Accès à distance avec Telnet et 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** : <code> $ sudo systemctl status inetd $ ss -nlt </code> Tapez Q pour quitter
- Prenez connaissance de l'adresse IP de votre VM <code> $ ip a </code>

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 un 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 shh** 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 suivangte :

```
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
```

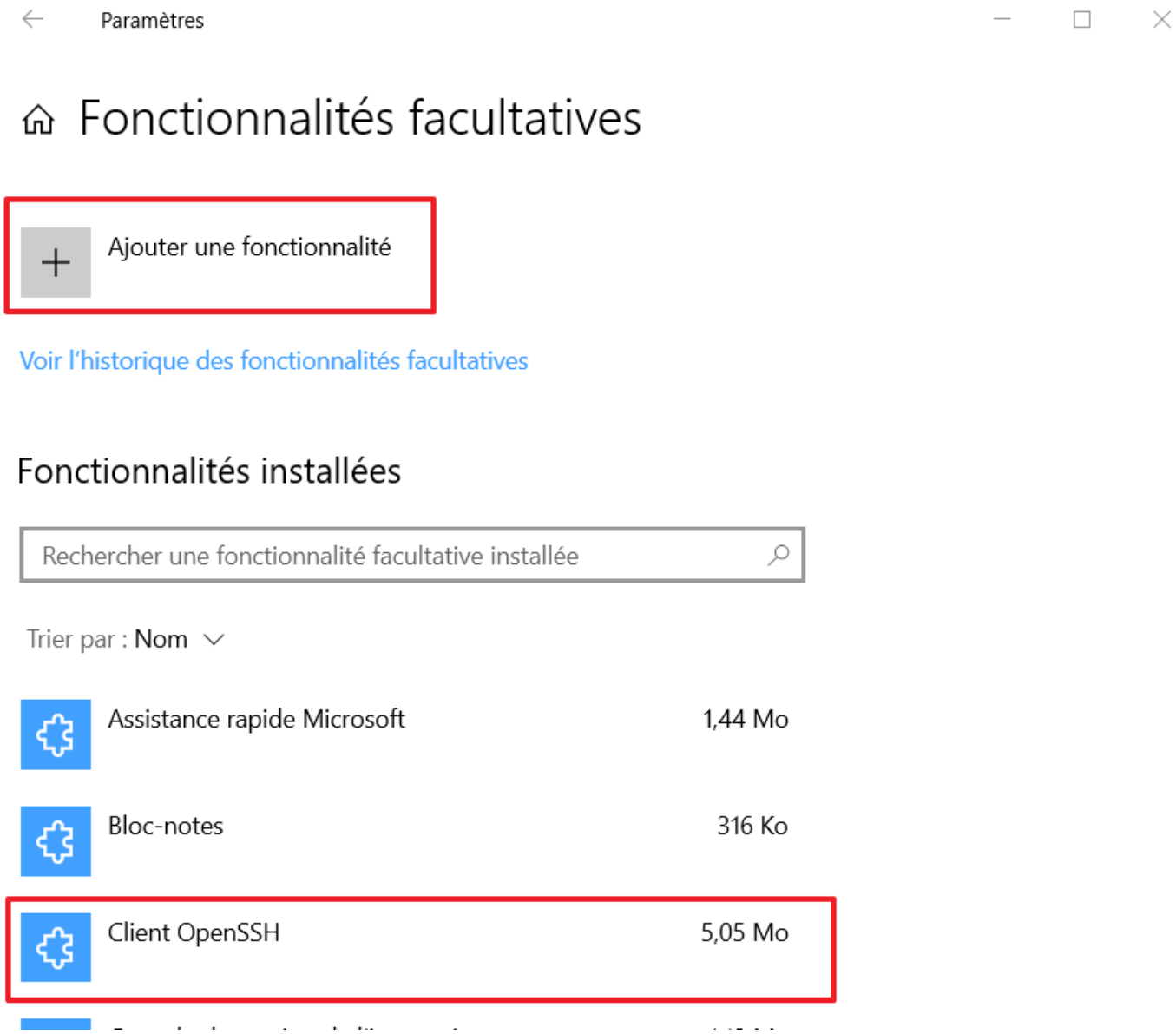
Accès au serveur en ligne de commande avec SSH

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**



- 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 : <code shell> C:>ssh compte@199.199.199.199 </code>
    - 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 configutation 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/ssh_config**.

Exemple :

```
Host serveurssh
  Hostname 192.168.10.20
  Port 4242
  User root
```

Explication :

- **Host** : défini un nom pour le serveur ssh
- **Hostname** adresse IP ou nom DNS du serveur
- **Port** : port ssh si différent du post SSH par defaut 22
- **User** : précise le nom de connexion

Utilisation

```
ssh serveurssh
```

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/id*dsa : 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 === 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/id*dsa: Identity added: /home/utilisateur/.ssh/id*dsa (/home/utilisateur/.ssh/id*dsa) </code> Lors de la connexion au serveur SSSH distant, il ne sera plus nécessaire de saisir la passphrase. === Lister toutes les clés dont les passphrases sont en mémoire === <code> ssh-add -l 2048 SHA256:XZvFr9RRRRRRRRRsIU6wuH7M0Tdr+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> === Afficher l'empreinte des clés en mémoires === L'option “-L”(majuscule) permet d'afficher l'empreinte des clés qui sont en mémoires. Syntaxe: ssh-add -L Exemple: $ ssh-add -L ssh-dss AAAAB3NzaC1kc3MAAACBAKCkD89unQxuhLVHObtC7Jfh4aKcEqxIvCbNUWDm2729lcfGVd6OS7e93y4bBq /i4HjP0nPIO344E4vCeojfcexhbj9wRQxn5L7gX25BgaE5rrbh66bB8VNrMotNWDpl0HCtKt8mTQmVX7tWeCkZkYot jrrwE6i/OMwF57tdxzt/AAAAFQC53V0c5s6R6rHQCI6Myyddw0Ng0wAAAIEAnS+xiRIy/JPL8nfTIXsPPPYpLpt3aR 69rBCJlOH762yIldYJbxzhxUSurx2YSByVHlvW7nrqTzmr0ipXql0MqvZeAlD3*AAgKMwgLuQnTpcNtf/leilR/8m Y/SnWlgDvYPXMDpuIWFcqOxUHi5VEP5f1fjcWryNzqE9Puba4UYAAACBAJM1N4/2CFNxycSdzetZq0fkwJqCU9FtTf 5XoVE0Xz89A1ea+gtL/jDiZUuydurqALK8b5BFzsja16Qg/icceiszcfmf6G6Neuxnmys32RpPOFyKiBwbO1d5Wz2o gMInLeod1OExYY0FC2ric2tuXCePnfQvsylp2tWYofi+EbsH cedric@pc-sport $ Retirer une passphase en mémoire L'option “-d”(minuscule) permet de supprimer la passphrase d'une clé qui est en mémoire. Syntaxe: ssh-add -d Exemple: $ ssh-add -d Identity removed: /home/cedric/.ssh/iddsa (/home/cedric/.ssh/iddsa.pub) $ Retirer toutes les passphrases de la mémoire L'option “-D”(majuscule) permet de supprimer toutes les passphrases de toutes les clés qui sont en mémoire. Syntaxe: ssh-add -D Exemple: $ ssh-add -D All identities removed. $ ==== 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.