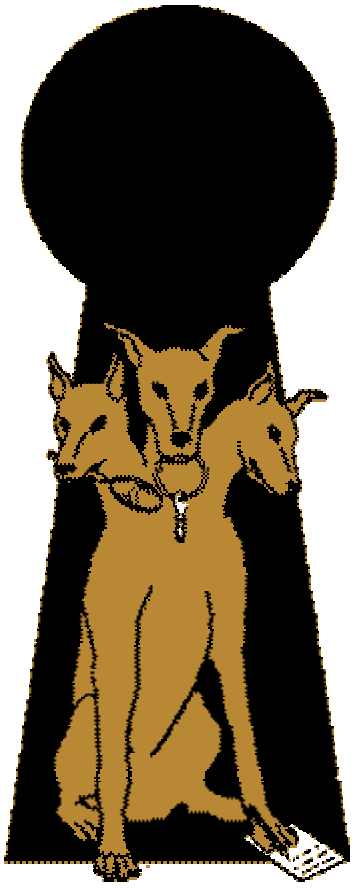
Kerberos



Besoins, architecture, fonctionnement, nouveautés de la version 5, implémentations, perspectives...

Guillaume BERTRAND
Mémoire de probatoire - CNAM 2003

contact : guillaume@agiletools.com

Introduction

L'expansion des réseaux de systèmes informatiques et l'avènement d'Internet pour l'interconnexion de ces réseaux ont amené de nombreux nouveaux problèmes à résoudre lors de la conception de systèmes informatiques.

Qu'il s'agisse de problèmes d'interopérabilité liés à la diversité des systèmes et des protocoles d'échanges réseaux, ou de problèmes liés à la sécurité des systèmes, des données, ou même des personnes, ils peuvent tous être résolus par l'adoption de protocoles d'échanges standardisés et fiables.

La sécurité des systèmes repose notamment en grande partie sur des standards prouvés, si possible de façon mathématiques, et permettant une interopérabilité basique.

Nous allons étudier dans ce document le protocole d'authentification Kerberos, ses principes d'architecture et de fonctionnement, les multiples fonctionnalités qu'il offre, ainsi que l'interopérabilité de systèmes qu'il favorise par ses multiples implémentations.

Mais auparavant, il nous est nécessaire de comprendre les concepts liés à la sécurité dans les réseaux informatiques, de manière à mieux appréhender la genèse de Kerberos et ainsi étudier au mieux ses propriétés.

La sécurité dans les réseaux de systèmes informatiques

Généralités

Les problèmes liés à la sécurité des réseaux informatiques sont nombreux et nécessitent une attention toute particulière des responsables de ces réseaux.

Pour les entreprises, la confidentialité des données, l'identification des parties prenantes lors de transactions, le contrôle de la validité des échanges et la garantie de l'unicité de chaque transaction, sont nécessaires pour assurer la bonne marche de l'activité économique.

Nous ne nous étendrons pas sur les répercussions que peuvent avoir une mauvaise gestion de la sécurité informatique en entreprise.

Les autres risques liés à la sécurité des réseaux informatiques, tels que « social engineering », attaques systèmes (hacking, virus…), ou attaques physiques (destruction de matériel…) ne font pas l'objet des parties qui vont suivre.

Acteurs et risques

Les acteurs, parties prenantes des problèmes de sécurité dans les réseaux informatiques peuvent être classés en deux grandes catégories : ceux qui tentent de maintenir la sécurité du réseau, et ceux qui tentent de la contourner pour arriver à leurs fins. Nous ne considérerons pas ici la catégorie de ceux qui mettent en péril la sécurité du réseau par ignorance, maladresse, ou inconscience.

La première famille recense principalement les ingénieurs systèmes consciencieux et les responsables de la sécurité. Ce sont eux qui vont devoir trouver des solutions aux potentiels risques amenés par la deuxième famille. Ils ont à leur avantage :

- l'évolution des technologies, favorisé par la concurrence et le développement de la recherche
- la découverte quotidienne de nouvelles attaques et de nouvelles failles dans leurs systèmes et le partage global, grâce notamment aux réseaux étendus, des moyens de contrecarrer ou d'inhiber ces éventuelles failles.

La deuxième famille d'acteurs englobe toutes les personnes qui peuvent avoir un intérêt à mettre en faute des systèmes, s'approprier ou falsifier des données ou encore consulter des informations auxquelles elles ne devraient pas avoir accès. Cela peut comprendre :

- les stagiaires, qui envisagent de tester leurs connaissances sur l'exploitation des failles de sécurité des réseaux informatiques
- les employés indiscrets, qui souhaitent consulter les données de leurs collègues
- les employés malhonnêtes, pour s'approprier le travail d'autres ou pour s'accorder une rémunération plus importante
- les clients malveillants, qui envisageraient de nier d'éventuelles commandes
- ou d'autres encore, pouvant tirer un quelconque bénéfice.

Il n'est pas rare qu'un acteur de la deuxième famille devienne par force d'expérience un acteur de la première. Car, d'une manière générale, tous les acteurs de la sécurité des réseaux se découvrent un point commun : la nécessité de comprendre les enjeux et les risques ou les

buts liés à la sécurité des réseaux, les types d'attaques qui peuvent être perpétrées et les solutions qui peuvent y être appliquées.

Solutions

La majorité des solutions de sécurité actuelles se basent sur le principe que les réseaux sont ouverts, c'est-à-dire qu'ils permettent l'ajout aisé de nouveaux systèmes, et qu'ils reposent sur des standards de communications, physiques et informatiques, connus de tous les acteurs de la sécurité et dont les forces et les faiblesses sont notoires.

Afin de garantir la sécurité des réseaux informatiques et des données qui y sont échangées, quatre axes principaux d'action doivent être envisagés :

- le secret : il est souvent nécessaire de s'assurer que les données échangées ne seront consultables que par leur destinataire. Pour cela, il faut que les instances en présence lors de la communication se soient accordées sur un protocole d'échange et cryptage de données de façon à ce que le contenu des messages échangés ne puissent être intelligible pour aucun autre intervenant sur le réseau.
- la non révocation : lorsqu'une transaction a été finalisée, aucune des parties prenantes ne doit pouvoir nier ou contester le contenu de ladite transaction. Pour cela tout au long de la transaction, des mesures diverses doivent être prises et acceptées de toutes les parties pour garantir la validité de la transaction. Ces mesures reposent notamment sur le contrôle de l'intégrité des données échangées et l'authentification et la signature des parties prenantes.
- le contrôle d'intégrité : dans un échange de données, lorsqu'un des tiers liés à l'échange reçoit un message, il lui faut s'assurer que les données ont bien été toutes transmises. Cela dans deux buts principaux ; premièrement de manière à être sûr que toutes les données ont bien été transmises (sans perte et sans dégradation du message) et deuxièmement pour garantir qu'un autre tiers n'ait pas modifié le message durant son transit sur le réseau.
- l'authentification : avant de laisser une entité (système ou personne) accéder à des informations (documents, programmes…) ou à des processus (commande, manipulation de bases de données,…) sensibles, il est nécessaire que l'identité de cette entité soit vérifiée et accréditée au préalable. C'est cet axe que nous développerons plus particulièrement lors de l'étude de Kerberos.

Ces quatre axes de recherche de solutions sont bien sûr interdépendants et reposent tous, en plus ou moins grande partie, sur l'utilisation de l'un ou de plusieurs des autres axes.

Le cryptage de données

Une manière de dissimuler des informations à des observateurs non autorisés, est de les rendre inintelligible pour quiconque ne connaît pas la manière de les lire. Léonard de Vinci notamment, en écrivant ses textes à l'aide d'un miroir, a ainsi pu écarter les curieux de ses écrits pendant très longtemps.

On appelle cryptographie, ou cryptage, ce moyen de protéger des données. Il existe de multiples méthodes pour crypter des données. Ces méthodes ont évolué au fil du temps et de l'augmentation de la gravité des enjeux (espionnage, guerres…) pour devenir de plus en plus sophistiquées, et de moins en moins faciles à dépasser.

Ces méthodes sont de 2 types : par application de calculs simples et par utilisation de calculs plus complexes pour un cryptage par clé.

Cryptage simple

Le cryptage simple consiste à appliquer une formule de conversion simple pour passer d'un texte à une séquence inintelligible de caractères. Lorsque l'on veut relire les messages, il suffit de connaître la formule de conversion employée et d'en appliquer l'inverse.

Les principaux cryptages simples sont :

- la substitution de caractères

  On définit une règle de conversion d'un caractère vers un autre (décalage de n caractères, table de conversion…) et on convertit l'ensemble du message vers un message codé. Cette méthode, assez simple à employer, est cependant assez simple à prendre en défaut : par analyse statistique du texte (probabilité d'apparition des lettres dans une langue, étude de la fréquence des doubles lettres,…), on arrive rapidement à casser le codage

- la transposition

  Au lieu de changer les caractères pour des autres, on change la place des caractères dans le texte en effectuant une lecture verticale du texte et en permutant les colonnes selon un algorithme spécial. Malheureusement, ce cryptage peut aussi être surmonté en essayant plusieurs combinaisons de permutations et en fondant des hypothèses sur le contenu du message

- la conversion par « ou exclusif »

  Cette méthode consiste à appliquer l'opérateur binaire « ou exclusif » (opérateur binaire XOR), bit à bit, entre la représentation binaire du message et la représentation binaire de la clé, choisie aléatoirement. Plus la longueur de la clé augmente, plus cette méthode est incassable, mais elle nécessite un partage des clés entre les parties de l'échange de messages et un stockage de ces clés en mémoire pendant tout le temps que dure la transaction

Cryptographie moderne

Les cryptages simples étaient « relativement » aisés à contourner par les crypto-analystes. Il a fallu trouver de nouvelles méthodes de cryptage. Ces nouvelles méthodes de la cryptographie moderne reposent sur des théorèmes et des preuves mathématiques et utilisent extensivement dans leurs opérations de cryptage, de grands nombres premiers (pour le RSA), qui permettent d'éviter les répétitions de séquences de caractères dans les messages codés.

- le DES

  Dans cet algorithme, développé par IBM pour le gouvernement américain, le texte à crypter est décomposé en blocs de 64 bits. Puis 19 opérations différentes sont appliquées à chaque bloc. Ces opérations comprennent des transpositions, des permutations de paquets de bits et des opérations XOR avec la clé de cryptage, elle-même modifiée au cours du

traitement. L'algorithme a été conçu pour permettre le décryptage du message avec la même clé. L'avantage principal du DES, c'est qu'il est possible de créer simplement des circuits électroniques prenant en charge le cryptage. L'inconvénient majeur de cet algorithme, c'est qu'il permet (nous n'allons pas le démontrer ici) la modification du message une fois crypté, par copie ou permutation de blocs de 64 bits. Pour empêcher cela, le DES chaîné a été introduit.

- le DES chaîné

Pour pallier à la faiblesse de l'algorithme DES, il a fallu trouver un moyen de rendre détectable toute modification du message après le cryptage. Pour cela, le DES chaîné effectue une opération XOR entre le bloc de 64 bits à crypter et le bloc crypté précédent. Plusieurs options de cryptage existent même pour permettre une adaptation de l'algorithme à des besoins différents. Cependant, en utilisant une méthode de cassage de code bien connue, la « loterie chinoise », il est envisageable que le DES, même chaîné simplement, puisse être cassé. Pour contrer cette méthode, le double cryptage (on re-crypte le message déjà crypté), puis le triple cryptage (faisant intervenir deux clés, avec double cryptage par la première clé et un décryptage par la deuxième clé) ont été imaginés.

- le RSA

Le RSA est un algorithme à clés privées et publiques qui permet qu'un message encrypté avec une clé publique puisse être décrypté en utilisant la clé privée correspondante. Les opérations effectuées sont assez simples (des calculs de modulo) mais font intervenir des grands nombres premiers ainsi que des opérations entre ces nombres, ce qui rend impossible (du moins dans un référentiel de temps humain) le cassage de la clé privée.

La signature électronique

La signature électronique est le procédé par lequel on permet de garantir qu'un message à bien été issu d'une personne, que cette personne ne puisse révoquer ce message, et que le destinataire du message ne puisse modifier le message sans que cela soit détecté.

Pour implémenter une solution à ces trois problèmes, il est possible soit de crypter le message à l'aide d'une clé secrète, soit de générer un bloc de texte supplémentaire de manière à ce que deux messages différents ne puissent générer ledit bloc et que l'on ne puisse décrypter ce hachage du message. Dans tous les cas, l'intervention d'une autorité en tant que tiers de confiance est nécessaire.

La signature électronique emploie fréquemment les principes de contrôle d'intégrité, pour l'assurance de la non-modification du message, et l'authentification pour la garantie de l'identité du signataire.

Le contrôle d'intégrité

Afin de valider l'intégrité du contenu d'un message, et s'assurer ainsi que personne n'a pu le modifier, il a été nécessaire d'imaginer des moyens de calculer un code de vérification de la validité du message.

Ces méthodes de contrôle d'intégrité sont dérivées principalement de la recherche sur les codes de contrôle d'erreurs (bits de parités, CRC et checksums) utilisés dans les communications de données sur les réseaux de systèmes informatiques afin de garantir, ou du moins détecter, que le message n'ait subi aucune perturbation.

Cependant, afin de garantir une sécurité optimale lors d'une transaction entre des parties, il a fallu trouver des techniques plus fiables et moins aisées à contourner. De façon basique, on va calculer, pour chaque bloc de caractères du message, une somme de bits nous décrivant le contenu du message ; et tous ces codes générés sont envoyés à la suite du message, pour que le destinataire, lui-même connaissant le mode de calcul des codes de vérification, puisse s'assurer de la cohérence du message reçu par rapport à celui qui a été envoyé.

Ces calculs d'intégrité sont appelés checksum (sommes de vérification) et peuvent répondre à deux conditions :

- sans collision : il faut s'assurer que deux messages sources différents, quels qu'ils soient, ne puissent jamais donner le même résultat de calcul d'intégrité
- checksum avec clé : le code de contrôle d'intégrité ainsi calculé peut lui-même être crypté à l'aide d'un algorithme à clé secrète afin que le code de contrôle ne puisse être modifié par un tiers

L'authentification

L'authentification des personnes, des machines et même des processus sur un réseau est devenu une nécessité dès l'apparition des premiers réseaux, de manière à tenter de garantir à chaque intervenant du réseau que rien ni personne ne peut s'approprier ses droits et que ses correspondants sont bien ceux qu'il pense être.

Les solutions d'authentification existantes sont multiples et très liées à l'environnement dans lequel elles sont utilisées et au niveau de sécurité que l'on souhaite obtenir par leur implémentation.

Authentification sur canal « sûr » ou « sécurisé »

Si l'on considère que le mode de communication entre les deux parties est sûr, ou que le niveau de sécurité requis est faible, il n'est pas obligatoirement nécessaire d'implémenter une méthode d'authentification particulière.

Cela peut être le cas, par exemple, sur les réseaux cryptés militaires, qui utilisent un cryptage considéré comme incassable, ou un mode de communication réputé inviolable. Sur des réseaux de ce type, deux applications peuvent communiquer « en clair » sans avoir à se soucier de la sécurité des données, assurée par la couche physique de transport des données sur le réseau.

Mais, plus simplement, c'est aussi le cas de beaucoup d'applications Internet, n'exposant que peu, ou prou, de données sensibles, et qui considère la connexion HTTP suffisamment sûre (il y a peu de chance qu'une personne malveillante « écoute » les communications d'un particulier) pour permettre l'échange de données non cryptées entre le serveur Web et le client. De même, dans cet environnement, les pièces justificatives de l'authentification sur la machine hôte (login et mot de passe) sont souvent envoyées sans un quelconque cryptage.

Authentification par clé partagée

Un des principes de base pour l'authentification, est de partager une clé de cryptage qu'avec une et une seule autre entité. Ainsi, pour la communication entre ces entités, on utilisera cette clé et on pourra garantir l'identité de la personne qui a crypté le message.

Cette méthode, fiable s'il en est, nécessite de générer une clé par entité avec laquelle on veut communiquer. Et c'est d'ailleurs son principal défaut. En effet, outre le nombre considérable, et constamment augmentant, de clés qu'il faut générer, il ne garantit pas l'identification d'une nouvelle entité avec laquelle on souhaite communiquer. Pour cela, il faut procéder à la génération des clés sur un vecteur physique « sécurisé » (oral, échange de documents…), ce qui s'avère rapidement fastidieux.

Authentification avec cryptage par clé publique / clé privée

Dans cette méthode d'authentification, le principe est que lorsqu'une entité A veut communiquer avec une entité B, elle encrypte ses messages avec la clé publique de cette entité (qui peut les décrypter à l'aide de sa clé privée). En retour l'entité B, encrypte ses messages avec la clé publique de A (dont elle disposait dans sa propre base de données), messages qui ne pourront être lus que par la vraie entité A.

A nouveau, cette méthode implique de disposer d'un grand nombre de clés, une pour chaque entité avec laquelle on désire communiquer. Ce problème est résolu par l'utilisation de centres de distribution de clés (KDC : Key Distribution Center), des tiers de confiance qui vont garder l'ensemble des clés publiques de toutes les entités du réseau dans leur base de données. Ainsi, on demande au KDC la clé de l'entité avec laquelle on souhaite communiquer.

Une autre variante, utilisant aussi un KDC, est de demander à celui-ci d'établir une communication avec l'autre entité. Ainsi, aucune clé ne circule jamais sur le réseau. Mieux encore, cette dernière méthode permet de réutiliser la méthode d'authentification par clés partagées, puisqu'il suffit à chaque entité du réseau de partager une clé unique avec le KDC ; et c'est alors lui qui va garantir les identités des parties. Cependant cette méthode implique que l'on fasse entièrement confiance au KDC, puisque c'est lui qui va décrypter et encrypter les messages à la demande. Il a donc accès à l'ensemble des messages échangés sur le réseau.

Panorama des solutions d'authentification

Les solutions d'authentification ont beaucoup évolué, à la mesure des évolutions des réseaux, notamment leur ouverture sur le reste du monde. Nous présentons ici un aperçu, loin d'être exhaustif de quelques solutions d'authentification qui sont utilisées, ou qui ont été utilisées, sans toutefois entrer dans le détail de leur implémentation.

- PAP (Password Authentication Protocol)

  Ce protocole fut conçu par Cisco, pour la connexion avec identification sur un serveur. Les premières connexions à l'Internet avec le fournisseur d'accès Internet se faisaient avec ce protocole. L'inconvénient majeur du protocole PAP est que le mot de passe transite « en clair » sur le réseau…

- CHAP (Challenge Handshake Authentication Protocol)

Ce protocole fut imaginé pour remplacer le protocole PAP, trop peu sûr. Le principe est que lors de la demande d'authentification, le serveur demande au client de décrypter un message crypté (le Challenge) avec une clé partagée par le serveur et le client seulement. Si le client arrive à décrypter le message et le renvoie décrypté, alors il y accord d'authentification (Handshake) entre les deux parties. Cet échange est réitéré plusieurs fois pendant la durée de la communication.

- Les infrastructures à clés publiques (Public Key Infrastructure)

Une infrastructure à clé publique est destinée à faire en sorte de faciliter et de fiabiliser l'utilisation des cryptages à clé publique / clé privée dans les échanges informatiques. Elle est composée de trois parties : l'autorité de certification qui émet les certificats des parties (clés publiques), l'autorité d'authentification qui s'attache à garantir l'identité d'une partie à certifier avant de soumettre une demande de certificat à l'autorité de certification, et un centre de distribution des clés.

- Kerberos

Kerberos est le système d'authentification qui fait l'objet de ce document. Nous le détaillerons plus en profondeur dans la deuxième partie du document, mais nous pouvons signaler ici qu'il s'agit d'un protocole d'échanges d'authentification à clés privées et partagées, conçu de manière à laisser le moins possible de failles ouvertes.

- Les cartes à puce

Cette solution d'authentification est souvent utilisée dans des milieux où l'accès à une ressource nécessite une authentification forte, et où le nombre de personnes à authentifier reste faible. Le principe est de distribuer à ces personnes des cartes à puces contenant la clé d'authentification et qu'un périphérique authentifie la personne, parfois à l'aide d'un mot de passe supplémentaire, de manière à lui donner accès ou non à une ressource.

L'authentification par Kerberos

Présentation

Kerberos est le nom grec de Cerbère, le chien à trois têtes, gardien des enfers, mais Kerberos est aussi et surtout un protocole d'authentification, basé sur des travaux décrits par Needham et Schroeder, destiné à sécuriser l'authentification sur des réseaux ouverts. Kerberos a été créé au MIT (Massachusetts Institute of Technology), dans le cadre du projet Athena, principalement par Miller et Neuman.

Kerberos est un protocole ouvert et une implémentation gratuite fournie avec son code source (afin de permettre la vérification de la fiabilité du logiciel) est disponible sur le site web du MIT, mais il en aussi existe plusieurs versions commerciales disponibles si l'on souhaite disposer d'un support contractuel.

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

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 personnes ne puisse se faire passer pour elles, les mots de passes choisis sont suffisamment longs et complexes pour résister pendant une durée acceptable à une attaque de type « force brute », les horloges internes des machines sont *relativement* synchrone (une grande partie du protocole repose sur ce principe, et c'est sa différence principale avec le protocole décrit par Needham et Schroeder) et, enfin, les identifiants uniques de parties ne sont pas recyclés, ou seulement sur une longue période.

Nous reviendrons plus en détail sur certaines de ces conditions lors de l'étude des fonctionnalités et des divers constituants du protocole Kerberos.

Architecture

L'architecture de Kerberos s'articule autour de deux services d'authentification principaux : le Service d'Authentification et le Service d'Attribution de Tickets.

Nous verrons plus tard que ce qui permet au serveur Kerberos de déterminer quel service doit répondre à une requête d'authentification est le type et le contenu du message inclus dans la requête.

Pour l'heure nous allons nous attacher à l'étude des différents constituants de Kerberos.

- 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 du réseau doit tout d'abord s'adresser au service d'authentification pour pouvoir communiquer avec le service d'attribution de tickets

- Le Service d'Attribution de Tickets

Ce service va permettre de distribuer, aux clients authentifiés à l'aide du service d'authentification, les tickets qui vont leur permettre de communiquer avec les autres parties du réseau. Nous verrons plus tard en quoi consistent ces tickets. Le service d'attribution de tickets va aussi rendre possible l'authentification inter-royaumes, que nous détaillerons un peu plus loin

- 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 stockée sur la même machine que le service d'authentification ou le service d'attribution de tickets

- 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

- Le Serveur d'Administration de Kerberos

Le serveur d'administration de Kerberos 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, changements de mots de passe…)

Dans la spécification originale de Kerberos, en langue anglaise, le Serveur d'Authentification est dénommé Authentication Server (AS), le Service d'Attribution de Tickets est appeleé Ticket Granting Service (TGS), le Royaume est le Realm, et le Serveur d'Administration de Kerberos est le Kerberos Administration Server (KADM). On retrouvera notamment toutes ces initiales dans les noms des messages Kerberos.

En ce qui concerne les interactions entre les différents éléments de l'architecture Kerberos, chacun de ces éléments a un rôle bien défini dans le fonctionnement de Kerberos. L'architecture du système Kerberos a été décomposée ainsi afin de garantir une sécurité optimale aux utilisateurs du système.

Fonctionnement

Après avoir étudié l'ensemble des problèmes de sécurité posés dans les réseaux de systèmes informatiques et leurs solutions dans la première partie du document, l'étude du fonctionnement de Kerberos devrait apparaître assez simple. En fait le protocole lui-même se trouve être très simple, même si la conception d'un système d'authentification fiable soit complexe.

Nota : 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'initiale du mot « Clé » lui-même et I la ou les initiales du possesseur de la clé ; le cryptage de données sera, quant à lui, noté C_I(DONNEES), où DONNEES représentera une série de données (séparées par des virgules) qui seront cryptées par la clé de I.

Survol

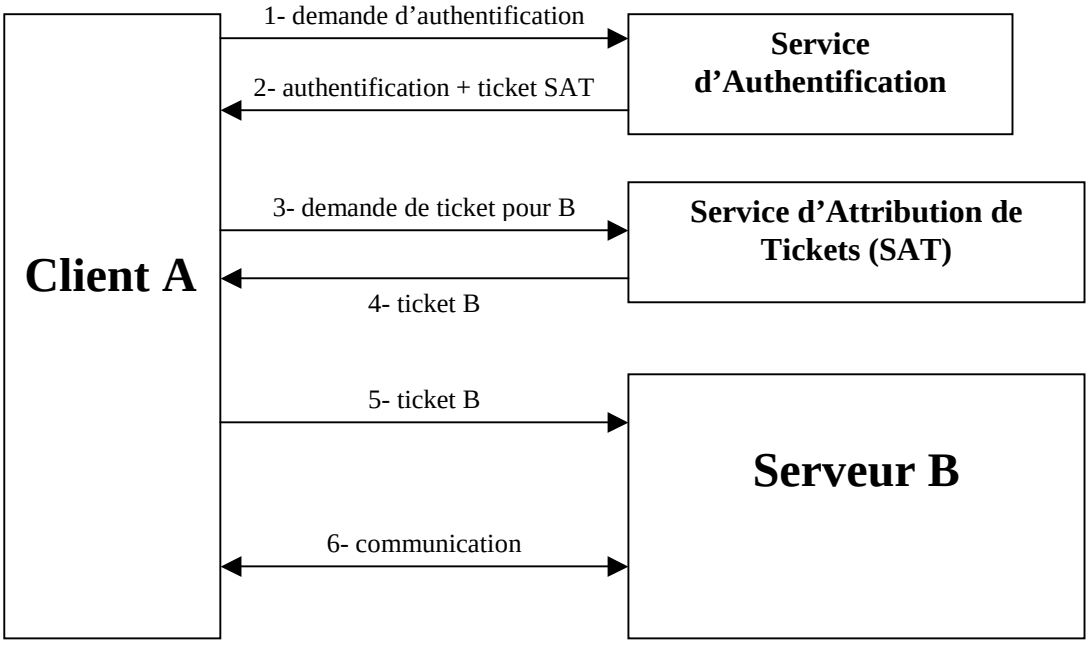
Nous allons décrire succinctement dans ce paragraphe l'utilisation de base du système Kerberos.

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



L'authentification

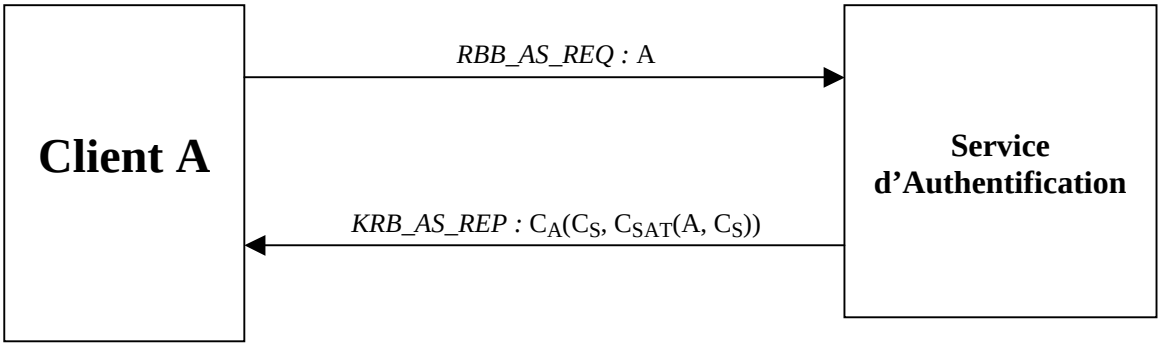
La phase d'authentification est la phase qui conditionne le reste de l'utilisation du système. Elle n'est effectuée qu'une seule fois, alors que nous verrons que les autres phases seront déroulées à chaque nouvelle authentification sur un nouveau serveur.

Pour débuter 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 gérées 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 Ticket 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.

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_REG 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 un timestamp, c'est-à-dire le nombre de millisecondes écoulées depuis le 1er Janvier 1970 à minuit.

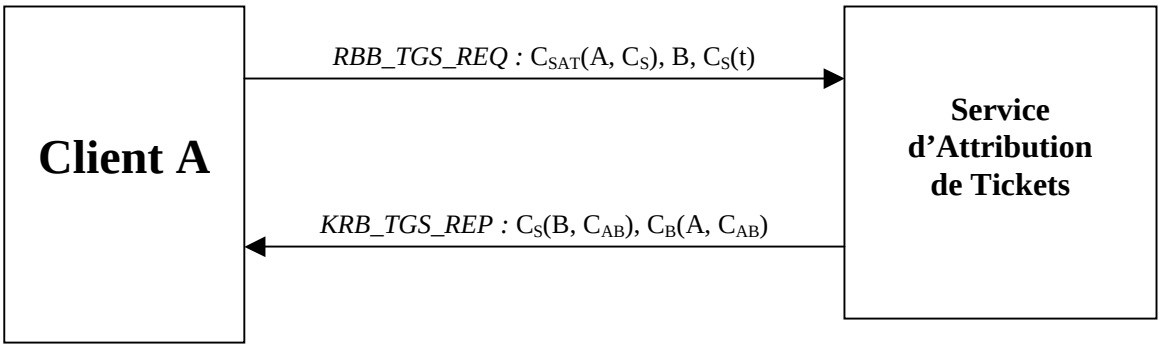
Quand le Service d'Attribution de Tickets reçoit le message KRB_TGS_REQ, il décrypte le ticket qui lui est destiné et qui contient l'identifiant du client A, ainsi que la clé de session CS avec laquelle il décrypte le timestamp pour vérifier que le message a été émis dans un intervalle de temps raisonnable après la récupération du ticket auprès du Service d'Authentification. Cela impose l'une des conditions que nous avons vues précédemment, qui est 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 quant lui principalement constitué de l'identité du client A et de la clé partagée C_{AB}, le tout crypté par la clé C_B du serveur B de manière à ce que seul lui puisse le décrypté.

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



L'échange d'attribution d'un ticket.

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 32bits 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 peu 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 persistant du ticket sur le système client

(le vol principalement). D'un autre côté, 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 trouve 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 post-daté (MAY-POSTDATE et POSTDATED)

Dans le cas d'application 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 d'utiliser le ticket dans un futur assez éloigné. Lorsque le client désire 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 certaines 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 couvert de l'identité (et donc de l'authentification) de l'entité qui a émis le ticket proxy, c'est-à-dire qui donné 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 validé par le serveur Kerberos, c'est-à-dire tant que la date d'utilisation du ticket n'est pas arrivée et que le possesseur du ticket ne l'a pas soumis à la validation du serveur Kerberos.

Utilisation du ticket pour l'accès à une ressource

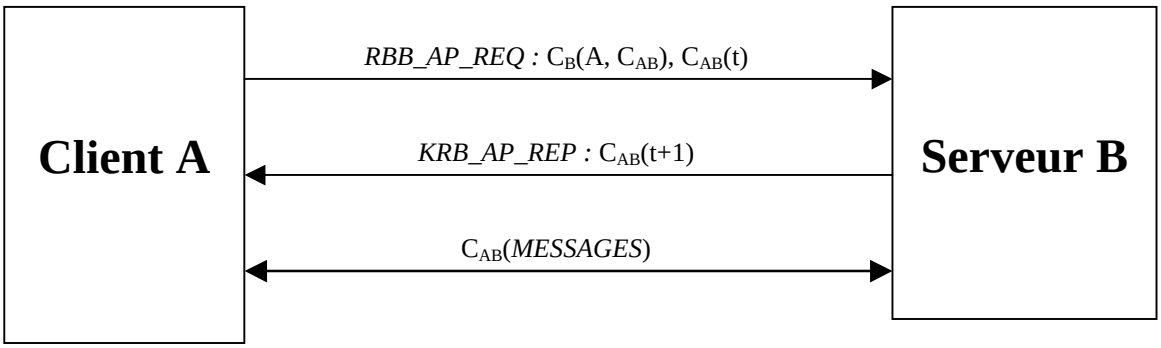
A partir du moment où le client a obtenu un ticket pour l'accès à une ressource, il peut envoyer un message KRB_AP_REQ au serveur auquel il désire accéder.

Le message KRB_AP_REQ contient le ticket pour le serveur, c'est-à-dire l'identifiant du client et la clé partagée entre le client et le serveur (dans notre exemple, il s'agit de CAB), 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 envoyer 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 construit de manière à prendre en charge bine plus que l'authentification : l'autorisation. En effet, il est possible de renseigner des « capacités » (« capability » dans la spécification originale de Kerberos).

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 affectés à une adresse réseau particulière comme le sont les autres tickets pour limiter l'usage des tickets dérobés.

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 serveur 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 types de situations peuvent se présenter : soit le client connaît le Royaume auquel appartient le serveur auprès duquel il souhaite s'authentifier, soit il ne le connaît pas. Autant que possible, le client essaye de fournir cette information au Service d'Attribution de Tickets de son Royaume. Le Service d'Attribution de Tickets, quant 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 Ticket va utiliser la clé du Service d'Attribution de Tickets du Royaume le plus « proche » (proximité basée sur des informations de configuration) pour crypter le ticket d'authentification. Et tant que le Royaume n'a pas été trouvé, le processus continue récursivement.

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 émis 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.

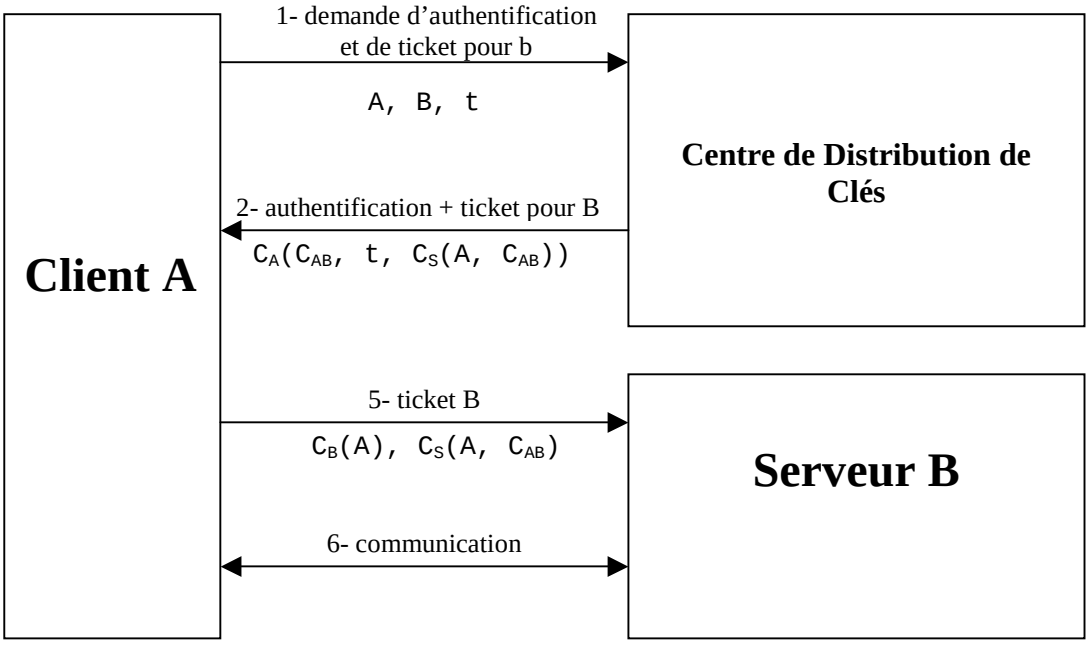
Nouveautés de Kerberos v5

Faiblesses 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.

- Faiblesses structurelles

Comme nous le voyons sur le schéma suivant qui détaille le mode de fonctionnement de Kerberos v4, le nombre de parties impliquées cette version est inférieur à celui utilisé dans la version 5. Et ce pour la simple raison que le Centre de Distribution de Clés (Key Distribution Center) 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 CS pour que le serveur B puisse recevoir le ticket émit par 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 à l'aide 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 dérober 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.

- 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 surcroît de besoin en puissance de traitement pour le Centre de Distribution de Clés.

- Faiblesses diverses

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

Améliorations apportées par Kerberos v5

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

- 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 CRC-32 (qui n'est pas sans collision) 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.

- 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.

- 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

- 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.

- 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.

- 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.

Implémentations

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 peu trouver la trace dans de nombreux systèmes. De plus, le MIT fournit une implémentation de référence, au code source ouvert, accessible à quiconque désire pousser plus loin l'étude ou l'implémentation de Kerberos.

Enfin Kerberos fait l'objet d'une RFC (Request For Comments), numéro 1510, qui détaille la spécification du protocole, des données et des messages. Cette RFC est très utile à toute personne qui souhaite étudier Kerberos.

Au coeur 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, se conformant à la RFC-1510.

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 ; et même si le nom « domaine » reste, il s'agit effectivement de royaumes qui sont configurés dans les environnements Windows 2000. L'implémentation de Kerberos faite par Microsoft reste compatible avec les implémentations qui suivent la RFC-1510, mais elle apporte des spécificités propres aux environnements Microsoft qui ne sont donc pas utilisables dans des environnements hétérogènes (extensions Service-for-User et Service-for-User-to-Proxy sous Windows 2003) et certaines fonctionnalités de la RFC-1510 ne sont pas supportées (tickets post-datés, DES-CBC-MD4 et cryptage NULL, transitivité inter-royaumes avec des royaumes non Windows 2000…).

Intégré aux langages de programmation

Pour développer des applications qui soient compatibles avec Kerberos et qui utilisent ses fonctionnalités d'authentification, le meilleur moyen est d'utiliser la GSS-API fournie par l'implémentation Kerberos du MIT. Cette librairie, écrite en C est directement utilisable dans tout programme C. C'est le choix que l'on fera la plupart du temps dans le développement d'applications pour les environnements Unix.

En ce qui concerne l'implémentation au sein de la plateforme Java, Kerberos est implémenté comme un plugin du Java Authentication and Authorization Service (JAAS). JAAS est un package qui permet d'offrir des fonctionnalités d'authentification et de contrôle d'accès aux développements d'applications Java, quelque soit la méthode d'authentification sous-jacente. En effet JAAS offre une architecture de haut niveau à laquelle il suffit de brancher des modules (plugins) pour disposer d'une implémentation quelconque d'une méthode d'authentification. JAAS a été intégré dans la version 1.4 du Java Development Kit et Java offre donc depuis des possibilités d'authentification par Kerberos.

L'environnement de développement Microsoft, quant à lui, offre une interface de programmation de haut niveau, nommée Security Support Provider Interface (SSPI), qui, comme JAAS, supporte Kerberos comme implémentation un package sous-jacent. Il est donc possible d'utiliser Kerberos pour utiliser l'authentification dans des applications sous Windows.

Enfoui dans les outils

Enfin, 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 des 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 (firewalls) 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.

L'authentification dans le futur

Pour clore ce document nous allons survoler rapidement les principes qui feront l'authentification du futur.

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 clés publiques, il sera sans doute nécessaire de l'y adapter
- cartes à puces : l'intérêt des cartes à puces, c'est qu'elles permettent de ne jamais mettre au jour la clé du l'utilisateur, surtout dans le cas où celui-ci utilise une station de travail en laquelle il ne peut avoir confiance (i.e. station publique)
- administration à distance : pour le moment, la spécification de Kerberos ne prend pas en charge l'administration de la base de données 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 Kerberos entre les serveurs Kerberos du domaine

Renforcement des cryptages

Les technologies évoluent rapidement, les puissances de calcul augmentent de façon linéaire et un cryptage que l'on 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, comme par exemple le cryptage quantique.

Utilisation de la biométrie

Enfin une des pistes actuellement poursuivie, et implémentée avec plus ou moins de succès, est la biométrie ; c'est-à-dire l'utilisation de la mesure d'une partie du corps humain pour la génération de l'authentification.

Le but recherché ici est de produire une clé de cryptage, un moyen d'authentification, qui ne puisse ni être falsifiée, ni dupliquée, ni volée. Les systèmes de biométrie actuellement en développement se basent sur les empreintes digitales, sur les empreintes vocales ou encore sur la mesure du fond de l'œil. Cela ressemble à de la science-fiction, et pourtant cela existe déjà !

Glossaire

attaque force brute : ce type d'attaque d'un cryptage consiste à essayer toutes les combinaisons de lettres possibles pour découvrir un mot de passe

denial of service : ce type d'attaque d'un système consiste à exploiter une faille réseau du système de manière à le rendre inopérant et, éventuellement, se faire passer pour lui pendant le temps où il le sera

loterie chinoise : ce type d'attaque consiste à effectuer une attaque force brute en répartissant la charge de calcul sur un nombre très important de petits calculateurs (postes de télévision, de radio…), par exemple, un dans chaque foyer chinois

social engineering : cette attaque consiste à appeler physiquement une secrétaire ou un agent d'entretien, pour obtenir une information (numéro de téléphone, mot de passe, adresse réseau,…) qui pourra aider dans l'attaque d'un système.

Bibliographie

Voici les différentes sources qui ont permit, à plus ou moins grande contribution, l'élaboration de ce document. Ne sont pas référencés ici nombreux articles de magazines, ou sites web qui ont largement permit une première approche du protocole Kerberos.

[1] Site web principal de Kerberos : http://web.mit.edu/kerberos/www

[2] The Evolution of the Kerberos Authentication Service – Kohl-Neuman-Ts'o – 1991

[3] RFC 1510 – Kohl-Neuman – 1993

[4] Computer Networks (3rd edition) – Andrew S. Tanenbaum – Prentice Hall – 1996