TP 02 : Linux : Utilisation distante

Introduction: Ce TP est primordial car, la plupart du temps, vous utiliserez les machines à distance. Il est donc capital que vous maîtrisiez l'utilisation d'OpenSSH.

1 Bash

- Qu'est-ce qu'une VARIABLE D'ENVIRONNEMENT ?
- Donner des exemples de variables d'environnement.
- Qu'est-ce que la sortie standard/la sortie d'erreur ? Comment mélanger la sortie standard avec la sortie d'erreur ?
- Dans un script bash, quelle est la différence entre :
  `VAR="coucou"`
  `echo $VAR`
  et
  `VAR = "coucou"`
  `echo $VAR`
  ?
- Créer un nouveau dossier et lancer la commande `touch ls && ` `ls` ` ` (utiliser le backquote de la touche 7 sur un azerty). Expliquer le résultat.
- Refaite l'expérience dans un nouveau dossier avec la commande `touch ` `ls` ` && ``ls`` `. Est-ce cohérent avec votre réponse à la question précédente ?
- Comment faire une sorte qu'un script soit lancé périodiquement en bash ?

2 Secure Shell

Dans cette première partie, nous allons réviser les commandes de connexion à distance via SSH. Vous pourrez à titre informatif, et/ou culturel, chercher l'équivalent de chaque commande sous MacOS et/ou Windows. OpenSSH est un outil de connexion à distance composé d'un client («ssh» sous unix, «putty» sous windows, ...) et d'un serveur («sshd» sous Linux). Les fichiers de configuration d'OpenSSH se trouvent dans le repertoire «/etc/ssh/». Les fichiers de personnalisation du client se trouvent dans le homedir de l'utilisateur, dans le répertoire «.ssh/».

Après vous être documenté sur internet, expliquer rapidement comment fonctionne la connexion d'un client SSH sur un serveur. Pourquoi peut-on dire que ce protocole est sécurisé ?

2.1 Premières utilisations

La syntaxe de l'utilisation du client «ssh» est la suivante :

```
1 ssh [ options ] [ login@hostname ]  [ commande ]
```

Si le fichier «~/.ssh/known_hosts» existe, renommez-le :

```
mv ~/.ssh/known_hosts{,.bak}
```

Expliquez la syntaxe proposée.
Connectez vous au serveur SSH du département. Que ce passe-t-il ? Expliquez.
Exécuter la commande suivante :

```
sed 's/.\{5\}$/aaaaa/' ~/.ssh/known_hosts
```

Se re-connecter au serveur. Que se passe-t-il ? Pourquoi ?
Utiliser «ssh-keygen» pour solutionner ce problème.

Questions bonus :

- Afficher le fichier `known_hosts`. Pourquoi il y a deux entrées ?
- Quel encodage est utilisé pour stocker la clé RSA ?
- À quoi correspond la partie avant `ssh-rsa` (regarder l'option `-F` de `ssh-keygen`) ? Pourquoi est-elle chiffrée à votre avis ?
- (Difficile) Implémenter l'option `-F` de ssh-keygen.

2.2 Génération d'une clef asymétrique

NB : Vous verrez en détail le fonctionnement du chiffrage par clé dans le cours de crypto du second semestre.

OpenSSH est une implémentation du protocole SSH. Cette implémentation est open source et est libre sous Linux. Elle utilise la commande «ssh-keygen» pour générer les clefs asymétriques. Utiliser la commande «man» pour trouver comment fonctionne la génération de clef.

Vous verrez qu'il vous est demandé, à la fin de la génération de la clef, une «passphrase». La passphrase est un secret qui sera utilisée avec un cryptosystème symétrique.

- Générez deux clefs asymétriques, l'une avec une passphrase, l'autre sans, respectivement dans les fichiers «unsecure» et «secure» de votre répertoire de configuration de «SSH».
- Identifiez la partie publique et la partie privée du secret asymétrique en regardant le contenu des fichiers.
- Déterminez quelle partie de la clef est chiffrée. A quoi sert-il de chiffrer cette partie de la clef asymétrique ?
- Pourquoi pas les deux parties ?
- Que se passe-t-il si vous perdez la passphrase ?
- (Bonus) Comment est chiffré la clé privée avec la passphrase ?

2.3 Authentification par clés

Une façon de voir le cryptosystème asymétrique est d'imaginer que la partie publique de la clé est une serrure et que la partie privée est la clé de cette serrure.

La mise en place de l'authentification par clés consiste à déposer, sur votre compte, dans votre répertoire «.ssh/», à l'intérieur du fichier «authorized_keys», la partie publique de la clé (la serrure).

Nous préciserons ensuite au programme ssh quelle clé utiliser pour déverrouiller la serrure. L'accès à la clé est protégée ou non par mot de passe.

- Déployez vos clés publiques (générées précédement) dans votre homedir (en une seule commande) ;
- votre home étant distribué, essayez d'accéder au serveur ssh du département (SSH, ssh.dptinfo.enscachan.fr). Analysez ce qui se passe et corrigez la commande en cas de besoin.

— que fait la commande «ssh-keygen -l -f unsecure» ? et «ssh-keygen -l -f unsecure.pub» ? A quoi sert cette sortie ?

2.4 Personnalisation de ssh

Il est possible de modifier un certain nombre de paramètres de fonctionnement d'OpenSSH via le fichier «~/.ssh/config».

La page de manuel de «ssh_config» donnne une liste des options disponibles.

Générer un fichier de configuration permettant de se connecter au serveur SSH en utilisant l'alias «secure» pour utiliser la clé protégée par mot de passe et l'alias «unsecure» pour utiliser la clé non protégée. Tester ces configurations.

Tester la configuration .

2.5 Utilisation d'un porte clés

Il est possible de gérer les clefs OpenSSH asymétriques avec un porte clef nommé «ssh-agent». Ce dernier maintient les clefs en mémoire pour éviter de devoir les recharger et éviter la resaisie de la «passphrase».

Quelles sont le dés/avantages de charger une clef en mémoire ?

Lancez l'agent et exportez les variables dans votre environnement. Trouvez comment ajouter une clef au porte clef. Utilisez le porte clef pour faire une connection SSH.

Pouvez-vous expliquer le fonctionnement de ssh-agent ?

2.6 Différents modes de connexions

Vous verrez en détail en deuxième année le fonctionnement du réseau dans le cours «réseaux». En attendant, imaginez une machine connectée au réseau comme un gros bloc de 65535 prises. Ces prises sont appelées en terminologie réseau, des ports. Un programme pouvant être accédé par le réseau utilise une prise. On dit qu'il «écoute» sur un port. Sauf cas particuliers, un port ne peut-être utilisé que par un programme à la fois. Les 1024 premiers ports sont réservés pour le système. Vous pouvez utiliser les autres ports. Un certain nombre de ports sont réservés pour des services particuliers, par exemple, le port 22 est réservé pour SSH, le port 80 est réservé pour le protocole http, le port 443 l'est pour le protocole https, etc... Lorsque nous utiliserons des ports pour les redirections de ports, ils devront donc «être libre», c'est à dire qu'aucun programme ne les utilisera déjà, et non réservés, donc supérieurs à 1024.

Les techniques évoquées ci dessous vous permettront de vous connecter depuis l'extérieur sur les machines de la salle 411 ou plus généralement de vous connecter sur un réseau depuis un autre au travers d'une machine accessible en SSH.

2.6.1 Connexion par rebond

Se connecter à une machine située dans un réseau local peut s'illustrer par la figure 1 page 4 :

1. le poste «Client 1» se connecte à «Passerelle SSH» ;
2. depuis le terminal ouvert sur «Passerelle SSH», l'utilisateur ouvre une connexion sur «Serveur cible».

La machine «Serveur cible» n'est pas forcément accessible depuis l'extérieur.

Cette méthode ne permet pas d'exporter les applications graphiques (via l'option '-X') ou d'effectuer des copies de fichiers directement sur le serveur cible.

Essayez de vous connecter en utilisant cette méthode sur la machine de vos voisins en passant par la passerelle SSH du département.

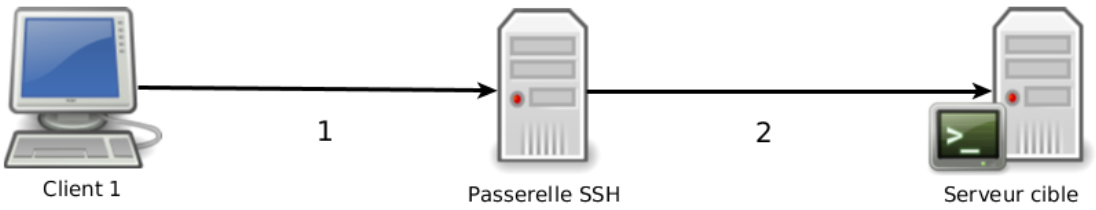

FIGURE 1 – Sans pivot

2.6.2 Re direction de port

La redirection de port permet de connecter le port X d'une machine distante sur le port Y de la machine locale au travers d'une tierce machine comme l'illustre la figure 2 page 4. Dans ce cas :

1. le poste «Client 1» se connecte à «Passerelle SSH» pour lui demander de créer un tunnel redirigeant le port X du serveur cible sur le port Y de la machine locale ;
2. la passerelle SSH assure la mise en place de ce tunnel qui restera actif tant que la connexion restera ouverte ;
3. à l'aide d'un autre terminal, le poste client se connecte sur l'entrée du tunnel qui redirige le trafic directement sur le port du serveur cible.

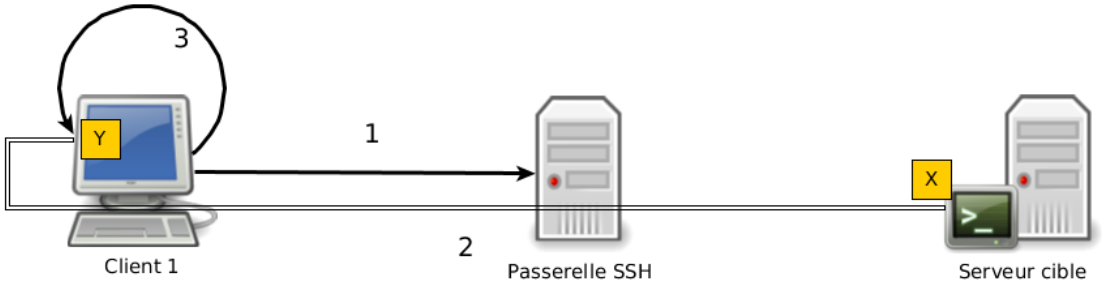

FIGURE 2 – Redirection de ports

NB : En simple utilisateur, le port Y doit être un port non privilégié (donc suppérieur à 1024).

L'exemple suivant redirige le port ssh (22) de la machine 22.dptinfo.ens-cachan.fr sur le port 2222 de la machine locale au travers du serveur SSH ssh.dptinfo.ens-cachan.fr :

```
ssh -L 2222:22.dptinfo.ens-cachan.fr:22 login@ssh.dptinfo.ens-cachan.fr
```

Dans un second terminal, il est alors possible de se connecter sur la machine 22 en empruntant le tunnel

```
ssh -p 2222 login@localhost
```

Connectez vous sur la machine de votre voisin en utilisant une redirection de port SSH sur le port 1000 puis le port 2000. Que ce passe-t-il ?

Utiliser la redirection précédente pour vous connecter sur votre machine.

2.6.3 Le pivot

La méthode du pivot, illustrée figure 3 page 5, permet d'accéder à une machine au travers d'une ou plusieurs autres de manière quasi transparente.

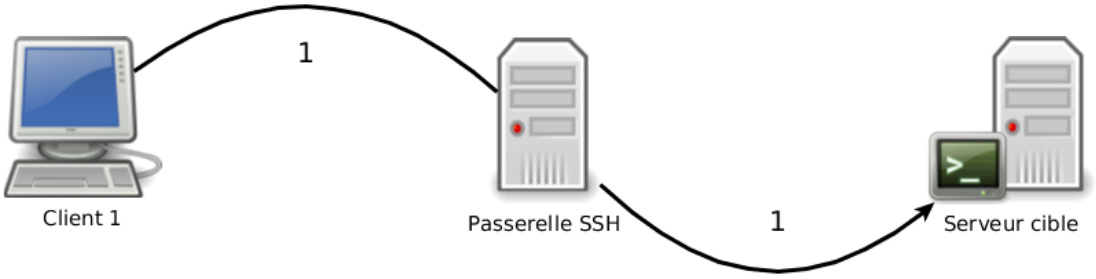

FIGURE 3 – Méthode du pivot

Au contraire de la connexion par rebond, l'utilisation de la passerelle est transparente pour l'utilisateur. La connexion étant redirigée à la manière de la redirection de port, l'export X fonctionne.

Via le fichier de configuration de ssh

L'option «ProxyCommand», ajoutée à la définition d'un alias dans le fichier de personnalisation du client ssh, permet de spécifier la machine utilisée comme pivot.

```
Host <alias_machine>
Hostname <machine.cible>
User <login_sur_machine cible>
ProxyCommand ssh <login>@<machine.ssh.pivot> -W %h:%p
```

où :
— <alias_machine> le nom qui sera utilisé pour appeler cette configuration ;
— <machine.cible> machine sur laquelle nous souhaitons arriver ;
— <login_sur_machine_cible> login à utiliser sur la machine cible, si non précisé, login courant utilisé ;
— <login>@<machine.ssh.pivot> paramètres de connexion sur la machine pivot.

Consulter la page de manuel de «ssh_config» pour vous documenter sur l'option «-W».

Configurez un alias vous permettant de vous connecter sur votre machine au travers du serveur SSH du département d'informatique. Ajouter l'utilisation de la clé SSH «secure» précédement générée.

NB : Vous pouvez utiliser les alias lors de la création de vos alias.

Via les nouvelles fonctionnalités d'ssh

Nous avons compilé, lors du TP précédent, une version récente d'OpenSSH qui offre une nouvelle option '-J' qui permet de spécifier un pivot lors d'une connexion.

La configuration précédente peut alors s'écrire

```
ssh -J <login>@<machine.ssh.pivot> <login_sur_machine_cible>@<machine.cible>
```

Testez la connexion à la machine de votre voisin au travers du serveur SSH du département.

NB : Vous pouvez également utiliser l'option «ProxyJump <login>@<machine.ssh.pivot>» dans votre fichier de configuration. Cette option à été introduite dans OpenSSH 7.3.

2.7 Proxy SOCKS

Vous pouvez utiliser un serveur SSH comme serveur proxy (serveur mandataire) comme illustré sur la figure 4 page 6, c'est a dire que votre trafic web passera par le serveur SSH et que vous naviguerez comme si vous vous trouviez dans les murs de l'ENSC.

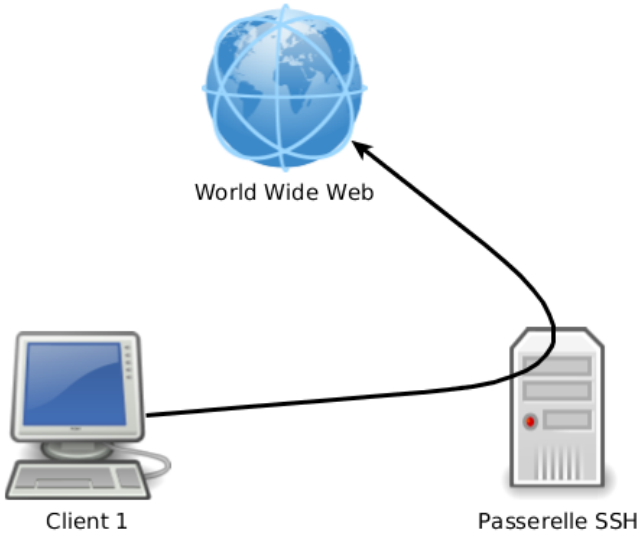

FIGURE 4 – Proxy SOCKS

— Cherchez sur Google, avec firefox, un site affichant votre adresse IP ;
— dans un terminal utiliser l'option «-D <port_non_reservé>» pour créer un proxy local (consulter le manuel de ssh pour voir comment utiliser l'option) ;
— configurer votre navigateur internet pour utiliser votre proxy local ; avec Firefox :
  — rendez-vous dans «préférences» puis «Avancé» (page about :preferences#advanced) et selectionnez l'onglet «réseaux» ;
  — configurez le mode de connexion au réseau en cliquant sur «configuration»
  — dans la nouvelle fenêtre activez «Configuration manuelle» et renseignez les champs «SOCKS Host» = localhost et «Port» = <port_non_reservé> spécifié précédement.
  — validez par «Ok»
— rafraichir la page du site affichant votre IP. Que constatez vous ?
— Fermez la connexion ssh ouverte puis rafraichissez la page web affichant votre IP. Que ce passe-t-il ? Pourquoi ?
— Quel est l'intérêt d'un Proxy ?

3 Multiplexeur de terminal

L'outil «tmux» vous permet d'ouvrir plusieurs instances du shell dans un même terminal et donc via SSH par exemple.

— Lancez «tmux» dans un terminal.
— Lancez la commande «top».
— Ouvrir une nouvelle instance du terminal via «Ctrl+b» (pour signaler que nous envoyons la commande à «tmux») puis «c» (pour create)
— Listez le contenu du répertoire courant
— Retourner sur le shell executant "top" via «Ctrl+b» puis «p» (previous). Utilisez «Ctrl+b» puis «n» pour l'écran suivant (next).
— «Splitez» l'écran verticalement «Ctrl+b» puis «"».

— «Splitez» l'écran horizontalement «Ctrl+b» puis «%».
— Détachez ce tmux du terminal «Ctrl+b» puis «d».
— Ré attachez votre tmux au terminal «tmux a».
— Vérifiez que «top» fonctionne toujours.

Le manuel de tmux et Google vous donneront toutes les astuces du fonctionnement de ce programme.

4 Pour aller plus loin

— Finir le TD de la semaine dernière
— Reprendre l'exercice du TD précédent pour compiler openssh
— Si vous avez votre ordinateur portable, essayer de vous connecter en ssh sur le wargame (overthewire) donné au TD précédent via une connexion `eduroam`
— Faire la question **difficile** du TD (n'hésitez-pas à demander de l'aide)