Préambule

Ce sujet de TP est fortement inspiré de l'ancien sujet de TP ainsi que du tutoriel anciennement présent à la page `http://techpubs.spinlocksolutions.com/dklar/kerberos.html` (RIP). Je remercie donc les auteurs du sujet précédent et Davor Ocelic, l'auteur du feu tutoriel.

Principe de fonctionnement de Kerberos

Vous trouverez sur Moodle un complément théorique sur Kerberos. Comme précédemment, ce complément n'est pas ma réalisation, j'en remercie chaleureusement les auteurs.

Mise en place du TP

Lors de ce TP, nous allons simuler une infrastructure avec deux VM :

- Un serveur Kerberos ;
- Un client contenant un serveur SSH dont l'authentification utilisera le serveur Kerberos.

L'expérience consistera à réussir une connexion SSH depuis le serveur vers le client sans passer par l'authentification directe sur le client.

Les deux VM ont chacune deux interfaces réseaux, une NATée par VirtualBox qui vous permet d'accéder à Internet et une autre présent dans un réseau interne à VirtualBox permettant d'interconnecter vos VM. Les VM sont normalement déjà configurées d'un point de vue réseau et nom de domaine, vous pouvez les ping entre elles. La machine qui joue le role de client s'appelle "client" et son nom de domaine complet est "client.master.home" tandis que la machine qui joue le rôle de serveur Kerberos s'appelle "kerberos" et son nom de domaine complet est "kerberos.master.home".

Importez les fichiers .ova présents dans le dossier que j'ai précisé au tableau sur VirtualBox.

Le mot de passe du serveur Kerberos est "tprli" tandis que le mot de passe de la machine Client est "ilrpt".

Manoeuvres

Nous allons d'abord installer les paquets pour Kerberos sur le serveur :

- krb5-kdc qui gère le KDC ;
- krb5-admin-server qui permet de manipuler la base de données Kerberos.

Pour cela vous pouvez utiliser la commande :

```
sudo apt install krb5-{admin-server,kdc}
```

Répondez aux questions avec les bonnes valeurs (royaume master.home , kerberos étant le serveur qui gère tout).

Maintenant on va créer notre royaume Kerberos (toujours sur le serveur) :

```
sudo krb5_newrealm
```

Choisissez un mot de passe approprié pour s'occuper de gérer la base de données Kerberos.

Il faut maintenant modifier le fichier /etc/krb5.conf qui contient la définition des différents royaumes Kerberos. Ajoutez le royaume qui nous correspond et enlever tous les autres :

- dans la section [realms], il faut que soit présent uniquement notre définition du KDC et de l'AS :

```
MASTER.HOME = {
    kdc = kerberos
    admin_server = kerberos
}
```

- dans la section [domain_realm], il faut que soit présent uniquement notre domaine :

```
.master.home = MASTER.HOME
master.home = MASTER.HOME
```

Note

Vous pouvez dire au serveur de rediriger les logs des différents services dans des fichiers. Rajoutez dans le fichier /etc/krb5.conf la section suivante :

```
[logging]
kdc = FILE:/var/log/kerberos/krb5kdc.log
admin_server = FILE:/var/log/kerberos/kadmin.log
default = FILE:/var/log/kerberos/krb5lib.log
```

Puis lancer les commandes

```
sudo mkdir /var/log/kerberos
sudo touch /var/log/kerberos/{krb5kdc,kadmin,krb5lib}.log
sudo chmod -R 750  /var/log/kerberos
cd /var/log
sudo tail -F kerberos/{krb5kdc,kadmin,krb5lib}.log
```

Votre terminal monitorera alors en permanence les fichiers de log de Kerberos.

Une fois le fichier krb5.conf modifié, relancer les services krb5-kdc et krb5-admin-server pour appliquer les modifications :

```
    sudo invoke-rc.d krb5-kdc restart
sudo invoke-rc.d krb5-admin-server restart
```

Les entrées dans la base de données Kerberos sont appelées des "principals" et contiennent des données telles que le nom du "principal", la clé secrete, des informations sur son expiration et d'autres données spécifiques à Kerberos. Nous allons d'abord utiliser la commande "kadmin.local" pour manipuler la base de données. Cette commande est une variante de la commande "kadmin" : "kadmin.local" ne peut s'exécuter que sur la même machine que le KDC et qu'avec les droits administrateurs. Commençons par affichier les principals existant en utilisant la commande "listprincs" :

```
    sudo kadmin.local
Authenticating as principal root/admin@MASTER.HOME with password.

kadmin.local:  listprincs

K/M@MASTER.HOME
kadmin/admin@MASTER.HOME
kadmin/changepw@MASTER.HOME
kadmin/kerberos.MASTER.HOME@MASTER.HOME
krbtgt/MASTER.HOME@MASTER.HOME
```

Les noms de principal suivent en général la syntaxe suivante :

- pour ceux associé à un utilisateur, le nom d'utilisateur séparé de son rôle par un / (exemple : myadmin/admin).
- pour ceux associé à un service, le nom du service séparé de l'hôte concerné par un / (exemple : ldap/toto.master.home).

le tout suivi d'un @NOM_DU_DOMAINE . Modifions maintenant le fichier /etc/kr5kdc/kadm5.acl qui permet de définir les droits d'accès à la base de données Kerberos. Décommentez la ligne :

```
    */admin *
```

cela permettra aux utilisateurs avec un rôle admin Kerberos d'avoir tous les droits.

Redémarrez le service pour prendre en compte le changement :

```
    sudo invoke-rc.d krb5-admin-server restart
```

Retournons sur l'outil "kadmin.local" pour pouvoir ajouter des "policies" Kerberos et qui seront associés à des "principals". Nous allons attribuer une longueur minimum au mot de passe et un nombre minimum de type différent de caractères qui doit être présent dans le mot de passe pour les principals admin, host, service et user :

```
    sudo kadmin.local
Authenticating as principal root/admin@MASTER.HOME with password.

kadmin.local:  add_policy -minlength 8 -minclasses 3 admin
```

```
kadmin.local:  add_policy -minlength 8 -minclasses 4 host
kadmin.local:  add_policy -minlength 8 -minclasses 4 service
kadmin.local:  add_policy -minlength 8 -minclasses 2 user
```

Depuis avant, nous manipulons la BDD Kerberos de manière locale avec le principal root/admin. Soyons plus propre et créeons un utilisateur qui sera admin et que nous pourrons utiliser depuis n'importe quel endroit. Toujours sur kadmin.local (remplacez tim par un nom d'utilisateur qui vous convient) :

```
    kadmin.local:  addprinc -policy admin tim/admin

Enter password for principal "tim/admin@MASTER.HOME":
Re-enter password for principal "tim/admin@MASTER.HOME":
Principal "tim/admin@MASTER.HOME" created.
```

Testons la connexion avec le compte :

```
    kadmin -p tim/admin
Authenticating as principal tim/admin@MASTER.HOME with password.

Password for tim/admin@MASTER.HOME: PASSWORD

kadmin:  listprincs

K/M@MASTER.HOME
tim/admin@MASTER.HOME
kadmin/admin@MASTER.HOME
kadmin/changepw@MASTER.HOME
kadmin/kerberos.MASTER.HOME@MASTER.HOME
krbtgt/MASTER.HOME@MASTER.HOME
```

On peut désormais créer un utilisateur non privilégié (remplacez toto par un nom d'utilisateur qui vous convient) :

```
    kadmin:  addprinc -policy user toto

Enter password for principal "toto@MASTER.HOME":
Re-enter password for principal "toto@MASTER.HOME":
Principal "toto@MASTER.HOME" created
```

Le compte toto existe dans la base de données Kerberos mais ne correspond en réalité pour l'instant à personne sur notre machine. Créons cet utilisateur sans mot de passe pour empêcher d'éventuel bug.

```
    sudo adduser --disabled-password toto
```

Maintenant, nous pouvons commencer à installer le serveur SSH sur notre machine client.

```
sudo apt install openssh-server
```

Tout service peut être lié à Kerberos soit de manière native lors de l'implémentation du service par ses créateurs soit en déléguant l'authentification au système PAM (Pluggable Authentication Modules). Nous allons utiliser PAM puisque le service SSH le supporte. Depuis le client, nous pouvons gérer à distance l'AS Kerberos avec la commande kadmin. Pour cela, nous allons installer le paquet de gestion et la configuration qui convient (càd la même que sur le serveur, ici) :

```
sudo apt install krb5-admin-server
sudo vim|gedit|nano|emacs|vi|votreediteurdetexte /etc/krb5.conf
```

Redémarrez le service krb5-admin-server et vérifiez que vous pouvez vous connecter avec le compte que vous avez créé et qui possède un role admin depuis le client en utilisant la commande kadmin.

Maintenant, il faut déclarer le service SSH dans la BDD Kerberos. Les services type SSH ou telnet ont par convention le nom de princpal "host".

```
kadmin: addprinc -policy service -randkey host/client.master.home
Principal "host/client.master.home@MASTER.HOME" created.
```

Comme indiqué dans la section théorique, Kerberos utilise des clés partagés entre les deux acteurs communicants. Il faut donc un exemplaire de la clé dans la BDD et un exemplaire de la clé sur l'hôte hébergeant le service. La commande ktadd permet d'exporter dans un fichier cette clé :

```
kadmin:  ktadd -k /etc/krb5.keytab host/client.master.home
Entry for principal host/client.master.home with kvno 2, encryption type
aes256-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/client.master.home with kvno 2, encryption type
aes128-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5.keytab.
```

> **Note**
>
> Chaque appel à ktadd va en réalité re-générer une nouvelle clé et invalider l'ancienne donc faites attention si vous utilisez cette commande plusieurs fois

On modifie la configuration SSH pour utiliser PAM et les bons paramètres (fichier /etc/ssh/sshd_config) et on redémarre le service. :

```
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
GSSAPIKeyExchange yes
UsePAM yes

sudo invoke-rc.d ssh restart
```

> **Note**
>
> Les configurations PAM sont assez sensibles et la moindre erreur peut bloquer l'accès complet à votre machine. Dans une situation de serveur en production, je vous conseille de copier la configuration initiale de PAM.
>
> ```
> cp -a /etc/pam.d /etc/pam.d.backup
> ```

Nous pouvons attaquer la configuration de PAM. D'abord, installons le module Kerberos pour PAM (sur le client).

```
sudo apt install libpam-krb5
```

L'installation du paquet a normalement déjà configuré PAM pour utiliser Kerberos et cette configuration correspond à ce qu'on souhaite. Vérifiez que les fichiers suivants contiennent la même configuration :

```
    /etc/pam.d/common-account
account [success=1 new_authtok_reqd=done default=ignore]        pam_unix.so
account requisite                       pam_deny.so
account required                        pam_permit.so
account required                        pam_krb5.so minimum_uid=1000

    /etc/pam.d/common-auth
auth    [success=2 default=ignore]      pam_krb5.so minimum_uid=1000
auth    [success=1 default=ignore]      pam_unix.so nullok_secure try_first_pass
auth    requisite                       pam_deny.so
auth    required                        pam_permit.so
autoh   optional                        pam_cap.so

    /etc/pamd.d/common-password
password        [success=2 default=ignore]      pam_krb5.so minimum_uid=1000
password        [success=1 default=ignore]      pam_unix.so obscure use_authtok try_firs
password        requisite                       pam_deny.so
password        required                        pam_permit.so
```

```
    /etc/pam.d/common-session
session [default=1]                     pam_permit.so
session requisite                       pam_deny.so
session required                        pam_permit.so
session optional                        pam_krb5.so minimum_uid=1000
session required        pam_unix.so
```

Si vous avez modifié ces fichiers, redémarrez le service SSH.

Sur le serveur, installez le client SSH :

```
    sudo apt install openssh-client
```

Tout est pratiquement prêt. Il nous reste à récupérer un ticket Kerberos pour pouvoir nous authentifier. Toujours sur le serveur, nous allons utiliser la commande kinit pour le récupérer.

```
     kinit toto
```

On vérifie avec la commande klist que l'on a bien un ticket.

```
    klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: toto@MASTER.HOME

Valid starting     Expires            Service principal
16/12/2022 15:57:17  17/12/2022 01:57:17 krbtgt/MASTER.HOME@MASTER.HOME
    renew until 17/12/2022 15:57:14
```

Maintenant que nous avons notre ticket, il ne reste plus qu'à nous connecter :

```
    ssh toto@client
Linux client 5.10.0-19-amd64 #1 SMP Debian 5.10.149-2 (2022-10-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Dec 16 16:10:19 2022 from 192.168.100.1
toto@client:~$
```