README du Serveur DNS Esclave

- Commande pour une bonne installation du script

Le script devra être placé dans le fichier /etc (Que l'on retrouve dans les dossiers de l'ordinateur)

La permission devra être accordé grâce au la commandes : sudo

-s

<Entrer le mot de passe admin> cd

/etc

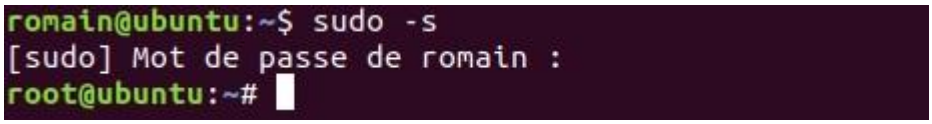
chmod 777 script.sh

- Pour que le ssh puisse fonctionner correctement nous devons l'exécuter en commande la première fois. Il nous suffit d'entrer la commande :

scp SAVEHTTP.csv <NomD'UtilisateurDuDestinataire>@<IpDuDestinataire> :<Chemin d'accès sur la machine du destinataire>

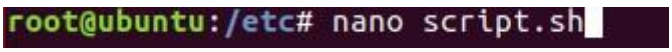
- Commentaire détaillé du code de supervision du DNS Esclave.

Dans un premier temps nous allons passer en mode root. Il faut donc entrer le mot de passe Administrateur.



Ensuite nous nous déplaçons dans le fichier où se trouve le script, ici la commande cd nous est utile. Dans notre cas le script se trouve dans le dossier etc. (cela dépend de l'emplacement que vous souhaitez).

La commande nano nous permettras d'éditer le script. (Dans notre cas son nom est script.sh)



(# !/bin/bash = Permet au script d'être exécuté en bash exclusivement).

Cette première partie nous permet de donner le chemin d'un fichier texte (ici MESS_ERROR.txt) à une variable. La commande echo nous permet d'écrire sur la première ligne du fichier et ainsi de le réinitialiser.

```
#!/bin/bash

#VARIABLE POUR RELEVER LES ERREURS
MESS_ERROR=/etc/MESS_ERROR.txt
echo "Problème(s): " > MESS_ERROR.txt
```

Dans ce second point on crée un dossier (du nom de DossierCSV) qui permettras de stocker les fichiers .csv plus tard. (Le chemin d'accès est personnel à votre machine).

```
#création d'un fichier pour stocker les .csv
DOSSIERCSV=/home/romain/Desktop/DossierCSV
if [ ! -e $DOSSIERCSV ]
then
mkdir $DOSSIERCSV
fi
```

Dans ce point on vient récupérer la date et l'heure et créer un fichier contenant ces informations.

```
#CREER LE FICHIER AVEC LA DATE DU JOUR
DATE=`date +"%kh%M-%d.%m.%y"`
touch $DATE.csv
```

Cet emplacement du code nous permet de définir l'adresse du serveur HTTP. (Celui-ci est personnel à votre installation et doit donc être modifié) Attention à ce que l'IP du serveur soit statique.

```
#DONNER L'ADRESSE DU SERVEUR HTTP
Add_HTTP=192.168.132.129
```

Cette commande nous permet de connaître le temps de réponse du serveur http et de l'inscrire dans le fichier .csv (qui a le nom de la date et de l'heure a laquelle le script a été lancé).

```
#TEMPS DE REPONSE

ping -c1 $Add_HTTP | cut -d"=" -f4 | sed -n '2p' > $DATE.csv
```

Ici il suffit de remplir le nom de domaine et l'IP du DNS. (Cela est personnel à votre installation).

```
#DONNER LE NOM DE DOMAINE ET L'IP DU DNS
DOMAIN=carnofluxe.domain
IPDOMAIN=178.170.102.194
```

Ici on test le serveur DNS, grâce à la commande nslookup. (Les variables sont définies au-dessus). On utilise aussi un fichier temporaire qui va nous permettre de stocker des valeurs pour les réutiliser en évitant les redirections ambiguës. Le sed et le cut vont nous permettre de cibler l'information que l'on cherche. Ensuite nous allons comparer cette donnée avec notre variable préalablement définie. Et renvoyer l'état du serveur DNS dans le fichier .csv. ET aussi renvoyer les erreurs dans le fichier MESS_ERROR.

```
#TEST SERVEUR DNS
tmpdns="`nslookup $DOMAIN | sed -n '6p' | cut -d" " -f2`"
echo $tmpdns > tmpdns
if [ "$tmpdns" = "$IPDOMAIN" ]
then
echo On >> $DATE.csv
else
echo Off >> $DATE.csv
echo "Le serveur DNS ne fonctionne pas" >> $MESS_ERROR
fi
```

Après avoir ping l'adresse du serveur http, on vérifie que celui-ci fonctionne. En utilisant [$? -eq 0] on vérifie que la valeur de la variable est « true » si elle renvoie 0. En fonction de la réponse on ajoute l'accès au serveur http au fichier .csv. Et on renvoie les erreurs dans le fichier MESS_ERROR.

```
#PING SERVEUR HTTP
ping -c1 $Add_HTTP
     if [ $? -eq 0 ]
     then echo On >> $DATE.csv
     else echo Off >> $DATE.csv
     echo "Le serveur HTTP ne fonctionne pas" >> $MESS_ERROR
     fi
```

Le wget nous permet de tester l'accès au site web en essayant de télécharger la page. (Le rm nous permet de supprimer la page téléchargée pour ne pas saturer le disque). Le cut et sed nous permettent encore une fois de récupérer seulement l'informations que l'on cherche et la comparer) Si tout est fonctionnel on écrit dans le fichier .csv que cela marche. On signale aussi si cela ne marche pas. Et on envoie les erreur une nouvelle fois dans le MESS_ERROR.

```
#ACCES AU SITE WEB

wget https://www.cesi.fr 2> tmp1.txt
tmp2="`cut tmp1.txt  -d" " -f10 | sed -n '4p'`"
        if [ "$tmp2" = "OK" ]
        then echo On >> $DATE.csv
        rm index.html
        else echo Off >> $DATE.csv
        echo "Le site WEB n'est pas en ligne" >> $MESS_ERROR
        fi
```

Ici on utilise une variable pour stocker notre fichier texte. Et on envoie un mail contenant les informations, avec le contenu du fichier texte, a l'administrateur.

```
#CREATION D'UNE VARIABLE QUI LIT LE FICHIER TEXTE
var=`cat MESS_ERROR.txt`

#ENVOIE MAIL ERREUR
echo "Ce mail fait suite au rapport $DATE si le message suivant est vide, aucun problèmes. $var" | mail -s "Message d'erreur" root
```

Ici on s'occupe de copier (cp) le fichier .csv pour le normaliser et permettre au site d'afficher les informations plus simplement. (en sachant que les informations sont personnel à votre installation). La commande sshpass nous permet de sauvegarder le fichier .csv sur une autre machine (ici le serveur http). Sans être obliger de devoir redonner le mot de passe a chaque renouvellement du script.

```
#SAVE .CSV SUR HTTP
cp -a $DATE.csv /etc/SAVEHTTP.csv
sshpass -p "bule" scp SAVEHTTP.csv romain@192.168.132.129:/home/romain/Desktop
```

Grâce a cette commande on déplace le .csv dans le Dossier « DossierCSV »

```
#DEPLACER LE FICHIER .CSV DANS LE BON DOSSIER
mv $DATE.csv /home/romain/Desktop/DossierCSV
```

Cette commande nous permet d'exécuter le script.

```
root@ubuntu:/etc# ./script.sh
```

Voici l'exécution du script lorsque tout est hors ligne.

```
root@ubuntu:/etc# ./script.sh
PING 192.168.132.129 (192.168.132.129) 56(84) bytes of data.
From 192.168.132.128 icmp_seq=1 Destination Host Unreachable

--- 192.168.132.129 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

ssh: connect to host 192.168.132.129 port 22: No route to host
lost connection
Vous avez du nouveau courrier dans /var/mail/root
```

Et le mail reçu :

```
Return-Path: <root@ubuntu>
X-Original-To: root@ubuntu
Delivered-To: root@ubuntu
Received: by ubuntu.localdomain (Postfix, from userid 0)
        id 295281000BA; Mon, 11 Feb 2019 06:45:18 -0800 (PST)
Subject: Message d'erreur
To: <root@ubuntu>
X-Mailer: mail (GNU Mailutils 3.4)
Message-Id: <20190211144518.295281000BA@ubuntu.localdomain>
Date: Mon, 11 Feb 2019 06:45:18 -0800 (PST)
From: root <root@ubuntu>

Ce mail fait suite au rapport  6h44-11.02.19 si le message suivant est vide, aucun problèmes. Problème(s):
Le serveur DNS ne fonctionne pas
Le serveur HTTP ne fonctionne pas
Le site WEB n'est pas en ligne
?
```