Authentification par clé sans passephrase

Pour commencer nous devons d'abord créer les conteneur pour **Backup-01** et **Backup-02** avec pour adresse **IP 10.31.192.73** et **10.31.192.74** sur notre réseau du LAN. Une fois cela fait on installe à l'intérieure de ces conteneur le service Backuppc.

- `apt-get install backuppc`

Ensuite on se connecte à l'utilisateur Backuppc.

- `su - backuppc`

Puis on crée une clé publique dans cette utilisateur sans ajouter de passephrase.

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

Nous allons copier car elle nous servira justement lors de la configuration pour qu'on puisse se connecter aux autres conteneur sans mot de passe.

Configuration et installation

1.Configuration

Dans cette étape nous allons d'abord commencer par installer rsync dans notre conteneur backup-01/02 et dans tous les autres conteneurs que l'on possède.

- `apt-get install rsync`

Ensuite dans notre PC local on va dans l'explorateur de fichier et on créer un dossier **SCRIPT** dans le Bureau par exemple et à l'intérieure on créer un fichier “monscript.py” ; un fichier “id_rsa.pub.txt” et un fichier “target.ip.txt”

- **Monscript.py**

```
import os
import re

######
###    Deploy
###  - the content of a copy id_rsa.pub from bacuppc server (user backuppc)
###  - to the authorized_keys file of a list of virtual machines (user
```

```
backup)
###  - Target directory : /var/backups/.ssh/authorized_keys
######
|

remote_dir = "/var/backups/.ssh/"
remote_file = "authorized_keys"
remote_path = remote_dir + remote_file

def is_valid_ipv4(ip):
    pattern = r'^(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.' \
              r'(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.' \
              r'(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.' \
              r'(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$'

    if re.match(pattern, ip):
        return True
    else:
        return False

with open("target.ip.txt", "r") as inputfile:
    for line in inputfile:
        ip = line.rstrip()

        if not is_valid_ipv4(ip):
            print(f"\n Skipping {ip} (- Unvalid IPv4 Address -) ")
            continue

        print("")
        print("*"*25)
        print("*** {ip}")
        print("*"*25)

        with open ("id_rsa.pub.txt", "r") as keyfile:
            mykey = keyfile.readlines()[0].rstrip()

        # creating .ssh folder if it doesn't exist
        print("-- Updating backup user shell (sh)...")
        print("-- Creating /var/backups/.ssh/ directory...")
        print("-- Adding pubkey in authorized_keys file...")
        print("-- Updating authorized_keys permissions...")
        os.system(f"ssh root@{ip} 'usermod --shell /bin/sh backup && mkdir -
p {remote_dir} && echo {mykey} >> {remote_path} && chown backup
{remote_path}'")

        # creating sudo permissions
        sudo_cfg = "backup ALL=NOPASSWD: /usr/bin/rsync"

        print("-- Adding sudo configuration for backup account")
        print(f"---- {sudo_cfg}")
        os.system(f"ssh root@{ip} 'echo {sudo_cfg} >
```

```
/etc/sudoers.d/backup'")
        print("-- DONE")

        print("*"*25)
```

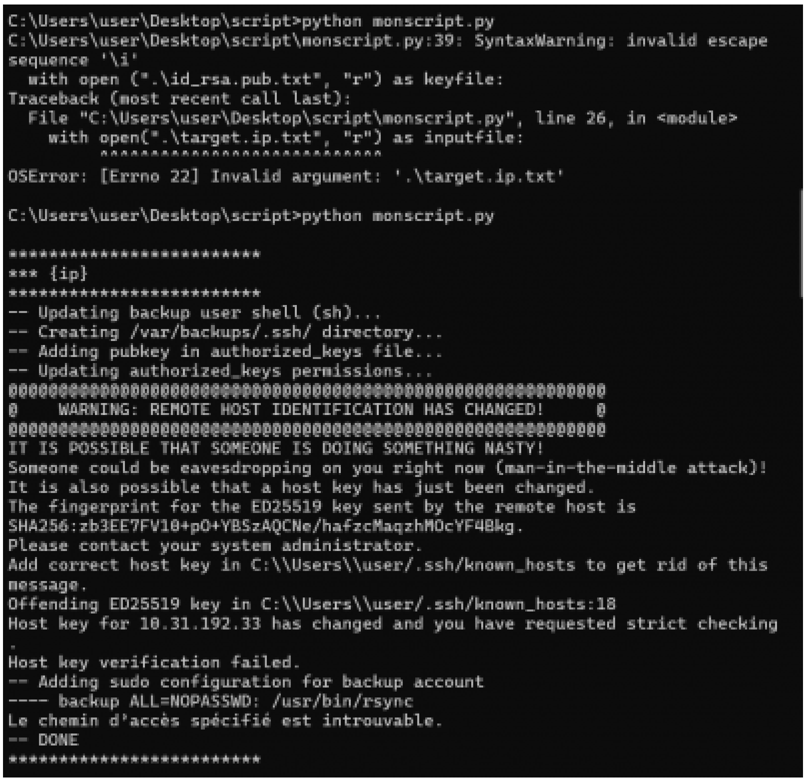
- **id_rsa.pub.txt**

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQDVQoCulXgqd/kpr07oTsDmFFKL5K/5AcoQJRxC2xu3adS2
JA51bmaa5RUwQM4NcUkwi5f8GwMuS60sDbCdolZ+VzlqYMe/IzUQIGJ577vAfUBenftju/tEaRGe
19jSfHdNJARPycUnC918BZx/Pbu16l+F8rL9CkG81jVYcIS9NYDjn2BdOppT4ahoiGmrWr35nRuX
8/YyyKJl0iQlA3Bjge0w4SqBvy7mZbeLH+1yWoBEbcAc1FtvKXN8A4BUU5Pc7EUWhbcaysB8iBCw
CkjuQCGJIgDtqWjE44btWpVeCkB7oLtu+BDVeBhRgVSQGFVL2P64EeG+yNdoREfn4TKfo4s6tDCK
c3jCh+nsXOptFUgNy8YC1b8etchgegZH98KUnYGuTxdWZY8lxQ3RCaGY81Txzc4WB8/4iSL0UPZc
2SpFMNWePxjZ24gtYwHeSLlgd/34UvwRO7iXBRdoaoNzhyfS/M6uUNPpCnXTHMkd3/SAaiylJIDj
PwVI7vTiX55pRLDe5LPMvTjI76FbfGPYp7EvMKwcWH2EQZkPtpN6bN3tjac0epKvyKNTk0BfY/UZ
ywksn3nRXYtRYRI2vyKEekPCMsYZPuhmFHldrKaRQlU/whwfaZMiGM9QHLXaBxNGX+aBTKjIYJRA
6hA+4sT8utsg8nLE8XL1clqMJNnuGw== backuppc@backup-02
```

- **Target.ip.txt**

```
10.31.192.33
10.31.192.34
10.31.200.67
10.31.192.2
10.31.200.54
10.31.200.53
10.31.192.74
10.31.192.73
```

Nous allons exécuter le script python dans notre machine locale pour qu'il puisse crée les répertoires et les connexions automatiques avec la clé publique de backuppc. Cela va générer un message d'avertissement. Il suffira à ce moment de faire un yes à chaque question concernant le fingerprint de chaque machines.



```
C:\Users\user\Desktop\script>python monscript.py
C:\Users\user\Desktop\script\monscript.py:39: SyntaxWarning: invalid escape
sequence '\i'
  with open (".\id_rsa.pub.txt", "r") as keyfile:
Traceback (most recent call last):
  File "C:\Users\user\Desktop\script\monscript.py", line 26, in <module>
    with open(".\target.ip.txt", "r") as inputfile:
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
OSError: [Errno 22] Invalid argument: '.\target.ip.txt'

C:\Users\user\Desktop\script>python monscript.py

*************************
*** {ip}
*************************
-- Updating backup user shell (sh)...
-- Creating /var/backups/.ssh/ directory...
-- Adding pubkey in authorized_keys file...
-- Updating authorized_keys permissions...
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ED25519 key sent by the remote host is
SHA256:zb3EE7FV10+pO+YBSzAQCNe/hafzcMaqzhMOcYF4Bkg.
Please contact your system administrator.
Add correct host key in C:\\Users\\user/.ssh/known_hosts to get rid of this
message.
Offending ED25519 key in C:\\Users\\user/.ssh/known_hosts:18
Host key for 10.31.192.33 has changed and you have requested strict checking
.
Host key verification failed.
-- Adding sudo configuration for backup account
---- backup ALL=NOPASSWD: /usr/bin/rsync
Le chemin d'accès spécifié est introuvable.
-- DONE
*************************
```

Puis on test les connexion avec les conteneurs depuis la machine backup avec l'utilisateur backuppc.

```
su - backuppc
ssh backup@10.31.192.67
```

2.Configuration

Dans cette étape nous allons créer un utilisateur que l'on va nommer europe-backup. Pour que cela fonctionne il :

- soit vous notez soigneusement le mot de passe attribué automatiquement à l'utilisateur backupPC
- soit vous préférez le modifier à l'aide de cette commande:

```
sudo htpasswd /etc/backuppc/htpasswd backuppc
```

Ajout de l'utilisateur dans le groupe backuppc :

Pour démarrer backuppc, il faut ajouter l'utilisateur de la session dans le groupe backuppc. Pour cela, il faut exécuter la ligne de commande suivante :

```
sudo adduser europe-backup backuppc
```

Ajout du fichier apache.conf

Comme l'installation ne copie pas le /etc/backuppc/apache.conf sur le serveur apache2, il faut le faire

soi-même en copiant le fichier dans le répertoire /etc/apache2/sites-available/ avant de rendre actif le site, pour cela utilisez la commande suivante :

```
sudo cp /etc/backuppc/apache.conf /etc/apache2/sites-available/backuppc.conf
```

Puis activez le site :

```
sudo a2ensite backuppc.conf
```

Un redémarrage du serveur web est nécessaire pour prendre en compte les modifications.

```
sudo /etc/init.d/apache2 restart
```

Si vous avez une page d'erreur du type Forbidden - You don't have permission to access this resource, il faut ajouter dans le fichier backuppc.conf la ligne

```
Require all granted<sxh>

et commenter cette ligne :

<sxh bash>require local
```

- **/etc/apache2/sites-available/backuppc.conf**

```
Alias /backuppc /usr/share/backuppc/cgi-bin/

<Directory /usr/share/backuppc/cgi-bin/>
        AllowOverride None

        # Uncomment the line below to ensure that nobody can sniff important
        # info from network traffic during editing of the BackupPC config or
        # when browsing/restoring backups.
        # Requires that you have your webserver set up for SSL (https)
access.
        #SSLRequireSSL

        Options ExecCGI FollowSymlinks
        AddHandler cgi-script .cgi
        DirectoryIndex index.cgi

        AuthUserFile /etc/backuppc/htpasswd
        AuthType basic
        AuthName "BackupPC admin"

        <RequireAll>
```

```
                # Comment out this line once you have setup HTTPS and
uncommented SSLRequireSSL
                #Require local

                # This line ensures that only authenticated users may access
your backups
                Require all granted
                Require valid-user
        </RequireAll>
</Directory>
```

puis redémarrer le serveur web:

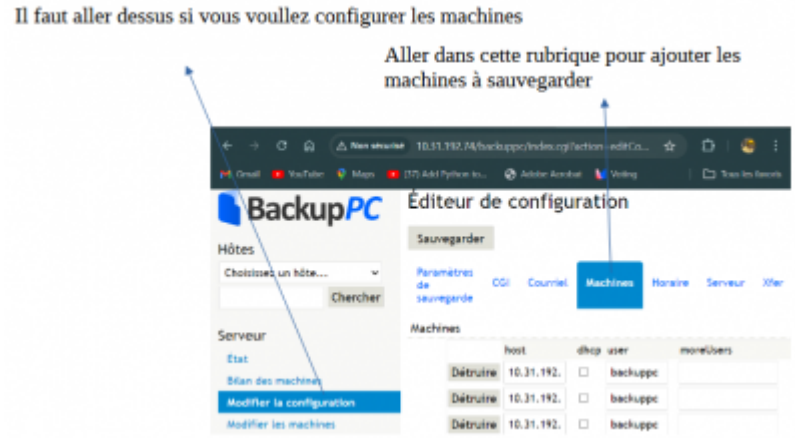
```
sudo /etc/init.d/apache2 restart
```

Si vous avez une erreur de type Forbiden vous devez impérativement ajouter dans le répertoire /etc/apache2/sites-available/backuppc.conf cette commmande dans le requireAll

```
Require all granted
```

Gestion de sauvegarde depuis Backuppc

Une fois les étapes correctement suivi le navigateur envoie cette page :



Après avoir ajouter les machines (Conteneur)qu'il faut sauvegarder nous devons maintenant configurer le Xfer

- Sur la xferMethod choisir Rsync
- Sur RsyncBackupPCPath mettre le chemin :

```
/usr/libexec/backuppc-rsync/rsync_bpc
```

- Sur RsyncClientPath mettre le chemin :

```
sudo /usr/bin/rsync
```