Kerberos, le SSO système

Benoit Métrot

Université de Poitiers

ANF Les systèmes d'authentification dans la communauté ESR : étude, mise en œuvre et interfaçage dans un laboratoire de Mathématique
Angers, 22 - 26 septembre 2014

Plan

Progression

Qu'est-ce que Kerberos

C'est un système d'authentification

- Sécurisé
- Authentification unique
- Basé sur un tiers de confiance
- Authentification mutuelle

Qu'est-ce que Kerberos

C'est un système d'authentification

- Sécurisé
  → Aucun mot de passe ne circule sur le réseau
- Authentification unique
  → Une seule saisie pour plusieurs ressources
- Basé sur un tiers de confiance
  → Serveur central connu de tous
- Authentification mutuelle
  → Le client est bien celui qu'il prétend être
  → Le serveur est bien celui qu'il prétend être

Objectifs

- Améliorer la sécurité
- Augmenter le confort d'utilisation
- Centraliser l'authentification
- Sécuriser l'authentification sur des réseaux de non-confiance
- Implémentation des *three A's*
  - Authorization
  - Auditing
  - Authentication

Objectifs

- Améliorer la sécurité
- Augmenter le confort d'utilisation
- Centraliser l'authentification
- Sécuriser l'authentification sur des réseaux de non-confiance
- Implémentation des *three A's*
  - Authorization
  - Auditing
  - Authentication

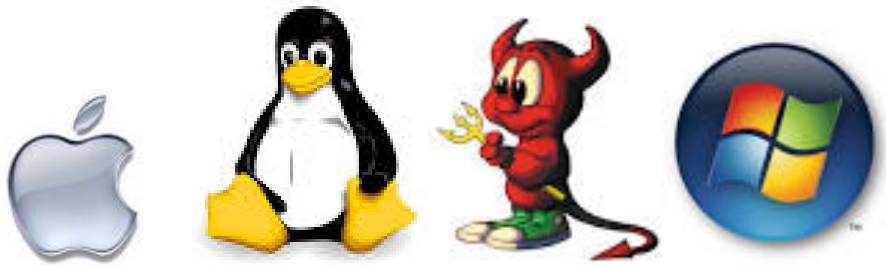
Kerberos n'est pas...

- ...une base d'identification d'utilisateurs
- ...un entrepot d'informations des utilisateurs
- ...un système de détection d'intrusion
- ...le gardien de votre système d'information

Périmètre

- Implanté au niveau système d'exploitation
- Transversal à différents services
- Inter-opérabilité des systèmes
- Politique de complexité et de durée de vie des mots de passe
- Applicable aux applications webs (support du serveur web)

Disponible sur :



Principe de fonctionnement

- Protocole client-serveur avec un tiers de confiance
- Le tiers de confiance est le serveur d'authentification
- Emission de tickets chiffrés
- Les tickets prouvent l'identité des utilisateurs et des ressources du réseau

Historique I

- Protocole développé par le MIT (Athena project)
- V1 à V3 sont restées internes au MIT
- Version 4 publiée le 24 janvier 1989
- Exportation du code interdite par les Etats-Unis (contient implémentation DES)
- MIT développa alors une version spécialisée pour l'exportation
- Version 5 publiée en 1993 (RFC 1510) ajoute :
  - la délégation et transfert d'autorisation
  - l'authentification entre domaines (relation d'approbation)
  - des mécanismes de chiffrement supplémentaires
  - la pré-authentification

Historique II

- Naissance d'une nouvelle implémentation en Suède, *Heimdal*, libre de toute restriction cryptographique (juillet 1997)
- En 2000, les restrictions cryptographiques des Etats-Unis sont levées sur les produits OpenSource

Progression

Royaume

Définition

Les utilisateurs, les ressources et les machines participant à un même système d'information sont regroupées au sein d'un royaume Kerberos.

- Toutes les entités d'un royaume partagent le même serveur d'authentification
- Le royaume définit la frontière de validité de l'authentification
- Le nom d'un royaume est souvent le même que le nom DNS (en majuscules)
  - mit.edu $\rightarrow$ MIT.EDU
  - crash.mit.edu $\rightarrow$ TEST.ATHENA.MIT.EDU
  - vbox.tp $\rightarrow$ VBOX.TP

Principal

Définition

Toute entité du royaume (ordinateur, utilisateur, service s'exécutant sur un serveur) est associée avec un principal. Chaque principal est désigné par un nom et contient un secret.

- Le nom d'un principal est unique dans le royaume
- Le secret est un mot de passe ou une passphrase

Nommage d'un principal

UserOrService /instance @ ROYAUME.TLD

- Nom d'utilisateur ou de service
- Instance du service ou de l'utilisateur
  - $\rightarrow$ précédé d'un *slash*
  - $\rightarrow$ optionnel
- Nom du royaume

Exemples :

- john@VBOX.TP
- john/admin@VBOX.TP
- host/filer.vbox.tp@VBOX.TP
- http/web.vbox.tp@VBOX.TP

3 types de principals

- Pour un utilisateur :
  - *john@VBOX.TP* $\rightarrow$ utilisateur standard
  - *john/admin@VBOX.TP* $\rightarrow$ utilisateur administrateur
- Pour les ordinateurs :
  - le nom de service est fixé à *host*
  - l'instance prend le nom DNS complètement qualifié
  - $\rightarrow$ host/clinux.vbox.tp@VBOX.TP
- Pour les services :
  - le nom de service correspond au protocole : *nfs*, *http*...
  - l'instance prend le nom DNS completèment qualifié
  - $\rightarrow$ nfs/clinux.vbox.tp@VBOX.TP

Key Distribution Center (KDC)

- 3 composants logiques
  - une base de données des principals
  - le *Authentication Server*
  - le *Ticket Granting Server*
- Au moins un KDC par royaume
- Pas de mécanisme de réplication standardisé : spécifique à chaque implémentation

Key Distribution Center (KDC)

Base de données des principals

- Contient tous les principals du royaume et les secrets associés
- Les secrets sont chiffrés sur disque (chiffrement symétrique)
- Informations complémentaires associées :
  - durée de vie des mots de passe
  - dernier changement de mot de passe
  - stratégie de mot de passe (complexité, durée)
- Enregistrée, en standard, dans un fichier spécialisé $\rightarrow$ système de base de données légère
- Possibilité d'utiliser un annuaire LDAP comme entrepot de stockage

Key Distribution Center (KDC)

Authentication Server

- Acronyme = AS
- Délivre des *Ticket Granting Ticket*

Key Distribution Center (KDC)

Ticket Granting Server

- A ne pas confondre avec le *Ticket Granting Ticket*
- Acronyme = TGS
- Délivre des tickets de service

Tickets

Définition

Un ticket est une structure de données chiffrée délivrée par le KDC. Il sert à prouver son identité vis à vis d'un utilisateur ou d'une ressource.

Il contient

- Le nom du *principal* du demandeur
- Le nom du *principal* de la ressource demandée
- Date de validité (début et fin)
- Adresses IP à partir desquelles il est utilisable
- Un secret partagé (chiffré) pour les communications utilisateurs $\leftrightarrow$ applications (clé de session)

Exemples de Tickets

- *Ticket Granting Ticket (TGT)*
  - Premier ticket remis au client
  - Emis par l'*Authentication Server*
  - Chiffré avec le mot de passe de l'utilisateur
  - Utilisé pour demander des tickets de services
  - Nommage = *krbtgt/VBOX.TP@VBOX.TP*
- *Ticket Service (TS)*
  - Contient le TGT du client demandeur
  - Contient le nom du principal de la ressource cible
  - Test de validité de la demande effectué sur le TGT par le TGS [1]

1. Ticket Granting Server

Credential Cache

- C'est un trousseau de stockage de tickets
- Contient l'ensemble des tickets obtenus par l'utilisateurs au cours de sa session
- Dans l'implémentation MIT c'est un fichier du /tmp
- Associé à un principal utilisateur
- Intégre les dates de validité des tickets (début et fin)

Keytab

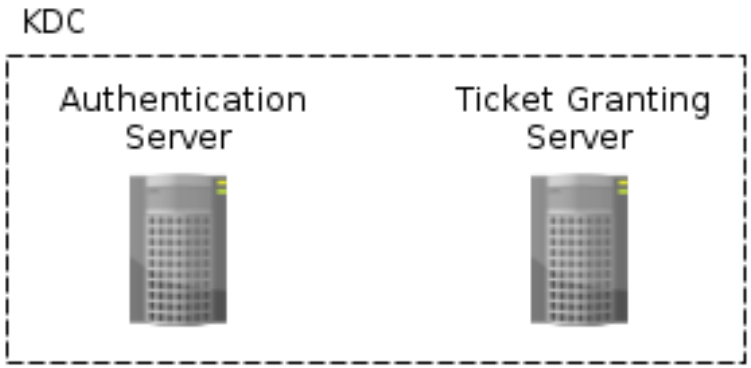
- Sert à enregistrer un secret Kerberos
- Contient le nom du *principal* (client)
- C'est un fichier, attention aux permissions !
- Utilisé pour les demandes de tickets sans saisie du secret (applications)

Protocole

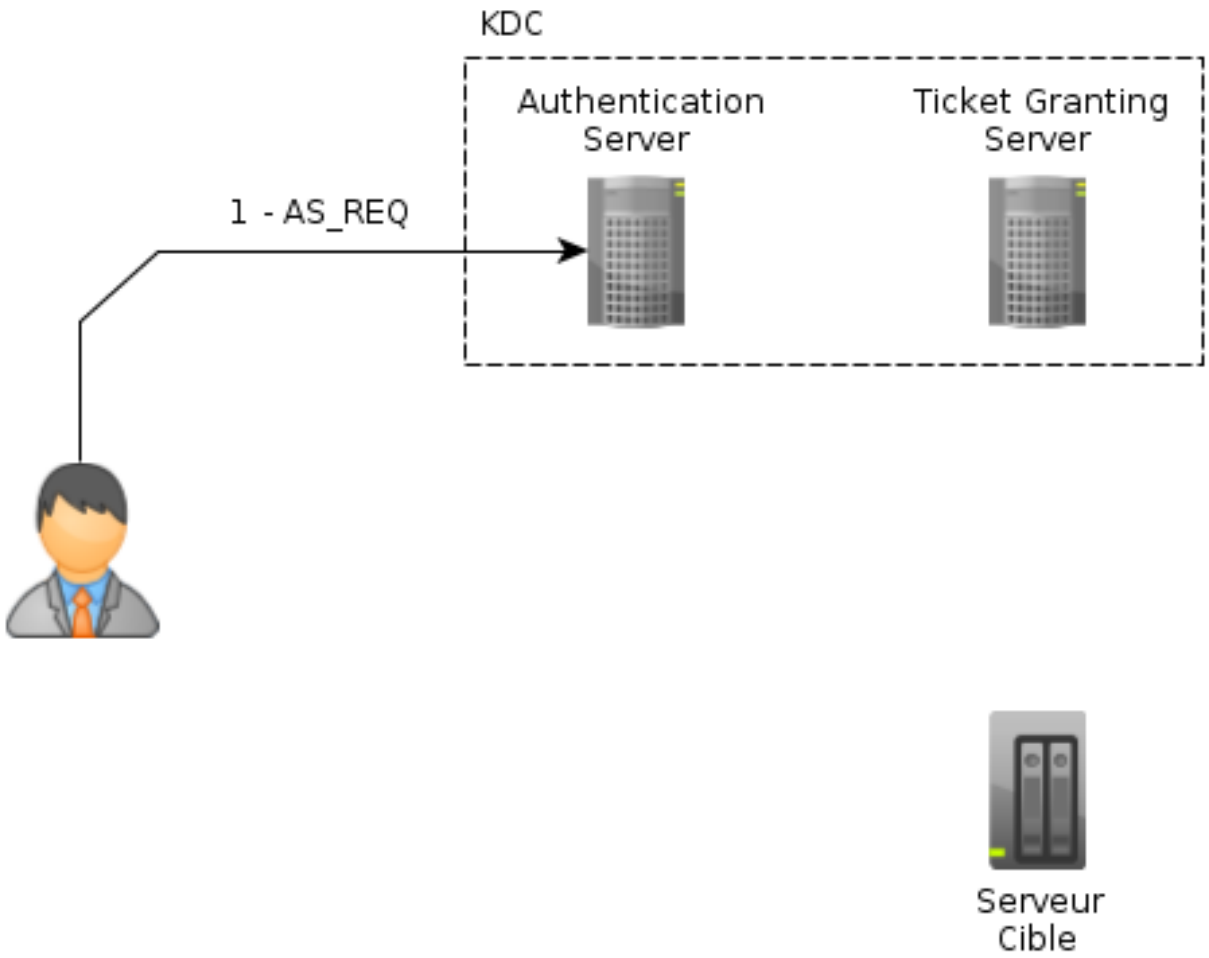
- Kerberos v5 documenté dans la RFC1510
- Basé sur l'algorithme proposé par *Roger Needham* et *Michael Schroder* en 1978
  Using encryption for authentication in large networks of computers
- Principe
  - Distribution d'une clé de chiffrement
  - Entre deux participants
  - Avec une durée de validité courte
  - Sans envoyer de mot de passe (même chiffré)

Progression

Les participants

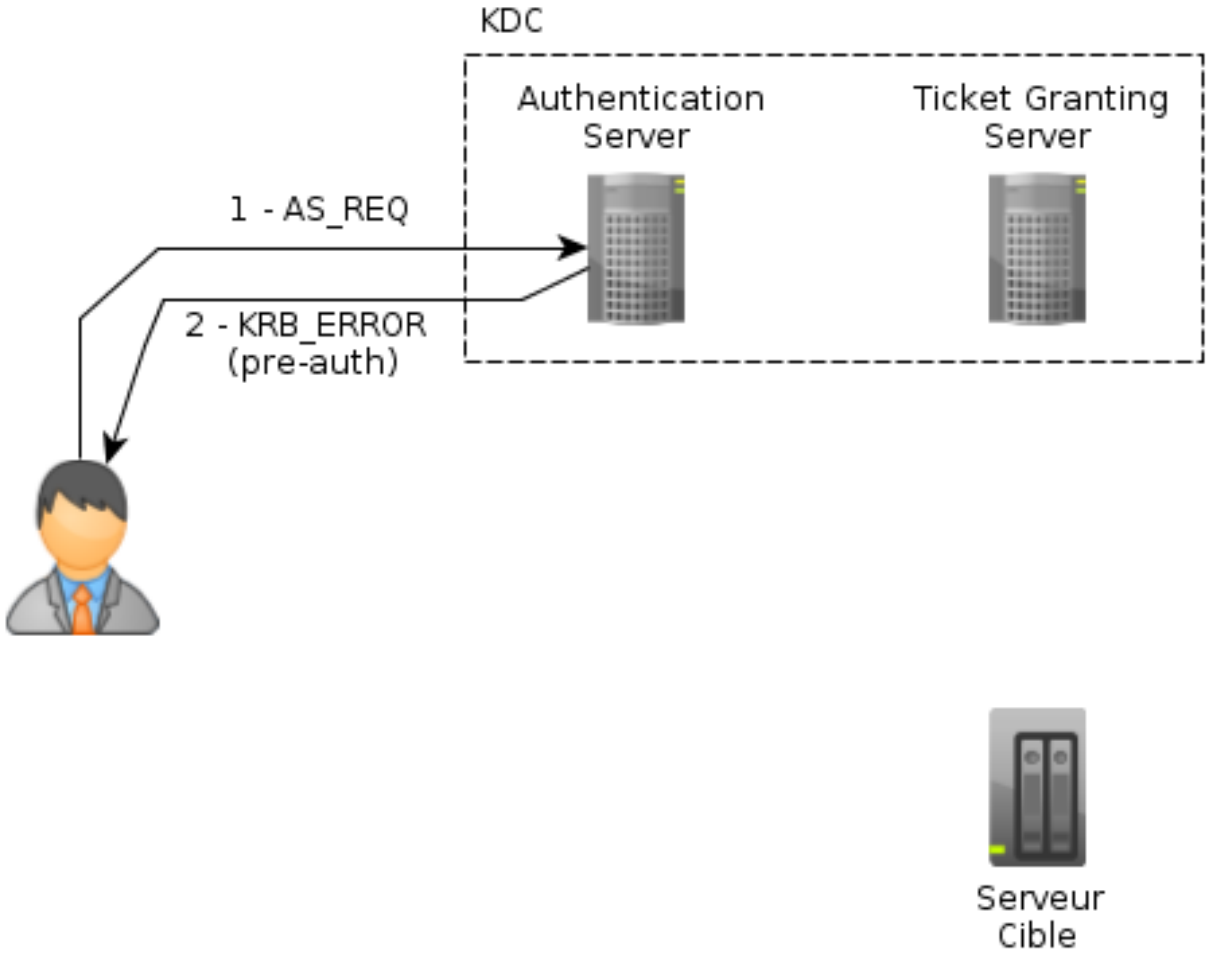


Connexion au royaume



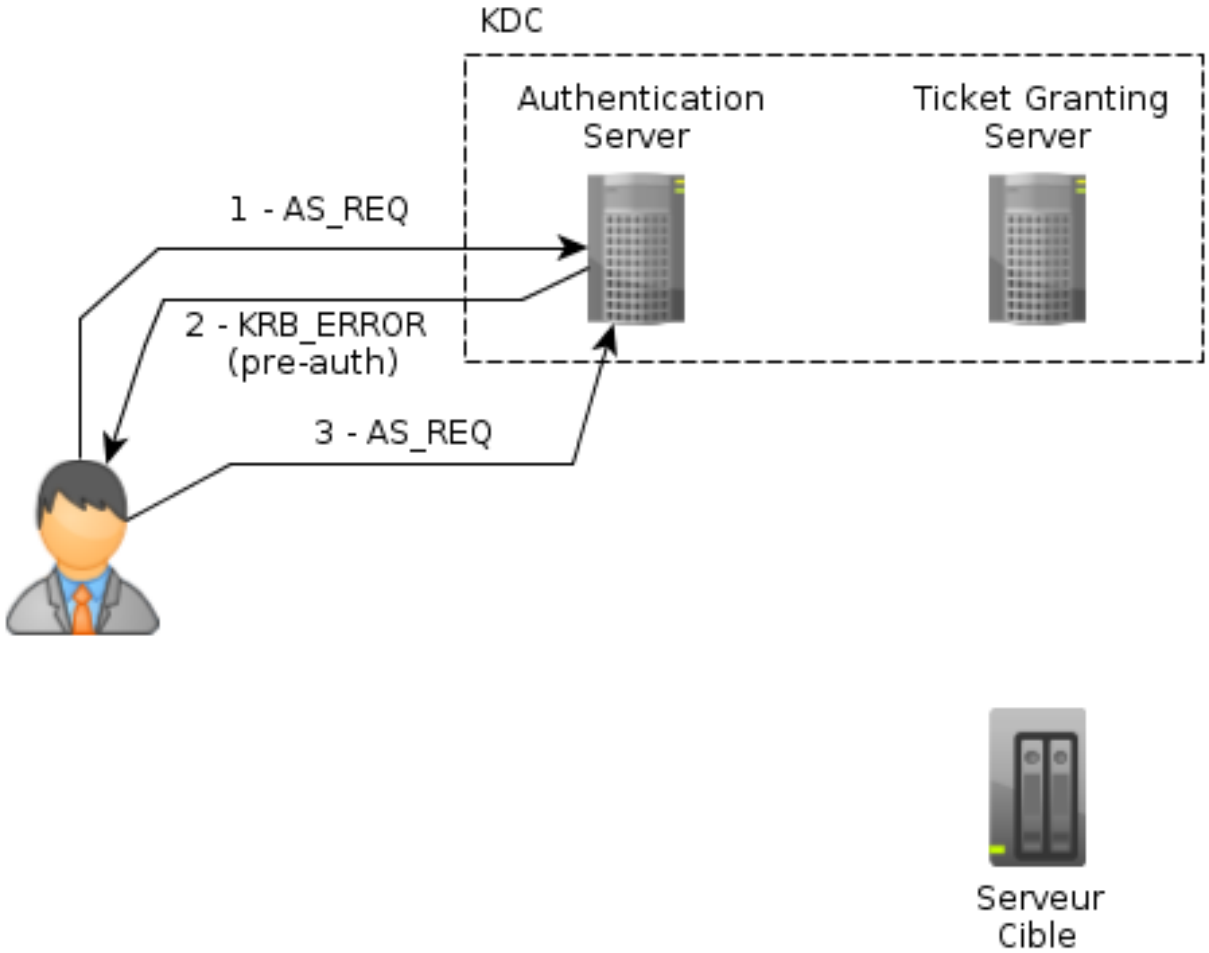
Demande d'un TGT au KDC

Connexion au royaume



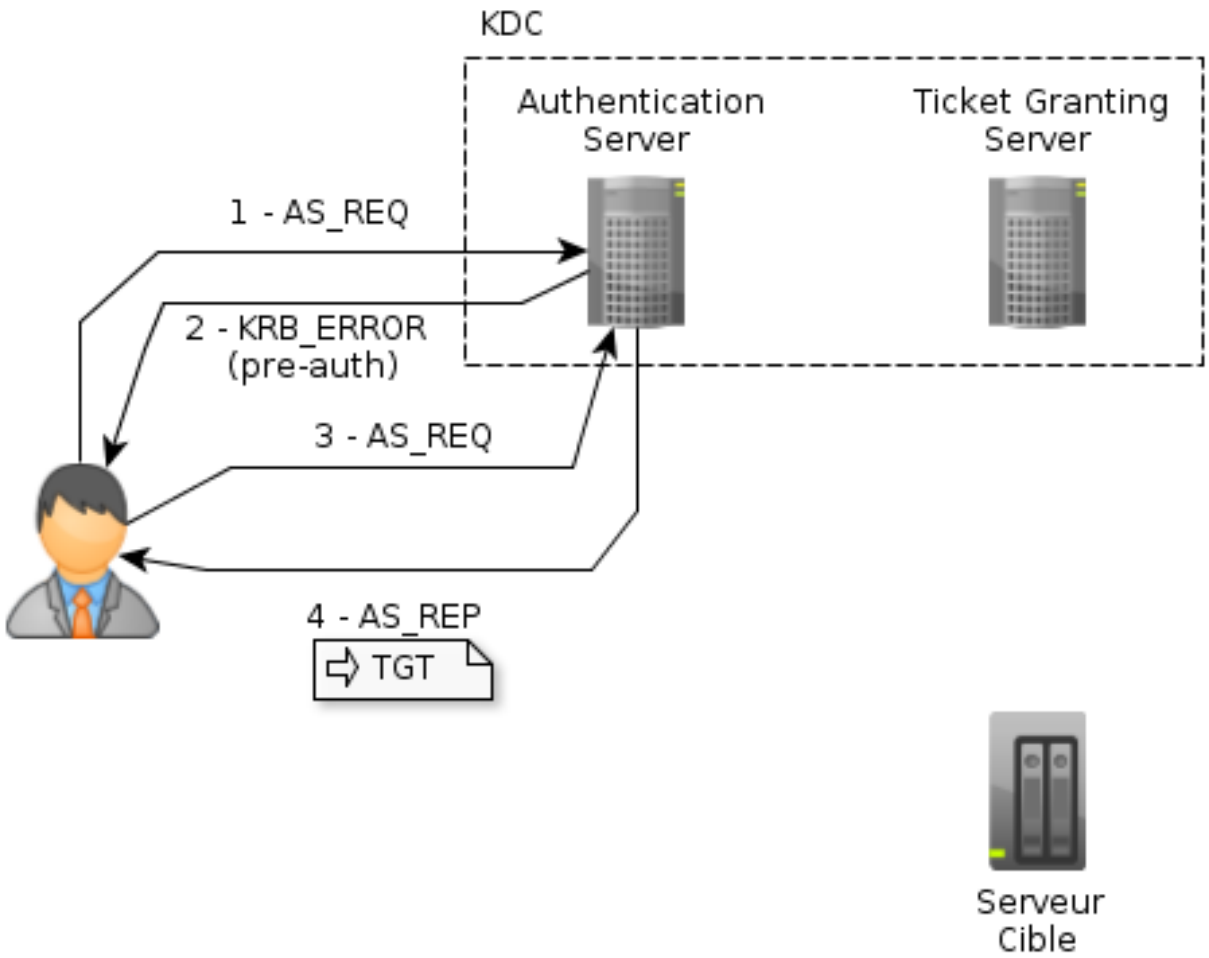
Le KDC refuse la première demande en exigeant une pré-authentification

Connexion au royaume



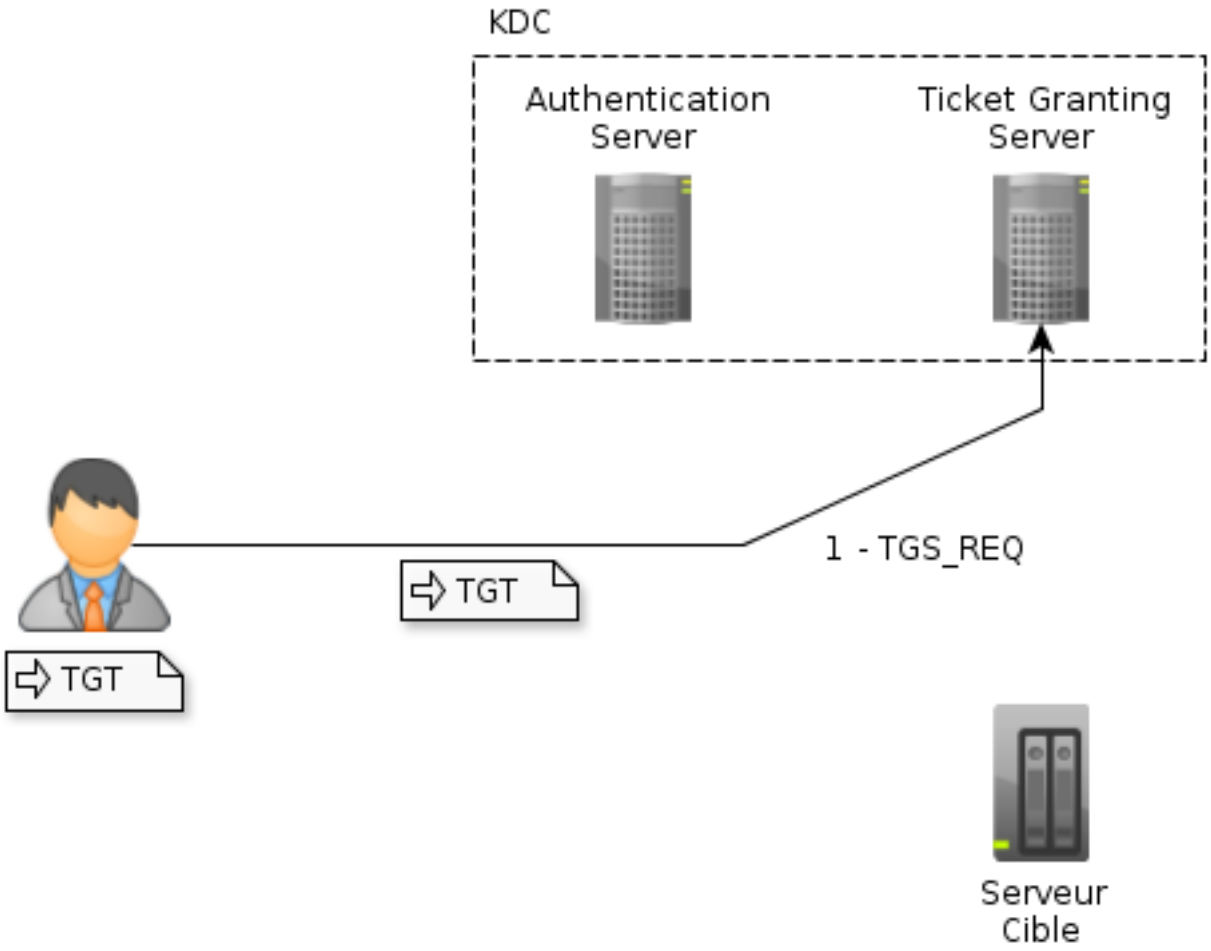
Nouvelle demande d'un TGT avec les attributs nécessaires à la pré-authentification

Connexion au royaume



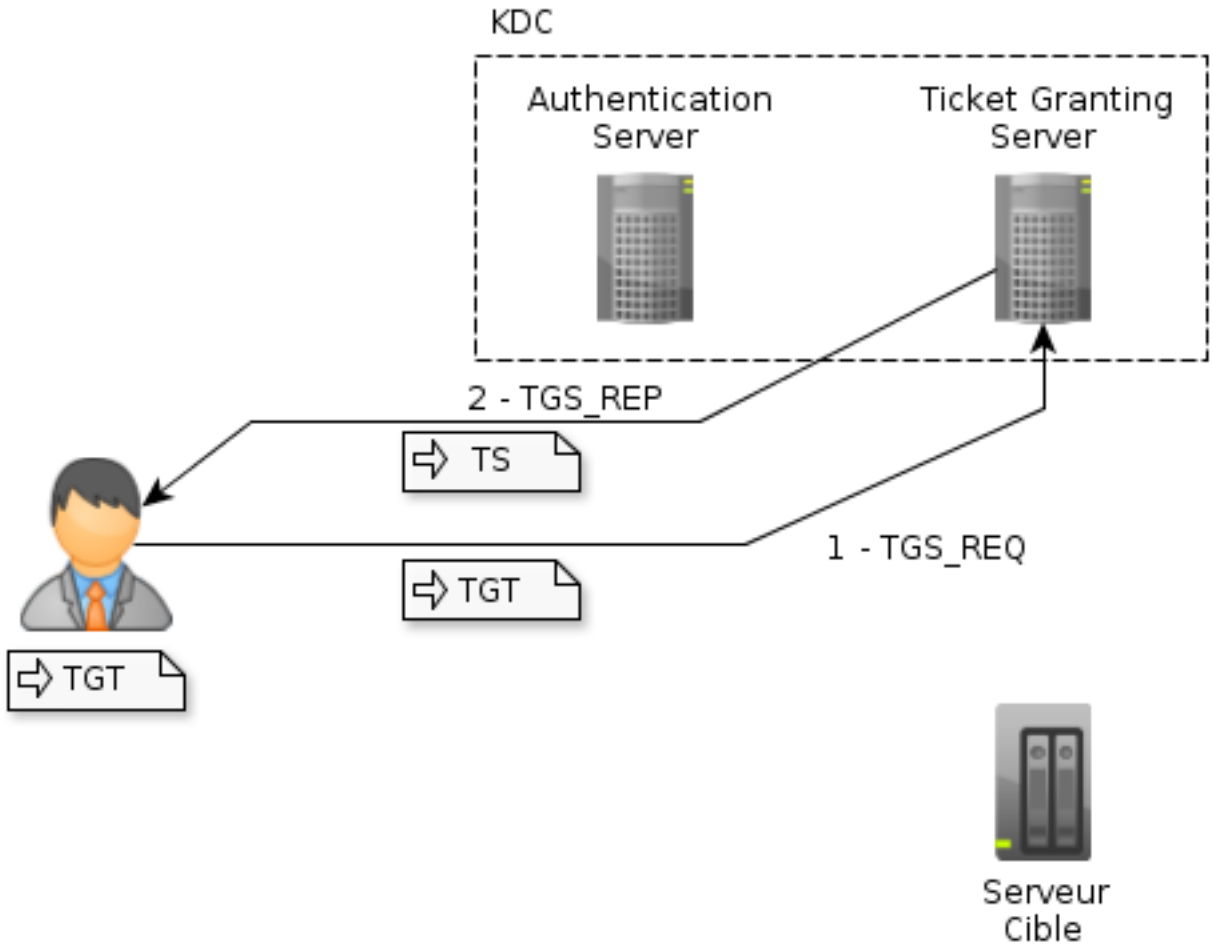
Emission du TGT (encodé avec un secret partagé)

Accès à une ressource



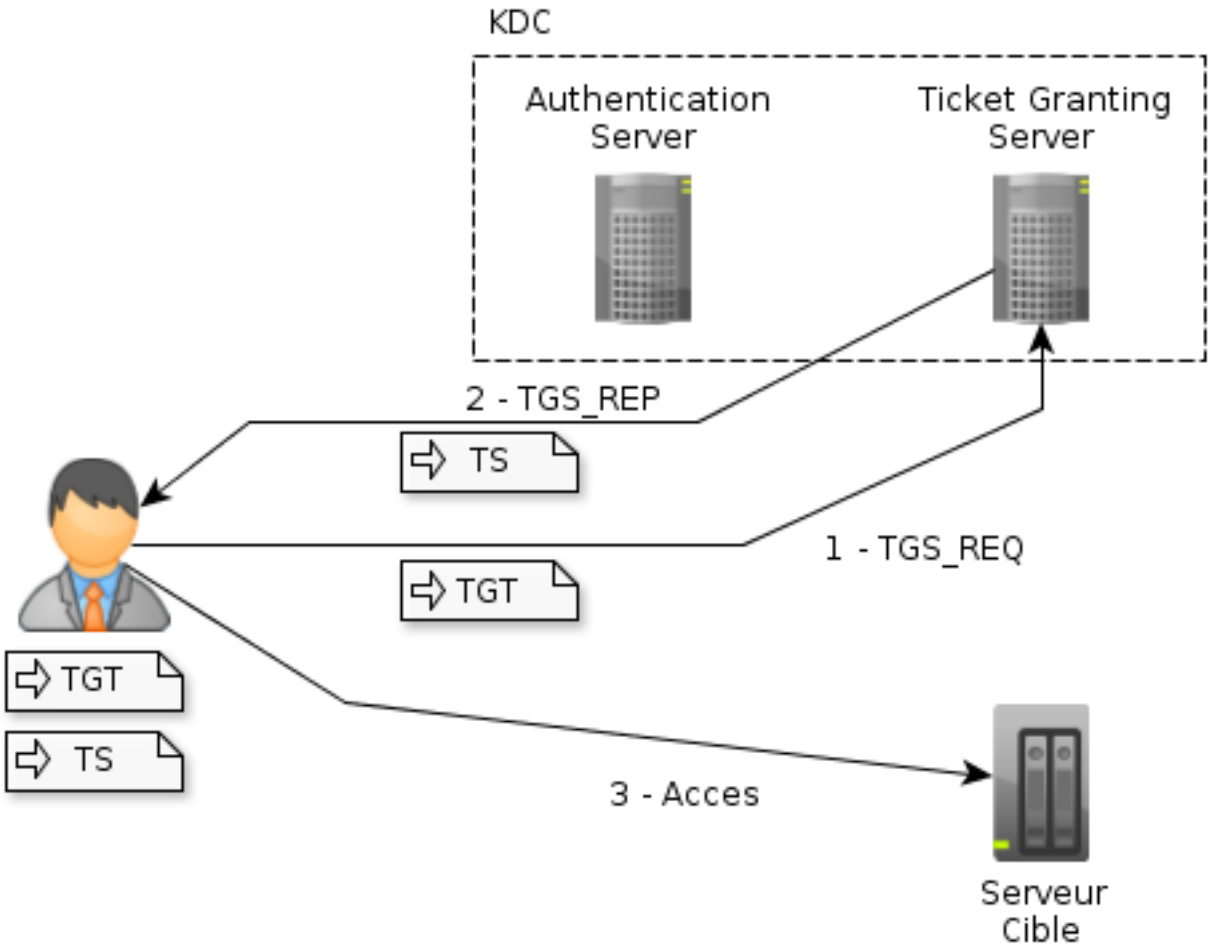
Demande d'un ticket de service (en fournissant le TGT) pour le serveur cible

Accès à une ressource



Si le TGT est valide, émission du ticket de service

Accès à une ressource



Demande d'accès à la ressource cible à l'aide du ticket de service

Bibliothèque GSSAPI

- Conçue pour les applications client/serveur souhaitant sécuriser leur echanges avec Kerberos
- Fournit un échange de message encapsulant une authentification
- Support de plusieurs mécanismes d'authentification dont Kerberos
- Indépendance du format des messages et du protocole d'authentification
- Décrit dans les RFCs 2743, 1509 et 1964

Progression

Attaques

- Vol des secrets par compromission du serveur
  - → Secrets chiffrés sur disque avec le *Master Key Password*
- Vol du secret d'un administrateur
  - → Mot de passe spécifique Kerberos
  - → Minimiser le nombre d'administrateurs
  - → Limiter la portée des comptes (ACL)
- Compromission d'un fichier Keytab
  - → Restreindre les permissions sur ces fichiers
- Compromission d'un *Credential Cache*
  - → Restreindre les permissions à l'utilisateur uniquement
- Attaque par dictionnaire ou force brute
  - → Complexifier les secrets (*policy*)

Pré-authentification

- L'*Authentication Server* ne délivre pas de TGT à un client ne prouvant pas son identité
- Refus de la première demande (AS_REQ) avec un code d'erreur particulier
  $\rightarrow$ *KRB5KDC_EER_PREAUTH_REQUIRED*
- A la seconde demande, ajout d'un timestamp chiffré avec le secret partagé

Accroitre la sécurité des mots de passe

Password policy

- Longueur minimale du mot de passe
- Définir un nombre de classe de caractères que doit contenir un mot de passe
  - Lettres minuscules
  - Lettres majuscules
  - Nombres
  - Caractères de ponctuation
  - Autres caractères
- Durée de vie du secret (expiration du mot de passe)
- Historique des mots de passe (pour ne pas remettre le même)

Réplication des KDC

- Aucun standard définit dans les RFCs
- Mécanisme propre à chaque implémentation
- Les replicas sont en lecture seule
- L'implémentation MIT propose l'outil *kprop*
- Réplication totale

Progression

Conclusion

- Solution d'authentication sécurisée
- Ne traite pas la problématique de l'identification
  $\rightarrow$ Annuaire LDAP sans champs userPassword
- Mise en oeuvre d'une politique de mot de passe
- Single Sign On