Rapport d'implementation:

1) La liste des choix faits:

Plusieurs décisions ont été prises avant l'implémentation de ce serveur. Premièrement l'utilisation d'un serveur d'authentification Kerberos plutôt que d'une autre solution tel que l'authentification à deux facteurs. Deuxièmement, pour des question de transparence et pour limiter les coûts de l'opération, le serveur mis en place utilise des solutions open source. Notamment Ubuntu-Serveur en tant qu'OS (version 18.04.01) et les package du MIT pour faire fonctionner Kerberos sur une machine Linux.

Ensuite, nous avons pris la décision de ne pas implémenter nous même le service de connection intégré avec l'application Web. Nous avons configuré les serveurs de sorte à ce qu'il soit possible d'utiliser le système d'authentification dans une application web. Mais la manière dont l'équipe de développeur décident de mettre cela en place est hors de notre domaine. Nous avons suggéré des librairies qui permettent d'effectuer une connection avec authentification utilisant un serveur Kerberos a la fin de ce document. Nous avons aussi mis en place un script qu'il suffit de lancer pour configurer un KDC sur un serveur Ubuntu.

Nous partons du principe que ce script sera utilisé sur un nouveau serveur séparé du serveur d'application suffisamment puissant pour les besoins de notre entreprise (en se basant sur le nombre de connexion horaire, par exemple). On suppose également qu'il n'y aura pas de soucis au niveau du réseau en considérant un réseau d'entreprise déjà efficace. Nous suggérons l'utilisation d'un SSD comme stockage permanent pour permettre un temps d'accès moindre au niveau de la base de donnée. La séparation physique entre le centre de distribution de clefs et le serveur d'application permet un renforcement de la sécurité car il faut dorénavant compromettre deux serveurs pour reussire a s'authentifier sans permission.

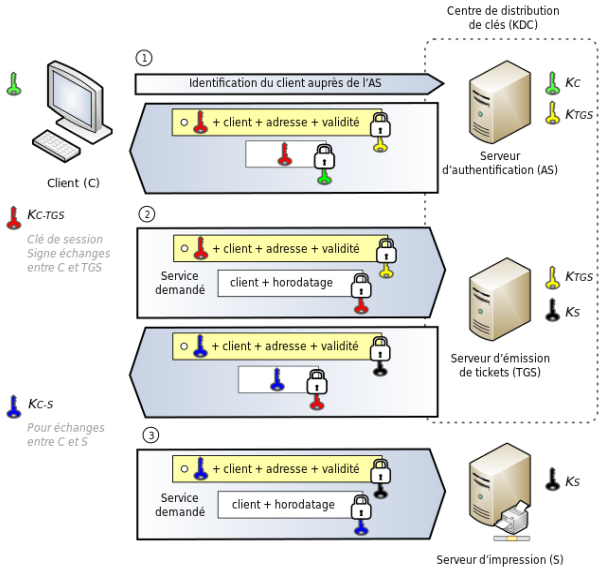
Nous n'avons pas inclu dans cette implémentation la création d'un serveur secondaire utilisé comme backup en cas de DDOS. Il est possible de créer un second serveur plutôt simplement pour atténuer les dégâts lors d'une attaque de type DDOS sur notre KDC, mais notre entreprise ne pouvant pas se permettre d'acheter un deuxième serveur uniquement utilisé comme backup, nous n'avons pas inclu ces démarches dans nos scripts. Si dans le futur notre

entreprise désire se procurer un autre centre de distribution de clef, la documentation Ubuntu de Kerberos propose une solution permettant de mettre en place un tel serveur.
Le dernier choix effectué lors du développement de cette solution était de ne pas imposer des identifiants admin prédéterminé lors de la configuration du serveur. Lors du démarrage du script, celui ci demande à l'utilisateur de choisir un nom d'utilisateur et un mot de passe. Ceux ci sont les seul (par défaut) correspondant à un compte administrateur et il est indispensable de les conserver.
Serveurs séparé tournant sur ubuntu spécifiquement pour connection
Server n'a pas besoin d'être très puissant.

2) <u>Une description des aspects techniques</u>

Lors du lancement du script fourni (install.sh), la machine va tout d'abord télécharger et mettre à jour ses librairies. Une fois que cela est fait, un formulaire va s'afficher demandant le nom du realm par défaut (si un utilisateur est créé sans spécifier un realm auquel il appartient).
Ensuite le formulaire va demander de définir les noms de host dans le realm créé précédemment. Finalement le formulaire va demander de définir le serveur administratif du realm (un des serveurs défini dans l'étape précédente).
Puis le script va attendre une commande de votre part, celle ci est affiché juste au dessus: addprinc <user>/admin. Cette commande va configurer un administrateur pour la base de donné associé à cette KDC. Vous devrez ensuite configurer un mot de passe.

Schema authentication Kerberos:



Nous suggérons la lecture de la documentation MIT:
https://web.mit.edu/kerberos/

Le centre de distribution des clés (KDC) tourne sur le nouveau serveur. Lorsqu'un client s'y connecte et réussi à s'authentifier, il obtient un ticket, qu'il peut donner au serveur d'application pour prouver qu'il a le droit d'accéder aux ressources qu'il essaye d'accéder.
En bref, Le package krb5-kdc permet la création de nouveau realm et initialise un KDC. Le package krb5-admin-server permet d'administrer le serveur. Et le script d'installation utilise ses package pour initialiser le KDC sur notre serveur.

Afin de configurer la machine "cliente" qui permettra de faire le lien entre le service web et les données protégées, il suffit de lancer le script donné (client.sh) sur le serveur concerné.

3) Nom des fichiers de configuration et description de leur contenu

Côté serveur KDC : - "kdc.conf" paramètre le royaume concerné, utilisé notamment par le daemon kadmind pour manager la base de donnée. Ce fichier contient par exemple les informations suivantes : la location de la base de donnée, la location du fichier des ACL sur les principals ou encore la durée maximale d'un ticket. En bref, ce fichier contient les informations principales concernant la base de donnée du serveur Kerberos.
- "kadm5.acl" ce fichier permet de gérer tout ce qui est permission/droit d'accès. Il est également utilisé par le daemon kadmind pour savoir quels principals sont autorisés à effectuer des actions administratives.

Côté serveur KDC et client : -"krb5.conf" ce fichier contient toutes les informations sur les différents royaumes existants. En particulier il donne les FQDN des machines sur lesquelles tourne le KDC ou encore le serveur d'administration. Il définit également quel est le royaume utilisé par défaut. Et surtout, il fournit la traduction d'un nom d'hôte(FQDN ou DN) vers un royaume Kerberos.

4) Commandes utiles

Pour l'administration de la base de données, les 2 commandes principales sont *kadmin* et *kadmin.local*. *kadmin.local* ne peut être utilisé que sur la

machine où la base de donnée est présente. kadmin en revanche peut être utilisée à distance et fait appel alors au démon kadmind qui administre alors localement la base de donnée. Dans les 2 cas ces commandes fournissent une interface à l'administrateur pour gérer le kdc. Il peut notamment ajouter/supprimer/modifier des principals selon les droits d'accès qui lui sont alloués.
Une commande intéressante pour se débarrasser des mots de passe est *ktadd*. Celle-ci nécessite de spécifier un fichier ".keytab" et le principal dont on souhaite remplacer le mot de passe par un ensemble de clés. Celles-ci sont donc stockées dans le fichier ".keytab" spécifié qui peut être partagé au client pour que celui-ci n'ait plus à s'identifier en utilisant son mot de passe.
La commande pour récupérer les tickets à partir du KDC est *kinit*. Celle-ci peut spécifier un principal ou laisser celui par défaut. Le mot de passe est alors demandé, sauf si l'on utilise le fichier keytab permettant l'authentification. Le ticket est alors stocké dans un cache situé le plus souvent dans /tmp.
La commande *klist* permet de lister les tickets présents en cache et les informations à propos de ceux-ci : leur date d'expiration, les principals concernés, etc..

5) Processus tournant en mémoire quand votre service est en cours d'exécution

Kadminserver : daemon pour administrer la base de donnée du master KDC.
Krb5kdc : daemon pour la gestion des tickets du KDC.
Kpropd : daemon pour gérer la base de donnée du serveur esclave s'il en existe un. Ce daemon écoute les requêtes en provenance du KDC principal.
Kdb5_util : daemon pour effectuer les opérations de maintenance sur la base de donnée.

6) Ports utilisés

Ports utilisés par le KDC : 750, 88

7) Scripts de démarrage

Tout ce qui est nécessaire pour démarrer le serveur est géré automatiquement par Ubuntu. Il n'y a donc pas de script de démarrage spécifique pour ce système.

8) Une liste détaillée des étapes d'implémentation (depuis les packages à installer ou à compiler, jusqu'aux fichiers à modifier)

Lancer le script install.sh avec les droits administrateur (sudo bash install.sh).
Ensuite, il faut remplir les formulaires et configurer un administrateur du KDC (instructions donnés dans la partie 2)).
Demander à tous les utilisateurs de mettre leurs mots de passe a jours.
Modifier l'application web pour permettre de se connecter avec ce serveur.
Une fois que tous les clients ont fait le changement, mettre en place la nouvelle application Web.

9) Les éventuelles opérations de maintenance à effectuer

Updates de Ubuntu / updates regulieres.
Changer les mots de passe des utilisateurs régulièrement.

10) Les fichiers de configuration commentés correspondants, ainsi que les éventuels scripts développés par vos soins

Veuillez trouver les scripts dans ce dossier.
Sinon, vous pouvez les obtenirs en clonant:
https://github.com/Jack-Portal/AdminSysKerberos

RESSOURCES:

MIT
https://web.mit.edu/kerberos/
Ubuntu help
https://help.ubuntu.com/lts/serverguide/kerberos.html.en
Tuto Francais
https://wiki.deimos.fr/Kerberos_:_Mise_en_place_d%27un_serveur_Kerberos

Wrapping LIBRAIRIES:

Python
http://python-notes.curiousefficiency.org/en/latest/python_kerberos.html
https://www.calendarserver.org/PyKerberos.html
JS
https://www.npmjs.com/package/kerberos
Script pour automatiser la création de nouveaux utilisateurs:
https://docs.oracle.com/cd/E36784_01/html/E37126/aadmin-192.html
AUTRE
https://blog.kloud.com.au/2015/06/04/kerberos-web-application-configuration/

OS:

Ubuntu Serveur
https://www.ubuntu.com/download/server