

Accès à distance avec Telnet et SSH

Ressources

Liens :

- <https://dev.to/gvelrajan/how-to-configure-and-setup-ssh-certificates-for-ssh-authentication-b52>
- <https://lackof.org/taggart/hacking/ssh/>
- <https://wiki.debian.org/fr/SSH>

Visualiser les services actifs

Lorsqu'un **service est actif** sur un serveur, un **port TCP ou UDP est en écoute** :

- le service Telnet utilise par défaut le port 23 ;
- le service SSH utilise par défaut le port 22.

Utilisez la commande suivante pour visualiser les ports UDP et TCP actif :

```
$ netstat -nltu  
$ ss -nltu
```

Si les ports 23 et 22 se sont pas en écoute → ces deux services ne sont pas installés. Lien :

<https://artheodoc.wordpress.com/2016/06/19/voir-les-ports-ouverts-sous-linux/>

Installer le service Telnet sur la VM Linux

- Lancer votre VM Linux et depuis un terminal, utilisez la commande suivante :

```
$ sudo apt -y install telnetd
```

- Visualisez le **statut** du service et le **port 23 en écoute** : `$ sudo systemctl status inetd $ ss -nlt` Tapez Q pour quitter
- Prenez connaissance de l'adresse IP de votre VM `$ ip a`

Configuration de l'accès Telnet en ligne de commande

Il est nécessaire de disposer d'un client Telnet.

Un client Telnet est disponible sous Windows.

Pour installer le client Telnet sous Windows :

- accédez à l'outil ou **Activer/désactiver des fonctionnalités Windows**
- puis cocher le **Client Telnet** et validez en cliquant sur le bouton **OK**

- Lancez une **invite de commandes** et saisissez le nom du client telnet suivi de l'adresse du serveur distant : `<code>shell> C:>telnet 199.199.199.199 </code>`
 - saisissez vos **identifiants** (login et mot de passe)

Configuration de l'accès Telnet au serveur avec Putty

Putty est un utilitaire qui permet d'ouvrir une session **Telnet** ou **SSH** sur un serveur distant, c'est-à-dire une session de terminal. Ce logiciel correspond à un seul fichier **putty.exe**.

- Télécharger l'utilitaire **putty.exe** à partir du partage **Classe**.

Putty est téléchargeable sur Internet à l'adresse du site <http://www.putty.org/>.

- Configurez Putty pour avoir un accès **console** à votre serveur.
- Indiquez l'**adresse IP** de votre serveur et le **port 23 (Telnet)**.
- précisez le compte de l'utilisateur existant dans votre système Linux **mabanque** pour vous connecter :
- Revenez sur la rubrique Session et sauvegardez les paramètres de votre connexion
- Puis cliquez sur **Open** pour lancer la connexion.
Saisissez le mot de passe du compte pour accéder à l'invite de commande (shell).

Installer le service SSH

- ouvrez une invite de commandes sur le serveur Debian et lancez l'installation d'OpenSSH.

```
$ sudo apt install openssh-server
```

Accéder à distance à la VM

Vous accédez à distance en SSH en utilisant la **commande ssh** sous Linux, MacOS et Windows ou en utilisant un logiciel comme **Putty** sous Windows.

Vous pouvez ouvrir une session afin d'administrer votre serveur :

- en utilisant le compte **root**, solution pratique mais **déconseillée** pour des raisons de sécurité ;
- ou en utilisant un **compte qui n'est pas root**, disposant de moins de droits, mais qui est configuré afin d'avoir la possibilité d'obtenir une **élévation de privilèges** quand cela est nécessaire avec l'utilisation de **sudo**.

Accès à distance avec le compte root (déconseillé)

Pour des raisons de sécurité, l'**accès en SSH avec le compte root est interdit**. Pour le permettre, le fichier **/etc/ssh/sshd_config** doit être modifié :

- la ligne suivante doit être commentée

```
# PermitRootLogin prohibit-password
```

- ajoutez la ligne suivante :

```
PermitRootLogin yes
```

Relancez le service ssh

```
# systemctl restart ssh
```

Pour ouvrir une session à distance avec SSH utilisez maintenant :

- le compte **root** ;
- le mot de passe que vous avez défini (**btssio** suggéré).

Accès à distance avec le compte qui n'est pas root

Si nécessaire **créez un compte linux** sur votre VM en utilisant la commande **adduser**. Renseignez le mot de passe et les autres informations demandées :

```
# adduser btssio
```

Donnez maintenant la possibilité à ce compte d'avoir une **élévation de privilèges** :

- installer sudo

```
# apt install sudo
```

- ajouter l'utilisateur nouvellement créé au groupe sudo

```
# adduser btssio sudo
```

ou

```
# usermod -aG sudo btssio
```

Accès au serveur en ligne de commande avec SSH

Il est nécessaire de disposer d'un client SSH.

Un client SSH disponible sous Windows.

Pour installer le client SSH sous Windows :

- accédez à l'outil aux **fonctionnalités facultatives** de Windows ;
- puis ajouter le **Client OpenSSH**



Paramètres



🏠 Fonctionnalités facultatives



Ajouter une fonctionnalité

[Voir l'historique des fonctionnalités facultatives](#)

Fonctionnalités installées

Rechercher une fonctionnalité facultative installée



Trier par : Nom ▾



Assistance rapide Microsoft

1,44 Mo



Bloc-notes

316 Ko



Client OpenSSH

5,05 Mo

- Lancez une **invite de commandes** et saisissez le nom du client OpenSSH suivi de l'adresse du serveur distant en précisant le compte de connexion : `C:>ssh compte@199.199.199.199`
 - saisissez ensuite le mot de passe du compte.

Il est possible de préciser le mot de passe à la connexion de la manière suivante (déconseillé)

```
C:>ssh compte:motdepasse@199.199.199.199
```

Configuration le client SSH

Lien : http://octetmalin.net/linux/tutoriels/ssh-fichier-etc-ssh_config-configuration-machine-client.php

Sur le client Linux, le fichier **/etc/ssh/ssh_config** permet de configurer les paramètres globaux du client pour toutes les connexions vers des serveurs **ssh**.

Pour chaque compte utilisateur, une configuration personnalisée se fait créant/modifiant le fichier **config** situé dans le répertoire utilisateur **.ssh** (/home/[nom_utilisateur]/.ssh/config sous Linux)

Les options utilisables sont les même que celles du fichier /etc/ssh/sshconfig. === Exemple : === <code shell> Host serveur**ssh**
Host**name** 192.168.10.20 Port 4242 User root IdentityFile clientkey CertificateFile clientkey-cert.pub </code> ===
Explication : === * Host : défini un nom pour le serveur **ssh** * Hostname adresse IP ou nom DNS du serveur * Port : port **ssh**
si différent du port **SSH** par défaut 22 * User : précise le nom de connexion === Utilisation === <code> ssh serveur**ssh**
</code> === Autres options === * Host * : Permet de définir vers quel machine les paramètres vont s'appliquer, l'étoile
veut dire toutes. * RSAAuthentication yes : indique de tenter une authentification **RSA**, clé publique/privé généré avec "ssh-
keygen" * PubKeyAuthentication yes : authentification avec une clé public. * PasswordAuthentication yes : autorise
l'authentification de base avec mot de passe. * CheckHostIP yes : Spécifie si le **ssh** doit vérifier l'adresse IP de l'hôte qui se
connectent au serveur pour détecter une usurpation **DNS**. * IdentityFile ~/.ssh/iddsa : définit la clé privé a utiliser pour
s'authentifier lors de la connexion au serveur * User nomducompte : définit le nom du compte utilisateur à distance à
utiliser pour ce connecter. * Port 22 : numéro de port du serveur **SSH** distant. === Mémoriser la clé privée === Lien :
<http://www.openssh.com/manual.html> La commande **ssh-add** permet de mémoriser une clé privée avec sa passphrase. La
passphrase est demandée lors de l'exécution de la commande. <code> ssh-add Enter passphrase for
/home/cedric/.ssh/iddsa: Identity added: /home/utilisateur/.ssh/iddsa (/home/utilisateur/.ssh/iddsa) </code> Lors de la
connexion au serveur **SSH** distant, il ne sera plus nécessaire de saisir la passphrase. === Lister les empreintes
(fingerprints) des clés privées (identités) en mémoire === <code> ssh-add -l 2048
SHA256:XZvFr9RRRRRRRRRRRsiU6wuH7M0Tdr+9eDYlut1pozxc ..ssh\id_rsa (RSA) </code> Si ce message suivant apparaît,
cela signifie qu'il n'y a pas de clé dont la passphrase est en mémoire <code> The agent has no identities. </code> ===
Lister les clés publiques associées aux clés privées en mémoires === <code> ssh-add -L ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAzWXO7r9vOclav80OG2OMLdRXYLx1L4+hf6hPwm/bMfvbJdLN5s3cyxf37/XzwfEqdhMA60IH
c7LeKtdBkadmocbmmggjvXoCZwY6J1RspY+KcY9oCDNsOFgPNyKF7I7YKXOKmc6TJxLg4R5ZeITpIN3SX+YyAulxOLT4KVDutNI2ia
1XsK527nijbXNtExgwlPJ3r1zkmyMI6eYMzQedtkpHntc8PFxTuMqHNDcbBUKAXcWMEwdDEAIgwelnpjup9BCePLGBwGPWNIVV6
mQEkW9psmWavdRfMsfblLweahSfE4rhoBd9qSXRJwfwf6yvKnAj+N8rTul+SlJl20LHAzw== ..ssh\id_rsa </code> === Supprimer
une identité en mémoire === <code> ssh-add -d </code> === Supprimer toutes les dentités en mémoire === <code> ssh-
add -D </code> ===== Configuration de l'accès **SSH** au serveur avec Putty ===== * Configurez Putty pour avoir un accès
console à votre serveur. * Indiquez l'adresse IP de votre serveur et le port 22 (**SSH**).

- * précisez le compte **root** pour vous connecter :
- * Cliquez sur **Open** pour lancer la connexion.

From:

/ - Les cours du BTS SIO

Permanent link:

</doku.php/reseau/debian/ssh?rev=1734531687>

Last update:

2024/12/18 15:21

