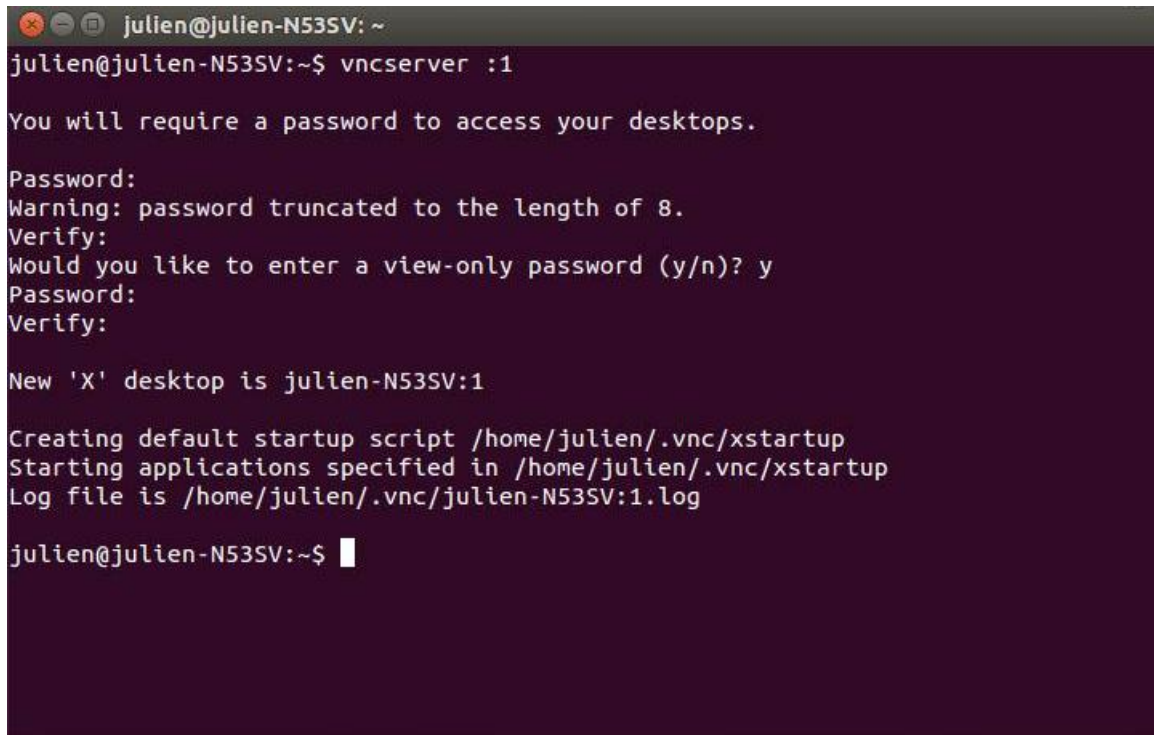
## 2.3 Le logiciel TightVNC

TightVNC est un logiciel offrant la possibilité de prendre le contrôle à distance d'un autre ordinateur (**le serveur**) en utilisant un réseau, qui peut être internet. Le serveur est présenté dans une fenêtre sur le poste client, ce qu'on appelle une **présentation graphique**, contrairement aux manipulations ne faisant intervenir qu'un mode console. Plusieurs personnes peuvent voir ce qui se passe sur un autre ordinateur sur lequel s'exécute le logiciel serveur. Pour notre part, nous utiliserons **VNCViewer**.

### 3 Manipulation et paramétrages

Dans cette section, on fera (enfin) référence aux lignes de commande entrées afin de réaliser le contrôle d'un ordinateur à distance de façon sécurisée.

#### 3.1 Création d'un serveur



```
julien@julien-N53SV: ~  
julien@julien-N53SV:~$ vncserver :1  
  
You will require a password to access your desktops.  
  
Password:  
Warning: password truncated to the length of 8.  
Verify:  
Would you like to enter a view-only password (y/n)? y  
Password:  
Verify:  
  
New 'X' desktop is julien-N53SV:1  
  
Creating default startup script /home/julien/.vnc/xstartup  
Starting applications specified in /home/julien/.vnc/xstartup  
Log file is /home/julien/.vnc/julien-N53SV:1.log  
  
julien@julien-N53SV:~$
```

Une fois TightVNC installé, on entre la commande **vncserver :1** où le **:1** est un indice se rapportant à l'ouverture d'un deuxième gestionnaire d'environnement graphique fenêtré sur l'ordinateur (le premier étant la gestion graphique actuelle de l'ordinateur). Le serveur est alors automatiquement relié à ce second gestionnaire.

N.B : Réaliser la même manipulation sur un serveur fonctionnant **uniquement** en mode console aurait donc pu se faire avec **vncserver :0**.

Nous avons à présent un serveur fonctionnel sur la machine **serveur** (la machine à contrôler), il faut maintenant s'y connecter grâce à une machine différente, la machine **client** (la machine qui contrôle).

## 3.2 Chiffrement des données

Avant toute connexion, il faut, **depuis la machine client s'assurer que la connexion est sécurisée**. Afin de réaliser le chiffrement des données via le Secure Shell, nous avons entré la ligne de commande suivante dans un premier terminal de la machine client :

```
sudo ssh -L 5901 :localhost :5900 -XC julien@52.552.36.25
```

- **sudo** : entrée de commande avec les droits de superutilisateur
- **ssh** : cette commande permet de créer un tunnel de connexion afin de **chiffrer les données**
- **-L** : redirige le port 5901 vers le port distant 5900
- **-X** : donne des droits à l'utilisateur qui se connecte (attention, la sécurité peut être diminuée)
- **-C** : permet de chiffrer les données afin de les protéger et d'économiser de la bande-passante
- **-XC** : utilisation simultanée de -X et -C
- **julien** : nom de l'utilisateur qui cherche à se connecter
- **52.552.36.25** : adresse IP du serveur auquel on veut accéder

Les connexions SSH entrant dans une machine passent automatiquement par le port 22 de cette machine afin de sécuriser ses communications (géré par le protocole SSH). Il suffit donc de laisser la machine serveur allumée et de laisser ouvert le terminal dans lequel nous avons rentré la commande ci-dessus afin de pouvoir s'y connecter.

On voit bien que le SSH fonctionne comme un trou de ver. Il récupère les données via le port 5901 du client, les transmet via une connexion sécurisée SSH au port 22, qui devient alors le **seul port d'entrée possible pour un autre utilisateur**. Les données sont ensuite reconstituées au niveau du port 5901 local du client. Il suffit donc simplement de se connecter au localhost :5901 en utilisant un client VNC.

## 3.3 Connexion à distance

Maintenant que la redirection SSH est configurée, nous pouvons connecter les deux machines l'une à l'autre. Pour ce faire, nous allons ouvrir un second terminal dans la machine client dans lequel nous allons taper cette commande :

```
vncviewer localhost :1
```

Cette commande utilise le logiciel vncviewer (que nous avons préalablement installé), ce logiciel se connecte en utilisant le port de sortie 5900+1 (c'est pourquoi nous avons mis localhost :1). La connexion est maintenant établie.

Un autre moyen de se connecter à distance est d'utiliser une interface graphique déjà disponible (Terminal Server Client pour Gnome ou Krdc pour KDE). Ceci a pour utilité principale de ne pas avoir à ouvrir un deuxième terminal pour se connecter au serveur. Pour effectuer cette manipulation, il est nécessaire de configurer la redirection de ports selon le protocole SSH, il suffit ensuite lancer le programme choisis en fonction de l'environnement graphique. Une fois la connexion établie **la machine client possède le contrôle sur la machine serveur**.

## 4 Sources de documentation

[http://fr.wikipedia.org/wiki/Port\\_\(logiciel\)](http://fr.wikipedia.org/wiki/Port_(logiciel))

[http://fr.wikipedia.org/wiki/Virtual\\_Network\\_Computing](http://fr.wikipedia.org/wiki/Virtual_Network_Computing)

<http://doc.ubuntu-fr.org/ssh>

[http://fr.wikipedia.org/wiki/Secure\\_Shell](http://fr.wikipedia.org/wiki/Secure_Shell)

<http://fr.wikipedia.org/wiki/TightVNC>

<http://www.latextemplates.com/>

<http://www.tuteurs.ens.fr/internet/loin/ssh.html>

<http://doc.ubuntu-fr.org/vnc>