

Aller au contenu

Activer l'agent openssh :

```
1 | eval $(ssh-agent)
```

Puis verifier si les clé sont deja synchronise pour les passphrase

```
1 | ssh-add -l
```

Pour les supprimer on utilise la commande

```
1 | ssh-add -D
```

Verifier que les clés ssh sont bien créés

```
1 | ls -l ~/.ssh/id_rsa*
```

Si ce n'est pas le cas les crée avec la commande

```
1 | ssh-keygen -t rsa -b 4096
```

Charger la clé pour que l'agent retienne la passphrase

```
1 | ssh-add ~/.ssh/id_rsa
```

Verifier si la clé a été synchronise

```
1 | ssh-add -l
```

Installer l'outil expect :

```
1 | apt install expect
```

Crée un script **send\_pubkey.exp** dans le dossier **/home/scripts**

```
1  #!/usr/bin/expect
2  # USAGE : ./send_pubkey.exp keyfile user password @ip
3  # Cohérence des arguments passés à tester (au moins le nombre et le fichier de clé)
4  set force_conservative 1
5  set timeout 9
6  # Ici on récupère les arguments passés en paramètres au script
7  set key [lindex $argv 3]
8  set user [lindex $argv 0]
9  set password [lindex $argv 1]
10 set hostname [lindex $argv 2]
11
12 #Verif de la config
13 #if { llength $argv != 4 }{
14 # do puts "invalid number of argument"
15 # exit 2
16 #}
17 #
18 #fi
19
20
21 # On envoie la clé $key avec spawn
22 spawn ssh-copy-id -i /home/std/.ssh/id_rsa.pub $user@$hostname
23 # On affiche un message à l'écran
24 send_user "Tentative d'envoi de la clé $key vers $user@$hostname avec le pass :
```

```

25 $password\n"
26 # On attend un retour du serveur ssh
27 expect {
28     # Sil y a un problème, on affiche une erreur
29     timeout { send_user "\nFailed to get password prompt\n"; exit 1 }
30     eof { send_user "\nSSH failure for $hostname\n"; exit 1 }
31     # Lors de la 1ère connexion ssh, un message apparaît avec yes à répondre, donc on gère :
32     "*re you sure you want to continue connecting" {
33         send "yes\r"
34         exp_continue
35     }
36     # Lorsque le mot de passe est demandé, le script l'envoie avec send
37     "*password:*" {
38         send "$password\r\r"
39         interact
40     }
41 }
42 exit 0

```

On a aussi créé un fichier hosts.ip pour repertorier tous les user, password, ip et location de la clé

```

1 std:password:10.31.240.33:~/.ssh/id_rsa.pub
2 std:password:10.31.240.67:~/.ssh/id_rsa.pub
3 std:password:10.31.240.68:~/.ssh/id_rsa.pub
4 std:password:10.31.240.53:~/.ssh/id_rsa.pub
5 std:password:10.31.240.54:~/.ssh/id_rsa.pub
6 std:password:10.31.240.80:~/.ssh/id_rsa.pub
7 std:password:10.31.240.13:~/.ssh/id_rsa.pub
8 std:password:10.31.248.80:~/.ssh/id_rsa.pub
9 std:password:10.31.248.20:~/.ssh/id_rsa.pub
10 std:password:10.31.240.34:~/.ssh/id_rsa.pub

```

Le script synchro.sh permet de rentrer les valeurs du script hosts.ip dans le script send\_pubkey.exp synchro.sh :

```

1 #!/bin/bash
2 filename=$1
3 while IFS= read -r line
4 do
5     user=$(echo $line | cut -d ":" -f1)
6     echo $user
7     password=$(echo $line | cut -d ":" -f2)
8     echo $password
9     hostname=$(echo $line | cut -d ":" -f3)
10    echo $hostname
11    key=$(echo $line | cut -d ":" -f4)
12    echo $key
13    echo $user / $password / $hostname / $key
14
15    ./send_pubkey.exp $user $password $hostname $key
16 done < "$filename"

```

Pour exécuter le script on lance

```
1 ./synchro.sh hosts.ip
```

**IMPORTANT : il faut créer les clés ssh et exécuter le script avec l'utilisateur std et non root**

Grâce au envoi de clé au vm j'ai pu mettre à jour le dhcp à distance :

```
1 ssh std@10.31.240.67 'sudo apt update'
```

Pour supprimer les clé a distance on crée un script delete\_key.sh

sir2-afrique/mission\_10\_hadj.txt · Dernière modification: 2021/11/16 15:07 de h-benzahaf