Vue d'ensemble de Kerberos – Un service d'authentification pour systèmes de réseaux ouverts

Contenu

Introduction

Dans un environnement informatique à réseau ouvert, un poste de travail ne peut pas identifier de manière fiable ses utilisateurs sur les services de réseau. Kerberos fournit une autre approche par laquelle un service d'authentification tiers de confiance est utilisé pour vérifier l'identité des utilisateurs. Ce document donne un aperçu du modèle d'authentification Kerberos mis en œuvre pour le projet Athena du MIT. Il décrit les protocoles utilisés par les clients, par les serveurs et par Kerberos pour réaliser l'authentification. Il décrit également la gestion et la reproduction requises de la base de données. Les fenêtres de Kerberos sont décrites telles que vues par l'utilisateur, le programmeur et l'administrateur. Enfin, le rôle de Kerberos est donné dans le contexte du projet Athena, avec une liste des applications qui utilisent actuellement Kerberos pour l'authentification des utilisateurs. Nous décrivons l'ajout de l'authentification Kerberos au système de fichiers en réseau de Sun dans la cadre d'une étude de cas pour intégrer Kerberos à une application existante.

Auteurs Kerberos

- Jennifer G. Steiner, Project Athena, Massachusetts Institute of Technology, Cambridge, MA 02139, steiner@ATHENA.MIT.EDU
- Clifford Neuman, Département d'informatique, FR-35, Université de Washington, Seattle, WA 98195, bcn@CS.WASHINGTON.EDU. Clifford Neuman était membre du personnel du Projet Athena pendant la phase de conception et de mise en oeuvre initiale de Kerberos.
- Jeffrey I. Schiller, Project Athena, Massachusetts Institute of Technology, Cambridge, MA 02139, jis@ATHENA.MIT.EDU

Introduction à Kerberos

Cet article donne un aperçu de Kerberos, un système d'authentification conçu par Miller et Neuman. pour les environnements d'informatique en réseau ouvert, et décrit notre expérience de l'utiliser au projet Athena du MIT. Dans la section Motivation, nous expliquons pourquoi un nouveau modèle d'authentification est nécessaire pour les réseaux ouverts et quelles sont ses exigences. Qu'est-ce que Kerberos ? répertorie les composants du logiciel Kerberos et décrit comment ils interagissent pour fournir le service d'authentification. Dans la section Noms Kerberos, nous décrivons le schéma d'attribution de noms Kerberos.

Comment fonctionne Kerberos présente les éléments de base de l'authentification Kerberos - le ticket et l'authentificateur. Cela mène à une discussion sur les deux protocoles d'authentification : l'authentification initiale d'un utilisateur à Kerberos (analogue à la connexion), et le protocole d'authentification mutuelle d'un consommateur potentiel et d'un producteur potentiel d'un service réseau.

Kerberos a besoin d'une base de données d'informations sur ses clients ; la section Base de données Kerberos décrit la base de données, sa gestion et le protocole pour sa modification. La section Kerberos From the Outside Look In décrit l'interface Kerberos à ses utilisateurs, programmeurs d'applications et administrateurs. Dans la section Plus grande image, nous décrivons comment le Projet Athena Kerberos s'intègre dans le reste de l'environnement Athéna. Nous décrivons également l'interaction de différents domaines d'authentification Kerberos, ou royaumes ; dans notre cas, la relation entre le Projet Athena Kerberos et le Kerberos qui fonctionne au Laboratoire d'informatique du MIT.

Dans la section Problèmes et problèmes ouverts, nous mentionnons les problèmes et problèmes non résolus. La dernière section donne le statut actuel de Kerberos au Projet Athena. Dans l'annexe, nous décrivons en détail comment Kerberos est appliqué à un service de fichiers réseau pour authentifier les utilisateurs qui souhaitent accéder à des systèmes de fichiers distants.

Concepts Kerberos

Tout au long de ce document, nous utilisons des termes qui peuvent être ambigus, nouveaux pour le lecteur ou utilisés différemment ailleurs. Ci-dessous, nous indiquons notre utilisation de ces termes.

Utilisateur, Client, Serveur - Par utilisateur, nous entendons un être humain qui utilise un programme ou un service. Un client utilise également quelque chose, mais n'est pas nécessairement une personne ; il peut s'agir d'un programme. Les applications réseau sont souvent composées de deux parties ; un programme qui s'exécute sur une machine et demande un service distant, et un autre qui s'exécute sur la machine distante et exécute ce service. Nous les appelons respectivement côté client et côté serveur de l'application. Souvent, un client contacte un serveur au nom d'un utilisateur.

Chaque entité qui utilise le système Kerberos, qu'il s'agisse d'un utilisateur ou d'un serveur réseau, est en un sens un client, puisqu'elle utilise le service Kerberos. Ainsi, pour distinguer les clients Kerberos des clients d'autres services, nous utilisons le terme principal pour désigner une telle entité. Notez qu'un principal Kerberos peut être un utilisateur ou un serveur. (Nous décrivons le nom des identités Kerberos dans une section ultérieure.)

Service vs. Server : nous utilisons le service comme spécification abstraite de certaines actions à effectuer. Un processus qui exécute ces actions est appelé serveur. À un moment donné, plusieurs serveurs (généralement exécutés sur différentes machines) peuvent exécuter un service donné. Par exemple, à Athena, il y a un serveur de connexion BSD UNIX qui fonctionne sur chacune de nos machines de partage de temps.

Clé, Clé privée, Mot de passe - Kerberos utilise le chiffrement de clé privée. Chaque principal Kerberos se voit attribuer un grand nombre, sa clé privée, connue uniquement de ce principal et de Kerberos. Dans le cas d'un utilisateur, la clé privée est le résultat d'une fonction unidirectionnelle appliquée au mot de passe de l'utilisateur. Nous utilisons la clé comme raccourci pour la clé privée.

Informations d'identification : malheureusement, ce mot a une signification particulière pour le système de fichiers Sun Network et le système Kerberos. Nous indiquons explicitement si nous entendons des informations d'identification NFS ou Kerberos, sinon le terme est utilisé dans le sens anglais normal.

Master et Slave : il est possible d'exécuter le logiciel d'authentification Kerberos sur plusieurs machines. Cependant, il n'y a toujours qu'une seule copie définitive de la base de données Kerberos. La machine qui héberge cette base de données est appelée la machine maître, ou simplement la machine maître. D'autres machines peuvent posséder des copies en lecture seule de la base de données Kerberos, appelées esclaves.

Motivation derrière Kerberos

Dans un environnement informatique personnel non connecté en réseau, les ressources et les

informations peuvent être protégées en sécurisant physiquement l'ordinateur personnel. Dans un environnement informatique de partage de temps, le système d'exploitation protège les utilisateurs les uns des autres et contrôle les ressources. Pour déterminer ce que chaque utilisateur peut lire ou modifier, il est nécessaire que le système de partage de temps identifie chaque utilisateur. Ceci est effectué lorsque l'utilisateur se connecte.

Dans un réseau d'utilisateurs nécessitant des services à partir de plusieurs ordinateurs distincts, il existe trois approches possibles pour le contrôle d'accès : On ne peut rien faire, en s'appuyant sur la machine à laquelle l'utilisateur est connecté pour empêcher tout accès non autorisé ; on peut demander à l'hôte de prouver son identité, mais faire confiance à la parole de l'hôte quant à l'identité de l'utilisateur ; ou l'utilisateur peut demander de prouver son identité pour chaque service requis.

Dans un environnement fermé où toutes les machines sont sous contrôle strict, on peut utiliser la première approche. Lorsque l'organisation contrôle tous les hôtes qui communiquent sur le réseau, il s'agit d'une approche raisonnable.

Dans un environnement plus ouvert, il est possible de ne faire confiance qu'aux hôtes placés sous contrôle organisationnel. Dans ce cas, chaque hôte doit être tenu de prouver son identité. Les programmes rlogin et rsh utilisent cette approche. Dans ces protocoles, l'authentification est effectuée en vérifiant l'adresse Internet à partir de laquelle une connexion a été établie.

Dans l'environnement Athena, nous devons être en mesure d'honorer les demandes des hôtes qui ne sont pas sous contrôle organisationnel. Les utilisateurs disposent d'un contrôle total sur leurs stations de travail : ils peuvent les redémarrer, les activer de manière autonome ou même démarrer sur leurs propres bandes. En tant que telle, la troisième approche doit être adoptée ; l'utilisateur doit prouver son identité pour chaque service souhaité. Le serveur doit également prouver son identité. Il ne suffit pas de sécuriser physiquement l'hôte exécutant un serveur réseau ; quelqu'un d'autre sur le réseau peut se faire passer pour le serveur donné.

Notre environnement impose plusieurs exigences à un mécanisme d'identification. Premièrement, elle doit être sécurisée. Il doit être assez difficile de contourner ce problème pour qu'un attaquant potentiel ne trouve pas le mécanisme d'authentification comme étant la liaison faible. Une personne qui surveille le réseau ne doit pas pouvoir obtenir les informations nécessaires pour se faire passer pour un autre utilisateur. Deuxièmement, elle doit être fiable. L'accès à de nombreux services dépendra du service d'authentification. Si elle n'est pas fiable, le système des services dans son ensemble ne le sera pas. Troisièmement, il devrait être transparent. Dans l'idéal, l'utilisateur ne doit pas savoir que l'authentification a lieu. Enfin, il devrait être évolutif. De nombreux systèmes peuvent communiquer avec les hôtes Athena. Tous ces mécanismes ne seront pas pris en charge, mais les logiciels ne devraient pas se casser s'ils le faisaient.

Kerberos est le résultat de notre travail pour satisfaire les exigences ci-dessus. Lorsqu'un utilisateur se connecte à une station de travail, il se connecte. Pour autant que l'utilisateur puisse le savoir, cette identification initiale est suffisante pour prouver son identité à tous les serveurs réseau requis pendant la durée de la session de connexion. La sécurité de Kerberos repose sur la sécurité de plusieurs serveurs d'authentification, mais pas sur le système à partir duquel les utilisateurs se connectent, ni sur la sécurité des serveurs finaux qui seront utilisés. Le serveur d'authentification fournit à un utilisateur correctement authentifié un moyen de prouver son identité aux serveurs dispersés sur le réseau.

L'authentification est un élément fondamental d'un environnement réseau sécurisé. Si, par exemple, un serveur connaît l'identité d'un client, il peut décider s'il doit fournir le service, si l'utilisateur doit bénéficier de privilèges spéciaux, qui doit recevoir la facture du service, etc. En

d'autres termes, les systèmes d'autorisation et de comptabilité peuvent être construits en plus de l'authentification que Kerberos fournit, ce qui donne une sécurité équivalente à l'ordinateur personnel isolé ou au système de partage de temps.

Qu'est-ce que Kerberos ?

Kerberos est un service d'authentification tiers de confiance basé sur le modèle présenté par Needham et Schroeder. On fait confiance à chacun de ses clients en ce sens que le jugement de Kerberos quant à l'identité de chacun de ses autres clients est juste. Des horodatages (grands nombres représentant la date et l'heure actuelles) ont été ajoutés au modèle d'origine pour faciliter la détection de relecture. La relecture se produit lorsqu'un message est volé sur le réseau et envoyé ultérieurement. Pour une description plus complète de la relecture et d'autres problèmes d'authentification, voir Voydock et Kent.

Que fait Kerberos ?

Kerberos conserve une base de données de ses clients et de leurs clés privées. La clé privée est un grand nombre connu uniquement de Kerberos et du client auquel elle appartient. Dans le cas où le client est un utilisateur, il s'agit d'un mot de passe chiffré. Les services réseau nécessitant une authentification s'enregistrent auprès de Kerberos, tout comme les clients souhaitant utiliser ces services. Les clés privées sont négociées lors de l'enregistrement.

Parce que Kerberos connaît ces clés privées, il peut créer des messages qui convainquent un client que l'autre est vraiment ce qu'il prétend être. Kerberos génère également des clés privées temporaires, appelées clés de session, qui sont données à deux clients et à personne d'autre. Une clé de session peut être utilisée pour chiffrer les messages entre deux parties.

Kerberos offre trois niveaux de protection distincts. Le programmeur d'applications détermine lequel est approprié, selon les exigences de l'application. Par exemple, certaines applications exigent uniquement que l'authenticité soit établie au début d'une connexion réseau et peuvent supposer que d'autres messages d'une adresse réseau donnée proviennent de la partie authentifiée. Notre système de fichiers réseau authentifié utilise ce niveau de sécurité.

D'autres applications nécessitent l'authentification de chaque message, mais peu importe si le contenu du message est divulgué ou non. Pour cela, Kerberos fournit des messages sécurisés. Pourtant, un niveau de sécurité plus élevé est fourni par les messages privés, où chaque message est non seulement authentifié, mais également chiffré. Les messages privés sont utilisés, par exemple, par le serveur Kerberos lui-même pour envoyer des mots de passe sur le réseau.

Composants logiciels Kerberos

La mise en oeuvre d'Athena comprend plusieurs modules :

- Bibliothèque d'applications Kerberos
- bibliothèque de chiffrement
- bibliothèque de bases de données
- programmes d'administration de bases de données
- serveur d'administration
- serveur d'authentification

- logiciel de propagation de base de données
- programmes utilisateur
- applications

La bibliothèque d'applications Kerberos fournit une interface pour les clients d'applications et les serveurs d'applications. Il contient, entre autres, des routines de création ou de lecture de demandes d'authentification, ainsi que les routines de création de messages sécurisés ou privés.

Le chiffrement dans Kerberos est basé sur DES, la norme de chiffrement des données. La bibliothèque de chiffrement implémente ces routines. Plusieurs méthodes de cryptage sont fournies, avec des compromis entre vitesse et sécurité. Une extension du mode de chaînage de bloc de chiffrement DES (CBC), appelé mode de propagation CBC, est également fournie. Dans CBC, une erreur est propagée uniquement par le bloc courant du chiffrement, alors que dans PCBC, l'erreur est propagée dans tout le message. Cela rend le message entier inutile si une erreur se produit, plutôt qu'une partie. La bibliothèque de cryptage est un module indépendant et peut être remplacée par d'autres implémentations DES ou une autre bibliothèque de cryptage.

Un autre module remplaçable est le système de gestion des bases de données. La mise en oeuvre actuelle de la bibliothèque de bases de données Athena utilise ndbm, bien qu'Ingres ait été utilisé à l'origine. D'autres bibliothèques de gestion de bases de données pourraient également être utilisées.

Les besoins de la base de données Kerberos sont simples ; un enregistrement est conservé pour chaque entité principale, contenant le nom, la clé privée et la date d'expiration du entité principale, ainsi que certaines informations administratives. (La date d'expiration est la date après laquelle une entrée n'est plus valide. Il est généralement fixé à quelques années à venir lors de l'inscription.)

D'autres informations utilisateur, telles que le nom réel, le numéro de téléphone, etc., sont conservées par un autre serveur, le serveur de noms Hesiod. De cette manière, les informations sensibles, à savoir les mots de passe, peuvent être traitées par Kerberos, en utilisant des mesures de sécurité relativement élevées ; tandis que les informations non sensibles conservées par Hesiod sont traitées différemment ; il peut, par exemple, être envoyé sans cryptage sur le réseau.

Les serveurs Kerberos utilisent la bibliothèque de base de données, tout comme les outils d'administration de la base de données.

Le serveur d'administration (ou serveur KDBM) fournit une interface réseau en lecture-écriture à la base de données. Le côté client du programme peut être exécuté sur n'importe quelle machine du réseau. Cependant, le serveur doit s'exécuter sur la machine hébergeant la base de données Kerberos afin d'apporter des modifications à la base de données.

Le serveur d'authentification (ou serveur Kerberos), en revanche, effectue des opérations en lecture seule sur la base de données Kerberos, à savoir l'authentification des principaux et la génération des clés de session. Comme ce serveur ne modifie pas la base de données Kerberos, il peut s'exécuter sur une machine hébergeant une copie en lecture seule de la base de données Kerberos maître.

Le logiciel de propagation de base de données gère la réplication de la base de données Kerberos. Il est possible d'avoir des copies de la base de données sur plusieurs machines différentes, avec une copie du serveur d'authentification exécuté sur chaque machine. Chacune de ces machines esclaves reçoit une mise à jour de la base de données Kerberos de la machine

maître à intervalles donnés.

Enfin, il existe des programmes d'utilisateur final pour se connecter à Kerberos, modifier un mot de passe Kerberos et afficher ou détruire des tickets Kerberos (les tickets sont expliqués plus loin).

Noms Kerberos

Une partie de l'authentification d'une entité est son nom. Le processus d'authentification est la vérification que le client est celui nommé dans une requête. En quoi consiste un nom ? Dans Kerberos, les utilisateurs et les serveurs sont nommés. En ce qui concerne le serveur d'authentification, ils sont équivalents. Un nom se compose d'un nom principal, d'une instance et d'un domaine, exprimé en name.instance@realm.

Le nom principal est le nom de l'utilisateur ou du service. L'instance est utilisée pour distinguer les variations du nom principal. Pour les utilisateurs, une instance peut comporter des privilèges spéciaux, tels que les instances « root » ou « admin ». Pour les services dans l'environnement Athena, l'instance est généralement le nom de la machine sur laquelle le serveur s'exécute. Par exemple, le service rlogin a différentes instances sur différents hôtes : rlogin.priam est le serveur rlogin sur l'hôte nommé priam. Un ticket Kerberos ne convient qu'à un seul serveur nommé. Ainsi, un ticket distinct est nécessaire pour accéder à différentes instances du même service. Le domaine est le nom d'une entité administrative qui gère les données d'authentification. Par exemple, différentes institutions peuvent avoir chacune leur propre machine Kerberos, qui héberge une base de données différente. Ils ont différents royaumes Kerberos. (Les royaumes sont abordés plus en détail dans Interaction avec d'autres Kerberi.)

Fonctionnement de Kerberos

Cette section décrit les protocoles d'authentification Kerberos. Comme mentionné ci-dessus, le modèle d'authentification Kerberos est basé sur le protocole de distribution de clé Needham et Schroeder. Lorsqu'un utilisateur demande un service, son identité doit être établie. Pour ce faire, un ticket est présenté au serveur, accompagné de la preuve que le ticket a été initialement émis à l'utilisateur, et non volé. Il existe trois phases d'authentification via Kerberos. Dans la première phase, l'utilisateur obtient des informations d'identification à utiliser pour demander l'accès à d'autres services. Dans la deuxième phase, l'utilisateur demande l'authentification d'un service spécifique. Dans la dernière phase, l'utilisateur présente ces informations d'identification au serveur final.

Informations d'identification Kerberos

Deux types d'informations d'identification sont utilisés dans le modèle d'authentification Kerberos : tickets et authentificateurs. Tous deux sont basés sur un cryptage à clé privée, mais ils sont cryptés à l'aide de clés différentes. Un ticket est utilisé pour transmettre en toute sécurité l'identité de la personne à laquelle le ticket a été émis entre le serveur d'authentification et le serveur d'extrémité. Un billet transmet également des informations qui peuvent être utilisées pour s'assurer que la personne qui utilise le billet est la même personne à laquelle il a été émis. L'authentificateur contient les informations supplémentaires qui, comparées à celles du ticket, prouvent que le client présentant le ticket est le même que celui auquel le ticket a été émis.

Un ticket est bon pour un seul serveur et un seul client. Il contient le nom du serveur, le nom du

client, l'adresse Internet du client, un horodatage, une durée de vie et une clé de session aléatoire. Ces informations sont chiffrées à l'aide de la clé du serveur pour lequel le ticket sera utilisé. Une fois le ticket émis, il peut être utilisé plusieurs fois par le client nommé pour accéder au serveur nommé, jusqu'à l'expiration du ticket. Notez que le ticket étant chiffré dans la clé du serveur, il est sûr de permettre à l'utilisateur de transmettre le ticket au serveur sans avoir à se soucier de la modification du ticket par l'utilisateur.

Contrairement au ticket, l'authentificateur ne peut être utilisé qu'une seule fois. Un nouveau doit être généré chaque fois qu'un client souhaite utiliser un service. Cela ne présente pas de problème car le client est capable de créer l'authentificateur lui-même. Un authentificateur contient le nom du client, l'adresse IP du poste de travail et l'heure actuelle du poste de travail. L'authentificateur est chiffré dans la clé de session qui fait partie du ticket.

Obtenir le ticket Kerberos initial

Lorsque l'utilisateur se rend à un poste de travail, une seule information peut prouver son identité : le mot de passe de l'utilisateur. L'échange initial avec le serveur d'authentification est conçu pour minimiser les risques de compromission du mot de passe, tout en ne permettant pas à un utilisateur de s'authentifier correctement sans connaître ce mot de passe. Le processus de connexion semble être le même que celui de connexion à un système de partage de temps. Mais en coulisses, c'est tout autre chose.

L'utilisateur est invité à saisir son nom d'utilisateur. Une fois saisie, une requête est envoyée au serveur d'authentification contenant le nom de l'utilisateur et le nom d'un service spécial appelé service d'octroi de tickets.

Le serveur d'authentification vérifie qu'il connaît le client. Si c'est le cas, il génère une clé de session aléatoire qui sera ensuite utilisée entre le client et le serveur d'octroi de tickets. Il crée ensuite un ticket pour le serveur d'octroi de tickets qui contient le nom du client, le nom du serveur d'octroi de tickets, l'heure actuelle, une durée de vie pour le ticket, l'adresse IP du client et la clé de session aléatoire qui vient d'être créée. Tout ceci est chiffré dans une clé connue uniquement du serveur d'octroi de tickets et du serveur d'authentification.

Le serveur d'authentification renvoie ensuite le ticket, accompagné d'une copie de la clé de session aléatoire et d'informations supplémentaires, au client. Cette réponse est chiffrée dans la clé privée du client, connue uniquement de Kerberos et du client, qui est dérivée du mot de passe de l'utilisateur.

Une fois la réponse reçue par le client, l'utilisateur est invité à saisir son mot de passe. Le mot de passe est converti en clé DES et utilisé pour déchiffrer la réponse du serveur d'authentification. Le ticket et la clé de session, ainsi que d'autres informations, sont stockés pour une utilisation future, et le mot de passe de l'utilisateur et la clé DES sont effacés de la mémoire.

Une fois l'échange terminé, le poste de travail possède les informations qu'il peut utiliser pour prouver l'identité de son utilisateur pendant la durée de vie du ticket d'octroi. Tant que le logiciel de la station de travail n'a pas été modifié auparavant, il n'existe aucune information permettant à quelqu'un d'autre d'emprunter l'identité de l'utilisateur au-delà de la durée de validité du ticket.

Demander un service Kerberos

Pour le moment, faisons semblant que l'utilisateur a déjà un ticket pour le serveur désiré. Afin d'accéder au serveur, l'application crée un authentificateur contenant le nom et l'adresse IP du

client, ainsi que l'heure actuelle. L'authentificateur est ensuite chiffré dans la clé de session qui a été reçue avec le ticket pour le serveur. Le client envoie ensuite l'authentificateur avec le ticket au serveur d'une manière définie par l'application individuelle.

Une fois que l'authentificateur et le ticket ont été reçus par le serveur, le serveur déchiffre le ticket, utilise la clé de session incluse dans le ticket pour déchiffrer l'authentificateur, compare les informations du ticket avec celles de l'authentificateur, l'adresse IP à partir de laquelle la demande a été reçue et l'heure actuelle. Si tout correspond, il permet à la demande de continuer.

On suppose que les horloges sont synchronisées en quelques minutes. Si l'heure de la requête est trop longue dans le futur ou dans le passé, le serveur traite la requête comme une tentative de relecture d'une requête précédente. Le serveur est également autorisé à suivre toutes les demandes passées avec des horodatages qui sont toujours valides. Pour éviter d'autres attaques de relecture, une demande reçue avec le même ticket et l'horodatage que celui déjà reçu peut être rejetée.

Enfin, si le client spécifie qu'il souhaite que le serveur prouve également son identité, le serveur en ajoute un à l'horodatage que le client a envoyé dans l'authentificateur, chiffre le résultat dans la clé de session et renvoie le résultat au client.

Àla fin de cet échange, le serveur est certain que, selon Kerberos, le client est celui qu'il dit être. Si une authentification mutuelle se produit, le client est également convaincu que le serveur est authentique. En outre, le client et le serveur partagent une clé que personne d'autre ne connaît, et peuvent supposer en toute sécurité qu'un message raisonnablement récent chiffré dans cette clé provient de l'autre partie.

Obtenir les tickets du serveur Kerberos

Rappelez-vous qu'un ticket ne convient qu'à un seul serveur. Il est donc nécessaire d'obtenir un ticket distinct pour chaque service que le client souhaite utiliser. Les billets pour des serveurs individuels peuvent être obtenus auprès du service de délivrance des billets. Puisque le service d'octroi de billets est lui-même un service, il utilise le protocole d'accès au service décrit dans la section précédente.

Lorsqu'un programme nécessite un ticket qui n'a pas encore été demandé, il envoie une requête au serveur d'octroi de tickets. La demande contient le nom du serveur pour lequel un ticket est demandé, ainsi que le ticket d'octroi et un authentificateur construit comme décrit dans la section précédente.

Le serveur d'octroi de tickets vérifie ensuite l'authentificateur et le ticket d'octroi comme décrit ci-dessus. S'il est valide, le serveur d'octroi de tickets génère une nouvelle clé de session aléatoire à utiliser entre le client et le nouveau serveur. Il crée ensuite un ticket pour le nouveau serveur contenant le nom du client, le nom du serveur, l'heure actuelle, l'adresse IP du client et la nouvelle clé de session qu'il vient de générer. La durée de vie du nouveau ticket correspond au minimum de la durée de vie restante pour le ticket d'octroi et à la valeur par défaut du service.

Le serveur d'octroi de tickets renvoie ensuite le ticket, ainsi que la clé de session et d'autres informations, au client. Cette fois, cependant, la réponse est chiffrée dans la clé de session qui faisait partie du ticket d'octroi de ticket. De cette façon, il n'est pas nécessaire que l'utilisateur entre à nouveau son mot de passe.

Base de données Kerberos

Jusqu'à présent, nous avons discuté des opérations nécessitant un accès en lecture seule à la base de données Kerberos. Ces opérations sont effectuées par le service d'authentification, qui peut s'exécuter sur les machines maître et esclave.

Dans cette section, nous aborderons les opérations nécessitant un accès en écriture à la base de données. Ces opérations sont effectuées par le service d'administration, appelé KDBM (Kerberos Database Management Service). La mise en oeuvre actuelle stipule que les modifications ne peuvent être apportées qu'à la base de données Kerberos maître ; les copies esclaves sont en lecture seule. Par conséquent, le serveur KDBM ne peut être exécuté que sur la machine Kerberos maître.

Notez que, bien que l'authentification puisse toujours avoir lieu (sur les esclaves), les demandes d'administration ne peuvent pas être traitées si la machine maître est en panne. D'après notre expérience, cela n'a pas posé de problème, car les demandes d'administration sont rares.

Le KDBM gère les demandes des utilisateurs de modifier leurs mots de passe. Le côté client de ce programme, qui envoie des requêtes au KDBM via le réseau, est le programme kpasswd. Le KDBM accepte également les requêtes des administrateurs Kerberos, qui peuvent ajouter des identifiants à la base de données, ainsi que modifier les mots de passe des identifiants existants. Le côté client du programme d'administration, qui envoie également des requêtes au KDBM via le réseau, est le programme kadmin.

Serveur KDBM

Le serveur KDBM accepte les demandes d'ajout de identifiants à la base de données ou de modification des mots de passe des identifiants existants. Ce service est unique en ce sens que le service de délivrance de billets n'émettra pas de billets pour lui. Au lieu de cela, le service d'authentification lui-même doit être utilisé (le même service qui est utilisé pour obtenir un ticket d'octroi de tickets). L'objectif est d'exiger que l'utilisateur entre un mot de passe. Si ce n'était pas le cas, alors si un utilisateur laissait son poste de travail sans surveillance, un passant pourrait se lever et changer son mot de passe pour eux, ce qui devrait être évité. De même, si un administrateur laisse son poste de travail sans surveillance, un passant peut changer n'importe quel mot de passe dans le système.

Lorsque le serveur KDBM reçoit une demande, il l'autorise en comparant le nom principal authentifié du demandeur de la modification au nom principal de la cible de la demande. Si elles sont identiques, la demande est autorisée. S'ils ne sont pas identiques, le serveur KDBM consulte une liste de contrôle d'accès (stockée dans un fichier du système Kerberos maître). Si le nom principal du demandeur est trouvé dans ce fichier, la demande est autorisée, sinon elle est refusée.

Par convention, les noms avec une instance NULL (l'instance par défaut) n'apparaissent pas dans le fichier de liste de contrôle d'accès ; à la place, une instance admin est utilisée. Par conséquent, pour qu'un utilisateur devienne administrateur de Kerberos, une instance d'administrateur pour ce nom d'utilisateur doit être créée et ajoutée à la liste de contrôle d'accès. Cette convention permet à un administrateur d'utiliser un mot de passe différent pour l'administration Kerberos, qu'il ou elle utiliserait pour se connecter normalement.

Toutes les demandes adressées au programme KDBM, qu'elles soient autorisées ou refusées, sont consignées.

Programmes kadmin et kpasswd

Les administrateurs de Kerberos utilisent le programme kadmin pour ajouter des identités à la base de données ou modifier les mots de passe des identités existantes. Un administrateur doit entrer le mot de passe de son nom d'instance admin lorsqu'il appelle le programme kadmin. Ce mot de passe est utilisé pour récupérer un ticket pour le serveur KDBM.

Les utilisateurs peuvent modifier leurs mots de passe Kerberos à l'aide du programme kpasswd. Ils doivent entrer leur ancien mot de passe lorsqu'ils appellent le programme. Ce mot de passe est utilisé pour récupérer un ticket pour le serveur KDBM.

Réplication de base de données Kerberos

Chaque domaine Kerberos dispose d'une machine Kerberos maître, qui héberge la copie maître de la base de données d'authentification. Il est possible (mais pas nécessaire) d'avoir des copies supplémentaires en lecture seule de la base de données sur les machines esclaves ailleurs dans le système. Les avantages d'avoir plusieurs copies de la base de données sont généralement cités pour la réplication : une disponibilité accrue et de meilleures performances. Si la machine maître est en panne, l'authentification peut toujours être réalisée sur l'une des machines esclaves. La possibilité d'effectuer l'authentification sur n'importe quelle machine réduit la probabilité d'un goulot d'étranglement sur la machine principale.

La conservation de plusieurs copies de la base de données pose un problème de cohérence des données. Nous avons constaté que des méthodes très simples suffisent pour remédier aux incohérences. La base de données principale est vidée toutes les heures. La base de données est envoyée, dans son intégralité, aux machines esclaves, qui mettent ensuite à jour leurs propres bases de données. Un programme sur l'hôte maître, appelé kprop, envoie la mise à jour à un programme homologue, appelé kpropd, exécuté sur chacune des machines esclaves. D'abord, kprop envoie une somme de contrôle de la nouvelle base de données qu'il est sur le point d'envoyer. La somme de contrôle est chiffrée dans la clé de base de données principale de Kerberos, que les machines Kerberos maître et esclave possèdent. Les données sont ensuite transférées sur le réseau vers le kpropd sur la machine esclave. Le serveur de propagation d'esclaves calcule une somme de contrôle des données qu'il a reçues et, s'il correspond à la somme de contrôle envoyée par le maître, les nouvelles informations sont utilisées pour mettre à jour la base de données de l'esclave.

Tous les mots de passe de la base de données Kerberos sont chiffrés dans la clé de base de données principale. Par conséquent, les informations transmises de maître à esclave sur le réseau ne sont pas utiles à un écouteur. Cependant, il est essentiel que seules les informations de l'hôte maître soient acceptées par les esclaves, et que la falsification des données soit détectée, donc la somme de contrôle.

Kerberos de l'extérieur en regardant vers l'intérieur

Cette section décrit Kerberos du point de vue pratique, d'abord comme vu par l'utilisateur, puis du point de vue du programmeur d'applications, et enfin, à travers les tâches de l'administrateur Kerberos.

Vue de l'oeil de l'utilisateur Kerberos

Si tout va bien, l'utilisateur ne remarquera pas que Kerberos est présent. Dans notre implémentation UNIX, le ticket d'octroi de tickets est obtenu à partir de Kerberos dans le cadre du processus de connexion. La modification du mot de passe Kerberos d'un utilisateur fait partie du

programme passwd. Et les tickets Kerberos sont automatiquement détruits lorsqu'un utilisateur se déconnecte.

Si la session d'ouverture de session de l'utilisateur dure plus longtemps que la durée de vie du ticket d'octroi (actuellement 8 heures), l'utilisateur remarquera la présence de Kerberos car la prochaine fois qu'une application authentifiée par Kerberos sera exécutée, elle échouera. Le ticket Kerberos a expiré. À ce stade, l'utilisateur peut exécuter le programme kinit pour obtenir un nouveau ticket pour le serveur d'octroi de tickets. Comme lors de la connexion, un mot de passe doit être fourni pour l'obtenir. Un utilisateur qui exécute la commande klist par curiosité peut être surpris par tous les tickets qui ont été obtenus en son nom et en silence pour des services nécessitant une authentification Kerberos.

Kerberos du point de vue du programmeur

Un programmeur qui écrit une application Kerberos ajoute souvent l'authentification à une application réseau existante composée d'un côté client et d'un côté serveur. Nous appelons ce processus « Kerberisation » un programme. La Kerberisation implique généralement d'appeler la bibliothèque Kerberos afin d'effectuer l'authentification à la demande initiale de service. Il peut également impliquer des appels à la bibliothèque DES pour chiffrer les messages et les données qui sont ensuite envoyés entre le client d'application et le serveur d'applications.

Les fonctions de bibliothèque les plus couramment utilisées sont krb_mk_req côté client et krb_rd_req côté serveur. La routine krb_mk_req prend comme paramètres le nom, l'instance et le domaine du serveur cible, qui sera demandé, et éventuellement une somme de contrôle des données à envoyer. Le client envoie ensuite le message retourné par l'appel krb_mk_req sur le réseau au côté serveur de l'application. Lorsque le serveur reçoit ce message, il passe un appel à la routine de bibliothèque krb_rd_req. La routine renvoie un jugement sur l'authenticité de l'identité présumée de l'expéditeur.

Si l'application exige que les messages envoyés entre le client et le serveur soient secrets, les appels de bibliothèque peuvent être passés à krb_mk_priv (krb_rd_priv) pour chiffrer (décrypter) les messages dans la clé de session que les deux parties partagent maintenant.

Tâche de l'administrateur Kerberos

La tâche de l'administrateur Kerberos commence par l'exécution d'un programme pour initialiser la base de données. Un autre programme doit être exécuté pour enregistrer les principaux essentiels dans la base de données, par exemple le nom de l'administrateur Kerberos avec une instance d'administrateur. Le serveur d'authentification Kerberos et le serveur d'administration doivent être démarrés. S'il existe des bases de données esclaves, l'administrateur doit faire en sorte que les programmes de propagation des mises à jour de base de données du maître vers les esclaves soient lancés périodiquement.

Une fois ces étapes initiales effectuées, l'administrateur manipule la base de données sur le réseau, à l'aide du programme kadmin. Grâce à ce programme, de nouveaux identités peuvent être ajoutés et des mots de passe peuvent être modifiés.

En particulier, lorsqu'une nouvelle application Kerberos est ajoutée au système, l'administrateur Kerberos doit prendre quelques mesures pour la faire fonctionner. Le serveur doit être enregistré dans la base de données et se voir attribuer une clé privée (généralement une clé aléatoire générée automatiquement). Ensuite, certaines données (y compris la clé du serveur) doivent être extraites de la base de données et installées dans un fichier sur la machine du serveur. Le fichier

par défaut est /etc/srvtab. La routine de bibliothèque krb_rd_req appelée par le serveur (voir la section précédente) utilise les informations de ce fichier pour déchiffrer les messages envoyés chiffrés dans la clé privée du serveur. Le fichier /etc/srvtab authentifie le serveur en tant que mot de passe tapé sur un terminal authentifie l'utilisateur.

L'administrateur Kerberos doit également s'assurer que les machines Kerberos sont physiquement sécurisées, et il serait également judicieux de conserver les sauvegardes de la base de données Master.

La plus grande image de Kerberos

Dans cette section, nous décrivons comment Kerberos s'intègre dans l'environnement Athena, y compris son utilisation par d'autres services et applications réseau, et comment il interagit avec les royaumes Kerberos distants. Pour une description plus complète de l'environnement Athéna, veuillez consulter G.W. Des arbres.

Utilisation de Kerberos par d'autres services réseau

Plusieurs applications réseau ont été modifiées pour utiliser Kerberos. Les commandes rlogin et rsh tentent d'abord de s'authentifier à l'aide de Kerberos. Un utilisateur disposant de tickets Kerberos valides peut se reconnecter à une autre machine Athena sans avoir à configurer des fichiers .rhosts. Si l'authentification Kerberos échoue, les programmes retombent sur leurs méthodes habituelles d'autorisation, dans ce cas, les fichiers .rhosts.

Nous avons modifié le protocole de poste pour utiliser Kerberos pour authentifier les utilisateurs qui souhaitent récupérer leur courrier électronique à partir du bureau de poste. Un programme de remise de messages, appelé Zephyr, a récemment été développé à Athena, et il utilise également Kerberos pour l'authentification.

Le programme d'inscription de nouveaux utilisateurs, appelé registre, utilise à la fois le système de gestion des services (SMS) et Kerberos. À partir de SMS, il détermine si les informations saisies par le nouvel utilisateur Athena potentiel, telles que le nom et le numéro d'identification du MIT, sont valides. Il vérifie ensuite avec Kerberos si le nom d'utilisateur demandé est unique. Si tout va bien, une nouvelle entrée est faite à la base de données Kerberos, contenant le nom d'utilisateur et le mot de passe.

Pour une discussion détaillée sur l'utilisation de Kerberos pour sécuriser le système de fichiers réseau de Sun, reportez-vous à l'annexe.

Interaction avec d'autres Kerberi

Il est prévu que différentes organisations administratives voudront utiliser Kerberos pour l'authentification des utilisateurs. Dans de nombreux cas, les utilisateurs d'une organisation voudront utiliser les services d'une autre organisation. Kerberos prend en charge plusieurs domaines administratifs. La spécification des noms dans Kerberos inclut un champ appelé domaine. Ce champ contient le nom du domaine administratif dans lequel l'utilisateur doit être authentifié.

Les services sont généralement enregistrés dans un domaine unique et n'acceptent que les informations d'identification émises par un serveur d'authentification pour ce domaine. Un utilisateur est généralement enregistré dans un seul domaine (le domaine local), mais il est

possible pour lui d'obtenir des informations d'identification émises par un autre domaine (le domaine distant), en utilisant l'authentification fournie par le domaine local. Les informations d'identification valides dans un domaine distant indiquent le domaine dans lequel l'utilisateur a été authentifié à l'origine. Les services du domaine distant peuvent choisir d'honorer ces informations d'identification, en fonction du niveau de sécurité requis et du niveau de confiance dans le domaine qui a initialement authentifié l'utilisateur.

Pour effectuer l'authentification entre domaines, il est nécessaire que les administrateurs de chaque paire de domaines sélectionnent une clé à partager entre leurs domaines. Un utilisateur du domaine local peut alors demander un ticket d'octroi de tickets au serveur d'authentification local pour le serveur d'octroi de tickets dans le domaine distant. Lorsque ce ticket est utilisé, le serveur d'octroi de tickets distant reconnaît que la demande ne provient pas de son propre domaine et utilise la clé précédemment échangée pour déchiffrer le ticket d'octroi de tickets. Il émet ensuite un ticket comme il le ferait normalement, sauf que le champ realm du client contient le nom du domaine dans lequel le client a été authentifié à l'origine.

Cette approche pourrait être étendue pour permettre de s'authentifier à travers une série de royaumes jusqu'à atteindre le royaume avec le service souhaité. Pour ce faire, cependant, il serait nécessaire d'enregistrer l'intégralité du chemin qui a été emprunté, et pas seulement le nom du domaine initial dans lequel l'utilisateur a été authentifié. Dans une telle situation, tout ce que connaît le serveur est que A dit que B dit que C dit que l'utilisateur est oui et non. Cette instruction ne peut être approuvée que si toutes les personnes sur le chemin sont également approuvées.

Problèmes Kerberos et problèmes ouverts

Il existe un certain nombre de problèmes et de problèmes ouverts associés au mécanisme d'authentification Kerberos. Parmi les problèmes, citons comment décider de la durée de vie correcte d'un ticket, comment autoriser les proxys et comment garantir l'intégrité de la station de travail.

Le problème de la durée de vie des billets consiste à choisir le compromis approprié entre la sécurité et la commodité. Si la durée de vie d'un ticket est longue, alors si un ticket et sa clé de session associée sont volés ou déplacés, ils peuvent être utilisés pendant une plus longue période. Ces informations peuvent être volées si un utilisateur oublie de se déconnecter d'une station de travail publique. Sinon, si un utilisateur a été authentifié sur un système qui autorise plusieurs utilisateurs, un autre utilisateur ayant accès à la racine peut trouver les informations nécessaires pour utiliser des tickets volés. Cependant, le problème avec l'attribution d'un ticket à court terme est que lorsqu'il expire, l'utilisateur devra en obtenir un nouveau qui exige que l'utilisateur entre à nouveau le mot de passe.

Un problème ouvert est le problème du proxy. Comment un utilisateur authentifié peut-il autoriser un serveur à acquérir d'autres services réseau en son nom ? L'utilisation d'un service permettant d'accéder directement à des fichiers protégés à partir d'un serveur de fichiers est un exemple d'importance. Un autre exemple de ce problème est ce que nous appelons le transfert d'authentification. Si un utilisateur est connecté à une station de travail et se connecte à un hôte distant, il serait bon que l'utilisateur ait accès aux mêmes services disponibles localement, tout en exécutant un programme sur l'hôte distant. Ce qui rend cela difficile, c'est que l'utilisateur peut ne pas faire confiance à l'hôte distant, de sorte que le transfert d'authentification n'est pas souhaitable dans tous les cas. Nous n'avons pas encore de solution à ce problème.

Un autre problème, important dans l'environnement Athena, est de savoir comment garantir l'intégrité du logiciel exécuté sur une station de travail. Ce n'est pas tant un problème sur les

stations de travail privées, car l'utilisateur qui l'utilisera en a le contrôle. Sur les postes de travail publics, cependant, quelqu'un peut être venu et avoir modifié le programme de connexion pour enregistrer le mot de passe de l'utilisateur. La seule solution actuellement disponible dans notre environnement est de rendre difficile la modification des logiciels exécutés sur les stations de travail publiques. Une meilleure solution exigerait que la clé de l'utilisateur ne quitte jamais un système dont l'utilisateur sait qu'il est fiable. Cela pourrait être possible si l'utilisateur possédait une carte à puce capable d'effectuer le chiffrement requis dans le protocole d'authentification.

État Kerberos

Une version prototype de Kerberos est entrée en production en septembre 1986. Depuis janvier 1987, Kerberos est le seul moyen de Project Athena d'authentifier ses 5 000 utilisateurs, 650 stations de travail et 65 serveurs. De plus, Kerberos est maintenant utilisé à la place des fichiers .rhosts pour contrôler l'accès dans plusieurs systèmes de partage de temps d'Athena.

Remerciements Kerberos

Kerberos a été initialement conçu par Steve Miller et Clifford Neuman avec des suggestions de Jeff Schiller et Jerry Saltzer. Depuis, de nombreuses autres personnes ont participé au projet. Parmi eux : Jim Aspnes, Bob Baldwin, John Barba, Richard Basch, Jim Bloom, Bill Bryant, Mark Colan, Rob French, Dan Geer, John Kohl, John Kubiatowicz, Bob Mckie, Brian Murphy, John Ostlund Ken Raeburn, Chris Reed, Jon Rochlis, Mike Shanzer, Bill Sommerfeld, Ted T'so, Win Treese et Stan Zanarotti.

Nous sommes reconnaissants à Dan Geer, Kathy Lieben, Josh Lubarr, Ken Raeburn, Jerry Saltzer, Ed Steiner, Robbert van Renesse et Win Treese, dont les suggestions ont grandement amélioré les versions antérieures de ce document.

Jedlinsky, J.T. Kohl et W.E. Sommerfeld, " The Zephyr Notification System « , dans Usenix Conference Proceedings (hiver 1988).

M.A. Rosenstein, D.E. Geer et P.J. Levine, dans Usenix Conference Proceedings (hiver 1988).

R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh et B. Lyon, « Design and Implementation of the Sun Network Filesystem », Usenix Conference Proceedings (été 1985).

Annexe : Application Kerberos au système de fichiers réseau (NFS) de SUN

Un élément clé du système de station de travail Project Athena est l'interconnexion du réseau entre la station de travail de l'utilisateur et son stockage de fichiers privés (répertoire d'origine). Tout le stockage privé réside sur un ensemble d'ordinateurs (actuellement des systèmes VAX 11/750) dédiés à cette fin. Cela nous permet d'offrir des services sur des stations de travail UNIX accessibles au public. Lorsqu'un utilisateur se connecte à l'une de ces stations de travail accessibles au public, plutôt que de valider son nom et son mot de passe par rapport à un fichier de mot de passe local, nous utilisons Kerberos pour déterminer son authenticité. Le programme de connexion demande un nom d'utilisateur (comme sur tout système UNIX). Ce nom d'utilisateur est utilisé pour récupérer un ticket Kerberos. Le programme de connexion utilise le mot de passe pour générer une clé DES pour déchiffrer le ticket. Si le déchiffrement est réussi, le répertoire

d'origine de l'utilisateur est situé en consultant le service d'attribution de noms Hesiod et monté via NFS. Le programme de connexion transfère ensuite le contrôle sur le shell de l'utilisateur, qui peut ensuite exécuter les fichiers de personnalisation traditionnels par utilisateur car le répertoire d'accueil est maintenant « attaché » à la station de travail. Le service Hesiod est également utilisé pour construire une entrée dans le fichier de mot de passe local. (Ceci est pour les programmes qui recherchent des informations dans /etc/passwd.)

Àpartir de plusieurs options pour la fourniture du service de fichiers à distance, nous avons choisi le système de fichiers réseau de Sun. Cependant, ce système ne répond pas à nos besoins d'une manière cruciale. NFS suppose que toutes les stations de travail se divisent en deux catégories (comme vue du serveur de fichiers) : approuvé et non approuvé. Les systèmes non fiables ne peuvent pas du tout accéder à des fichiers, les systèmes fiables le peuvent. Les systèmes fiables sont totalement fiables. On suppose qu'un système fiable est géré par une gestion conviviale. Plus précisément, il est possible à partir d'une station de travail de confiance de se déguiser en n'importe quel utilisateur valide du système de service de fichiers et donc d'accéder à presque tous les fichiers du système. (Seuls les fichiers appartenant à « root » sont exemptés.)

Dans notre environnement, la gestion d'une station de travail (dans le sens traditionnel de la gestion du système UNIX) est entre les mains de l'utilisateur qui l'utilise actuellement. Nous ne cachons pas le mot de passe root sur nos stations de travail, car nous réalisons qu'un utilisateur vraiment peu amical peut se déconnecter par le fait même qu'il est assis au même endroit physique que la machine et a accès à toutes les fonctions de console. Par conséquent, nous ne pouvons pas vraiment faire confiance à nos stations de travail dans l'interprétation NFS de la confiance. Pour permettre des contrôles d'accès appropriés dans notre environnement, nous avons dû apporter quelques modifications au logiciel NFS de base, et intégrer Kerberos dans le schéma.

NFS non modifié Kerberos

Dans la mise en oeuvre de NFS avec laquelle nous avons commencé (de l'Université du Wisconsin), l'authentification a été fournie sous la forme d'un morceau de données inclus dans chaque requête NFS (appelé « informations d'identification » dans la terminologie NFS). Cette information d'identification contient des informations sur l'identifiant utilisateur unique (UID) du demandeur et une liste des identificateurs de groupe (GID) de l'appartenance du demandeur. Ces informations sont ensuite utilisées par le serveur NFS pour le contrôle d'accès. La différence entre une station de travail approuvée et une station de travail non approuvée est de savoir si ses informations d'identification sont acceptées ou non par le serveur NFS.

NFS modifié Kerberos

Dans notre environnement, les serveurs NFS doivent accepter les informations d'identification d'une station de travail si et seulement si les informations d'identification indiquent l'UID de l'utilisateur de la station de travail, et aucun autre.

Une solution évidente consisterait à changer la nature des informations d'identification, passant de simples indicateurs d'UID et de GID à des données authentifiées Kerberos entièrement détruites. Toutefois, une pénalité importante serait versée si cette solution était adoptée. Les informations d'identification sont échangées sur chaque opération NFS, y compris toutes les activités de lecture et d'écriture sur disque. L'inclusion d'une authentification Kerberos sur chaque transaction de disque ajouterait un nombre suffisant de codages complets (effectués dans le logiciel) par transaction et, selon nos calculs d'enveloppe, aurait produit des performances inacceptables. (Il

aurait également fallu placer les routines de bibliothèque Kerberos dans l'espace d'adressage du noyau.)

Nous avions besoin d'une approche hybride, décrite ci-dessous. L'idée de base est de faire recevoir les informations d'identification du mappage du serveur NFS des stations de travail clientes, à une information d'identification valide (et peut-être différente) sur le système serveur. Ce mappage est effectué dans le noyau du serveur sur chaque transaction NFS et est configuré au moment du « montage » par un processus au niveau de l'utilisateur qui s'engage dans l'authentification modérée Kerberos avant d'établir un mappage d'informations d'identification du noyau valide.

Pour implémenter cela, nous avons ajouté un nouvel appel système au noyau (requis uniquement sur les systèmes serveur, et non sur les systèmes client) qui permet le contrôle de la fonction de mappage qui mappe les informations d'identification entrantes des stations de travail clientes aux informations d'identification valides pour utilisation sur le serveur (le cas échéant). La fonction de mappage de base mappe le tuple :

```
<CLIENT-IP-ADDRESS, UID-ON-CLIENT>
```

àune identification NFS valide sur le système serveur. L'adresse IP-CLIENT est extraite du paquet de requête NFS fourni par le système client. Note: toutes les informations contenues dans les informations d'identification générées par le client, à l'exception de l'UID-ON-CLIENT, sont ignorées.

Si aucun mappage n'existe, le serveur réagit de deux manières, selon sa configuration. Dans notre configuration conviviale, nous utilisons par défaut les requêtes inmappables dans les informations d'identification de l'utilisateur « personne » qui n'a pas d'accès privilégié et qui a un UID unique. Les serveurs non conviviaux retournent une erreur d'accès NFS lorsqu'aucun mappage valide n'est trouvé pour les informations d'identification NFS entrantes.

Notre nouvel appel système est utilisé pour ajouter et supprimer des entrées de la carte de résident du noyau. Il permet également de vider toutes les entrées qui correspondent à un UID spécifique sur le système serveur, ou de vider toutes les entrées d'une ADRESSE-IP-CLIENT donnée.

Nous avons modifié le démon de montage (qui gère les demandes de montage NFS sur les systèmes serveur) pour accepter un nouveau type de transaction, la demande de mappage d'authentification Kerberos. En principe, dans le cadre du processus de montage, le système client fournit un authentificateur Kerberos ainsi qu'une indication de son UID-ON-CLIENT (crypté dans l'authentificateur Kerberos) sur la station de travail. Le démon de montage du serveur convertit le nom principal Kerberos en nom d'utilisateur local. Ce nom d'utilisateur est ensuite recherché dans un fichier spécial pour obtenir l'UID et la liste des GID de l'utilisateur. Pour plus d'efficacité, ce fichier est un fichier de base de données ndbm avec le nom d'utilisateur comme clé. À partir de ces informations, un identifiant NFS est construit et remis au noyau comme mappage valide du tuple <CLIENT-IP-ADDRESS, CLIENT-UID> pour cette demande.

Au moment du démontage, une requête est envoyée au démon de montage pour supprimer le mappage précédemment ajouté du noyau. Il est également possible d'envoyer une demande au moment de la déconnexion pour invalider tous les mappages pour l'utilisateur actuel sur le serveur en question, nettoyant ainsi les mappages restants (mais ils ne devraient pas) avant que la station de travail ne soit mise à disposition pour l'utilisateur suivant.

Implications de sécurité Kerberos du NFS modifié

Cette mise en oeuvre n'est pas totalement sécurisée. Tout d'abord, les données utilisateur sont toujours envoyées sur le réseau sous une forme non cryptée et donc interceptable. L'authentification de bas niveau par transaction est basée sur une paire <CLIENT-IP-ADDRESS, CLIENT-UID> fournie non chiffrée dans le paquet de requête. Ces informations pourraient être falsifiées et donc compromises sur la sécurité. Toutefois, il est à noter que seuls les mappages valides sont en place lorsqu'un utilisateur utilise activement ses fichiers (c'est-à-dire lorsqu'il est connecté) et que cette forme d'attaque est donc limitée à l'utilisateur connecté. Lorsqu'un utilisateur n'est pas connecté, aucune falsification d'adresse IP ne permet un accès non autorisé à ses fichiers.

Références Kerberos

1. S.P. Miller (C.-B.) Neuman, J.I. Schiller et J.H. Sel, section E.2.1 : Kerberos Authentication and Authorization System, M.I.T. Projet Athena, Cambridge, Massachusetts (21 décembre 1987).
2. e. Balkovich, S.R. Lerman et R.P. Parmelee, « Computing in Higher Education : The Athena Experience, " Communications of the ACM, vol. 28(11), p. 1214 à 1224, ACM (novembre 1985).
3. R.M. Needham et M.D. Schröder, « Using Encryption for Authentication in Large Networks of Computers », Communications of the ACM, vol. 21(12), p. 993-999 (décembre 1978).
4. V.L. Voydock et S.T. Kent, Security Mechanisms in High-Level Network Protocols, Computing Surveys, vol. 15(2), ACM (juin 1983).
5. National Bureau of Standards, « Data Encryption Standard », Federal Information Processing Standards Publication 46, Government Printing Office, Washington, DC (1977).
6. SP Dyer, Hesiod, in Usenix Conference Proceedings (Hiver, 1988).
7. W.J. Bryant, Kerberos Programmer's Tutorial, MIT Project Athena (En préparation).
8. W.J. Bryant, Kerberos Administrator's Manual, MIT Project Athena (En préparation).
9. G.W. Treese, Berkeley Unix on 1000 Workstations : Athena Changes to 4.3BSD, " dans Usenix Conference Proceedings (hiver 1988).
10. C.A. DellaFera, M.W. Eichin, R.S. French, D.C. Jedlinsky, J.T. Kohl et W.E. Sommerfeld, " The Zephyr Notification System « , dans Usenix Conference Proceedings (hiver 1988).
11. M.A. Rosenstein, D.E. Geer et P.J. Levine, dans Usenix Conference Proceedings (hiver 1988).
12. R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh et B. Lyon, « Design and Implementation of the Sun Network Filesystem », Usenix Conference Proceedings (été 1985).

Informations connexes

- Page d'assistance de Kerberos
- Support et documentation techniques - Cisco Systems