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 de Kerberos

- Jennifer G. Steiner, projet Athéna, Massachusetts Institute of Technology, Cambridge, mA 02139, steiner@ATHENA.MIT.EDU
- Clifford Neuman, service de l'informatique, FR-35, université de Washington, Seattle, WA 98195, bcn@CS.WASHINGTON.EDU. Le Clifford Neuman était un membre de l'équipe du projet Athena pendant la conception et la phase d'implémentation initiale du Kerberos.
- Jeffrey I. Schiller, projet Athéna, Massachusetts Institute of Technology, Cambridge, mA 02139, jis@ATHENA.MIT.EDU

Introduction au Kerberos

Ce document donne un aperçu de Kerberos, un système d'authentification conçu par Miller et Neuman. pour des environnements d'informatique à réseau ouvert, et décrit notre expérience utilisant elle au projet Athéna du MIT. Dans la section sur la motivation, nous expliquons pourquoi un nouveau modèle d'authentification est nécessaire pour des réseaux ouverts, et ce que sont ses conditions requises. Ce qui est Kerberos ? la section répertorie les composants du Kerberos logiciel et décrit comment ils interagissent en fournissant le service d'authentification. Dans les noms Kerberos section, nous décrivons le Kerberos nommant le schéma.

Comment le Kerberos fonctionne des présents les modules de l'authentification Kerberos - le ticket et l'authentificateur. Ceci mène à un examen des deux Protocoles d'authentification : l'authentification initiale d'un utilisateur au Kerberos (analogue à ouvrir une session), et le protocole pour l'authentification mutuelle d'un consommateur potentiel et d'un producteur potentiel d'un service réseau.

Le Kerberos exige une base de données des informations sur ses clients ; la section de base de données Kerberos décrit la base de données, sa Gestion, et le protocole pour sa modification. Le Kerberos du regard extérieur dans la section décrit le Kerberos relie à ses utilisateurs, programmeurs d'applications, et administrateurs. Dans la section de plus grande image, nous décrivons comment les adaptations de projet Athena Kerberos dans le reste de l'environnement Athena. Nous décrivons également l'interaction de différents domaines d'authentification Kerberos, ou de royaumes ; dans notre cas, la relation entre le projet Athena Kerberos et le Kerberos s'exécutant au laboratoire du MIT pour de l'informatique.

Dans les problèmes et question non résolus section, nous mentionnons les questions ouvertes et les problèmes jusqu'à présent non résolus. La dernière section donne l'état actuel du Kerberos au projet Athéna. Dans l'annexe, nous décrivons en détail comment le Kerberos est appliqué à un d'archivage de réseau pour authentifier les utilisateurs qui souhaitent accéder aux systèmes de fichiers distants.

Concepts de Kerberos

Dans tout ce document nous utilisons les termes qui peuvent être ambigus, nouveau au lecteur, ou utilisé différemment ailleurs. Au-dessous de nous énonçons notre utilisation de ces termes.

Utilisateur, client, serveur ? Par l'utilisateur, nous voulons dire un être humain qui utilise un programme ou un service. Un client utilise également quelque chose, mais n'est pas nécessairement une personne ; ce peut être un programme. Souvent les applications réseau se composent de deux parts ; un un programme qui fonctionne sur un ordinateur et demande un service distant, et un programme différent qui des passages sur l'ordinateur distant et assure ce service. Nous appelons ceux le côté client et le côté serveur de l'application, respectivement. Souvent, un client contactera un serveur au nom d'un utilisateur.

Chaque entité qui utilise le système Kerberos, que ce soit un utilisateur ou un serveur de réseau, est dans un sens un client, puisqu'elle utilise le service de Kerberos. Distinguer ainsi des clients Kerberos des clients d'autre service, nous employons le principal de terme pour indiquer une telle entité. Notez qu'un principal Kerberos peut être un utilisateur ou un serveur. (Nous décrivons nommer des principal Kerberos dans une section postérieure.)

Entretenez contre le serveur ? Nous utilisons le service comme une spécification abstraite de quelques actions d'être exécuté. Un processus qui exécute ces actions s'appelle un serveur. À un moment donné, il peut y avoir plusieurs serveurs (s'exécutant habituellement sur différents ordinateurs) exécutant un service donné. Par exemple, chez Athéna il y a un serveur exécutant de rlogin BSD UNIX sur chacun de nos ordinateurs de temps partagé.

Clé, clé privée, mot de passe ? Chiffrement à clé privé d'utilisations de Kerberos. Chaque principal Kerberos est assigné un grand nombre, sa clé privée, connue seulement à celui principal et au Kerberos. Dans le cas d'un utilisateur, la clé privée est le résultat d'une fonction univoque appliquée au mot de passe d'utilisateur. Nous utilisons la clé comme sténographie pour la clé privée.

Qualifications ? Malheureusement, ce mot a une signification particulière pour le système de fichiers en réseau de Sun et le système Kerberos. Nous énonçons explicitement si nous voulons dire des qualifications NFS ou des qualifications de Kerberos, autrement le terme est utilisé dans le sens d'anglais normal.

Maître et esclave ? Il est possible d'exécuter le logiciel d'authentification Kerberos sur plus d'un ordinateur. Cependant, il y a toujours seulement une copie définitive de la base de données Kerberos. L'ordinateur qui loge cette base de données s'appelle l'ordinateur maître, ou juste le maître. D'autres ordinateurs peuvent posséder des copies en lecture seule de la base de données Kerberos, et ceux-ci s'appellent les esclaves.

Motivation derrière le Kerberos

Dans un environnement non-en réseau d'informatique personnelle, des ressources et les informations peuvent être protégées en sécurisant physiquement le PC. Dans un environnement informatique de temps partagé, le système d'exploitation protège des utilisateurs les uns des autres et contrôle des ressources. Afin de déterminer ce que chaque utilisateur peut lire ou modifier, il est que le système à temps partagé identifie chaque utilisateur. Ce fait quand l'utilisateur ouvre une session.

Dans un réseau des utilisateurs ayant besoin des services à partir de beaucoup d'ordinateurs distincts, il y a trois approches une peut prendre au contrôle d'accès : On peut ne faire rien, comptant sur l'ordinateur auquel l'utilisateur est ouvert une session pour empêcher l'accès non autorisé ; on peut exiger de l'hôte de prouver son identité, mais fait confiance au mot de l'hôte quant à qui l'utilisateur est ; ou on peut exiger de l'utilisateur de prouver le son identité pour chaque service exigé.

Dans un environnement fermé où tous les ordinateurs sont sous le contrôle strict, on peut utiliser la première approche. Quand les contrôles d'organisation tous les hôtes communiquant du réseau, ceci est une approche raisonnable.

Dans plus d'environnement ouvert, on pourrait sélectivement faire confiance seulement à ces hôtes sous le contrôle organisationnel. Dans ce cas, chaque hôte doit être exigé pour prouver son identité. Les programmes de rlogin et de rsh utilisent cette approche. Dans ces protocoles, l'authentification est faite en vérifiant l'adresse Internet dont une connexion a été établie.

Dans l'environnement Athena, nous devons pouvoir honorer des demandes des hôtes qui ne sont pas sous le contrôle organisationnel. Les utilisateurs ont le contrôle complet de leurs postes de travail : ils peuvent les redémarrer, les amener autonome, ou même démarrer outre de leurs propres moyens des bandes. En soi, la troisième approche doit être adoptée ; l'utilisateur doit prouver le son identité pour chaque service désiré. Le serveur doit également prouver son identité. Il n'est pas suffisant de sécuriser physiquement l'hôte exécutant un serveur de réseau ; quelqu'un ailleurs sur le réseau peut déguiser en tant que serveur donné.

Notre environnement place des plusieurs conditions sur un mécanisme d'identification. D'abord, il doit être sécurisé. La mise en échec de lui doit être assez difficile qu'un attaquant potentiel ne trouve pas le mécanisme d'authentification pour être le maillon faible. Quelqu'un qui observe le réseau ne devrait pas pouvoir obtenir les informations nécessaires pour personnifier un autre utilisateur. En second lieu, il doit être fiable. Access à beaucoup de services dépendra du service d'authentification. S'il n'est pas fiable, le système des services dans son ensemble ne sera pas. Troisièmement, il devrait être transparent. Dans le meilleur des cas, l'utilisateur ne devrait pas se rendre compte de l'authentification ayant lieu. En conclusion, il devrait être extensible. Beaucoup de systèmes peuvent communiquer avec des hôtes d'Athéna. Pas toute la ces derniers prendra en charge notre mécanisme, mais le logiciel ne devrait pas se casser si elles faisaient.

Le Kerberos est le résultat de notre travail pour répondre aux exigences ci-dessus. Quand un utilisateur marche à un poste de travail ils ouvrent une session. Dans la mesure où l'utilisateur peut dire, cette première identification est suffisante pour prouver leur identité à tous les serveurs de réseau exigés pour la durée de la session d'ouverture de connexion. La Sécurité du Kerberos se fonde sur les sécurités de plusieurs serveurs d'authentification, mais pas sur le système duquel les utilisateurs ouvrent une session, ni sur le degré de sécurité des serveurs d'extrémité qui seront utilisés. Le serveur d'authentification fournit à un utilisateur correctement authentifié une manière de prouver le son identité aux serveurs dispersés à travers le réseau.

L'authentification est un élément constitutif essentiel pour un environnement en réseau sécurisé. Si, par exemple, un serveur connaît pour certain l'identité d'un client, elle peut décider si fournir le service, si l'utilisateur devrait être des privilèges spéciaux donnés, qui devraient recevoir la facture pour le service, et ainsi de suite. En d'autres termes, des schémas d'autorisation et de comptabilité peuvent être établis sur l'authentification que le Kerberos fournit, ayant pour résultat la Sécurité équivalente au PC solitaire ou au système à temps partagé.

Quel est Kerberos ?

Le Kerberos est un service en fonction de confiance d'authentification de tiers sur le modèle présenté par Needham et Schroeder. Il est de confiance dans le sens que chacun de ses clients pense le jugement du Kerberos quant à l'identité de chacun de ses autres clients pour être précis. Des horodateurs (grands nombres représentant la date et heure actuelles) ont été ajoutés au modèle d'origine pour faciliter la détection de la rediffusion. La rediffusion se produit quand un message est dérobé outre du réseau et plus tard renvoyé. Pour une description plus complète de rediffusion, et d'autres questions de l'authentification, voir Voydock et le Kent.

Que le Kerberos fait-il ?

Le Kerberos garde une base de données de ses clients et de leurs clés privées. La clé privée est un grand nombre connu seulement au Kerberos et au client qu'il appartient à. Dans le cas que le client est un utilisateur, c'est un mot de passe chiffré. Les services réseau ayant besoin de l'authentification s'inscrivent au Kerberos, de même que faites des clients souhaitant utiliser ces services. Les clés privées sont négociées à l'enregistrement.

Puisque le Kerberos connaît ces clés privées, il peut créer les messages qui convainquent un client qu'un autre est vraiment qui il prétend être. Le Kerberos génère également des clés privées provisoires, appelées les clés de session, qui sont données à deux clients et à personne d'autre. Une clé de session peut être utilisée pour chiffrer des messages entre deux interlocuteurs.

Le Kerberos fournit trois niveaux de protection distincts. Le programmeur d'application détermine ce qui est approprié, selon les conditions requises de l'application. Par exemple, quelques applications exigent seulement que l'authenticité soit établie à l'initiation d'une connexion réseau, et peuvent supposer que les messages suivants d'une adresse de réseau donnée proviennent de l'interlocuteur authentifié. Notre Network File System authentifié utilise ce niveau de sécurité.

D'autres applications exigent l'authentification de chaque message, mais ne s'inquiètent pas, que le contenu du message soit révélé ou pas. Pour ces derniers, le Kerberos fournit les messages sûrs. Pourtant un niveau supérieur de Sécurité est fourni par les messages privés, où chaque message est non seulement authentifié, mais également est chiffré. Des messages privés sont utilisés, par exemple, par le serveur de Kerberos lui-même pour envoyer des mots de passe au-dessus du réseau.

Composants logiciels de Kerberos

L'implémentation d'Athéna comporte plusieurs modules :

- Bibliothèque d'applications de Kerberos
- bibliothèque de chiffrement
- bibliothèque de la base de données
- programmes d'administration de base de données
- serveur de gestion
- serveur d'authentification
- logiciel de propagation DB
- programmes utilisateur
- applications

La bibliothèque d'applications de Kerberos fournit une interface pour des clients et des serveurs d'applications d'application. Il contient, notamment, des routines pour des demandes d'authentification de création ou de lecture, et les routines pour créer les messages sûrs ou privés.

Le cryptage dans le Kerberos est basé sur le DES, la norme de chiffrement de données. La bibliothèque de chiffrement implémente ces routines. Plusieurs méthodes de cryptage sont équipées, de compromis entre la vitesse et la Sécurité. Une extension au bloc de chiffrement DES enchaînant le mode (CBC), appelé le mode propageant CBC, est également fournie. Dans le CBC, une erreur est propagée seulement par le bloc en cours du chiffrement, tandis que dans PCBC, l'erreur est propagée dans tout le message. Ceci rend le message entier inutile si une erreur se produit, plutôt que juste une partie de lui. La bibliothèque de chiffrement est un module indépendant, et peut être remplacée par d'autres réalisations DES ou une bibliothèque de chiffrement différente.

Un autre module remplaçable est le système de gestion de bases de données. L'implémentation actuelle d'Athena de la bibliothèque de la base de données utilise le ndbm, bien qu'Ingres ait été initialement utilisé. D'autres bibliothèques de gestion de bases de données ont pu être aussi bien utilisées.

Les nécessités de la base de données Kerberos sont simples ; un record est détenu pour chaque principal, contenant le nom, la clé privée, et la date d'expiration du principal, avec quelques informations d'administration. (La date d'expiration est la date après quoi une entrée n'est plus valide. Il est habituellement placé à quelques années dans le futur à l'enregistrement.)

D'autres informations utilisateur, telles que le nom réel, numéro de téléphone, et ainsi de suite, sont gardées par un autre serveur, le nameserver de Hesiod. De cette façon, les informations confidentielles, à savoir des mots de passe, peut être manipulée par le Kerberos, utilisant assez la sécurité élevée mesure ; tandis que les informations non sensibles gardées par Hesiod sont traitées différemment ; il peut, par exemple, être envoyé à décrypté au-dessus du réseau.

Les serveurs de Kerberos utilisent la bibliothèque de la base de données, de même que font les outils pour gérer la base de données.

Le serveur de gestion (ou le serveur KDBM) fournit une interface réseau lecture/écriture à la base de données. Le côté client du programme peut être exécuté sur n'importe quel ordinateur sur le réseau. Le côté serveur, cependant, doit s'exécuter sur l'ordinateur logeant la base de données Kerberos afin d'apporter des modifications à la base de données.

Le serveur d'authentification (ou le serveur de Kerberos), d'autre part, exécute des opérations en lecture seule sur la base de données Kerberos, à savoir, l'authentification des principaux, et de la génération des clés de session. Puisque ce serveur ne modifie pas la base de données Kerberos, elle peut fonctionner sur un ordinateur logeant une copie en lecture seule de la base de données Kerberos principale.

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 différents ordinateurs, avec une copie du serveur exécutant d'authentification sur chaque ordinateur. Chacun de ces ordinateurs slaves reçoit une mise à jour de la base de données Kerberos de l'ordinateur maître aux intervalles donnés.

En conclusion, il y a des programmes d'utilisateur pour ouvrir une session au Kerberos, changer un mot de passe de Kerberos, et afficher ou la destruction des tickets Kerberos (des tickets sont expliqués plus tard).

Noms Kerberos

Une partie d'authentifier une entité la nomme. Le processus de l'authentification est la vérification que le client est celui nommé dans une demande. De queest-ce qu'un nom se compose ? Dans le Kerberos, des 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'un exemple, et d'un royaume, exprimé comme name.instance@realm.

Le nom principal est le nom d'utilisateur ou le service. L'exemple est utilisé pour distinguer parmi des variations sur le nom principal. Pour des utilisateurs, un exemple peut nécessiter des privilèges spéciaux, tels que les exemples de « racine » ou de « admin ». Pour des services dans l'environnement Athena, l'exemple est habituellement le nom de la machine sur lequel le serveur fonctionne. Par exemple, le service de rlogin a différents exemples sur des différents hôtes : rlogin.priam est le serveur de rlogin sur l'hôte nommé Priam. Un ticket Kerberos est seulement bon pour un serveur Désigné simple. En soi, un ticket séparé est exigé pour accéder à différents exemples du même service. Le royaume est le nom d'une entité administrative qui met à jour des données d'authentification. Par exemple, les différentes institutions peuvent chacune avoir leur propre ordinateur de Kerberos, logeant une base de données différente. Ils ont différents royaumes de Kerberos. (Des royaumes sont discutés plus loin dans Interactionwith l'autre Kerberi.)

Comment le Kerberos fonctionne

Cette section décrit les protocoles d'authentification Kerberos. Comme mentionné ci-dessus, le modèle d'authentification Kerberos est basé protocole sur de Needham et de Schroeder distribution de clé. Quand des demandes d'utilisateur un service, son identité doivent être établies. Pour faire ceci, un ticket est présenté au serveur, avec la preuve que le ticket a été initialement émis à l'utilisateur, pas dérobé. Il y a trois phases à l'authentification par le Kerberos. Pendant la première phase, l'utilisateur obtient des qualifications à utiliser pour demander l'accès à d'autres services. Pendant la deuxième étape, l'authentification de demandes d'utilisateur pour un service spécifique. Pendant la phase finale, l'utilisateur présente ces qualifications au serveur d'extrémité.

Qualifications de Kerberos

Il y a deux types de qualifications utilisées dans le modèle d'authentification Kerberos : tickets et authentificateurs. Chacun des deux sont basés sur le chiffrement à clé privé, mais elles sont chiffrées utilisant différentes clés. Un ticket est utilisé pour passer sécurisé l'identité de la personne à qui le ticket a été fourni entre le serveur d'authentification et le serveur d'extrémité. Un ticket passe également les informations qui peuvent être utilisées pour s'assurer que la personne utilisant le ticket est la même personne à laquelle il a été fourni. L'authentificateur contient les informations complémentaires qui, une fois comparées contre cela dans le ticket montrent que le client présent le ticket est le même auquel le ticket a été fourni.

Un ticket est bon pour un serveur unique et un client simple. Il contient le nom du serveur, le nom du client, l'adresse Internet du client, un horodateur, une vie, et une clé de session aléatoire. Ces informations sont chiffrées utilisant la clé du serveur pour lequel le ticket sera utilisé. Une fois le ticket a été émis, il peut être utilisé de plusieurs périodes par le client Désigné d'accéder au serveur Désigné, jusqu'à ce que le ticket expire. Notez que parce que le ticket est chiffré dans la clé du serveur, il est sûr de permettre à l'utilisateur de passer le ticket en fonction au serveur sans devoir s'inquiéter de l'utilisateur modifiant le ticket.

Àla différence du ticket, l'authentificateur peut seulement être utilisé une fois. Un neuf doit être généré chaque fois que un client veut utiliser un service. Ceci ne présente pas un problème parce

que le client peut établir l'authentificateur lui-même. Un authentificateur contient le nom du client, de l'adresse IP du poste de travail, et de l'heure actuelle du poste de travail. L'authentificateur est chiffré dans la clé de session qui fait partie du ticket.

Obtenez le ticket Kerberos initial

Quand l'utilisateur marche à un poste de travail, seulement l'une seule pièce des informations peut prouver le son identité : le mot de passe d'utilisateur. L'échange initial avec le serveur d'authentification est conçu pour réduire l'occasion que le mot de passe sera compromis, tout en en même temps ne permettant pas à un utilisateur pour authentifier correctement elle/lui-même sans connaissance de ce mot de passe. Le processus d'ouvrir une session semble à l'utilisateur être identique qu'ouvrant une session à un système à temps partagé. Dans les coulisses, bien que, il soit très différent.

L'utilisateur est incité pour son nom d'utilisateur. Une fois qu'il a été entré, une demande est envoyée au serveur d'authentification contenant le nom d'utilisateur et le nom d'un service spécial connu sous le nom de service de distribution de tickets.

Les contrôles de serveur d'authentification qu'ils connaissent le client. Si oui, il génère une clé de session aléatoire qui plus tard sera utilisée entre le client et le serveur-distributeur de tickets. Il crée alors un ticket pour le serveur-distributeur de tickets qui contient le nom du client, le nom du serveur-distributeur de tickets, le moment en cours, une vie pour le ticket, l'adresse IP du client, et la clé de session aléatoire juste créée. Ceci est tout chiffré dans une clé connue seulement au serveur-distributeur de tickets et au serveur d'authentification.

Le serveur d'authentification envoie alors le ticket, avec une copie de la clé de session aléatoire et de quelques informations complémentaires, de nouveau au client. Cette réponse est chiffrée dans la clé privée du client, connue seulement au Kerberos et au client, qui est dérivé du mot de passe d'utilisateur.

Une fois que la réponse a été reçue par le client, l'utilisateur est demandé 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, avec certaines des autres informations, sont enregistrés pour une utilisation future, et le mot de passe d'utilisateur et la clé DES sont effacés de la mémoire.

Une fois l'échange a été terminé, le poste de travail possède les informations qu'il peut employer pour prouver l'identité de son utilisateur pour la vie du ticket distribué. Tant que le logiciel sur le poste de travail n'avait pas été précédemment trifouillé, aucune informations n'existe qui permettra à quelqu'un d'autre pour personnifier l'utilisateur au delà de la vie du ticket.

Demandez un service de Kerberos

Pour le moment, permettez-nous feignent que l'utilisateur a déjà un ticket pour le serveur désiré. Afin d'accéder au serveur, l'application établit un authentificateur contenant le nom et l'adresse IP du client, et le temps en cours. L'authentificateur est alors chiffré dans la clé de session qui a été reçue avec le ticket pour le serveur. Le client envoie alors l'authentificateur avec le ticket au serveur en quelque sorte défini 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 dans le ticket à celle dans l'authentificateur, l'adresse IP desquels la demande a été

reçue, et l'époque actuelle. Si tout s'assortit, il permet la demande de poursuivre.

On le suppose que des horloges sont synchronisées à dans plusieurs minutes. Si le temps dans la demande est trop lointain à l'avenir ou le passé, le serveur traite la demande comme tentative de rejouer une demande précédente. On permet également au le serveur pour maintenir tous des demandes de passé avec les horodateurs qui sont encore valides. Afin de déjouer plus loin des attaques par relecture, une demande reçue avec le même ticket et l'horodateur qu'un déjà reçu peut être jeté.

En conclusion, si le client spécifie qu'il veut que le serveur prouve son identité aussi, le serveur additionne un à l'horodateur le client introduit l'authentificateur, chiffre le résultat dans la clé de session, et envoie le résultat de nouveau au client.

Àla fin de cet échange, le serveur est certain que, selon le Kerberos, le client soit qui il indique qu'il est. Si l'authentification mutuelle se produit, le client est également convaincu que le serveur est authentique. D'ailleurs, le partage de client et serveur une clé que personne d'autre connaît, et peut sans risque supposer qu'un message raisonnablement récent chiffré dans cette clé a commencé avec l'autre interlocuteur.

Obtenez les tickets de serveur de Kerberos

Rappelez-vous qu'un ticket est seulement bon pour un serveur unique. En soi, il est nécessaire d'obtenir un ticket séparé pour chaque service que le client veut l'utiliser. Des tickets pour différents serveurs peuvent être obtenus du service de distribution de tickets. Puisque le service de distribution de tickets est lui-même un service, il se sert du protocole d'accès de service décrit dans la section précédente.

Quand un programme exige un ticket qui n'a pas été déjà demandé, il envoie une demande au serveur-distributeur de tickets. La demande contient le nom du serveur pour lequel un ticket est demandé, avec le ticket distribué et d'un authentificateur construit comme décrit dans la section précédente.

Le serveur-distributeur de tickets vérifie alors l'authentificateur et le ticket distribué comme décrit ci-dessus. Si valide, le serveur-distributeur de tickets génère une nouvelle clé de session aléatoire à utiliser entre le client et le nouveau serveur. Il construit alors un ticket pour le nouveau serveur contenant le nom du client, le nom du serveur, le temps en cours, l'adresse IP du client et la nouvelle clé de session qu'elle a juste générée. La vie du nouveau ticket est le minimum de la vie restante pour le ticket distribué et du par défaut pour le service.

Le serveur-distributeur de tickets envoie alors le ticket, avec la clé de session et d'autres informations, de nouveau au client. Cette fois, cependant, la réponse est chiffrée dans la clé de session qui faisait partie du ticket distribué. De cette façon, là n'est aucun besoin de l'utilisateur d'entrer son mot de passe de nouveau.

La base de données Kerberos

Jusqu'à ce point, nous avons discuté des exécutions exigeant l'accès en lecture seule à la base de données Kerberos. Ces exécutions sont exécutées par le service d'authentification, qui peut fonctionner sur des ordinateurs de maître et d'esclave.

Dans cette section, nous discutons les exécutions qui exigent l'accès en écriture à la base de

données. Ces exécutions sont exécutées par le service de gestion, appelé le service de supervision de base de données Kerberos (KDBM). L'implémentation en cours stipule que des modifications peuvent seulement être apportées à la base de données Kerberos principale ; les copies d'esclave sont en lecture seule. Par conséquent, le serveur KDBM peut seulement fonctionner sur l'ordinateur principal de Kerberos.

Notez que, alors que l'authentification peut encore se produire (sur des esclaves), des demandes de gestion ne peuvent pas être entretenues si l'ordinateur maître est vers le bas. Dans notre expérience, ceci n'a pas présenté un problème, car les demandes de gestion sont peu fréquentes.

Le KDBM traite des demandes des utilisateurs de changer leurs mots de passe. Le côté client de ce programme, qui envoie des demandes au KDBM au-dessus du réseau, est le programme kpasswd. Le KDBM reçoit également des demandes des administrateurs de Kerberos, qui peuvent ajouter des principaux à la base de données, aussi bien que change des mots de passe pour les principaux existants. Le côté client du programme de gestion, qui envoie également des demandes au KDBM au-dessus du réseau, est le programme de kadmin.

Le serveur KDBM

Le serveur KDBM reçoit des demandes d'ajouter des principaux à la base de données ou de changer les mots de passe pour les principaux existants. Ce service est seul parce que le service de distribution de tickets n'émettra pas des tickets 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 distribué). Le but de ceci est d'exiger de l'utilisateur d'entrer un mot de passe. Si ce n'étaient pas aussi, alors si un utilisateur partait de son poste de travail sans surveillance, un passant pourrait marcher et changer son mot de passe pour eux, quelque chose qui devrait être empêchée. De même, si un administrateur partait de son poste de travail sans surveillance, un passant pourrait changer n'importe quel mot de passe dans le système.

Quand le serveur KDBM reçoit une demande, elle l'autorise en comparant le nom principal authentifié du demandeur de la modification au nom principal de la cible de la demande. S'ils sont identiques, on permet la demande. S'ils ne sont pas identiques, le serveur KDBM consulte une liste de contrôle d'accès (enregistrée dans un fichier sur le système Kerberos principal). Si le nom principal du demandeur est trouvé dans ce fichier, la demande est permise, autrement on lui refuse.

Par la convention, les noms avec un exemple NUL (l'exemple par défaut) n'apparaissent pas dans le fichier de liste de contrôle d'accès ; au lieu de cela, un exemple d'admin est utilisé. Par conséquent, pour qu'un utilisateur devienne un administrateur de Kerberos qu'un admin citent pour ce nom d'utilisateur doit être créé, et ajouté à la liste de contrôle d'accès. Cette convention permet à un administrateur pour utiliser un mot de passe différent pour la gestion de Kerberos alors qu'il l'utiliserait pour la procédure de connexion normale.

Toutes les demandes au programme KDBM, si laissé ou refusé, sont enregistré.

Le kadmin et les programme kpasswd

Les administrateurs du Kerberos emploient le programme de kadmin pour ajouter des principaux à la base de données, ou changent les mots de passe des principaux existants. Un administrateur est requis d'entrer le mot de passe pour leur nom d'exemple d'admin quand ils appellent le programme de kadmin. Ce mot de passe est utilisé pour chercher un ticket pour le serveur KDBM.

Les utilisateurs peuvent changer leurs mots de passe de Kerberos utilisant le programme kpasswd. Ils sont exigés pour entrer leur ancien mot de passe quand ils appellent le programme. Ce mot de passe est utilisé pour chercher un ticket pour le serveur KDBM.

Réplication de base de données Kerberos

Chaque royaume de Kerberos a un ordinateur principal de Kerberos, qui loge la copie principale de la base de données d'authentification. Il est possible (bien que non nécessaire) d'avoir supplémentaire, des copies en lecture seule de la base de données sur les ordinateurs slaves ailleurs dans le système. Les avantages de avoir de plusieurs copies de la base de données sont ceux habituellement citée pour la réplication : une représentation plus facilement disponible et meilleure. Si l'ordinateur maître est vers le bas, l'authentification peut encore être réalisée sur un des ordinateurs slaves. La capacité d'exécuter l'authentification sur des n'importe quels de plusieurs ordinateurs réduit la probabilité d'un étranglement à l'ordinateur maître.

La conservation de plusieurs copies de la base de données introduit le problème de la cohérence de données. Nous avons constaté que les méthodes très simples suffisent pour traiter l'incohérence. La base de données principale est vidée chaque heure. La base de données est envoyée, en sa totalité, aux ordinateurs slaves, qui mettent à jour alors leurs propres bases de données. Un programme sur l'hôte principal, appelé le kprop, envoie la mise à jour à un programme de pair, appelé le kpropd, s'exécutant sur chacun des ordinateurs slaves. Le premier kprop envoie une somme de contrôle de la nouvelle base de données qu'elle est sur le point d'envoyer. La somme de contrôle est chiffrée dans la clé de base de données principale de Kerberos, que les ordinateurs de Kerberos de maître et d'esclave possèdent. Les données sont alors transférées au-dessus du réseau vers le kpropd sur l'ordinateur slave. Le serveur slave de propagation calcule une somme de contrôle des données qu'elle a reçues, et si elles apparient 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 dans la base de données Kerberos sont chiffrés dans la clé de base de données principale par conséquent, les informations passées du maître pour asservir au-dessus du réseau n'est pas utile à une oreille indiscrète. Cependant, il est essentiel que seulement les informations de l'hôte principal soient reçues par les esclaves, et que le trifouillage des données soit détecté, ainsi la somme de contrôle.

Kerberos du regard extérieur dedans

Cette section décrit le Kerberos du point de vue pratique, d'abord comme vu par l'utilisateur, puis du point de vue du programmeur d'application, et en conclusion, par les tâches de l'administrateur de Kerberos.

Vue de l'oeil de l'utilisateur de Kerberos

Si tout va bien, l'utilisateur notera à peine que le Kerberos est présent. Dans notre implémentation UNIX, le ticket distribué est obtenu du Kerberos en tant qu'élément du processus de procédure de connexion. Changer du mot de passe du Kerberos d'un utilisateur fait partie du programme de passwd. Et des tickets Kerberos sont automatiquement détruits quand des journaux de l'utilisateur.

Si la session d'ouverture de connexion de l'utilisateur dure plus long que la vie du ticket distribué (actuellement 8 heures), l'utilisateur notera la présence du Kerberos parce que la prochaine fois

qu'une application Kerberos-authentifiée est exécutée, elle échouera. Le ticket Kerberos pour lui aura expiré. À ce moment là, l'utilisateur peut lancer le programme de kinit pour obtenir un nouveau ticket pour le serveur-distributeur de tickets. Comme en ouvrant une session, un mot de passe doit être fourni afin de l'obtenir. Un utilisateur exécutant la commande de klist hors de la curiosité peut être étonné à tous les tickets ce qui ont été silencieusement obtenus en son nom pour des services ce qui exigent l'authentification Kerberos.

Kerberos du point de vue du programmeur

Un programmeur écrivant une application de Kerberos ajoutera souvent l'authentification déjà à une application de réseau existant se composant d'un côté de client et serveur. Nous appelons ce « Kerberizing » de processus un programme. Kerberizing implique habituellement de faire un appel à la bibliothèque de Kerberos afin d'exécuter l'authentification à la requête initiale pour le service. Il peut également impliquer des appels à la bibliothèque DES pour chiffrer les messages et les données qui sont ultérieurement envoyés entre le client d'application et le serveur d'applications.

Les fonctions de bibliothèque les plus utilisées généralement sont krb_mk_req sur le côté client, et krb_rd_req sur le côté serveur. La routine de krb_mk_req prend comme paramètres le nom, l'exemple, et le royaume du serveur de cible, qui sera prié, et probablement une somme de contrôle des données à envoyer. Le client envoie alors le message retourné par l'appel de krb_mk_req au-dessus du réseau au côté serveur de l'application. Quand le serveur reçoit ce message, il fait un appel au krb_rd_req de routine de bibliothèque. La routine renvoie un jugement au sujet de l'authenticité de l'identité alléguée de l'expéditeur.

Si l'application exige que les messages envoyés entre le client et serveur soient secrets, alors des appels de bibliothèque peuvent être faits au krb_mk_priv (krb_rd_priv) pour chiffrer des messages (de déchiffrage) dans la clé de session que les deux côtés partagent maintenant.

Le travail de l'administrateur de Kerberos

Le travail de l'administrateur de Kerberos commence par exécuter un programme pour initialiser la base de données. Un autre programme doit être lancé pour enregistrer les principaux essentiels dans la base de données, telle que le nom de l'administrateur de Kerberos avec un exemple d'admin. Le serveur d'authentification Kerberos et le serveur de gestion doivent être démarrés. S'il y a les bases de données slaves, l'administrateur doit arranger que les programmes pour propager des mises à jour de base de données de maître aux esclaves soient donnés un coup de pied hors fonction périodiquement.

Après que ces mesures initiales aient été prises, l'administrateur manipule la base de données au-dessus du réseau, utilisant le programme de kadmin. Par ce programme, de nouveaux principaux peuvent être ajoutés, et des mots de passe peuvent être changés.

En particulier, quand une nouvelle application de Kerberos est ajoutée au système, l'administrateur de Kerberos doit prendre quelques mesures pour l'obtenir fonctionnant. Le serveur doit être enregistré dans la base de données, et a assigné une clé privée (habituellement c'est une clé aléatoire automatiquement générée). Puis, quelques données (clé y compris du serveur) doivent être extraites de la base de données et être installées dans un fichier sur l'ordinateur du serveur. Le fichier par défaut est /etc/srvtab. La routine de bibliothèque de krb_rd_req appelée par le serveur (voyez la section précédente) emploie les informations dans ce fichier pour déchiffrer des messages envoyés chiffrés dans la clé privée du serveur. Le fichier de /etc/srvtab authentifie le serveur pendant qu'un mot de passe tapé sur un terminal authentifie l'utilisateur.

L'administrateur de Kerberos doit également s'assurer que les ordinateurs de Kerberos sont physiquement sécurisés, et serait également sage de mettre à jour des sauvegardes de la base de données principale.

L'image plus grande de Kerberos

Dans cette section, nous décrivons comment des adaptations de Kerberos dans l'environnement Athena, y compris son utilisation par d'autres services réseau et applications, et comment elle interagit avec des royaumes distants de Kerberos. Pour une description plus complète de l'environnement Athena, voir s'il vous plaît le G.W. Treese.

L'utilisation d'autres services réseau du Kerberos

Plusieurs applications réseau ont été modifiées d'utiliser le Kerberos. Les commandes de rlogin et de rsh essayent d'abord d'authentifier utilisant le Kerberos. Un utilisateur avec les tickets Kerberos valides mettent en boîte le rlogin à un autre ordinateur d'Athéna sans devoir installer des fichiers .rhosts. Si l'authentification Kerberos échoue, les programmes tombent de retour sur leurs méthodes habituelles d'autorisation, dans ce cas, les fichiers .rhosts.

Nous avons modifié le Post Office Protocol pour utiliser le Kerberos pour authentifier les utilisateurs qui souhaitent récupérer leur courrier électronique du « bureau de poste. » Un programme de la livraison de message, appelé Zephyr, a été développé récemment chez Athéna, et il utilise le Kerberos pour l'authentification aussi bien.

Le programme pour s'inscrire de nouveaux utilisateurs, appelé le registre, utilise le système de gestion des services (SMS) et le Kerberos. À partir du SMS, il détermine si l'information saisie par le nouvel utilisateur potentiel d'Athéna, tel que le nom et le numéro d'identification MIT, est valide. Il vérifie alors avec le Kerberos pour voir si le nom d'utilisateur demandé est seul. Si tout va bien, une nouvelle entrée est faite à la base de données Kerberos, contenant le nom d'utilisateur et mot de passe.

Pour une analyse détaillée de l'utilisation du Kerberos de sécuriser le système de fichiers en réseau du Sun, référez-vous s'il vous plaît à l'annexe.

Interaction avec l'autre Kerberi

On s'attend à ce que les différents organismes administratifs veuillent utiliser le Kerberos pour l'authentification de l'utilisateur. On s'attend à ce également que dans de nombreux cas, les utilisateurs dans une organisation veuillent utiliser des services dans des autres. Domaines administratifs de multiple de supports de Kerberos. La spécification des noms dans le Kerberos inclut un champ appelé le royaume. Ce champ contient le nom du domaine administratif dans lequel l'utilisateur doit être authentifié.

Des services sont habituellement enregistrés dans un royaume simple et recevront seulement des qualifications émises par un serveur d'authentification pour ce royaume. Un utilisateur est habituellement enregistré dans un royaume simple (le royaume local), mais il est possible à elle/à lui pour obtenir des qualifications émises par un autre royaume (le royaume distant), en vertu de l'authentification fournie par le royaume local. Les qualifications valides dans un royaume distant indiquent le royaume dans lequel l'utilisateur a été initialement authentifié. Les services dans le royaume distant peuvent choisir si honorer ces qualifications, selon le degré de Sécurité exigé et du niveau de confiance dans le royaume qui a au commencement authentifié l'utilisateur.

Afin d'exécuter l'authentification de croix-royaume, il est nécessaire que les administrateurs de chaque paire de royaumes sélectionnent une clé à partager entre leurs royaumes. Un utilisateur dans le royaume local peut alors demander un ticket distribué du serveur d'authentification locale pour le serveur-distributeur de tickets dans le royaume distant. Quand ce ticket est utilisé, le serveur-distributeur de tickets distant identifie que la demande n'est pas de son propre royaume, et elle emploie la clé précédemment permutée pour déchiffrer le ticket distribué. Il émet alors un ticket comme il normalement, sauf que le champ de royaume pour le client contient le nom du royaume dans lequel le client a été initialement authentifié.

Cette approche a pu être étendue pour permettre à un pour s'authentifier par une gamme de royaumes jusqu'à atteindre le royaume avec le service désiré. Afin de faire ceci, bien que, il soit nécessaire d'enregistrer le chemin entier qui a été pris, et pas simplement le nom du royaume initial dans lequel l'utilisateur a été authentifié. Dans une telle situation, tout ce qui est connu par le serveur est qu'A indique que B indique que le C indique que l'utilisateur est untel. Cette déclaration peut sont de confiance seulement si chacun le long du chemin est également fait confiance.

Problèmes et question non résolus de Kerberos

Il y a un certain nombre de problèmes et question non résolus associés avec le mécanisme d'authentification Kerberos. Parmi les questions soyez comment décider la vie correcte pour un ticket, comment permettre des proxys, et comment garantir l'intégrité de poste de travail.

Le problème de vie de ticket est une question de choisir le compromis approprié entre la Sécurité et la commodité. Si la vie d'un ticket est longue, alors si un ticket et sa clé de session associée sont dérobés ou mal placés, ils peuvent être utilisés pendant une plus longue période. Une telle informations peuvent être dérobées si un utilisateur oublie de se déconnecter d'un poste de travail public. Alternativement, si un utilisateur a été authentifié sur un système qui permet des plusieurs utilisateurs, un autre utilisateur avec l'accès à enraciner pourrait pouvoir trouver les informations requises pour utiliser les tickets dérobés. Le problème avec donner à un ticket une vie courte, cependant, est que quand elle expire, l'utilisateur devra obtenir un neuf qui exige de l'utilisateur d'entrer le mot de passe de nouveau.

Un problème non résolu est le problème de proxy. Comment un utilisateur authentifié peut-il permettre à un serveur pour saisir d'autres services réseau en son nom ? Un exemple où ce serait important est l'utilisation d'un service qui accédera aux fichiers protégés directement d'un serveur de fichiers. Est un autre exemple de ce problème ce que nous appelons expédition d'authentification. Si un utilisateur est enregistré dans un poste de travail et des logins à un serveur distant, il serait gentil si l'utilisateur avait accès aux mêmes services disponibles localement, tout en exécutant un programme sur le serveur distant. Ce qui fait ce difficile est que l'utilisateur ne pourrait pas faire confiance au serveur distant, ainsi l'expédition d'authentification n'est pas desirable dans des tous les cas. Nous n'avons pas actuellement une solution au problème.

Un autre problème, et un qui sont importants dans l'environnement Athena, est comment garantir l'intégrité de l'exécution de logiciel sur un poste de travail. Ce n'est pas tellement d'un problème sur les postes de travail privés puisque l'utilisateur qui l'utilisera a le contrôle de lui. Sur les postes de travail publics, cependant, quelqu'un pourrait avoir été livré le long et modifié le programme de procédure de connexion pour sauvegarder le mot de passe d'utilisateur. La seule solution actuellement disponible dans notre environnement est de le rendre difficile pour que les personnes modifient l'exécution de logiciel sur les postes de travail publics. Une meilleure solution exigerait que la clé de l'utilisateur ne laissent jamais un système que l'utilisateur connaît peut sont

de confiance. Une manière que ceci pourrait être fait serait si l'utilisateur possédait une carte à puce capable de faire les cryptages priés dans le protocole d'authentification.

État de Kerberos

Une version de prototype de Kerberos est entrée dans la production en septembre de 1986. Depuis janvier de 1987, le Kerberos a été les moyens uniques d'Athéna de projet d'authentifier ses 5,000 utilisateurs, 650 postes de travail, et 65 serveurs. En outre, le Kerberos maintenant est utilisé au lieu des fichiers .rhosts pour l'accès de contrôle dans plusieurs des systèmes à temps partagé d'Athéna.

Accusés de réception de Kerberos

Le Kerberos a été au commencement conçu par Steve Miller et Clifford Neuman avec des suggestions de Jeff Schiller et de Jerry Saltzer. Depuis lors, nombreux d'autres personnes ont été comportées du projet. Parmi eux sont JIM Aspnes, Bob Baldwin, John Barba, Richard Basch, fleur de JIM, facture Bryant, marque Colan, Français de Rob, Dan Geer, John Kohl, John Kubiatowicz, Bob Mckie, Brian Murphy, John Ostlund Ken Raeburn, Chris Reed, Jon Rochlis, Mike Shanzer, facture Sommerfeld, Ted T'so, victoire Treese, et Stan Zanarotti.

Nous sommes reconnaissants à Dan Geer, à Kathy Lieben, à Josh Lubarr, à Ken Raeburn, à Jerry Saltzer, à Ed Steiner, à Robbert van Renesse, et à victoire Treese dont les suggestions ont beaucoup amélioré des ébauches plus tôt de ce document.

Jedlinsky, J.T. Kohl, et W.E. Sommerfeld, « le système de notification de zéphyr, » dans des actes de la conférence d'Usenix (Winter, 1988).

M.A. Rosenstein, D.E. Geer, et P.J. Levine, dans des actes de la conférence d'Usenix (Winter, 1988).

R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, et B. Lyon, « conception et réalisation du Network File System de Sun, » dans des actes de la conférence d'Usenix (Summer, 1985).

Annexe : Application de Kerberos au Systèmes de fichiers en réseau (NFS) du Sun

Un élément clé du système de poste de travail d'Athéna de projet est l'interposition du réseau entre le poste de travail de l'utilisateur et son stockage de fichier privé (répertoire home). Tout le stockage privé réside sur un ensemble d'ordinateurs (actuellement VAX 11/750s) qui sont dédiés à ce but. Ceci nous permet pour offrir les postes de travail Unix disponibles de services en fonction publiquement -. Quand un utilisateur ouvre une session à un de ces derniers publiquement - les postes de travail disponibles, valident plutôt alors son nom et mot de passe contre un fichier de mot de passe de riverain, nous employons le Kerberos pour déterminer son authenticité. Le programme de procédure de connexion incite pour un nom d'utilisateur (comme sur tout système Unix). Ce nom d'utilisateur est utilisé pour chercher un ticket distribué de Kerberos. Le programme de procédure de connexion emploie le mot de passe pour générer une clé DES pour déchiffrer le ticket. Si le déchiffrement est réussi, le répertoire home de l'utilisateur se trouve en consultant le Hesiod nommant le service et monté par le NFS. Le programme de procédure de connexion fait tourner alors le contrôle au shell de l'utilisateur, qui alors peut

exécuter les fichiers traditionnels de personnalisation de par-utilisateur parce que le répertoire home « est maintenant relié » au poste de travail. Le service de Hesiod est également utilisé pour construire une entrée dans le fichier local de mot de passe. (C'est au profit des programmes ces données pour la consultation dans /etc/passwd.)

De plusieurs options pour la prestation du d'archivage distant, nous avons choisi le système de fichiers en réseau du Sun. Cependant ce système n'engrène pas avec nos besoins d'une manière cruciale. Le NFS suppose que tous les postes de travail se rangent dans deux catégories (comme visualisé du point de vue d'un serveur de fichiers) : fait confiance et non approuvé. Les systèmes non approuvés ne peuvent accéder à aucun fichiers du tout, fait confiance peuvent. Des systèmes de confiance sont complètement faits confiance. On le suppose qu'un système de confiance est géré par la Gestion amicale. Spécifiquement, il est possible d'un poste de travail de confiance de déguiser car n'importe quel utilisateur valide du système d'archivage et accède ainsi à juste au sujet de chaque fichier sur le système. (Seulement des fichiers possédés par la « racine » sont exemptés.)

Dans notre environnement, la Gestion d'un poste de travail (dans le sens traditionnel de la gestion du système UNIX) est aux mains de l'utilisateur actuellement utilisant elle. Nous ne faisons aucun secret du mot de passe root sur nos postes de travail, car nous nous rendons compte qu'un utilisateur véritablement peu amical peut se casser dedans par le fait même qu'il s'assied dans le même emplacement physique que l'ordinateur et a accès à toutes les fonctions de console. Par conséquent nous ne pouvons pas vraiment faire confiance à nos postes de travail dans la traduction NFS de la confiance. Pour nous permettre aux contrôles d'accès appropriés dans notre environnement avons dû apporter quelques modifications au logiciel NFS de base, et intégrons le Kerberos dans le schéma.

NFS non modifié de Kerberos

Dans l'implémentation du NFS par laquelle nous avons commencé (de l'université du Wisconsin), l'authentification a été fournie sous forme de partie de données incluses dans chaque demande NFS (appelée un « laisser-passer » en terminologie NFS). Ce laisser-passer contient des informations sur l'identifiant d'utilisateur unique (UID) du demandeur et une liste des identifiants de groupe (GIDs) de l'adhésion du demandeur. Ces informations sont alors utilisées par le serveur NFS pour vérifier d'accès. La différence entre un poste de travail de confiance et non-fait confiance est si ses qualifications sont reçues par le serveur NFS.

Le Kerberos a modifié le NFS

Dans notre environnement, les serveurs NFS doivent recevoir des qualifications d'un poste de travail si et seulement si les qualifications indiquent l'UID de l'utilisateur du poste de travail, et pas d'autre.

Une solution évidente serait de changer la nature des qualifications de simples indicateurs d'UID et GIDs au véritable Kerberos a authentifié des données. Cependant une baisse de performances significative serait payée si cette solution étaient adoptées. Des qualifications sont permutées sur chaque exécution NFS comprenant tout le disque lu et écrivent des activités. Y compris une authentification Kerberos sur chaque disque la transaction ajouterait un nombre équitable de véritables cryptages (faits en logiciel) par transaction et, selon nos calculs d'enveloppe, aurait fourni la représentation inacceptable. (Il aurait également exigé placer les routines de bibliothèque de Kerberos dans l'espace d'adressage de noyau.)

Nous avons eu besoin d'une approche hybride, décrite ci-dessous. L'idée de base est d'avoir les qualifications de carte de serveur NFS reçues des postes de travail de client, à un laisser-passer valide (et probablement différent) sur le système serveur. Ce mappage est effectué au noyau du serveur sur chaque transaction NFS et est installé au temps de « support » par un processus de niveau utilisateur qui s'engage dans l'authentification modérée par Kerberos avant d'établir un mappage de créance de noyau valide.

Pour implémenter ceci nous avons ajouté un nouvel appel système au noyau (requis seulement sur des systèmes serveurs, pas sur des systèmes client) qui prévoit le contrôle de la fonction de mappage cette les qualifications entrantes de cartes des postes de travail de client aux qualifications valides pour l'usage sur le serveur (le cas échéant). La fonction de mappage de base trace le tuple :

àun laisser-passer valide NFS sur le système serveur. L'ADRESSE IP DU CLIENT est extraite du paquet de demandes NFS fourni par le système client. Remarque: toutes les informations dans le laisser-passer client-généré excepté l'UID-ON-CLIENT sont jetées.

Si aucun mappage n'existe, le serveur réagit dans une de deux manières, dépendant il est configuré. Dans notre configuration amicale nous transférons les demandes unmappable dans les qualifications pour l'utilisateur « personne » qui n'a aucun accès privilégié et a un seul UID. Les serveurs peu amicaux renvoient une erreur d'accès NFS quand aucun mappage valide ne peut être trouvé pour un laisser-passer entrant NFS.

Notre nouvel appel système est utilisé pour ajouter et supprimer des entrées de la carte résidente de noyau. Il fournit également la capacité de vider toutes les entrées qui tracent à un UID spécifique sur le système serveur, ou vide toutes les entrées d'une ADRESSE IP DU CLIENT donnée.

Nous avons modifié le démon de support (qui traite les demandes de montage NFS sur des systèmes serveurs) pour recevoir un nouveau type de transaction, la demande de mappage d'authentification Kerberos. Fondamentalement, en tant qu'élément du processus de support, le système client fournit un authentificateur de Kerberos avec une indication de son UID-ON-CLIENT (chiffré dans l'authentificateur de Kerberos) sur le poste de travail. Le démon du support du serveur convertit le nom de principal Kerberos en nom d'utilisateur local. Ce nom d'utilisateur est alors recherché dans un fichier spécial pour rapporter l'UID de l'utilisateur et la liste de GIDs. Pour l'efficacité, ce fichier est un fichier de base de données de ndbm avec le nom d'utilisateur comme clé. De ces informations, un laisser-passer NFS est construit et remis au noyau comme mappage valide du <CLIENT-IP-ADDRESS, tuple CLIENT-UID> pour cette demande.

Au temps d'unmount une demande est envoyée au démon de support d'enlever le mappage précédemment ajouté du noyau. Il est également possible d'envoyer une demande au temps de déconnexion d'infirmer tout le mappage pour l'utilisateur courant sur le serveur en question, de ce fait nettoyant tous mappages restants qui existent (cependant ils ne devraient pas) avant que le poste de travail soit rendu disponible pour le prochain utilisateur.

Implications en matière de sécurité de Kerberos du NFS modifié

Cette implémentation n'est pas complètement sécurisée. Pour commencer, des données d'utilisateur sont encore envoyées à travers le réseau dans un décrypté, et donc interceptable, forme. Le bas niveau, authentification de par-transaction est basé sur un <CLIENT-IP-ADDRESS, décrypté fourni par paires CLIENT-UID> dans le paquet de demandes. Ces informations ont pu être modifiées et ainsi Sécurité être compromises. Cependant, il convient noter que seulement

tandis qu'un utilisateur utilise activement son les fichiers (c'est-à-dire, tandis qu'ouvert une session) sont les mappages valides en place et donc cette forme d'attaque est limités à quand l'utilisateur en question est ouvert une session. Quand un utilisateur n'est pas ouvert une session, aucune quantité de contrefaçon d'adresse IP ne permettra à accès non autorisé au son des fichiers.

Références de Kerberos

1. S.P. Miller, BC Neuman, J.I. Schiller, et J.H. Saltzer, section E.2.1 : Système d'authentification Kerberos et d'autorisation, M.I.T. Projet Athéna, Cambridge, le Massachusetts (décembre 21, 1987).
2. E. Balkovich, S.R. Lerman, et R.P. Parmelee, « calculant dans l'enseignement supérieur : L'expérience d'Athéna, » transmissions de l'ACM, vol. 28(11), Pp. 1214-1224, ACM (novembre, 1985).
3. R.M. Needham et M.D. Schroeder, « utilisant le cryptage pour l'authentification dans de grands réseaux d'ordinateurs, » transmissions de l'ACM, vol. 21(12), Pp. 993-999 (décembre, 1978).
4. V.L. Voydock et S.T. Kent, « mécanismes de sécurité dans des protocoles réseau de haut niveau, » calculant des analyses, vol. 15(2), ACM (juin 1983).
5. National Bureau of Standards, « norme de chiffrement de données, » publication 46 de Normes fédérales pour le traitement de l'information, bureau d'impression du gouvernement, Washington, DC (1977).
6. Tinctorial de fournisseur de services, « Hesiod, » dans des actes de la conférence d'Usenix (Winter, 1988).
7. W.J. Bryant, le tutoriel du programmeur de Kerberos, projet Athéna MIT (en préparation).
8. W.J. Bryant, le manuel de l'administrateur de Kerberos, projet Athéna MIT (en préparation).
9. G.W. Treese, « Berkeley Unix sur 1000 postes de travail : Athéna change à 4.3BSD," dans des actes de la conférence d'Usenix (Winter, 1988).
10. C.A. DellaFera, M.W. Eichin, R.S. French, D.C. Jedlinsky, J.T. Kohl, et W.E. Sommerfeld, « le système de notification de zéphyr, » dans des actes de la conférence d'Usenix (Winter, 1988).
11. M.A. Rosenstein, D.E. Geer, et P.J. Levine, dans des actes de la conférence d'Usenix (Winter, 1988).
12. R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, et B. Lyon, « conception et réalisation du Network File System de Sun, » dans des actes de la conférence d'Usenix (Summer, 1985).

Informations connexes

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