Station Blanche

Document d'exploitation

Kim LAUGAUDIN

Compétences validées :

5. Mettre à disposition des utilisateurs un service informatique

- Réaliser les tests d'intégration et d'acceptation d'un service
- Déployer un service

Table des matières

1. Introduction

Une station blanche est un ordinateur sur lequel un script de décontamination de clé USB tourne. Ce script analyse les fichiers PDF présents sur la clé USB qui est branchée sur l'ordinateur. Il va lancer un scan antivirus, si le fichier est contaminé, il part en quarantaine ou est supprimé en fonction des choix indiqués dans le script. Il va également vérifier la présence de JavaScript, s'il y en a, il sera désactivé.

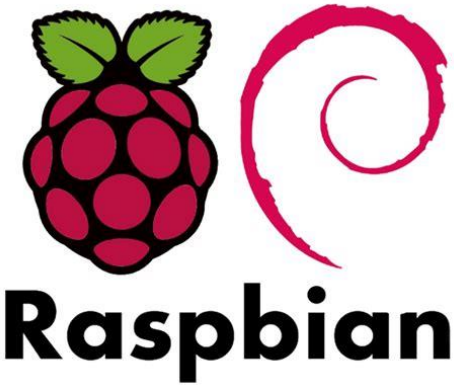
Dans le cadre de mon alternance, on m'a demandé de créer une station blanche qui permettrait de non seulement d'analyser les fichiers PDF présents sur la clé USB mais également de copier ces fichiers et uniquement ceux là sur une autre clé USB qui serait amenée à être plugger sur les postes des utilisateurs.

Le script est écrit en bash. Il s'agit d'un puissant outil de scripting. C'est également l'acronyme de Bourne Again Shell.

2. Prérequis

L'ordinateur utilisé pour la station blanche est un Raspberry Pi. Il s'agit d'un nano ordinateur de la taille d'une carte de crédit à processeur ARM.

Le système d'exploitation à installer dessus est Raspbian. Il s'agit d'une distribution Linux basée sur Debian.



Une fois monté et configuré, nous pouvons passer à l'installation des applications et services nécessaires au script de décontamination.

Nous avons donc besoin d'un logiciel qui permet de parser les PDF et qui désarme les entêtes JavaScript, et nous avons également besoin d'un antivirus.

3. PDFID

PDFID, est l'application qui permet de parser les PDF, c'est-à-dire qu'elle permet d'analyser les noms et extensions des fichiers présents dans le chemin indiqué et qu'elle va récupérer ceux avec l'extension .pdf. Elle permet également de désactiver JavaScript dans les fichiers PDF.

Pour installer PDFID, nous avons tout d'abord besoin d'ajouter les dépôts de Kali sur Raspbian, car PDFID ne fait pas parti de ce dernier. Pour cela il nous faut ouvrir le fichier sources.list situé dans /etc/apt/

Il faut donc ouvrir un terminal et taper :

```
pi@raspberry $~: nano /etc/apt/sources.list
```

Puis rajouter dedans la ligne suivante qui permettra d'indiquer à la commande apt où aller chercher les dépôts de Kali.

```
Deb http://kali.org/kali kali-rolling main non free contrib
```

Une fois fait on sauvegarde en faisant ctrl+o puis entrer, et quitte en faisant ctrl+x puis entrer.

Nous lançons une mise à jour des paquets pour télécharger les dépôts Kali.

```
pi@raspberry $~ sudo apt update && apt upgrade
```

La première commande télécharge les paquets, la seconde les mets à jour.

Une fois fait, nous pouvons passer à l'installation de PDFID en entrant la commande :

```
pi@raspberry $~ sudo apt install PDFID
```

L'installation de l'application se lance. Une fois installée nous pouvons passer à l'installation de l'antivirus.

4. Clamav

Clamav est un antivirus qui va analyser tous les fichiers présents sur l'ordinateur ou dans le chemin indiqué.

Il s'agit d'un antivirus principalement utilisé sur les systèmes d'exploitation Linux et MacOS, mais il existe également une version pour Windows.

Cette fois, nous n'avons pas besoin d'importer les dépôts d'un autre système d'exploitation, Clamav est disponible sur Raspbian.

Nous pouvons donc lancer la commande suivante :

```
pi@raspberry $~ sudo apt install clamav
```

L'installation se lance.

5. Fichier Fonctions.sh

Maintenant que nous avons téléchargé PDFID et Clamav nous pouvons passer à la création du script. Pour cela, nous ouvrons un terminal et créer le fichier fonctions.sh

La commande pour créer un fichier est la suivante :

```
pi@raspberry ~$ sudo nano fonctions.sh
```

Puis nous devons rendre ce fichier executable avec la commande :

```
pi@raspberry ~$ sudo chmod u+x fonctions.sh
```

Sans cela, nous ne pourrons exécuter les fichiers en .sh, il ne faut donc pas oublier de le faire.

Ce dossier va accueillir toutes les fonctions que nous créerons pour le script, cela permet d'avoir un fichier de script qui ne fera qu'appeler les fonctions.

Le fichier .sh doit toujours commencer par

" #!/bin/bash"

En effet, cela indique au fichier que le langage utilisé sera du bash.

5.1. PDFID

Avant de commencer la fonction, on indique au fichier que l'on se déplace dans la clé, les indications situées après le « # » sont des indications commentées qui n'impactent pas le script, elles ne sont là qu'à destination du lecteur/créateur du script :

_____

#PARSING DE PDF

```
desarme(){
log=/var/log/StationBlanche

zenity --info --text="Le désarmement va commencer.."

OLDIFS=$IFS

IFS=$'\n'

files=$(find -depth -name '*.pdf')

for f in $files
do
        check=$(pdfid $f | grep JavaScript | awk '{print $2}') > /dev/null 2>&1

        if [ $check != "0" ]; then
        echo "`date +%D-%H-%M` $f is armed" >> $log/PDFarmed.log
        pdfid $f --disarm > /dev/null 2>&1

        else
                echo "Fichier saint" >> $log/PDFsaints.log
        fi
done


FS=$OLDIFS

}
```

_____

OLDIF permet d'échapper les espace des noms des fichiers.
On indique à PDFID dans une variable files quelle-ci correspond au nom des fichiers qui finissent par .pdf, et nous réutilisons cette variable plus loin.

Nous demandons à PDFID d'analyser si JavaScript est activé, puis d'entrer les résultats dans deux fichiers de log différents en fonction de si le fichier est saint ou JavaScript est activé. Le nom du fichier de log contiendra la date de l'analyse, le nom du fichier concerné et s'il est armé ou non. Ensuite nous demandons à PDFID de désarmer le fichier s'il est armé.

5.2. Analyse Clamav

La fonction d'analyse de la clé est la suivante :

```
#analyse et décontamination de la clé usb - modifier chemin de la clé

virscan(){

log=/var/log/StationBlanche

echo "Analyse des fichiers en cours..."

clamscan -r --quiet -i --move=/home/pi/Quarantaine --log=$log/clamav.log /media/pi/*


}
```

Nous demandons à Clamav d'analyser tous les fichiers présents sur la clé USB, puisqu'au début du script nous indiquerons à celui-ci de se déplacer dans la clé USB.

Si un fichier est contaminé, celui-ci est déplacé en quarantaine et un fichier de log est créé.

5.3. Détection des clés USB

Le script doit pouvoir tourner en arrière-plan au démarrage de la machine et doit donc pouvoir détecter le branchement de la première clé USB pour pouvoir lancer le script. Deux fonctions ont été créées, l'une pour détecter la première clé, la seconde fonction pour détecter la seconde clé.

Une boucle est créée, celle-ci indique que tant qu'aucune clé USB n'est détectée, la boucle cherche une clé. Si une clé est détectée on annonce la suite et on sort de la boucle.

Première fonction :

```
detectKeyIn(){

while true
do
        echo $var
        var=$(($var+1))
        sleep 0.2

        {
                cd /media/pi/* &> /dev/null &&
                isThere=1
                firstUSB=$((pwd))
        } || {
                isThere=0
        }

        if [ $isThere = 1 ]; then
                zenity --info --title="Analyse USB" --text="Start analyse !" --width="500"
        else
                detectKeyIn
        fi

        break
done

}
```

Deuxième function :

```
detectKeyOut(){
while true
do
	echo $var
	var=$(($var+1))
	sleep 1

	{
		cd /media/pi/* &> /dev/null &&
		isThere=1
		secondUSB=$((pwd))
	} || {
		isThere=0
	}

	if [ $isThere = 1 ]; then
		zenity --question --title="Analyse USB" --text="Confirmer la copie ?"
	else
		detectKeyOut
	fi

	break
done

}
```

5.4. Copie des fichiers

5.4.a. Copie des fichiers de la 1ère clé USB

Une fois les analyses terminées, les fichiers de la première clé sont copiés dans un dossier temporaire pour ensuite être copié sur la seconde clé.

Nous réutilisons la variable OLDIFS pour échapper les espaces des fichiers.

```
filecopy(){

echo ""

zenity --info --text="Copie des fichiers en cours"

echo ""

files=$(find -depth -name '*disarmed.pdf')

OLDIFS=$IFS

IFS=$'\n'

cp $files /home/pi/Documents/Temporaires/

        if [ ls -a /home/pi/Documents/Temporaires/ ]; then
                echo "+%D-%H-%M - Copie de la clé usb au dossier Temporaire " >>
/var/log/StationBlanche/usbkey.log
                #detectKey
        else
                zenity --info --text="Echec"
        fi

IFS=$OLDIFS

}
```

Le dossier Temporaire a été créé au préalable sur le poste.
Un fichier de log est créé lorsque les fichiers sont copiés dans le dossier.
S'il y a une erreur lors de la copie des fichiers un message d'échec s'affiche.

5.4.b. Copie des fichiers sur la 2nde clé USB

Une fois les fichiers copiés dans le dossier temporaire et la seconde clé USB détectée nous pouvons copiés les fichiers du dossier temporaire à la seconde clé USB.

OLDIF est de nouveau utilisé et un fichier de log est également créé.

```
#COPIE DES FICHIERS DU DOSSIER TEMPORAIRE SUR LA SECONDE CLE


filepaste(){

zenity --info --text="Copie en cours"

echo ""

dir=/home/pi/Documents/Temporaires

cd $dir

files=$(find -depth -name '*.disarmed.pdf')

OLDIFS=$IFS

IFS=$'\n'

cp $files /media/pi/*

	if [ ls -a /media/pi/ ];then
		zenity --info --text="Copie réussie !"
		echo "+%D-%H-%M - Copie de fichiers effectué du dossier Temporaire à clé usb" >>
/var/log/StationBlanche/usbkey.log
	else
		echo "Echec"
	fi

}
```

6. Le script

Le script est relativement simple et ne fait qu'appel aux fonctions. Une solution doit être trouvée pour l'activer au démarrage du poste, mais celle-ci n'a pas encore été trouvée.

```
#!/bin/bash

#set -x

source /home/pi/Documents/Scripts/fonctions.sh

#se déplacer dans la media pi
cd /media/pi/

# détection de la clé 1ère clé USB

detectKeyIn

# analyse PDFID

desarme

#analyse CLAMAV

virscan

#Copie de fichier de la première clé

filecopy

#détection de la seconde clé

echo "Veuillez retirer la clé"

sleep 10

if [ -d "/media/pi/*" ]; then

        echo "Vous n'avez pas retiré la clé. Veuillez la retirer"
        sleep 30
        firstkey

else
        echo "Veuillez insérer la seconde clé"
        detectKeyOut
fi

#Copie des fichiers sur la seconde clé

Filepaste
```