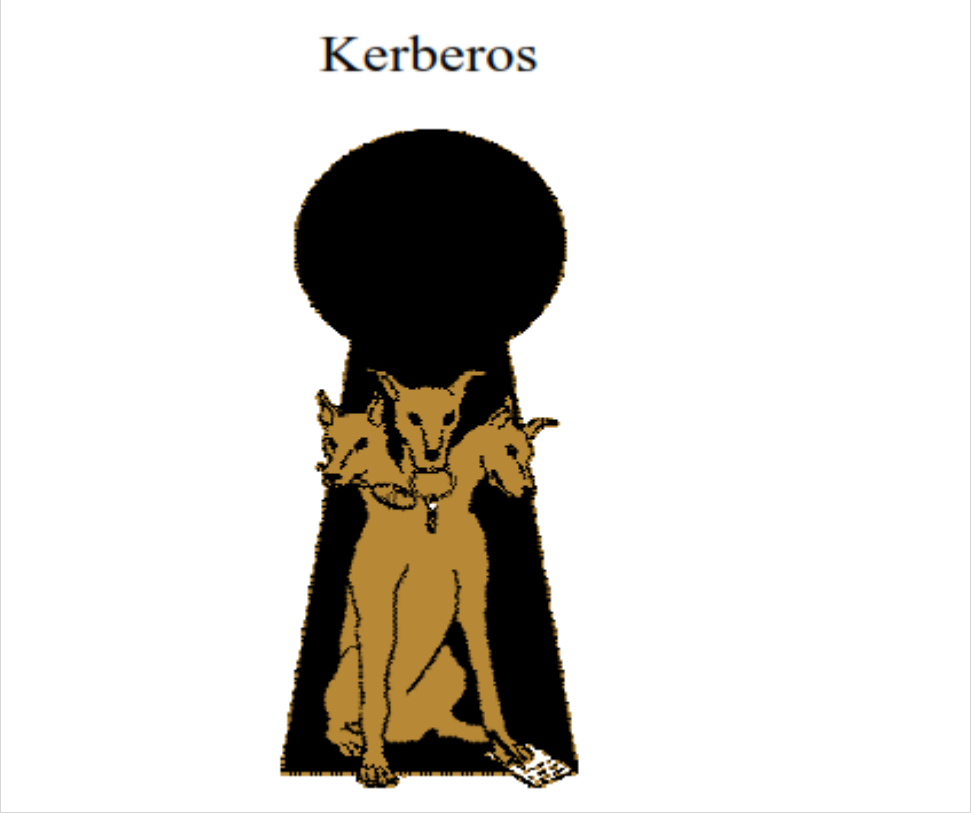
Sécurité informatique :
Le protocole d'authentification

Kerberos

Exposé réalisé par :
- Kemouguette Aya du groupe 7
- Lehamel titem du groupe 5

Encadré par Mr : Anane Mohamed
Année universitaire : 2013/2.14



1. Introduction

La sécurité et l'intégrité d'un système au sein d'un réseau peut être une lourde tâche. En effet, elle peut monopoliser le temps de plusieurs administrateurs rien que pour effectuer le suivi des services en cours d'exécution sur un réseau et surveiller la manière selon laquelle ils sont utilisés. De plus, l'authentification des utilisateurs auprès des services réseau peut s'avérer être une opération dangereuse lorsque la méthode utilisée par le protocole est par essence non-sécurisée, comme c'est le cas avec les protocoles FTP et Telnet lors du transfert de mots de passe de manière non-cryptée sur le réseau.

L'authentification :

C'est la vérification d'informations relatives à une personne ou à un processus informatique. L'authentification complète le processus d'identification dans le sens où l'authentification permet de prouver une identité déclarée. Dans un serveur, un processus de contrôle valide l'identité et après authentification, il donne accès aux données, applications, bases de données, fichiers ou sites Internet. Dans le cas contraire, l'accès est refusé.

L'authentification peut se faire de multiples manières, et notamment par la vérification de :

- « ce que je sais », un mot de passe par exemple.
- « ce que je sais faire », une signature manuscrite sur un écran tactile / digital.
- « ce que je suis », une caractéristique physique comme une empreinte digitale.

- « ce que je possède », une carte à puce par exemple.

Le choix de telle ou telle technique dépend en grande partie de l'usage que l'on souhaite en faire : authentification de l'expéditeur d'un email, authentification d'un utilisateur qui se connecte à distance, authentification d'un administrateur au système, authentification des parties lors d'une transaction de B2B (Business to Busines), etc.

La combinaison de plusieurs de ces méthodes (aussi appelées facteurs d'authentification) permet de renforcer le processus d'authentification, on parle alors d'authentification forte.

Par ailleurs, l'authentification peut se reposer sur un protocole d'authentification réseau, le protocole Kerberos, qui permet de sécuriser les mots de passe statiques (mots de passe qui restent identiques pour plusieurs connexions sur un même compte) lorsqu'ils sont transmets sur le réseau.

2. Kerberos

2.1. Définition

Kerberos est le nom grec de Cerber, le chien à trois têtes, gardien des enfers, mais aussi et surtout un nom donné au protocole d'authentification, basé sur des travaux décrits par Needham et Schreoder, conçu spécialement pour fournir une authentification forte au niveau des applications client/serveur à l'aide des clés symétriques. Il a était créé au MIT (Massachusetts Institute of Technology), dans le cadre du projet Athena, principalement par Miller et Neuman.

2.2. Présentation

Le principe de base de base de toute authentification sur un réseau est que chaque partie puisse être sûre de l'identité des autres parties. Pour garabtir cela, le protocole Kerberos met en œuvre, dans le processus d'authentification entre deux parties, deux tiers destinés à effectuer l'authentification entre les deux parties : le serveur d'authentification et le serveur d'attribution des tickets. De plus, les principes de cryptographie modernes (cryptage fort par clé secrète) qui sont utilisés afin de prouver l'identité des parties et d'occulter les messages aux yeux d'éventuels pirates qui les interceptent.

En fait, la conception du protocole Kerberos présume du fait qu'un certain nombre de conditions sont réunies pour garantir la sécurité du réseau :

- Les machines intervenant dans le protocole d'échange sont stables (insensibles à des attaques de type « denial of service ») et physiquement inviolables

- Les parties prenantes ne divulguent jamais leurs clés secrètes afin que personne ne puisse de faire passer pour elles
- Les mots de passe choisis sont suffisamment longs et complexes pour résister pendant une durée acceptable à une attaque
- Les horloges internes des machines sont relativement synchrones
- Les identifiants uniques de parties ne sont pas recyclés, ou seulement sur une longue période.

2.2. Architecture

L'architecture de Kerberos s'articule sur deux services d'authentification principaux : le service d'authentification et le service d'attribution de tickets.

2.2.1. Le service d'authentification

Cette partie du serveur Kerberos est destinée à prendre en charge les requêtes d'authentification sur le réseau. Tout client qui souhaite d'authentifier auprès d'autres parties doit tout d'abord s'adresser au service d'authentification pour pouvoir communiquer avec le service d'attribution de tickets.

2.2.2. le service d'attribution de tickets

Ce service va permettre de destribuer, aux clients authentifiés à l'aide du service d'authentification, les tickets qui vont leur permettre de communiquer avec les autres parties du réseau. Ce service va aussi rendre possible l'authentification inter-royaumes, que nous détaillerons un peu plus tard.

2.2.3. la base de données Kerberos

Toutes les clés des parties du réseau sont stockées dans une base de données cryptée qui contient, pour chaque partie, son nom (son identifiant), sa clé et diverses informations de validité de la clé. La base de données Kerberos n'a nullement le besoin d'être implémentée sur la même machine que le service d'authentification ou le service d'attribution de tickets.

2.2.4. le royaume

Pour organiser la structure des réseaux d'authentification de Kerberos, chaque ensemble de parties prenantes est inclus dans un royaume. D'autres systèmes d'authentification appelle ce regroupement un domaine. Il va donc falloir établir des relations de confiance entre les royaumes, le plus souvent organisé hiérarchiquement, afin que les clients puissent s'authentifier et utiliser des ressources situées dans un autre royaume que celui auquel ils appartiennent.

2.2.5. le serveur d'administration de Kerberos

C'est un service particulier, mis en place sur le serveur Kerberos, qui permet d'effectuer des opérations de maintenance sur la base de données Kerberos (ajout et suppression d'utilisateurs, changement de mots de passe, …)

Dans la spécification originale de Kerberos, en langue anglaise, le serveur d'authentification est dénommé Authentification Server (AS), le serveur d'attribution de tickets est appelé Ticket Granting Server (TGS), le royaume est le Realm, et le serveur d'administration de Kerberos est le Kerberos Administration Server (KADM).

En ce qui concerne les interactions entre les différents éléments de l'architecture Kerberos, chaqu'un de ces éléments a un rôle bien défini dans le fonctionnement de Kerberos.

L'architecture Kerberos est ainsi découpée afin de garantir une sécurité optimale aux utilisateurs du système.

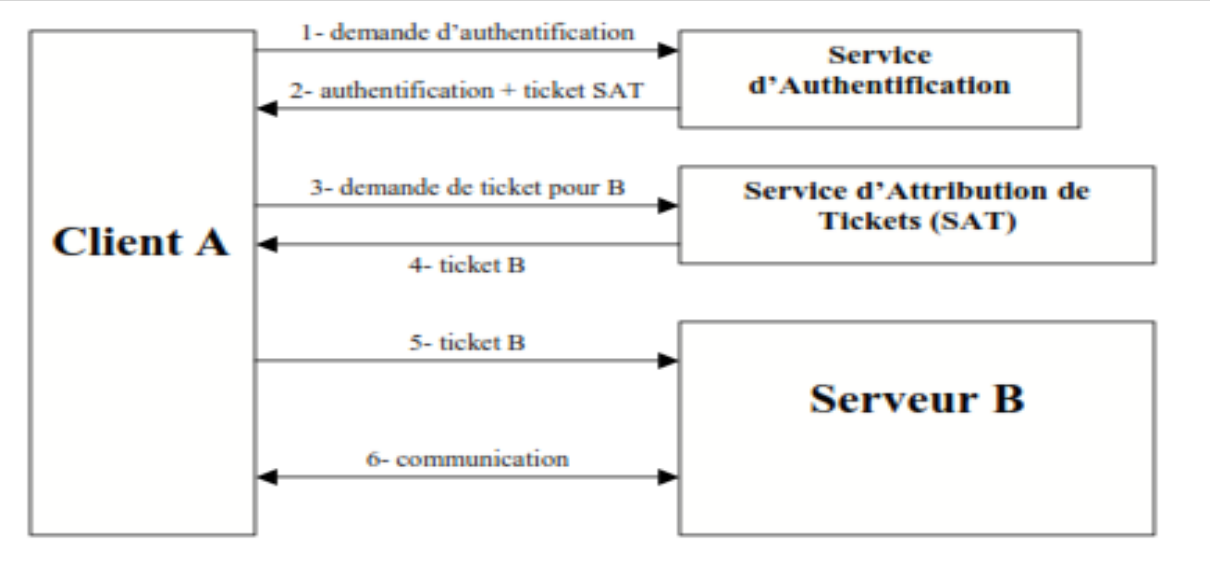
2.3. Fonctionnement

Note : dans les schémas que nous allons montrer, et plus généralement dans les textes explicatifs qui vont suivre, on utilisera sensiblement la même notation pour représenter une clé et pour exprimer l'action ou l'état de cryptage avec une clé. Ainsi, une clé sera notée C_I où C est l'initial du mot « clé » lui-même et I la ou les initiales du processeur de la clé ; le cryptage de données sera, quand à lui, noté C_I (DONNEES), où DONEES représentera une série de données (séparées par des virgules) qui seront cryptées par la clé de I.

Nous allons décrire succinctement dans ce paragraphe l'utilisation de base du système Kerberos.

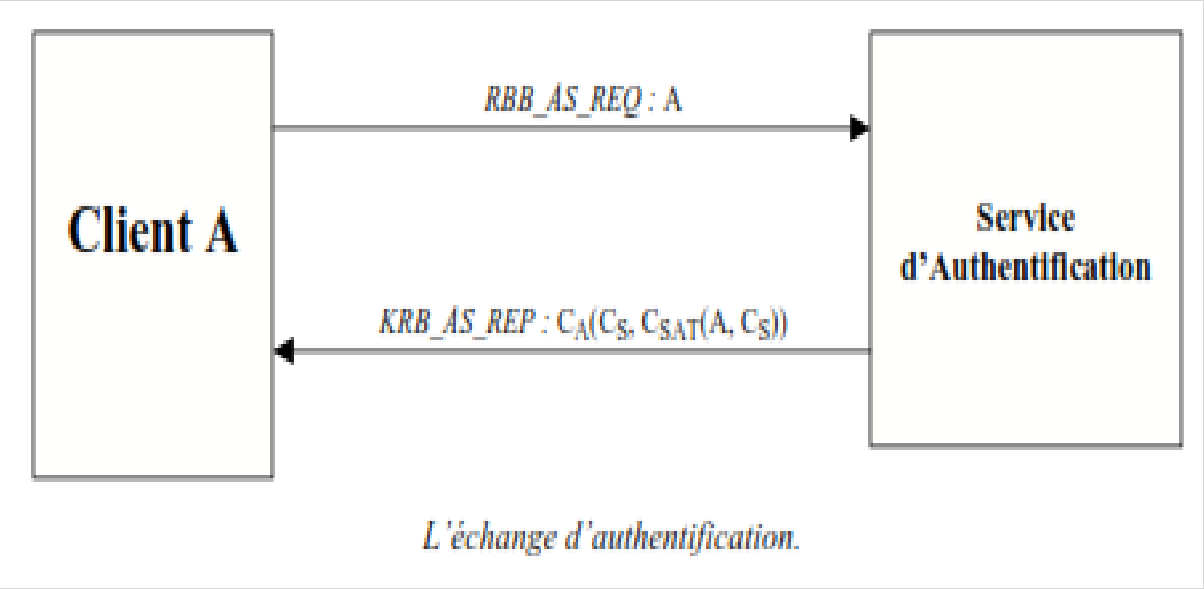
La séquence d'authentification « classique » d'un client A auprès d'un serveur B se déroule en trois phases :

- Le client A demande au service d'authentification de l'authentifier et de lui fournir un ticket pour s'authentifier auprès du service d'attribution de tickets
- Puis, A demande au service d'attribution de tickets de lui fournir un ticket pour s'authentifier auprès du serveur B
- Enfin, le client A fourni le ticket au serveur B afin de s'authentifier auprès de celui-ci, et les échanges transactionnels peuvent commencer.



2.3.1. L'authentification

La phase d'authentification est la phase qui conditionne le reste de l'utilisation du système. Elle n'est effectuée qu'une seule fois, alors que nous verrons que les autres phases seront déroulées à chaque nouvelle authentification sur un nouveau serveur.
Pour débouter l'ensemble du processus d'authentification sur le réseau, le client A envoie son identifiant au service d'authentification inclus dans un message KRB_AS_REQ.
Le service d'authentification recherche alors la clé du client A dans la base de données Kerberos en prenant celle qui a le plus grand numéro de version (les clés étant générées dans la base en tenant compte des changements divers grâce à une gestion de version). Il recherche aussi la clé du service d'attribution de tickets pour pouvoir continuer le protocole.
Si l'une des clés n'est pas trouvée, un message KRB_ERROR, contenant les informations d'erreur appropriées, est renvoyé au client.
Sinon, le service d'authentification génère une clé de session C_S aléatoire, c'est-à-dire qui soit impossible à deviner à l'avance par quiconque en possèderait une autre. Puis il retourne un message KRB_AS_REP contenant la clé de session et un bloc de données à destination du service d'attribution de tickets. Ce bloc de données, appelé « ticket pour le SAT » ou « ticket d'attribution de tickets », est crypté avec la clé C_{SAT} du service d'attribution de tickets de façon à ce que lui seul puisse le lire, et contient l'identifiant du client A ainsi que la clé de session C_S générée précédemment. L'ensemble du corps du message KRB_AS_REP est lui-même crypté à l'aide de la clé C_A du client A.



L'échange d'authentification.

2.3.2. L'attribution d'un ticket

Une fois que le client A s'est authentifié sur le réseau à l'aide du service d'authentification, il peut demander un ticket d'authentification, pour un serveur B, au service d'attribution de tickets.

Un ticket est un message crypté qui assure, par son cryptage, à une entité sur le réseau que l'entité qui tente de démarrer une conversation est bien celle qu'elle prétend être. Ce ticket garantit que les échanges d'authentification préalables ont été effectués avec succès car l'entité qui l'utilise n'aurait pas pu se le procurer autrement.

Ainsi, pour obtenir un ticket pour un serveur B, le client A envoie le message KRB_TGS_REQ au service d'attribution de tickets. Ce message contient le ticket pour le service d'attribution de tickets, de façon à amener la preuve à celui-ci que la phase d'authentification par le service d'authentification s'est effectivement déroulée. Le message contient aussi l'identifiant (le nom) du serveur B sur lequel le client souhaite s'authentifier, ainsi qu'une indication temporelle(t dans le schéma ci-dessous), cryptée à l'aide de la clé de session C_s, destinée à ce que le service d'attribution de tickets s'assure que le message ne soit pas une « redite » d'un message précédent. Cette indication est timestamp, c'est-à-dire le nombre de millisecondes écoulées depuis le 1er Janvier 1970 à minuit.

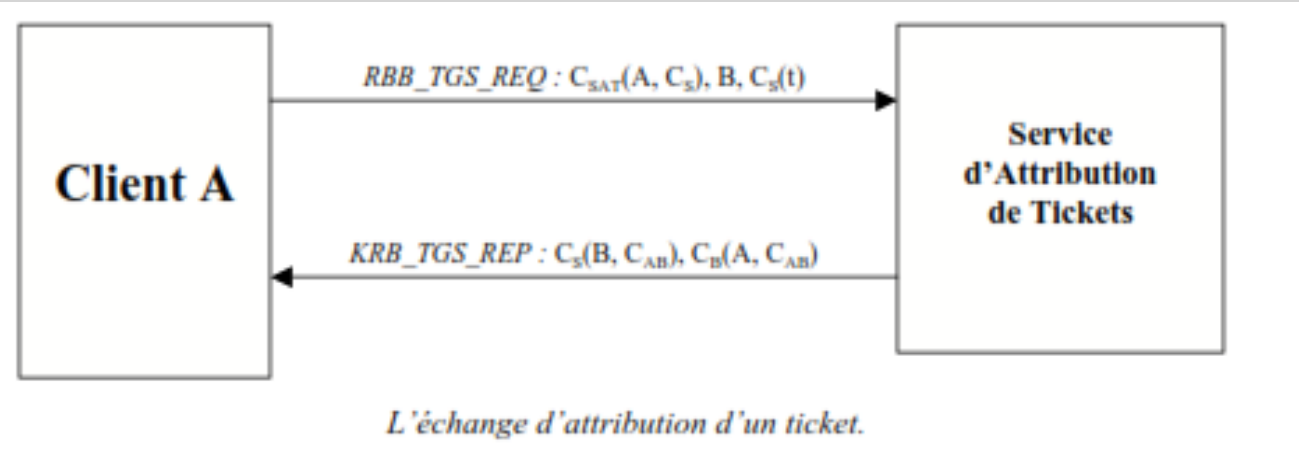
Quand le service d'attribution de tickets reçoit le message KRB_TGS_REQ, il décrypte le ticket qui lui est destiné et qui contient l'identifiant du client A, ainsi que la clé de session C_S, avec laquelle il décrypte le timestamp pour vérifier que le message a été émis dans un intervalle de temps raisonnable après la récupération du ticket auprès du service

d'authentification. Cela impose que les systèmes sur le réseau se doivent d'être plus ou moins synchrones pour que le système Kerberos fonctionne effectivement.

Une fois que le message KRB_TGS_REQ a été validé par le service d'attribution de tickets, celui-ci recherche la clé C_B du système auquel souhaite accéder dans la base de données Kerberos. Puis le service d'attribution de tickets génère une clé partagée C_{AB} entre le client A et le serveur B, et prépare un message KRB_TGS_REP.

Le message KRB_TGS_REP contient deux parties principales : la communication de la clé partagée entre A et B au client A et le ticket à fournir au serveur B. La clé partagée C_{AB} est cryptée, en même temps que l'identifiant de B, à l'aide de la clé de session C_S générée pendant la phase d'authentification et dont seuls le serveur Kerberos et le client A ont connaissance. Le ticket à fournir au serveur B est quand à lui principalement constitué de l'identité du client A et de la clé partagée C_{AB}, le tout crypté par la clé C_B de serveur B de manière à ce que seul lui puisse le décrypter.

Le but de ce message est de préparer à ce que seul le client A et le serveur B aient connaissance (et soient en mesure de connaitre) une clé partagée par eux seuls.



L'échange d'attribution d'un ticket.

2.3.3. Les différents types de tickets

Le système Kerberos permet un certain nombre de possibilités de configuration de la sécurité, par l'intermédiaire du changement du type de ticket qu'il va fournir pour l'authentification. Le type du ticket est stocké dans les messages Kerberos sous la forme d'un mot de 32 bits et une checksum cryptée est calculée pour être sûr que le message n'a pas été modifié.

Les types de tickets peuvent être demandés par le client pour obtenir tel ou tel service de la part d'une autre entité du réseau qui peut requérir ce type de ticket. Mais ils sont aussi positionnés par le système Kerberos pour indiquer la nature du ticket et la manière dont il doit être utilisé.

Les différents types de tickets sont les suivants (entre parenthèses le nom de la donnée à positionner dans la spécification originale de Kerberos) :

- Ticket initial (INITIAL)

Il s'agit d'un ticket qui a été émis par le service d'authentification lui-même. C'est-à-dire qu'il a été généré lors du protocole d'authentification entre le client et le service d'authentification. Certaines applications désirant s'assurer que l'authentification vient d'être effectuée, comme par exemple une application de changement de mot de passe, peuvent requérir ce type de ticket.

- Ticket renouvelable (RENEWABLE)

Dans le système Kerberos, les tickets ont une durée de validité limitée de manière à limiter par là même les risques liés au stockage du ticket sur le système client (le vol principalement). D'un autre coté, une durée de validité trop courte imposerait à l'utilisateur d'avoir à ressaisir trop fréquemment sa clé.

Pour pallier aux deux problèmes, il est possible d'autoriser le renouvellement de tickets en positionnant la donnée RENEWABLE. Ainsi, le ticket se retrouve crédité de deux durées de validité : une durée renouvelable et une durée au-delà de laquelle le ticket ne peut plus être renouvelé. Ainsi à chaque expiration de la première durée, le ticket est retourné au serveur Kerberos afin d'être renouvelé, et ce jusqu'à expiration de la deuxième durée de validité.

Dans le cas où un vol de ticket aurait été signalé, le serveur Kerberos refuserait de renouveler le ticket. Ceci permettrait ainsi de limiter la durée d'utilisation d'un ticket dérobé.

- Ticket postdaté (MAY-POSTDATE et POSTDATED)

Dans le cas d'applications asynchrones, où la clé d'authentification ne peut pas être présente au moment du besoin d'authentification, on peut avoir besoin d'effectuer une authentification à l'avance.

En positionnant l'attribut MAY-POSTDATE en effectuant sa demande de ticket auprès du service d'attribution de tickets, le client déclare qu'il prévoit utiliser le ticket dans un futur assez éloigné. Lorsque le client désir utiliser le ticket, le serveur Kerberos l'active (s'il n'a pas fait l'objet d'un vol) et positionne l'attribut POSTDATED sur le ticket de manière à ce que certain applications puissent refuser le ticket si elles le souhaitent.

- Ticket proxy (PROXIABLE et PROXY)

Le ticket proxy est destiné à être utilisé pour permettre à une application d'agir sous couvet de l'identité (et donc de l'authentification) de l'entité qui a émis le ticket proxy, c'est-à-dire qui donne procuration à l'application, pour une activité donnée.

Pour émettre un ticket proxy, le client doit positionner l'attribut PROXIABLE sur le ticket et le serveur Kerberos, lui, positionnera l'attribut PROXY sur le ticket. Ensuite, le client peut fournir ce ticket à un tiers pour qu'il dispose de son authentification pour une ressource particulière. Ainsi, la tierce entité disposant de ce ticket pourra s'authentifier auprès de la ressource et endosser ainsi l'identité du client, et donc ses droits sur la ressource.

Pour rendre inopérant le vol de tels tickets, ou du moins pour en rendre plus difficile l'utilisation, on peut aussi joindre, au sein du ticket pour la ressource, la description de l'entité supposée utiliser le ticket. Ainsi, la ressource traitant un ticket proxy qui déclare une adresse différente de l'entité qui l'utilise, pourra refuser l'authentification par ce ticket.

Le ticket proxy ne permet d'effectuer des demandes de tickets d'attribution de tickets, à la différence du ticket transférable.

- Ticket transférable (FORWARDABLE et FORWARDED)

Le ticket transférable a une signification très proche de celle du ticket proxy, si ce n'est qu'il permet d'endosser complètement l'identité de l'entité qui a émis le ticket. De plus, ce ticket permet même d'effectuer des demandes de tickets d'attribution de tickets auprès du serveur Kerberos.

- Ticket invalide (INVALID)

Lorsque l'attribut INVALID d'un ticket est positionné, c'est que le ticket est déclaré invalide et ne doit pas être accepté par aucun tiers sur le réseau. Dans le cas d'un ticket postdaté, l'attribut INVALID est positionné tant que le ticket n'a pas été arrivé et que le processeur du ticket ne l'a pas soumis à la validation du serveur Kerberos.

2.4. Utilisation du ticket pour l'accès à une ressource

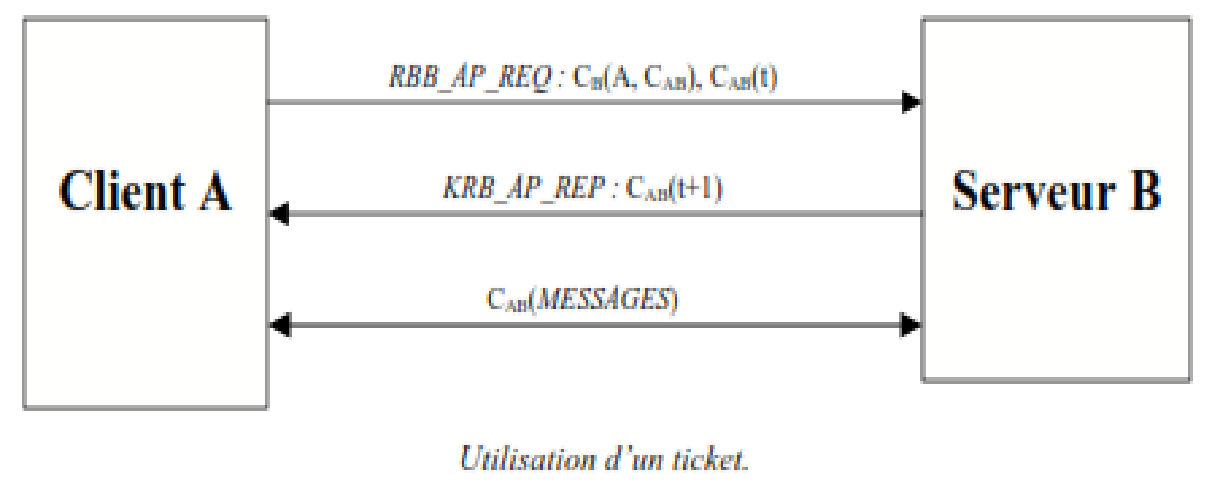
A partir du moment où le client a obtenu un ticket pour l'accès à une ressource, il peut envoyer un message KRB_AP_REQ au serveur auquel il désire accéder.

Le message KRB_AP_REQ contient le ticket pour le serveur, c'est-à-dire l'identifiant du client et la clé et la clé partagée entre le client et le serveur (dans notre exemple, il s'agit de C_{AB}) , le tout crypté à l'aide de la clé du serveur de manière à ce qu'il soit le seul à pouvoir décrypter le ticket.

De plus, le message KRB_AP_REQ contient aussi un timestamp, crypté à l'aide de la clé partagée entre le client et le serveur. Ainsi le serveur peut vérifier que le message qui lui est soumis pour authentification, ne soit pas une redite d'un message précédent.

Une fois le message vérifié par le serveur, celui-ci peut renvoyer au client un message d'accréditation de l'authentification qui consiste en un timestamp plus récent crypté à l'aide de la clé partagée.

Désormais, puisque l'ensemble du processus d'authentification a été déroulé, les échanges transactionnels, entre le client A et le serveur B, peuvent commencer, sous couvert du cryptage à l'aide de la clé partagée, et ce pendant toute la durée des échanges.



Utilisation d'un ticket.

Dans l'échange d'authentification, nous n'avons considéré que l'authentification elle-même, c'est-à-dire l'apport de la preuve de l'identité des parties amené avant les échanges transactionnels liés aux besoins applicatifs qui ont nécessité une authentification.

Or, dans le protocole Kerberos, les tickets sont construits de manière à prendre en charge bine plus que l'authentification : l'autorisation. En effet, il est possible de renseigner des « capacités »

Une capacité est une donnée qui permet à son porteur d'accéder à certaines permissions sur le système auquel il fournit le ticket. Les capacités ne sont pas affectées à une adresse réseau particulière comme le sont les autres tickets pour limiter l'usage des tickets dérobés.

2.5. Authentification sur un autre royaume

Un des principaux aspects de l'architecture de Kerberos, c'est qu'elle est organisée autour d'un royaume qui va contenir un ou plusieurs serveurs Kerberos. Cette structure permet de décomposer les responsabilités de sécurité lors de la mise en œuvre d'une architecture d'authentification au sein d'une organisation importante. D'où la nécessité, puisque le réseau de l'organisation se trouve explosé en plusieurs parties, d'un protocole d'authentification sur les autres royaumes.

Kerberos permet l'authentification sur un autre royaume en créant des relations de confiance entre les services d'attribution de tickets des différents royaumes. Ainsi, si un client veut accéder à un service qui se trouve dans un autre royaume, il va demander au

service d'attribution de tickets de son propre royaume de lui fournir un ticket pour le service d'attribution de tickets de l'autre royaume.

En fait, deux situations peuvent se présenter : soit le client connait le royaume auquel appartient le serveur auprès duquel il souhaite s'authentifier, soit il ne le connait pas. Autant que possible, le client essaie de fournir cette information au service d'attribution de tickets de son royaume. Le service d'attribution de tickets, quand à lui, donne donc deux types de réponses à la requête du client : s'il dispose, dans sa base de données d'authentification, de la clé du service d'attribution de tickets de l'autre royaume , c'est cette clé qu'il va utiliser pour fournir un ticket à son client : par contre, si le royaume sur lequel désire s'authentifier le client n'existe pas dans sa base de données, ou si le client n'a pas pu fournir son nom, le service d'attribution de tickets va utiliser la clé de service d'attribution de tickets du royaume le plus proche pour crypté le ticket d'authentification. Et tant que le royaume n'a pas été trouvé, le processus continue récursivement.

2.6. Structure du ticket Kerberos

Afin de mieux appréhender les données et les processus qui ont été décrits précédemment nous présentons ici la structure d'un ticket Kerberos lorsqu'il est mis par le service d'attribution de tickets et conservé ou utilisé par le client.

Nom du champ	Description
Les trois premiers champs du ticket ne sont pas cryptés afin que le client puisse gérer sa propre base de tickets.	
tkt-version	Version du ticket Kerberos (actuellement « 5 »)
realm	Nom du royaume qui a émis le ticket
sname	Nom du serveur pur lequel le ticket est destiné
Les champs suivants sont cryptés et contiennent des informations à destination du serveur.	
flags	Attributs du ticket (mot de 32 bits)
key	Clé de session
crealm	Nom du Royaume du client
cname	Nom du client
transited	Liste des Royaumes par lesquels le client a dû passer pour s'authentifier
authtime	Date de l'authentification initiale du client
startime	Date de début de validité du ticket
endtime	Date de fin de validité du ticket
renew-till	Date jusqu'à laquelle le ticket peut être renouvelé (dans le cas d'un ticket avec l'attribut RENEWABLE positionné)
caddr	Liste des adresses réseau à partir desquelles le ticket peut être utilisé. Si ce champ est omis, le ticket peut être utilisé à partir d'une adresse réseau quelconque.
authorization-data	Ce champ n'est pas utilisé à proprement parler par le protocole Kerberos, il s'agit de données à destination du service auquel s'adresse le client, par exemple dans le cas de l'utilisation d'une capacité.

Structure d'un ticket Kerberos.

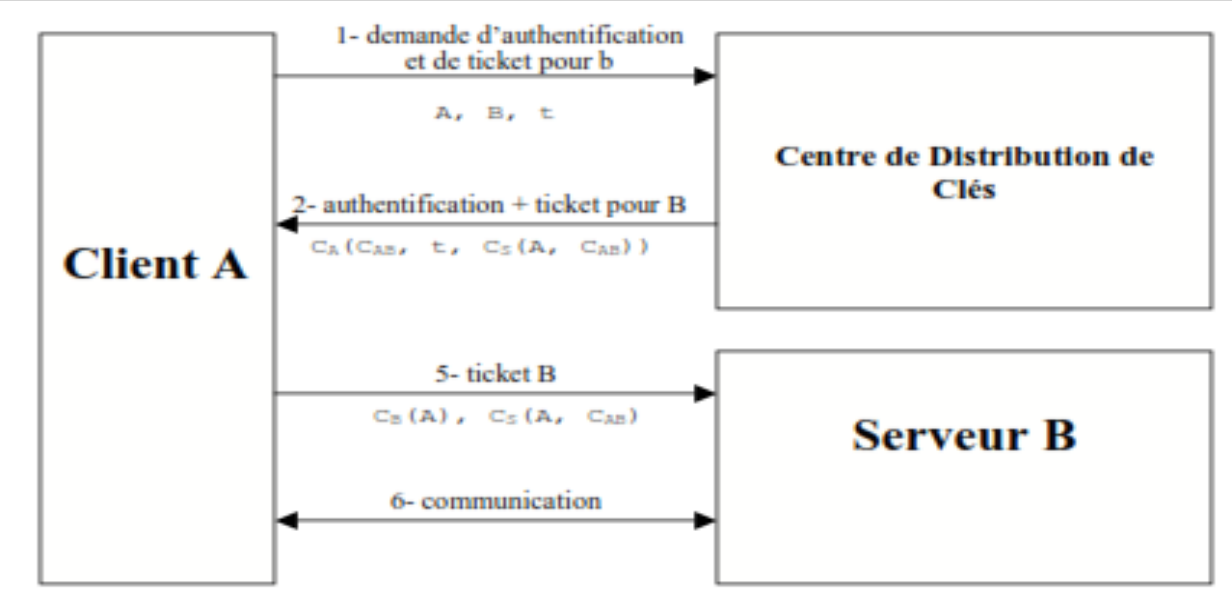
3. Nouveautés de Kerberos v5

3.1. Faiblesse de Kerberos v4

Très tôt après l'apparition de Kerberos v4, son évolution fut réfléchie de manière à compenser les faiblesses dont il était sujet. Nous couvrons ici les principales :

3.1.1. Faiblesses structurelles

Comme nous le voyons sur le schéma suivant qui détaille le mode de fonctionnement de Kerberos v4, le nombre de parties impliquées dans cette version est inferieur à celui utilisé dans la version 5. Et ce pour la simple raison que le centre de distribution de clés occupe une place centrale et unique dans le système Kerberos v4. Il est en charge de l'authentification du client et de la fourniture de tickets pour l'accès aux serveurs.



Echange d'authentification avec Kerberos v4.

D'autre part, comme nous pouvons le voir sur ce schéma d'interaction, le serveur B et le centre de distribution de clés doivent avoir en commun une clé partagée C_S pour que le serveur B puisse recevoir le ticket émit par le centre de distribution des clés à destination du client A. Cela induit des risques supplémentaires dus au stockage permanent de telles clés.

Enfin, à chaque demande de ticket de la part du client A, un message crypté de la clé de ce client transite sur le réseau, a donc plus de risques de se faire intercepter par une personne qui aurait pu dérobé la clé. D'autre part, pour finir, à chaque nouvelle demande de ticket, le client A doit présenter sa clé (soit sous la forme d'une invite de mot de passe, soit par stockage semi-permanent), ce qui augmente les risques de vols de clés.

3.1.2. Faiblesses cryptographiques

La version 4 de Kerberos utilisant la méthode de cryptage triple DES, un attaquant peut tenter une approche probabiliste pour générer un ticket Kerberos valide (et notamment un ticket d'attribution de tickets) et ainsi agir sous couvert de l'identité d'un autre utilisateur du système.

De plus, comme nous l'avons vu dans le schéma précédent, le ticket d'authentification à destination du serveur est encrypté deux fois : une fois à l'aide de la clé de session partagée entre le serveur et le centre de distribution de clés et une seconde fois à l'aide de la clé du client ; alors que cela ne s'avère pas nécessaire. Ce double cryptage induit un surcroit de besoin en puissance de traitement pour le centre de distribution de clés.

Le protocole Kerberos v4 comporte d'autres faiblesses que nous ne détaillerons pas. En voici quelques une : dépendance vis-à-vis du protocole IP, dépendance vis-à-vis de la méthode de cryptage, difficulté de l'authentification être les royaumes …

3.2. Améliorations apportées par Kerberos v5

Kerberos a été amélioré considérablement avec l'évènement de la version 5 voici présentés ici les changements principaux dans le protocole.

3.2.1. Interchangeabilité des algorithmes de cryptage

Pour faciliter l'exportation du système Kerberos en dehors des Etats-Unis et assurer une extensibilité du protocole, il est désormais possible de spécifier la méthode de cryptage que l'on souhaite utiliser. A cette fin, le système Kerberos version 5 propose d'ailleurs un certain nombre de systèmes de cryptages par défaut.

Il est ainsi possible d'utiliser l'algorithme de cryptage DES en mode CBC (Cipher-Block-Chaining), soit avec un contrôle d'intégrité MD4, soit avec un contrôle d'intégrité MD5. Il est même possible d'utiliser le protocole Kerberos sans cryptage de façon à ne conserver que les capacités d'identification, par exemple sur un réseau sécurisé.

3.2.2. Utilisation de l'encodage ASN.1

Désormais, l'ensemble de la spécification Kerberos utilise ASN.1 (Abstract Syntax Notation number One) pour décrire les données ainsi que les échanges réseaux. Cela le rend plus aisé à implémenter et rend ses implémentations plus simples à valider.

3.2.3. Extension du support de l'adressage réseau

Le protocole Kerberos en version 5 autorise l'utilisation du type de réseau que l'on souhaite par la présence dans les tickets d'un champ que l'on peut utiliser pour déclarer le type de réseau que l'on utilise et ainsi permettre l'interprétation du champ d'adressage réseau aux utilisateurs du ticket.

3.2.4. Introduction de nouveaux types de tickets

Dans la nouvelle version de Kerberos, un champ « flags » a été introduit dans la structure du ticket de manière à pouvoir spécifier un type et un état du ticket. On a vu précédemment quels types de tickets pouvaient exister, ainsi que leur utilisation possible.

3.2.5. Authentification inter-royaumes

L'authentification sur une machine d'un royaume différent de celui dans lequel on se situe a été considérablement simplifiée en mettant en place une structure hiérarchique des royaumes. Ainsi pour s'authentifier sur une machine d'un autre royaume, il faut demander un ticket au service d'attribution de tickets supérieur pour accéder au service d'attribution de tickets situé dans une autre branche de la hiérarchie, réduisant ainsi le nombre d'échanges effectués pour obtenir une authentification sur une machine. Cela réduit aussi la complexité de gestion des relations inter-royaumes.

3.2.6. Introduction de la GSS-API

La nouvelle version de Kerberos fournit en standard une interface de programmation d'application (API) dérivée de la GSS-API (General Security Services) afin de permettre à des développeurs d'application d'étendre et de modifier le schéma d'authentification de Kerberos.

4. Implémentation

Pour être largement utilisé, un système, quel qu'il soit, se doit de disposer d'un nombre étendu d'implémentations sur de nombreux systèmes et s'intégrer dans les environnements de programmation des développeurs d'applications.

C'est le cas de Kerberos, dont on peut trouver la trace dans nombreux systèmes. De plus, le MIT fournit une implémentation de référence, au code source ouvert, accessible à quiconque désirant pousser plus loin l'étude ou l'implémentation de Kerberos.

4.1. Au cœur des systèmes d'exploitation

Grâce à sa conception intelligente et évolutive, le système Kerberos a été largement adopté pour implémenter les besoins d'authentification sur les réseaux.

Ainsi, dans le monde UNIX, qu'il s'agisse de Sun Solaris, de MacOS X, ou de Linux, l'implémentation utilisée est, généralement, celle du MIT.

De même, dans les environnements Microsoft, Kerberos a fait son apparition, depuis Windows 2000, de manière à remplacer le système d'authentification qui existait alors. L'organisation des domaines de Windows NT 4 a été complètement revue pour prendre en charge l'authentification par Kerberos dans Windows 2000.

4.2. Enfui dans les outils

Kerberos est présent dans de nombreux outils en tant que protocole d'authentification sur des ressources.

Par exemple, Eudora implémente un client Kerberos pour authentifier ses utilisateurs sur des serveurs de messageries (POP, SMTP, NNTP …) qui utiliseraient Kerberos comme moyen d'authentification.

De même de serveurs d'impression peuvent utiliser Kerberos avec les tickets proxys pour accéder à une ressource possédée par l'utilisateur qui fait la demande d'impression.

Certains pare-feux permettent aussi l'utilisation de Kerberos pour authentifier et contrôler l'accès au réseau Internet.

La liste pourrait être longue, car l'authentification par Kerberos se retrouve au sein de beaucoup de systèmes et outils sur tous types de plateformes.

5. Conclusion : L'authentification dans le futur

Pour clore ce rapport, nous allons survoler rapidement les principes qui feront l'authentification du futur :

5.1. Amélioration de Kerberos

Le système Kerberos est un système ouvert, c'est-à-dire que tout le monde peut s'atteler à son amélioration et c'est ce qui fait principalement que Kerberos soit en perpétuelle évolution.

De plus, les concepteurs de Kerberos, au MIT, sont à l'écoute des besoins des utilisateurs, ainsi que des failles et faiblesses dont le système pourrait faire preuve.

Les principales évolutions attendues sont les suivantes :

- Cryptage à clé publique : à l'heure actuelle, Kerberos est basé sur des principes de cryptage à clé privée, mais au vu de l'importance que prennent les infrastructures à clé publiques, il sera sans doute nécessaire de l'y adapter.
- Cartes à puce : l'intérêt des cartes à puce, c'est qu'elles permettent de ne jamais mettre au jour la clé de l'utilisateur, surtout dans le cas où celui-ci utilise une station de travail en laquelle il ne peut avoir confiance.
- Administration à distance : pour le moment, la spécification de Kerberos ne prend pas en charge l'administration de la base de données de Kerberos sur une autre machine que le serveur Kerberos, même si des applications le permettant existent.
- Réplication de base : il est nécessaire de concevoir un mécanisme de réplication sécurisée de la base de données de Kerberos entre les serveurs Kerberos du domaine.

5.2. Renforcement des cryptages

Les technologies évoluent rapidement, les puissances de calcul augmentent de façon linéaire et un cryptage que l'in supposait inviolable peut se retrouver percé du jour au lendemain, soit par la force brute de plusieurs machines travaillant de concert, soit par la découverte d'une quelconque faille dans l'algorithme de cryptage.

La recherche en cryptographie continue donc et s'attache soit à trouver les failles des méthodes de cryptage existantes, soit à trouver de nouvelles méthodes de cryptage.

Bibliographie

- Site principal de Kerberos : http://web.mit.edu/Kerberos/www
- The Evolution ok Kerberos Authentification Service –Kohl-Neuman-Ts'o-1991
- RFC 1510-Kohl-Neuman-1993
- Computer Networks (3rd edition) –Andrew S.Tanenbaum-Prentice Hall-1996