Gestion de Kerberos et d'autres services d'authentification dans Oracle® Solaris 11.2

Référence: E53967-02
Septembre 2014

ORACLE

Copyright © 2002, 2014, Oracle et/ou ses affiliés. Tous droits réservés.

Ce logiciel et la documentation qui l'accompagne sont fournis sous concédés sous licence et soumis à des restrictions d'utilisation et de divulgation et, sont protégés par les lois sur la propriété intellectuelle. Sauf disposition expresse de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, accorder de licence, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est livré sous licence au Gouvernement des Etats-Unis, ou à quiconque qui aurait souscrit la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour des applications dangereuses.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation.

Table des matières

Liste des tableaux

Liste des exemples

Utilisation de la présente documentation

- **Présentation** : décrit comment gérer l'authentification sécurisée sur un ou plusieurs systèmes Oracle Solaris. Ce manuel traite de PAM (Pluggable Authentication Modules), Kerberos, Simple Authentication and Security Layer-authentification (SASL) et RPC sécurisé pour NFS et NIS. Une annexe décrit les sondes DTrace pour Kerberos avec des exemples de leur utilisation.
- **Public visé** : système, sécurité et les administrateurs de sécurité de réseau.
- **Connaissances requises** : exigences en matière d'accès et de sécurité de réseau.

Bibliothèque de documentation du produit

Les informations de dernière minute et les problèmes connus pour ce produit sont inclus dans la bibliothèque de documentation accessible à l'adresse : http://www.oracle.com/pls/topic/lookup?ctx=E56338.

Accès aux services de support Oracle

Les clients Oracle ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info ou le site http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs si vous êtes malentendant.

Commentaires

Faites part de vos commentaires sur cette documentation à la page suivante : http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CHAPITRE 1

Utilisation de modules d'authentification enfichables

Ce chapitre traite du PAM (Pluggable Authentication Module, module d'authentification enfichable). PAM offre une structure dans laquelle il est possible d'insérer des vérifications pour les utilisateurs d'applications sur le Oracle Solaris (SE). PAM fournit une structure centrale pour gérer l'utilisation des applications, notamment l'authentification de l'utilisateur, les modifications des mots de passe, la fermeture et l'ouverture de la session utilisateur et le suivi des limitations de comptes, tels que l'heure du jour. PAM est extensible vers les applications tierces et peut donc offrir une gestion transparente de l'accès aux services sur un système.

Ce chapitre comprend les sections suivantes :

Nouveautés propres à l'authentication dans Oracle Solaris 11.2

Cette section met en évidence sur des informations importantes pour les utilisateurs existants et Kerberos nouvelles fonctions de technologies PAM dans cette version.

Nouveautés dans PAM

Les nouvelles fonctionnalités de structure PAM dans la version Oracle Solaris 11.2 incluent les fonctionnalités suivantes :

- Le module `pam_unix_account` prend en charge le contrôle d'accès basé sur des données temporelles à tous les services PAM spécifiés. Les opérations par conséquent, qui

appellent liées à la sécurité peut être limité à spécifique PAM les jours et les heures. Vous pouvez également fournir un fuseau horaire. Reportez-vous à la page de manuel `pam_unix_account`(5).

- Les mots-clés `access_times` et `access_tz` peuvent être affectés aux utilisateurs et aux rôles. A l'instar de tous les mots-clés des attributs de sécurité de l'utilisateur sont pris en charge dans, ces tous les services de noms. Pour plus d'informations, reportez-vous à la page de manuel `user_attr`(4).

Nouveautés dans Kerberos

Les nouvelles fonctionnalités de Kerberos dans la version Oracle Solaris 11.2 incluent les fonctionnalités suivantes :

- Kerberos prend en charge les tâches par lots s'exécutant à des moments quelconques. Les commandes qui permettent une exécution retardée, `at`, `batch` et`cron`, peuvent faire appel à des services Kerberos sans aucun besoin d'une intervention manuelle pour fournir l'authentification. Pour plus d'informations, reportez-vous à la section “Configuration de l'exécution retardée pour l'accès aux services Kerberos” à la page 127.
- Configurer les clients Kerberos vous pouvez automatiquement avec le programme d'installation automatisée (AI). Ces clients sont immédiatement capables d'héberger des services utilisant Kerberos. Fourniture d'infos de paramétrage pour un système automatiquement au cours du client Kerberos permet aux administrateurs de rapidement sécurisé et infrastructure déployer aisément une système importantes, ce qui permet de libérer des ressources administrateur.
  - Les systèmes client installés ne nécessitent pas d'étapes supplémentaires pour spécifier la configuration Kerberos et créer des clés de service Kerberos pour le système.
  - Lors de la saisie de différentes options sont le service Kerberos est - à - dire les dates clés pour les systèmes clients.

  Pour plus d'informations sur la configuration de Kerberos par l'intermédiaire du programme d'installation automatisée, reportez-vous à la section “ Configuration des clients Kerberos à l’aide de l’AI ” du manuel “ Installation des systèmes Oracle Solaris 11.2 ”.
- Le service `ktkt_warn` est désactivé par défaut. Vous devez explicitement activer initiale il TGT afin d'avertir les utilisateurs d'expiration ou de renouvelle automatiquement d'informations d'identification et de départ d'un utilisateur d'informations d'identification et de connexion. Pour plus d'informations, reportez-vous à la section “Renouvellement automatique de tous les tickets d'octroi de tickets” à la page 117.

Pour plus d'informations sur l'historique de Kerberos dans Oracle Solaris, reportez-vous à “ Composants des différentes versions Kerberos ” dans “ Administration Oracle Solaris 11.1 : Services de sécurité ”.

A propos de PAM

Les applications fournit une structure PAM pour effectuer différentes pour les fonctions d'authentification. L'authentification, il permet de centraliser la gestion des mots de passe, la gestion de session et, pour les utilisateurs d'applications les limitations de comptes, y compris programmes système. PAM permet à ces programmes, tels que `login`, `su` et `ssh` de demeurer inchangés si les informations de gestion de l'utilisateur changent. PAM site pour gérer les applications peuvent utiliser, d'informations d'identification et de leur propre compte, de session et les exigences de mot de passe. "Est" enfichés PAM à ces applications.

Présentation de la structure PAM

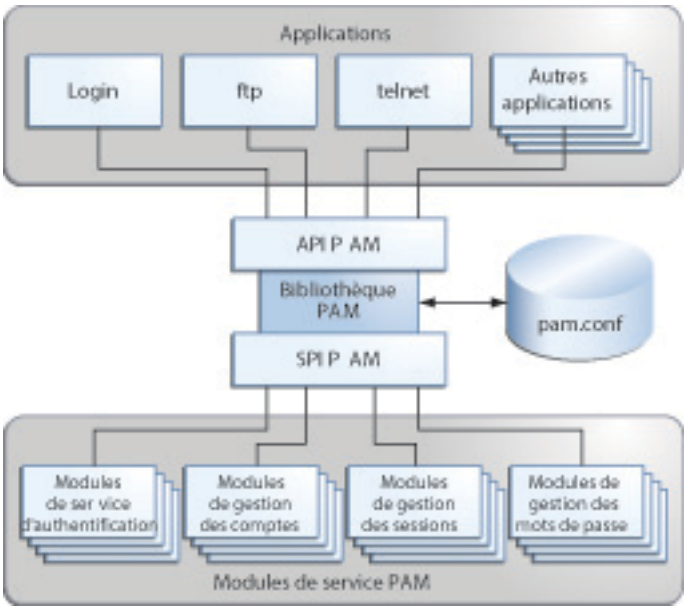
La structure PAM s'organise autour de quatre composants :

- Les applications qui utilisent PAM
- Structure PAM
- Modules de service PAM
- PAM modules, y compris une option de configuration et une affectation d'utilisateur

La structure permet d'uniformiser les activités liées à l'authentification. Cette approche permet aux développeurs d'applications d'utiliser les services PAM sans connaissance préalable de la sémantique de la stratégie d'authentification. Grâce à PAM, les administrateurs peuvent adapter le processus d'authentification aux besoins d'un système particulier sans avoir à modifier aucune application. PAM dans ce cas, il est préférable de configuration, les administrateurs ajustent le.

La figure ci-dessous illustre l'architecture PAM.

FIGURE 1-1 Architecture PAM



L'architecture fonctionne comme suit :

- Les applications communiquent avec la structure PAM par l'intermédiaire de l'API (Application Programming Interface, interface de programmation d'application) PAM.

  Pour plus d'informations sur l'utilisation de l'API, reportez-vous à `pam`(3PAM) man page and Chapitre 3, “ Writing PAM Applications and Services ” du manuel “ Developer’s Guide to Oracle Solaris 11 Security ”.
- Les modules de service PAM communiquent avec la structure PAM par l'intermédiaire de la SPI (Service Provider Interface, interface de fournisseur de services) PAM. Pour plus d'informations, reportez-vous à la page de manuel `pam_sm`(3PAM).

  Pour obtenir une brève description des modules de service sélectionnés, reportez-vous à la section “Modules de service PAM” à la page 40 et aux pages de manuel `pam.conf`(4) et `pam_user_policy`(5).

Les administrateurs peuvent configurer des séries de modules pour gérer les conditions requises pour le site. Cette série de modules est appelée une *pile* PAM. La pile de est évaluée dans l'ordre. Si une application requiert une ou plusieurs pile PAM, le développeur d'application doit créer plus d'un *nom de service*. Par exemple, le démon `sshd` fournit et nécessite plusieurs noms de service pour le PAM. Pour obtenir la liste des noms de

service PAM pour le démon`sshd`, lancez une recherche pour le terme `PAM` dans la page de manuel`sshd`(1M). Pour plus d'informations sur la pile PAM, reportez-vous aux étapes “Superposition PAM” à la page 35. “Exemple de superposition PAM” à la page 39 concernant une pile d'authentification PAM.

Avantages de l'utilisation de PAM

La vous permet de configurer PAM possédant une structure que les utilisateurs doivent satisfaire les conditions requises pour utiliser une application. Voici certains avantages qui PAM, vous obtenez les résultats suivants :

- Flexibilité de la stratégie de configuration PAM
  - Stratégie d'authentification par nom de service
  - Stratégie à l'échelle du site et par utilisateur PAM stratégie PAM
  - Choix administratif d'une stratégie d'authentification par défaut
  - Mise en oeuvre de plusieurs besoins de l'utilisateur sur les systèmes haute sécurité
- Facilité d'utilisation pour l'utilisateur final
  - Pas de nouvelle saisie des mots de passe identiques pour les services d'authentification différents
  - Invite en plusieurs services d'authentification utilisateur plutôt que d'imposer un utilisateur à taper plusieurs commandes
- Pour l'administrateur une plus grande facilité d'de configuration
  - Possibilité de transmettre d'informations sur les options de module de service PAM
  - Possibilité de mettre en place une stratégie de sécurité spécifique au site sans avoir à modifier les applications

Planification d'une configuration PAM propre à votre site

A la livraison, le PAM met en oeuvre une stratégie de sécurité standard de configuration qui couvre les services systèmes qui nécessitent une authentification, tels que`login` et `ssh`. Si vous avez besoin d'appliquer une stratégie de sécurité différente pour certains services. vous pouvez également créer une stratégie système pour les applications tierces, tenez compte des points suivants :

- Les fichiers de configuration fournis déterminer que le ne satisfont pas à vos besoins.

  Tester la configuration par défaut. Testez les fichiers par utilisateur dans le répertoire`/etc/security/pam_policy` . Testez si le nom de service par défaut `other` traite vos exigences. “Exemple de superposition PAM” à la page 39 vous guide dans la pile `other`.

- Pile identifier les besoins dont la modification des noms de service. Pour obtenir un exemple de modification de pile PAM de nom de service, reportez-vous à la section “Création d'un fichier de configuration PAM propre à votre site” à la page 23.
- Pour toute application tierce codé pour utiliser la structure PAM, déterminez les noms de service utilisés par les applications.
- Pour chaque nom de service, déterminent les modules PAM à utiliser.

  Reportez - vous à la section PAM 5 pages de manuel d 'pour les modules. Ces pages de manuel expliquent le fonctionnement de chaque module, la disponibilité des options et les interactions entre modules empilés. Pour obtenir une brève synthèse des modules sélectionnés, reportez-vous à la section “Modules de service PAM” à la page 40. Les modules sont également disponibles à partir de PAM en provenance de sources externes.
- Par nom de service, choisir leur ordre dans lequel vous souhaitez lancer l'modules.
- Sélectionnez l'indicateur de contrôle pour chaque module. Pour plus d'informations sur les indicateurs de contrôle, reportez-vous à la section “Superposition PAM” à la page 35. Notez que le contrôle les indicateurs peuvent avoir des problèmes de sécurité.

  Pour obtenir une représentation visuelle, reportez-vous aux schémas Figure 1-2, “Superposition PAM : effet des indicateurs de contrôle” et Figure 1-3, “Superposition PAM : détermination de la valeur intégrée”.
- Choisissez les options nécessaires pour chaque module. La page de manuel pour chaque module répertorie les options sont disponibles pour ce module.
- L'utilisation de l'application test avec la configuration PAM. En tant que rôle `root`, testez d'autres rôles, des utilisateurs privilégiés et des utilisateurs standard. Si certains utilisateurs ne sont pas autorisés à utiliser l'application, tester ces utilisateurs.

Affectation d'une stratégie PAM par utilisateur

Le module PAM `pam_user_policy` permet aux administrateurs système d'attribuer les configurations PAM spécifiques à chaque utilisateur. Lorsque ce module est le premier dans une pile PAM et que l'attribut de sécurité `pam_policy` pour l'utilisateur spécifie un fichier de configuration PAM, ce fichier spécifie la stratégie PAM pour l'utilisateur.

Pour plus d'informations, reportez-vous aux références suivantes :

- Page de manuel `pam_user_policy`(5)
- “Superposition PAM” à la page 35
- “Création d'un fichier de configuration PAM propre à votre site” à la page 23
- Exemple 1-3, “Définition d'une stratégie PAM par utilisateur à l'aide d'un profil de droits”

Configuration de PAM

Vous pouvez utiliser PAM en l'état. PAM cette section, vous trouverez des exemples de qui ne sont prises en compte les configurations par défaut.

TABLEAU 1-1 Liste des tâches PAM

Tâche	Description	Pour obtenir des instructions
Planification de l'installation PAM	PAM couvre les procédures de planification la personnalisation pour votre site.	“Planification d'une configuration PAM propre à votre site” à la page 21
Affectation d'une nouvelle stratégie PAM à un utilisateur.	Les paramètres d'authentification obligatoires personnalise par utilisateur pour plusieurs services.	“Création d'un fichier de configuration PAM propre à votre site” à la page 23
Répertoires créer des utilisateurs disposant de personnel chiffré.	Modifie une pile PAM crypté pour activer la création de répertoires personnels.	Exemple 1-1, “Utilisation d'une pile PAM modifiée pour créer un répertoire d'accueil chiffré”
Ajout de nouveaux modules PAM	Ce guide décrit la procédure d'installation et de tester les modules PAM personnalisée.	“Ajout d'un module PAM” à la page 25
Affecter une stratégie PAM autre que celui par défaut aux utilisateurs.	Montre comment ajouter une stratégie à un profil de droits PAM pour l'affectation à une plage d'utilisateurs pour les sites qui utilisent Kerberos, LDAP ou une combinaison de connexions.	“Assignation d'une stratégie PAM modifiée” à la page 28
Affecter une stratégie PAM autre que celui par défaut aux utilisateurs.	Les piles PAM personnalisée à tous les distribue des images système.	“Assignation d'une stratégie PAM modifiée” à la page 28
Initialisation de la journalisation des erreurs	Consigne les messages d'erreur PAM par l'intermédiaire de `syslog`.	“Journalisation de rapports d'erreur PAM” à la page 31
Résoudre des erreurs PAM.	Fournit les étapes permettant de localiser PAM, la résolution des erreurs de configuration, et enfin, vous pouvez tester.	“Dépannage des erreurs de configuration PAM” à la page 31

▼ Création d'un fichier de configuration PAM propre à votre site

Dans la configuration par défaut, les services d'entrée `ssh` et `telnet` sont traités par le nom de service `other`. Le fichier de configuration PAM dans cette procédure modifie les conditions requises pour `ssh` et `telnet`.

Avant de commencer Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. **Créez un fichier de configuration de stratégie PAM.**

Utilisez la commande `pfedit` pour créer le fichier. Placez le fichier dans un répertoire de configuration de site tel que `/opt`. Vous pouvez également le placer dans le répertoire `/etc/security/pam_policy`.

Remarque - Ne modifiez pas les fichiers existants dans le répertoire `/etc/security/pam_policy`.

Inclure des commentaires explicatifs dans le fichier.

```
# pfedit /opt/local_pam/ssh-telnet-conf
#
# PAM configuration which uses UNIX authentication for console logins,
# (see pam.d/login), and LDAP for SSH keyboard-interactive logins
# This stack explicitly denies telnet logins.
#
sshd-kbdint  auth requisite          pam_authtok_get.so.1
sshd-kbdint  auth binding            pam_unix_auth.so.1 server_policy
sshd-kbdint  auth required           pam_unix_cred.so.1
sshd-kbdint  auth required           pam_ldap.so.1
#
telnet auth     requisite    pam_deny.so.1
telnet account  requisite    pam_deny.so.1
telnet session  requisite    pam_deny.so.1
telnet password requisite    pam_deny.so.1
```

2. **Protégez le fichier.**

   Protégez le fichier avec l'appartenance `root` et les autorisations `444`.

```
# ls -l /opt/local_pam

total 5
-r--r--r--   1 root         4570 Jun 21 12:08 ssh-telnet-conf
```

3. **Affecter la stratégie.**

   Reportez-vous à la section “Assignation d'une stratégie PAM modifiée” à la page 28.

Exemple 1-1 Utilisation d'une pile PAM modifiée pour créer un répertoire d'accueil chiffré

Par défaut, le module `zfs_pam_key` ne se trouve pas dans le fichier `/etc/security/pam_policy/unix`. Dans cet exemple, l'administrateur crée une version de la stratégie PAM par utilisateur `unix`, puis utilise la nouvelle version pour créer les utilisateurs dont les répertoires personnels sont chiffrés.

```
# cp /etc/security/pam_policy/unix /opt/local_pam/unix-encrypt
# pfedit /opt/local_pam/unix-encrypt.conf
...
other   auth required           pam_unix_auth.so.1
other   auth required           pam_unix_cred.so.1
```

```
## pam_zfs_key auto-creates an encrypted home directory
##
other auth required           pam_zfs_key.so.1 create
```

Fichier l'administrateur utilisant cette stratégie lors de l'ajout d'utilisateurs. Notez que le chiffrement ne peut pas être ajouté à un système de fichiers. Le système de fichiers doit être créé avec le chiffrement activé. Pour plus d'informations, reportez-vous à `zfs_encrypt`(1M).

L'administrateur crée un utilisateurs et assigne un mot de passe.

```
# useradd -K pam_policy=/opt/local_pam/unix-encrypt.conf jill
# passwd jill
New Password: xxxxxxxx
Re-enter new Password: xxxxxxxx
passwd: password successfully changed for jill
```

Ensuite, l'administrateur crée le répertoire personnel chiffré en vous connectant en tant que l'utilisateur.

```
# su - jill
Password: xxxxxxxx
Creating home directory with encryption=on.
Your login password will be used as the wrapping key.
Oracle Corporation      SunOS 5.11      11.2    July 2014

# logout
```

Pour les options au module de service ZFS, reportez-vous à la page de manuel `pam_zfs_key`(5).

Enfin, l'administrateur vérifie que le nouveau répertoire d'origine Oracle Home est codée d'un système de fichiers.

```
# mount -p | grep ~jill
rpool/export/home/jill - /export/home/jill zfs - no
rw,devices,setuid,nonbmand,exec,rstchown,xattr,atime
# zfs get encryption,keysource rpool/export/home/jill
NAME                   PROPERTY    VALUE              SOURCE
rpool/export/home/jill  encryption  on                 local
rpool/export/home/jill  keysource   passphrase,prompt  local
```

▼ Ajout d'un module PAM

Cette procédure indique comment ajouter, protéger et tester un nouveau module PAM. Pour de nouveaux modules il peut s'avérer nécessaire ou des stratégies de sécurité propres au site pour prendre en charge des applications tierces. Pour créer un module PAM, reportez-vous à la section Chapitre 3, “ Writing PAM Applications and Services ” du manuel “ Developer’s Guide to Oracle Solaris 11 Security ”.

Remarque - Vous devez installer une version 32 bits et une version 64 bits du module de service PAM.

Avant de commencer

Terminer “Planification d'une configuration PAM propre à votre site” à la page 21.

Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. **Installez puis protégez les deux versions du module de service PAM sur le disque.**

   Assurez-vous que la propriété et les autorisations protègent les fichiers de module avec l'appartenance `root` et les autorisations `444`.

```
# cd /opt/pam_modules
# ls -lR
.:
total 4
-r--r--r--   1 root     root        4570 Nov 27 12:34 pam_app1.so.1
drwxrwxrwx   2 root     root           3 Nov 27 12:38 sparcv9

./64:
total 1
-r--r--r--   1 root     root        4862 Nov 27 12:38 pam_app1.so.1
```

   Le module 32 bits se trouve dans le répertoire `/opt/pam_modules` et le module 64 bits se trouve dans le sous-répertoire `64`.

2. **Ajouter le module au fichier de configuration PAM approprié.**

   Dans l'exemple suivant, le module est valable pour une nouvelle application `app1`. Son nom de service est identique au nom de l'application. Créez un fiche de nom de service `app1` dans le répertoire `/etc/pam.d`. La première entrée du fichier permet au service `app1` d'être affecter à des utilisateurs individuels.

```
# cd /etc/pam.d
# pfedit app1
...
# PAM configuration
#
# app1 service
#
auth definitive         pam_user_policy.so.1
auth required           /opt/pam_modules/$ISA/pam.app1.so.1  debug
```

   Le jeton `$ISA` du chemin d'accès au module dirige la structure PAM vers la version appropriée 32 bits ou 64 bits de l'architecture du module de service pour l'application appelante. Pour les applications 32 bits, */a/b/*$ISA/*Module*.so devient donc */a/b/module*.so. Et pour les applications 64 bits, il devient */a/b/*64/*Module*.so. Dans cet exemple, vous avez installé le module de service

32 bits `pam.app1.so.1` dans le répertoire `/opt/pam_modules` et le module 64 bits dans le répertoire `/opt/pam_modules/64`.

Pour plus d'informations, reportez-vous aux pages de manuel `pfedit`(1M) et `pam.conf`(4).

Pour limiter la stratégie PAM `app1` aux utilisateurs sélectionnés, reportez-vous à Exemple 1-2, “Ajout d'un nouveau module à un fichier de stratégie PAM par utilisateur”.

3. **Votre nouveau service de test.**
   Connectez-vous directement à l'aide de `login` ou `ssh`. Ensuite, exécutez les commandes qui sont affectées par le nouveau module. Les utilisateurs autorisés de test et la personne à l'utilisation de affectées ne le sont pas n'ont pas l'autorisation des commandes. Pour obtenir une assistance de dépannage, reportez-vous à la section “Dépannage des erreurs de configuration PAM” à la page 31.

4. **Affecter la stratégie.**
   Reportez-vous à la section “Assignation d'une stratégie PAM modifiée” à la page 28.

Exemple 1-2 Ajout d'un nouveau module à un fichier de stratégie PAM par utilisateur

Dans cet exemple, le service `app1` n'est pas utilisé par tous les utilisateurs, de sorte que l'administrateur ajoute le service en tant que stratégie par utilisateur.

```
# cd /etc/pam.d
# cp app1 /opt/local_pam/app1-conf
# pfedit /opt/local_pam/app1-conf

## app1 service
##
app1 auth definitive         pam_user_policy.so.1
app1 auth required           /opt/pam_modules/$ISA/pam_app1.so.1  debug
```

L'administrateur supprime le fichier `app1` à partir du répertoire `pam.d`.

```
# rm /etc/pam.d/app1
```

L'administrateur ajoute ensuite la stratégie `app1-conf` à la stratégie PAM de l'administrateur système.

```
# rolemod -K pam_policy=/opt/local_pam/app1-conf sysadmin
```

Exemple 1-3 Définition d'une stratégie PAM par utilisateur à l'aide d'un profil de droits

Cet exemple utilise l'attribut de sécurité `pam_policy` pour permettre aux utilisateurs d'être authentifiés à partir de différents services de noms. Le fichier de stratégie PAM `any` est fourni dans le répertoire `/etc/security/pam_policy`. Les commentaires qu'il contient décrivent cette stratégie.

Ne modifiez pas des fichiers dans ce répertoire.

```
# profiles -p "PAM Per-User Policy of Any" \
'set desc="Profile which sets pam_policy=any";
set pam_policy=any; exit;'
```

Pour affecter ce profil de droits, reportez-vous à la section “Assignation d'une stratégie PAM modifiée” à la page 28.

▼ Assignation d'une stratégie PAM modifiée

Dans cette procédure, vous devez configurer une stratégie PAM autre que celui par défaut sur toutes les images du système. Une fois que tous les fichiers sont copiés, vous pouvez appliquer la nouvelle stratégie PAM ou la stratégie modifiée à des utilisateurs individuels ou à tous les utilisateurs.

Avant de commencer Vous avez modifié les fichiers de configuration et testez la qui répondent aux conditions spécifiques au PAM nouvelle stratégie.

Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. **Ajouter les fichiers PAM par défaut à toutes les images du système.**

   Vous devez ajouter tous les nouveaux modules PAM et des fichiers de configuration nouveaux et modifiés à toutes les images du système.

   a. **Ajoutez tout d'abord à chaque toute nouvelle image système modules PAM.**

      i. **Ajoutez le module PAM 32 bits au répertoire d'architecture appropriée.**

      ii. **Ajoutez le module PAM 64 bits au répertoire d'architecture appropriée.**

      Pour obtenir un exemple de configuration d'annuaire, reportez-vous à Étape 1 dans la section “Ajout d'un module PAM” à la page 25.

   b. **Ensuite, ajoutez de nouveaux fichiers dans chaque PAM l'image du système.**

      Par exemple, ajoutez le fichier `/opt/local_pam/ssh-telnet-conf` à chaque image de système.

   c. **Il vous suffit ensuite de copier toute modification apportée dans chaque système PAM image les fichiers de configuration.**

      Par exemple, copiez un fichier `/etc/pam.conf` modifié et n'importe quel `/etc/pam.d/`*service-name-files* modifié sur chaque image de système.

2. **Affecter une stratégie PAM autre que celui par défaut à tous les utilisateurs.**

   a. **Modifiez le fichier `policy.conf` de l'une des façons suivantes :**

   - **Ajoutez un fichier de configuration PAM au mot-clé `PAM_POLICY` dans le fichier `policy.conf`.**

     ```
     # pfedit /etc/security/policy.conf
     ...
     # PAM_POLICY=
     PAM_POLICY=/opt/local_pam/ssh-telnet-conf
     ...
     ```

   - **Ajoutez un profil de droits au mot-clé `PROFS_GRANTED` dans le fichier `policy.conf`.**

     Par exemple, affectez la stratégie PAM par utilisateur de tout profil de droits à partir de Exemple 1-3, “Définition d'une stratégie PAM par utilisateur à l'aide d'un profil de droits”.

     ```
     # pfedit /etc/security/policy.conf
     ...
     AUTHS_GRANTED=
     # PROFS_GRANTED=Basic Solaris User
     PROFS_GRANTED=PAM Per-User Policy of Any,Basic Solaris User
     ...
     ```

   b. **Copiez le fichier `policy.conf` modifié sur chaque image du système.**

3. **Pour affecter une stratégie PAM autre que celui par défaut à chaque utilisateur, vous pouvez affecter le privilège directement à un utilisateur ou à ajouter la stratégie à un profil de droits assigné à des utilisateurs.**

   - **Affectez le privilège directement PAM à des utilisateurs individuels.**

     ```
     # usermod -K pam_policy="/opt/local_pam/ssh-telnet-conf" jill
     ```

   - **Stratégie dans une inclure le profil de droits PAM et affectez le profil à des utilisateurs individuels.**

     Cet exemple utilise la stratégie PAM `ldap`.

     ```
     # profiles -p "PAM Per-User Policy of LDAP" \
     'set desc="Profile which sets pam_policy=ldap";
     set pam_policy=ldap; exit;'
     ```

     Affectez alors le profil de droits à un utilisateur.

     ```
     # usermod -P +"PAM Per-User Policy of LDAP" jill
     ```

Exemple 1-4 Restriction de la pile PAM `ktelnet` aux utilisateurs sélectionnés

L'administrateur souhaite autoriser un nombre limité d'utilisateurs la possibilité d'utiliser dans un domaine Kerberos `telnet`. Par conséquent, avant que le service `telnet` soit activé, l'administrateur modifie le fichier de configuration `ktelnet` par défaut, et place le fichier `ktelnet` par défaut dans le répertoire `pam_policy`.

Tout d'abord, l'administrateur configure un fichier par utilisateur `ktelnet`.

```
# cp /etc/pam.d/ktelnet /etc/security/pam_policy/ktelnet-conf
# pfedit /etc/security/pam_policy/ktelnet-conf
...
# Kerberized telnet service
#
ktelnet  auth required          pam_unix_cred.so.1
ktelnet  auth required          pam_krb5.so.1
```

L'administrateur protège le fichier avec les autorisations `444`.

```
# chmod 444 /etc/security/pam_policy/ktelnet-conf
# ls -l /etc/security/pam_policy/ktelnet-conf
-r--r--r--   1 root     root         228 Nov 27 15:04 ktelnet-conf
```

Puis, l'administrateur modifie le fichier `ktelnet` dans le répertoire `pam.d`.

- Pour effectuer un ajustement en fonction de la première entrée permet active l'affectation de l'utilisateur.
- La seconde entrée rejette l'utilisation de `ktelnet` sauf si `pam_policy=ktelnet` vous est affecté par l'administrateur.

```
# cp /etc/pam.d/ktelnet /etc/pam.d/ktelnet.orig
              # pfedit /etc/pam.d/ktelnet
...
# Denied Kerberized telnet service
#
auth definitive         pam_user_policy.so.1
auth required          pam_deny.so.1
```

L'administrateur teste la configuration avec un utilisateur privilégié, un utilisateur standard, et le rôle root. Une fois la configuration réussie, l'administrateur active le service`telnet` et affecte la stratégie par utilisateur aux administrateurs Kerberos.

```
# svcadm enable telnet
# rolemod -S ldap -K pam_policy=ktelnet-conf kerbadmin
```

L'administrateur copie les fichiers modifiés sur tous les serveurs Kerberos et active `telnet` sur ces serveurs.

▼ Journalisation de rapports d'erreur PAM

Avant de commencer Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Déterminez quelle instance de service `system-log` est en ligne.**

```
# svcs system-log
STATE          STIME    FMRI
disabled       13:11:55 svc:/system/system-log:rsyslog
online         13:13:27 svc:/system/system-log:default
```

2. **Configurez le fichier `syslog.conf` pour le niveau de connexion nécessaire.**

   Reportez-vous à la section `DESCRIPTION` de la page de manuel `syslog.conf`(4) pour plus d'inforamtions sur les niveaux de consignation. La plupart des erreurs PAM sont signalées par l'intermédiaire de l'utilitaire `LOG_AUTH`.

   Par exemple, créez un fichier pour la sortie de débogage.

```
# touch /var/adm/pam_debuglog
```

   Puis, ajoutez l'entrée `syslog.conf` pour envoyer la sortie de débogage vers ce fichier.

Remarque - Si l'instance de service `rsyslog` est en ligne, modifiez le fichier `rsyslog.conf`.

```
# pfedit /etc/syslog.conf
...
*.debug        /var/adm/pam_debuglog
...
```

3. **Actualisez les informations de configuration du service `system-log`.**

```
# svcadm refresh system-log:default
```

Remarque - Actualisez l'instance de service `system-log:rsyslog` si le service `rsyslog` est en ligne.

▼ Dépannage des erreurs de configuration PAM

Avant de commencer Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Pour chaque entrée PAM faisant l'objet d'un dépannage, ajoutez l'option `debug`.**

   Par exemple, les entrées suivantes dans le fichier `/etc/pam.d/cron` permettent de créer une sortie de débogage pour le service.

   ```
   account definitive      pam_user_policy.so.1     debug
   account required        pam_unix_account.so.1    debug
   ```

2. **Consignez les erreurs PAM au niveau approprié et actualisez le démon `syslog`.**

   Pour plus d'informations, reportez-vous à la section “Journalisation de rapports d'erreur PAM” à la page 31.

3. **Une archive endommagée s'il s'agit d'un problème de configuration PAM, procédez comme suit :**

   a. **Exécutez l'application à partir de la fenêtre de terminal et modifier un fichier de configuration PAM dans une autre fenêtre.**

   b. **Vérifiez que les erreurs sont en cours de correction, testez les modifications dans la fenêtre d'application.**

4. **Une archive endommagée s'il s'agit d'un problème de connexion PAM, configuration qui empêche une initialisation en mode monoutilisateur, puis corrigez le fichier, ré - initialiser, et exécutez le test.**

   - **Pour initialiser un système SPARC, tapez la commande suivante à l'invite PROM :**

     ```
     ok > boot -s
     ```

   - **Pour initialiser un système x86, ajoutez l'option-`s` à la ligne des options du noyau dans le menu GRUB.**

   Pour plus d'informations, reportez-vous aux pages de manuel `boot`(1M) et `grub`(5).

5. **Vérifiez que les erreurs corrigées.**

   Connectez-vous directement à l'aide de `login` ou `ssh`. Test, les utilisateurs dotés de privilèges que les utilisateurs standard, et les rôles peuvent utiliser les commandes affectées.

Référence de configuration PAM

Cette section fournit des détails supplémentaires sur l'empilage, y compris l'empilage PAM.

Fichiers de configuration PAM

Des applications de système, telles que `login` et `ssh`, qui utilisent la structure PAM sont configurées dans les fichiers de configuration PAM du répertoire `/etc/pam.d`. Le fichier `/etc/pam.conf` peut également être utilisé. Les modifications apportées à ces fichiers concernent tous les utilisateurs sur le système.

En outre, le répertoire `/etc/security/pam_policy` regroupe des fichiers de configuration PAM. Plusieurs services et de ces fichiers d'accompagnement sont conçues pour par l'affectation de l'utilisateur. Fichiers de ce répertoire ne doit pas être modifiée.

- `/etc/pam.d` **Répertoire –** Contient des fichiers de configuration PAM spécifiques au service, y compris le fichier à caractères génériques, `other`. Pour ajouter un service pour une application, ajoutez un seul fichier *service-name* qui est le nom de service utilisé par l'application. Le cas échéant, votre application peut utiliser la pile PAM dans le fichier `other`.

  Le service de fichiers du répertoire `/etc/pam.d` fournit la configuration par défaut dans la plupart des implémentations PAM. Ils sont préassemblés à l'aide du mécanisme IPS comme décrit dans la page de manuel `pkg`(5). Cette valeur par défaut simplifie l'interaction avec d'autres applications PAM multi-plates-formes . Pour plus d'informations, reportez-vous à la page de manuel `pam.conf`(4).
- `/etc/pam.conf` **Fichier –** Le fichier de configuration et de stratégie PAM hérité. Ce fichier, est concédé sous licence vide. Le mécanisme recommandé pour la configuration PAM est l'utilisation de fichiers dans le répertoire `/etc/pam.d`. Pour plus d'informations, reportez-vous à la page de manuel `pam.conf`(4).
- `/etc/security/pam_policy` **directory –** Contient les fichiers de stratégie PAM comportant les stratégies pour plusieurs services. Ces fichiers peuvent être attribuées à un particulier, à un groupe de personnes, ou à tous les utilisateurs, en fonction des besoins. Une telle affectation remplace les fichiers de configuration PAM du système dans `pam.conf` ou le répertoire`/etc/pam.d`. Ne modifiez pas ces fichiers. Pour ajouter un fichier par utilisateur, reportez-vous à la section “Création d'un fichier de configuration PAM propre à votre site” à la page 23. Pour plus d'informations sur les fichiers par utilisateur, reportez-vous à la page de manuel`pam_user_policy`(5).

L'administrateur de sécurité gère tous les fichiers de configuration PAM. Un ordre d'entrée incorrect, c'est-à-dire, une pile PAM incorrecte peut entraîner des effets secondaires imprévus. Par exemple, un fichier mal configuré risque de verrouiller les utilisateurs, de sorte que le mode monoutilisateur s'impose pour effectuer des réparations. Pour obtenir de l'aide, reportez-vous à la section “Superposition PAM” à la page 35 and “Dépannage des erreurs de configuration PAM” à la page 31.

Ordre de recherche de la configuration PAM

Application aux appels au configuré PAM pour la structure des services de recherche dans l'ordre suivant :

1. Le nom de service est recherché dans le fichier `/etc/pam.conf`.
2. Des services spécifiques sont utilisés dans le fichier `/etc/pam.d/`*service-name*.
3. Le nom de service `other` est vérifié dans le fichier`/etc/pam.conf`.
4. Le fichier `/etc/pam.d/other` est utilisé.

Cet ordre permet d'utiliser un fichier `/etc/pam.conf` existant avec les fichiers de configuration PAM par service dans le répertoire `/etc/pam.d` .

Syntaxe du fichier de configuration PAM

Le fichier`pam.conf` et les fichiers par utilisateur PAM utilisent une syntaxe propre à chaque fichier spécifique au service dans le répertoire `pam.d`.

- Les entrées dans le fichier `/etc/pam.conf` et dans les fichiers `/etc/security/pam_policy` se trouvent dans l'un des deux formats suivants :

  service-name module-type control-flag module-path module-options

  service-name module-type `include` *path-to-included-PAM-configuration*
- Les entrées dans les fichiers *service-name* dans le répertoire `/etc/pam.d` omettent le nom de service. Le nom du fichier fournit le nom de service.

  module-type control-flag module-path module-options

  module-type `include` *path-to-included-PAM-configuration*

Les éléments de syntaxe le fichier de configuration PAM sont les suivantes :

service-name

Le nom du service sensible à la casse, par exemple, `login` ou `ssh`. Une application peut utiliser différents noms pour les services offerts par l'application. Par exemple, lancer une recherche pour le terme `PAM` dans la page de manuel`sshd`(1M) pour les noms de service pour les différents services que le démon `sshd` fournit.

Le nom de service prédéfini "`other`" est le nom de service par défaut en l'absence de configuration de service spécifique.

module-type

Type de service, c'est-à-dire `auth`, `account`, `session` ou `password`.

control-flag

Rôle du module dans la détermination de la valeur de réussite ou d'échec pour le service. Les indicateurs de contrôle valides sont décrits dans “Superposition PAM” à la page 35.

module-path

Le chemin d'accès au module qui met le type de module. S'il n'est pas absolu, on suppose qu'il est relatif au chemin d'accès `/usr/lib/security/$ISA/`. La macro ou le jeton `$ISA` incite la structure PAM à regarder dans le répertoire spécifique à l'architecture du chemin d'accès du module.

module-options

Les options telles que `nowarn` et `debug` pouvant être transmises aux modules de service. Une page de manuel du module décrit les options de ce module.

path-to-included-PAM-configuration

Représente le chemin complet à un fichier de configuration PAM ou un nom de fichier au répertoire `/usr/lib/security`.

Superposition PAM

Lorsqu'une applicaiton appelle une des fonctions suivantes, la structure PAM lit les fichiers de configuration PAM afin de déterminer les modules qui mettent en oeuvre le nom de service PAM de cette application :

- `pam_authenticate`(3PAM)
- `pam_acct_mgmt`(3PAM)
- `pam_setcred`(3PAM)
- `pam_open_session`(3PAM)
- `pam_close_session`(3PAM)
- `pam_chauthtok`(3PAM)

Si les fichiers de configuration ne contiennent qu'un seul module, le résultat de ce module détermine celui de l'opération. Par exemple, l'opération d'authentification par défaut pour l'application `passwd` contient un module, `pam_passwd_auth.so.1`, dans le fichier `/etc/pam.d/passwd`.

```
auth required          pam_passwd_auth.so.1
```

Si, en revanche, plusieurs modules mettent en oeuvre un service, on parle de modules *empilés*, c'est- à-dire qu'une pile PAM existe pour ce nom de service. Par exemple, prenons les entrées dans un exemple de service `/etc/pam.d/login` :

```
auth definitive         pam_user_policy.so.1
auth requisite          pam_authtok_get.so.1
auth required           pam_unix_auth.so.1
auth required           pam_dhkeys.so.1
auth required           pam_unix_cred.so.1
auth required           pam_dial_auth.so.1
```

Ces entrées créent une pile `auth` pour le nom de service `login`. Pour déterminer le résultat de cette pile, les codes de résultat de chaque module requièrent un *processus d'intégration*.

Dans le processus d'intégration, les modules sont exécutés dans l'ordre du fichier. Chaque code de réussite ou d'échec est intégré dans le résultat global en fonction de l'indicateur de contrôle du module. L'indicateur de contrôle peut entraîner la fin anticipée de la pile. Par exemple, l'échec d'un module `requisite` ou `definitive` met fin à la pile. En cas d'absence d'échecs précédent, la réussite d'un module `sufficient`, `definitive` ou `binding` met également fin à la pile. Une fois la pile traitée, tous les résultats sont regroupés en un résultat global unique, qui est transmis à l'application. Pour une vue graphique du flux, reportez-vous à la seciton Figure 1-2, “Superposition PAM : effet des indicateurs de contrôle” et Figure 1-3, “Superposition PAM : détermination de la valeur intégrée”.

L'indicateur de contrôle précise le rôle joué par un module PAM pour déterminer la réussite ou l'échec. Les indicateurs de contrôle ont les effets suivants :

- **Liaison** : lorsque les exigences d'un module binding (liaison) sont satisfaites, la réussite est immédiatement renvoyée à l'application si aucun échec n'a été enregistré. Si ces conditions sont vérifiées, aucun autre module n'est exécuté.

  En cas d'échec, un échec required est enregistré et le traitement des modules se poursuit.
- **Définitif** : lorsque les exigences d'un module definitive (définitif) sont satisfaites, la réussite est immédiatement renvoyée à l'application si aucun échéc précédent n'a été enregistré.

  Si un échec précédent a été enregistré, cet échec est immédiatement renvoyé à l'application sans exécution supplémentaire de modules. Un échec entraîne le renvoi immédiat d'une erreur et l'arrêt de l'exécution des modules.
- **Include** : ajoute des lignes d'un autre fichier de configuration PAM à utiliser à ce stade de la pile PAM. Cet indicateur ne contrôle pas les comportements de réussite ni d'échec. Lorsqu'un nouveau fichier est lu, la pile PAM `include` est incrémentée. Au terme de la vérification de la pile dans le nouveau fichier, la valeur de pile `include` est décrémentée. Une fois la fin du fichier atteinte et la valeur de la pile PAM `include` définie sur `0`, le traitement de la pile prend fin. Le nombre maximum pour la pile PAM `include` est `32`.
- **Optional** : il n'est pas nécessaire que les exigences d'un module optional (facultatif) soient satisfaites pour que le service puisse être utilisé.

  En cas d'échec, un échec optional est enregistré.
- **Required** : les exigences d'un module required (requis) doivent être satisfaites pour que la pile réussisse. La réussite finale de la pile n'est renvoyée que si aucun module binding ou required n'a signalé d'échec.

  Un échec entraîne le renvoi d'une erreur après l'exécution des modules restants pour ce service.

- **Requisite** : les exigences d'un module requisite (indispensable) doivent être satisfaites pour que la pile réussisse. Tous les modules requisite dans la pile doivent renvoyer un résultat positif pour que la pile puisse renvoyer une réussite à l'application.

  Un échec entraîne le renvoi immédiat d'une erreur et l'arrêt de l'exécution des modules.
- **Sufficient** : si aucun échec required précédent n'a été enregistré, la réussite dans un module sufficient (suffisant) renvoie immédiatement un résultat positif sans aucune autre exécution de modules.

  En cas d'échec, un échec optional est enregistré.

Les deux diagrammes connectés suivants indiquent comment un résultat est déterminé lors du processus d'intégration.

- Le premier diagramme montre comment la réussite ou l'échec sont enregistrés pour chaque type d'indicateur de contrôle. Les résultats sont affichés dans le second diagramme.
- Le second diagramme indique comment la valeur intégrée est déterminée. Un échec facultatif et un échec requis renvoient un échec en retour, et une réussite renvoie une réussite. L'application détermine comment gérer ces codes de retour.

FIGURE 1-2 Superposition PAM : effet des indicateurs de contrôle

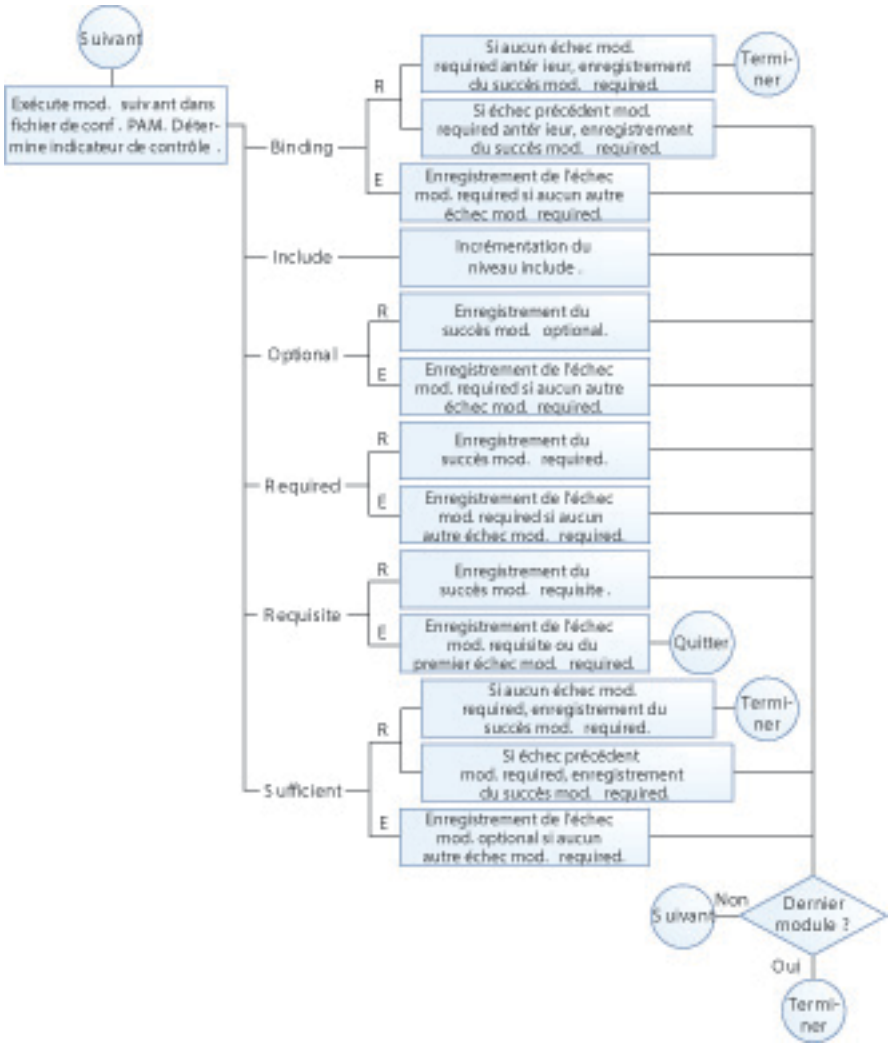
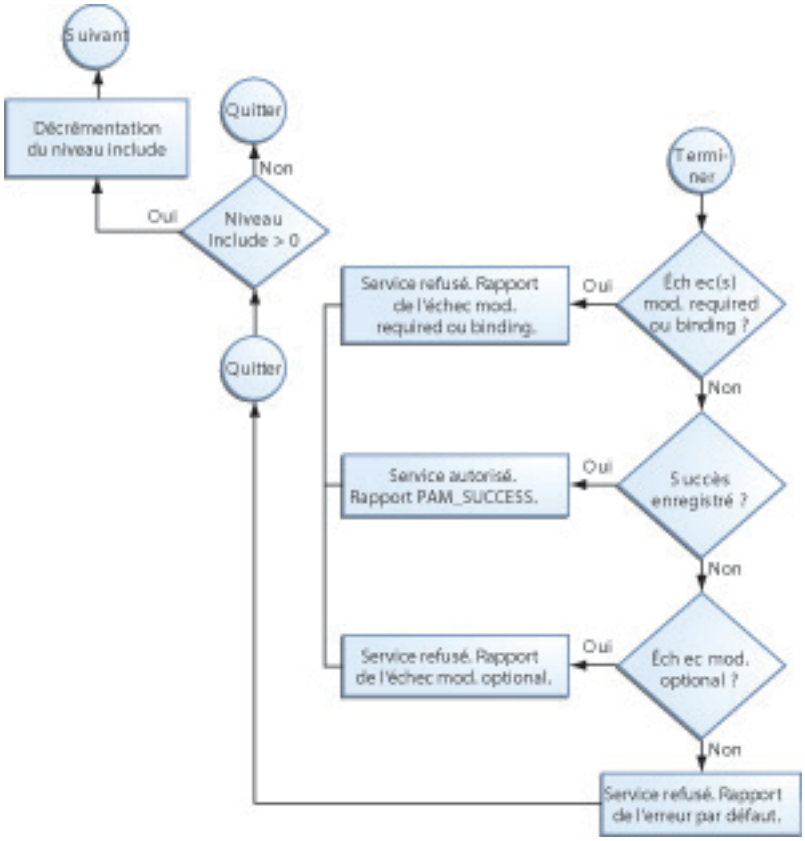


FIGURE 1-3 Superposition PAM : détermination de la valeur intégrée



Exemple de superposition PAM

L'exemple suivant indique les définitions par défaut de la gestion des authentifications dans un exemple de fichier `/etc/pam.d/other`. Ces définitions sont utilisées pour l'authentification si aucune définition d'authentification spécifique au service n'a été configurée.

```
##
# Default definitions for Authentication management
# Used when service name is not explicitly mentioned for authentication
#
auth definitive         pam_user_policy.so.1
auth requisite          pam_authtok_get.so.1
```

```
auth required           pam_dhkeys.so.1
auth required           pam_unix_auth.so.1
auth required           pam_unix_cred.so.1
```

Tout d'abord, la stratégie PAM de l'utilisateur est vérifiée à l'aide du module `pam_user_policy.so`. L'indicateur de contrôle `definitive` prévoit que si l'évaluation de la pile PAM configurée réussit, le résultat positif est renvoyé à l'application, car aucun autre module n'a été vérifié à ce stade. Si l'évaluation de la pile configuré n'est trouvée, une PAM code d'échec est renvoyé à l'application et aucune autre vérification n'est effectuée. Si aucune stratégie PAM par utilisateur est affecté à cet utilisateur, le module suivant est exécuté.

Si aucune stratégie PAM par utilisateur n'est affecté à cet utilisateur, le module `pam_authtok_get` est exécuté. L'indicateur de contrôle de ce module est défini sur `requisite`. Si `pam_authtok_get` échoue, le processus d'authentification se termine et l'échec est renvoyé à l'application.

Si `pam_authtok_get` n'échoue pas, les trois modules suivants sont exécutés. Ces modules sont configurés avec l'indicateur de contrôle `required` de sorte que le processus d'intégration se poursuit même si un échec individuel est renvoyé. Après l'exécution de `pam_unix_cred`, il ne reste plus aucun module. A ce stade, si tous les modules ont réussi, la réussite est renvoyée à l'application. Si `pam_dhkeys`, `pam_unix_auth` ou `pam_unix_cred` a renvoyé un échec, celui-ci est renvoyé à l'application.

Modules de service PAM

Cette section répertorie une sélection de modules de service PAM. Les modules sont répertoriés en fonction de leur page de manuel suivis d'une brève description emplacement et le moment où elles sont utilisées. Pour plus d'informations, reportez-vous à la page de manuel.

Pour une liste des modules de services PAM fournis par Oracle Solaris, reportez-vous à la section 5 des pages de manuel. De nouveaux modules sont ajoutés à intervalles réguliers. Par exemple, dans cette version, un certain nombre de modules sont ajoutés pour l'authentification avec les systèmes Windows. Votre site peut également ajouter des modules PAM émanant de tiers.

`pam_allow(5)` — Renvoie `PAM_SUCCESS` pour tous les appels. Reportez-vous également à la page de manuel `pam_deny(5)`.

`pam_authtok_check(5)` — Jeton valide le mot de passe pour modifier le mot de passe.

`pam_authtok_get(5)` — Les fonctions de celle fournit PAM pour la demande du mot de pile.

`pam_authtok_store(5)` — Met à jour le jeton du mot de passe pour `PAM_USER`.

pam_deny(5)	Type de module d'échec par défaut renvoie le code retour pour tous les appels. Reportez-vous également à la page de manuel pam_allow(5).
pam_dhkeys(5)	Fournit une fonctionnalité à deux services PAM : l'authentification RPC sécurisé et la gestion des jetons de l'authentification RPC sécurisé.
pam_krb5(5)	Fournit des fonctions pour vérifier l'identité d'un utilisateur Kerberos, et pour gérer le cache des informations d'identification Kerberos.
pam_krb5_migrate(5)	Permet de migrer PAM_USER vers le domaine Kerberos local du client.
pam_ldap(5)	Fournit la fonctionnalité pour l'authentification PAM et les piles de gestion de compte par le serveur d'annuaire LDAP configuré.
pam_list(5)	Fournit des fonctions pour valider le compte de l'utilisateur sur cet hôte. La validation est basée sur une liste d'utilisateurs et groupes réseau sur l'hôte.
pam_passwd_auth(5)	Ajoute la fonctionnalité d'authentification à la pile de mot de passe.
pam_pkcs11(5)	Permet à un utilisateur de se connecter à un système à l'aide d'un certificat X.509 et sa clé privée dédiée PKCS # 11 stockées dans un jeton.
pam_roles(5)	Vérifie qu'un utilisateur est autorisé à prendre un rôle et empêche la connexion directe par un rôle.
pam_smb_passwd(5)	Prend en charge la modification et l'ajout de mots de passe SMB pour les utilisateurs Oracle Solaris locaux. Reportez-vous également à la page de manuel smb(4).
pam_smbfs_login(5)	Synchronise les mots de passe entre Oracle Solaris et leurs serveurs CIFS/SMB.
pam_tsol_account(5)	Vérifie les limitations de compte Trusted Extensions liées aux étiquettes.
pam_tty_tickets(5)	Fournit un mécanisme permettant de contrôler un ticket créé par une authentification réussie antérieure.
pam_unix_account(5)	Fournit des fonctions pour valider le fait que le compte de l'utilisateur n'est pas verrouillé ou n'a pas expiré et que le mot de passe de l'utilisateur n'a pas besoin d'être modifié. Comprend les vérifications de access_times et access_tz.
pam_unix_auth(5)	Fournit des fonctions pour vérifier que le mot de passe est valide pour PAM_USER.

pam_unix_cred(5)	Fournit les fonctions qui permettent de définir les informations d'identification de l'utilisateur. Permet à la fonctionnalité d'authentification d'être remplacée indépendamment de la fonctionnalité des informations d'identification.
pam_unix_session(5)	Permet d'ouvrir et de fermer une session et également de mettre à jour le fichier `/var/adm/lastlog`.
pam_user_policy(5)	Appelle une configuration PAM propre à l'utilisateur.
pam_zfs_key(5)	Fournit des fonctions ZFS pour charger et modifier la phrase secrète de chiffrement pour un répertoire personnel chiffré de l'utilisateur.

♦ ♦ ♦ CHAPITRE 2

A propos du service Kerberos

Ce chapitre présente le service Kerberos. Le présent chapitre contient les informations suivantes :

Description du service Kerberos

Le *service Kerberos* est une architecture client-serveur qui fournit des transactions sécurisées sur les réseaux. Le service assure l'authentification fiable des utilisateurs, ainsi que l'intégrité et la confidentialité. L'*authentification* garantit que l'identité de l'expéditeur et du destinataire d'une transaction réseau sont toutes deux réelles. Le service peut également vérifier la validité des données transmises (*intégrité*) et le chiffrement des données lors de la transmission (*confidentialité*). A l'aide du service Kerberos, vous pouvez vous connecter à d'autres machines, exécuter des commandes, échanger des données et transférer des fichiers en toute sécurité. En outre, ce service offre des services d'*autorisation*, ce qui permet aux administrateurs de limiter l'accès aux services et aux machines. Par ailleurs, en tant qu'utilisateur Kerberos, vous pouvez réguler l'accès d'autres personnes à votre compte.

Le service Kerberos est un système à *connexion unique*, ce qui signifie que vous ne devez vous authentifier auprès du service qu'une fois par session. Toutes les transactions ultérieures au cours de la session sont automatiquement protégées. Une fois authentifié par le service, vous n'avez plus besoin de vous authentifier à chaque utilisation d'une commande basée sur Kerberos, comme `ftp` ou `ssh`, ou pour accéder à des données sur un système de fichiers NFS. Par conséquent, vous n'avez pas à envoyer votre mot de passe sur le réseau, où il peut être intercepté à chaque fois que vous utilisez ces services.

Le service Kerberos dans Oracle Solaris est basé sur le protocole d'authentification de réseau Kerberos V5 développé au Massachusetts Institute of Technology (MIT). Si vous avez utilisé le produit Kerberos V5, vous trouverez donc la version Oracle Solaris très familière. Le protocole Kerberos V5 étant une norme *de facto* de l'industrie en matière de sécurité de réseau, la version Oracle Solaris permet des transactions sécurisées sur des réseaux hétérogènes. En outre, le service assure l'authentification et la sécurité entre les domaines et au sein d'un domaine unique.

Le service Kerberos offre une plus grande flexibilité dans l'exécution des applications Oracle Solaris. Vous pouvez configurer ce service pour autoriser les demandes de services réseau Kerberos et non Kerberos, notamment les services NFS et `ftp`. Par conséquent, les applications actuelles fonctionnent toujours, même si elles sont en cours d'exécution sur des systèmes sur lesquels le service Kerberos n'est pas activé. Bien entendu, vous pouvez également configurer le service Kerberos pour n'autoriser que les demandes de réseau Kerberos.

Le service Kerberos fournit un mécanisme de sécurité permettant d'utiliser Kerberos pour l'authentification, l'intégrité et la confidentialité lors de l'utilisation d'applications ayant recours à GSS-API (API générique de services de sécurité). Toutefois, il n'est pas nécessaire que les applications restent dédiées au service Kerberos si d'autres mécanismes de sécurité sont développés. Etant donné que le service est conçu pour s'intégrer de façon modulaire à GSS-API, les applications qui l'utilisent peuvent sélectionner le mécanisme de sécurité le plus adapté à leurs besoins.

Fonctionnement du service Kerberos

La présente section donne un aperçu de l'authentification Kerberos. Pour obtenir une description plus détaillée, reportez-vous à la section “Accès à des services à l'aide des informations d'identification et de connexion Kerberos” à la page 56.

Du point de vue de l'utilisateur, le service Kerberos est pratiquement invisible une fois la session Kerberos démarrée. Les commandes telles que `ssh` ou `ftp` fonctionnent de la même façon. L'initialisation d'une session Kerberos n'implique souvent rien de plus que la connexion et l'indication du mot de passe Kerberos.

Le système Kerberos est basé sur le concept de *ticket*. Un ticket est un ensemble d'informations électroniques qui identifient un utilisateur ou un service tel que le service NFS. Tout comme votre permis de conduire vous identifie et indique les privilèges de conduite dont vous disposez, un ticket vous identifie ainsi que vos privilèges d'accès au réseau. Lorsque vous effectuez une transaction basée sur Kerberos (par exemple, si vous demandez un fichier monté NFS), vous envoyez de façon transparente une demande de ticket à un *Key Distribution Center, Centre de distribution des clés* ou KDC. Le KDC accède à une base de données pour authentifier votre identité et renvoie un ticket qui vous autorise à accéder à l'NFS serveur. "Transparently" signifie que vous n'avez pas à demander explicitement un ticket. La demande s'effectue lorsque vous tentez d'accéder à le serveur. Puisque seuls des clients authentifiés peuvent obtenir un ticket

d'un service particulier, un autre client ne peut pas accéder au serveur NFS sous une fausse identité.

Des tickets disposent de certains attributs qui leur sont associés. Par exemple, un ticket peut être *transmissible*, ce qui signifie qu'il peut être utilisé sur un autre ordinateur sans nouveau processus d'authentification. Un ticket peut également être *postdaté*, ce qui signifie qu'il n'est pas valide avant une heure spécifiée. Les utilisations des tickets, par exemple, pour spécifier quels utilisateurs sont autorisés à obtenir quels types de ticket, sont définies par des *stratégies*. Les stratégies sont déterminées lors de l'installation ou de l'administration de Kerberos.

Remarque - Vous verrez fréquemment les termes *informations d'identification* et*ticket*. Dans l'univers Kerberos, ils sont souvent utilisés de façon interchangeable. D'un point de vue technique, cependant, les informations d'identification correspondent à un ticket et à la *clé de session* pour cette session. Cette différence est expliquée plus en détails dans la section “Accès à des services à l'aide des informations d'identification et de connexion Kerberos” à la page 56.

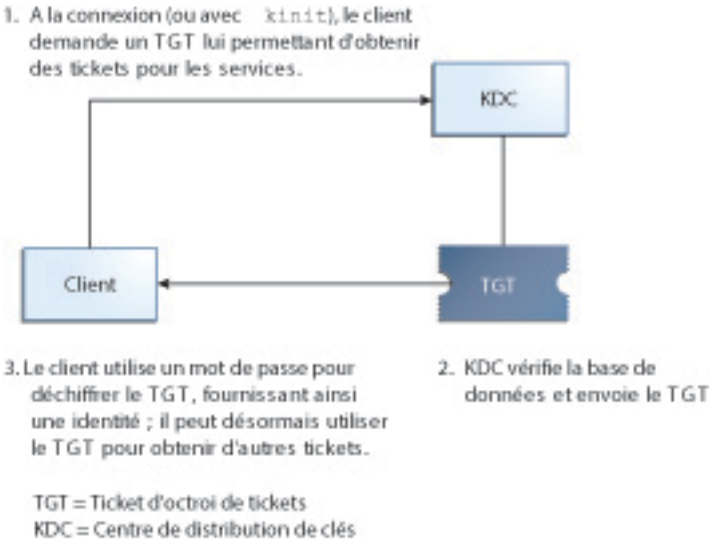
Les sections suivantes expliquent davantage les processus d'authentification Kerberos.

Authentification initiale : le TGT

L'authentification Kerberos comprend deux phases : une authentification initiale qui autorise toutes les authentifications, puis les authentifications suivantes.

La figure ci-dessous illustre la manière dont l'authentification initiale a lieu.

FIGURE 2-1 Authentification Kerberos initiale pour une session



1. Un client (un utilisateur, ou un service comme NFS) commence une session Kerberos en demandant un *TGT* (ticket d'octroi de tickets) dans le KDC (centre de distribution de clés). Cette demande est souvent effectuée automatiquement à la connexion.

   Un TGT est nécessaire pour obtenir d'autres tickets pour des services spécifiques. Considérez le TGT comme étant semblable à un passeport. Comme un passeport, le TGT vous identifie et vous permet d'obtenir de nombreux "visas" (tickets) plutôt que d'accorder d'accès, qui vers ou depuis des pays étrangers vous permettent d'accéder aux machines distantes ou des services réseau. Comme les passeports et les visas, le TGT et les autres différents tickets ont une durée de vie limitée. La seule différence réside dans le fait que les commandes Kerberos voient que vous avez un passeport et qu'elles obtiennent les visas pour vous. Vous n'avez pas à effectuer les transactions vous-même.

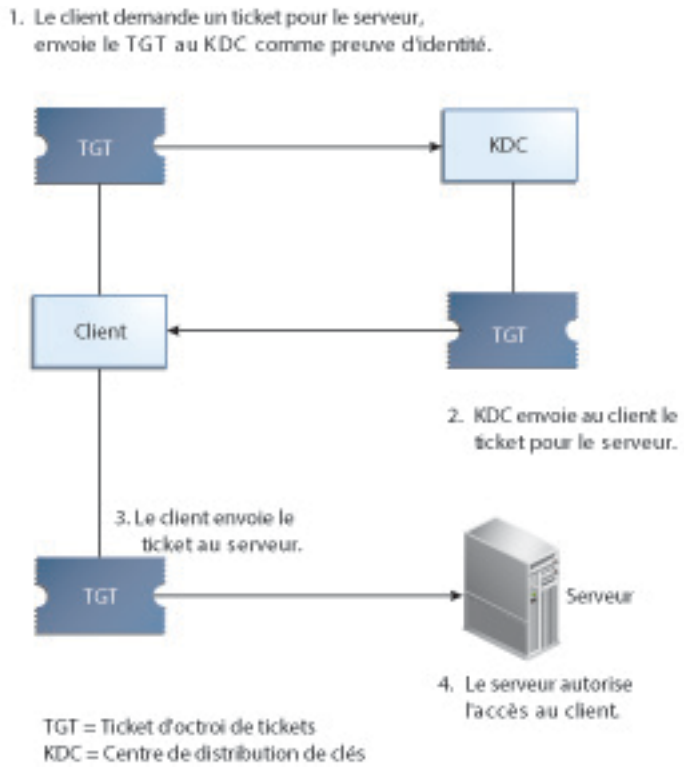
   Une autre analogie pour le TGT est celle du passe de ski de trois jours valable dans quatre différentes stations. Vous pouvez montrer le passe dans la station de votre choix et vous recevez un ticket pour les pistes correspondantes, tant que le passe n'a pas expiré. Une fois que vous avez accès aux pistes, vous pouvez skier autant que vous le voulez dans cette station. Si vous passez à une autre station le jour suivant, vous devez montrer votre passe à nouveau, et vous obtenez l'accès aux pistes de la nouvelle station. La différence réside dans le fait que les commandes Kerberos voient que vous avez un passe de ski et obtiennent l'accès aux pistes pour vous de sorte que vous n'avez pas à effectuer les opérations vous-même .
2. Le KDC crée un TGT qu'il renvoie, sous forme chiffrée, au client. Le client déchiffre le TGT en utilisant le mot de passe du client.

3. Maintenant en possession d'un TGT en cours de validité, le client peut demander des tickets pour toutes sortes d'opérations réseau, telles que `nfs` ou `ssh`, aussi longtemps que le TGT est valide. Ce ticket dure généralement quelques heures. A chaque fois que le client effectue une opération réseau unique, il demande un ticket pour cette opération au KDC.

Authentifications Kerberos suivantes

Une fois que le client a reçu l'authentification initiale, chaque nouvelle authentification suit le modèle indiqué dans la figure ci-dessous.

FIGURE 2-2 Obtention de l'accès à un service à l'aide de l'authentification Kerberos



1. Le client demande un ticket au KDC pour un service particulier, par exemple, pour se connecter à distance à une autre machine, en envoyant au KDC son TGT comme preuve d'identité.
2. Le KDC envoie le ticket pour le service spécifique au client.

Supposons qu'un utilisateur `joe` demande l'accès à un système de fichiers NFS qui a été partagé avec `krb5`, l'authentification est nécessaire. Puisque `jdoe` est déjà authentifié (c'est-à-dire que `jdoe` possède déjà un TGT), lorsque `jdoe` tente d'accéder aux fichiers, le système client NFS obtnient automatiquement et de façon transparente un ticket du KDC pour le service NFS. Pour utiliser un autre service utilisant Kerberos, `jdoe` obtient un autre ticket, comme à l'étape 1.

3. Le client envoie le ticket au serveur.

   Lors de l'utilisation du service NFS, le client NFS envoie le ticket automatiquement et de manière transparente au service NFS pour le serveur NFS.

4. Le serveur autorise l'accès au client.

Bien que les étapes ci-dessous indiquent que la KDC jamais communique avec le serveur, le serveur ne permet peut s'inscrire auprès de la même manière que le KDC, le premier client. A des fins de simplification, cette section ont été omises.

Authetification Kerberos des tâches par lots

Les tâches par lots telles que cron, at et batch sont des processus à exécution retardée. Dans un environnement Kerberos, tous les processus, y compris les processus nécessitent les informations d'identification et de l'exécution différée. Cependant, des utilisateurs sont relativement courte durée les informations d'identification et de connexion. Les informations d'identification et de connexion par défaut, l'utilisateur et sont valides pour 8 heures en vigueur tant qu 'renouvelable une semaine. Ces heures sont destinée à introduire les clés prevents users from assigning themselves to and à des utilisateurs malveillants peuvent se produire de sensitive, mais risque d'empêcher le fois l'exécution de jobs à arbitraire.

Dans Oracle Solaris, les tâches par lots qui accèdent aux services Kerberos peuvent s'exécuter sans révéler la clé à long terme de l'utilisateur. Les informations d'identification et de la solution implique le stockage qui incluent le service Kerberos, le nom d'utilisateur et le nom d'hôte du client utilisateur dans un cache d'informations d'identification et de connexion par session. Un module PAM est utilisé pour authentifier la tâche par lots. Les services qui un hôte peut obtenir des tickets pour peuvent être stockés de manière centrale dans le serveur d'annuaire LDAP.

Pour plus d'informations, reportez-vous aux pages de manuel `pam_krb5_keytab`(5) et `pam_gss_s4u`(5) et à la section “Configuration de l'exécution retardée pour l'accès aux services Kerberos” à la page 127.

Kerberos, DNS et le service de noms

Le service Kerberos est compilé pour utiliser le DNS pour la résolution des noms d'hôte. Le service `nsswitch` n'est pas du tout consulté pour la résolution des noms d'hôte.

Composants Kerberos

Programmes réseau Kerberos, inclut des principaux domaines de gestion des identités,, les serveurs et les autres composants. Pour obtenir la liste des commandes et modules, reportez-vous à la section “Utilitaires Kerberos” à la page 52.

Programmes réseau Kerberos

Remarque - Dans cette version Oracle Solaris, toutes les commandes de connexion à distance à l'exception de `ssh` sont en phase d'abandon. Si vous souhaitez utiliser l'une des opérations suivantes en phase d'abandon pour atteindre un système plus ancien des commandes, vous pouvez le faire. Si vous souhaitez utiliser une commande sur ce système en phase d'abandon car la commande est utilisé dans un script hérité, vous devez activer le service SMF pour cette commande, comme indiqué dans `svcadm enable login:rlogin` Vous pouvez également altérer les scripts pour utiliser la commande `ssh`. De même, pour un système plus ancien pour atteindre cette système à l'aide d'une commande, le service en phase d'abandon pour la commande doit être activée sur ce système.

Des commandes en phase d'abandon telles que `telnet` peuvent nécessiter des clés de chiffrement faibles. Ces clés ne sont pas autorisées par défaut dans le fichier `krb5.conf`. Pour plus d'informations, reportez-vous à la section“Types de chiffrement pris en charge dans Kerberos” à la page 65 et à la page de manuel `krb5.conf`(4).

Pour plus d'informations, reportez-vous à la section “ Contrôle de l’accès aux ressources de la machine ” du manuel “ Sécurisation des systèmes et des périphériques connectés dans Oracle Solaris 11.2 ”. Reportez-vous également aux pages de manuel qui sont répertoriées dans la section “Commandes utilisateur Kerberos” à la page 176.

(The Kerberos basée sur les commandes utilisant Kerberos à la disposition des utilisateurs sont) les éléments suivants :

- `ftp`
- `rcp`, `rlogin`, `rsh`
- `ssh`, `scp`, `sftp`
- `telnet`

Ces applications sont les mêmes que l'application Oracle Solaris du même nom. Cependant, elles ont été étendues pour utiliser les principaux Kerberos pour authentifier les transactions, afin que vous disposiez d'une sécurité Kerberos. Pour plus d'informations sur les principaux, reportez-vous à la section “Principaux Kerberos” à la page 50.

Ces commandes sont traitées plus en détail à la section “Commandes utilisateur Kerberos” à la page 176.

Principaux Kerberos

Un client dans le service Kerberos est identifié par son *principal*. Un principal est une identité unique à laquelle le KDC peut affecter les tickets. Un principal peut être un utilisateur, tel que `joe`, ou un service, tel que `nfs`.

Par convention, un nom de principal est divisé en trois composants : le *primaire*, l'*instance* et le *domaine*. Un principal Kerberos type peut être, par exemple, `jdoe/admin@CORP.EXAMPLE.COM`. Dans cet exemple :

- `jdoe` est le principal. Le primaire peut être un nom d'utilisateur, comme ici, ou un service, comme `nfs`. Le primaire peut également être le mot `host`, ce qui signifie que ce principal est un principal de service qui est configuré pour fournir divers services réseau, `ftp`, `scp`, `ssh`, etc.
- `admin` est l'instance. Une instance est facultative dans le cas de principaux d'utilisateur, mais est nécessaire pour les principaux de service. Par exemple, si l'utilisateur `jdoe` agit parfois en tant qu'administrateur système, le principal `jdoe/admin` est capable de distinguer l'identité de l'administrateur de celle de l'utilisateur. De même, si `joe` a des comptes sur deux hôtes différents, les comptes peuvent utiliser deux noms de principal avec différentes instances, par exemple, `jdoe/denver.example.com` et `jdoe/boston.example.com`. Notez que le service Kerberos traite `jdoe` et `jdoe/admin` comme deux principaux totalement différents.

  Dans le cas d'un principal de service, l'instance est le nom d'hôte complet. `bigmachine.corp.example.com` est un exemple d'une telle instance. Le principal ou l'instance pour cet exemple pourrait être `ftp/bigmachine.corp.example.com` ou `host/bigmachine.corp.example.com`.
- `CORP.EXAMPLE.COM` est le domaine Kerberos. Les domaines sont abordés dans “Domaines Kerberos” à la page 50.

Les éléments suivants sont tous les noms de principaux valides :

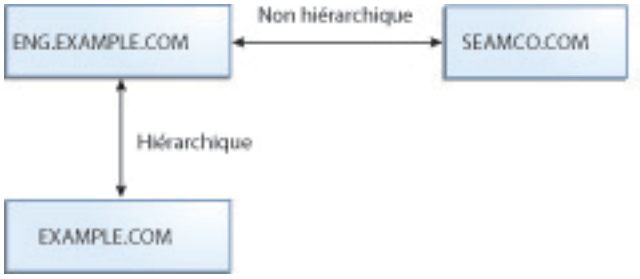
- `jdoe`
- `jdoe/admin`
- `jdoe/admin@CORP.EXAMPLE.COM`
- `nfs/host.corp.example.com@CORP.EXAMPLE.COM`
- `host/corp.example.com@CORP.EXAMPLE.COM`

Domaines Kerberos

Un *domaine* est un réseau logique qui définit un groupe de systèmes sous le même *KDC maître*. La Figure 2-3, “Domaines Kerberos” montre de quelle manière les domaines peuvent

se rapporter les uns aux autres. Certains domaines sont hiérarchiques, où un domaine est un surensemble de l'autre domaine. Dans le cas contraire, les domaines sont non hiérarchiques (ou "directs") et le mappage entre les deux domaines doit être défini. L'*authentification inter-domaine* active l'authentification au sein des domaines. Chaque domaine a seulement besoin de disposer d'une entrée de principal pour l'autre domaine dans son KDC.

FIGURE 2-3 Domaines Kerberos



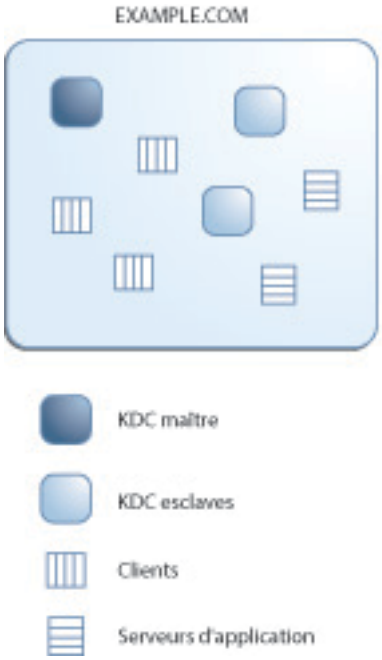
Serveurs Kerberos

Chaque domaine doit inclure un serveur qui gère la copie principale de la base de données du principal. Ce serveur est appelé le *serveur KDC maître*. En outre, chaque domaine doit contenir au moins un *serveur KDC esclave*, qui contient une copie de la base de données du principal. Le serveur KDC maître et le serveur KDC esclave créent tous deux des tickets utilisés pour établir l'authentification.

Le domaine peut également inclure un *serveur d'application* Kerberos. Ce serveur permet d'accéder aux services utilisant Kerberos tels que `ftp`, `ssh` et NFS.

La figure suivante illustre le contenu possible d'un domaine.

FIGURE 2-4 Domaine Kerberos typique



Utilitaires Kerberos

Similaire à la distribution par MIT du produit Kerberos V5, le service Kerberos dans la version Oracle Solaris comprend les éléments suivants :

- KDC (Key Distribution Center, centre de distribution de clés) :
  - Démon d'administration de base de données Kerberos : `kadmind`.
  - Démon de traitement des tickets Kerberos : `krb5kdc`.
  - Programmes d'administration de base de données : `kadmin` (maître uniquement), `kadmin.local` et `kdb5_util`.
  - Logiciel de propagation de base de données : `kprop` (esclave uniquement) et `kpropd`.
- Programmes utilisateur de gestion des informations d'identification : `kinit`, `klist` et `kdestroy`.
- Programme utilisateur de modification du mot de passe Kerberos : `kpasswd`.
- Applications réseau : `ftp`, `rcp`, `rlogin`, `rsh`, `scp`, `sftp`, `ssh` et `telnet`.

- Démons d'application à distance : `ftpd`, `rlogind`, `rshd`, `sshd` et `telnetd`.
- Utilitaire d'administration de keytab : `ktutil`.
- GSS-API (API de service de sécurité générique) : permet aux applications d'utiliser plusieurs mécanismes de sécurité sans avoir à recompiler l'application à chaque fois qu'un nouveau mécanisme est ajouté. GSS-API utilise les interfaces standard permettant aux applications d'être portables pour de nombreux systèmes d'exploitation. GSS-API permet aux applications d'inclure les services de sécurité d'intégrité et de confidentialité, ainsi que l'authentification. Les commandes `ftp` et `ssh` utilisent GSS-API.
- RPCSEC_GSS API (Application Programming Interface, Interface de programmation d'application) : permet d'activer les services NFS afin d'utiliser l'authentification Kerberos. Fournit des services de sécurité RPCSEC_GSS API la qui sont indépendants des mécanismes utilisés. RPCSEC_GSS est installé sur la couche GSS-API. N'importe quel mécanisme de sécurité enfichable basé sur GSS_API peut être utilisé par des applications utilisant RPCSEC_GSS.

De plus, le service Kerberos dans Oracle Solaris inclut les éléments suivants :

- Modules de service Kerberos V5 pour PAM : assurent l'authentification, la gestion de compte, la gestion de session et la gestion des mots de passe pour le service Kerberos. Les modules que l'authentification Kerberos soit transparente pour l'utilisateur.
- Piles PAM par utilisateur Kerberos V5 : fournit des fichiers de configuration PAM pour différents scénarios dans le répertoire `/etc/security/pam_policy`.
- Modules de noyau : fournissent des implémentations basées sur le noyau du service Kerberos pour une utilisation par le service NFS, ce qui améliore considérablement les performances.
- GUI d'administration Kerberos (`gkadmin`) : permet d'administrer les principaux et les stratégies des principaux dans une interface graphique basée sur la technologie Java™ en tant qu'alternative à la commande `kadmin`.

Pour plus d'informations, reportez-vous au Chapitre 7, Référence de service Kerberos.

Services de sécurité Kerberos

En plus d'assurer l'authentification sécurisée des utilisateurs, le service Kerberos fournit deux services de sécurité :

- **Intégrité :** tout comme l'authentification permet de s'assurer que les clients sur un réseau sont bien ceux qu'ils prétendent être, l'intégrité des données permet de s'assurer que les données qu'ils envoient sont valides et qu'elles n'ont pas été falsifiées pendant le transport. L'intégrité est assurée par l'intermédiaire de la somme de contrôle des données. L'intégrité inclut également l'authentification de l'utilisateur.
- **Confidentialité :** la confidentialité renforce la sécurité. La confidentialité n'inclut pas seulement la vérification de l'intégrité des données transmises, mais elle chiffre en outre les

données avant leur transmission, afin de les protéger contre les écoutes électroniques. La confidentialité permet également d'authentifier les utilisateurs.

Types de chiffrement Kerberos

Les types de chiffrement identifient les algorithmes cryptographiques et le mode à utiliser lors d'opérations cryptographiques. Pour obtenir une liste des types de chiffrement pris en charge, reportez-vous aux pages de manuel `krb5.conf`(4) et `kdb5_util`(1M).

Lorsqu'un client demande un ticket au KDC, le KDC doit utiliser des clés dont le type de chiffrement est compatible à la fois avec le client et le serveur. Le protocole Kerberos permette au client de demander à ce que le KDC utilise certains types de chiffrement pour la partie de réponse du ticket du client. Le protocole n'autorise pas le serveur à spécifier des types de chiffrement au KDC.

Tenez compte des points suivants avant de modifier les types de chiffrement, procédez comme suit :

- Le KDC suppose que le premier type de clé/chiffrement associé à l'entrée du principal du serveur dans la base de données du principal est pris en charge par le serveur.
- Sur le KDC, vous devez vous assurer que les clés qui sont générées pour le principal sont compatibles avec les systèmes afin d'authentifier l'identité utilisateur (principal). Par défaut, la commande `kadmin` crée des clés pour tous les types de chiffrement pris en charge. Si les systèmes sur lesquels le principal est utilisé ne prennent pas en charge cette configuration par défaut des types de chiffrement, vous devez restreindre les types de chiffrement lors de la création d'un principal. Les deux méthodes recommandées de limitation des types de chiffrement sont l'utilisation de l'indicateur `-e` dans `kadmin addprinc` ou la définition du paramètre `supported_enctypes` dans le fichier `kdc.conf` à ce sous-ensemble. Utilisez le paramètre `supported_enctypes` parameter lorsque la plupart des systèmes d'un domaine Kerberos prennent en charge un sous-ensemble de l'ensemble par défaut des types de chiffrement. Le paramètre `supported_enctypes` spécifie l'ensemble par défaut des types de chiffrement `kadmin addprinc` utilisés lorsqu'il crée un principal pour un domaine particulier.
- Au moment de déterminer les types de chiffrement pris en charge par un système, pensez à la fois à la version de Kerberos exécutée sur le système et aux algorithmes cryptographiques pris en charge par l'application du serveur pour lequel un principal de serveur est en cours de création. Par exemple, lors de la création d'un principal de service `nfs/hostname`, il est conseillé de limiter les types de chiffrement aux types pris en charge par le serveur NFS de l'hôte.
- Le paramètre `master_key_enctype` dans le fichier `kdc.conf` peut être utilisé pour contrôler le type de chiffrement de la clé principale qui chiffre les entrées dans la base de données du principal. N'utilisez pas ce paramètre si la base de données du principal KDC a déjà été

créée. Le paramètre `master_key_enctype` peut être utilisé au moment de la création de la base de données afin de faire passer le type de chiffrement de la clé principale par défaut de `aes256-cts-hmac-sha1-96` à un type de chiffrement différent. Assurez-vous que tous les KDC esclaves prennent en charge le type de chiffrement choisi et qu'ils ont la même entrée `master_key_enctype` dans leur fichier `kdc.conf` lors de la configuration des KDC esclaves. Assurez-vous également que le paramètre `master_key_enctype` est défini sur l'un des types de chiffrement dans `supported_enctypes`, si `supported_enctypes` est défini dans `kdc.conf`. Si l'une ou l'autre de ces situations n'est pas gérée correctement, le KDC maître peut ne pas être en mesure de travailler avec les KDC esclaves.

- Sur le client, vous pouvez contrôler les requêtes de type de chiffrement à partir du KDC à l'aide des paramètres dans le fichier `krb5.conf`. Le paramètre `default_tkt_enctypes` spécifie les types de chiffrement que le client est disposé à utiliser lorsqu'il demande un ticket d'octroi de tickets (TGT) au KDC. Le TGT est utilisé par le client pour acquérir d'autres tickets de serveur d'une manière plus efficace. L'effet du paramètre `default_tkt_enctypes` est de donner au client un contrôle sur les types de chiffrement utilisés pour protéger les communications entre le client et le KDC, lorsque le client demande un ticket de serveur via TGT (aussi appelé demande TGS). Notez que les types de chiffrement spécifiés dans `default_tkt_enctypes` doivent correspondre à au moins un des types de chiffrement de clé principal dans la base de données du principal stockée sur le KDC. Dans le cas contraire, la demande TGS échoue. Dans la plupart des cas, vous ne devez pas définir `default_tkt_enctypes` car ce paramètre peut être une source de problèmes d'interopérabilité. Par défaut, le code client demande tous les types de chiffrement pris en charge et le KDC choisit les types de chiffrement en fonction des clés que le KDC trouve dans la base de données du principal.
- Le paramètre `default_tgs_enctypes` restreint les types de chiffrement demandés par le client dans ses demandes TGS, qui sont utilisées pour l'acquisition de tickets de serveur. Ce paramètre restreint également les types de chiffrement utilisés par le KDC lors de la création de la clé de session partagée par le client et le serveur. Par exemple, si un client souhaite utiliser uniquement le chiffrement 3DES pour un NFS sécurisé, vous devez définir `default_tgs_enctypes` = `des3-cbc-sha1`. Assurez-vous que les principaux du client et du serveur ont une clé `des-3-cbc-sha1` dans la base de données du principal. Comme avec `default_tkt_enctype`, vous ne devez pas, dans la plupart des cas, définir ce paramètre, car il peut provoquer des problèmes d'interopérabilité si les informations d'identification ne sont pas configurées correctement sur le KDC et le serveur.
- Sur le serveur, vous pouvez contrôler les types de chiffrement qu'il accepte avec le paramètre `permitted_enctypes` dans le fichier `kdc.conf`. De plus, vous pouvez spécifier les types de chiffrement utilisés lors de la création d'entrées keytab. Evitez d'utiliser l'une de ces méthodes pour contrôler les types de chiffrement et laisser plutôt le KDC déterminer les types de chiffrement à utiliser, car le KDC ne communique pas avec l'application de serveur pour déterminer la clé ou le type de chiffrement à utiliser.

Algorithmes FIPS 140 et types de chiffrement Kerberos

Vous pouvez configurer Kerberos pour s'exécuter en Mode FIPS 140 dans Oracle Solaris. Si votre domaine contient des applications ou des systèmes qui ne sont pas conformes FIPS 140, le domaine ne pourra pas s'exécuter en Mode FIPS 140.

Lorsqu'il s'exécute en Mode FIPS 140, on dit de Kerberos qu'il est un *consommateur* du *fournisseur* FIPS 140. Le fournisseur dans Oracle Solaris est la structure cryptographique. Le seul type de chiffrement Kerberos validé-FIPS 140 pour la structure cryptographique est `des3-cbc-sha1`. Ce n'est pas le type de chiffrement par défaut. Pour obtenir des instructions, reportez-vous à “Configuration de Kerberos afin qu'il s'exécute en mode FIPS 140” à la page 77.

Remarque - Si l'exigence stricte est de n'utiliser qu'une cryptographie validée FIPS 140-2, vous devez exécuter la version Oracle Solaris 11.1 SRU 5.5 ou la version Oracle Solaris 11.1 SRU 3. Oracle a obtenu la validation FIPS 140-2 relative à la structure cryptographique dans ces deux versions spécifiques. Oracle Solaris 11.2 s'appuie sur cette base validée pour apporter des améliorations logicielles concernant la performance, les fonctionnalités et la fiabilité. Configurez Oracle Solaris 11.2 chaque fois que possible en mode FIPS 140-2 pour bénéficier de ces améliorations.

Accès à des services à l'aide des informations d'identification et de connexion Kerberos

Les services distants vous permettre d'accéder si vous pouvez fournir un ticket apportant la preuve de votre identité et la clé de session correspondante. La clé de session contient des informations spécifiques à l'utilisateur et au service en cours d'accès. Un ticket et une clé de session sont créés par le KDC pour tous les utilisateurs lors de leur première connexion. Le ticket et la clé de session correspondante constituent des informations d'identification. Lors de l'utilisation de plusieurs services réseau, un utilisateur peut obtenir de nombreuses informations d'identification. L'utilisateur doit disposer d'informations d'identification pour chaque service s'exécutant sur un serveur particulier. Par exemple, l'accès au service `nfs` sur un serveur nommé `boston` nécessite des informations d'identification. L'accès au service `nfs` sur un autre serveur nécessite des informations d'identification séparées.

Pour accéder à un service spécifique sur un serveur spécifique, l'utilisateur doit obtenir deux types d'informations d'identification. Les premières informations d'identificaiton correspondent au ticket d'octroi de tickets (TGT). Une fois que le service d'octroi de ticket a déchiffré ces informations d'identification, le service crée d'autres informations d'identification pour le serveur auquel l'utilisateur demande l'accès. Ces informations d'identification peuvent ensuite être utilisées pour demander l'accès à ce service sur le serveur. Une fois que le serveur a déchiffré ces informations d'identification, l'utilisateur obtient l'accès.

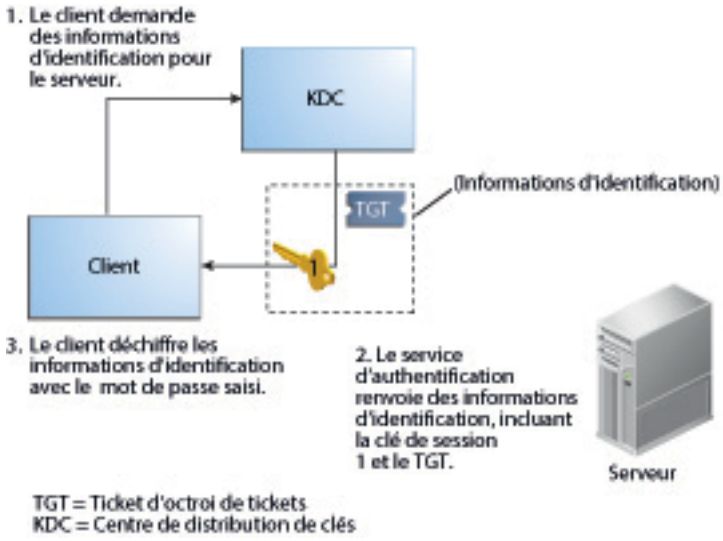
Le processus de création et de stockage des informations d'identification est transparent. Les informations d'identification sont créées par le KDC, qui les envoie aux demandeurs. Une fois reçues, les informations d'identification sont stockées dans un cache d'informations d'identification.

Obtention d'informations d'identification pour le service d'octroi de tickets

1. Pour démarrer le processus d'authentification, le client envoie une demande au serveur d'authentification pour un principal d'utilisateur spécifique. Cette demande est envoyée sans chiffrement. Aucune information sécurisée n'est incluse dans la demande, de sorte que le chiffrement est inutile.
2. Lorsque la demande est reçue par le service d'authentification, le nom de principal de l'utilisateur est recherché dans la base de données KDC. Si un principal correspond à l'entrée dans la base de données, le service d'authentification obtient la clé privée pour ce principal. Le service d'authentification génère ensuite une clé de session utilisable par le client et le service d'octroi de tickets (appelez-la Clé de session 1) et un ticket pour le service d'octroi de tickets (Ticket 1). Ce ticket est également qualifié de *ticket d'octroi de tickets* (TGT). La clé de session et le ticket sont chiffrés à l'aide de la clé privée de l'utilisateur, et les informations sont renvoyées au client.
3. Le client utilise ces informations pour déchiffrer la Clé de session 1 et Ticket 1 à l'aide de la clé privée pour le principal de l'utilisateur. Comme la clé privée doit uniquement être connue de l'utilisateur et de la base de données KDC, les informations du paquet doivent être sécurisées. Le client stocke les informations dans le cache d'informations d'identification.

Au cours de ce processus, l'utilisateur est invité à saisir un mot de passe normalement. Si le mot de passe spécifié par l'utilisateur est le même que celui utilisé pour créer la clé privée stockée dans la base de données KDC, alors le client peut déchiffrer les informations envoyées par le service d'authentification. Le client dispose d'informations d'identification à utiliser avec le service d'octroi de tickets et est prêt à demander des informations d'identification pour un serveur.

FIGURE 2-5 Obtention d'informations d'identification pour le service d'octroi de tickets

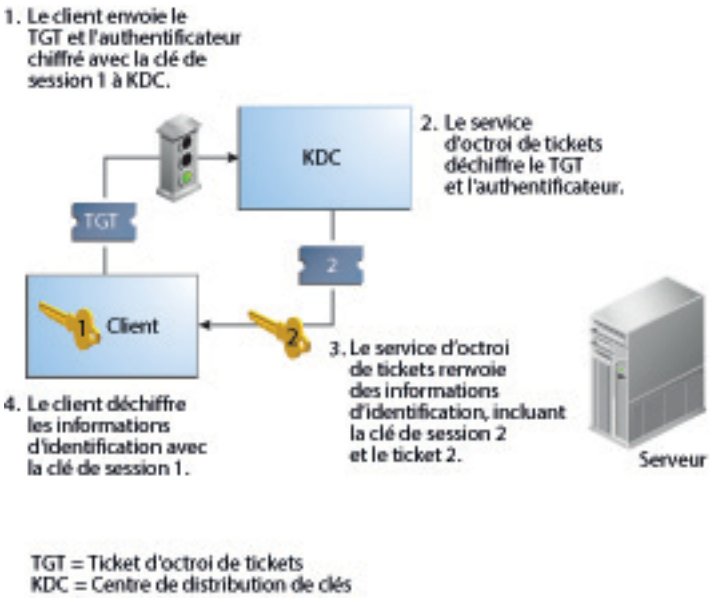


Obtention d'informations d'identification pour un serveur utilisant Kerberos

1. Pour demander l'accès à un serveur spécifique, un client doit d'abord avoir obtenu des informations d'identification pour ce serveur à partir du service d'authentification. Reportez-vous à la section “Obtention d'informations d'identification pour le service d'octroi de tickets” à la page 57. Le client envoie ensuite une demande au service d'octroi de ticket, qui inclut le nom du principal de service, Ticket 1, et un authentificateur chiffré avec Session Key 1. Ticket 1 a été initialement chiffré par le service d'authentification à l'aide de la clé du service d'octroi de tickets.
2. Comme la clé de service du service d'octroi de tickets est connue du service d'octroi de tickets, Ticket 1 peut être déchiffré. Les informations de Ticket 1 comprennent Session Key 1, de sorte que le service d'octroi de tickets peut déchiffrer l'authentificateur. A ce stade, le principal de l'utilisateur est authentifié avec le service d'octroi de tickets.
3. Une fois l'authentification réussie, le service d'octroi de tickets génère une clé de session pour le principal de l'utilisateur et le serveur (Clé de session 2), et un ticket pour le serveur (Ticket 2). Session Key 2 et Ticket 2 sont ensuite chiffrés à l'aide de Session Key 1. Etant donné que Session Key 1 est uniquement connue du client et du service d'octroi de tickets, cette information est sécurisée et peut être envoyée sans problème sur le réseau.

4. Lorsque le client reçoit ce paquet d'informations, il déchiffre les informations en utilisant Session Key 1, qui était stockée dans le cache des informations d'identification. Le client dispose d'informations d'identification à utiliser avec le serveur. Maintenant, le client est prêt à demander l'accès à un service particulier sur ce serveur.

FIGURE 2-6 Obtention d'informations d'identification pour un serveur

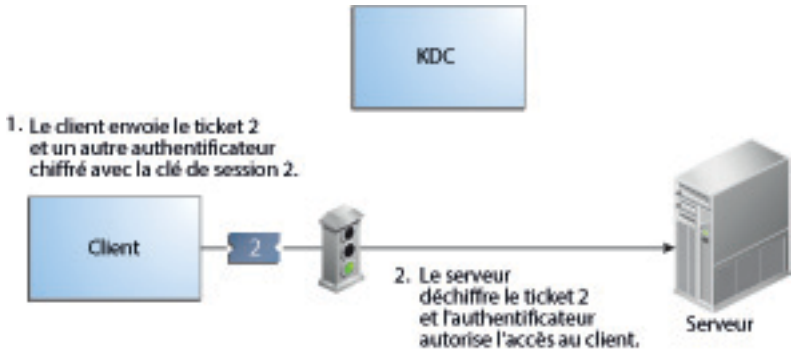


Obtention de l'accès à un service Kerberos donné

1. Pour demander l'accès à un service spécifique, le client doit d'abord avoir obtenu des informations d'identification pour le service d'octroi de tickets à partir du serveur d'authentification, et des informations d'identification de serveur à partir du service d'octroi de tickets. Reportez-vous à la section “Obtention d'informations d'identification pour le service d'octroi de tickets” à la page 57 et à la section“Obtention d'informations d'identification pour un serveur utilisant Kerberos” à la page 58. Le client peut alors envoyer une demande au serveur en incluant Ticket 2 et un autre authentificateur. L'authentificateur est chiffré à l'aide de Session key 2.
2. Ticket 2 a été chiffré par le service d'octroi de tickets avec la clé de service pour le service. Etant donné que la clé de service est appelée par le principal de service, le service peut déchiffrer Ticket 2 et obtenir Session Key 2. Session Key 2 peut alors être utilisée pour

déchiffrer l'authentificateur. Si l'authentificateur est correctement déchiffré, le client obtient l'accès au service.

FIGURE 2-7 Obtention de l'accès à un service donné



Différences notables entre Oracle Solaris Kerberos et MIT Kerberos

Les améliorations comprises dans Oracle Solaris mais non incluses dans MIT Kerberons sont les suivantes :

- Prise en charge par Kerberos des applications distantes Oracle Solaris
- Propagation incrémentielle de la base de données KDC
- Script de configuration client
- Messages d'erreur localisés
- Prise en charge des enregistrements d'audit Oracle Solaris
- Utilisation de Kerberos sécurisée par thread GSS-API à l'aide de
- Utilisation de la structure cryptographique pour la cryptographie

♦ ♦ ♦ CHAPITRE 3

Planification du service Kerberos

Ce chapitre couvre plusieurs options d'installation et de configuration que les administrateurs doivent résoudre avant de configurer et de déployer le service Kerberos :

Planification d'un déploiement Kerberos

Avant d'installer le service Kerberos, vous devez résoudre plusieurs problèmes de configuration. Bien que vous puissiez modifier la configuration après l'installation initiale, certaines modifications peuvent être difficiles à implémenter. En outre, certaines modifications impliquent que le KDC soit reconstruit, de sorte examiner les objectifs à long terme avant de déployer Kerberos.

Le déploiement d'une infrastructure Kerberos implique d'effectuer des tâches telles que l'installation de KDC, de créer des clés pour les hôtes et de faire migrer des utilisateurs. La reconfiguration d'un déploiement Kerberos peut être aussi difficile que l'exécution d'un déploiement initial, donc planifiez un déploiement avec soin pour éviter d'avoir à reconfigurer.

Planification de domaines Kerberos

Un *domaine* est réseau logique qui définit un groupe de systèmes qui sont sous le même KDC maître. Comme lors de l'établissement d'un nom de domaine DNS, vous devez résoudre les problèmes tels que le nom de domaine, le nombre et la taille de chaque domaine, ainsi que

la relation d'un domaine à d'autres domaines pour l'authentification inter-domaine avant de configurer le service Kerberos.

Noms de domaine Kerberos

Les noms de domaine peuvent être constitués de n'importe quelle chaîne de caractères ASCII. En général, le nom de domaine est le même que celui de votre nom de domaine DNS sauf que le nom de domaine est en majuscules. Cette convention permet de différencier les problèmes avec le service Kerberos des problèmes avec l'espace de noms DNS, tout en gardant un nom familier. Vous pouvez utiliser n'importe quelle chaîne, mais pourraient alors nécessitent des opérations de configuration et de maintenance davantage de travail. Utilisez les noms de domaine qui suivent les conventions de désignation d'Internet dans une grille de classification.

Nombre de domaines Kerberos

Le nombre de domaines requis par votre installation dépend de plusieurs facteurs :

- Le nombre de clients à prendre en charge. Trop de clients dans un domaine rend l'administration plus difficile et finit par vous forcer à diviser le domaine. Les principaux facteurs qui déterminent le nombre de clients pouvant être pris en charge sont les suivants :
  - Quantité de trafic générée par chaque client Kerberos
  - Bande passante du réseau physique
  - Vitesse de l'hôte

  Etant donné que chaque installation aura différentes limitations, aucune règle n'existe pour déterminer le nombre maximum de clients.
- La distance qui les sépare des clients. Configurer plusieurs petits domaines peut avoir un sens si les clients sont dans différentes régions.
- Le nombre d'hôtes disponibles pour être installés en tant que KDC. Envisagez de créer au moins deux serveurs par KDC, un serveur maître et de domaine de gestion des identités au moins un serveur esclave.

L'alignement des domaines Kerberos avec les domaines d'administration est recommandé. Il convient de noter qu'un domaine Kerberos V peut s'étendre sur plusieurs sous-domaines du domaine DNS auquel le domaine correspond.

Hiérarchie des domaines Kerberos

Lorsque vous configurez plusieurs domaines pour l'authentification inter-domaine, vous devez décider comment lier les domaines entre eux. Vous pouvez établir une relation hiérarchique

entre les domaines, ce qui fournit automatiquement les chemins d'accès aux domaines associés. Lorsque tous les domaines dans la chaîne hiérarchique sont correctement configurés, ces chemins d'accès automatiques peuvent alléger la charge administrative. Cependant, s'il existe de nombreux niveaux de domaines, il se peut que vous ne souhaitiez pas utiliser le chemin automatique car cela nécessite trop de transactions.

Vous pouvez également choisir d'établir la relation de confiance de manière directe. Une relation de confiance directe est utile lorsqu'un trop grand nombre de niveaux hiérarchiques existent entre deux domaines ou lorsqu'il n'existe aucune relation hiérarchique. La connexion doit être définie dans le fichier `/etc/krb5/krb5.conf` sur tous les hôtes qui utilisent la connexion, certains travaux supplémentaires sont alors nécessaires. La relation de confiance directe est également appelée relation transitive. Pour obtenir un exemple, reportez-vous à “Domaines Kerberos” à la page 50. Pour les procédures de configuration, reportez-vous à la section “Configuration de l'authentification inter-domaine” à la page 129.

Mappage de noms d'hôtes avec des domaines Kerberos

Le mappage de noms d'hôtes sur des noms de domaines est défini dans la section `domain_realm` du fichier `krb5.conf`. Ces mappages peuvent être définis pour un domaine ou pour des hôtes spécifiques, selon les besoins.

Vous pouvez également utiliser DNS pour chercher des informations sur le KDC. L'utilisation de DNS permet de changer plus facilement les informations car il est inutile de modifier le fichier `krb5.conf` sur tous les clients Kerberos pour chaque modification.

Remarque - Les procédures présentées dans ce manuel partent du principe l'utilisation de DNS.

Pour plus d'informations, reportez-vous à la page de manuel `krb5.conf`(4).

Les clients Kerberos dans Oracle Solaris interagissent correctement avec les serveurs Active Directory. Les serveurs Active Directory peuvent être configurés de façon à fournir le domaine au mappage d'hôte.

Noms de client et de principal de service Kerberos

Kerberos dans Oracle Solaris n'utilise pas le service `name-service/switch`. Utilise, le service Kerberos dans ce cas, il est préférable pour la résolution des noms d'hôte DNS. Par conséquent, DNS doit être activé sur tous les hôtes. Avec le FQDNDNS, le principal doit contenir le nom de domaine complet (FQDN, fully qualified domain name) de chaque hôte. Par exemple, si

le nom d'hôte est `boston`, le nom de domaine DNS est `example.com` et le nom de domaine est `EXAMPLE.COM`, le nom de principal de l'hôte doit être `host/boston.example.com@EXAMPLE.COM`. Les exemples de ce guide nécessitent que le DNS soit configuré et que le nom de domaine complet soit utilisé pour chaque hôte.

Le service Kerberos normalise les noms d'alias d'hôtes par l'intermédiaire du DNS, et utilise le formulaire normalisé (cname) lors de la construction du principal de service pour le service associé. C'est pourquoi, lors de la création d'un principal de service, le composant nom d'hôte des noms des principaux de service est la forme normalisée du nom d'hôte du système qui fournit le service.

L'exemple qui suit illustre de quelle façon le service Kerberos normalise les noms d'hôte. Si un utilisateur exécute la commande `ssh alpha.example.com` où `alpha.example.com` est un alias de nom d'hôte DNS pour le cname `beta.example.com`, le service Kerberos normalise `alpha.example.com` à `beta.example.com`. Le KDC traite le ticket en tant que demande pour le principal de service *host*`/beta.example.com`.

Pour les noms de principal qui comprennent le nom de domaine complet (FQDN) de l'hôte, assurez-vous de faire correspondre la chaîne qui décrit le nom de domaine DNS dans le fichier `/etc/resolv.conf`. Le service Kerberos requiert que le nom de domaine DNS soit en minuscules lorsque vous spécifiez le nom de domaine complet (FQDN) pour un principal. Le nom de domaine DNS peuvent inclure des majuscules et des minuscules, mais n'utilisez des lettres minuscules que lorsque vous créez un principal d'hôte. Par exemple, le nom du domaine DNS peut être `example.com`, `Example.COM` ou toute autre variante. Le nom du principal d'hôte sera toujours `host/boston.example.com@EXAMPLE.COM`.

En outre, l'utilitaire de gestion des services (SMF) a été configuré de manière à ce qu'un grand nombre de ces démons ou commandes ne démarre pas si le service de client DNS n'est pas en cours d'exécution. Les démons `kdb5_util`, `kadmind` et `kpropd`, ainsi que la commande `kprop`, sont configurés pour dépendre du service DNS. Pour exploiter au mieux les fonctionnalités disponibles à partir du service Kerberos et du SMF, vous devez activer le service de client DNS sur tous les hôtes.

Synchronisation de l'horloge dans un domaine Kerberos

Les horloges internes de tous les hôtes qui participent au système d'authentification Kerberos doivent être synchronisées dans une durée maximale spécifiée. Appelée *écart d'horloge*, cette fonction fournit un autre contrôle de sécurité Kerberos. Si l'écart d'horloge est dépassé entre des hôtes participants, les demandes sont rejetées.

Une façon de synchroniser toutes les horloges est d'utiliser le logiciel NTP (Network Time Protocol). Pour plus d'informations, reportez-vous à la section “Synchronisation des horloges

entre les KDC et les clients Kerberos” à la page 132. D'autres façons de synchroniser les horloges peut être utilisée. toutefois, une forme ou une autre de synchronisation est obligatoire.

Types de chiffrement pris en charge dans Kerberos

Un *type de chiffrement* est un identificateur qui spécifie l'algorithme de chiffrement, le mode de chiffrement et les algorithmes de hachage utilisés dans le service Kerberos. Les clés dans le service Kerberos sont associées à un type de chiffrement qui indique l'algorithme et le mode cryptographiques à utiliser lorsque le service effectue des opérations cryptographiques à l'aide de la clé. Pour obtenir une liste des types de chiffrement pris en charge, reportez-vous aux pages de manuel `krb5.conf`(4) et `kdb5_util`(1M).

Si vous souhaitez modifier le type de chiffrement, faites-le lors de la création d'une nouvelle base de données de principal. En raison de l'interaction entre le KDC, le serveur et le client, il est difficile de modifier le type de chiffrement sur une base de données existante. Pour plus d'informations, reportez-vous à la section “Types de chiffrement Kerberos” à la page 54.

Les types de chiffrement faible, tels que `des`, sont interdits par défaut. Si vous devez utiliser des types de chiffrement faible pour des raisons de compatibilité descendante ou d'interopérabilité, définissez l'entrée `allow_weak_crypto` dans la section `libdefaults` du fichier `/etc/krb5/krb5.conf` sur `true`.

Planification de KDC

Utilisez des ports spécifiques, KDC serveurs permettant de gérer plus grande taille supplémentaire est nécessaire, puis ticket nécessitent charge techniques pour conserver la propagation des serveurs synchronisés. En outre, types de chiffrement gérés de manière centralisée. Vous disposez de plusieurs options pour la configuration initiale d'votre KDC.

Ports pour les services d'administration et le KDC

Par défaut, les ports `88` et `750` sont utilisés pour le KDC, et le port `749` est utilisé pour le démon d'administration du KDC. Vous pouvez utiliser différents numéros de port. Toutefois, si vous modifiez les numéros de port, vous devez alors modifier les fichiers `/etc/services` et `/etc/krb5/krb5.conf` sur chaque client. En outre, vous devez mettre à jour le fichier `/etc/krb5/kdc.conf` sur chaque KDC.

Nombre de KDC esclaves

Les KDC esclaves génèrent des informations d'identification tout comme le KDC maître. Les KDC esclaves fournissent une sauvegarde si le maître n'est plus disponible. Envisagez de créer au moins un esclave KDC par domaine de gestion des identités.

Des KDC esclaves supplémentaires peuvent être nécessaires, selon les facteurs suivants :

- Nombre de segments physiques dans le domaine. Normalement, le réseau doit être défini de manière à ce que chaque segment puisse fonctionner, au moins de manière minimale, sans le reste du domaine. Pour ce faire, un KDC doit être accessible à partir de chaque segment. Le KDC dans cette instance pourrait être un maître ou un esclave.
- Nombre de clients dans le domaine. En ajoutant plusieurs serveurs KDC esclaves, vous pouvez réduire la charge des serveurs actuels.

Ne pas ajouter un trop grand nombre de KDC esclaves. La base de données KDC doit être diffusée vers chaque serveur. Par conséquent, plus le nombre de serveurs KDC installés est grand, plus la mise à jour des données dans l'ensemble du domaine peut être longue. En outre, puisque chaque esclave conserve une copie de la base de données KDC, le risque de violation de sécurité augmente avec le nombre d'esclaves.

Configurer un ou plusieurs KDC esclaves à être échangés avec le KDC maître. L'avantage de configurer au moins un KDC esclave de cette manière est de pouvoir disposer d'un système préconfiguré à devenir le KDC maître en cas de panne de celui-ci. Pour obtenir des instructions, reportez-vous à la section“Echange d'un KDC maître et d'un KDC esclave” à la page 133.

Propagation de la base de données Kerberos

La base de données stockée sur le KDC maître doit être régulièrement propagée aux KDC esclaves. Vous pouvez configurer la propagation de la base de données pour qu'elle soit incrémentielle. Le processus incrémentiel propage uniquement les informations mises à jour dans les KDC esclaves plutôt que l'intégralité de la base de données. Pour plus d'informations sur la propagation de base de données, reportez-vous à la section “Administration de la base de données Kerberos” à la page 138.

Si vous n'utilisez pas la propagation incrémentielle, l'une des premières questions à résoudre est la fréquence de mise à jour des KDC esclaves. Le besoin d'avoir des informations à jour disponibles pour tous les clients doit être comparé à la durée nécessaire pour effectuer la mise à jour.

Dans les installations de grande taille avec de nombreux KDC dans un domaine, un ou plusieurs esclaves peuvent propager les données de manière à ce que le processus s'effectue en parallèle.

Cette stratégie permet de réduire le temps de la mise à jour, mais il augmente également administration. Pour plus d'informations, reportez-vous à la section “Configuration de la propagation parallèle pour Kerberos” à la page 147.

Options de configuration de KDC

Il existe plusieurs façons de configurer un KDC. Les façons les plus simples utilisent l'utilitaire `kdcmgr` pour configurer le KDC automatiquement ou de façon interactive. La version automatique nécessite que vous utilisiez les options de ligne de commande afin de définir les paramètres de configuration. Cette méthode est particulièrement utile pour les scripts. La version interactive vous invite à fournir toutes les informations nécessaires. Pour accéder à des pointeurs vers les instructions d'utilisation de cette commande, reportez-vous à la section “Configuration des serveurs KDC” à la page 75.

Vous pouvez également utiliser les fichiers pour gérer la base de données LDAP pour Kerberos. Pour obtenir des instructions, reportez-vous à la section “Configuration du KDC maître pour utiliser un serveur d'annuaires LDAP” à la page 90. Le LDAP simplifie l'administration des sites qui nécessitent une coordination entre les bases de données Kerberos et la configuration existante de leur serveur d'annuaire.

Planification des clients Kerberos

Les clients Kerberos, installé peuvent être installés automatiquement à partir d'un script, ou en modifiant les fichiers de configuration configurés manuellement. Pour des connexions de réseau protégées, la structure PAM fournit le module `pam_ldap`. Voir la page de manuel `pam_ldap`(5). En outre, lorsque le client demande un service, ce service peuvent être accordés par un serveur autre que le KDC maître.

Planification de l'installation automatique de clients Kerberos

Les clients Kerberos peuvent être rapidement et facilement configurés à l'aide de la fonctionalité d'installation automatique (AI, Automated Installer) d'Oracle Solaris. Créer et affecter AI aux administrateurs de serveur pour les clients AI Profils de configuration Kerberos. En outre, le serveur en question fournit les clés du client AI. Par conséquent, lors de l'installation, le client Kerberos est un système Kerberos pleinement approvisionné,

capable d'héberger des services sécurisés. L'aide du programme d'installation automatisée et de maintenance peut diminuer les coûts d'administration du système.

Vous pouvez utiliser la commande `kclient` pour la création d'installation automatisée pour des clients de tout type de KDC.

- Vous pouvez utiliser AI pour les clients qui ne sont pas clients d'un KDC Oracle Solaris. Pour obtenir la liste des fournisseurs KDC, reportez-vous à la page de manuel `kclient`(1M).

  Pour tous les types de KDC, le transfert de keytab pré-généré est pris en charge. Oracle Solaris KDC et MS AD prennent également en charge l'inscription automatique.
- Vous exécutez la commande `kclient` pour créer des profils de configuration Kerberos pour AI. Pour plus d'informations, reportez-vous à la page de manuel `kclient`(1M) et à la section “Options de configuration du client Kerberos” à la page 68. Pour obtenir des instructions pour configurer les clients Kerberos par l'intermédiaire du AI, reportez-vous à la section “ Configuration des clients Kerberos à l’aide de l’AI ” du manuel “ Installation des systèmes Oracle Solaris 11.2 ”.

Options de configuration du client Kerberos

Vous pouvez configurer des clients Kerberos à l'aide de l'utilitaire de configuration `kclient` ou en modifiant des fichiers manuellement. L'utilitaire s'exécute en mode interactif et non interactif. En mode interactif, vous êtes invité à saisir des valeurs de paramètre spécifiques aux Kerberos, de sorte que vous puissiez le modifier lors de la configuration du client. En mode non interactif, vous devez fournir un fichier avec les valeurs des paramètres. De plus, vous pouvez ajouter des options de ligne de commande en mode non interactif. Modes car le programme d'installation en mode et non interactif nécessitent moins d'étapes que la configuration manuelle, ils sont plus rapide et moins sujette à erreur.

Si la configuration suivante est activée, cela signifie qu'aucune configuration de votre client Kerberos explicite est nécessaire :

- Le service DNS est configuré de façon à renvoyer des enregistrements SRV aux KDC.
- Le nom de domaine correspond au nom de domaine DNS ou le KDC prend en charge les références.
- Le client Kerberos ne requiert aucune clé différente des clés du serveur KDC.

Vous quand même souhaiterez peut - être configurer explicitement le client Kerberos pour les raisons suivantes :

- Le processus sans configuration effectue plus les recherches DNS que les clients configurés directement, et est ainsi moins efficace que la configuration directe.
- Si les références ne sont pas utilisées, la logique d'absence de configuration se base sur le nom de domaine DNS de l'hôte pour déterminer le domaine. Cette configuration introduit

un petit risque pour la sécurité, mais celui-ci est beaucoup plus faible que l'activation de `dns_lookup_realm`.

- Le module `pam_krb5` s'appuie sur une entrée de clé d'hôte dans le fichier keytab. Bien que cette condition puisse être désactivée dans le fichier `krb5.conf`, cette opération n'est pas recommandé pour des raisons de sécurité. Pour plus d'informations, reportez-vous à la section “Sécurité de connexion du client Kerberos” à la page 69 et à la page de manuel `krb5.conf`(4).

Pour obtenir une description complète de la configuration client, reportez-vous à la section “Configuration des clients Kerberos” à la page 99.

Sécurité de connexion du client Kerberos

A l'ouverture de la session, un client utilise le module `pam_krb5` pour vérifier que le KDC ayant émis le dernier TGT est le même KDC ayant émis le principal d'hôte du client stocké dans le fichier `/etc/krb5/krb5.keytab`. Le module `pam_krb5` vérifie le KDC lorsque le module est configuré dans la pile d'authentification. Pour certaines configurations, telles que des clients DHCP ne stockant pas de principal d'hôte de client, cette vérification doit être désactivée. Pour désactiver cette vérification, vous devez définir l'option `verify_ap_req_nofail` dans le fichier `krb5.conf` sur `false`. Pour plus d'informations, reportez-vous à la section “Désactivation de la vérification du ticket d'octroi de tickets” à la page 111.

Services délégués de confiance dans Kerberos

Pour certaines applications, un client peut avoir besoin de déléguer l'autorité à un serveur pour qu'il agisse en son nom lorsqu'il contacte d'autres services. Le client doit transférer des informations d'identification à un serveur intermédiaire. La capacité du client à obtenir un ticket de service à un serveur ne communique aucune information au client permettant de savoir si le serveur peut être autorisé à accepter des références déléguées. L'option `ok_to_auth_as_delegate` de la commande `kadmin` permet à un KDC de communiquer la stratégie du domaine local à un client indiquant si un serveur intermédiaire est autorisé à accepter de telles informations d'identification.

La partie chiffrée de la réponse KDC au client peut inclure une copie des indicateurs de ticket d'informations d'identification avec l'option `ok_to_auth_as_delegate`. Un client peut utiliser ce paramètre pour déterminer s'il faut attribuer ou non des informations d'identification (en octroyant un proxy ou un TGT transmis) à ce serveur. Lors de la définition de cette option, prenez en compte la sécurité et la position du serveur sur lequel le service s'exécute, et si le service requiert l'utilisation d'informations d'identification déléguées.

Planification de l'usage par Kerberos des noms et des informations d'identification et de connexion UNIX

Le service Kerberos fournit un mappage par défaut des noms d'informations d'identification GSS sur les noms d'utilisateurs UNIX (UID) pour les applications GSS nécessitant ce mappage, comme NFS. Les noms d'informations d'identification GSS sont équivalents aux noms de principaux Kerberos lors de l'utilisation du service Kerberos. De même, les utilisateurs UNIX ne disposant pas de comptes utilisateur valables dans le domaine Kerberos par défaut peuvent être automatiquement migrés à l'aide de la structure PAM.

Mappage d'informations d'identification GSS avec des informations d'identification UNIX

L'algorithme de mappage par défaut utilise le nom principal de l'utilisateur ou l'hôte principal Kerberos sous lequel rechercher l'UID. La recherche est effectuée dans le domaine par défaut ou tout domaine autorisé par le paramètre `auth_to_local_realm` dans le fichier `/etc/krb5/krb5.conf`. Par exemple, le nom de principal d'utilisateur `jdoe@EXAMPLE.COM` est mappé sur l'UID de l'utilisateur UNIX nommé `jdoe` à l'aide de la table de mots de passe. Le nom de principal d'utilisateur `jdoe/admin@EXAMPLE.COM` n'est pas mappé car le nom de principal comprend le composant de l'instance `admin`.

Si les mappages par défaut pour les informations d'identification de l'utilisateur sont suffisants, la table des informations d'identification GSS n'a pas besoin d'être renseignée. Si le mappage par défaut n'est pas suffisant, comme lorsque vous souhaitez mapper un nom de principal contenant un composant d'instance, d'autres méthodes sont requises. Pour plus d'informations, reportez-vous aux références suivantes :

- “Création et modification d'une table d'informations d'identification” à la page 123
- “Mappage d'informations d'identification entre domaines” à la page 124
- “Observation du mappage d'informations d'identification GSS sur des informations d'identification UNIX” à la page 201

Table `gsscred`

La table `gsscred` est utilisée par un serveur NFS lorsque le serveur tente d'identifier un utilisateur Kerberos, si les mappages par défaut ne sont pas suffisants. Le service NFS utilise des UID UNIX pour identifier les utilisateurs. Ces ID ne font pas partie d'un principal ou d'informations d'identification d'utilisateur. La table `gsscred` assure le mappage supplémentaires d'informations d'identification GSS aux UID UNIX (à partir du fichier de

mots de passe). La table doit être créée et administrée une fois que la base de données KDC est remplie.

Lorsqu'une demande de client arrive, le service NFS tente de faire correspondre le nom des informations d'identification avec un UID UNIX. Si le mappage échoue, la table `gsscred` est vérifiée.

Migration automatique d'utilisateurs vers un domaine Kerberos

Les utilisateurs UNIX ne disposant pas de comptes utilisateur valables dans le domaine Kerberos par défaut peuvent être automatiquement migrés à l'aide de la structure PAM. Vous ajoutez, en particulier, le module `pam_krb5_migrate.so` à la pile d'authentification du service PAM. Les services sont alors configurés de sorte que, chaque fois qu'un utilisateur ne disposant pas d'un principal Kerberos parvient à se connecter à un système à l'aide de son mot de passe, un principal Kerberos est automatiquement créé pour celui-ci. Le mot de passe du nouveau principal est le même que le mot de passe UNIX. Pour obtenir des instructions à l'aide du module `pam_krb5_migrate.so`, reportez-vous à la section “Configuration de la migration automatique des utilisateurs dans un domaine Kerberos” à la page 112.

♦ ♦ ♦ CHAPITRE 4

Configuration du service Kerberos

Ce chapitre fournit les procédures de configuration pour les serveurs KDC, les serveurs d'application réseau, les serveurs NFS et les clients Kerberos. L'accès `root` est nécessaire pour un grand nombre de ces procédures, elles doivent donc être réalisées par les administrateurs système ou les utilisateurs avancés. Les procédures de configuration inter-domaines et d'autres sujets liés aux serveurs KDC sont également abordés.

Ce chapitre comprend les sections suivantes :

Configuration du service Kerberos

Dans la mesure où certaines procédures décrites dans la procédure de configuration dépendent d'autres procédures, ils doivent être effectuées selon un ordre spécifique. Ces procédures établissent souvent les services requis pour utiliser le service Kerberos. D'autres procédures ne sont pas dépendantes de l'ordre et peuvent être effectuées si nécessaire. La liste des tâches ci-dessous présente un ordre suggéré en vue d'une installation Kerberos.

Remarque - Les exemples dans ces sections utilisent des types de chiffrement par défaut, qui ne sont pas validés FIPS 140 pour Oracle Solaris. Pour une exécution en Mode FIPS 140, vous devez limiter les types de chiffrement au type de chiffrement `des3-cbc-sha1` pour la base de données, les serveurs et les communications client. Avant de créer le KDC , modifiez les fichiers dans “Configuration de Kerberos afin qu'il s'exécute en mode FIPS 140” à la page 77.

TABLEAU 4-1 Configuration de la liste des tâches de service Kerberos

Tâche	Description	Pour obtenir des instructions
1. Planification de votre installation Kerberos	Résout les problèmes de configuration avant de démarrer le processus de configuration du logiciel. La planification permet d'économiser du temps et d'autres ressources ultérieurement.	Chapitre 3, Planification du service Kerberos
2. Configuration des serveurs KDC	Configure et construit des serveurs KDC maître et KDC esclave ainsi que la base de données KDC pour un domaine.	“Configuration des serveurs KDC” à la page 75
2a. (Facultatif) Configurez Kerberos pour qu'il s'exécute en Mode FIPS 140.	Active l'utilisation d'algorithmes validés FIPS 140 uniquement.	“Configuration de Kerberos afin qu'il s'exécute en mode FIPS 140” à la page 77
2b. (Optionnel) Configuration de Kerberos pour qu'il s'exécute sur LDAP.	KDC pour utiliser un configure le serveur d'annuaire LDAP.	“Gestion d'un KDC sur un serveur d'annuaire LDAP” à la page 97
3. Installation du logiciel NTP (Network Time Protocol).	Crée un temps de l'horloge qui fournit l'central pour tous les systèmes sur le réseau.	“Synchronisation des horloges entre les KDC et les clients Kerberos” à la page 132
4. (Facultatif) Configuration des serveurs KDC échangeables	Facilite la tâche d'échange du serveur KDC maître et esclave.	“Configuration d'un KDC échangeable” à la page 134
4. (Facultatif) Augmentation de la sécurité sur les serveurs KDC	Permet d'éviter les violations de sécurité sur le serveur KDC.	“Restriction de l'accès aux serveurs KDC” à la page 152

Configuration de services Kerberos supplémentaires

Une fois les étapes requises effectuées, réalisez les procédures suivantes le cas échéant.

TABLEAU 4-2 Configuration d'une liste de tâches des services Kerberos supplémentaires

Tâche	Description	Pour obtenir des instructions
Configuration de l'authentification inter-domaine	Active les communications entre deux domaines.	“Configuration de l'authentification inter-domaine” à la page 129
Configuration de serveurs d'application Kerberos	Permet à un serveur de prendre en charge des services tels que `ftp` en utilisant l'authentification Kerberos.	“Configuration des serveurs d'application réseau Kerberos” à la page 118
Les services d'exécution configurer retardée.	Permet à un hôte `cron` d'exécuter des tâches à des moments arbitraires.	“Configuration de l'exécution retardée pour l'accès aux services Kerberos” à la page 127

Tâche	Description	Pour obtenir des instructions
Configuration du serveur NFS Kerberos	Permet à un serveur de partager un système de fichiers requérant l'authentification Kerberos.	“Configuration de serveurs NFS Kerberos” à la page 121
Configuration des clients Kerberos	Permet à un client d'utiliser des services Kerberos.	“Configuration des clients Kerberos” à la page 99
Synchroniser les horloges pour aider l'authentification.	Configure un protocole de synchronisation d'horloge.	“Synchronisation des horloges entre les KDC et les clients Kerberos” à la page 132
Gérer la base de données Kerberos.	Permet de tenir à jour la base de données Kerberos.	“Administration de la base de données Kerberos” à la page 138
Gérer le fichier stash de la base de données Kerberos.	Gère les clés pour la base de données Kerberos.	“Administration du fichier stash pour la base de données Kerberos” à la page 149

Configuration des serveurs KDC

Une fois le logiciel Kerberos installé, vous devez configurer les serveurs KDC (Key Distribution Centre, Centre de distribution des clés). La configuration du KDC maître et d'au moins un KDC esclave fournit le service émetteur des informations d'identification. Ces informations d'identification étant la base du service Kerberos, les KDC doivent être configurés avant de tenter d'effectuer d'autres tâches.

La différence principale entre un KDC maître et un KDC esclave est que seul le KDC maître peut traiter les demandes d'administration de base de données. Par exemple, la modification d'un mot de passe ou l'ajout d'un nouveau principal doivent s'effectuer sur le KDC maître. Ces modifications peuvent alors être propagées aux KDC esclaves. Les KDC esclaves et maître génèrent tous deux des informations d'identification. Les KDC esclaves KDC assurer la redondance que l'EdC principal ne répond pas.

Vous pouvez choisir de configurer et générer les serveur KDC maître, la base de données, et d'autres serveurs de plusieurs manières :

- Automatique - recommandé pour les scripts
- Interactif - suffisant pour la plupart des installations
- Manuel - nécessaire pour des installations plus complexes
- Manuel avec LDAP - nécessaire lors de l'utilisation de LDAP avec KDC

TABLEAU 4-3 Configuration de la liste des tâches des serveurs KDC

Tâche	Description	Pour obtenir des instructions
Installation du package KDC.	Package obligatoire pour la création de KDC.	“Installation du package KDC” à la page 76
(Facultatif) Configurez Kerberos pour qu'il s'exécute en Mode FIPS 140.	Active l'utilisation d'algorithmes validés FIPS 140 uniquement.	“Configuration de Kerberos afin qu'il s'exécute en mode FIPS 140” à la page 77

Tâche	Description	Pour obtenir des instructions
Utilisez un script qui permet de configurer le KDC maître.	Simplifie la configuration initiale.	“Utilisation de `kdcmgr` pour configurer le KDC maître” à la page 78 Exemple 4-1, “Exécution de la commande `kdcmgr` sans arguments”
Utilisez un script pour configurer un serveur KDC esclave.	Simplifie la configuration initiale.	“Utilisation de `kdcmgr` pour configurer un KDC esclave” à la page 80
Configurez manuellement le serveur KDC maître.	Fournit le contrôle de toutes les entrées de la les fichiers de configuration KDC durant l'installation initiale.	“Configuration manuelle d'un KDC maître” à la page 81
Configuration manuelle d'un serveur KDC esclave.	Fournit le contrôle de toutes les entrées de la les fichiers de configuration KDC durant l'installation initiale.	“Configuration manuelle d'un KDC esclave” à la page 86
Pour configurer manuellement les utiliser LDAP KDC maître.	Configure le serveur pour utiliser LDAP KDC maître.	“Configuration du KDC maître pour utiliser un serveur d'annuaires LDAP” à la page 90
Remplacement des clés principales sur un serveur KDC.	Met à jour la clé de session sur un serveur KDC hérité pour utiliser des types de chiffrement renforcés.	“Remplacement des clés TGS sur un serveur maître” à la page 96

▼ Installation du package KDC

Logiciel client par défaut, ces derniers est installé sur votre système. Pour installer un KDC (Key Distribution Center), vous devez ajouter le package KDC.

Avant de commencer Vous devez disposer du profil de droits d'installation de logiciels pour ajouter des packages au système. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. Installation du package KDC.

```
$ pkg install system/security/kerberos-5
```

Pour plus d'informations, reportez-vous à la page de manuel`pkg`(1).

2. (Facultatif) Liste les services Kerberos.

A la différence près que le package du serveur votre système dispose de deux supplémentaires, les services Kerberos. Comme Kerberos logiciel client, ces services sont désactivés par défaut. Avant d'activer ces vous configurez des services Kerberos.

```
$ svcs krb5
STATE          STIME    FMRI
disabled       Sep_10   svc:/network/security/krb5kdc:default
disabled       Sep_10   svc:/network/security/krb5_prop:default
$ svcs | grep kerb
```

```
STATE          STIME    FMRI
disabled       Sep_07   svc:/system/kerberos/install:default
```

▼ Configuration de Kerberos afin qu'il s'exécute en mode FIPS 140

Avant de commencer

Pour que Kerberos s'exécute en Mode FIPS 140, vous devez activer le Mode FIPS 140 sur votre système. Reportez-vous à " Création d'un environnement d'initialisation compatible FIPS 140 " du manuel " Gestion du chiffrement et des certificats dans Oracle Solaris 11.2 ".

1. **Modifiez les types de chiffrement pour le KDC.**

   Dans la section `[realms]` du fichier `kdc.conf`, définissez le type de clé maître pour la base de données KDC :

   ```
   # pfedit  /etc/krb5/kdc.conf
   ...
   master_key_type = des3-cbc-sha1-kd
   ```

2. **Dans le même fichier, interdisez d'autres types de chiffrement de manière explicite.**

   Etant donné que vous pouvez également définir le chiffrement en exécutant une commande, les fichiers de configuration doivent empêcher l'utilisation d'un argument d'algorithme non FIPS 140 dans une commande.

   ```
   supported_enctypes = des3-cbc-sha1-kd:normal
   ```

3. **Modifiez les types de chiffrement pour les transactions dans la section `[libdefaults]` du fichier `krb5.conf`.**

   Ces paramètres limitent les types de chiffrement pour les serveurs, services et clients Kerberos.

   ```
   # pfedit /etc/krb5/krb5.conf
          default_tgs_enctypes = des3-cbc-sha1-kd
          default_tkt_enctypes = des3-cbc-sha1-kd
          permitted_enctypes = des3-cbc-sha1-kd
   ```

4. **Dans le même fichier, interdisez des types de chiffrement faible de manière explicite.**

   ```
   allow_weak_enctypes = false
   ```

Erreurs fréquentes

Reportez-vous à "Types de chiffrement Kerberos" à la page 54

Utilisation de kdcmgr pour configurer le KDC maître

Le script kdcmgr fournit une interface de ligne de commande pour installer les KDC maître et esclave. Pour le serveur maître, vous devez créer un mot de passe pour la base de données Kerberos et un mot de passe de l'administrateur. Sur les KDC esclaves, vous devez fournir ces mots de passe pour terminer l'installation. Pour plus d'informations sur ces mots de passe, reportez-vous à la page de manuel kdcmgr(1M).

Avant de commencer Vous devez prendre le rôle root. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. Créer le KDC maître.

Sur la ligne de commande, exécutez la commande kdcmgr et attribuez un nom à l'administrateur et au domaine.

Vous êtes invité à saisir le mot de passe de la base de données Kerberos, c'es-à-dire la *clé maître* ainsi que le mot de passe du principal d'administration. Le script affiche pour les mots de passe.

```
kdc1# kdcmgr -a kws/admin -r EXAMPLE.COM create master

Starting server setup
---------------------------------------

Setting up /etc/krb5/kdc.conf

Setting up /etc/krb5/krb5.conf

Initializing database '/var/krb5/principal' for realm 'EXAMPLE.COM',
master key name 'K/M@EXAMPLE.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:     /** Type strong password **/
Re-enter KDC database master key to verify: xxxxxxxx

Authenticating as principal root/admin@EXAMPLE.COM with password.
WARNING: no policy specified for kws/admin@EXAMPLE.COM; defaulting to no policy
Enter password for principal "kws/admin@EXAMPLE.COM":     /** Type strong password **/
Re-enter password for principal "kws/admin@EXAMPLE.COM": xxxxxxxx
Principal "kws/admin@EXAMPLE.COM" created.

Setting up /etc/krb5/kadm5.acl.

--------------------------------------------------
Setup COMPLETE.

kdc1#
```

Remarque - Enregistrer et stocker ces mots de passe dans un endroit sûr.

2. **(Facultatif) Afficher le statut du KDC maître.**

```
# kdcmgr status
```

3. **Synchroniser l'horloge de ce système avec d'autres horloges dans le domaine à l'aide du NTP ou d'un autre mécanisme.**

Pour une authentification réussie, chaque horloge doit être réglée sur l'heure par défaut définie dans la section `libdefaults` du fichier `krb5.conf`. Pour plus d'informations, reportez-vous à la page de manuel `krb5.conf`(4). Pour plus d'informations sur le NTP (Network Time Protocol), reportez-vous à la section “Synchronisation des horloges entre les KDC et les clients Kerberos” à la page 132.

Remarque - Ne peut pas être le la NTP KDC maître serveur. Si vous ne disposez pas d'un serveur NTP KDC, revenez à la une fois le serveur NTP maître est installé et que vous rendez les maître NTP KDC un client du serveur.

Exemple 4-1 Exécution de la commande `kdcmgr` sans arguments

Dans cet exemple, l'administrateur et les approvisionnements de principal admin du nom de domaine (realm) lorsque vous y êtes invité par le script.

```
kdc1# kdcmgr create master

Starting server setup
-------------------------------------

Enter the Kerberos realm: EXAMPLE.COM

Setting up /etc/krb5/kdc.conf

Setting up /etc/krb5/krb5.conf

Initializing database '/var/krb5/principal' for realm 'EXAMPLE.COM',
master key name 'K/M@EXAMPLE.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:     /** Type strong password **/
Re-enter KDC database master key to verify: xxxxxxxx

Enter the krb5 administrative principal to be created: kws/admin

Authenticating as principal root/admin@EXAMPLE.COM with password.
WARNING: no policy specified for kws/admin@EXAMPLE.COM; defaulting to no policy
Enter password for principal "kws/admin@EXAMPLE.COM":     /** Type strong password **/
Re-enter password for principal "kws/admin@EXAMPLE.COM": xxxxxxxx
Principal "kws/admin@EXAMPLE.COM" created.

Setting up /etc/krb5/kadm5.acl.
```

```
------------------------------------------------
Setup COMPLETE.

kdc1#
```

▼ Utilisation de kdcmgr pour configurer un KDC esclave

Avant de commencer

Le serveur KDC maître est configuré.

Vous devez prendre le rôle root. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. Créer un KDC esclave.

Sur la ligne de commande, exécutez la commande kdcmgr et attribuez un nom à l'administrateur, au domaine et au KDC maître.

Le script vous invite à saisir les deux mots de passe créés à la section "Utilisation de kdcmgr pour configurer le KDC maître" à la page 78, l'un pour le principal d'administration et l'autre pour la base de données KDC.

```
kdc2# kdcmgr -a kws/admin -r EXAMPLE.COM create -m kdc1 slave

Starting server setup
--------------------------------------

Setting up /etc/krb5/kdc.conf

Setting up /etc/krb5/krb5.conf
Obtaining TGT for kws/admin ...
Password for kws/admin@EXAMPLE.COM: xxxxxxxx

Setting up /etc/krb5/kadm5.acl.

Setting up /etc/krb5/kpropd.acl.

Waiting for database from master...
Waiting for database from master...
Waiting for database from master...
kdb5_util: Cannot find/read stored master key while reading master key
kdb5_util: Warning: proceeding without master key
Enter KDC database master key: xxxxxxxx

------------------------------------------------
Setup COMPLETE.

kdc2#
```

2. **(Facultatif) Afficher le statut du KDC.**

```
# kdcmgr status
```

3. **Synchroniser l'horloge de ce système avec d'autres horloges dans le domaine à l'aide du NTP ou d'un autre mécanisme.**
Si vous ne disposez pas d'un serveur NTP, vous pouvez utiliser ce système que votre serveur NTP.
Pour une authentification réussie, chaque horloge doit être réglée sur l'heure par défaut définie dans la section `libdefaults` du fichier `krb5.conf`. Pour plus d'informations, reportez-vous à la page de manuel `krb5.conf`(4). Pour plus d'informations sur le NTP (Network Time Protocol), reportez-vous à la section “Synchronisation des horloges entre les KDC et les clients Kerberos” à la page 132.

Étapes suivantes Revenir à la NTP KDC maître KDC après la le serveur KDC maître à un client NTP sur le serveur NTP.

▼ Configuration manuelle d'un KDC maître

Dans cette procédure, la propagation incrémentielle est configurée. Cette procédure utilise les paramètres de configuration ci-dessous :

- Nom de domaine = `EXAMPLE.COM`
- Nom de domaine DNS = `example.com`
- KDC maître = `kdc1.example.com`
- `admin` principal = `kws/admin`
- URL de l'aide en ligne = `http://docs.oracle.com/cd/E23824_01/html/821-1456/aadmin-23.html`

Remarque - Ajuster l'URL de sorte qu'il pointe vers l'emplacement de l'aide en ligne, comme décrit dans “Interface graphique `gkadmin`” à la page 157.

Avant de commencer L'hôte soit configuré pour utiliser DNS. Pour obtenir des instructions de nommage spécifiques afin de déterminer si ce maître doit être échangeable, reportez-vous à la section “Echange d'un KDC maître et d'un KDC esclave” à la page 133.

Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. **Installation du package KDC.**

Suivez les instructions de la section “Installation du package KDC” à la page 76.

2. **Edition du fichier de configuration Kerberos, `krb5.conf`.**

   Pour plus d'informations sur ce fichier, reportez-vous à la page de manuel `krb5.conf`(4). Dans cet exemple, l'administrateur modifie les lignes pour `default_realm`, `kdc`, `admin_server` et toutes les entrées `domain_realm` et édite l'entrée `help_url`.

```
kdc1 # pfedit /etc/krb5/krb5.conf
...
[libdefaults]
default_realm = EXAMPLE.COM

[realms]
EXAMPLE.COM = {
kdc = kdc1.example.com
admin_server = kdc1.example.com
}

[domain_realm]
.example.com = EXAMPLE.COM
#
# if the domain name and realm name are equivalent,
# this entry is not needed
#
[logging]
default = FILE:/var/krb5/kdc.log
kdc = FILE:/var/krb5/kdc.log

[appdefaults]
gkadmin = {
help_url = http://docs.oracle.com/cd/E23824_01/html/821-1456/aadmin-23.html
}
```

Remarque - Si vous devez communiquer à l'aide d'une version antérieure, vous aurez sans doute besoin Kerberos pour limiter les types de chiffrement. Pour une description des problèmes liés à la restriction des types de chiffrement, reportez-vous à la section “Types de chiffrement Kerberos” à la page 54.

3. **Attribution d'un nom au fichier de configuration du domaine, dans le KDC. `kdc.conf`**

   Pour plus d'informations sur ce fichier, reportez-vous à la page de manuel `kdc.conf`(4). Dans cet exemple, en plus de la définition du nom de domaine, l'administrateur modifie la propagation incrémentielle et erreurs de journalisation.

```
kdc1 # pfedit /etc/krb5/kdc.conf
[kdcdefaults]
kdc_ports = 88,750
```

```
[realms]
EXAMPLE.COM = {
profile = /etc/krb5/krb5.conf
database_name = /var/krb5/principal
acl_file = /etc/krb5/kadm5.acl
kadmind_port = 749
max_life = 8h 0m 0s
max_renewable_life = 7d 0h 0m 0s
sunw_dbprop_enable = true
sunw_dbprop_master_ulogsize = 1000
}
```

Remarque - Si vous devez communiquer à l'aide d'une version antérieure, vous aurez sans doute besoin Kerberos pour limiter les types de chiffrement. Pour une description des problèmes liés à la restriction des types de chiffrement, reportez-vous à la section “Types de chiffrement Kerberos” à la page 54.

4. Créer la base de données KDC à l'aide de la commande `kdb5_util`.

La commande `kdb5_util` crée la base de données KDC. En outre, lorsqu'elle est utilisée avec l'option `-s`, cette commande crée un fichier stash utilisé pour authentifier le KDC à lui-même avant le lancement des démons `kadmind` et `krb5kdc`. Pour plus d'informations, reportez-vous aux pages de manuel `kdb5_util`(1M), `kadmind`(1M) et`krb5kdc`(1M).

```
kdc1 # /usr/sbin/kdb5_util create -s
Initializing database '/var/krb5/principal' for realm 'EXAMPLE.COM'
master key name 'K/M@EXAMPLE.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:     /** Type strong password **/
Re-enter KDC database master key to verify: xxxxxxxx
```

Astuce - Si cette étape échoue, vérifiez que le est identifié à l'aide d'un KDC FQDN du principal.

```
# getent hosts IP-address-of-KDC
IP-address-of-KDC  kdc       /** This entry does not include FQDN **/
```

Ensuite, ajoutez le FQDN sous forme de première entrée KDC de votre fichier `/etc/hosts`, par exemple :

IP-address-of-KDC **kdc.***kdc-principal***.example.com kdc**

5. Editer le fichier de la liste de contrôle d'accès Kerberos, `kadm5.acl`.

Une fois renseigné, le fichier `/etc/krb5/kadm5.acl` doit contenir tous les noms de principaux autorisés à administrer le KDC.

```
kws/admin@EXAMPLE.COM    *
```

L'entrée précédente donne au principal `kws/admin` du domaine `EXAMPLE.COM` la possibilité de modifier les principaux et les stratégies dans le KDC. La valeur par défaut de l'entrée du principal est un astérisque (*), qui renvoie à tous les principaux `admin`. Cette entrée peut se présenter un risque en matière de sécurité. Modifiez le fichier de façon explicite afin de pouvoir répertorier chaque principal `admin` ainsi que leurs droits. Pour plus d'informations, reportez-vous à la page de manuel `kadm5.acl`(4).

6. Ajoutez les principaux d'administration à la base de données.

Vous pouvez ajouter autant de principaux `admin` que nécessaire. Vous devez ajouter au moins un principal `admin` pour terminer le processus de configuration du KDC. Pour cet exemple, un principal `kws/admin` est ajouté. Vous pouvez remplacer "`kws`" par le nom de principal approprié.

```
kadmin.local: addprinc kws/admin
Enter password
for principal kws/admin@EXAMPLE.COM:      /** Type strong password **/
Re-enter password
for principal kws/admin@EXAMPLE.COM: xxxxxxxx
Principal "kws/admin@EXAMPLE.COM" created.
kadmin.local:
```

Pour plus d'informations, reportez-vous à la page de manuel `kadmin`(1M).

7. Démarrez les démons Kerberos.

```
kdc1 # svcadm enable -r network/security/krb5kdc
kdc1 # svcadm enable -r network/security/kadmin
```

8. Démarrez `kadmin` et ajoutez d'autres principaux.

```
kdc1 # /usr/sbin/kadmin -p kws/admin
Enter password: xxxxxxxx
kadmin:
```

a. Créer le KDC maître `host` Principal.

L'hôte principal est utilisé par les applications utilisant Kerberos, notamment `kprop`, pour propager les modifications aux KDC esclaves. Ce principal est également utilisé pour fournir un accès à distance sécurisé au serveur KDC à l'aide d'applications de réseau, comme `ssh`. Notez que lorsque l'instance de principal est un nom d'hôte, le nom de domaine complet (FQDN) doit être spécifié en minuscules, quelle que soit la casse du nom de domaine dans le service de noms.

```
kadmin: addprinc -randkey host/kdc1.example.com
Principal "host/kdc1.example.com@EXAMPLE.COM" created.
kadmin:
```

b. (Facultatif) Créer le principal `kclient`.

Ce principal est utilisé par l'utilitaire `kclient` au cours de l'installation d'un client Kerberos. Si vous n'avez pas l'intention d'utiliser cet utilitaire, vous n'avez pas besoin d'ajouter le principal. Les utilisateurs de l'utilitaire `kclient` doivent utiliser ce mot de passe. Pour plus d'informations, reportez-vous à la page de manuel `kclient`(1M).

```
kadmin: addprinc clntconfig/admin
Enter password for principal clntconfig/admin@EXAMPLE.COM:      /** Type strong  password **/
Re-enter password for principal clntconfig/admin@EXAMPLE.COM: xxxxxxxx
Principal "clntconfig/admin@EXAMPLE.COM" created.
kadmin:
```

Remarque - Enregistrer et stocker ce mot de passe dans un endroit sûr.

c. Ajouter des privilèges au principal `clntconfig/admin`.

Modifiez le fichier `kadm5.acl` afin d'accorder au principal `clntconfig` les privilèges suffisants pour réaliser les tâches d'installation `kclient`.

```
# pfedit /etc/krb5/kadm5.acl
...
clntconfig/admin@EXAMPLE.COM  acdilm
```

d. Ajouter le principal `host` du KDC maître au fichier keytab du KDC maître.

L'ajout du principal d'hôte au fichier keytab permet aux serveurs d'application d'utiliser automatiquement ce principal, comme par exemple `sshd`.

```
kadmin: ktadd host/kdc1.example.com
Entry for principal host/kdc1.example.com with kvno 3, encryption type AES-256 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal host/kdc1.example.com with kvno 3, encryption type AES-128 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal host/kdc1.example.com with kvno 3, encryption type Triple DES cbc
mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:
```

e. Quitter `kadmin`.

```
kadmin: quit
```

9. L'horloge synchroniser ce du système dans le domaine de gestion des identités avec d'autres à l'aide des horloges NTP ou un autre mécanisme.

Pour une authentification réussie, chaque horloge doit être réglée sur l'heure par défaut définie dans la section `libdefaults` du fichier `krb5.conf`. Pour plus d'informations, reportez-vous à la page de manuel `krb5.conf`(4). Pour plus d'informations sur le NTP (Network Time Protocol), reportez-vous à la section “Synchronisation des horloges entre les KDC et les clients Kerberos” à la page 132.

Remarque - Ne peut pas être le la NTP KDC maître serveur. Si vous ne disposez pas d'un serveur NTP KDC, revenez à la une fois le serveur NTP maître est installé et que vous rendez les maître NTP KDC un client du serveur.

10. **Configurez les KDC esclaves.**

    Pour assurer la redondance, veillez à installer au moins un KDC esclave. Suivez les instructions de la section “Utilisation de `kdcmgr` pour configurer un KDC esclave” à la page 80 ou “Configuration manuelle d'un KDC esclave” à la page 86.

▼ Configuration manuelle d'un KDC esclave

Dans cette procédure, un nouveau KDC esclave nommé `kdc2` est configuré. En outre, la propagation incrémentielle est configurée. Cette procédure utilise les paramètres de configuration ci-dessous :

- Nom de domaine = `EXAMPLE.COM`
- Nom de domaine DNS = `example.com`
- KDC maître = `kdc1.example.com`
- KDC esclave = `kdc2.example.com`
- `admin` principal = `kws/admin`

Avant de commencer Le KDC maître est configuré. Si cet esclave doit être échangeable, suivez les instructions de la section “Echange d'un KDC maître et d'un KDC esclave” à la page 134.

Vous devez prendre le rôle `root` sur le serveur KDC. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. **Sur le KDC maître, démarrez `kadmin`.**

   Vous devez vous connecter à l'aide de l'un des noms de principal `admin` que vous avez créé lors de la configuration du KDC maître.

```
kdc1 # /usr/sbin/kadmin -p kws/admin
Enter password: xxxxxxxx
kadmin:
```

   Pour plus d'informations, reportez-vous à la page de manuel `kadmin`(1M).

   a. **Sur le KDC maître, ajoutez des principaux d'hôtes esclaves à la base de données, si ce n'est pas déjà fait.**

Pour que l'esclave fonctionne, il doit avoir un hôte principal. Notez que lorsque l'instance de principal est un nom d'hôte, le nom de domaine complet (FQDN) doit être spécifié en minuscules, quelle que soit la casse du nom de domaine dans le service de noms.

```
kadmin: addprinc -randkey host/kdc2.example.com
Principal "host/kdc2.example.com@EXAMPLE.COM" created.
kadmin:
```

b. Sur le maître KDC, créez le principal pour la propagation incrémentielle.

Le principal `kiprop` est utilisé pour autoriser la propagation incrémentielle à partir du KDC maître.

```
kadmin: addprinc -randkey kiprop/kdc2.example.com
Principal "kiprop/kdc2.example.com@EXAMPLE.COM" created.
kadmin:
```

c. Quitter `kadmin`.

```
kadmin: quit
```

2. Sur le KDC maître, modifiez le fichier de configuration Kerberos`krb5.conf`.

Vous devez ajouter une entrée pour chaque esclave. Pour plus d'informations sur ce fichier, reportez-vous à la page de manuel `krb5.conf`(4).

```
kdc1 # pfedit /etc/krb5/krb5.conf
.
.
[realms]
EXAMPLE.COM = {
kdc = kdc1.example.com
kdc = kdc2.example.com
admin_server = kdc1.example.com
}
```

3. Sur le KDC maître, ajoutez une entrée `kiprop` au fichier `kadm5.acl`.

Cette entrée permet au KDC maître de recevoir des demandes de propagation incrémentielle pour le serveur `kdc2`.

```
kdc1 # pfedit /etc/krb5/kadm5.acl
*/admin@EXAMPLE.COM *
kiprop/kdc2.example.com@EXAMPLE.COM p
```

4. Sur le KDC maître, redémarrez le service `kadmind` pour utiliser les nouvelles entrées dans le fichier `kadm5.acl`.

```
kdc1 # svcadm restart network/security/kadmin
```

5. Sur tous les KDC esclaves, copiez les fichiers d'administration KDC à partir du serveur KDC maître.

Chaque esclave KDC doit contenir des informations à jour sur les serveur KDC maître. Vous pouvez utiliser `ftp` ou tout autre mécanisme de transfert similaire pour obtenir des copies des fichiers suivants à partir du KDC maître :

- `/etc/krb5/krb5.conf`
- `/etc/krb5/kdc.conf`

6. **Sur tous les KDC esclaves, ajoutez une entrée pour le KDC maître et le KDC esclave dans le fichier de configuration de propagation de base de données, `kpropd.acl`.**

   Ces informations doivent être mises à jour sur tous les serveurs KDC esclaves.

```
kdc2 # pfedit /etc/krb5/kpropd.acl
host/kdc1.example.com@EXAMPLE.COM
host/kdc2.example.com@EXAMPLE.COM
```

7. **Sur tous les KDC esclaves, assurez-vous que le fichier d'ACL Kerberos, `kadm5.acl`, n'est pas renseigné.**

   Un fichier `kadm5.acl` non modifié ressemble à l'exemple suivant :

```
kdc2 # pfedit /etc/krb5/kadm5.acl
*/admin@___default_realm___ *
```

   Si le fichier contient des entrées `kiprop`, supprimez-les.

8. **Sur le nouvel esclave, définissez son intervalle d'interrogation dans le fichier `kdc.conf`.**

   Remplacez l'entrée`sunw_dbprop_master_ulogsize` par une entrée qui définit l'intervalle d'interrogation de l'esclave. L'entrée suivante définit la durée d'interrogation sur deux minutes :

```
kdc1 # pfedit /etc/krb5/kdc.conf
[kdcdefaults]
kdc_ports = 88,750

[realms]
EXAMPLE.COM= {
profile = /etc/krb5/krb5.conf
database_name = /var/krb5/principal
acl_file = /etc/krb5/kadm5.acl
kadmind_port = 749
max_life = 8h 0m 0s
max_renewable_life = 7d 0h 0m 0s
sunw_dbprop_enable = true
sunw_dbprop_slave_poll = 2m
}
```

9. **Sur le nouvel esclave, démarrez la commande `kadmin`.**

Connectez-vous à l'aide de l'un des noms de principal `admin` créés lors de la configuration du KDC maître.

```
kdc2 # /usr/sbin/kadmin -p kws/admin
Enter password: xxxxxxxx
kadmin:
```

a. Ajoutez l'hôte principal de l'esclave au fichier keytab de l'esclave en utilisant `kadmin`.

Cette entrée permet à la commande `kprop` et à d'autres applications utilisant Kerberos de fonctionner. Notez que lorsque l'instance de principal est un nom d'hôte, le nom de domaine complet (FQDN) doit être spécifié en minuscules, quelle que soit la casse du nom de domaine dans le service de noms. Pour plus d'informations, reportez-vous à la page de manuel kprop(1M).

```
kadmin: ktadd host/kdc2.example.com
Entry for principal host/kdc2.example.com with kvno 3, encryption type AES-256 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal host/kdc2.example.com with kvno 3, encryption type AES-128 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal host/kdc2.example.com with kvno 3, encryption type Triple DES cbc
mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:
```

b. Ajoutez le principal `kiprop` au fichier keytab du KDC esclave.

L'ajout du principal `kiprop` au fichier `krb5.keytab` permet à la commande `kpropd` de s'authentifier elle-même lors du lancement de la propagation incrémentielle.

```
kadmin: ktadd kiprop/kdc2.example.com
Entry for principal kiprop/kdc2.example.com with kvno 3, encryption type AES-256 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal kiprop/kdc2.example.com with kvno 3, encryption type AES-128 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal kiprop/kdc2.example.com with kvno 3, encryption type Triple DES cbc
mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:
```

c. Quitter `kadmin`.

```
kadmin: quit
```

10. Sur le nouvel esclave, démarrez le démon de propagation Kerberos.

```
kdc2 # svcadm enable network/security/krb5_prop
```

11. Sur le nouvel esclave, créer un fichier stash à l'aide de la commande `kdb5_util`.

```
kdc2 # /usr/sbin/kdb5_util stash
kdb5_util: Cannot find/read stored master key while reading master key
```

```
kdb5_util: Warning: proceeding without master key

Enter KDC database master key: xxxxxxxx
```

Pour plus d'informations, reportez-vous à la page de manuel kdb5_util(1M).

12. **Synchroniser l'horloge de ce système avec d'autres horloges dans le domaine à l'aide du NTP ou d'un autre mécanisme.**

    Pour une authentification réussie, chaque horloge doit être réglée sur l'heure par défaut définie dans la section `libdefaults` du fichier `krb5.conf`. Pour plus d'informations, reportez-vous à la page de manuel krb5.conf(4). Pour plus d'informations sur le NTP (Network Time Protocol), reportez-vous à la section “Synchronisation des horloges entre les KDC et les clients Kerberos” à la page 132.

13. **Sur le nouvel esclave, démarrer le démon KDC.**

```
kdc2 # svcadm enable network/security/krb5kdc
```

Étapes suivantes Revenir à la NTP KDC maître KDC après la le serveur KDC maître à un client NTP sur le serveur NTP.

▼ Configuration du KDC maître pour utiliser un serveur d'annuaires LDAP

Cette procédure utilise les paramètres de configuration ci-dessous :

- Nom de domaine = `EXAMPLE.COM`
- Nom de domaine DNS = `example.com`
- KDC maître = `kdc1.example.com`
- Serveur d'annuaire = `dsserver.example.com`
- `admin` principal = `kws/admin`
- FMRI pour le service LDAP = `svc:/application/sun/ds:ds--var-opt-SUNWdsee-dsins1`
- URL de l'aide en ligne = `http://docs.oracle.com/cd/E23824_01/html/821-1456/aadmin-23.html`

Remarque - Ajuster l'URL de sorte qu'il pointe vers l'emplacement de l'aide en ligne, comme décrit dans “Interface graphique gkadmin” à la page 157.

Avant de commencer

L'hôte soit configuré pour utiliser DNS. Pour de meilleures performances, installez le KDC et le service d'annuaire LDAP sur le même serveur. En outre, un Serveur d'annuaire doit être en cours d'exécution. La procédure ci-dessous fonctionne avec des serveurs utilisant Oracle Directory Server Enterprise Edition. Pour plus d'informations, reportez-vous à Oracle Identity Management - Documentation.

Vous devez prendre le rôle `root` sur le serveur KDC. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Configurez le KDC maître afin qu'il utilise SSL pour atteindre le Serveur d'annuaire.**

   Les étapes suivantes permettent de configurer le KDC afin qu'il utilise le certificat autosigné du Serveur d'annuaire.

   a. **Sur le Serveur d'annuaire, exportez le certificat autosigné.**

   ```
   # /export/sun-ds6.1/ds6/bin/dsadm show-cert -F der /export/sun-ds6.1/directory2 \
   defaultCert > /tmp/defaultCert.cert.der
   ```

   b. **Sur le KDC maître, importez le certificat du Serveur d'annuaire.**

   ```
   # pktool setpin keystore=nss dir=/var/ldap
   # chmod a+r /var/ldap/*.db
   # pktool import keystore=nss objtype=cert trust="CT" \
   infile=/tmp/defaultCert.cert.der \
   label=defaultCert dir=/var/ldap
   ```

   Pour plus d'informations, reportez-vous à la page de manuel `pktool`(1).

   c. **Sur le KDC maître, vérifiez que SSL fonctionne.**

   Cet exemple suppose que l'entrée `cn=directory manager` dispose de privilèges d'administration.

   ```
   master# /usr/bin/ldapsearch -Z -P /var/ldap -D "cn=directory manager" \
   -h dsserver.example.com -b "" -s base objectclass='*'
   Subject:
   "CN=dsserver.example.com,CN=636,CN=Directory Server,O=Example Corporation
   ```

   Notez que l'entrée `CN=dsserver.example.com` doit inclure le nom d'hôte complet, pas une version abrégée.

2. **Renseignez l'annuaire LDAP, si nécessaire.**

3. **Ajoutez le schéma Kerberos au schéma existant de LDAP.**

   ```
   # ldapmodify -h dsserver.example.com -D "cn=directory manager" \
   -f /usr/share/lib/ldif/kerberos.ldif
   ```

4. **Créez le conteneur Kerberos dans l'annuaire LDAP.**

   Ajoutez les entrées suivantes au fichier `krb5.conf`.

   a. **Définissez le type de base de données.**

      Ajoutez une entrée pour définir le `database_module` sur la section `realms`.

      ```
      database_module = LDAP
      ```

   b. **Définissez le module de base de données.**

      ```
      [dbmodules]
      LDAP = {
      ldap_kerberos_container_dn = "cn=krbcontainer,dc=example,dc=com"
      db_library = kldap
      ldap_kdc_dn = "cn=kdc service,ou=profile,dc=example,dc=com"
      ldap_kadmind_dn = "cn=kadmin service,ou=profile,dc=example,dc=com"
      ldap_cert_path = /var/ldap
      ldap_servers = ldaps://dsserver.example.com
      }
      ```

   c. **Créez le KDC dans l'annuaire LDAP.**

      Cette commande crée `krbcontainer` et plusieurs autres objets. Elle crée également une clé principale `/var/krb5/.k5.EXAMPLE.COM` et un fichier stash pour la clé. Pour plus d'informations sur les options de la commande, reportez-vous à la page de manuel kdb5_ldap_util(1M).

      ```
      # kdb5_ldap_util -D "cn=directory manager" create
         -P master-key -r EXAMPLE.COM -s
      ```

5. **Dissimulez les mots de passe KDC de liaison DN (Distinguished Name, nom distinctif).**

   Ces mots de passe sont utilisés par le KDC lorsqu'il se connecte au serveur d'annuaire. Le KDC utilise des rôles différents en fonction du type d'accès utilisé par le KDC.

   ```
   # kdb5_ldap_util stashsrvpw "cn=kdc service,ou=profile,dc=example,dc=com"
   # kdb5_ldap_util stashsrvpw "cn=kadmin service,ou=profile,dc=example,dc=com"
   ```

6. **Ajoutez des rôles de service KDC.**

   a. **Créez un fichier `kdc_roles.ldif` avec un contenu comme suit :**

      ```
      dn: cn=kdc service,ou=profile,dc=example,dc=com
      cn: kdc service
      sn: kdc service
      objectclass: top
      objectclass: person
      userpassword: xxxxxxxx
      ```

```
dn: cn=kadmin service,ou=profile,dc=example,dc=com
cn: kadmin service
sn: kadmin service
objectclass: top
objectclass: person
userpassword: xxxxxxxx
```

b. Créez des entrées de rôle dans l'annuaire LDAP

```
# ldapmodify -a -h dsserver.example.com -D "cn=directory manager" -f kdc_roles.ldif
```

7. Définissez les ACL pour les rôles liés à `kadmin`.

```
# cat << EOF | ldapmodify -h dsserver.example.com -D "cn=directory manager"
# Set kadmin ACL for everything under krbcontainer.
dn: cn=krbcontainer,dc=example,dc=com
changetype: modify
add: aci
aci: (target="ldap:///cn=krbcontainer,dc=example,dc=com")(targetattr="krb*")(version 3.0;\
acl kadmin_ACL; allow (all)\
userdn = "ldap:///cn=kadmin service,ou=profile,dc=example,dc=com";)

# Set kadmin ACL for everything under the people subtree if there are
# mix-in entries for krb princs:
dn: ou=people,dc=example,dc=com
changetype: modify
add: aci
aci: (target="ldap:///ou=people,dc=example,dc=com")(targetattr="krb*")(version 3.0;\
acl kadmin_ACL; allow (all)\
userdn = "ldap:///cn=kadmin service,ou=profile,dc=example,dc=com";)
EOF
```

8. Edition du fichier de configuration Kerberos, `krb5.conf`.

Vous devez attribuer un nom au les domaines (realms) et sur les serveurs. Pour plus d'informations sur ce fichier, reportez-vous à la page de manuel `krb5.conf`(4).

```
kdc1 # pfedit /etc/krb5/krb5.conf
[libdefaults]
default_realm = EXAMPLE.COM

[realms]
EXAMPLE.COM = {
kdc = kdc1.example.com
admin_server = kdc1.example.com
}

[domain_realm]
.example.com = EXAMPLE.COM
#
# if the domain name and realm name are equivalent,
# this entry is not needed
#
```

```
[logging]
default = FILE:/var/krb5/kdc.log
kdc = FILE:/var/krb5/kdc.log

[appdefaults]
gkadmin = {
help_url = http://docs.oracle.com/cd/E23824_01/html/821-1456/aadmin-23.html
}
```

Remarque - Ajuster l'URL de sorte qu'il pointe vers l'emplacement de l'aide en ligne, comme décrit dans “Interface graphique `gkadmin`” à la page 157.

Dans cet exemple, les lignes pour `default_realm`, `kdc`, `admin_server` et toutes les entrées `domain_realm` ont été modifiées. En outre, l'aide en ligne URL a été modifié.

Remarque - Si vous devez communiquer à l'aide d'une version antérieure, vous aurez sans doute besoin Kerberos pour limiter les types de chiffrement. Pour une description des problèmes liés à la restriction des types de chiffrement, reportez-vous à la section “Types de chiffrement Kerberos” à la page 54.

9. **Editez le fichier de configuration KDC, `kdc.conf`.**

   Vous devez attribuer un nom au domaine. Pour plus d'informations sur ce fichier, reportez-vous à la page de manuel `kdc.conf`(4).

   Dans cet exemple, en plus de la définition du nom de domaine, l'administrateur modifie la propagation incrémentielle et erreurs de journalisation.

```
kdc1 # pfedit /etc/krb5/kdc.conf
[kdcdefaults]
kdc_ports = 88,750

[realms]
EXAMPLE.COM = {
profile = /etc/krb5/krb5.conf
database_name = /var/krb5/principal
acl_file = /etc/krb5/kadm5.acl
kadmind_port = 749
max_life = 8h 0m 0s
max_renewable_life = 7d 0h 0m 0s
sunw_dbprop_enable = true
sunw_dbprop_master_ulogsize = 1000
}
```

Remarque - Si vous devez communiquer à l'aide d'une version antérieure, vous aurez sans doute besoin Kerberos pour limiter les types de chiffrement. Pour une description des problèmes liés à la restriction des types de chiffrement, reportez-vous à la section “Types de chiffrement Kerberos” à la page 54.

10. Editer le fichier de la liste de contrôle d'accès Kerberos, `kadm5.acl`.

Une fois renseigné, le fichier `/etc/krb5/kadm5.acl` doit contenir tous les noms de principaux autorisés à administrer le KDC.

```
kws/admin@EXAMPLE.COM   *
```

L'entrée précédente donne au principal `kws/admin` du domaine `EXAMPLE.COM` la possibilité de modifier les principaux et les stratégies dans le KDC. La valeur par défaut de l'entrée du principal est un astérisque (*), qui renvoie à tous les principaux `admin`. Cette entrée peut se présenter un risque en matière de sécurité. Modifiez le fichier de façon explicite afin de pouvoir répertorier chaque principal `admin` ainsi que leurs droits. Pour plus d'informations, reportez-vous à la page de manuel `kadm5.acl`(4).

11. Lancez la commande `kadmin.local` et créez les principaux `admin`.

```
kdc1 # /usr/sbin/kadmin.local
kadmin.local:
```

a. Ajoutez des principaux d'administration à la base de données.

Vous pouvez ajouter autant de principaux `admin` que nécessaire. Vous devez créer au moins un principal `admin` pour terminer le processus de configuration du KDC. Pour cet exemple, vous créez le principal `kws/admin`. Vous pouvez remplacer "`kws`" par le nom de principal approprié.

```
kadmin.local: addprinc kws/admin
Enter password for principal kws/admin@EXAMPLE.COM:     /** Type strong password **/
Re-enter password for principal kws/admin@EXAMPLE.COM: xxxxxxxx
Principal "kws/admin@EXAMPLE.COM" created.
kadmin.local:
```

b. Quittez `kadmin.local`.

```
kadmin.local: quit
```

12. (Facultatif) Configurez les dépendances LDAP pour les services Kerberos.

Si les serveurs LDAP et KDC sont en cours d'exécution sur le même hôte et si le service LDAP est configuré avec SMF, ajoutez une dépendance au service LDAP pour les démons Kerberos. Cette dépendance redémarre le service KDC si le service LDAP est redémarré.

a. Ajoutez la dépendance au service `krb5kdc`.

```
# svccfg -s security/krb5kdc
svc:/network/security/krb5kdc> addpg dsins1 dependency
svc:/network/security/krb5kdc> setprop dsins1/entities = \
fmri: "svc:/application/sun/ds:ds--var-opt-SUNWdsee-dsins1"
svc:/network/security/krb5kdc> setprop dsins1/grouping = astring: "require_all"
svc:/network/security/krb5kdc> setprop dsins1/restart_on = astring: "restart"
```

```
svc:/network/security/krb5kdc> setprop dsins1/type = astring: "service"
svc:/network/security/krb5kdc> exit
```

b. Ajoutez la dépendance au service `kadmin`.

```
# svccfg -s security/kadmin
svc:/network/security/kadmin> addpg dsins1 dependency
svc:/network/security/kadmin> setprop dsins1/entities =\
fmri: "svc:/application/sun/ds:ds--var-opt-SUNWdsee-dsins1"
svc:/network/security/kadmin> setprop dsins1/grouping = astring: "require_all"
svc:/network/security/kadmin> setprop dsins1/restart_on = astring: "restart"
svc:/network/security/kadmin> setprop dsins1/type = astring: "service"
svc:/network/security/kadmin> exit
```

13. **Terminez la configuration Kerberos dans LDAP en effectuant l'Étape 7 à Étape 9 dans “Configuration manuelle d'un KDC maître” à la page 81.**

14. **Configurez les KDC esclaves.**
Pour assurer la redondance, veillez à installer au moins un KDC esclave. Pour obtenir des instructions, reportez-vous à la section “Configuration manuelle d'un KDC esclave” à la page 86.

Remplacement des clés TGS sur un serveur maître

Remarque - Remplacer les clés lorsque vous souhaitez utiliser un nouveau type de chiffrement renforcés pour toutes les clés de session.

Lorsque le service d'octroi de tickets (TGS DES) Principal n'a qu'une clé, la clé restreint le type de chiffrement du ticket d'octroi de tickets (TGT DES) clé de session. Lorsque le KDC est mis à jour pour une version qui prend en charge les types de chiffrement renforcés, vous devez remplacer le TGS DES clé du principal de manière à ce que le principal peut générer pour toutes les clés de session de chiffrement supérieur.

Vous pouvez remplacer ou la clé à distance sur le serveur maître. Vous devez être un principal `admin` auquel est assigné le privilège `changepw`.

- Pour remplacer la clé de principal du service TGS à partir de n'importe quel système Kerberos, utilisez la commande `kadmin`.

```
kdc1 % /usr/sbin/kadmin -p kws/admin
Enter password: xxxxxxxx
kadmin: cpw -randkey krbtgt/EXAMPLE.COM@EXAMPLE.COM
Enter TGS key: xxxxxxxx
Enter new TGS key:      /** Type strong password **/
Re-enter TGS key to verify: xxxxxxxx
```

`cpw` est un alias pour la commande `change_password`. L'option `-randkey` vous invite à saisir le nouveau mot de passe.

- Si vous n'êtes pas connecté au KDC maître en tant que `root`, vous pouvez utiliser la commande `kadmin.local`. Vous êtes invité à saisir le nouveau mot de passe de base de données.

```
kdc1 # kadmin.local -q 'cpw -randkey krbtgt/EXAMPLE.COM@EXAMPLE.COM'
```

Remarque - Enregistrer et stocker ce mot de passe dans un endroit sûr.

Gestion d'un KDC sur un serveur d'annuaire LDAP

La plupart des tâches d'administration du KDC qui utilisent un serveur d'annuaire LDAP sont identiques à celles du serveur DB2. Certaines nouvelles tâches sont spécifiques à l'utilisation de LDAP.

TABLEAU 4-4 Configuration des serveurs KDC pour l'utilisation de la liste des tâches LDAP

Tâche	Description	Pour obtenir des instructions
Configuration d'un KDC maître.	Configure et construit le serveur KDC maître et une base de données pour un domaine à l'aide d'un processus manuel et à l'aide de LDAP avec le KDC.	“Configuration du KDC maître pour utiliser un serveur d'annuaires LDAP” à la page 90
Association des attributs de principaux Kerberos aux types de classe objet non Kerberos.	Permet de partager les informations stockées avec les enregistrements Kerberos avec d'autres bases de données LDAP.	“Association des attributs de principaux Kerberos dans un type de classe d'objet non Kerberos” à la page 97
Destruction d'un domaine.	Supprime toutes les données associées à un domaine.	“Suppression d'un domaine d'un serveur d'annuaire LDAP” à la page 98

▼ Association des attributs de principaux Kerberos dans un type de classe d'objet non Kerberos

Dans cette procédure, les attributs `krbprincipalaux`, `krbTicketPolicyAux` et `krbPrincipalName` sont associés à la classe d'objet "personnes".

Cette procédure utilise les paramètres de configuration ci-dessous :

- Serveur d'annuaire = `dsserver.example.com`
- principal d'utilisateur = `mre@EXAMPLE.COM`

Avant de commencer

Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. **Préparez chaque entrée dans les classes d'objet personnes.**

   Sur le serveur d'annuaire, répétez cette étape pour chaque entrée.

   ```
   cat << EOF | ldapmodify -h dsserver.example.com -D "cn=directory manager"
   dn: uid=mre,ou=people,dc=example,dc=com
   changetype: modify
   objectClass: krbprincipalaux
   objectClass: krbTicketPolicyAux
   krbPrincipalName: mre@EXAMPLE.COM
   EOF
   ```

2. **Ajoutez un attribut de sous-arborescence pour le conteneur de domaine.**

   Cet exemple permet d'effectuer des recherches d'entrées de principal dans le conteneur `ou=people,dc=example,dc=com`, ainsi que le conteneur `EXAMPLE.COM` par défaut.

   ```
   # kdb5_ldap_util -D "cn=directory manager" modify \
      -subtrees 'ou=people,dc=example,dc=com' -r EXAMPLE.COM
   ```

3. **(Facultatif) Si les enregistrements de KDC sont stockés dans DB2, migrez les entrées DB2.**

   a. **Videz les entrées DB2.**

      ```
      # kdb5_util dump > dumpfile
      ```

   b. **Chargez la base de données dans le serveur LDAP.**

      ```
      # kdb5_util load -update dumpfile
      ```

4. **(Facultatif) Ajoutez les attributs de principal au KDC.**

   ```
   # kadmin.local -q 'addprinc mre'
   ```

▼ Suppression d'un domaine d'un serveur d'annuaire LDAP

Cette procédure peut être utilisée si un autre serveur d'annuaire LDAP a été configuré pour gérer un domaine.

Avant de commencer Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

- **Supprimez le domaine.**

  ```
  # kdb5_ldap_util -D "cn=directory manager" destroy
  ```

Configuration des clients Kerberos

Les clients Kerberos incluent tout hôte sur le réseau qui n'est pas un serveur KDC et qui doit utiliser des services Kerberos. Cette section décrit les procédures d'installation d'un client Kerberos, ainsi que des informations sur l'utilisation de l'authentification `root` pour monter des systèmes de fichiers NFS.

Les options de configuration sont similaires aux client les options de serveur, à la différence près que le programme d'installation automatisée (AI), procédez comme suit :

- AI rapidement et facilement-recommandé pour l'installation de plusieurs clients Kerberos
- Automatique - recommandé pour les scripts
- Interactif - suffisant pour la plupart des installations
- Manuel - nécessaire pour des installations plus complexes

La liste ci-dessous décrit les tâches traitées dans cette section.

TABLEAU 4-5 Configuration d'une liste des tâches des clients Kerberos

Tâche	Description	Pour obtenir des instructions
L'installation de clients à l'aide du programme d'installation automatisée (AI).	Kerberos recommandés lorsque vous souhaitez que le client soit configuré lors de l'installation du serveur.	“ Configuration des clients Kerberos à l’aide de l’AI ” du manuel “ Installation des systèmes Oracle Solaris 11.2 ”
Un profil d'installation du même type créer les clients Kerberos.	Crée un profil d'installation client réutilisables.	“Création d'un profil d'installation de client Kerberos” à la page 100
Les clients d'installation d'un script.	Lorsque les paramètres d'installation appropriées sont identiques pour tous les clients.	“Configuration automatique d'un client Kerberos” à la page 100
Les clients d'installation en répondant aux invites.	Si seul un petit nombre approprié des paramètres d'installation doivent être modifiés.	“Configuration interactive d'un client Kerberos” à la page 102
Les clients d'installation manuellement.	Lorsque chaque installation de client requiert appropriée des paramètres d'installation uniques.	“Configuration manuelle d'un client Kerberos” à la page 105
Un client Kerberos à une jointure Active Directory Server.	Installe automatiquement un client Kerberos d'un serveur Active Directory.	“Connexion d'un client Kerberos à un serveur Active Directory” à la page 104
Désactivation de la vérification du KDC qui a émis un TGT (Ticket Granting Ticket) client	La vérification du KDC Kerberos rationalise les clients n'ont pas lors d'un principal d'hôte stocké dans le fichier keytab local.	“Désactivation de la vérification du ticket d'octroi de tickets” à la page 111
Activation d'un client pour qu'il puisse accéder à un système de fichier NFS en tant qu'utilisateur `root`	Permet au client de monter un système de fichiers NFS avec accès `root`. Active également le NFS au client d'accéder au système de fichiers de sorte que les tâches cron puissent s'exécuter.	“Accès à un système de fichiers NFS protégé par Kerberos en tant qu'utilisateur `root`” à la page 111

Création d'un profil d'installation de client Kerberos

Cette procédure permet de créer un profil kclient à utiliser lorsque vous installez un client Kerberos. L'utilisation du profil vous permet de réduire les risques de faute de frappe. En outre, le profil permet de réduire l'intervention de l'utilisateur par rapport au processus interactif.

Remarque - Pour créer des systèmes qui sont initialisés en tant que clients Kerberos entièrement configurés, reportez-vous à la section “ Configuring Security ” dans “ Installation des systèmes Oracle Solaris 11.2 Systems ”.

Avant de commencer Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. **Créez un profil d'installation `kclient`.**

   Vous trouverez ci-dessous un exemple de profil`kclient` :

   ```
   client# pfedit kcprofile
   REALM EXAMPLE.COM
   KDC kdc1.example.com
   ADMIN clntconfig
   FILEPATH /net/denver.example.com/export/install/krb5.conf
   NFS 1
   DNSLOOKUP none
   ```

2. **Protéger le fichier et conservez-la pour sera utilisée par d'autres clients.**

   ```
   client# cp kcprofile /net/denver.example.com/export/install
   denver# chown root kcprofile; chmod 644 kcprofile
   ```

Configuration automatique d'un client Kerberos

Avant de commencer Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. **Créer un profil kclient.**

   Utilisez le profil d'installation à partir de la section“Création d'un profil d'installation de client Kerberos” à la page 100

2. **Exécutez la commande`kclient` avec un argument de profil.**

   Vous devez fournir le mot de passe pour le principal `clntconfig` afin de terminer le processus. Vous avez créé ce mot de passe lorsque vous avez configuré le KDC maître dans la section

“Configuration des serveurs KDC” à la page 75. Pour plus d'informations, reportez-vous à la page de manuel `kclient`(1M).

```
client# /usr/sbin/kclient -p /net/denver.example.com/export/install/kcprofile

Starting client setup
-----------------------------------------------------

kdc1.example.com

Setting up /etc/krb5/krb5.conf.

Obtaining TGT for clntconfig/admin ...
Password for clntconfig/admin@EXAMPLE.COM: xxxxxxxx

nfs/client.example.com entry ADDED to KDC database.
nfs/client.example.com entry ADDED to keytab.

host/client.example.com entry ADDED to KDC database.
host/client.example.com entry ADDED to keytab.

Copied /net/denver.example.com/export/install/krb5.conf.

-----------------------------------------------------
Setup COMPLETE.

client#
```

Exemple 4-2 Utilisation d'un profil d'installation pour configurer un client Kerberos

L'exemple suivant utilise le profil de client `kcprofile` et le remplacement de deux lignes de commande pour configurer le client.

```
# /usr/sbin/kclient -p /net/denver.example.com/export/install/kcprofile \
-d dns_fallback -k kdc2.example.com

Starting client setup
-----------------------------------------------------

kdc1.example.com

Setting up /etc/krb5/krb5.conf.

Obtaining TGT for clntconfig/admin ...
Password for clntconfig/admin@EXAMPLE.COM:  xxxxxxxx

nfs/client.example.com entry ADDED to KDC database.
nfs/client.example.com entry ADDED to keytab.

host/client.example.com entry ADDED to KDC database.
host/client.example.com entry ADDED to keytab.

Copied /net/denver.example.com/export/install/krb5.conf.
```

```
------------------------------------------------
Setup COMPLETE.

client#
```

Configuration interactive d'un client Kerberos

Cette procédure utilise l'utilitaire d'installation `kclient` sans profil d'installation. Si le client doit se connecter à un serveur Active Directory, passez à la section “Connexion d'un client Kerberos à un serveur Active Directory” à la page 104.

Avant de commencer Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. **Exécutez la commande `kclient` sans arguments.**

   ```
   client# /usr/sbin/kclient
   ```

   Le script vous invite à entrer les informations suivantes :

   - Nom de domaine Kerberos
   - Nom d'hôte KDC maître
   - Nom d'hôte KDC esclave
   - Domaines à mettre en correspondance avec le domaine local
   - Noms de service PAM et options à utiliser pour l'authentification Kerberos

   Pour plus d'informations, reportez-vous à la page de manuel `kclient`(1M).

2. **Si le serveur KDC n'exécute pas une version Oracle Solaris, répondez `y` et définissez le type de serveur qui exécute le KDC.**

   Pour obtenir la liste des serveurs disponibles, reportez-vous à l'option `-T` dans la page de manuel `kclient`(1M).

3. **Si DNS doit être utilisé pour les recherches Kerberos, répondez `y` et indiquez l'option de recherche DNS à utiliser.**

   Les options valides sont `dns_lookup_kdc`, `dns_lookup_realm` et `dns_fallback`. Pour plus d'informations sur ces valeurs, reportez-vous à la page de manuel `krb5.conf`(4).

4. **Définissez le nom du domaine Kerberos et le nom d'hôte du KDC maître.**

   Ces informations sont ajoutées au fichier de configuration `/etc/krb5/krb5.conf`.

5. **Si des KDC esclaves sont dans le domaine , répondez `y` et fournissez les noms d'hôte KDC esclaves.**
   Ces informations sont utilisées pour créer d'autres entrées KDC dans le fichier de configuration du client.

6. **Si des clés de service ou d'hôte sont obligatoires, répondez `y`.**
   En général, aucune clé de service ou d'hôte n'est requise, sauf si le système client héberge des services utilisant Kerberos.

7. **Si le client est membre d'un cluster, répondez `y` et fournissez le nom logique du cluster.**
   Le nom d'hôte logique est utilisé lorsque vous créez les clés de service, ce qui est requis lors de l'hébergement de services Kerberos de clusters.

8. **Identifiez les domaines ou hôtes à mapper avec le domaine actuel.**
   Ce mappage permet à d'autres domaines d'appartenir au domaine par défaut du client.

9. **Indiquez si le client utilisera le NFS utilisant Kerberos.**
   Les clés de service NFS doivent être créées si le client héberge des services NFS utilisant Kerberos.

10. **Indiquez si une nouvelle stratégie PAM doit être créée.**
    Les services de définir l'utilisation de Kerberos pour l'authentification PAM, vous fournissez le nom du service et un indicateur signalant comment l'authentification Kerberos est utilisée. Les options d'indicateur valides sont les suivantes :

    - `first` : utilisez d'abord l'authentification Kerberos, puis utilisez uniquement UNIX si l'authentification Kerberos échoue
    - `only` : utilisez uniquement l'authentification Kerberos
    - `optional` : utilisez l'authentification Kerberos de manière optionnelle

    Pour plus d'informations sur les services PAM fournis pour Kerberos, examinez les fichiers dans `/etc/security/pam_policy`.

11. **Indiquez si le fichier maître `/etc/krb5/krb5.conf` doit être copié.**
    Cette option permet d'utiliser des informations de configuration spécifiques lorsque les arguments de `kclient` ne sont pas suffisants.

Exemple 4-3 Exemple d'exécution du script `kclient`

```
...
Starting client setup
-------------------------------------------------
```

```
Is this a client of a non-Solaris KDC ? [y/n]: n
No action performed.
Do you want to use DNS for kerberos lookups ? [y/n]: n
No action performed.
Enter the Kerberos realm: EXAMPLE.COM
Specify the KDC host name for the above realm: kdc1.example.com

Note, this system and the KDC's time must be within 5 minutes of each other for
Kerberos to function. Both systems should run some form of time synchronization
system like Network Time Protocol (NTP).
Do you have any slave KDC(s) ? [y/n]: y
Enter a comma-separated list of slave KDC host names: kdc2.example.com

Will this client need service keys ? [y/n]: n
No action performed.
Is this client a member of a cluster that uses a logical host name ? [y/n]: n
No action performed.
Do you have multiple domains/hosts to map to realm ? [y/n]: y
Enter a comma-separated list of domain/hosts to map to the default
 realm: corphdqtrs.example.com, \
example.com

Setting up /etc/krb5/krb5.conf.

Do you plan on doing Kerberized nfs ? [y/n]: y
Do you want to update /etc/pam.conf ? [y/n]: y
Enter a comma-separated list of PAM service names in the following format:
service:{first|only|optional}: xscreensaver:first
Configuring /etc/pam.conf.

Do you want to copy over the master krb5.conf file ? [y/n]: n
No action performed.

-------------------------------------------------
Setup COMPLETE.
```

▼ Connexion d'un client Kerberos à un serveur Active Directory

Cette procédure utilise la commande `kclient` sans profil d'installation.

Avant de commencer

Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. **(Facultatif) Activez la création d'enregistrements de ressource DNS pour le client.**

```
client# sharectl set -p ddns_enable=true smb
```

2. **Exécutez la commande `kclient`.**

   La sortie suivante montre un exemple de sortie à partir de l'exécution de la commande `kclient` pour connecter le client au domaine AD, `EXAMPLE.COM`.

   L'option `-T` permet de sélectionner un type de serveur KDC, dans ce cas, un type de serveur Microsoft Active Directory (AD) Par défaut, vous devez saisir le mot de passe du principal d'administration du serveur AD.

```
client# /usr/sbin/kclient -T ms_ad
Starting client setup
-------------------------------------------------

Attempting to join 'CLIENT' to the 'EXAMPLE.COM' domain.
Password for Administrator@EXAMPLE.COM: xxxxxxxx
Forest name found: example.com
Looking for local KDCs, DCs and global catalog servers (SVR RRs).

Setting up /etc/krb5/krb5.conf

Creating the machine account in AD via LDAP.
-------------------------------------------------
Setup COMPLETE.
#
```

   Pour plus d'informations, reportez-vous à la page de manuel `kclient`(1M).

▼ Configuration manuelle d'un client Kerberos

Cette procédure utilise les paramètres de configuration ci-dessous :

- Nom de domaine = `EXAMPLE.COM`
- Nom de domaine DNS = `example.com`
- KDC maître = `kdc1.example.com`
- KDC esclave = `kdc2.example.com`
- Serveur NFS = `denver.example.com`
- Client = `client.example.com`
- `admin` principal = `kws/admin`
- Utilisateur principal = `mre`
- URL de l'aide en ligne = `http://docs.oracle.com/cd/E23824_01/html/821-1456/aadmin-23.html`

Avant de commencer

Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Editez le fichier de configuration Kerberos, `krb5.conf`.**

   Modifier les noms de domaine et les noms de serveur dans le fichier de configuration Kerberos. Vous pouvez également indiquer le chemin d'accès aux fichiers d'aide pour `gkadmin`.

```
kdc1 # pfedit /etc/krb5/krb5.conf
[libdefaults]
default_realm = EXAMPLE.COM

[realms]
EXAMPLE.COM = {
kdc = kdc1.example.com
kdc = kdc2.example.com
admin_server = kdc1.example.com
}

[domain_realm]
.example.com = EXAMPLE.COM
#
# if the domain name and realm name are equivalent,
# this entry is not needed
#
[logging]
default = FILE:/var/krb5/kdc.log
kdc = FILE:/var/krb5/kdc.log

[appdefaults]
gkadmin = {
help_url = http://www.example.com/doclib/OSMKA/aadmin-23.html
```

   Remarque - Si vous devez communiquer à l'aide d'une version antérieure, vous aurez sans doute besoin Kerberos pour limiter les types de chiffrement. Pour une description des problèmes liés à la restriction des types de chiffrement, reportez-vous à la section “Types de chiffrement Kerberos” à la page 54.

2. **(Facultatif) Modifiez le processus utilisé pour localiser les KDC.**

   Par défaut, le domaine Kerberos pour le mappage KDC est effectué dans l'ordre suivant :

   - Définition de la section `realms` dans `krb5.conf`.
   - Recherche d'enregistrements SRV dans le DNS.

   Vous pouvez modifier ce comportement en ajoutant `dns_lookup_kdc` ou `dns_fallback` à la section `libdefaults` du fichier `krb5.conf`. Pour plus d'informations, reportez-vous à `krb5.conf`(4). Notez que les références sont toujours tentées en premier.

3. **(Facultatif) Modifiez le processus utilisé pour déterminer le domaine d'un hôte.**

   Par défaut, le mappage de l'hôte au domaine est déterminé dans l'ordre suivant :

- Si le KDC prend en charge les références, le KDC peut indiquer au client le domaine auquel appartient l'hôte.
- La définition de `domain_realm` dans le fichier `krb5.conf`.
- Le nom de domaine DNS de l'hôte.
- Le domaine par défaut.

Vous pouvez modifier ce comportement en ajoutant `dns_lookup_kdc` ou `dns_fallback` à la section `libdefaults` du fichier `krb5.conf`. Pour plus d'informations, reportez-vous à la page de manuel `krb5.conf`(4). Notez que les références sont toujours tentées en premier.

4. Synchronisez l'horloge du client avec l'horloge du KDC maître à l'aide de NTP ou d'un autre mécanisme de synchronisation d'horloge.

Pour une authentification réussie, chaque horloge doit être synchronisée avec l'heure du serveur KDC ne pouvant dépasser un écart maximum défini dans la relation `clockskew` dans le fichier `krb5.conf`. Pour plus d'informations, reportez-vous à la page de manuel `krb5.conf`(4). Pour plus d'informations sur le NTP (Network Time Protocol), reportez-vous à la section “Synchronisation des horloges entre les KDC et les clients Kerberos” à la page 132.

5. Créer les principaux Kerberos.

```
denver # /usr/sbin/kadmin -p kws/admin
Enter password: xxxxxxxx
kadmin:
```

Pour plus d'informations, reportez-vous à la page de manuel `kadmin`(1M).

a. (Facultatif) Créez un principal d'utilisateur s'il n'en existe pas déjà.

Vous devez créer un principal d'utilisateur uniquement si aucun principal n'est affecté à l'utilisateur associé à cet hôte.

```
kadmin: addprinc mre
Enter password for principal mre@EXAMPLE.COM:     /** Type strong password **/
Re-enter password for principal mre@EXAMPLE.COM: xxxxxxxx
kadmin:
```

b. (Facultatif) Créez un principal `root` et ajoutez le principal au fichier keytab du serveur.

Remarque - Si le client ne nécessite pas d'accès `root` à un système de fichiers monté via NFS distant, vous pouvez ignorer cette étape.

Si l'accès `root` non interactif est nécessaire, par exemple pour l'exécution des tâches cron en tant qu'utilisateur `root`, réalisez cette étape.

Le principal `root` doit être un principal à deux composants. Le second composant doit être le nom d'hôte du système du client Kerberos pour éviter la création d'un principal `root` à l'échelle du domaine. Lorsque l'instance de principal est un nom d'hôte, le nom de domaine complet (FQDN) doit être spécifié en minuscules, quelle que soit la casse du nom de domaine dans le service de noms.

```
kadmin: addprinc -randkey root/client.example.com
Principal "root/client.example.com" created.
kadmin: ktadd root/client.example.com
Entry for principal root/client.example.com with kvno 3, encryption type AES-256 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal root/client.example.com with kvno 3, encryption type AES-128 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal root/client.example.com with kvno 3, encryption type Triple DES cbc
mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:
```

c. Créez un principal `host` et ajoutez le principal au fichier keytab du serveur.

Le principal `host` est utilisé par les services d'accès à distance pour fournir l'authentification. Le principal permet à `root` d'acquérir des informations d'identification, si elles n'existent pas déjà dans le fichier keytab.

```
kadmin: addprinc -randkey host/denver.example.com
Principal "host/denver.example.com@EXAMPLE.COM" created.
kadmin: ktadd host/denver.example.com
Entry for principal host/denver.example.com with kvno 3, encryption type AES-256 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal host/denver.example.com with kvno 3, encryption type AES-128 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal host/denver.example.com with kvno 3, encryption type Triple DES cbc
mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:
```

d. (Facultatif) Ajoutez le principal de service du serveur NFS au fichier keytab du serveur.

Cette étape n'est requise que si le client a besoin d'accéder aux systèmes de fichiers NFS à l'aide de l'authentification Kerberos.

```
kadmin: ktadd nfs/denver.example.com
Entry for principal nfs/denver.example.com with kvno 3, encryption type AES-256 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal nfs/denver.example.com with kvno 3, encryption type AES-128 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal nfs/denver.example.com with kvno 3, encryption type Triple DES cbc
mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:
```

e. Quitter `kadmin`.

```
kadmin: quit
```

6. **(Facultatif) Activez Kerberos avec NFS.**

   a. **Activez les modes de sécurité Kerberos dans le fichier `/etc/nfssec.conf`.**

   Dans le fichier`/etc/nfssec.conf`, supprimez le "#" qui met en commentaire les modes de sécurité Kerberos.

```
# pfedit /etc/nfssec.conf
.
.
#
# Uncomment the following lines to use Kerberos V5 with NFS
#
krb5            390003  kerberos_v5     default -               # RPCSEC_GSS
krb5i           390004  kerberos_v5     default integrity       # RPCSEC_GSS
krb5p           390005  kerberos_v5     default privacy         # RPCSEC_GSS
```

   b. **Activez DNS.**

   Si le service `svc:/network/dns/client:default` n'est pas activé, activez-le. Pour plus d'informations, reportez-vous à la page de manuel `resolv.conf`(4).

```
# svcadm enable network/dns/client:default
```

   c. **Redémarrez le service `gss`.**

```
# svcadm restart network/rpc/gss
```

7. **(Facultatif) Pour que le client renouvelle automatiquement le TGT ou pour avertir les utilisateurs concernant l'expiration du ticket Kerberos, créez une entrée dans le fichier `/etc/krb5/warn.conf`.**

   Pour plus d'informations, reportez-vous à la page de manuel `warn.conf`(4) et à la section “Renouvellement automatique de tous les tickets d'octroi de tickets” à la page 117.

Exemple 4-4 Configuration d'un client Oracle Solaris pour une utilisation avec un KDC à plusieurs maîtres

Le service Kerberos Microsoft Active Directory (AD) fournit un KDC qui s'exécute sur plusieurs serveurs maîtres. Pour qu'un client Oracle Solaris mette à jour les informations, `admin_server` ou la déclaration `kpasswd_server` dans le fichier `/etc/krb5/krb5.conf` doit reprendre tous les serveurs. Cet exemple montre comment autoriser le client à mettre à jour les informations concernant le KDC partagé par `kdc1` et `kdc2`.

```
[realms]
EXAMPLE.COM = {
kdc = kdc1.example.com
kdc = kdc2.example.com
admin_server = kdc1.example.com
```

```
admin_server = kdc2.example.com
}
```

Exemple 4-5 Configuration d'un client Kerberos pour un KDC non-Oracle Solaris

Un client Kerberos peut être configuré pour travailler avec un KDC non Oracle Solaris, en ajoutant une ligne au fichier `/etc/krb5/krb5.conf` dans la section `realms`. Cette ligne change le protocole utilisé lorsque le client est en cours de communication avec le serveur de changement de mot de passe Kerberos. L'extrait suivant montre le format de cette ligne.

```
[realms]
EXAMPLE.COM = {
kdc = kdc1.example.com
kdc = kdc2.example.com
admin_server = kdc1.example.com
kpasswd_protocol = SET_CHANGE
}
```

Exemple 4-6 Enregistrements DNS TXT pour le mappage de l'hôte et du nom de domaine au domaine Kerberos

```
@ IN SOA kdc1.example.com root.kdc1.example.com (
1989020501   ;serial
10800        ;refresh
3600         ;retry
3600000      ;expire
86400 )      ;minimum

IN      NS      kdc1.example.com.
kdc1                    IN      A       192.146.86.20
kdc2                    IN      A       192.146.86.21

_kerberos.example.com.             IN      TXT     "EXAMPLE.COM"
_kerberos.kdc1.example.com.        IN      TXT     "EXAMPLE.COM"
_kerberos.kdc2.example.com.        IN      TXT     "EXAMPLE.COM"
```

Exemple 4-7 Enregistrements DNS SRV pour les emplacements de serveur Kerberos

Cet exemple définit les enregistrements pour l'emplacement des KDC, du serveur `admin` et du serveur `kpasswd`, respectivement.

```
@ IN SOA kdc1.example.com root.kdc1.example.com (
1989020501   ;serial
10800        ;refresh
3600         ;retry
3600000      ;expire
86400 )      ;minimum

IN      NS      kdc1.example.com.
kdc1                               IN      A       192.146.86.20
kdc2                               IN      A       192.146.86.21
```

```
_kerberos._udp.EXAMPLE.COM         IN      SRV 0 0 88  kdc2.example.com
_kerberos._tcp.EXAMPLE.COM         IN      SRV 0 0 88  kdc2.example.com
_kerberos._udp.EXAMPLE.COM         IN      SRV 1 0 88  kdc1.example.com
_kerberos._tcp.EXAMPLE.COM         IN      SRV 1 0 88  kdc1.example.com
_kerberos-adm._tcp.EXAMPLE.COM     IN      SRV 0 0 464 kdc1.example.com
_kpasswd._udp.EXAMPLE.COM          IN      SRV 0 0 464 kdc1.example.com
```

Désactivation de la vérification du ticket d'octroi de tickets

Par défaut, Kerberos vérifie que le KDC du principal d'hôte stocké dans le fichier local `/etc/krb5/krb5.keytab` file est identique au KDC qui a émis le ticket d'octroi de tickets (TGT). Ce contrôle, `verify_ap_req_nofail`, empêche les attaques par usurpation de DNS.

Toutefois, cette vérification doit être désactivée afin que les données de configuration client dans lequel l'hôte principal n'est pas disponible. Vérifiez les configurations suivantes a besoin de celui-ci à désactiver :

- Le client adresse IP est affectée de manière dynamique DHCP, par exemple, un client.
- Le client n'est pas configuré pour héberger les services, de sorte qu'aucun hôte principal n'a été créé.
- La clé d'hôte n'est pas stockée sur le client.

Pour désactiver la vérification TGT, réglez l'option `verify_ap_req_nofail` sur `false` dans le fichier `krb5.conf`. L'option `verify_ap_req_nofail` peut être saisie dans la section `[libdefaults]` ou la section `[realms]` du fichier `krb5.conf`. Dans le section `[libdefaults]`, le paramètre est utilisé pour tous les domaines :

```
client # pfedit /etc/krb5/krb5.conf
[libdefaults]
default_realm = EXAMPLE.COM
verify_ap_req_nofail = false
...
```

Si l'option se trouve dans la section `[realms]`, le paramètre s'applique uniquement au domaine défini. Pour plus d'informations sur cette option, reportez-vous à la page de manuel `krb5.conf`(4).

▼ Accès à un système de fichiers NFS protégé par Kerberos en tant qu'utilisateur `root`

Cette procédure permet à un client d'accéder à un système de fichiers NFS requérant l'authentification Kerberos avec le privilège d'ID `root`. En particulier, lorsque le système de fichiers NFS est partagé avec des options comme `-o sec=krb5,root=client1.example.com`.

1. **Exécutez la commande `kadmin`.**

```
denver # /usr/sbin/kadmin -p kws/admin
Enter password: xxxxxxxx
kadmin:
```

2. **Créez un principal `root` pour le client NFS.**

   Ce principal permet de fournir un accès équivalent à root aux systèmes de fichiers montés NFS requérant l'authentification Kerberos. Le principal `root` doit être un principal à deux composants. Le second composant doit être le nom d'hôte du système du client Kerberos, pour éviter la création d'un principal root à l'échelle du domaine (realm). Notez que lorsque l'instance de principal est un nom d'hôte, le nom de domaine complet (FQDN) doit être spécifié en minuscules, quelle que soit la casse du nom de domaine dans le service de noms.

```
kadmin: addprinc -randkey root/client.example.com
Principal "root/client.example.com" created.
kadmin:
```

3. **Ajoutez le principal `root` au fichier keytab du serveur.**

   Cette étape est nécessaire au client pour disposer d'un accès `root` aux systèmes de fichiers montés à l'aide du service NFS. Cette étape est également nécessaire pour un accès `root` non interactif, par exemple pour l'exécution des tâches cron en tant que `root`.

```
kadmin: ktadd root/client.example.com
Entry for principal root/client.example.com with kvno 3, encryption type AES-256 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal root/client.example.com with kvno 3, encryption type AES-128 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal root/client.example.com with kvno 3, encryption type Triple DES cbc
mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:
```

4. **Quitter `kadmin`.**

```
kadmin: quit
```

▼ Configuration de la migration automatique des utilisateurs dans un domaine Kerberos

Les utilisateurs ne disposant pas d'un principal Kerberos peuvent être automatiquement migrés vers un domaine Kerberos existant à l'aide de PAM. PAM vous personnalisez les fichiers de configuration par système sur le serveur le serveur de migration et du UNIX pour gérer et la reconnaissance des l'authentifier à nouveau les informations d'identification et de connexion dans le domaine Kerberos.

Pour plus d'informations sur le PAM, reportez-vous à la section Chapitre 1, Utilisation de modules d'authentification enfichables et à la page de manuel `pam.conf`(4).

Dans cette procédure, les noms de service de connexion sont configurés en vue de l'utilisation de la migration. Cet exemple utilise les paramètres de configuration suivants :

- Nom de domaine = `EXAMPLE.COM`
- KDC maître = `kdc1.example.com`
- Machine hébergeant le service de migration = `server1.example.com`
- Principal du service de migration = `host/server1.example.com`

Avant de commencer Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Assurez-vous qu'un principal de service d'hôte pour `server1` existe.**

   L'hôte principal de service dans le fichier `keytab` de `server1` est utilisé pour authentifier le serveur auprès du KDC maître.

   ```
   server1 # klist -k
   Keytab name: FILE:/etc/krb5/krb5.keytab
   KVNO Principal
   ---- -------------------------------------------------
   3 host/server1.example.com@EXAMPLE.COM
   ...
   ```

   Pour plus d'informations sur les options de la commande, reportez-vous à la page de manuel `klist`(1).

2. **Si `server1` n'est pas répertorié, configurez-le en tant que client Kerberos du domaine `EXAMPLE.COM`.**

   Reportez-vous aux exemples dans la section "Configuration des clients Kerberos" à la page 99 pour connaître la procédure à suivre.

3. **Modifiez la stratégie PAM pour `server1`.**

   Pour plus d'informations, reportez-vous à la section "Affectation d'une stratégie PAM par utilisateur" à la page 22.

   a. **Déterminez la stratégie Kerberos utilisée sur `server1`.**

      ```
      % grep PAM_POLICY /etc/security/policy.conf
      # PAM_POLICY specifies the system-wide PAM policy (see pam_user_policy(5))
      ...
      PAM_POLICY=krb5_first
      ```

b. Copiez ce fichier de stratégie PAM, puis modifiez le nouveau fichier de stratégie pour ajouter le module `pam_krb5_migrate.so.1` à chaque pile d'authentification.

```
server1 # cd /etc/security/pam_policy/; cp krb5_first krb5_firstmigrate
server1 # pfedit /etc/security/pam_policy/krb5_firstmigrate.
# login service (explicit because of pam_dial_auth)
#
login auth requisite    pam_authtok_get.so.1
...
login auth required  pam_unix_auth.so.1
login    auth optional  pam_krb5_migrate.so.1
#
# rlogin service (explicit because of pam_rhost_auth)
#
rlogin auth sufficient  pam_rhosts_auth.so.1
...
rlogin auth required    pam_unix_auth.so.1
rlogin   auth optional        pam_krb5_migrate.so.1
#
# Kerberized rlogin service
#
krlogin auth required  pam_unix_cred.so.1
krlogin auth required  pam_krb5.so.1
krlogin auth optional  pam_krb5_migrate.so.1
#
# rsh service (explicit because of pam_rhost_auth)
#
rsh auth sufficient  pam_rhosts_auth.so.1
rsh auth required    pam_unix_cred.so.1
rsh auth optional    pam_krb5_migrate.so.1
#
# Kerberized rsh service
#
krsh auth required  pam_unix_cred.so.1
krsh auth required  pam_krb5.so.1
krsh auth optional  pam_krb5_migrate.so.1
#
# Kerberized telnet service
#
ktelnet auth required  pam_unix_cred.so.1
ktelnet auth required  pam_krb5.so.1
ktelnet auth optional  pam_krb5_migrate.so.1
#
# PPP service (explicit because of pam_dial_auth)
#
ppp auth requisite    pam_authtok_get.so.1
...
ppp auth required  pam_unix_auth.so.1
ppp auth optional  pam_krb5_migrate.so.1
#
# GDM Autologin (explicit because of pam_allow).  These need to be
```

```
# here as there is no mechanism for packages to amend pam.conf as
# they are installed.
#
gdm-autologin auth  required    pam_unix_cred.so.1
gdm-autologin auth  sufficient  pam_allow.so.1
gdm-autologin auth optional    pam_krb5_migrate.so.1
#
# Default definitions for Authentication management
# Used when service name is not explicitly mentioned for authentication
#
OTHER auth requisite    pam_authtok_get.so.1
...
OTHER auth required  pam_unix_auth.so.1
OTHER auth optional  pam_krb5_migrate.so.1
#
# passwd command (explicit because of a different authentication module)
#
passwd auth required  pam_passwd_auth.so.1
#
# cron service (explicit because of non-usage of pam_roles.so.1)
#
cron account required pam_unix_account.so.1
#
# cups service (explicit because of non-usage of pam_roles.so.1)
#
cups account required pam_unix_account.so.1
#
# GDM Autologin (explicit because of pam_allow) This needs to be here
# as there is no mechanism for packages to amend pam.conf as they are
# installed.
#modified
gdm-autologin account  sufficient  pam_allow.so.1
#
.
.
.
```

c. (Facultatif) Forcez une modification immédiate du mot.

Pour les nouveaux comptes Kerberos, définissez le délai d'expiration du mot de passe à l'heure actuelle en ajoutant l'option `expire_pw` aux entrées `pam_krb5_migrate`. Pour plus d'informations, reportez-vous à la page de manuel `pam_krb5_migrate`(5).

```
service-name auth optional     pam_krb5_migrate.so.1 expire_pw
```

d. Dans ce fichier de configuration, modifiez la pile de compte `OTHER` afin de bloquer l'accès si le mot de passe Kerberos est arrivé à expiration.

```
# Definition for Account management
# Used when service name is not explicitly mentioned for account management
# Re-ordered pam_krb5 causes password expiration in Kerberos to block access
#
```

```
OTHER account requisite   pam_roles.so.1
OTHER account required pam_krb5.so.1
OTHER account required pam_unix_account.so.1
OTHER account required pam_tsol_account.so.1
# OTHER account required pam_krb5.so.1
#
.
.
.
```

e. Modifiez l'entrée `PAM_POLICY` dans le fichier `policy.conf` pour utiliser le fichier de configuration modifié.

```
server1 # pfedit /etc/security/policy.conf
...
# PAM_POLICY=krb5_first
PAM_POLICY=krb5_firstmigrate
```

Pour plus d'informations, consultez le fichier `policy.conf`.

4. Sur le KDC maître, mettez à jour le fichier de contrôle d'accès `kadm5.acl`.

Les entrées suivantes accordent des privilèges de consultation et de migration au principal de service `host/server1.example.com` pour tous les utilisateurs à l'exception de l'utilisateur `root`. Utilisez le privilège `U` pour dresser la liste des utilisateurs ne devant pas faire l'objet d'une migration. Ces entrées doivent précéder l'entrée "permit all" ou `ui`. Pour plus d'informations, reportez-vous à la page de manuel`kadm5.acl`(4).

```
kdc1 # pfedit /etc/krb5/kadm5.acl
host/server1.example.com@EXAMPLE.COM U root
host/server1.example.com@EXAMPLE.COM ui *
*/admin@EXAMPLE.COM *
```

5. Sur le KDC maître, activez le démon `kadmind` pour utiliser le service PAM `k5migrate`.

Si un fichier de service `k5migrate` ne figure pas dans le répertoire `/etc/pam.d`, ajoutez le fichier de service au répertoire. Pour plus d'informations, reportez-vous à la page de manuel `pam.d`(4).

La validation de cette modification UNIX permet au mots de passe utilisateur pour les comptes qui nécessitent une migration.

```
kdc1 # pfedit /etc/pam.d/k5migrate
...
# Permits validation of migrated UNIX accounts
auth    required        pam_unix_auth.so.1
account required        pam_unix_account.so.1
```

Remarque - `k5migrate` est le nom d'un service PAM. Le fichier doit être appelé `k5migrate`.

6. **Tester la configuration. avant d'insérer-le dans l'environnement de production.**

   - **En tant qu'utilisateur standard, service PAM tester chaque modification.**
   - **En tant que `root`, testez chaque service PAM modifié.**
   - **Forcer la modification du mot de passe modifié PAM, puis testez les services.**

Renouvellement automatique de tous les tickets d'octroi de tickets

Pour une plus grande facilité d'administration, vous pouvez configurer le renouvellement des tickets et les messages d'avertissement à propos de l'expiration Ticket Granting Ticket (TGT). Les administrateurs peuvent définir des avertissements et servant de seuil pour tous les utilisateurs, et les utilisateurs puissent personnaliser leur propre des avertissements. Pour plus d'informations, reportez-vous aux pages de manuel `warn.conf`(4) et `ktkt_warnd`(1M).

Remarque - Le service `ktkt_warn` est désactivé par défaut. Pour activer le service sur des clients Kerberos existants, exécutez la commande `svcadm enable ktkt_warn`.

EXEMPLE 4-8 Configuration des messages d'expiration de TGT pour tous les utilisateurs

L'exemple ci-après illustre plusieurs méthodes de configuration du renouvellement et du système de message pour les TGT.

```
# pfedit /etc/krb5/warn.conf
##
## renew the TGT 30 minutes before expiration and send message to users terminal
##
mre@EXAMPLE.COM renew:log terminal 30m
##
## send a warning message to a specific email address 20 minutes before TGT expiration
##
mre@EXAMPLE.COM mail 20m mre@example2.com
##
# renew the TGT 20 minutes before expiration and send an email message on failure
##
bricker@EXAMPLE.COM renew:log-failure mail 20m -
##
## catch-all: any principal not matched above will get an email warning
* mail 20m -
```

Après avoir configuré les messages, exécutez la commande `kclient` sur les nouveaux clients.

```
client# /usr/sbin/kclient -p /net/denver.example.com/export/install/kcprofile
```

Sur les clients existants, activez le service.

```
# svcadm enable network/security/ktkt_warn
```

EXEMPLE 4-9 Configuration des messages d'expiration de TGT pour un utilisateur

Chaque utilisateur peut configurer un fichier de configuration `warnd` individuel, appelé `/var/user/`*`$USER`*`/krb-warn.conf`. L'existence de ce fichier empêche la lecture du fichier de l'administrateur.

```
% pfedit /var/user/mre/krb-warn.conf
mre@EXAMPLE.COM renew:log mail 25m &
```

Le TGT est renouvelé 25 minutes avant l'expiration, le renouvellement est consigné et l'utilisateur Kerberos `mre` envoie un courrier électronique à ce moment.

Configuration des serveurs d'application réseau Kerberos

Les serveurs d'application réseau sont des hôtes fournissant un accès à l'aide d'une ou plusieurs applications réseau parmi les suivantes : `ftp`, `rcp`, `rlogin`, `rsh`, `ssh`, and `telnet`. Seules quelques étapes sont nécessaires pour activer la version Kerberos de ces applications sur un serveur.

▼ Configuration d'un serveur d'application réseau Kerberos

Cette procédure utilise les paramètres de configuration ci-dessous :

- Serveur d'application = `boston`
- `admin` principal = `kws/admin`
- Nom de domaine DNS = `example.com`
- Nom de domaine = `EXAMPLE.COM`

Avant de commencer

Le KDC maître est configuré. Les horloges sont synchronisées, comme décrit dans la section “Synchronisation des horloges entre les KDC et les clients Kerberos” à la page 132. Pour tester complètement le processus, vous avez besoin de plusieurs clients.

Vous devez prendre le rôle `root` sur le serveur d'application. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. **Déterminer si un principal d'hôte existe pour le nouveau serveur.**

   La commande suivante indique l'existence de l'hôte principal :

   ```
   boston # klist -k | grep host
   4 host/boston.example.com@EXAMPLE.COM
   4 host/boston.example.com@EXAMPLE.COM
   4 host/boston.example.com@EXAMPLE.COM
   4 host/boston.example.com@EXAMPLE.COM
   ```

   Si la commande ne renvoie un principal, vous avez terminé. Si elle ne renvoie pas de principal, créez de nouveaux principaux en suivant les étapes ci-dessous.

2. **Connectez-vous au serveur en utilisant un des noms de principal `admin` créé lors de la configuration du KDC maître.**

   ```
   boston # /usr/sbin/kadmin -p kws/admin
   Enter password: xxxxxxxx
   kadmin:
   ```

3. **Créez l'`host` principal du serveur.**

   ```
   kadmin: addprinc -randkey host/boston.example.com
   Principal "host/boston.example.com" created.
   kadmin:
   ```

   L'`host` principal est utilisé dans les cas suivants :

   - Pour authentifier le trafic lors de l'utilisation des commandes à distance, comme `rsh` et `ssh`.
   - Par `pam_krb5` afin d'empêcher les attaques par mystification de KDC en utilisant l'`host` principal pour vérifier que les informations d'identification Kerberos d'un utilisateur ont été obtenues auprès d'un KDC de confiance.
   - Pour autoriser l'utilisateur `root` à acquérir automatiquement des informations d'identification Kerberos en l'absence d'un principal `root`. Cette fonctionnalité peut être utile lors d'un montage NFS manuel où le partage requiert des informations d'identification Kerberos.

   Ce principal est obligatoire si le trafic qui utilise l'application distante doit être authentifié à l'aide du service Kerberos. Si le serveur a plusieurs noms d'hôte associés, créez un principal à l'aide de la pour chaque formulaire FQDN nom d'hôte du nom d'hôte.

4. **Ajoutez l'`host` principal du serveur au fichier keytab du serveur.**

   Si la commande `kadmin` n'est pas en cours d'exécution, redémarrez-la avec une commande similaire à la suivante : `/usr/sbin/kadmin -p` *kws*`/admin`

   Si le serveur a plusieurs noms d'hôte associés, ajoutez un principal au fichier keytab pour chaque nom d'hôte.

   ```
   kadmin: ktadd host/boston.example.com
   Entry for principal host/boston.example.com with kvno 3, encryption type AES-256 CTS mode
   ```

```
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal host/boston.example.com with kvno 3, encryption type AES-128 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal host/boston.example.com with kvno 3, encryption type Triple DES cbc
mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:
```

5. **Quitter `kadmin`.**

```
kadmin: quit
```

▼ Utilisation du service de sécurité générique avec Kerberos lors de l'exécution FTP

Le service de sécurité générique (GSS) peut être utilisé par des applications de réseau Kerberos pour l'authentification, l'intégrité et la confidentialité. Les étapes suivantes montrent comment activer le service GSS pour ProFTPD.

Avant de commencer Vous devez prendre le rôle `root` sur le serveur FTP. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Ajoutez des principaux pour le serveur FTP et créez le fichier keytab du serveur FTP.**

   Ces étapes ne sont peut-être pas nécessaires si les modifications ont été effectuées précédemment.

   a. **Démarrez la commande `kadmin`.**

   ```
   ftpserver1 # /usr/sbin/kadmin -p kws/admin
   Enter password: xxxxxxxx
   kadmin:
   ```

   b. **Ajoutez le principal de service `ftp` pour le serveur FTP.**

   ```
   kadmin: addprinc -randkey ftp/ftpserver1.example.com
   ```

   c. **Ajoutez le principal de service `ftp` à un nouveau fichier keytab.**

   Un nouveau fichier keytab permet à ces informations d'être disponibles pour le service ftp, sans exposer l'ensemble des informations dans le fichier keytab du serveur.

   ```
   kadmin: ktadd -k /etc/krb5/ftp.keytab ftp/ftpserver1.example.com
   ```

   Pour plus d'informations, reportez-vous à la commande `ktadd` dans la page de manuel`kadmin`(1M).

2. **Modifiez la propriété du nouveau fichier keytab.**

```
ftpserver1 # chown ftp:ftp /etc/krb5/ftp.keytab
```

3. **Activez GSS pour le serveur FTP.**

Apportez les modifications suivantes au fichier `/etc/proftpd.conf` .

```
# pfedit /etc/proftpd.conf
LoadModule     mod_gss.c

GSSEngine      on
GSSKeytab      /etc/krb5/ftp.keytab
```

4. **Redémarrez le serveur FTP.**

```
# svcadm restart network/ftp
```

Configuration de serveurs NFS Kerberos

Les services NFS utilisent les ID d'utilisateur (UID) UNIX pour identifier un utilisateur et ne peuvent pas utiliser directement les informations d'identification GSS. Un pour traduire les données d'identification en UID, vous devrez peut - être créer une table des informations d'identification qui mappe les informations d'identification et les UID UNIX Pour plus d'informations sur le mappage des informations d'identification par défaut, reportez-vous à la section “Mappage d'informations d'identification GSS avec des informations d'identification UNIX” à la page 70. Les procédures décrites dans cette section se concentrent sur les tâches nécessaires pour configurer un serveur Kerberos NFS, administrer la table d'informations d'identification Kerberos, et initier des modes de sécurité pour les systèmes de fichiers montés sur NFS. La liste ci-dessous décrit les tâches traitées dans cette section.

TABLEAU 4-6 Configuration de la liste des tâches des serveurs Kerberos NFS

Tâche	Description	Pour obtenir des instructions
Configuration d'un serveur NFS Kerberos	Permet à un serveur de partager un système de fichiers requérant l'authentification Kerberos.	“Configuration des serveurs NFS Kerberos” à la page 122
Créer une table des informations d'identification et de le modifier.	Pour plus d'informations sur le mappage crée une table des informations d'identification UID UNIX GSS lorsque les informations d'identification et de connexion à mappage par défaut n'est pas suffisant, puis ajoute une entrée.	“Création et modification d'une table d'informations d'identification” à la page 123
Mise en correspondance des utilisateurs UNIX à partir d'un autre domaine de gestion des identités à des informations d'identification et de connexion les UID.	Met à jour les informations de la table d'informations d'identification.	Exemple 4-10, “Ajout d'un principal à un domaine différent de la table d'informations d'identification Kerberos”
Mappage des informations d'identification entre deux domaines similaires	Mappe les UID d'un domaine à l'autre avec laquelle les domaines partagent un fichier de mots de passe.	“Mappage d'informations d'identification entre domaines” à la page 124

Tâche	Description	Pour obtenir des instructions
Partage d'un système de fichiers à l'aide de l'authentification Kerberos	Partage un système de fichiers avec des modes de sécurité de sorte que l'authentification Kerberos est requise.	“Configuration d'un environnement NFS sécurisé avec plusieurs modes de sécurité Kerberos” à la page 125

▼ Configuration des serveurs NFS Kerberos

Cette procédure utilise les paramètres de configuration ci-dessous :

- Nom de domaine = `EXAMPLE.COM`
- Nom de domaine DNS = `example.com`
- Serveur NFS = `denver.example.com`
- `admin` principal = `kws/admin`

Avant de commencer

Vous devez prendre le rôle `root` sur le serveur NFS. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

Le KDC maître est configuré. Les horloges sont synchronisées, comme décrit dans la section “Synchronisation des horloges entre les KDC et les clients Kerberos” à la page 132. Pour tester complètement le processus, vous avez besoin de plusieurs clients.

1. **Configurez le serveur NFS en tant que client Kerberos.**

   Suivez les instructions de la section “Configuration des clients Kerberos” à la page 99.

2. **Ajoutez le principal de service NFS.**

   Utilisez la commande `kadmin`.

```
denver # /usr/sbin/kadmin -p kws/admin
Enter password: xxxxxxxx
kadmin:
```

   a. **Créez le principal de service NFS.**

      Notez que lorsque l'instance de principal est un nom d'hôte, le nom de domaine complet (FQDN) doit être spécifié en minuscules, quelle que soit la casse du nom de domaine dans le service de noms.

      Répétez cette étape pour chaque interface unique sur le système susceptible d'être utilisée pour accéder aux données NFS. Si un hôte possède plusieurs interfaces avec des noms uniques, chaque nom unique doit avoir son propre principal de service NFS.

```
kadmin: addprinc -randkey nfs/denver.example.com
Principal "nfs/denver.example.com" created.
kadmin:
```

b. Ajoutez le principal de service du serveur NFS au fichier keytab du serveur.

Répétez cette étape pour chaque principal de service unique créé dans Étape 2.a.

```
kadmin: ktadd nfs/denver.example.com
Entry for principal nfs/denver.example.com with kvno 3, encryption type AES-256 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal nfs/denver.example.com with kvno 3, encryption type AES-128 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal nfs/denver.example.com with kvno 3, encryption type Triple DES cbc
mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:
```

c. Quitter `kadmin`.

```
kadmin: quit
```

3. Créez les mappages d'informations d'identification GSS spéciaux, si nécessaire.

Normalement, le service Kerberos génère des mappages appropriés entre les informations d'identification GSS et les UID UNIX. Le mappage par défaut est décrit dans la section “Mappage d'informations d'identification GSS avec des informations d'identification UNIX” à la page 70. Si le mappage par défaut est insuffisant, reportez-vous à la section “Création et modification d'une table d'informations d'identification” à la page 123 pour plus d'informations.

4. Partagez le système de fichiers NFS avec les modes de sécurité Kerberos.

Pour plus d'informations, reportez-vous à la section “Configuration d'un environnement NFS sécurisé avec plusieurs modes de sécurité Kerberos” à la page 125.

▼ Création et modification d'une table d'informations d'identification

La table d'informations d'identification `gsscred` est utilisée par un serveur NFS pour mapper les informations d'identification Kerberos à une ID utilisateur UNIX. Par défaut, la première partie du nom du principal est mappée avec un nom de connexion UNIX. Vous créez cette table si le mappage par défaut n'est pas suffisant.

Avant de commencer

Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. Assurez-vous que le mécanisme de sécurité indiqué dans `/etc/gss/gsscred.conf` est `files`.

```
# cat /etc/gss/gsscred.conf
...
#
files
#
#
# Syslog (auth.debug) a message for GSS cred to Unix cred mapping
#SYSLOG_UID_MAPPING=yes
```

2. **Créez la table d'informations d'identification à l'aide de la commande `gsscred`.**

   ```
   # gsscred -m kerberos_v5 -a
   ```

   La commande `gsscred` rassemble les informations provenant de l'ensemble des sources répertoriées avec l'entrée `passwd` dans le fichier `svc:/system/name-service/switch:default`. Si vous ne souhaitez pas que les entrées de mot de passe local soient comprises dans la table d'informations d'identification, vous pouvez supprimer temporairement l'entrée `files`. Pour plus d'informations, reportez-vous à la page de manuel `gsscred`(1M).

3. **(Facultatif) Ajoutez une entrée dans la table d'informations d'identification.**
   Par exemple, comme le rôle `root` sur le serveur NFS, ajoutez une entrée pour effectuer le mappage du principal `sandy/admin` à l'ID utilisateur `3736`. L'option `-a` ajoute l'entrée à la table d'informations d'identification.

   ```
   # gsscred -m kerberos_v5 -n sandy/admin -u 3736 -a
   ```

Exemple 4-10 Ajout d'un principal à un domaine différent de la table d'informations d'identification Kerberos

Dans cet exemple, vous utilisez un nom de domaine complet (FQDN) pour spécifier un principal sur un autre domaine.

```
# gsscred -m kerberos_v5 -n sandy/admin@EXAMPLE.COM -u 3736 -a
```

▼ Mappage d'informations d'identification entre domaines

Cette procédure assure le mappages d'informations d'identification approprié entre des domaines utilisant le même fichier de mots de passe. Dans cet exemple, les domaines `CORP.EXAMPLE.COM` et `SALES.EXAMPLE.COM` utilisent le même fichier de mot de passe. Les informations d'identification pour *username*`@CORP.EXAMPLE.COM` et *username*`@SALES.EXAMPLE.COM` sont mappées avec la même ID utilisateur.

Avant de commencer Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

- **Sur le système client, ajoutez les entrées `default_realm` et`auth_to_local_realm` au fichier `krb5.conf`.**

```
# pfedit /etc/krb5/krb5.conf
[libdefaults]
default_realm = CORP.EXAMPLE.COM
.
[realms]
CORP.EXAMPLE.COM = {
.
auth_to_local_realm = SALES.EXAMPLE.COM
.
}
```

Erreurs fréquentes Pour résoudre les éventuels problèmes de mappage d'informations d'identification, reportez-vous à la section “Observation du mappage d'informations d'identification GSS sur des informations d'identification UNIX” à la page 201.

▼ Configuration d'un environnement NFS sécurisé avec plusieurs modes de sécurité Kerberos

Cette procédure permet d'activer dans le but de sécuriser un serveur NFS, grâce aux différents modes de sécurité d'accès. Lorsqu'un client négocie une serveur le mode de sécurité, le client NFS utilise le premier mode proposée par le serveur. Ce mode est utilisé pour toutes les demandes de client suivantes du système de fichiers partagé par ce serveur.

Avant de commencer Vous devez prendre le rôle `root` sur le serveur NFS. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. **Vérifiez que le fichier keytab comporte un principal de service NFS.**

   La commande `klist` signale s'il existe un fichier keytab et affiche les principaux. Si les résultats indiquent qu'aucun fichier keytab ou principal de service NFS n'existe, vous devez vérifier que toutes les étapes décrites à la section “Configuration des serveurs NFS Kerberos” à la page 122 ont bien été effectuées dans leur totalité.

```
# klist -k
Keytab name: FILE:/etc/krb5/krb5.keytab
KVNO Principal
---- --------------------------------------------------------
3 nfs/denver.example.com@EXAMPLE.COM
3 nfs/denver.example.com@EXAMPLE.COM
3 nfs/denver.example.com@EXAMPLE.COM
3 nfs/denver.example.com@EXAMPLE.COM
```

   Pour plus d'informations, reportez-vous à la page de manuel `klist`(1).

2. **Activez les modes de sécurité Kerberos dans le fichier `/etc/nfssec.conf`.**

   Dans le fichier`/etc/nfssec.conf`, supprimez le "#" qui met en commentaire les modes de sécurité Kerberos.

```
# pfedit /etc/nfssec.conf
.
.
#
# Uncomment the following lines to use Kerberos V5 with NFS
#
krb5            390003  kerberos_v5     default -               # RPCSEC_GSS
krb5i           390004  kerberos_v5     default integrity       # RPCSEC_GSS
krb5p           390005  kerberos_v5     default privacy         # RPCSEC_GSS
```

3. **Partagez les systèmes de fichiers avec les modes de sécurité appropriés.**

   `share -F nfs -o sec=`*mode file-system*

   mode — Spécifie les modes de sécurité à utiliser lors du partage du système de fichiers. Lorsque plusieurs modes de sécurité sont utilisés, le premier mode figurant dans la liste est utilisé comme valeur par défaut.

   file-system — Définit le chemin d'accès au système de fichiers à partager.

   Tous les clients tentant d'accéder à des fichiers dans le système de fichiers nommé requièrent l'authentification Kerberos. Pour accéder aux fichiers, le principal d'utilisateur sur le client NFS doit être authentifié.

4. **(Facultatif) Monter un système de fichiers à l'aide d'un mode de sécurité autre que la valeur par défaut.**

   Vous ne devez pas suivre cette procédure si la valeur par défaut mode de sécurité est acceptable.

   - **Si l'agent de montage automatique est en cours d'utilisation, modifiez la base de données `auto_master` pour saisir un mode de sécurité autre que le mode par défaut.**

     file-system `auto_home -nosuid,sec=`*mode*

   - **Emettez manuellement la commande `mount` pour accéder au système de fichiers à l'aide d'un autre mode que celui par défaut.**

     `# mount -F nfs -o sec=`*mode file-system*

Exemple 4-11 Partage d'un système de fichiers avec un mode de sécurité Kerberos

Dans cet exemple, l'authentification avec le mode de sécurité `krb5` doit être terminé pour que les fichiers puissent être accessibles via le service NFS.

```
# share -F nfs -o sec=krb5 /export/home
```

Exemple 4-12 Partage d'un système de fichiers avec plusieurs modes de sécurité Kerberos

Dans cet exemple, les trois modes de sécurité Kerberos ont été sélectionnés. Le mode utilisé est négocié entre le client et le serveur NFS. Si le premier mode dans la commande échoue, le mode suivant est essayé. Pour plus d'informations, reportez-vous à la page de manuel `nfssec`(5).

```
# share -F nfs -o sec=krb5:krb5i:krb5p /export/home
```

Configuration de l'exécution retardée pour l'accès aux services Kerberos

Dans la valeur par défaut Kerberos expirent après un environnement limité, les informations d'identification et de la période indiquée. Pour les processus qui peuvent être exécutés à des moments arbitraires, tels que `cron` et `at`, le temps limité présente un problème. Cette procédure décrit la configuration de l'environnement Kerberos permettant la prise en charge des processus d'exécution retardée qui nécessitent des services authentifiés par le biais de Kerberos. Oracle Solaris fournit des modules PAM, utilise des clés de service et utilise des options de configuration `kclient` pour effectuer une exécution retardée avec authentification Kerberos possible et plus sûre que d'autres solutions.

Remarque - Si le serveur`cron` est compromis, un individu malveillant peut usurper l'identité d'utilisateurs pour accéder aux services cibles configurés pour le serveur `cron`. Par conséquent, considérez que l'hôte `cron` est configuré dans cette procédure en tant que système plus sensible, car il dispose de services intermédiaires pour les utilisateurs.

▼ Configuration d'un hôte cron pour l'accès aux services Kerberos

Cette procédure utilise les paramètres de configuration ci-dessous :

- `cron` hôte = `host1.example.com`
- Serveur NFS = `host2.example.com`
- Serveur LDAP = `host3.example.com`

1. **Configurez le service `cron` pour prendre en charge Kerberos.**

- **Si l'hôte `cron` n'est pas configuré pour Kerberos, exécutez la commande `kclient` sur le système.**

  Pour plus d'informations, reportez-vous à la page de manuel `kclient`(1M).

  Par exemple, la commande suivante configure le client dans le domaine `EXAMPLE.COM`. La commande inclut le fichier `pam_gss_s4u` dans le fichier de service `/etc/pam.d/cron` en utilisant le mécanisme `include`.

  ```
  # kclient -s cron:optional -R EXAMPLE.COM
  ```

- **Si l'hôte`cron` est déjà configuré pour Kerberos, vous devez modifier manuellement la configuration PAM du service `cron` sur cet hôte.**

  Assurez-vous que la configuration PAM pour le service `cron` inclut le fichier `pam_gss_s4u`.

  ```
  # cd /etc/pam.d ; cp cron cron.orig
  # pfedit cron
        # PAM include file for optional set credentials
        # through Kerberos keytab and GSS-API S4U support
        auth include          pam_gss_s4u
  ```

2. **Activez l'hôte `cron` pour agir en tant que délégué.**

   Ainsi,

   ```
   # kadmin -p kws/admin
   Enter password: xxxxxxxx
   kadmin: modprinc +ok_as_delegate host/host1.example.com@EXAMPLE.COM
   Principal “host/host1.example.com@EXAMPLE.COM” modified.
   ```

3. **Activez l'hôte `cron` pour demander des tickets pour lui-même pour le compte de l'utilisateur qui a créé le travail `cron`.**

   ```
   kadmin: modprinc +ok_to_auth_as_delegate host/host1.example.com@EXAMPLE.COM
   Principal “host/host1.example.com@EXAMPLE.COM” modified.
   kadmin: quit
   ```

4. **Dans LDAP, configurez l'hôte `cron` pour indiquer les services qu'il utilise en tant que délégué.**

   Par exemple, pour permettre à l'hôte `cron` d'accéder au répertoire personnel de l'utilisateur sur `host2`, un serveur NFS utilisant Kerberos, ajoutez l'hôte NFS au paramètre `krbAllowedToDelegateTo` dans la définition LDAP du serveur `cron`.

   a. **Créer des messages par procuration affectation.**

      ```
      # pfedit /tmp/delghost.ldif
      dn: krbprincipalname=host/
      host1.example.com@EXAMPLE.COM,cn=EXAMPLE.COM,cn=krbcontainer,dc=example,dc=com
      ```

```
changetype: modify
krbAllowedToDelegateTo: nfs/host2.example.com@EXAMPLE.COM
```

b. Ajouter l'affectation à LDAP.

```
# ldapmodify -h host3 -D "cn=directory manager" -f delghost.ldif
```

Configuration de l'authentification inter-domaine

Il existe plusieurs manières de relier les domaines pour que les utilisateurs d'un domaine puissent être authentifiés dans un autre domaine. L'authentification inter-domaine est réalisée par la mise en place d'une clé secrète partagée par les deux domaines. La relation entre les domaines peut être hiérarchique ou directionnelle. Pour plus d'informations, reportez-vous à la section “Hiérarchie des domaines Kerberos” à la page 62.

▼ Etablissement de l'authentification inter-domaine hiérarchique

L'exemple de cette procédure établit une authentification inter-domaine entre `CORP.EAST.EXAMPLE.COM` et `EAST.EXAMPLE.COM` dans les deux sens. Cette procédure doit être réalisée sur le KDC maître dans les deux domaines.

Avant de commencer

Le KDC maître de chaque domaine est configuré. Pour tester l'ensemble du processus d'authentification, vous avez besoin de plusieurs clients.

Vous devez prendre le rôle `root` sur les deux serveurs KDC. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. Créez des principaux de service de TGT pour les deux domaines.

Vous devez vous connecter à l'aide de l'un des noms de principal `admin` que vous avez créé lorsque vous avez configuré le KDC maître.

```
# /usr/sbin/kadmin -p kws/admin
Enter password: xxxxxxxx
kadmin: addprinc krbtgt/CORP.EAST.EXAMPLE.COM@EAST.EXAMPLE.COM
Enter password for principal krbtgt/CORP.EAST.EXAMPLE.COM@EAST.EXAMPLE.COM:      /** Type strong
 password **/
kadmin: addprinc krbtgt/EAST.EXAMPLE.COM@CORP.EAST.EXAMPLE.COM
Enter password for principal krbtgt/EAST.EXAMPLE.COM@CORP.EAST.EXAMPLE.COM:      /** Type strong
 password **/
```

```
kadmin: quit
```

Remarque - Enregistrer et stocker ces mots de passe dans un endroit sûr.

2. **Ajoutez des entrées au fichier de configuration Kerberos pour définir les noms de domaine pour chaque domaine.**

```
# pfedit /etc/krb5/krb5.conf
[libdefaults]
.
.
[domain_realm]
.corp.east.example.com = CORP.EAST.EXAMPLE.COM
.east.example.com = EAST.EXAMPLE.COM
```

Dans cet exemple, les noms de domaine pour les domaines `CORP.EAST.EXAMPLE.COM` et `EAST.EXAMPLE.COM` sont définis. Le sous-domaine doit précéder le nom de domaine dans le fichier, car la recherche dans le fichier s'effectue du haut vers le bas.

3. **Copiez le fichier de configuration Kerberos pour tous les clients dans ce domaine.**

Pour que l'authentification inter-domaine fonctionne, tous les systèmes (y compris les KDC esclaves et les autres serveurs) doivent utiliser la version de KDC maître de `/etc/krb5/krb5.conf`.

4. **Répétez cette procédure dans le second domaine.**

Remarque - Le mot de passe que vous avez spécifié pour chaque service de principal doit être identique dans les deux KDC. Par conséquent, le mot de passe pour le principal de service `krbtgt/CORP.EAST.EXAMPLE.COM@EAST.EXAMPLE.COM` doit être identique dans les deux domaines.

▼ Etablissement de l'authentification inter-domaine directe

L'exemple de cette procédure utilise deux domaines, `CORP.EAST.EXAMPLE.COM` et `SALES.WEST.EXAMPLE.COM`. L'authentification inter-domaine est établie dans les deux directions. Cette procédure doit être effectuée sur le KDC maître dans les deux domaines.

Avant de commencer Le KDC maître de chaque domaine est configuré. Pour tester l'ensemble du processus d'authentification, vous avez besoin de plusieurs clients.

Vous devez prendre le rôle `root` sur les deux serveurs KDC. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Créez des principaux de service de TGT pour les deux domaines.**

   Vous devez vous connecter à l'aide de l'un des noms de principal `admin` que vous avez créé lorsque vous avez configuré le KDC maître.

```
# /usr/sbin/kadmin -p kws/admin
Enter password: xxxxxxxx
kadmin: addprinc krbtgt/CORP.EAST.EXAMPLE.COM@SALES.WEST.EXAMPLE.COM
Enter password for principal
krbtgt/CORP.EAST.EXAMPLE.COM@SALES.WEST.EXAMPLE.COM:      /** Type strong password **/
kadmin: addprinc krbtgt/SALES.WEST.EXAMPLE.COM@CORP.EAST.EXAMPLE.COM
Enter password for principal
krbtgt/SALES.WEST.EXAMPLE.COM@CORP.EAST.EXAMPLE.COM:      /** Type strong password **/
kadmin: quit
```

2. **Ajoutez des entrées dans le fichier de configuration Kerberos pour définir le chemin d'accès direct au domaine distant.**

   Cet exemple représente les clients dans le domaine `CORP.EAST.EXAMPLE.COM`. Pour ajouter les définitions adéquates dans le domaine `SALES.WEST.EXAMPLE.COM`, changez les noms de domaine.

```
# pfedit /etc/krb5/krb5.conf
[libdefaults]
.
.
[capaths]
CORP.EAST.EXAMPLE.COM = {
SALES.WEST.EXAMPLE.COM = .
}

SALES.WEST.EXAMPLE.COM = {
CORP.EAST.EXAMPLE.COM = .
}
```

3. **Copiez le fichier de configuration Kerberos pour tous les clients dans le domaine actuel.**

   Pour que l'authentification inter-domaine fonctionne, la nouvelle version du fichier de configuration Kerberos, `/etc/krb5/krb5.conf`, doit être installée sur tous les systèmes (y compris les KDC esclaves et les autres serveurs).

4. **Répétez cette procédure pour le second domaine.**

Synchronisation des horloges entre les KDC et les clients Kerberos

Tous les hôtes participant au système d'authentification Kerberos doivent avoir leurs horloges internes synchronisées dans une quantité maximale de temps spécifiée (appelée *écart d'horloge*). Cette exigence constitue un autre contrôle de sécurité Kerberos. Si l'écart d'horloge est dépassé entre des hôtes participants, les demandes du client sont rejetées.

L'écart d'horloge détermine également la durée pendant laquelle les serveurs d'application doivent assurer le suivi des messages du protocole Kerberos, afin de reconnaître et de rejeter les demandes rediffusées. Ainsi, plus l'écart d'horloge est élevé, plus les serveurs d'application doivent collecter d'informations.

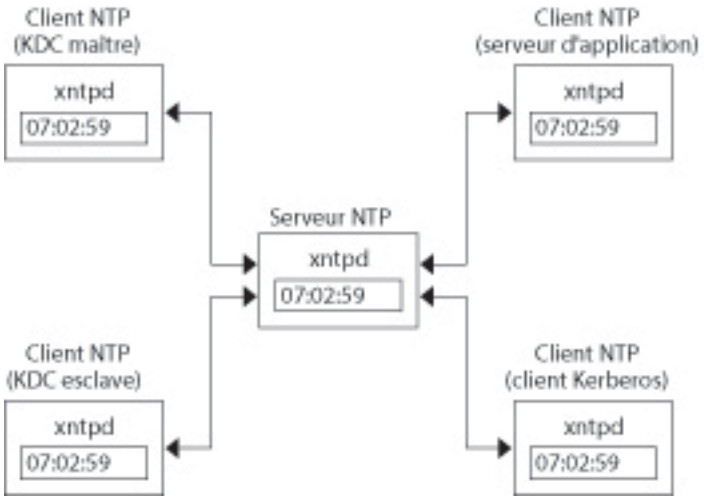
La valeur par défaut pour l'écart d'horloge maximal est de 300 secondes (5 minutes). Vous pouvez modifier cette valeur par défaut dans la section `libdefaults` du fichier `krb5.conf`.

Remarque - Pour des raisons de sécurité, l'écart d'horloge ne doit pas dépasser 300 secondes.

Dans la mesure où il est important de conserver la synchronisation des horloges entre les KDC et les clients Kerberos, utilisez le logiciel NTP (Network Time Protocol) pour les synchroniser. Le logiciel NTP du domaine public de l'Université du Delaware est inclus dans le logiciel Oracle Solaris. La documentation est disponible sur l'adresse NTP Documentation.

NTP vous permet de gérer avec précision la synchronisation de l'heure ou de l'horloge du réseau, ou les deux, dans un environnement réseau. Serveur - client NTP est un protocole. Un système est NTP l'horloge principale, le serveur. Les clients NTP tous les autres systèmes sont qui synchronisent leurs horloges avec l'horloge principale. Pour synchroniser les horloges, NTP utilise le démon `xntpd` qui définit et actualise l'heure du jour d'un système UNIX en accord avec les serveurs de temps standard Internet. Le schéma suivant montre un exemple de cette implémentation NTP serveur-client.

FIGURE 4-1 Synchronisation des horloges à l'aide de NTP



S'assurer que les KDC et les clients Kerberos maintiennent leurs horloges synchronisées implique la mise en oeuvre des étapes suivantes :

1. Configuration d'un serveur NTP sur votre réseau. Ce serveur peut être n'importe quel système, à l'exception du KDC maître.
2. Lorsque vous configurez les clients KDC et Kerberos sur le réseau, définissez-les de sorte qu'ils soient des clients NTP sur le serveur NTP. Revenir à la maître en tant que KDC NTP : si vous optez pour une configuration client.
3. L'activation du service NTP sur tous les systèmes.

Echange d'un KDC maître et d'un KDC esclave

Les procédures décrites dans cette section facilitent l'échange d'un KDC maître avec un KDC esclave. Vous devez remplacer le KDC maître par un KDC esclave uniquement si le serveur du KDC maître est en panne pour une raison quelconque, ou si le KDC maître doit être réinstallé (par exemple, en cas d'installation de nouveau matériel).

Configuration d'un KDC échangeable

Que vous puissiez effectuer la procédure suivante à l'aide du domaine de gestion des identités, c'est - à - KDC la propagation incrémentielle serveur sur le serveur esclave que vous souhaitez libérez pour devenir le KDC maître.

Avant de commencer Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Utilisez les noms d'alias pour le KDC maître et le KDC esclave échangeable pendant l'installation du KDC.**

   Lorsque vous définissez les noms d'hôte des KDC, assurez-vous que chaque système possède un alias inclus dans le DNS. Vous pouvez également utiliser les noms d'alias lorsque vous définissez les hôtes dans le fichier `/etc/krb5/krb5.conf`.

2. **Installer un KDC esclave.**

   Avant tout échange, ce serveur doit fonctionner comme n'importe quel autre KDC esclave dans le domaine. Pour plus d'informations, reportez-vous à la section "Configuration manuelle d'un KDC esclave" à la page 86.

3. **Après l'installation des commandes, déplacez le KDC maître.**

   Commandes le KDC maître ne doit pas être exécuté à partir de ce KDC esclave.

```
kdc4 # mv /usr/lib/krb5/kprop /usr/lib/krb5/kprop.save
kdc4 # mv /usr/lib/krb5/kadmind /usr/lib/krb5/kadmind.save
kdc4 # mv /usr/sbin/kadmin.local /usr/sbin/kadmin.local.save
```

Echange d'un KDC maître et d'un KDC esclave

Dans cette procédure, le serveur de KDC maître échangé est appelé `kdc1`. Le KDC esclave qui devient le nouveau KDC maître est appelé `kdc4`. Cette procédure suppose que vous utilisez la propagation incrémentielle.

Avant de commencer Terminez la procédure "Configuration d'un KDC échangeable" à la page 134.

Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Sur le nouveau KDC maître, démarrez `kadmin`.**

```
kdc4 # /usr/sbin/kadmin -p kws/admin
```

```
Enter password: xxxxxxxx
kadmin:
```

2. **Créez de nouveaux principaux pour le service `kadmind`.**

   L'exemple suivant montre la première commande `addprinc` sur deux lignes, mais elle doit être saisie sur une seule ligne.

```
kadmin: addprinc -randkey -allow_tgs_req +password_changing_service -clearpolicy \
changepw/kdc4.example.com
Principal "changepw/kdc4.example.com@EXAMPLE.COM" created.
kadmin: addprinc -randkey -allow_tgs_req -clearpolicy kadmin/kdc4.example.com
Principal "kadmin/kdc4.example.com@EXAMPLE.COM" created.
kadmin:
```

3. **Quitter `kadmin`.**

```
kadmin: quit
```

4. **Sur le nouveau KDC maître, forcez la synchronisation.**

   Les étapes suivantes forcent une mise à jour complète de KDC sur le serveur esclave.

   a. **Désactivez le service `krb5kdc` et supprimez son fichier journal.**

```
kdc4 # svcadm disable network/security/krb5kdc
kdc4 # rm /var/krb5/principal.ulog
```

   b. **Vérifiez que la mise à jour est terminée.**

```
kdc4 # /usr/sbin/kproplog -h
```

   c. **Redémarrez le service KDC.**

```
kdc4 # svcadm enable -r network/security/krb5kdc
```

   d. **Réinitialisez le journal de mise à jour pour le nouveau serveur KDC maître.**

```
kdc4 # svcadm disable network/security/krb5kdc
kdc4 # rm /var/krb5/principal.ulog
```

5. **Sur l'ancien KDC maître, arrêtez les services `kadmin` et `krb5kdc`.**

   Lorsque vous interrompez le service `kadmind`, vous empêchez toute modification apportée à la base de données KDC.

```
kdc1 # svcadm disable network/security/kadmin
kdc1 # svcadm disable network/security/krb5kdc
```

6. **Sur l'ancien KDC maître, spécifiez la durée d'interrogation pour demander les propagations.**

Mettez en commentaire l'entrée `sunw_dbprop_master_ulogsize` dans `/etc/krb5/kdc.conf` et ajoutez une entrée qui définit l'intervalle d'interrogation de l'esclave. Cette entrée définit la durée d'interrogation sur deux minutes.

```
kdc1 # pfedit /etc/krb5/kdc.conf
[kdcdefaults]
kdc_ports = 88,750

[realms]
EXAMPLE.COM= {
profile = /etc/krb5/krb5.conf
database_name = /var/krb5/principal
acl_file = /etc/krb5/kadm5.acl
kadmind_port = 749
max_life = 8h 0m 0s
max_renewable_life = 7d 0h 0m 0s
sunw_dbprop_enable = true
#  sunw_dbprop_master_ulogsize = 1000
sunw_dbprop_slave_poll = 2m
}
```

7. **Sur l'ancien KDC maître, déplacez les commandes de KDC maître et le fichier `kadm5.acl`.**

   Commandes le KDC maître ne doivent pas être exécutées à partir de l'ancien KDC maître.

   ```
   kdc1 # mv /usr/lib/krb5/kprop /usr/lib/krb5/kprop.save
   kdc1 # mv /usr/lib/krb5/kadmind /usr/lib/krb5/kadmind.save
   kdc1 # mv /usr/sbin/kadmin.local /usr/sbin/kadmin.local.save
   kdc1 # mv /etc/krb5/kadm5.acl /etc/krb5/kadm5.acl.save
   ```

8. **Sur le serveur DNS, modifiez les noms d'alias du KDC maître.**

   Pour changer de serveurs, modifiez le fichier de zone `example.com` et modifiez l'entrée pour `masterkdc`.

   ```
   masterkdc IN CNAME kdc4
   ```

9. **Serveur sur le recharger les nouvelles DNS, les informations sur les alias.**

   ```
   # svcadm refresh network/dns/server
   ```

10. **Sur le nouveau KDC maître, déplacez les commandes de KDC maître et le fichier esclave `kpropd.acl`.**

    Vous avez déplacé les commandes KDC maître dansÉtape 3 de la section“Configuration d'un KDC échangeable” à la page 134,

    ```
    kdc4 # mv /usr/lib/krb5/kprop.save /usr/lib/krb5/kprop
    kdc4 # mv /usr/lib/krb5/kadmind.save /usr/lib/krb5/kadmind
    kdc4 # mv /usr/sbin/kadmin.local.save /usr/sbin/kadmin.local
    kdc4 # mv /etc/krb5/kpropd.acl /etc/krb5/kpropd.acl.save
    ```

11. **Sur le nouveau KDC maître, créez le fichier de liste de contrôle d'accès Kerberos, `kadm5.acl`.**

    Une fois renseigné, le fichier `/etc/krb5/kadm5.acl` doit contenir tous les noms de principaux autorisés à administrer le KDC. Ce fichier doit également répertorier tous les esclaves qui peuvent émettre des demandes de propagation incrémentielle. Pour plus d'informations, reportez-vous à la page de manuel `kadm5.acl`(4).

```
kdc4 # pfedit /etc/krb5/kadm5.acl
kws/admin@EXAMPLE.COM   *
kiprop/kdc1.example.com@EXAMPLE.COM p
```

12. **Sur le nouveau KDC maître, spécifiez la taille de journal de mise à jour dans le fichier `kdc.conf`.**

    Mettez en commentaire l'entrée `sunw_dbprop_slave_poll` et ajoutez une entrée qui définit `sunw_dbprop_master_ulogsize`. Cette entrée définit la taille du journal à 1000 entrées.

```
kdc1 # pfedit /etc/krb5/kdc.conf
[kdcdefaults]
kdc_ports = 88,750

[realms]
EXAMPLE.COM= {
profile = /etc/krb5/krb5.conf
database_name = /var/krb5/principal
acl_file = /etc/krb5/kadm5.acl
kadmind_port = 749
max_life = 8h 0m 0s
max_renewable_life = 7d 0h 0m 0s
sunw_dbprop_enable = true
#               sunw_dbprop_slave_poll = 2m
sunw_dbprop_master_ulogsize = 1000
}
```

13. **Sur le nouveau KDC maître, activez les services `kadmin` et `krb5kdc`.**

```
kdc4 # svcadm enable -r network/security/krb5kdc
kdc4 # svcadm enable -r network/security/kadmin
```

14. **Sur l'ancien KDC maître, ajoutez le principal de service `kiprop`.**

    L'ajout du principal `kiprop` au fichier `krb5.keytab` permet au démon `kpropd` de s'authentifier auprès du service de propagation incrémentielle.

```
kdc1 # /usr/sbin/kadmin -p kws/admin
Authenticating as principal kws/admin@EXAMPLE.COM with password.
Enter password: xxxxxxxx
kadmin: ktadd kiprop/kdc1.example.com
Entry for principal kiprop/kdc1.example.com with kvno 3,
encryption type AES-256 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal kiprop/kdc1.example.com with kvno 3,
```

```
encryption type AES-128 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal kiprop/kdc1.example.com with kvno 3, encryption type Triple DES cbc
mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin: quit
```

15. **Sur l'ancien KDC maître, ajoutez une entrée pour chaque KDC dans le fichier `krb5.conf` au fichier de configuration de propagation.**

```
kdc1 # pfedit /etc/krb5/kpropd.acl
host/kdc1.example.com@EXAMPLE.COM
host/kdc2.example.com@EXAMPLE.COM
host/kdc3.example.com@EXAMPLE.COM
host/kdc4.example.com@EXAMPLE.COM
```

16. **Sur l'ancien KDC maître, activez les services `kpropd` et `krb5kdc`.**

```
kdc1 # svcadm enable -r network/security/krb5_prop
kdc1 # svcadm enable -r network/security/krb5kdc
```

Administration de la base de données Kerberos

La base de données Kerberos est l'épine dorsale de Kerberos et sa maintenance doit s'effectuer correctement. Cette section présente certaines procédures d'administration de la base de données Kerberos, telles que la sauvegarde et la restauration de la base de données, la définition de la propagation incrémentielle ou parallèle, ainsi que l'administration du fichier stash. Les étapes de configuration initiale de la base de données sont détaillées dans la section “Configuration manuelle d'un KDC maître” à la page 81.

Sauvegarde et propagation de la base de données Kerberos

La propagation de la base de données Kerberos du KDC maître aux KDC esclaves est l'une des tâches de configuration les plus importantes. Si la propagation n'est pas suffisamment fréquente, la synchronisation entre le KDC maître et les KDC esclaves est perdue. Par conséquent, en cas de défaillance du KDC maître, les KDC esclaves n'auront pas les informations de base de données les plus récentes. En outre, si un KDC esclave a été configuré en tant que KDC maître à des fins d'équilibrage de charge, les clients qui utilisent le KDC esclave en tant que KDC maître ne disposeront pas des dernières informations.

Assurez-vous que la propagation est suffisamment fréquente ou configurez les serveurs pour une propagation incrémentielle, en fonction de la fréquence à laquelle vous modifiez la base de données Kerberos. La propagation incrémentielle est préférable à d'autres méthodes de

propagation. Une propagation manuelle de la base de données exige la propagation complète élimine les frais d'administration liés évacués ne fonctionne pas et.

Attention - Pour éviter les pertes de données, vous devez propager manuellement la base de données si vous ajoutez des mises à jour importantes de la base de données Kerberos avant une propagation programmée.

Fichier `kpropd.acl`

Le fichier `kpropd.acl` sur un KDC esclave fournit une liste de noms d'hôte principal, un nom par ligne, qui spécifie les systèmes à partir desquels le KDC peut recevoir une base de données mise à jour par la propagation. Si le KDC maître est utilisé pour propager tous les KDC esclaves, le fichier `kpropd.acl` sur chaque esclave doit contenir uniquement le nom d'hôte principal du KDC maître.

Toutefois, l'installation Kerberos et les étapes de configuration suivantes décrites dans ce guide vous indiquent que vous devez ajouter le même fichier `kpropd.acl` sur le KDC maître et les KDC esclaves. Ce fichier contient tous les noms de principaux d'hôtes KDC. Cette configuration vous permet de propager à partir de n'importe quel KDC, dans le cas où la propagation des KDC serait temporairement indisponible. La copie identique sur tous les KDC facilite la maintenance.

Commande `kprop_script`

La commande `kprop_script` utilise la commande `kprop` pour propager la base de données Kerberos à d'autres KDC. Si la commande `kprop_script` est exécutée sur un KDC esclave, elle se propage à la copie de la base de données Kerberos du KDC esclave vers d'autres KDC. La commande `kprop_script` accepte une liste de noms d'hôte pour les arguments, séparés par des espaces, qui indiquent les KDC à propager.

Quand `kprop_script` est exécutée, elle crée une copie de sauvegarde de la base de données Kerberos pour le fichier `/var/krb5/slave_datatrans` et copie le fichier dans les KDC spécifiés. La base de données Kerberos est verrouillée jusqu'à ce que la propagation soit terminée.

Sauvegarde de la base de données Kerberos

Lorsque vous configurez le KDC maître, vous configurez la commande `kprop_script` dans une tâche `cron` pour sauvegarder automatiquement la base de données Kerberos dans le fichier dump `/var/krb5/slave_datatrans` et la propager vers les KDC esclaves. Mais, comme avec n'importe quel fichier, la base de données Kerberos peut être altérée. N'est pas un problème

d'endommagement des données sur un serveur esclave KDC, étant donné que la propagation automatique suivante de la base de données permet d'installer une nouvelle copie. Toutefois, si l'altération se produit sur le KDC maître, la base de données altérée est propagée à l'ensemble des KDC esclaves pendant la propagation suivante. La sauvegarde altérée remplace également le fichier de sauvegarde non altéré précédent sur le KDC maître.

Configurez un travail `cron` pour copier à intervalles réguliers le fichier de vidage `slave_datatrans` sur un autre emplacement ou pour créer une autre copie de sauvegarde séparée en utilisant la commande `dump` de `kdb5_util`afin de vous protéger contre ce scénario. Puis, si votre base de données est endommagée, vous pouvez restaurer la sauvegarde la plus récente sur le KDC maître en utilisant la commande `load` de `kdb5_util`.

Autre remarque importante : puisque le fichier dump de la base de données contient les clés de principal, vous devez protéger le fichier de tout accès par des utilisateurs non autorisés. Par défaut, la base de données du fichier de vidage dispose d'autorisations de lecture et d'écriture uniquement en tant que `root`. Afin de les protéger contre tout accès non autorisé, propagez le fichier de vidage de la base de données avec la commande`kprop`, étant donné que cette commande chiffre les données en cours de transfert. En outre, `kprop` propage les données uniquement aux KDC esclaves, ce qui réduit les risques d'envoi par inadvertance du fichier dump de la base de données à des hôtes non autorisés.

EXEMPLE 4-13 Sauvegarde manuelle de la base de données Kerberos

Vous utilisez la commande `dump` de la commande `kdb5_util` pour sauvegarder la base de données. Exécutez cette commande dans un répertoire qui appartient à l'utilisateur `root`.

```
# /usr/sbin/kdb5_util dump
```

Dans l'exemple suivant, la base de données Kerberos est sauvegardée dans un fichier appelé `dumpfile`. Dans la mesure où l'option `-verbose` est spécifiée, chaque principal est imprimé lorsqu'il est sauvegardé. Aucune identité utilisateur sont indiquées, car l'intégralité de la base de données est sauvegardé.

```
# kdb5_util dump -verbose /var/user/kadmin/dumpfile
kadmin/kdc1.corp.example.com@CORP.EXAMPLE.COM
krbtgt/CORP.EXAMPLE.COM@CORP.EXAMPLE.COM
kadmin/history@CORP.EXAMPLE.COM
pak/admin@CORP.EXAMPLE.COM
pak@CORP.EXAMPLE.COM
changepw/kdc1.corp.example.com@CORP.EXAMPLE.COM
```

Dans l'exemple suivant, le vidage contient uniquement les principaux `pak` et `pak/admin`.

```
# kdb5_util dump -verbose pakfile pak/admin@CORP.EXAMPLE.COM pak@CORP.EXAMPLE.COM
pak/admin@CORP.EXAMPLE.COM
pak@CORP.EXAMPLE.COM
```

Pour plus d'informations, reportez-vous à la page de manuel `kdb5_util`(1M).

Restauration d'une sauvegarde de la base de données Kerberos

Avant de commencer Sur le maître KDC, vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Sur le serveur maître, arrêtez les démons KDC.**

```
kdc1 # svcadm disable network/security/krb5kdc
kdc1 # svcadm disable network/security/kadmin
```

2. **Restaurez la base de données Kerberos à l'aide de la commande `load` de la commande `kdb_util`.**

   Par exemple, chargez la sauvegarde `dumpfile` à partir de Exemple 4-13, "Sauvegarde manuelle de la base de données Kerberos".

```
# /usr/sbin/kdb5_util load /var/user/kadmin/dumpfile
```

3. **Démarrez les démons KDC.**

```
kdc1 # svcadm enable -r network/security/krb5kdc
kdc1 # svcadm enable -r network/security/kadmin
```

Exemple 4-14 Restauration de la base de données Kerberos

Dans l'exemple suivant, la base de données appelée `database1` est restaurée dans le répertoire courant à partir du fichier `dumpfile`. Comme l'option `-update` n'est pas spécifiée, une nouvelle base de données est créée.

```
# kdb5_util load -d database1 dumpfile
```

Conversion d'une base de données Kerberos après une mise à niveau du serveur

Si votre base de données KDC a été créée sur un serveur exécutant une ancienne version, la conversion de la base de données vous permet de tirer parti du format de base de données améliorée.

Avant de commencer Cette procédure s'utilise en cas de base de données utilise un format ancien.

Sur le maître KDC, vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Sur le serveur maître, arrêtez les démons KDC.**

```
kdc1 # svcadm disable network/security/krb5kdc
kdc1 # svcadm disable network/security/kadmin
```

2. **Créez un répertoire pour stocker une copie temporaire de la base de données.**

```
kdc1 # mkdir /var/krb5/tmp
kdc1 # chmod 700 /var/krb5/tmp
```

3. **Videz la base de données KDC.**

```
kdc1 # kdb5_util dump /var/krb5/tmp/prdb.txt
```

4. **Enregistrez des copies des fichiers de la base de données actuelle.**

```
kdc1 # cd /var/krb5
kdc1 # mv princ* tmp/
```

5. **Chargez la base de données.**

```
kdc1 # kdb5_util load /var/krb5/tmp/prdb.txt
```

6. **Démarrez les démons KDC.**

```
kdc1 # svcadm enable -r network/security/krb5kdc
kdc1 # svcadm enable -r network/security/kadmin
```

▼ Reconfiguration d'un KDC maître pour l'utilisation de la propagation incrémentielle

Les étapes de cette procédure peuvent être utilisées pour reconfigurer un KDC maître pour qu'il utilise la propagation incrémentielle. Cette procédure utilise les paramètres de configuration ci-dessous :

- Nom de domaine = `EXAMPLE.COM`
- Nom de domaine DNS = `example.com`
- KDC maître = `kdc1.example.com`
- KDC esclave = `kdc2.example.com`
- `admin` principal = `kws/admin`

Avant de commencer

Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Ajoutez des entrées à `kdc.conf`.**

Vous devez activer la propagation incrémentielle et sélectionner le nombre de mises à jour que le KDC maître conserve dans le journal. Pour plus d'informations, reportez-vous à la page de manuel kdc.conf(4).

```
kdc1 # pfedit /etc/krb5/kdc.conf
[kdcdefaults]
kdc_ports = 88,750

[realms]
EXAMPLE.COM= {
profile = /etc/krb5/krb5.conf
database_name = /var/krb5/principal
acl_file = /etc/krb5/kadm5.acl
kadmind_port = 749
max_life = 8h 0m 0s
max_renewable_life = 7d 0h 0m 0s
sunw_dbprop_enable = true
sunw_dbprop_master_ulogsize = 1000
}
```

2. **Créez le principal `kiprop`.**

   Le principal `kiprop` est utilisé pour authentifier le serveur KDC maître et autoriser les mises à jour depuis le KDC maître.

```
kdc1 # /usr/sbin/kadmin -p kws/admin
Enter password: xxxxxxxx
kadmin: addprinc -randkey kiprop/kdc1.example.com
Principal "kiprop/kdc1.example.com@EXAMPLE.COM" created.
kadmin: addprinc -randkey kiprop/kdc2.example.com
Principal "kiprop/kdc2.example.com@EXAMPLE.COM" created.
kadmin:
```

3. **Sur le KDC maître, ajoutez une entrée kiprop au fichier `kadm5.acl`.**

   Cette entrée permet au KDC maître de recevoir des demandes de propagation incrémentielle du serveur kdc2.

```
kdc1 # pfedit /etc/krb5/kadm5.acl
*/admin@EXAMPLE.COM *
kiprop/kdc2.example.com@EXAMPLE.COM p
```

4. **Mettez en commentaire la ligne `kprop` dans le fichier crontab `root`.**

   Cette étape permet d'éviter que le KDC maître ne propage sa copie de la base de données KDC.

```
kdc1 # crontab -e
#ident  "@(#)root       1.20    01/11/06 SMI"
#
# The root crontab should be used to perform accounting data collection.
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
```

```
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
#10 3 * * * /usr/lib/krb5/kprop_script kdc2.example.com
```

5. **Redémarrez `kadmind`.**

```
kdc1 # svcadm restart network/security/kadmin
```

6. **Reconfigurez tous les serveurs KDC esclaves qui utilisent la propagation incrémentielle.**

   Pour obtenir des instructions complètes, reportez-vous à la section “Reconfiguration d'un KDC esclave pour l'utilisation de la propagation incrémentielle” à la page 144.

▼ Reconfiguration d'un KDC esclave pour l'utilisation de la propagation incrémentielle

Avant de commencer

Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. **Ajoutez des entrées à `kdc.conf`.**

   La première nouvelle entrée permet active la propagation incrémentielle. La deuxième nouvelle entrée définit la durée d'interrogation sur deux minutes.

```
kdc2 # pfedit /etc/krb5/kdc.conf
[kdcdefaults]
kdc_ports = 88,750

[realms]
EXAMPLE.COM= {
profile = /etc/krb5/krb5.conf
database_name = /var/krb5/principal
acl_file = /etc/krb5/kadm5.acl
kadmind_port = 749
max_life = 8h 0m 0s
max_renewable_life = 7d 0h 0m 0s
sunw_dbprop_enable = true
sunw_dbprop_slave_poll = 2m
}
```

2. **Ajoutez le principal `kiprop` au fichier `krb5.keytab` .**

```
kdc2 # /usr/sbin/kadmin -p kws/admin
Enter password: xxxxxxxx
kadmin: ktadd kiprop/kdc2.example.com
```

```
Entry for principal kiprop/kdc2.example.com with kvno 3,
encryption type AES-256 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal kiprop/kdc2.example.com with kvno 3,
encryption type AES-128 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal kiprop/kdc2.example.com with kvno 3, encryption type Triple DES cbc
mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin: quit
```

3. **Redémarrez `kpropd`.**

```
kdc2 # svcadm restart network/security/krb5_prop
```

▼ Vérification de la synchronisation des serveurs KDC

Si la propagation incrémentielle a été configurée, cette procédure permet de s'assurer que les informations concernant le KDC esclave ont été mises à jour.

Avant de commencer Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Sur le serveur KDC maître, exécutez la commande `kproplog`.**

```
kdc1 # /usr/sbin/kproplog -h
```

2. **Sur un serveur KDC esclave, exécutez la commande `kproplog`.**

```
kdc2 # /usr/sbin/kproplog -h
```

3. **Vérifiez que les valeurs du dernier numéro de série et du dernier horodatage correspondent.**

Exemple 4-15 Vérification de la synchronisation des serveurs KDC

L'exemple suivant est un exemple des résultats de l'exécution de la commande `kproplog` sur le serveur KDC maître.

```
kdc1 # /usr/sbin/kproplog -h

Kerberos update log (/var/krb5/principal.ulog)
Update log dump:
Log version #: 1
Log state: Stable
Entry block size: 2048
Number of entries: 2500
```

```
First serial #: 137966
Last serial #: 140465
First time stamp: Wed Dec 4 00:59:27 2013
Last time stamp: Wed Dec 4 01:06:13 2013
```

L'exemple suivant est un exemple des résultats de l'exécution de la commande `kproplog` sur le serveur KDC esclave.

```
kdc2 # /usr/sbin/kproplog -h

Kerberos update log (/var/krb5/principal.ulog)
Update log dump:
Log version #: 1
Log state: Stable
Entry block size: 2048
Number of entries: 0
First serial #: None
Last serial #: 140465
First time stamp: None
Last time stamp: Wed Dec 4 01:06:13 2013
```

Notez que les valeurs pour le dernier numéro de série et le dernier horodatage sont identiques, ce qui indique que l'esclave est synchronisé avec le serveur KDC maître.

Dans la sortie du serveur KDC esclave, notez qu'aucune entrée de mise à jour n'existe dans le journal de mise à jour du serveur KDC esclave. Il n'y a pas d'entrées dans la mesure où le serveur KDC esclave ne conserve pas de jeu de mises à jour, à l'inverse du serveur KDC maître. En outre, le serveur KDC esclave n'inclut pas d'informations concernant le premier numéro de série ou le premier horodatage car il ne s'agit pas d'informations pertinentes.

Propagation manuelle de la base de données Kerberos aux KDC esclaves

En règle générale, un travail `cron` propage la base de données Kerberos dans les KDC esclaves. Si vous devez synchroniser un KDC esclave avec le KDC maître en dehors du travail périodique `cron`, deux possibilités s'offrent à vous, la commande `/usr/lib/krb5/kprop_script` et la commande `kprop`. Pour plus d'informations, examinez le script et la page de manuel `kprop`(1M).

Attention - Commandes suivants ne doivent pas être utilisés si la propagation incrémentielle KDC est activée sur le serveur esclave.

▼ Propagation manuelle de la base de données Kerberos à un KDC esclave

1. **Vérifiez que la propagation incrémentielle n'est pas activée sur le KDC esclave.**

```
slave# grep sunw_dbprop_enable /etc/krb5/kdc.conf
sunw_dbprop_enable = true
```

2. **si la valeur est `true`, désactivez la propagation incrémentielle et redémarrez le service `krb5_prop`.**

```
slave# cp /etc/krb5/kdc.conf /etc/krb5/kdc.conf.sav
slave# pfedit /etc/krb5/kdc.conf
...
sunw_dbprop_enable = false
...

slave# svcadm restart krb5_prop
```

3. **Sur le maître KDC, utilisez l'une des commandes suivantes pour propager la base de données de maître KDC vers l'esclave KDC.**

   - **La commande `kprop_script` permet de sauvegarder la base de données avant de synchroniser le KDC esclave.**

     ```
     master# /usr/lib/krb5/kprop_script slave-KDC
     ```

   - **La commande `kprop` propage la sauvegarde actuelle de la base de données sans nouvelle sauvegarde préalable de la base de données Kerberos.**

     ```
     master# /usr/lib/krb5/kprop -f /var/krb5/slave_datatrans slave-KDC
     ```

4. **(Facultatif) Une fois une propagation manuelle terminée, restaurez le fichier d'origine `krb5.conf`.**

```
slave# mv /etc/krb5/kdc.conf.sav /etc/krb5/kdc.conf
```

Configuration de la propagation parallèle pour Kerberos

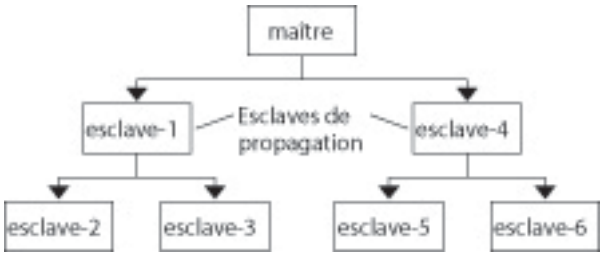
Dans la plupart des cas, le KDC maître est utilisé exclusivement pour propager sa base de données Kerberos aux KDC esclaves. Cependant, si votre site comporte beaucoup de KDC esclaves, vous pouvez envisager le partage de la charge du processus de propagation, aussi appelé *propagation parallèle*.

Attention - Ne configurez pas la propagation parallèle si vous utilisez la propagation incrémentielle.

La propagation parallèle permet aux KDC esclaves spécifiques de partager les fonctions de propagation avec le KDC maître. Ce partage permet à la propagation de s'effectuer plus rapidement et d'alléger la charge du KDC maître.

Par exemple, supposons que votre site dispose d'un KDC maître et de six KDC esclaves (illustrés dans Figure 4-2, “Exemple de configuration de propagation parallèle dans Kerberos”), où `slave-1` jusqu'à `slave-3` correspondent à un regroupement logique et `slave-4` jusqu'à `slave-6` correspondent à un autre regroupement logique. Pour configurer une propagation parallèle, vous pouvez faire en sorte que le KDC maître propage la base de données à `slave-1` et `slave-4`. A leur tour, ces KDC esclaves pourraient propager la base de données aux KDC esclaves dans leur groupe.

FIGURE 4-2 Exemple de configuration de propagation parallèle dans Kerberos



Etapes de configuration d'une propagation parallèle

Les étapes de configuration de haut niveau permettant d'activer la propagation parallèle sont les suivantes :

1. Su le KDC maître, modifiez l'entrée `kprop_script` dans ce travail `cron` pour inclure des arguments uniquement pour les KDC esclaves qui effectueront la propagation suivante (les *esclaves de propagation*).
2. Sur chaque esclave de propagation, l'ajoutez une entrée `kprop_script` dans sa tâche `cron`, qui doit inclure les arguments pour que les esclaves réalisent la propagation. Pour que la propagation parallèle s'effectue, la tâche `cron` doit être configurée de sorte qu'elle s'exécute après que la nouvelle base de données Kerberos soit propagée à l'esclave de propagation.

Remarque - Le temps que prend un esclave de propagation à se propager dépend de facteurs tels que la bande passante du réseau et la taille de la base de données Kerberos.

---

3. Sur chaque esclave KDC, configurez les autorisations appropriées à propager en ajoutant le nom du principal d'hôte de son KDC de propagation à son fichier `kpropd.acl`.

EXEMPLE 4-16 Configuration de la propagation parallèle dans Kerberos

A l'aide de l'exemple dans Figure 4-2, “Exemple de configuration de propagation parallèle dans Kerberos”, l'entrée `kprop_script` du KDC maître doit ressembler à ce qui suit :

```
0 3 * * * /usr/lib/krb5/kprop_script slave-1.example.com slave-4.example.com
```

L'entrée `kprop_script` du `slave-1` doit ressembler à ce qui suit :

```
0 4 * * * /usr/lib/krb5/kprop_script slave-2.example.com slave-3.example.com
```

Notez que la propagation sur l'esclave démarre une heure après sa propagation par le maître.

Le fichier `kpropd.acl` sur les esclaves de propagation doit contenir l'entrée suivante :

```
host/master.example.com@EXAMPLE.COM
```

Le fichier `kpropd.acl` sur les KDC esclaves propagés par `slave-1` contient l'entrée suivante :

```
host/slave-1.example.com@EXAMPLE.COM
```

Administration du fichier stash pour la base de données Kerberos

Le *fichier stash* contient la clé principale de la base de données Kerberos, qui est créée automatiquement lorsque vous créez une base de données Kerberos. Si le fichier stash est endommagé, vous pouvez utiliser la commande `stash` de l'utilitaire `kdb5_util` pour remplacer le fichier endommagé. Le seul moment où vous devez supprimer un fichier stash est après la suppression de la base de données Kerberos avec la commande `destroy` de `kdb5_util`. Dans la mesure où le fichier stash n'est pas automatiquement supprimé avec la base de données, vous devez d'abord supprimer le fichier stash manuellement.

Pour supprimer le fichier stash, utilisez la commande `rm` :

```
# rm stash-file
```

Où *stash-file* est le chemin d'accès du fichier stash. Par défaut, le fichier stash est situé dans `/var/krb5/.k5.` *realm*.

Remarque - Si vous devez recréer le fichier stash, vous pouvez utiliser l'option `-f` de la commande `kdb5_util`.

▼ Création, utilisation et stockage d'une nouvelle clé maître pour la base de données Kerberos

Avant de commencer Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Créez une nouvelle clé principale.**

   Cette commande ajoute une nouvelle clé principale, générée de façon aléatoire. L'option `-s` requiert que la nouvelle clé principale soit stockée dans le fichier keytab par défaut.

   ```
   # kdb5_util add_mkey -s

   Creating new master key for master key principal 'K/M@EXAMPLE.COM'
   You will be prompted for a new database Master Password.
   It is important that you NOT FORGET this password.
   Enter KDC database master key:     /** Type strong password **/
   Re-enter KDC database master key to verify: xxxxxxxx
   ```

2. **Vérifiez que la nouvelle clé principale existe.**

   ```
   # kdb5_util list_mkeys
   Master keys for Principal: K/M@EXAMPLE.COM
   KNVO: 2, Enctype: AES-256 CTS mode with 96-bit SHA-1 HMAC, No activate time set
   KNVO: 1, Enctype: AES-128 CTS mode with 96-bit SHA-1 HMAC, Active on: Fri Dec 31 18:00:00 CST
    2011 *
   ```

   L'astérisque dans cette sortie identifie la clé principale actuellement active.

3. **Définissez le moment auquel la nouvelle clé principale deviendra active.**

   ```
   # date
   Fri Jul 11 17:57:00 CDT 2014
   # kdb5_util use_mkey 2 'now+2days'
   # kdb5_util list_mkeys
   Master keys for Principal: K/M@EXAMPLE.COM
   KNVO: 2, Enctype: AES-256 CTS mode with 96-bit SHA-1 HMAC,
   Active on: Sun Jul 13 17:57:15 CDT 2014
   KNVO: 1, Enctype: AES-128 CTS mode with 96-bit SHA-1 HMAC,
   Active on: Fri Dec 31 18:00:00 CST 2011 *
   ```

   Dans cet exemple, la date est définie deux jours plus tard pour laisser le temps à la nouvelle clé principale de se propager à tous les KDC. Ajustez la date selon les besoins de votre environnement.

4. **(Facultatif) Après la création d'un nouveau principal, vérifiez que la nouvelle clé principale est en cours d'utilisation.**

   ```
   # kadmin.local -q 'getprinc tamiko' |egrep 'Principal|MKey'
   Authenticating as principal root/admin@EXAMPLE.COM with password.
   ```

```
Principal: tamiko@EXAMPLE.COM
MKey: vno 2
```

Dans cet exemple, `MKey: vno 2` indique que la clé secrète du principal est protégée par la nouvelle clé principale 2.

5. **Re-chiffrez les clés secrètes de l'utilisateur principal avec la nouvelle clé principale.**

   Si vous ajoutez un argument de modèle à la fin de la commande, les principaux qui correspondent au modèle seront mis à jour. Ajoutez l'option -n à la syntaxe de cette commande afin d'identifier les principaux qui seront mis à jour.

   ```
   # kdb5_util update_princ_encryption -f -v
   Principals whose keys WOULD BE re-encrypted to master key vno 2:
   updating: host/kdc1.example.com@EXAMPLE.COM
   skipping:   tamiko@EXAMPLE.COM
   updating: kadmin/changepw@EXAMPLE.COM
   updating: kadmin/history@EXAMPLE.COM
   updating: kdc/admin@EXAMPLE.COM
   updating: host/kdc2.example.com@EXAMPLE.COM
   6 principals processed: 5 updated, 1 already current
   ```

6. **Purgez l'ancienne clé principale.**

   Lorsqu'une clé principale n'est plus utilisée pour protéger les clés secrètes des principaux, elle peut être purgée du principal de la clé principale. Cette commande ne purge pas la clé si celle-ci est encore utilisée par des principaux. Ajoutez l'option -n à cette commande pour vérifier que la clé principale appropriée sera purgée.

   ```
   # kdb5_util purge_mkeys -f -v
   Purging the follwing master key(s) from K/M@EXAMPLE.COM:
   KNVO: 1
   1 key(s) purged.
   ```

7. **Vérifiez que l'ancienne clé principale a été purgée.**

   ```
   # kdb5_util list_mkeys
   Master keys for Principal: K/M@EXAMPLE.COM
   KNVO: 2, Enctype: AES-256 CTS mode with 96-bit SHA-1 HMAC,
   Active on: Sun Jul 13 17:57:15 CDT 2014 *
   ```

8. **Mettez à jour le fichier stash.**

   ```
   # kdb5_util stash
   Using existing stashed keys to update stash file.
   ```

9. **Vérifiez que le fichier stash a été mis à jour.**

   ```
   # klist -kt /var/krb5/.k5.EXAMPLE.COM
   Keytab name: FILE:.k5.EXAMPLE.COM
   KVNO Timestamp              Principal
   ---- --------------- ------------------------------------------------------------
   ```

```
2 05/11/2014 18:03 K/M@EXAMPLE.COM
```

Renforcement de la sécurité des serveurs Kerberos

Cette section fournit des conseils sur l'augmentation de la sécurité sur les serveurs d'applications Kerberos et sur les serveurs KDC.

Restriction de l'accès aux serveurs KDC

Les serveurs KDC maîtres et esclaves disposent de copies de la base de données KDC stockées localement. La restriction de l'accès à ces serveurs pour sécuriser les bases de données est importante pour la sécurité globale de l'installation Kerberos.

- Limitez l'accès physique au matériel prenant en charge le KDC.

  Assurez-vous que le serveur KDC et son moniteur se trouvent dans un site sécurisé. Les utilisateurs standard ne doit pas être en mesure d'accéder à ce serveur de quelque façon que ce soit.
- Stockez les sauvegardes de la base de données KDC sur des disques locaux ou les KDC esclaves.

  Réalisez des sauvegardes sur bande de votre KDC uniquement si les bandes sont stockées en toute sécurité. Suivez la même pratique pour les copies de fichiers keytab.

  Il est recommandé de stocker ces fichiers sur un système de fichiers local non partagé avec d'autres systèmes. Le système de stockage de fichiers peut être le serveur KDC maître ou n'importe lequel des KDC esclaves.

Utilisation d'un fichier dictionnaire pour augmenter la sécurité de mot de passe

Un fichier dictionnaire peut être utilisé par le service Kerberos pour éviter qu'un mot dans le dictionnaire soit utilisé en tant que mot de passe pour de nouvelles informations d'identification. Pour rendre plus difficile le fait de deviner un mot de passe, il est judicieux d'éviter l'utilisation de mots du dictionnaire en tant mots de passe. Par défaut, le fichier `/var/krb5/kadm5.dict` est utilisé, mais il est vide.

Vous devez ajouter une ligne au fichier de configuration KDC, `kdc.conf` pour informer le service d'utiliser un fichier de dictionnaire. Dans cet exemple, l'administrateur utilise le dictionnaire qui est inclus dans l'utilitaire `spell`, puis le redémarre avec les services Kerberos.

Pour obtenir une description complète du fichier de configuration, reportez-vous à la page de manuel kdc.conf(4).

```
kdc1 # pfedit /etc/krb5/kdc.conf
[kdcdefaults]
kdc_ports = 88,750

[realms]
EXAMPLE.COM = {
profile = /etc/krb5/krb5.conf
database_name = /var/krb5/principal
acl_file = /etc/krb5/kadm5.acl
kadmind_port = 749
max_life = 8h 0m 0s
max_renewable_life = 7d 0h 0m 0s
sunw_dbprop_enable = true
sunw_dbprop_master_ulogsize = 1000
dict_file = /usr/share/lib/dict/words
}
kdc1 #
kdc1 # svcadm restart -r network/security/krb5kdc
kdc1 # svcadm restart -r network/security/kadmin
```

♦ ♦ ♦ CHAPITRE 5

Administration des principaux et des stratégies Kerberos

Ce chapitre décrit les procédures d'administration des principaux et des stratégies qui leur sont associées. Ce chapitre indique également comment administrer un fichier keytab d'hôte.

Ce chapitre comprend les sections suivantes :

Pour plus d'informations d'ordre général concernant les principaux et les stratégies, reportez-vous au Chapitre 2, A propos du service Kerberos et auChapitre 3, Planification du service Kerberos.

Méthodes d'administration des principaux et des stratégies Kerberos

La base de données Kerberos sur le KDC maître contient tous les principaux Kerberos du domaine, leurs mots de passe, les stratégies et d'autres informations administratives. Pour créer et supprimer des principaux et pour modifier leurs attributs, vous pouvez utiliser les commandes `kadmin` ou `gkadmin`.

La commande `kadmin` fournit une interface de ligne de commande interactive qui permet de mettre à jour les principaux, les stratégies et les fichiers keytab de Kerberos. Vous pouvez exécuter des scripts qui permettent d'automatiser la création du principal. La commande `kadmin` comporte une version locale et une version distante :

- `kadmin` : utilise l'authentification Kerberos pour fonctionner en toute sécurité de n'importe où sur le réseau
- `kadmin.local` : doit être directement exécuté sur le KDC maître

Les capacités de ces deux versions sont identiques. Vous devez configurer la base de données assez de la version locale avant de pouvoir utiliser la version distante.

La version Oracle Solaris fournit également une interface d'utilisateur graphique interactive (GUI), `gkadmin`.

Cette section décrit les des fonctionnalités de script de la commande `kadmin.local`, puis compare la ligne de commande et les interfaces d'utilisateur graphique.

Automatisation de la création de principaux Kerberos

Vous pouvez utiliser la commande `kadmin.local` dans un script pour automatiser la création de nouveaux principaux Kerberos. L'automatisation est utile lorsque vous voulez ajouter un grand nombre de nouvelles des principaux sont ajoutés à la base de données.

La ligne de script shell suivante indique comment automatiser la création de nouveaux principaux :

```
awk '{ print "ank +needchange -pw", $2, $1 }' < /tmp/princnames |
time /usr/sbin/kadmin.local> /dev/null
```

L'exemple précédent est réparti sur deux lignes pour une meilleure lisibilité.

- Le script lit un fichier appelé `princnames`. Ce fichier contient les noms de principaux et leurs mots de passe, et les ajoute à la base de données Kerberos.

  Vous devez créer le fichier `princnames` contenant un nom de principal et son mot de passe sur chaque ligne, séparés par un ou plusieurs espaces.
- La commande`ank` ajoute une nouvelle clé. `ank` est un alias pour la commande `add_principal`.
- L'option `+needchange` configure le principal de manière à ce que l'utilisateur qui est le principal soit invité à saisir un nouveau mot de passe lors de sa première connexion.

  La nécessité de modifier un mot de passe permet d'assurer que les mots de passe dans le fichier `princnames` ne représentent pas un risque pour la sécurité.

Vous pouvez construire des scripts plus élaborés. Par exemple, le script peut utiliser les informations contenues dans le service de noms pour obtenir la liste des noms d'utilisateur pour les noms de principaux. Ce que vous faite et la manière dont vous le faites est déterminé par les besoins de votre site et votre expérience en script.

Interface graphique `gkadmin`

L'interface graphique Kerberos `gkadmin`fournit la plupart des fonctionnalités de la CLI et comprend une aide en ligne. Le tableau suivant décrit les différences entre les GUI et la CLI.

TABLEAU 5-1 Equivalents de ligne de commande de l'interface graphique `gkadmin`

Procédure Gkadmin, GUI	**Equivalent de la commande `kadmin`**
Affichage de la liste de principaux.	`list_principals` ou `get_principals`
Affichage des attributs d'un principal.	`get_principal`
Création d'un principal.	`add_principal`
Duplication d'un principal.	Pas d'équivalent de ligne de commande
Modification d'un principal.	`modify_principal` ou `change_password`
Suppression d'un principal.	`delete_principal`
Définition des valeurs par défaut pour la création de principaux.	Pas d'équivalent de ligne de commande
Affichage de la liste des stratégies.	`list_policies` ou `get_policies`
Affichage des attributs d'une stratégie.	`get_policy`
Création d'une stratégie.	`add_policy`
Duplication d'une stratégie.	Pas d'équivalent de ligne de commande
Modification d'une stratégie.	`modify_policy`
Supprimer une stratégie.	`delete_policy`

L'interface graphique Kerberos `gkadmin` fournit une aide contextuelle. Pour accéder au navigateur, utilisez l'Oracle URL (http://docs.oracle.com/cd/E23824_01/html/821-1456/aadmin-23.html). Vous pouvez copier URL ce fichier vers un site. Spécifiez l'URL de l'aide en ligne dans le fichier `krb5.conf` lors de la configuration d'un hôte pour utiliser l'interface graphique `gkadmin`.

Administration des principaux Kerberos

Cette section fournit des exemples d'utilisation de la commande `kadmin` et de l'interface graphique `gkadmin` pour gérer les principaux. Pour plus d'informations, reportez-vous aux pages de manuel `kadmin`(1M) et `gkadmin`(1M).

Visualisation des principaux de Kerberos et leurs attributs

Les exemples suivants montrent comment obtenir la liste des principaux et leurs attributs. Vous pouvez utiliser des caractères génériques pour construire les listes prévues à cet effet. Pour plus d'informations sur les éventuelles caractères génériques, examinez la définition de `expression` dans la page de manuel `kadmin`(1M).

EXEMPLE 5-1 Visualisation des principaux de Kerberos

Dans cet exemple, la sous-commande `list_principals` est utilisée pour répertorier tous les principaux correspondant à `kadmin*`. Sans un argument, `list_principals` répertorie tous les principaux définis dans la base de données Kerberos.

```
# /usr/sbin/kadmin
kadmin: list_principals kadmin*
kadmin/changepw@EXAMPLE.COM
kadmin/kdc1.example.com@EXAMPLE.COM
kadmin/history@EXAMPLE.COM
```

EXEMPLE 5-2 Visualisation des attributs de principaux Kerberos

L'exemple suivant affiche les attributs du principal `jdb/admin`.

```
kadmin: get_principal jdb/admin
Principal: jdb/admin@EXAMPLE.COM

Expiration date: [never]
Last password change: [never]

Password expiration date: Fri Sep 13 11:50:10 PDT 2013
Maximum ticket life: 1 day 16:00:00
Maximum renewable life: 1 day 16:00:00
Last modified: Thu Aug 15 13:30:30 PST 2013 (host/admin@EXAMPLE.COM)
Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 1
Key: vno 1, AES-256 CTS mode with 96-bit SHA-1 HMAC, no salt
Key: vno 1, AES-128 CTS mode with 96-bit SHA-1 HMAC, no salt
Key: vno 1, Triple DES with HMAC/sha1, no salt
Key: vno 1, ArcFour with HMAC/md5, no salt
Attributes: REQUIRES_HW_AUTH
Policy: [none]
kadmin: quit
```

EXEMPLE 5-3 Utilisation de l'interface graphique `gkadmin` pour répertorier et définir par défaut les principaux Kerberos

Dans cet exemple, l'administrateur souhaite montrer au nouvel administrateur, la liste des principaux et leurs attributs, utilise donc l'interface graphique `gkadmin`. L'administrateur définit également de nouvelles valeurs par défaut pour les futures principaux.

```
# /usr/sbin/gkadmin
```

La fenêtre affiche les le nom du principal, et le mot de passe KDC maître, un domaine ainsi que les champs prévus à cet effet.

Il accède à la liste de tous les noms de principaux administrateur, puis présente les d'informations sur l'utilisation de la nouvelle entre les majuscules et les minuscules filtre.

Ensuite, l'administrateur clique sur le menu Edition et choisit Propriétés. Modification du mot de passe une fois que vous avez cliqué sur, l'administrateur Exiger appliquera la modification.

Pour voir les attributs d'une identité utilisateur en cours, l'administrateur permet d'accéder à la liste des principaux et choisit un principal dans la liste. La première boîte de dialogue affiche les attributs de base. L'administrateur clique sur le bouton Suivant pour afficher tous les attributs.

Création d'un principal Kerberos

Si un nouveau principal exige une nouvelle stratégie, vous devez d'abord créer cette stratégie l'identité utilisateur. Pour la création d'une stratégie, reportez-vous à Exemple 5-10, “Création d'une nouvelle stratégie de mot de passe Kerberos.”. Indiquez la plupart des informations sur les mots de passe. les stratégies Kerberos

EXEMPLE 5-4 Création d'un principal Kerberos

L'exemple suivant crée un nouveau principal appelé `pak` et définit la stratégie du principal sur `testuser`. Les autres valeurs requises, comme type de chiffrement, utiliser des valeurs par défaut.

```
# /usr/sbin/kadmin
kadmin: add_principal -policy testuser pak
Enter password for principal "pak@EXAMPLE.COM": xxxxxxxx
Re-enter password for principal "pak@EXAMPLE.COM": xxxxxxxx
Principal "pak@EXAMPLE.COM" created.
kadmin: quit
```

En règle générale, peu d'utilisateurs disposent de privilèges pour gérer la base de données Kerberos. Si ce nouveau principal a besoin de privilèges d'administration, poursuivez avec “Modification des privilèges d'administration Kerberos des principaux” à la page 161.

Modification d'un principal Kerberos

Les exemples suivants montrent comment modifier le mot de passe attributs d'un principal Kerberos.

EXEMPLE 5-5 Modification du nombre maximal de nouvelles tentatives de mot de passe pour un principal Kerberos

```
# /usr/sbin/kadmin
kadmin: modify_principal jdb
kadmin: maxtries=5
kadmin: quit
```

EXEMPLE 5-6 Modification du mot de passe d'un principal Kerberos

```
# /usr/sbin/kadmin
kadmin: change_password jdb
Enter password for principal "jdb": xxxxxxxx
Re-enter password for principal "jdb": xxxxxxxx
Password for "jdb@EXAMPLE.COM" changed.
kadmin: quit
```

Suppression d'un principal Kerberos

L'exemple suivant montre comment supprimer un principal. Conforme à la messagerie en ligne avant de continuer.

```
# /usr/sbin/kadmin
kadmin: delete_principal pak
Are you sure you want to delete the principal "pak@EXAMPLE.COM"? (yes/no): yes
Principal "pak@EXAMPLE.COM" deleted.
Make sure that you have removed this principal from all ACLs before reusing.
kadmin: quit
```

Pour supprimer ce principal de toutes les ACL, reportez-vous à la section “Modification des privilèges d'administration Kerberos des principaux” à la page 161.

Duplication d'un principal Kerberos à l'aide de l'interface graphique `gkadmin`

Vous pouvez dupliquer un principal à l'aide de l'interface graphique `gkadmin`. Il n'existe pas d'équivalent de ligne de commande pour cette procédure.

1. Après le lancement de l'interface graphique à l'aide de la commande `/usr/sbin/gkadmin`, cliquez sur l'onglet Principaux et sélectionnez un principal à dupliquer.
2. Cliquez sur le bouton Dupliquer. De cette action la duplication de tous les attributs du principal sélectionné à l'exception pour le nom de principal et un mot de passe.
3. Permet d'indiquer un nouveau nom et votre mot de passe, puis cliquez sur Enregistrer pour créer une copie exacte.
4. Pour modifier ce double, indiquez les valeurs des attributs du principal.

   Trois fenêtres contiennent les informations sur les attributs. Choisissez l'aide contextuelle dans le menu d'aide pour obtenir plus d'informations sur les divers attributs dans chaque fenêtre.

En règle générale, peu d'utilisateurs disposent de privilèges pour gérer la base de données Kerberos. Si ce nouveau principal a besoin de privilèges d'administration, poursuivez avec “Modification des privilèges d'administration Kerberos des principaux” à la page 161.

Modification des privilèges d'administration Kerberos des principaux

Les utilisateurs qui sont privilégiées et informations d'identification et de les quelques d'administrer la base de données Kerberos sont spécifiés dans la liste de contrôle d'accès (ACL Kerberos). Cette liste est tenue à jour en tant qu'entrées dans un fichier, `/etc/krb5/kadm5.acl`. Pour plus d'informations, reportez-vous à la page de manuel `kadm5.acl`(4).

Pour ajouter des entrées au fichier `kadm5.acl`, utilisez la commande `pfedit`.

```
# pfedit /usr/krb5/kadm5.acl
```

Une entrée dans le fichier `kadm5.acl` adopte le format suivant :

principal privileges [*principal-target*]

- *principal* : spécifie le principal auquel les privilèges sont accordés. N'importe quelle partie du nom du principal peut inclure le caractère générique "*", ce qui est utile pour fournir les mêmes privilèges pour un groupe de principaux. Par exemple, si vous souhaitez spécifier tous les principaux avec l'instance `admin`, vous devez utiliser `*/admin@`*realm*.

  Notez qu'une utilisation commune d'une instance `admin` consiste à accorder des privilèges séparés (tels que l'accès à l'administration de la base de données Kerberos) à un principal Kerberos séparé. Par exemple, l'utilisateur `jdb` peut avoir un principal pour l'utilisation administrative, appelé `jdb/admin`. Le fait d'avoir deux principaux, l'utilisateur `jdb` obtient `jdb/admin` uniquement les tickets lorsque vous devez disposer de privilèges d'administration.
- *privileges* : cette option permet de spécifier les opérations pouvant être réalisées par le principal. Ce champ est constitué d'une chaîne d'un ou plusieurs caractères de la liste

suivante. Si le caractère est majuscule ou non spécifié, l'opération n'est pas autorisée. Si le caractère est minuscule, l'opération est autorisée.

- [A]a : autorise ou interdit l'ajout des principaux ou des stratégies.
- [C]c : autorise ou interdit le changement de mots de passe des principaux.
- [D]d : autorise ou interdit la suppression des principaux ou des stratégies.
- [I]i : autorise ou interdit la consultation de la base de données Kerberos.
- [L]l : autorise ou interdit la liste des principaux ou stratégies.
- [M]m : autorise ou interdit la modification de principaux ou stratégies.
- x ou * : autorise tous les privilèges (admcil).

- *principal-target* : lorsqu'un principal est spécifié dans ce champ, les privilèges de principal s'appliquent à ce principal uniquement. Pour affecter des privilèges à un groupe de principaux, utilisez le caractère générique "*" dans *principal-target*.

EXEMPLE 5-7 Modification des privilèges d'un principal Kerberos

L'entrée suivante dans le fichier kadm5.acl accorde à tout principal dans le domaine EXAMPLE.COM avec l'instance admin tous les privilèges de la base de données Kerberos :

```
*/admin@EXAMPLE.COM *
```

L'entrée suivante dans le fichier kadm5.acl donne au principal jdb@EXAMPLE.COM les privilèges pour répertorier et consulter tout principal qui a l'instance root.

```
jdb@EXAMPLE.COM li */root@EXAMPLE.COM
```

Administration des stratégies Kerberos

Cette section fournit des exemples d'utilisation de la commande kadmin et de l'interface graphique gkadmin pour gérer des stratégies Kerberos. Indiquez la plupart des informations sur les mots de passe. les stratégies Kerberos

Sont les étapes permettant d'administrer similaire à celle des stratégies d'administration des principaux. Pour plus d'informations, reportez-vous aux pages de manuel kadmin(1M) et gkadmin(1M).

EXEMPLE 5-8 Affichage de la liste des stratégies Kerberos

Dans cet exemple, la sous-commande list_policies est utilisée pour répertorier toutes les stratégies qui correspondent à *user*. Sans un argument, list_policies répertorie toutes les stratégies définis dans la base de données Kerberos.

```
# kadmin
kadmin: list_policies *user*
testuser
financeuser
kadmin: quit
```

EXEMPLE 5-9 Visualisation des attributs d'une stratégie Kerberos

Dans cet exemple, la sous-commande `get_policy` est utilisée pour afficher les attributs de la stratégie `financeuser`.

```
# /usr/sbin/kadmin.local
kadmin.local: get_policy financeuser
Policy: financeuser
Maximum password life: 13050000
Minimum password life: 10886400
Minimum password length: 8
Minimum number of password character classes: 2
Number of old keys kept: 3
Reference count: 8
Maximum password failures before lockout: 5
Password failure count reset interval: 200
Password lockout duration: 300
kadmin: quit
```

Le nombre de références est le nombre de principaux qui est affecté à cette stratégie.

EXEMPLE 5-10 Création d'une nouvelle stratégie de mot de passe Kerberos.

Dans cet exemple, la sous-commande `add_policy` est utilisée pour créer la stratégie `build11`. Cette stratégie requiert au moins trois classes de caractères dans un mot de passe.

```
# kadmin
kadmin: add_policy -minclasses 3 build11
kadmin: quit
```

EXEMPLE 5-11 Le verrouillage du compte du traitement d'une stratégie Kerberos

Dans cet exemple, trois échecs d'authentification en l'espace de `300` secondes déclenche un verrouillage de compte de `900` secondes.

```
kadmin: add_policy -maxfailure 3 -failurecountinterval "300 seconds"\
-lockoutduration "900 seconds" default
```

Pour libérer le verrou externe requiert le segment de 15 minutes au sein d'une action d'administration.

```
# /usr/sbin/kadmin -p kws/admin
```

```
Enter password: xxxxxxxx
kadmin: modify_principal -unlock principal
```

EXEMPLE 5-12 Modification d'une stratégie Kerberos

Dans cet exemple, la sous-commande `modify_policy` est utilisée pour modifier la longueur minimum d'un mot de passe à huit caractères pour la stratégie `build11`.

```
# kadmin
kadmin: modify_policy -minlength 8 build11
kadmin: quit
```

EXEMPLE 5-13 Suppression d'une stratégie Kerberos

Dans cet exemple, la sous-commande `delete_policy` est utilisée pour supprimer la stratégie `build11`.

1. L'administrateur supprime la stratégie à partir de tous les principaux qui l'utilisent.

   ```
   # kadmin
   kadmin: modify_principal -policy build11 *admin*
   ```

2. Ensuite, il supprime la stratégie.

   ```
   kadmin: delete_policy build11
   Are you sure you want to delete the policy "build11"? (yes/no): yes
   kadmin: quit
   ```

La commande `delete_policy` échoue si la stratégie est affectée à un principal.

EXEMPLE 5-14 Duplication d'une stratégie Kerberos à l'aide de l'interface graphique `gkadmin`

Dans l'interface graphique `gkadmin`, vous pouvez dupliquer une stratégie sélectionnée en cliquant sur le bouton Dupliquer. Champ Nom de la police. entrez le nom de la nouvelle stratégie Vous pouvez également modifier les attributs de la stratégie que vous avez dupliqué. La procédure est similaire à la procédure dans la section “Duplication d'un principal Kerberos à l'aide de l'interface graphique `gkadmin`” à la page 160.

Administration des fichiers keytab

Tous les hôtes qui fournissent un service doivent disposer d'un fichier local, appelé un fichier keytab, qui est l'abréviation de "key table" (table des clés). Le fichier keytab contient le principal pour le service approprié, appelé clé de service. Une clé de service est utilisée par un

service pour s'authentifier auprès du KDC et est uniquement connue de Kerberos et du service lui-même. Par exemple, si vous avez un serveur NFS utilisant Kerberos, le serveur doit avoir un fichier keytab qui contient le principal de service pour le principal de service `nfs`.

Pour ajouter une clé de service à un fichier keytab, vous ajoutez le principal de service approprié à un fichier keytab de l'hôte à l'aide de la commande `ktadd` décrite dans un processus `kadmin`. Comme vous êtes en train d'ajouter un principal de service à un fichier keytab, le principal doit déjà exister dans la base de données Kerberos. Sur les serveurs d'applications qui fournissent des services utilisant Kerberos, le fichier keytab se trouve dans `/etc/krb5/krb5.keytab`, par défaut.

Un fichier keytab est comparable à un mot de passe d'utilisateur. La même façon que les utilisateurs doivent protéger leurs mots de passe, serveurs d'application doit protéger leurs fichiers keytab. Vous devez toujours stocker les fichiers keytab sur un disque local et les rendre lisibles uniquement par l'utilisateur `root`. En outre, vous ne devez jamais envoyer un fichier keytab par le biais d'un réseau non sécurisé.

Des circonstances particulières peuvent exiger que vous ajoutiez un principal `root` à un fichier keytab d'hôte. Si vous souhaitez qu'un utilisateur sur un client Kerberos monte des systèmes de fichiers NFS utilisant Kerberos qui nécessitent un accès équivalent à `root`, vous devez ajouter le principal `root` du client au fichier keytab du client. Dans le cas contraire, les utilisateurs doivent utiliser la commande `kinit` en tant que `root` pour obtenir des informations d'identification pour le principal `root` du client lorsqu'ils souhaitent monter un système de fichiers NFS utilisant Kerberos avec accès `root`, même lorsqu'ils utilisent l'agent de montage automatique.

Attention - Le montage des serveurs NFS en tant qu'utilisateur `root` peut poser des problèmes de sécurité.

Vous pouvez également utiliser la commande `ktutil` pour administrer les fichiers keytab. Cette commande interactive vous permet de gérer un fichier keytab d'un hôte local sans disposer de privilèges d'administration Kerberos, car cette commande n'interagit pas avec la base de données Kerberos comme le fait `kadmin`. Après l'ajout d'un principal à un fichier keytab, vous pouvez utiliser `ktutil` pour visualiser la liste de clés dans le fichier keytab ou pour désactiver temporairement l'authentification d'un service.

Remarque - Lorsque vous modifiez un principal dans un fichier keytab à l'aide de la commande `ktadd` dans `kadmin`, une nouvelle clé est générée et ajoutée au fichier keytab.

Ajout d'un principal de service Kerberos dans un fichier keytab

Vous pouvez ajouter un principal à un fichier keytab après avoir vérifié que le principal existe dans la base de données Kerberos. Pour plus d'informations, reportez-vous à la section “Visualisation des principaux de Kerberos et leurs attributs” à la page 158.

Sur l'hôte qui a besoin qu'un principal soit ajouté à son fichier keytab, vous exécutez la commande `ktadd` dans un processus `kadmin`. Pour plus d'informations, reportez-vous à la page de manuel `kadmin`(1M).

```
# /usr/sbin/kadmin
kadmin: ktadd [-e enctype] [-k keytab] [-q] [principal | -glob principal-exp]
```

-e *enctype*	Remplace la liste des types de chiffrement définie dans le fichier `krb5.conf`.
-k *keytab*	Spécifie le fichier keytab. Par défaut, `/etc/krb5/krb5.keytab` est utilisé.
-q	Affiche des informations moins détaillées.
principal	Spécifie le principal à ajouter au fichier keytab. Vous pouvez ajouter les principaux de service suivants : `host`, `root`, `nfs` et `ftp`.
-glob *principal-exp*	Spécifie les expressions de principal. Tous les principaux qui correspondent à *principal-exp* sont ajoutés au fichier keytab. Les règles qui régissent l'expression de principal sont les mêmes que pour la commande `list_principals` de `kadmin`. Pour obtenir les expressions possibles, examinez la définition de `expression` dans la page de manuel`kadmin`(1M)

EXEMPLE 5-15 Ajout d'un principal de service dans un fichier keytab

Dans cet exemple, le principal `host` de `denver` est ajouté au fichier keytab de `denver` de sorte que le KDC puisse authentifier les services réseau de `denver`.

```
denver # /usr/sbin/kadmin
kadmin: ktadd host/denver.example.com
Entry for principal host/denver.example.com with kvno 3,
encryption type AES-256 CTS
mode with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal host/denver.example.com with kvno 3,
encryption type AES-128 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal host/denver.example.com with kvno 3,
encryption type Triple DES cbc mode
```

```
with HMAC/sha1 added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin: quit
```

Suppression d'un principal de service d'un fichier keytab.

Vous pouvez supprimer un principal dans un fichier keytab. Sur l'hôte qui a besoin qu'un principal soit supprimé de son fichier keytab, vous affichez d'abord la liste de principaux. Reportez-vous à la section “Affichage des principaux dans un fichier Keytab” à la page 167.

Puis, vous exécutez la commande `ktadd` dans un processus `kadmin`. Pour plus d'informations, reportez-vous à la page de manuel `kadmin`(1M).

```
# /usr/sbin/kadmin
kadmin: ktremove [-k keytab] [-q] principal [kvno | all | old ]
```

`-k` *keytab*	Spécifie le fichier keytab. Par défaut, `/etc/krb5/krb5.keytab` est utilisé.
`-q`	Affiche des informations moins détaillées.
principal	Spécifie le principal à supprimer du fichier keytab.
kvno	Supprime toutes les entrées pour le principal spécifié dont le numéro de version de clé correspond à *kvno*.
`all`	Supprime toutes les entrées pour le principal spécifié.
`old`	Supprime toutes les entrées pour le principal spécifié, à l'exception des principaux avec les numéros de version de clé les plus élevés.

EXEMPLE 5-16 Suppression d'un principal de service d'un fichier keytab.

Dans cet exemple, le principal `host` de `denver` est supprimé du fichier keytab de `denver`.

```
denver # /usr/sbin/kadmin
kadmin: ktremove host/denver.example.com@EXAMPLE.COM
kadmin: Entry for principal host/denver.example.com@EXAMPLE.COM with kvno 3
removed from keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin: quit
```

Affichage des principaux dans un fichier Keytab

Sur l'hôte avec le fichier keytab, exécutez la commande `ktutil`, lisez le keytab, puis répertoriez les principaux. Les principaux sont également appelés *keylist*.

Remarque - Bien qu'il soit possible de créer des fichiers keytab détenus par d'autres utilisateurs, l'utilisation de l'emplacement par défaut pour le fichier keytab requiert la propriété `root`.

EXEMPLE 5-17 Affichage de la liste de clés (principaux) dans un fichier keytab

L'exemple suivant affiche la liste de clés dans le fichier `/etc/krb5/krb5.keytab` sur l'hôte `denver`.

```
denver # /usr/bin/ktutil
ktutil: read_kt /etc/krb5/krb5.keytab
ktutil: list
slot KVNO Principal
---- ---- ---------------------------------------
1    5 host/denver@EXAMPLE.COM
ktutil: quit
```

Désactivation temporaire d'un service Kerberos sur un hôte

Parfois, il peut s'avérer nécessaire de désactiver temporairement le mécanisme d'authentification d'un service, tel que `ssh` ou `ftp`, sur un serveur d'application réseau. Par exemple, si vous le souhaitez, vous pouvez empêcher les utilisateurs de se connecter à un système pendant que vous êtes en train d'effectuer des procédures de maintenance.

Remarque - Par défaut, la plupart des services exige l'authentification. Si un service n'a pas d'authentification nécessitent une authentification, la désactivation du n'a aucun effet. Le service est toujours disponible.

▼ Désactivation temporaire de l'authentification d'un service Kerberos sur un hôte

La commande `ktutil` permet à un utilisateur qui ne dispose pas des privilèges `kadmin` de désactiver un service. Cet utilisateur peut également le restaurer. Pour plus d'informations, reportez-vous à la page de manuel `ktutil`(1).

Avant de commencer

Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Enregistrez le fichier keytab actuel dans un fichier temporaire.**

   Vous utilisez ce fichier temporaire pour réactiver l'authentification dans Étape 9.

2. **Sur l'hôte avec le fichier keytab, démarrez la commande `ktutil`.**

   Remarque - Bien qu'il soit possible de créer des fichiers keytab détenus par d'autres utilisateurs, l'utilisation de l'emplacement par défaut pour le fichier keytab requiert la propriété `root`.

   ```
   # /usr/bin/ktutil
   ```

3. **Lisez le fichier keytab dans le tampon de la liste des clés.**

   ```
   ktutil: read_kt keytab
   ```

4. **Affichez le tampon de liste des clés.**

   ```
   ktutil: list
   ```

   Le tampon de la liste de clés actuel s'affiche. Notez le numéro d'emplacement du service que vous voulez désactiver.

5. **Désactivez temporairement un service d'hôte en supprimant le principal de service spécifique du tampon de la liste des clés.**

   ```
   ktutil: delete_entry slot-number
   ```

   Où *slot-number* indique le numéro d'emplacement du principal de service à supprimer dans la sortie `list`.

6. **Le tampon de la liste de clés en écriture à un nouveau fichier keytab.**

   ```
   ktutil: write_kt new-keytab
   ```

7. **Quittez la commande `ktutil`.**

   ```
   ktutil: quit
   ```

8. **Utilisez le nouveau fichier keytab d'authentification pour désactiver le du principal.**

   ```
   # mv new-keytab keytab
   ```

9. **(Facultatif) Pour réactiver le service, copiez le fichier keytab temporaire vers son emplacement d'origine.**

   ```
   # cp original-keytab keytab
   ```

Exemple 5-18 Désactivation temporaire d'un hôte Kerberos

Dans cet exemple, le service `host` sur l'hôte`denver` est temporairement désactivé. Pour réactiver le service d'hôte sur `denver`, l'administrateur copie le fichier enregistré keytab vers son emplacement d'origine.

```
denver # cp /etc/krb5/krb5.keytab /etc/krb5/krb5.keytab.save
denver # /usr/bin/ktutil
ktutil:read_kt /etc/krb5/krb5.keytab
ktutil:list
slot KVNO Principal
---- ---- ---------------------------------------
1    8 root/denver@EXAMPLE.COM
2    5 host/denver@EXAMPLE.COM
ktutil:delete_entry 2
ktutil:list
slot KVNO Principal
---- ---- ---------------------------------------
1    8 root/denver@EXAMPLE.COM
ktutil:write_kt /etc/krb5/nodenverhost.krb5.keytab
ktutil: quit
denver # cp /etc/krb5/nodenverhost.krb5.keytab /etc/krb5/krb5.keytab
```

L'hôte reste indisponible jusqu'à l'utilisateur copie le fichier enregistré jusqu'à son emplacement d'origine.

```
denver # cp /etc/krb5/krb5.keytab.save /etc/krb5/krb5.keytab
```

♦ ♦ ♦ CHAPITRE 6

Utilisation des applications Kerberos

Ce chapitre explique comment utiliser les commandes et les services basés sur Kerberos fournis dans Oracle Solaris. Vous devez être déjà familiarisé avec ces commandes (dans leurs versions d'origine) avant de lire les descriptions ci-après.

Etant donné que ce chapitre est destiné aux utilisateurs d'application, il inclut des informations sur les tickets : obtention, affichage et destruction. Ce chapitre inclut également des informations concernant la sélection ou la modification d'un mot de passe Kerberos.

Ce chapitre comprend les sections suivantes :

Pour une présentation de Kerberos, reportez-vous au Chapitre 2, A propos du service Kerberos.

Gestion des tickets Kerberos

Cette section explique comment obtenir, afficher et détruire les tickets. Pour une présentation des tickets, reportez-vous à la section "Fonctionnement du service Kerberos" à la page 44.

Dans Oracle Solaris, Kerberos est intégré à la commande `login`. Cependant, pour obtenir des tickets automatiquement, vous devez configurer le service PAM pour les services de connexion appropriés. Pour plus d'informations, reportez-vous à la page de manuel`pam_krb5`(5).

La commande `ssh` peut être configurée de manière à fournir des copies de vos tickets aux autres hôtes, ce qui vous évite de devoir demander explicitement des billets pour obtenir l'accès à ces hôtes. Pour des raisons de sécurité, votre administrateur peut interdire cette opération. Pour plus d'informations, reportez-vous à la rubrique concernant le transfert d'agent dans la page de manuel`ssh`(1).

Pour plus d'informations concernant la durée de vie des tickets, reportez-vous à la section "Durée de vie des tickets" à la page 184.

Création d'un ticket Kerberos

Si le PAM est configuré correctement, un ticket est créé automatiquement lorsque vous vous connectez, et vous n'avez pas besoin d'effectuer une action spécifique pour obtenir un ticket. Cependant, vous devrez peut-être créer un ticket si votre ticket arrive à expiration. En outre, vous devrez peut-être utiliser un principal différent en plus de votre principal par défaut, par exemple, si vous utilisez `ssh -l` pour vous connecter à un système sous l'identité d'un autre utilisateur.

Pour créer un ticket, utilisez la commande `kinit`.

```
% /usr/bin/kinit
```

La commande `kinit` vous invite à saisir votre mot de passe. Pour la syntaxe complète de la commande `kinit`, reportez-vous à la page de manuel `kinit(1)`.

EXEMPLE 6-1 Création d'un ticket Kerberos

Cet exemple illustre un utilisateur, `kdoe`, créant un ticket sur son propre système.

```
% kinit
Password for kdoe@CORP.EXAMPLE.COM: xxxxxxxx
```

Dans cet exemple, l'utilisateur `kdoe` crée un ticket qui est valide pendant trois heures avec l'option `-l`.

```
% kinit -l 3h kdoe@EXAMPLE.ORG
Password for kdoe@EXAMPLE.ORG: xxxxxxxx
```

Affichage des tickets Kerberos

Tous les tickets ne sont pas similaires. Par exemple, un seul ticket peut être `transmissible`, un autre ticket peut être `postdaté` et un troisième ticket peut être à la fois transmissible et postdaté. Vous pouvez voir les billets que vous avez, ainsi que leurs attributs, en utilisant la commande `klist` avec l'option `-f` :

```
% /usr/bin/klist -f
```

Les symboles suivants indiquent les attributs qui sont associés à chaque ticket, tel qu'affiché par `klist` :

A	Préauthentifié (Preauthenticated)
D	Postdatable

d	Postdaté (Postdated)
F	Transmissible (Forwardable)
f	Transmis (Forwarded)
I	Initial
i	Non valide
P	Utilisable avec proxy (Proxiable)
p	Proxy
R	Renouvelable

La section “Types de tickets” à la page 182 décrit les différents attributs qu'un ticket peut avoir.

EXEMPLE 6-2 Affichage des tickets Kerberos

Cet exemple montre que l'utilisateur `kdoe` a un ticket `initial`, qui est `transmissible` (F) et `postdaté` (d), mais il n'a pas encore été validé (i).

```
% /usr/bin/klist -f
Ticket cache: /tmp/krb5cc_74287
Default principal: kdoe@EXAMPLE.COM

Valid starting                Expires                Service principal
09 Feb 14 15:09:51  09 Feb 14 21:09:51  nfs/EXAMPLE.COM@EXAMPLE.COM
renew until 10 Feb 14 15:12:51, Flags: Fdi
```

L'exemple suivant montre que l'utilisateur `kdoe` a deux tickets qui ont été `transmis` (f) à son hôte à partir d'un autre hôte. Les tickets sont également `transmissibles` (F).

```
% klist -f
Ticket cache: /tmp/krb5cc_74287
Default principal: kdoe@EXAMPLE.COM

Valid starting                Expires                Service principal
07 Feb 14 06:09:51  09 Feb 14 23:33:51  host/EXAMPLE.COM@EXAMPLE.COM
renew until 10 Feb 14 17:09:51, Flags: fF

Valid starting                Expires                Service principal
08 Feb 14 08:09:51  09 Feb 14 12:54:51  nfs/EXAMPLE.COM@EXAMPLE.COM
renew until 10 Feb 14 15:22:51, Flags: fF
```

L'exemple suivant montre comment afficher les types de chiffrement de la clé de session et du ticket en utilisant l'option `-e`. L'option `-a` est utilisée pour mapper l'adresse de l'hôte à un nom d'hôte si le service de noms peut faire la conversion.

```
% klist -fea
Ticket cache: /tmp/krb5cc_74287
Default principal: kdoe@EXAMPLE.COM

Valid starting                Expires                Service principal
07 Feb 14 06:09:51  09 Feb 14 23:33:51  krbtgt/EXAMPLE.COM@EXAMPLE.COM
renew until 10 Feb 14 17:09:51, Flags: FRIA
Etype(skey, tkt): AES-256 CTS mode with 96-bit SHA-1 HMAC
Addresses: client.example.com
```

Destruction des tickets Kerberos

Si vous souhaitez détruire tous les tickets Kerberos acquis au cours de votre session en cours, utilisez la commande `kdestroy`. La commande détruit votre cache des informations d'identification, ce qui détruit toutes vos informations d'identification et tous les tickets. Bien que cette destruction ne soit pas généralement nécessaire, l'exécution de `kdestroy` réduit le risque de compromettre le cache des informations d'identification pendant que vous n'êtes pas connecté.

Pour détruire vos tickets, utilisez la commande `kdestroy`.

```
% /usr/bin/kdestroy
```

La commande `kdestroy` détruit vos tickets *all*. Vous ne pouvez pas utiliser cette commande pour détruire sélectivement un ticket donné.

Si vous vous éloignez du système, vous devez utiliser la commande `kdestroy` ou verrouiller l'écran avec un économiseur d'écran.

Gestion de mot de passe Kerberos

Avec le service Kerberos configuré, vous avez maintenant deux mots de passe : votre mot de passe Oracle Solaris normal et un mot de passe Kerberos. Vous pouvez faire en sorte que les deux mots de passe soient le même, ou ils peuvent être différents.

Changement de mot de passe

Si le PAM est correctement configuré, vous pouvez modifier votre mot de passe Kerberos de deux manières.

- Utilisez la commande `passwd`. Avec le service Kerberos configuré, la commande `passwd` vous invite également automatiquement à entrer un nouveau mot de passe Kerberos.

  A l'aide de `passwd`, vous pouvez définir les deux mots de passe Kerberos et UNIX en même temps. Toutefois, vous pouvez ne modifier qu'un seul mot de passe avec `passwd` et ne pas modifier l'autre.

Remarque - Le comportement de `passwd` dépend de la façon dont le module PAM est configuré. Vous pouvez être amené à modifier les deux mots de passe dans certaines configurations. Pour certains sites, le mot de passe UNIX doit être modifié, alors que d'autres sites nécessitent la modification du mot de passe Kerberos.

- Utilisez la commande `kpasswd`. `kpasswd` modifie uniquement les mots de passe Kerberos. Vous devez utiliser `passwd` si vous souhaitez modifier votre mot de passe UNIX.

  Une utilisation principale de `kpasswd` consiste à modifier un mot de passe pour un principal Kerberos qui n'est pas un utilisateur UNIX valide. Par exemple, `jdoe/admin` est un principal Kerberos mais pas un utilisateur UNIX réel, vous devez donc utiliser `kpasswd` pour modifier le mot de passe.

Après avoir modifié votre mot de passe, le mot de passe doit se propager par le biais du réseau. En fonction de la taille du réseau Kerberos, le temps requis pour la propagation peut être comprise entre quelques minutes à une heure ou plus. Si vous avez besoin d'obtenir de nouveaux tickets Kerberos peu de temps après avoir modifié votre mot de passe, essayez d'abord le nouveau mot de passe. Si le nouveau mot de passe ne fonctionne pas, essayez de nouveau à l'aide de l'ancien mot de passe.

La *stratégie* Kerberos définit les critères des mots de passe. Une stratégie peut être défini pour chaque utilisateur ou une stratégie par défaut peut s'appliquer. Pour plus d'informations sur les stratégies, reportez-vous à la section “Administration des stratégies Kerberos” à la page 162. Pour obtenir un exemple qui recense les critères pouvant être définis pour les mots de passe Kerberos, reportez-vous àExemple 5-9, “Visualisation des attributs d'une stratégie Kerberos”. Caractères de mot de passe, les sessions de cours sont des nombres, des minuscules, des majuscules, et tous les autres caractères de ponctuation.

Connexions à distance dans Kerberos

Comme dans Oracle Solaris, Kerberos fournit la commande `ssh` pour l'accès à distance. Après l'installation, la commande `ssh` constitue le seul service réseau acceptant les demandes réseau. Par conséquent, d'autres services réseau qui sont modifiés pour Kerberos, tels que `rlogin` et `telnet`, sont désactivés. Pour plus d'informations, reportez-vous à la section “Programmes réseau Kerberos” à la page 49 et à la section “Commandes utilisateur Kerberos” à la page 176.

Commandes utilisateur Kerberos

Le produit Kerberos V5 est un système *à connexion unique*, ce qui signifie que vous n'avez à saisir votre mot de passe qu'une seule fois lors de l'utilisation d'applications réseau. Les applications Kerberos V5 effectuent l'authentification et éventuellement le chiffrement pour vous car Kerberos a été intégré dans chacune des suites d'applications réseau conviviales existantes. Les applications de Kerberos V5 sont des versions d'applications réseau UNIX existantes auxquels des fonctions Kerberos ont été ajoutées.

Remarque - En général, l'application `ssh` doit suffire pour vos besoins de connexion à distance. Reportez-vous à la page de manuel `ssh`(1). Pour plus d'informations sur l'activation d'une application réseau en phase d'abandon, reportez-vous à la section “Programmes réseau Kerberos” à la page 49.

Pour les fonctions Kerberos dans les applications réseau existantes, reportez-vous aux pages de manuel suivantes et en particulier, aux sections `OPTIONS` :

- `ftp`(1)
- `rcp`(1)
- `rlogin`(1)
- `rsh`(1)
- `telnet`(1)

Lorsque vous utilisez une application utilisant Kerberos pour vous connecter à un système distant, l'application, le KDC, et si le système distant effectuent un ensemble de négociations rapides. Lorsque ces négociations sont terminées et que votre application a prouvé votre identité en votre nom au système distant, l'hôte distant vous accorde l'accès.

Pour obtenir des exemples d'utilisation des commandes de connexion à distance en phase d'abandon, reportez-vous à "Utilisation des applications Kerberos (tâches)" dans *Administration Oracle Solaris 11.1 : Services de sécurité*.

♦ ♦ ♦ CHAPITRE 7

Référence de service Kerberos

Ce chapitre présente un grand nombre de fichiers, de commandes et de démons faisant partie du produit Kerberos.

Ce chapitre traite des informations de référence suivantes :

Fichiers Kerberos

Les fichiers suivants sont utilisés par le service Kerberos.

`~/.gkadmin`	Valeurs par défaut pour la création de nouveaux principaux dans l'interface graphique d'administration Kerberos
`~/.k5login`	Liste des principaux qui accordent l'accès à un compte Kerberos
`/etc/krb5/kadm5.acl`	Fichier de liste de contrôle d'accès Kerberos, qui inclut des noms de principal des administrateurs KDC et leurs privilèges d'administration Kerberos
`/etc/krb5/kdc.conf`	Fichier de configuration KDC
`/etc/krb5/kpropd.acl`	Fichier de configuration de base de données Kerberos
`/etc/krb5/krb5.conf`	Fichier de configuration de domaine Kerberos

`/etc/krb5/krb5.keytab`	Fichier Keytab pour serveurs d'application de réseau
`/etc/krb5/warn.conf`	Avertissement d'expiration de ticket Kerberos et fichier de configuration de renouvellement automatique
`/etc/pam.conf`	Fichier de configuration PAM
`/tmp/krb5cc_`*UID*	Cache d'informations d'identification par défaut, où *UID* est l'ID utilisateur décimal de l'utilisateur
`/tmp/ovsec_adm.`*xxxxxx*	Cache d'informations d'identification temporaire pour la durée de vie des opérations de modification de mot de passe, où *xxxxxx* est une chaîne aléatoire
`/var/krb5/.k5.`*REALM*	Le fichier stach de KDC qui contient une copie de la clé principale de KDC
`/var/krb5/kadmin.log`	Fichier journal pour `kadmind`
`/var/krb5/kdc.log`	Fichier journal pour le KDC
`/var/krb5/principal`	Base de données du principal Kerberos
`/var/krb5/principal.kadm5`	Base de données d'administration Kerberos qui contient des informations de stratégie
`/var/krb5/principal.kadm5.lock`	Fichier de verrouillage de la base de données d'administration Kerberos
`/var/krb5/principal.ok`	Fichier d'initialisation de la base de données de principaux Kerberos, créé lors de l'initialisation réussie de la base de données Kerberos
`/var/krb5/principal.ulog`	Fichier journal de mise à jour Kerberos contenant des mises à jour pour la propagation incrémentielle
`/var/krb5/slave_datatrans`	Sauvegarde du KDC que le script `kprop_script` utilise pour la propagation
`/var/krb5/slave_datatrans_`*slave*	Fichier de vidage temporaire créé lorsque des mises à jour complètes sont effectuées sur l'esclave spécifié
`/var/user/`*$USER*`/krb-warn.conf`	Fichier de configuration de renouvellement automatique et d'avertissement d'expiration des tickets par utilisateur

Commandes Kerberos

Les commandes suivantes incluses dans le produit Kerberos. Elles sont répertoriées par leur page de manuel.

`ftp`(1)	Application de protocole de transfert de fichier
`kdestroy`(1)	Détruit les tickets Kerberos
`kinit`(1)	Obtient et met en cache les tickets d'octroi de tickets Kerberos
`klist`(1)	Affiche les tickets actuels Kerberos
`kpasswd`(1)	Modifie un mot de passe Kerberos
`ktutil`(1)	Gère les fichiers keytab Kerberos
`kvno`(1)	Répertorie les numéros de version clé pour les principaux Kerberos
`rcp`(1)	Application de copie de fichier à distance
`rlogin`(1)	Application de connexion à distance
`rsh`(1)	Application shell distante
`scp`(1)	Application de copie de fichier distant sécurisé
`sftp`(1)	Application de transfert de fichier sécurisé
`ssh`(1)	Shell sécurisé pour connexion à distance
`telnet`(1)	Application `telnet` utilisant Kerberos
`kprop`(1M)	Programme de propagation de base de données Kerberos
`gkadmin`(1M)	Programme d'interface graphique d'administration de base de données Kerberos utilisé pour gérer les principaux et les stratégies
`gsscred`(1M)	Gérez les entrées de la table `gsscred`
`kadmin`(1M)	Programme d'administration de base de données Kerberos à distance qui requiert l'authentification Kerberos, utilisé pour gérer les principaux, les stratégies et les fichiers keytab

kadmin.local(1M)	Programme d'administration de base de données Kerberos local qui s'exécute sur le KDC maître qui est utilisé pour gérer les principaux, les stratégies et les fichiers keytab
kclient(1M)	Script d'installation du client Kerberos utilisé avec ou sans profil d'installation
kdb5_ldap_util(1M)	Crée des conteneurs LDAP pour les bases de données Kerberos
kdb5_util(1M)	Crée des bases de données Kerberos et des fichiers stash
kdcmgr(1M)	Configure les KDC maître et esclave Kerberos
kproplog(1M)	Présente un récapitulatif des entrées de mise à jour dans le journal de mise à jour

Démons Kerberos

Les démons suivants sont utilisés par le produit Kerberos.

/usr/lib/inet/proftpd	Démon de protocole de transfert de fichier sécurisé
/usr/lib/ssh/sshd	Démon de shell sécurisé
/usr/lib/krb5/kadmind	Démon d'administration de base de données Kerberos
/usr/lib/krb5/kpropd	Démon de propagation de base de données Kerberos
/usr/lib/krb5/krb5kdc	Démon de traitement des tickets Kerberos
/usr/lib/krb5/ktkt_warnd	Démon de renouvellement automatique et d'avertissement d'expiration des tickets Kerberos
/usr/sbin/in.rlogind	Démon de connexion à distance
/usr/sbin/in.rshd	Démon de shell à distance
/usr/sbin/in.telnetd	telnet, démon

Terminologie Kerberos

Ce qui suit les termes Kerberos sont utilisés dans la documentation Kerberos. Une bonne compréhension de ces termes est essentielle pour appréhender les concepts Kerberos.

Terminologie spécifique à Kerberos

Vous devez comprendre les termes de cette section pour pouvoir administrer les KDC.

Le *Centre de distribution des clés* ou *KDC* est le composant de Kerberos responsable de l'émission des informations d'identification. Ces informations d'identification sont créées à l'aide des informations stockées dans la base de données KDC. Chaque domaine a besoin d'au moins deux KDC, un maître et au moins un esclave. Tous les KDC génèrent des informations d'identification, mais seul le KDC maître gère toutes les modifications apportées à la base de données KDC.

Un *Fichier stash* contient la clé principale pour le KDC. Cette clé est utilisée lorsqu'un serveur est redémarré pour authentifier automatiquement le KDC avant qu'il ne démarre les commandes `kadmind` et `krb5kdc`. Etant donné que ce fichier inclut la clé principale, ce fichier et toutes ses sauvegardes doivent être sécurisés. Le fichier est créé avec des autorisations en lecture seule pour `root`. Pour garder le fichier sécurisé, vous ne devez pas modifier les autorisations. Si le fichier est compromis, la clé peut être utilisée pour accéder à la base de données KDC ou la modifier.

Terminologie spécifique à l'authentification

Vous devez connaître les termes de cette section pour comprendre le processus d'authentification. Les programmeurs et les administrateurs système doivent être familiarisés avec ces termes.

Un *client* est un logiciel qui s'exécute sur le poste de travail d'un utilisateur. Le logiciel Kerberos qui s'exécute sur le client effectue de nombreuses demandes au cours de ce processus. Par conséquent, il est important de différencier les actions de ce logiciel de celles de l'utilisateur.

Les termes *server* et *service* sont souvent interchangeables. Pour clarifier, le terme *server* est utilisé pour définir le système physique sur lequel le logiciel Kerberos est exécuté. Le terme *service* correspond à une fonction particulière prise en charge sur un serveur (par exemple, `ftp` ou `nfs`). Dans la plupart des cas, la documentation mentionne les serveurs dans le cadre d'un service, mais cette définition obscurcit la signification des termes. Par conséquent, le terme *server* fait référence au système physique. Le terme *service* fait référence au logiciel.

Le produit Kerberos utilise deux types de clés. Un type de clé est la clé est dérivée d'un mot, qui est attribué à chaque principal de l'utilisateur et est uniquement connue du KDC et à l'utilisateur. L'autre type de clé est une clé aléatoire qui n'est pas associée à un mot de passe et donc n'est pas adaptée à l'utilisation par les principaux d'utilisateur. Les clés aléatoires sont généralement utilisées pour les principaux de service qui ont des entrées dans un fichier keytab et dont les clés de session sont générées par le KDC. Les principaux de service peuvent utiliser des clés aléatoires étant donné que le service peut accéder à la clé dans le fichier keytab qui lui permet de fonctionner de manière non interactive. Les clés de session sont générées par le KDC (et partagées entre le client et le service) pour assurer la sécurité des transactions entre un client et un service.

Un *ticket* est un paquet d'informations servant à transmettre en toute sécurité l'identité d'un utilisateur à un serveur ou un service. Un ticket n'est valable que pour un client et un service particulier sur un serveur spécifique. Un ticket contient les informations suivantes :

- Nom du principal du service
- Nom du principal de l'utilisateur
- Adresse IP de l'hôte de l'utilisateur
- Timestamp
- Valeur définissant la durée de vie du ticket.
- Copie de la clé de session

Toutes ces données sont chiffrées dans la clé de service du serveur. Le KDC émet le ticket intégré dans les informations d'identification. Une fois le ticket émis, il peut être réutilisé jusqu'à son expiration.

Les *informations d'identification* sont un paquet d'informations comprenant un ticket ainsi que la clé de session correspondante. Les informations d'identification sont chiffrées avec la clé du principal effectuant la demande. En règle générale, le KDC génère des informations d'identification en réponse à une demande de ticket d'un client.

Un *authentificateur* correspond à des informations que le serveur utilise pour authentifier le principal de l'utilisateur du client. Un authentificateur inclut le nom du principal de l'utilisateur, un horodatage et d'autres données. A la différence d'un ticket, un authentificateur ne sert qu'une seule fois, généralement lorsque l'accès à un service est demandé. Un authentificateur est chiffré à l'aide de la clé de session partagée par le client et le serveur. En général, le client crée l'authentificateur et l'envoie à l'aide du ticket du serveur ou du service pour s'authentifier auprès du serveur ou du service.

Types de tickets

Les tickets ont des propriétés qui régissent la façon dont ils peuvent être utilisés. Ces propriétés sont assignées au ticket lors de sa création et peuvent être modifiées ultérieurement.

Par exemple, un ticket peut passer de `forwardable` à `forwarded`. Vous pouvez visualiser les propriétés de ticket à l'aide de la commande `klist`. Voir “Affichage des tickets Kerberos” à la page 172.

Les tickets peuvent être décrits par un ou plusieurs des termes suivants :

Transmissible/ transmis
: Un ticket transmissible peut être envoyé à partir d'un hôte vers un autre, supprimant la nécessité pour un client de se réauthentifier. Par exemple, si l'utilisateur `david` obtient un ticket transmissible sur la machine de l'utilisateur `jennifer`, `david` peut se connecter à sa propre machine sans devoir obtenir un nouveau ticket (et donc s'authentifier à nouveau). Reportez-vous à Exemple 6-1, “Création d'un ticket Kerberos” pour consulter un exemple de ticket transmissible.

Initial
: Un ticket initial est un ticket émis directement, et non sur la base d'un ticket d'octroi de tickets. Certains services, tels que les applications modifiant les mots de passe, peuvent nécessiter des tickets marqués comme étant initiaux afin d'assurer que le client puisse démontrer qu'il connaît sa clé secrète. Un ticket initial indique que le client s'est récemment authentifié et ne dépend pas d'un ticket d'octroi de tickets qui peut être plus ancien.

Non valide
: Un ticket non valide est un ticket postdaté qui n'est pas encore devenu utilisable. Un ticket non valide est rejeté par un serveur d'application jusqu'à ce qu'il soit validé. Pour être validé, un ticket doit être présenté au KDC par le client dans une demande de TGS, avec l'indicateur `VALIDATE`, après l'heure de début.

Postdatable/ postdaté
: Un ticket postdaté est un ticket qui ne devient valide qu'après une période spécifiée après sa création. Par exemple, un tel ticket peut être utile avec les tâches exécutées par lots la nuit car le ticket, s'il est volé, ne peut pas être utilisé tant que l'exécution de ces tâches n'a pas eu lieu. Lorsqu'un ticket postdaté est émis, il est émis en tant que non valide et le reste jusqu'à ce que son heure de début soit dépassée, et que le client demande la validation par le KDC. Un ticket postdaté est normalement valide jusqu'à l'heure d'expiration du ticket d'octroi de tickets. Toutefois, si le ticket est marqué comme renouvelable, sa durée de vie est normalement égale à la durée de vie entière du ticket d'octroi de tickets.

Utilisable avec proxy/proxy
: A certains moments, principal peut s'avérer nécessaire de permettre à un service d'effectuer une opération en son nom. Le nom du principal du proxy doit être spécifié lors de la création du ticket. Oracle Solaris ne prend pas en charge les tickets utilisables avec proxy ou les tickets proxy.

  Un ticket utilisable avec proxy est similaire à un ticket transmissible à ceci près qu'il n'est valide que pour un seul service. Un ticket

transmissible accorde le service auquel la utilisation complète de l'identité du client. Un ticket transmissible peut par conséquent être considéré comme une sorte de super-proxy.

Renouvelable

Comme les tickets avec de très longues durées de vie impliquent un risque en matière de sécurité, les tickets peuvent être désignés comme renouvelables. Un ticket renouvelable possède deux moments d'expiration : l'heure à laquelle l'instance courante du ticket expire et la durée de vie maximale de tout ticket (une semaine). Si un client souhaite continuer à utiliser un ticket, il peut le renouveler avant sa première expiration. Par exemple, supposons qu'un ticket peut être valide pendant une heure, et que tous les tickets ont une durée de vie maximale de dix heures. Si le client détenant le ticket souhaite le conserver plus d'une heure, il doit le renouveler dans l'heure. Lorsqu'un ticket atteint sa durée de vie maximale (dix heures), celui-ci expire automatiquement et ne peut pas être renouvelé.

Pour plus d'informations concernant la façon de visualiser les attributs des tickets, reportez-vous à la section “Affichage des tickets Kerberos” à la page 172.

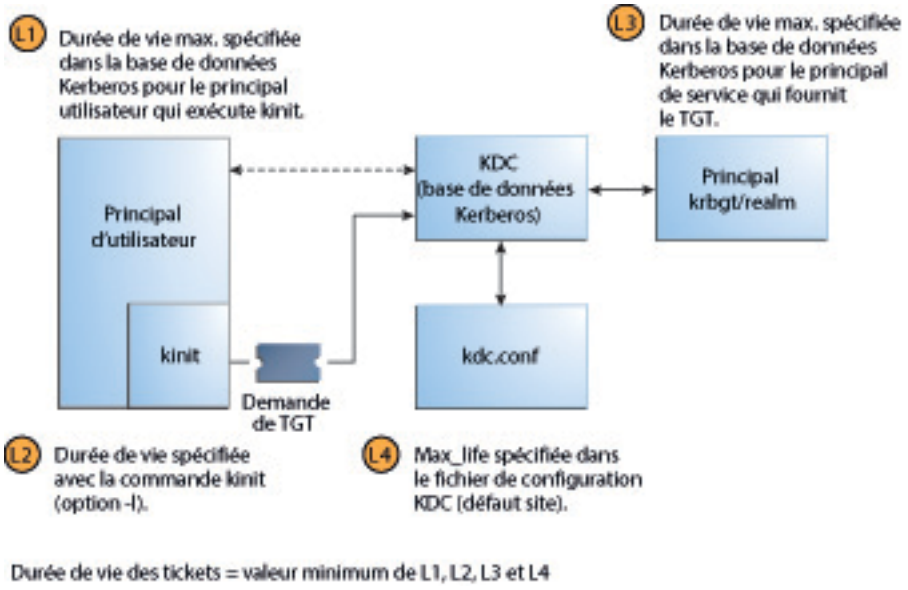
Durée de vie des tickets

Lorsqu'un principal obtient un ticket, y compris un ticket d'octroi de tickets (TGT), la durée de vie du ticket est définie comme la plus petite des valeurs de durée de vie suivantes :

- La valeur de la durée de vie qui est spécifiée par l'option `-l` de `kinit`, si `kinit` est utilisée pour obtenir le ticket. Par défaut, `kinit` utilise la valeur de durée de vie maximale
- La valeur de durée de vie maximale (`max_life`) spécifiée dans le fichier `kdc.conf`.
- Valeur de durée de vie maximale spécifiée dans la base de données Kerberos pour le principal de service fournissant le ticket. Dans le cas de `kinit`, le principal de service est `krbtgt/`*realm*.
- Valeur de durée de vie maximale spécifiée dans la base de données Kerberos pour le principal d'utilisateur demandant le ticket.

La capture d'écran suivante illustre une durée de vie d'un TGT est déterminée et d'où les quatre valeurs de durée de vie de ces dernières. Par un principal obtient un ticket, la durée de vie du ticket est déterminé de la même manière. Les deux différences sont que `kinit` n'offre pas une valeur de durée de vie et que le principal de service fournissant le ticket offre une valeur de durée de vie maximale au lieu du principal `krbtgt/`*realm*.

FIGURE 7-1 Détermination de la durée de vie d'un TGT



La durée de vie d'un ticket renouvelable est déterminée à partir de la plus basse de quatre valeurs, comme suit :

- Valeur de durée de vie renouvelable spécifiée par l'option `-r` de `kinit`, si `kinit` est utilisée pour obtenir ou renouveler le ticket.
- Valeur de durée de vie renouvelable maximale (`max_renewable_life`) spécifiée dans le fichier `kdc.conf`.
- Valeur de durée de vie renouvelable maximale spécifiée dans la base de données Kerberos pour le principal de service fournissant le ticket. Dans le cas de `kinit`, le principal de service est `krbtgt/`*realm*.
- Valeur de durée de vie renouvelable maximale spécifiée dans la base de données Kerberos pour le principal d'utilisateur demandant le ticket.

Noms de principaux Kerberos

Chaque ticket est identifié par un nom de principal. Le nom de principal peut identifier un utilisateur ou un service. Les exemples suivants montrent les noms de principal standard, procédez comme suit :

`changepw/kdc1.example.com@EXAMPLE.COM`	Principal pour le serveur KDC maître qui permet l'accès au KDC lorsque vous modifiez les mots de passe.
`clntconfig/admin@EXAMPLE.COM`	Principal utilisé par l'utilitaire d'installation `kclient`.
`ftp/boston.example.com@EXAMPLE.COM`	Un principal utilisé par le service `ftp`. Ce principal peut être utilisé à la place d'un principal `host`.
`host/boston.example.com@EXAMPLE.COM`	Principal utilisé par des applications utilisant Kerberos (`klist` et `kprop`, par exemple) et des services (tels que `ftp` et `telnet`). Ce principal est appelé `host` ou principal de service. Le principal est utilisé pour authentifier les montages NFS. Ce principal est également utilisé par un client pour vérifier que le TGT émis au client provient du bon KDC.
`K/M@EXAMPLE.COM`	Nom de principal de la clé principale. Un nom de principal de clé principale est associé à chaque KDC maître.
`kadmin/history@EXAMPLE.COM`	Principal incluant une clé utilisée pour conserver l'historique des mots de passe d'autres principaux. Chaque KDC maître possède l'un de ces principaux.
`kadmin/kdc1.example.com@EXAMPLE.COM`	Principal pour le serveur KDC maître qui permet l'accès au KDC à l'aide de la commande `kadmind`.
`kadmin/changepw.example.com@EXAMPLE.COM`	Principal utilisé pour accepter les demandes de modification de mots de passe émises par les clients qui n'exécutent pas une version d'Oracle Solaris.
`krbtgt/EXAMPLE.COM@EXAMPLE.COM`	Ce principal est utilisé lors de la génération d'un ticket d'octroi de tickets.
`krbtgt/EAST.EXAMPLE.COM@WEST.EXAMPLE.COM`	Ce principal est un exemple de ticket d'octroi de tickets inter-domaine.
`nfs/boston.example.com@EXAMPLE.COM`	Principal utilisé par le service NFS. Ce principal peut être utilisé à la place d'un principal `host`.
`root/boston.example.com@EXAMPLE.COM`	Principal associé au compte `root` sur un client. Ce principal est appelé principal `root` et fournit un accès `root` aux systèmes de fichiers montés NFS.
username`@EXAMPLE.COM`	Principal d'un utilisateur.
username`/admin@EXAMPLE.COM`	Principal `admin` pouvant être utilisé pour l'administration de la base de données KDC.

♦ ♦ ♦ CHAPITRE 8

Messages d'erreur et dépannage de Kerberos

Ce chapitre fournit les solutions aux messages d'erreur que vous pouvez recevoir lorsque vous utilisez le service Kerberos. Ce chapitre fournit également quelques conseils de dépannage pour différents problèmes.

Ce chapitre comprend les sections suivantes :

Messages d'erreur Kerberos

Cette section fournit des informations sur les messages d'erreur Kerberos, y compris la raison de chaque erreur et une façon de corriger la corriger.

Messages d'erreur Gkadmin, GUI

`Unable to view the list of principals or policies; use the Name field.`

Origine : Le principal `admin` avec lequel vous vous êtes connecté n'a pas de privilège de liste (`l`) dans le fichier ACL de Kerberos (`kadm5.acl`). Par conséquent, vous ne pouvez pas afficher la liste des principaux ou la liste des stratégies.

Solution : Vous devez saisir les noms des principaux et des stratégies dans le champ de nom pour travailler sur ces derniers, ou vous devez vous connecter à l'aide d'un principal qui dispose des privilèges appropriés.

`JNI: Java * failed`

Origine : Un grave problème existe avec l'interface native Java utilisée par l'interface graphique `gkadmin`.

Solution : Quittez `gkadmin` et redémarrez-le. Si le problème persiste, signalez un bogue.

Messages d'erreur Kerberos courants (A-M)

Cette section fournit une liste alphabétique (A-M) des messages d'erreur courants pour les commandes Kerberos, les démons Kerberos, la structure PAM, l'interface GSS, le service NFS et la bibliothèque Kerberos.

`Bad lifetime value`

Origine : La valeur de durée de vie fournie n'est pas valide ou incorrectement formatée.

Solution : Assurez-vous que la valeur fournie est cohérente avec la section des formats d'heure de la page de manuel `kinit`(1).

`Bad start time value`

Origine : La valeur d'heure de démarrage fournie n'est pas valide ou incorrectement formatée.

Solution : Assurez-vous que la valeur fournie est cohérente avec la section des formats d'heure de la page de manuel `kinit`(1).

`Cannot contact any KDC for requested realm`

Origine : Aucun KDC n'a répondu dans le domaine demandé.

Solution : Assurez-vous qu'au moins un KDC (maître ou esclave) est accessible ou que le démon `krb5kdc` est en cours d'exécution sur les KDC. Consultez le fichier `/etc/krb5/krb5.conf` pour la liste de KDC configurés (`kdc =` *kdc-name*).

`Cannot determine realm for host: host is '`*hostname*`'`

Origine : Kerberos n'est pas en mesure de déterminer le nom de domaine de l'hôte.

Solution : Assurez-vous qu'il y a un nom de domaine par défaut ou que les mappages de nom de domaine sont définis dans le fichier de configuration Kerberos (`krb5.conf`).

`Cannot find a kadmin KDC entry in krb5.conf(4) or DNS Service Location records for realm '`*realmname*`'`

`Cannot find a kpassword KDC entry in krb5.conf(4) or DNS Service Location records for realm '`*realmname*`'`

`Cannot find a master KDC entry in krb5.conf(4) or DNS Service Location records for realm '`*realmname*`'`

`Cannot find any KDC entries in krb5.conf(4) or DNS Service Location records for realm '`*realmname*`'`

Origine : Le fichier `krb5.conf` ou l'enregistrement du serveur DNS n'est pas correctement configuré.

Solution : Assurez-vous que le fichier de configuration Kerberos (`/etc/krb5/krb5.conf`) ou les enregistrements du serveur DNS pour le KDC sont correctement configurés.

`Cannot find address for '`*hostname*`': '`*error-string* `'`

Origine : Aucune adresse n'a été trouvée dans les enregistrements du DNS pour le nom d'hôte donné.

Solution : Corrigez l'enregistrement d'hôte dans le DNS ou corrigez l'erreur dans la recherche DNS.

`cannot initialize realm` *realm-name*

Origine : Le KDC n'a peut-être pas de fichier stash.

Solution : Assurez-vous que le KDC dispose d'un fichier stash. Si tel n'est pas le cas, créez un fichier stash en utilisant la commande `kdb5_util` et essayez de redémarrer la commande `krb5kdc`.

`Cannot resolve KDC for requested realm`

Origine : Kerberos n'est pas en mesure de déterminer un KDC pour le domaine.

Solution : Assurez-vous que le fichier de configuration Kerberos (`krb5.conf`) indique un KDC dans la section `realm`.

`Cannot resolve network address for KDCs '`*hostname*`' discovered via DNS Service Location records for realm '`*realm-name*`'`

`Cannot resolve network address for KDCs '`*hostname*`' specified in krb5.conf(4) for realm '`*realm-name*`'`

Origine : Le fichier `krb5.conf` ou l'enregistrement du serveur DNS n'est pas configuré correctement.

Solution : Assurez-vous que le fichier de configuration Kerberos (`/etc/krb5/krb5.conf`) et les enregistrements du serveur DNS pour le KDC sont correctement configurés.

`Can't open/find Kerberos configuration file`

Origine : Le fichier de configuration Kerberos (`krb5.conf`) n'était pas disponible.

Solution : Assurez-vous que le `krb5.conf` est disponible au bon emplacement et qu'il dispose des autorisations nécessaires. Ce fichier doit être accessible en écriture par `root` et lisible par tout le monde.

`Client '`*principal*`' pre-authentication failed`

Origine : L'authentification du principal a échoué.

Solution : Assurez-vous que l'utilisateur utilise le mot de passe correct.

`Client or server has a null key`

Origine : Le principal a une clé nulle.

Solution : Modifiez le principal afin qu'il dispose d'une clé non nulle en utilisant la commande `cpw` de `kadmin`.

`Communication failure with server while initializing kadmin interface`

Origine : Le démon `kadmind` n'était pas en cours d'exécution sur l'hôte qui était spécifié pour le KDC maître.

Solution : Assurez-vous d'avoir spécifié le nom d'hôte correct pour le KDC maître. Si vous avez spécifié le nom d'hôte correct, assurez-vous que `kadmind` est en cours d'exécution sur le KDC maître que vous avez spécifié.

`Credentials cache file permissions incorrect`

Origine : Vous ne disposez pas des autorisations de lecture ou d'écriture sur le cache d'informations d'identification (`/tmp/krb5cc_`*uid*).

Solution : Assurez-vous que vous disposez des autorisations de lecture ou d'écriture sur le cache d'informations d'identification.

`Credentials cache I/O operation failed` *XXX*

Origine : Kerberos a rencontré un problème lors de l'écriture dans le cache d'informations d'identification du système (`/tmp/krb5cc_`*uid*).

Solution : Assurez-vous que le cache d'informations d'identification n'a pas été supprimé et qu'il reste de l'espace sur le périphérique en utilisant la commande `df`.

`Decrypt integrity check failed`

Origine : Vous avez peut-être un ticket non valide.

Solution : Vérifiez les deux conditions suivantes :

- Assurez-vous que vos informations d'identification sont valides. Détruisez vos tickets avec `kdestroy` et créez de nouveaux tickets avec `kinit`.
- Assurez-vous que l'hôte cible dispose d'un fichier keytab avec la version correcte de la clé du service. Utilisez `kadmin` pour afficher le numéro de version de la clé du principal service (par exemple, `host/`*FQDN-hostname*) dans la base de données Kerberos. Utilisez également la commande `klist -k` sur l'hôte cible pour vérifier que le numéro de version de la clé est identique.

`Decrypt integrity check failed for client 'principal' and server 'hostname'`

Origine : Vous avez peut-être un ticket non valide.

Solution : Assurez-vous que vos informations d'identification sont valides. Détruisez vos tickets avec la commande `kdestroy`, puis créez de nouveaux tickets avec la commande `kinit`.

`failed to obtain credentials cache`

Origine : Pendant l'initialisation de `kadmin`, une panne s'est produite lorsque `kadmin` a essayé afin d'obtenir des informations d'identification pour le principal `admin`.

Solution : Assurez-vous d'avoir utilisé le bon principal et le bon mot de passe lorsque vous avez exécuté la commande `kadmin`.

`Field is too long for this implementation`

Origine : Le message qui a été envoyé par une application utilisant Kerberos était trop long. Cette erreur peut être générée si le protocole de transport est UDP, dont la taille maximale de message par défaut est 65535 octets. En outre, il existe des limites sur les champs individuels dans un message de protocole qui est envoyé par le service Kerberos.

Solution : Vérifiez que vous n'avez pas restreint le transport à UDP dans le fichier `/etc/krb5/kdc.conf` du serveur KDC.

`GSS-API (or Kerberos) error`

Origine : Ce message est un message d'erreur GSS-API ou Kerberos générique pouvant être causé par divers problèmes.

Solution : Vérifiez le fichier `/var/krb5/kdc.log` pour trouver le message d'erreur plus spécifique qui a été enregistré lorsque l'erreur s'est produite.

`Improper format of Kerberos configuration file`

Origine : Le fichier de configuration Kerberos a des entrées non valides.

Solution : Assurez-vous que toutes les relations dans le fichier `krb5.conf` sont suivies du signe "=" et d'une valeur. En outre, vérifiez que les crochets sont présents dans les paires pour chaque sous-section.

`Invalid credential was supplied`
`Service key not available`

Origine : Le ticket de service du cache d'informations d'identification est peut-être incorrect.

Solution : Détruisez le cache d'informations d'identification actuel et exécutez de nouveau la commande `kinit` avant d'essayer d'utiliser ce service.

`Invalid flag for file lock mode`

Origine : Une erreur Kerberos interne s'est produite.

Solution : Veuillez signaler un bogue.

`Invalid message type specified for encoding`

Origine : Kerberos n'a pas reconnu le type de message qui a été envoyé par l'application utilisant Kerberos.

Solution : Si vous utilisez une application utilisant Kerberos qui a été développé par votre site ou un fournisseur, assurez-vous qu'elle utilise Kerberos correctement.

`kadmin: Bad encryption type while changing host/`*FQDN*`'s key`

Origine : Plusieurs types de chiffrement par défaut sont inclus dans la version de base dans les dernières versions. Les clients peuvent demander des types de chiffrement qui ne sont peut-être pas pris en charge par un KDC exécuté sur une version antérieure du logiciel.

Solution : Définissez `permitted_enctypes` dans `krb5.conf` sur le client pour que le type de chiffrement `aes256` ne soit pas inclus. Cette étape doit être effectuée sur chaque nouveau client.

`KDC can't fulfill requested option`

Origine : Le KDC n'a pas autorisé l'option demandée. Il est possible que des options postdatables ou transmissibles soient demandées et que le KDC refuse. Un autre problème peut être une demande de renouvellement d'un TGT, sans avoir de TGT renouvelable.

Solution : Déterminez si vous demandez une option que le KDC n'autorise pas ou un type de ticket qui n'est pas disponible.

```
KDC reply did not match expectation: KDC not found. Probably got an unexpected realm
referral
```

Origine : La réponse du KDC ne contient pas le nom de principal attendu ou d'autres valeurs dans la réponse n'étaient pas correctes.

Solution : Assurez-vous que le KDC avec lequel vous communiquez est conforme à RFC4120, que la demande envoyée est une demande Kerberos V5 ou que le KDC est disponible.

```
kdestroy: Could not obtain principal name from cache
```

Origine : Le cache d'informations d'identification est manquant ou endommagé.

Solution : Vérifiez que l'emplacement du cache fourni est correct. Supprimez et obtenez un nouveau TGT via `kinit`, si nécessaire.

```
kdestroy: No credentials cache file found while destroying cache
```

Origine : Le cache d'informations d'identification (`/tmp/krb5c_`*uid*) est manquant ou endommagé.

Solution : Vérifiez que l'emplacement du cache fourni est correct. Supprimez et obtenez un nouveau TGT via `kinit`, si nécessaire.

```
kdestroy: TGT expire warning NOT deleted
```

Origine : Le cache d'informations d'identification est manquant ou endommagé.

Solution : Vérifiez que l'emplacement du cache fourni est correct. Supprimez et obtenez un nouveau TGT via `kinit`, si nécessaire.

```
Kerberos authentication failed
```

Origine : Le mot de passe Kerberos est incorrect ou ne peut pas être synchronisé avec le mot de passe UNIX.

Solution : Si les mots de passe ne sont pas synchronisés, vous devez spécifier le mot de passe Kerberos pour terminer l'authentification. Il est possible que l'utilisateur ait oublié son mot de passe d'origine.

`Key version` *number* `is not available for principal` *principal*

Origine : La version des clés ne correspond pas à la version des clés sur le serveur d'application.

Solution : Vérifiez la version des clés sur le serveur d'application à l'aide de la commande `klist -k`.

`Key version number for principal in key table is incorrect`

Origine : La version de clé d'un principal dans le fichier keytab est différente de la version dans la base de données Kerberos. Soit la clé d'un service a été modifiée, soit vous utilisez un ancien ticket de service.

Solution : Si la clé d'un service a été modifiée (par exemple, en utilisant `kadmin`), vous devez extraire la nouvelle clé et la stocker dans le fichier keytab de l'hôte où le service est en cours d'exécution.

Sinon, vous devrez peut-être utiliser un ancien ticket de service disposant d'une ancienne clé. Vous aurez peut-être besoin d'exécuter la commande `kdestroy`, puis la commande `kinit` à nouveau.

`kinit: gethostname failed`

Origine : Une erreur dans la configuration réseau local provoque l'échec de `kinit`.

Solution : Assurez-vous que l'hôte est correctement configuré.

`login: load_modules: can not open module /usr/lib/security/pam_krb5.so.1`

Origine : Le module PAM de Kerberos est manquant ou il ne s'agit pas d'un binaire exécutable valide.

Solution : Assurez-vous que le module PAM de Kerberos est dans les `/usr/lib/security` Directory et qu'il s'agit d'un fichier exécutable valide binaire. De même, assurez-vous que votre fichier de configuration PAM pour `login` contient le chemin d'accès correct à `pam_krb5.so.1`.

`Looping detected getting initial creds: '`*client-principal*`' requesting ticket '`*service-principal*`'. Max loops is` *value*`. Make sure a KDC is available.`

Origine : Kerberos a tenté à plusieurs reprises d'obtenir les tickets initiaux mais n'a pas réussi.

Solution : Assurez-vous qu'au moins un KDC répond aux demandes d'authentification.

`Master key does not match database`

Origine : Le vidage de base de données chargé n'a pas été créé à partir d'une base de données qui contient la clé principale. La clé principale est située dans `/var/krb5/.k5.`*REALM*.

Solution : Assurez-vous que la clé principale dans le vidage de base de données chargé correspond à la clé principale qui se trouve dans `/var/krb5/.k5.`*REALM*.

`Matching credential not found`

Origine : Les informations d'identification correspondant à votre demande n'ont pas été trouvées. Votre demande exige l'utilisation d'informations d'authentification qui ne sont pas disponibles dans le cache d'informations d'identification.

Solution : Détruisez vos tickets avec `kdestroy` et créez de nouveaux tickets avec `kinit`.

`Message out of order`

Origine : Les messages qui ont été envoyés lors de l'utilisation d'une confidentialité à ordre séquentielle ont été livrés sans tenir compte de l'ordre. Certains messages ont peut-être été perdus dans le processus.

Solution : Vous devez réinitialiser la session Kerberos.

`Message stream modified`

Origine : La somme de contrôle calculée et la somme de contrôle du message ne correspondent pas. Le message a peut-être été modifié au cours de la transmission, ce qui peut indiquer un problème de sécurité.

Solution : Assurez-vous que les messages sont envoyés sur le réseau correctement. Etant donné que ce message peut également indiquer la possible altération des messages pendant leur envoi, détruisez vos tickets et réinitialisez les services Kerberos que vous êtes en train d'utiliser.

Messages d'erreur Kerberos courants (N-Z)

Cette section fournit une liste alphabétique (N-Z) des messages d'erreur courants pour les commandes Kerberos, les démons Kerberos, la structure PAM, l'interface GSS, le service NFS et la bibliothèque Kerberos.

`No credentials cache file found`

Origine : Kerberos n'a pas trouvé le cache d'informations d'identification (`/tmp/krb5cc_`*uid*).

Solution : Assurez-vous que le fichier d'informations d'identification existe et qu'il est lisible. S'il n'y figure pas, essayez d'exécuter à nouveau la commande `kinit`.

`No credentials were supplied, or the credentials were unavailable or inaccessible`
`No credential cache found`

Origine : Le cache d'informations d'identification de l'utilisateur est incorrect ou n'existe pas.

Solution : L'utilisateur doit exécuter `kinit` avant d'essayer de démarrer le service.

`No credentials were supplied, or the credentials were unavailable or inaccessible`
`No principal in keytab ('`*filename*`') matches desired name` *principal*

Origine : Une erreur s'est produite au cours d'une tentative d'authentification du serveur.

Solution : Assurez-vous que l'hôte ou le principal de service est dans le fichier keytab du serveur.

`Operation requires “`*privilege*`” privilege`

Origine : Le principal `admin` qui était en cours d'utilisation ne s'est pas vu attribuer les privilèges appropriés dans le fichier `kadm5.acl`.

Solution : Utilisez une identité qui dispose des privilèges appropriés. Vous pouvez également configurer le principal qui était en cours d'utilisation afin qu'il dispose des privilèges appropriés. Généralement, un principal contenant `/admin` dans son nom est doté des privilèges appropriés.

`PAM-KRB5 (auth): krb5_verify_init_creds failed: Key table entry not found`

Origine : L'application distante a tenté de lire le principal de service de l'hôte dans le fichier `/etc/krb5/krb5.keytab` local, mais il n'existe pas.

Solution : Ajoutez le principal de service de l'hôte au fichier keytab du serveur.

`Permission denied in replay cache code`

Origine : Le cache de rediffusion du système n'a pas pu être ouvert. Votre serveur peut avoir été exécuté pour la première fois sous un ID utilisateur différent de votre ID d'utilisateur actuel.

Solution : Assurez-vous que le cache de rediffusion possède les autorisations appropriées. Le cache de rediffusion est stocké sur l'hôte sur lequel l'application de serveur utilisant Kerberos est en cours d'exécution. Le fichier du cache de rediffusion est appelé `/var/krb5/rcache/rc_`*service_name_uid* pour les utilisateurs non`root`. Pour les utilisateurs root, le fichier du cache de rediffusion est appelé `/var/krb5/rcache/root/rc_` *service_name*.

`Protocol version mismatch`

Origine : Une demande Kerberos V4 a probablement été envoyée au KDC. Le service Kerberos ne prend en charge que le protocole Kerberos V5.

Solution : Assurez-vous que vos applications utilisent le protocole Kerberos V5.

`Request is a replay`

Origine : La demande a déjà été envoyée à ce serveur et traitée. Les tickets ont peut-être été volés et quelqu'un d'autre essaie de les réutiliser.

Solution : Attendez quelques minutes et relancez la demande.

`Requested principal and ticket don't match: Requested principal is '`*service-principal*`' and TGT principal is '`*TGT-principal*`'`

Origine : Le principal de service auquel vous vous connectez et le ticket de service que vous avez ne correspondent pas.

Solution : Assurez-vous que le service DNS fonctionne correctement. Si vous utilisez un logiciel d'un autre fabricant, assurez-vous qu'il utilise correctement les noms de principaux.

`Server refused to negotiate authentication, which is required for encryption. Good bye.`

Origine : L'application distante n'est pas capable ou a été configuré de manière à ne pas accepter l'authentification Kerberos du client.

Solution : Fournissez une application distante qui peut négocier l'authentification ou configurez l'application pour qu'elle utilise les indicateurs appropriés pour activer l'authentification.

`Server rejected authentication (during sendauth exchange)`

Origine : Le serveur avec lequel vous tentez de communiquer a rejeté l'authentification. La plupart du temps, cette erreur se produit pendant la propagation de base de données kerberos. Certaines causes courantes peuvent être des problèmes avec le fichier `kpropd.acl`, DNS ou le fichier keytab.

Solution : Si vous recevez ce message d'erreur lorsque vous exécutez des applications autres que `kprop`, cherchez à savoir si le fichier keytab du serveur est correct.

`Target name principal '`*principal*`' does not match` *service-principal*

Origine : Le principal de service en cours d'utilisation ne correspond pas au principal de service utilisé par le serveur d'application.

Solution : Sur le serveur d'application, veillez à ce que le principal de service soit inclus dans le fichier keytab. Pour le client, assurez-vous que le principal de service correct est utilisé.

```
The ticket isn't for us
Ticket/authenticator do not match.
```

Origine : Le nom du principal de la demande peut ne pas correspondre au nom du principal de service. Soit le ticket a été envoyé avec un nom FQDN du principal alors que le service attendait un autre nom, soit le service attendait un FQDN et a reçu un autre nom.

Solution : Si vous recevez ce message d'erreur lorsque vous exécutez des applications autres que `kprop`, cherchez à savoir si le fichier keytab du serveur est correct.

```
Truncated input file detected
```

Origine : Le fichier de vidage de base de données qui a été utilisé dans l'opération n'est pas un fichier de vidage complet.

Solution : Recréez le fichier de vidage ou utilisez-en un autre.

Dépannage de Kerberos

Cette section fournit des informations de dépannage pour le logiciel Kerberos.

Problèmes de numéros de version de clé

Parfois, le numéro de version de clé (KVNO) utilisé par le KDC et les clés du principal de service stockées dans le fichier `/etc/krb5/krb5.keytab` pour les services hébergés sur le système ne correspondent pas. Le KVNO peut être désynchronisé lors de la création d'un nouvel ensemble de clés sur le KDC sans mettre à jour le fichier keytab avec les nouvelles clés. Une fois le problème identifié, actualisez le fichier `krb5.keytab`.

1. Répertoriez les entrées keytab .

   Le KVNO pour chaque principal est le premier élément de chaque entrée.

```
# klist -k
Keytab name: FILE:/etc/krb5/krb5.keytab
KVNO Principal
---- --------------------------------------------------------------------------
2 host/denver.example.com@EXAMPLE.COM
2 host/denver.example.com@EXAMPLE.COM
2 host/denver.example.com@EXAMPLE.COM
2 nfs/denver.example.com@EXAMPLE.COM
2 nfs/denver.example.com@EXAMPLE.COM
2 nfs/denver.example.com@EXAMPLE.COM
2 nfs/denver.example.com@EXAMPLE.COM
```

2. Faites l'acquisition d'autorisations initiales à l'aide de la clé `host`.

```
# kinit -k
```

3. Déterminez le KVNO utilisé par le KDC.

```
# kvno nfs/denver.example.com
nfs/denver.example.com@EXAMPLE.COM: kvno = 3
```

Notez que le KVNO répertorié ici est 3 au lieu de 2.

Problèmes avec le format du fichier `krb5.conf`

Si le fichier `krb5.conf` n'est pas formaté correctement, le message d'erreur suivant peut s'afficher dans une fenêtre de terminal ou être enregistrée dans le fichier journal :

```
Improper format of Kerberos configuration file while initializing krb5 library
```

Si le format est incorrect, les services associés sont vulnérables aux attaques. Vous devez résoudre le problème avant d'autoriser l'utilisation des fonctions de Kerberos.

Problèmes de propagation de la base de données Kerberos

Si la propagation de la base de données Kerberos échoue, essayez `/usr/bin/rlogin -x` entre le KDC esclave et le KDC maître, et depuis le serveur KDC maître vers le serveur KDC esclave.

Si les KDC sont sécurisés par défaut, la commande `rlogin` est désactivée et ne peut pas être utilisée pour résoudre ce problème. Pour activer `rlogin` sur un KDC, vous devez activer le service `eklogin`.

```
# svcadm enable svc:/network/login:eklogin
```

Une fois le problème résolu, désactivez le service `eklogin`.

```
# svcadm disable svc:/network/login:eklogin
```

Si l'accès à distance ne fonctionne pas, les problèmes proviennent probablement des fichiers keytab des KDC. Si l'accès à distance ne fonctionne pas, le problème ne provient pas du fichier keytab ou du service de noms, étant donné que `rlogin` et le logiciel de propagation utilisent le même principal `host/`*host-name*. Dans ce cas, assurez-vous que le fichier `kpropd.acl` est correct.

Problèmes de montage d'un système de fichiers NFS utilisant Kerberos

- Si un montage de système de fichiers NFS utilisant Kerberos échoue, assurez-vous que le fichier `/var/rcache/root` existe sur le serveur NFS. Si le système de fichiers n'est pas détenu par `root`, supprimez-le et essayez de le monter une nouvelle fois.
- Si vous avez un problème d'accès à un système de fichiers NFS utilisant Kerberos, assurez-vous que le service `gssd` est activé sur votre système et le serveur NFS.
- Si vous voyez le message d'erreur `invalid argument` ou `bad directory` lorsque vous tentez d'accéder à un système de fichiers NFS utilisant Kerberos, le problème provient peut-être du fait que vous n'utilisez pas un nom DNS complet lorsque vous essayez de monter le système de fichiers NFS. L'hôte qui est en cours de montage n'est pas le même que le composant du nom de l'hôte du principal de service dans le fichier keytab du serveur.

  Ce problème peut également se produire si votre serveur dispose de plusieurs interfaces Ethernet, et que vous avez configuré DNS pour qu'il utilise un plan de type "nom par interface" au lieu de "plusieurs enregistrements d'adresses par hôte". Pour le service Kerberos, vous devez configurer plusieurs enregistrements d'adresses par hôte comme suit[1]:

  ```
  my.host.name.   A       1.2.3.4
  A       1.2.4.4
  A       1.2.5.4

  my-en0.host.name.       A       1.2.3.4
  my-en1.host.name.       A       1.2.4.4
  my-en2.host.name.       A       1.2.5.4

  4.3.2.1         PTR     my.host.name.
  4.4.2.1         PTR     my.host.name.
  4.5.2.1         PTR     my.host.name.
  ```

Dans cet exemple, la configuration autorise une référence pour les différentes interfaces et un seul principal de service au lieu de trois principaux de service dans le fichier keytab du serveur.

[1]Ken Hornstein, "FAQ Kerberos," [http://www.cmf.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html#kerbdns], consulté le 10 mars 2010.

Problèmes liés à l'authentification en tant qu'utilisateur `root`

En cas d'échec de l'authentification lorsque vous tentez de devenir `root` sur votre système et que vous avez déjà ajouté le principal `root` au fichier keytab de votre hôte, vérifiez deux zones. Tout d'abord, assurez-vous que le principal `root` dans le fichier keytab a un nom d'hôte complet comme instance. Si c'est le cas, vérifiez le fichier `/etc/resolv.conf` pour vous assurer que le système est correctement configuré en tant que client DNS.

Observation du mappage d'informations d'identification GSS sur des informations d'identification UNIX

Pour être en mesure de contrôler les correspondances d'informations d'identification, commencez par décommenter cette ligne du fichier `/etc/gss/gsscred.conf`.

```
SYSLOG_UID_MAPPING=yes
```

Ensuite, le service `gssd` doit lire le fichier `/etc/gss/gsscred.conf`.

```
# pkill -HUP gssd
```

A présent, vous pouvez contrôler les mappages d'informations d'identification comme `gssd` le demande. Les mappages sont enregistrés par le démon `syslog`, si le fichier `syslog.conf` est configuré pour l'utilitaire système `auth` avec le niveau de gravité - `debug`.

Remarque - Si l'instance de service `rsyslog` est activée, les mappages sont enregistrés par le démon `rsyslog`.

Utilisation de DTrace avec le service Kerberos

Le mécanisme Kerberos prend en charge plusieurs sondes DTrace pour le décodage de différents messages de protocole. Pour obtenir une liste, reportez-vous à la section Annexe A, Sondes DTrace pour Kerberos. Avoir un avantage de sondes DTrace rapport aux autres inspecteurs de protocole, par un utilisateur doté de car pour DTrace permet de consulter facilement les données d'application et Kerberos non chiffrées.

Les exemples suivants indiquent ce qui peut être consulté avec Kerberos sondes DTrace.

EXEMPLE 8-1 Utilisation de DTrace pour le suivi des messages Kerberos

Le script suivant utilise les sondes DTrace pour afficher des détails sur les messages Kerberos qui sont envoyées et reçues par le système. Notez que les champs affichés sont basés sur les structures affectées aux sondes `krb_message-recv` et`krb_message-send`. Pour plus d'informations, reportez-vous à la section “Définitions des sondes Kerberos DTrace” à la page 222.

```
kerberos$target:::krb_message-recv
{
    printf("<- krb message recved: %s\n", args[0]->krb_message_type);
    printf("<- krb message remote addr: %s\n", args[1]->kconn_remote);
    printf("<- krb message ports: local %d remote %d\n",
            args[1]->kconn_localport, args[1]->kconn_remoteport);
    printf("<- krb message protocol: %s transport: %s\n",
            args[1]->kconn_protocol, args[1]->kconn_type);
}

kerberos$target:::krb_message-send
{
    printf("-> krb message sent: %s\n", args[0]->krb_message_type);
    printf("-> krb message remote addr: %s\n", args[1]->kconn_remote);
    printf("-> krb message ports: local %d remote %d\n",
            args[1]->kconn_localport, args[1]->kconn_remoteport);
    printf("-> krb message protocol: %s transport: %s\n",
            args[1]->kconn_protocol, args[1]->kconn_type);
    printf("\n");
}

kerberos$target:::krb_error-read
{
    printf("<- krb error code: %s\n", args[1]->kerror_error_code);
    printf("<- krb error client: %s server: %s\n", args[1]->kerror_client,
            args[1]->kerror_server);
    printf("<- krb error e-text: %s\n", args[1]->kerror_e_text);
    printf("\n");
}
```

Le script précédent peut être appelé à partir de la ligne de commande ou à l'aide du démon `krb5kdc`. Voici un exemple de l'appel du script, qui est appelé `krb-dtrace.d`, à partir de la ligne de commande. La commande entraîne Kerberos pour envoyer et recevoir des messages. Notez que `LD_NOLAZYLOAD=1` est nécessaire pour forcer le chargement de la bibliothèque Kerberos `mech_krb5.so` qui contient les sondes DTrace de Kerberos.

```
# LD_NOLAZYLOAD=1 dtrace -s ./krb\-dtrace.d -c kinit
dtrace: script './krb-dtrace' matched 4 probes
kinit: Client 'root@DEV.ORACLE.COM' not found in Kerberos database while g
etting initial credentials
dtrace: pid 3750 has exited
 CPU     ID                    FUNCTION:NAME
   2  74782 k5_trace_message_send:krb_message-send -> krb message sent: KR
B_AS_REQ(10)
-> krb message remote addr: 10.229.168.163
```

```
-> krb message ports: local 62029 remote 88
-> krb message protocol: ipv4 transport: udp


   2  74781 k5_trace_message_recv:krb_message-recv <- krb message recved:
KRB_ERROR(30)
<- krb message remote addr: 10.229.168.163
<- krb message ports: local 62029 remote 88
<- krb message protocol: ipv4 transport: udp

   2  74776     krb5_rd_error:krb_error-read <- krb error code: KDC_ERR_C_
PRINCIPAL_UNKNOWN(6)
<- krb error client: root@DEV.ORACLE.COM server: krbtgt/DEV.ORACLE.COM@DEV
.ORACLE.COM
<- krb error e-text: CLIENT_NOT_FOUND
```

Pour utiliser le script à l'aide du démon `krb5kdc`, le service `svc:/network/security/krb5kdc:default` doit être activé et en ligne. Notez que la commande suivante n'utilise pas `LD_NOLAZYLOAD=1` étant donné que la bibliothèque `mech_krb5.so` charge le démon `krb5kdc`.

```
# dtrace -s ./krb\-dtrace.d -p $(pgrep -x krb5kdc)
```

EXEMPLE 8-2 Utilisation de DTrace pour visualiser les types de pré-authentification Kerberos

L'exemple ci-dessous illustre quelle pré-authentification est sélectionnée par le client. La première étape consiste à créer un script DTrace similaire à l'exemple ci-dessous :

```
cat krbtrace.d
kerberos$target:::krb_message-recv
{
printf("<- krb message recved: %s\n", args[0]->krb_message_type);
printf("<- krb message remote addr: %s\n", args[1]->kconn_remote);
printf("<- krb message ports: local %d remote %d\n",
args[1]->kconn_localport, args[1]->kconn_remoteport);
printf("<- krb message protocol: %s transport: %s\n",
args[1]->kconn_protocol, args[1]->kconn_type);
}

kerberos$target:::krb_message-send
{
printf("-> krb message sent: %s\n", args[0]->krb_message_type);
printf("-> krb message remote addr: %s\n", args[1]->kconn_remote);
printf("-> krb message ports: local %d remote %d\n",
args[1]->kconn_localport, args[1]->kconn_remoteport);
printf("-> krb message protocol: %s transport: %s\n",
args[1]->kconn_protocol, args[1]->kconn_type);
printf("\n");
}

kerberos$target:::krb_kdc_req-make
{
printf("-> krb kdc_req make msg type: %s\n", args[0]->krb_message_type);
printf("-> krb kdc_req make pre-auths: %s\n", args[1]->kdcreq_padata_types);
```

```
printf("-> krb kdc_req make auth data: %s\n", args[1]->kdcreq_authorization_data);
printf("-> krb kdc_req make client: %s server: %s\n", args[1]->kdcreq_client,
args[1]->kdcreq_server );
}

kerberos$target:::krb_kdc_req-read
{
/* printf("<- krb kdc_req msg type: %s\n", args[0]->krb_message_type); */
printf("<- krb kdc_req client: %s server: %s\n", args[1]->kdcreq_client,
args[1]->kdcreq_server );
printf("\n");
}

kerberos$target:::krb_kdc_rep-read
{
/* printf("<- krb kdc_rep msg type: %s\n", args[0]->krb_message_type); */
printf("<- krb kdc_rep client: %s server: %s\n", args[1]->kdcrep_client,
args[1]->kdcrep_enc_server );
printf("\n");
}

kerberos$target:::krb_ap_req-make
{
printf("-> krb ap_req make server: %s client: %s\n", args[2]->kticket_server,
args[2]->kticket_enc_client );
}

kerberos$target:::krb_error-read
{
printf("<- krb error code: %s\n", args[1]->kerror_error_code);
printf("<- krb error client: %s server: %s\n", args[1]->kerror_client,
args[1]->kerror_server);
printf("<- krb error e-text: %s\n", args[1]->kerror_e_text);
printf("\n");
}
```

Ensuite, exécutez le script `krbtrace.d` en tant qu'utilisateur disposant de privilèges sur le système Kerberos en tapant la commande suivante :

```
# LD_BIND_NOW=1 dtrace -qs krbtrace.d -c "kinit -k"
.
.
-> krb kdc_req make pre-auths: FX_COOKIE(133) ENC_TIMESTAMP(2) REQ_ENC_PA_REP(149)
```

Les types de pré-authentification sont affichés dans la sortie. Pour plus d'informations sur les divers types de pré-authentification, consultez la page RFC 4120.

EXEMPLE 8-3 Utilisation de DTrace pour vider un message d'erreur Kerberos

```
# dtrace -n 'krb_error-make {
printf("\n{");
printf("\n\tctime = %Y", (uint64_t)(args[1]->kerror_ctime * 1000000000));
printf("\n\tcusec = %d", args[1]->kerror_cusec);
```

```
printf("\n\tstime = %Y", (uint64_t)(args[1]->kerror_stime * 1000000000));
printf("\n\tsusec = %d", args[1]->kerror_susec);
printf("\n\terror_code = %s", args[1]->kerror_error_code);
printf("\n\tclient = %s", args[1]->kerror_client);
printf("\n\tserver = %s", args[1]->kerror_server);
printf("\n\te_text = %s", args[1]->kerror_e_text);
printf("\n\te_data = %s", "");
printf("\n}");
}'
dtrace: description 'krb_error-make ' matched 1 probe
CPU     ID                    FUNCTION:NAME
0  78307     krb5_mk_error:krb_error-make
{
ctime = 2012 May 10 12:10:20
cusec = 0
stime = 2012 May 10 12:10:20
susec = 319090
error_code = KDC_ERR_C_PRINCIPAL_UNKNOWN(6)
client = testuser@EXAMPLE.COM
server = krbtgt/EXAMPLE.COM@EXAMPLE.COM
e_text = CLIENT_NOT_FOUND
e_data =
}
```

EXEMPLE 8-4 Utilisation de DTrace pour afficher le ticket de service d'un serveur SSH

```
# LD_PRELOAD_32=/usr/lib/gss/mech_krb5.so.1 dtrace -q -n '
kerberos$target:::krb_kdc_req-make {
printf("kdcreq_server: %s",args[1]->kdcreq_server);
}' -c "ssh local@four.example.com" -o dtrace.out
Last login: Wed Sep 10 10:10:20 2014
Oracle Solaris 11 X86    July 2014
$ ^D
# cat dtrace.out
kdcreq_server: host/four.example.com@EXAMPLE.COM
```

EXEMPLE 8-5 Utilisation de DTrace pour afficher l'adresse et le port d'un KDC non disponible lors d'une demande de TGT initial

```
# LD_BIND_NOW=1 dtrace -q -n '
kerberos$target:::krb_message-send {
printf("%s:%d\n",args[1]->kconn_remote, args[1]->kconn_remoteport)
}' -c "kinit local4"

10.10.10.14:88
10.10.10.14:750
10.10.10.14:88
10.10.10.14:750
10.10.10.14:88
10.10.10.14:750
kinit(v5): Cannot contact any KDC for realm 'EXAMPLE.COM'
```

```
while getting initial credentials
```

EXEMPLE 8-6 Utilisation de DTrace pour afficher les demandes des principaux Kerberos

```
# LD_BIND_NOW=1 dtrace -qs /opt/kdebug/mykdtrace.d \
-c 'kadmin -p kdc/admin -w test123 -q listprincs'
Authenticating as principal kdc/admin with password.
krb kdc_req msg type: KRB_AS_REQ(10)
krb kdc_req make client: kdc/admin@TEST.NET server:
kadmin/interop1.example.com@TEST.NET
krb message sent: KRB_AS_REQ(10)
krb message recved: KRB_ERROR(30)
Err code: KDC_ERR_PREAUTH_REQUIRED(25)
Err msg client: kdc/admin@TEST.NET server: kadmin/interop1.example.com@TEST.NET
Err e-text: NEEDED_PREAUTH
krb kdc_req msg type: KRB_AS_REQ(10)
krb kdc_req make client: kdc/admin@TEST.NET server:
kadmin/interop1.example.com@TEST.NET
krb message sent: KRB_AS_REQ(10)
krb message recved: KRB_AS_REP(11)
kadmin: Database error! Required KADM5 principal missing while
initializing kadmin interface
krb kdc_req msg type: KRB_AS_REQ(10)
krb kdc_req make client: kdc/admin@TEST.NET server:
kadmin/interop2.example.com@TEST.NET
krb message sent: KRB_AS_REQ(10)
krb message recved: KRB_ERROR(30)
Err code: KDC_ERR_S_PRINCIPAL_UNKNOWN(7)
Err msg client: kdc/admin@TEST.NET server: kadmin/interop2.example.com@TEST.NET
Err e-text: SERVER_NOT_FOUND
krb kdc_req msg type: KRB_AS_REQ(10)
krb kdc_req make client: kdc/admin@TEST.NET server:
kadmin/interop2.example.com@TEST.NET
```

Le script suivant a été utilisée pour produire la sortie ci-dessus .

```
kerberos$target:::krb_message-recv
{
printf("krb message recved: %s\n", args[0]->krb_message_type);
}

kerberos$target:::krb_message-send
{
printf("krb message sent: %s\n", args[0]->krb_message_type);
}

kerberos$target:::krb_kdc_req-make
{
printf("krb kdc_req msg type: %s\n", args[0]->krb_message_type);
printf("krb kdc_req make client: %s server: %s\n", args[1]->kdcreq_client,
args[1]->kdcreq_server );
}
```

```
kerberos$target:::krb_ap_req-make
{
printf("krb ap_req make server: %s client: %s\n", args[2]->kticket_server,
args[2]->kticket_enc_client );
}

kerberos$target:::krb_error-read
{
printf("Err code: %s\n", args[1]->kerror_error_code);
printf("Err msg client: %s server: %s\n", args[1]->kerror_client,
args[1]->kerror_server);
printf("Err e-text: %s\n", args[1]->kerror_e_text);
}
```

♦ ♦ ♦ CHAPITRE 9

Utilisation de l'authentification simple et de la couche de sécurité

Ce chapitre contient des informations sur SASL (Simple Authentication and Security Layer, couche d'authentification et de sécurité simple).

A propos de SASL

La couche SASL (Simple Authentication and Security Layer) est une structure fournissant des services d'authentification et de sécurité facultatifs aux protocoles réseau. Une application appelle la bibliothèque SASL, `/usr/lib/libsasl.so`, qui fournit une couche de collage entre l'application et les divers mécanismes SASL. Les mécanismes sont utilisés dans le processus d'authentification et lors de la fourniture de services de sécurité facultatifs. La version de SASL est dérivée de la couche Cyrus SASL avec quelques changements.

SASL fournit les services suivants :

- Chargement de plug-ins
- Détermination des options de sécurité nécessaires dans l'application afin de faciliter le choix d'un mécanisme de sécurité
- Liste des plug-ins disponibles pour l'application
- Choix du meilleur mécanisme dans une liste des mécanismes disponibles pour une tentative d'authentification particulière
- Routage des données d'authentification entre l'application et le mécanisme sélectionné
- Renvoi des informations sur la négociation SASL à l'application

SASL (référence)

La section suivante fournit des informations sur l'implémentation de SASL.

Plug-ins SASL

Les plug-ins SASL assurent la prise en charge des mécanismes de sécurité, la normalisation utilisateur et la récupération des propriétés auxiliaires. Par défaut, les plug-ins 32 bits chargés de manière dynamique sont installés dans `/usr/lib/sasl` et les plug-ins 64 bits sont installés dans `/usr/lib/sasl/` *$ISA*. Les plug-ins de mécanisme de sécurité suivants sont fournis :

`crammd5.so.1`	CRAM-MD5, qui prend en charge l'authentification uniquement, et non l'autorisation.
`digestmd5.so.1`	DIGEST-MD5, qui prend en charge l'authentification, l'intégrité et la confidentialité, ainsi que l'autorisation.
`gssapi.so.1`	GSSAPI, qui prend en charge l'authentification, l'intégrité et la confidentialité, ainsi que l'autorisation. Le mécanisme de sécurité GSSAPI requiert une infrastructure Kerberos en état de fonctionnement.
`plain.so.1`	PLAIN, qui prend en charge l'authentification et l'autorisation.

En outre, le plug-in de mécanisme de sécurité EXTERNE et les plug-ins de normalisation utilisateur INTERNE sont intégrés dans `libsasl.so.1`. Le mécanisme EXTERNE prend en charge l'authentification et l'autorisation. Le mécanisme prend en charge l'intégrité et la confidentialité si la source de sécurité externe les fournit. Le plug-in INTERNE ajoute le nom de domaine au nom d'utilisateur, le cas échéant.

La version Oracle Solaris ne fournit aucun plug-ins `auxprop` actuellement. Pour que les plug-ins de mécanisme CRAM-MD5 et DIGEST-MD5 soient entièrement opérationnels côté serveur, l'utilisateur doit fournir un plug-in `auxprop` pour récupérer des mots de passe en clair. Le plug-in PLAIN requiert une prise en charge supplémentaire pour vérifier le mot de passe. La prise en charge de la vérification du mot de passe peut se faire à travers l'un des éléments suivants : un rappel à l'application de serveur, un plug-in `auxprop`, `saslauthd` ou `pwcheck`. Les démons `salauthd` et `pwcheck` ne sont pas fournis dans les versions Oracle Solaris. Pour une meilleure interopérabilité, limitez les applications de serveur aux mécanismes qui sont entièrement opérationnel en utilisant l'option SASL `mech_list`.

Variable d'environnement SASL

Par défaut, le nom d'authentification du client est défini sur `getenv("LOGNAME")`. Cette variable peut être réinitialisée par le client ou par le plug-in.

Options SASL

Le comportement de `libsasl` et les plug-ins peuvent être modifiés côté serveur à l'aide d'options pouvant être définies dans le fichier `/etc/sasl/`*app*`.conf`. La variable *app* est le nom défini côté serveur pour l'application. La documentation du serveur *app* doit indiquer le nom de l'application.

Les options suivantes sont prises en charge :

`auto_transition`	Effectue la transition automatique de l'utilisateur vers d'autres mécanismes lorsque celui-ci réalise une authentification en texte simple réussie.
`auxprop_login`	Dresse la liste des noms de plug-ins de propriétés auxiliaires à utiliser.
`canon_user_plugin`	Sélectionne le plug-in canon_user à utiliser.
`mech_list`	Dresse la liste des mécanismes autorisés à être utilisés par l'application de serveur.
`pwcheck_method`	Dresse la liste des mécanismes utilisés pour vérifier les mots de passe. Actuellement, `auxprop` est la seule valeur autorisée.
`reauth_timeout`	Définit la durée, en minutes, pendant laquelle les informations d'authentification sont mises en mémoire cache pour une réauthentification rapide. Cette option est utilisée par le plug-in DIGEST-MD5. La définition de cette option sur la valeur 0 désactive la réauthentification.

Les options suivantes ne sont pas prises en charge :

`plugin_list`	Dresse la liste des mécanismes disponibles. Non utilisé, car l'option modifie le comportement de chargement dynamique des plug-ins.
`saslauthd_path`	Définit l'emplacement de la porte saslauthd, qui est utilisée pour la communication avec le démon `saslauthd`. Le démon `saslauthd` n'est pas inclus dans la version Oracle Solaris. Par conséquent, cette option n'est pas incluse non plus.
`keytab`	Définit l'emplacement du fichier `keytab` utilisé par le plug-in GSSAPI. Utilisez à la place la variable d'environnement `KRB5_KTNAME` pour définir l'emplacement `keytab` par défaut.

Les options suivantes ne figurent pas dans Cyrus SASL. Cependant, elles ont été ajoutées dans la version Oracle Solaris :

`use_authid`	Permet d'obtenir les informations d'identification du client plutôt que d'utiliser les informations d'identification par défaut lors de la création du contexte de sécurité client GSS. Par défaut, l'identité Kerberos du client est utilisée.
`log_level`	Définit le niveau souhaité de journalisation d'un serveur.

♦ ♦ ♦ CHAPITRE 10

Configuration des services réseau d'authentification

Ce chapitre fournit des informations sur la façon d'utiliser le RPC sécurisé pour authentifier un hôte et un utilisateur sur un montage NFS, et traite des sujets suivants :

A propos de RPC sécurisé

Sécurisé (Remote Procedure Call-Appel de procédure asynchrone) RPC sécurisé protège les procédures distantes par le biais d'un mécanisme d'authentification. Le mécanisme d'authentification Diffie-Hellman authentifie à la fois l'hôte et l'utilisateur à l'origine d'une demande de service. Le mécanisme d'authentification utilise le chiffrement Data Encryption Standard (DES). Les applications qui utilisent le RPC sécurisé incluent le service de noms NFS et NIS.

Services NFS et RPC sécurisé

NFS permet à plusieurs hôtes de partager des fichiers sur le réseau. Dans le cadre du service NFS, un serveur contient les données et les ressources pour plusieurs clients. Les clients ont accès aux systèmes de fichiers que le serveur partage avec les clients. Les utilisateurs connectés aux systèmes client peuvent accéder aux systèmes de fichiers en montant les systèmes de fichiers à partir du serveur. Pour l'utilisateur sur le système client, il s'affiche comme si les fichiers étaient des fichiers locaux pour le client. L'une des utilisations les plus courantes de NFS permet aux systèmes d'être installés dans les bureaux, tout en stockant tous les fichiers utilisateur dans un emplacement central. Certaines fonctions du service NFS, telles que l'option `-nosuid` de la commande `mount` peuvent être utilisées pour interdire l'ouverture des périphériques et systèmes de fichiers par des utilisateurs non autorisés.

Le service NFS utilise le RPC sécurisé afin d'authentifier les utilisateurs adressant des demandes sur le réseau. Ce processus est appelé *NFS sécurisé*. Le mécanisme d'authentification

Diffie-Hellman `AUTH_DH` utilise des fonctions de chiffrement DES pour garantir un accès autorisé. Le mécanisme `AUTH_DH` a également été appelé `AUTH_DES`. Pour plus d'informations, reportez-vous aux références suivantes :

- Pour configurer et administrer le service NFS sécurisé, reportez-vous à la section " Administration du système NFS sécurisé " du manuel " Gestion des systèmes de fichiers réseau dans Oracle Solaris 11.2 ".

Authentification Kerberos

Kerberos est un système d'authentification qui a été développé au MIT. Une mise en oeuvre côté client et côté serveur de Kerberos V5, qui utilise RPCSEC_GSS, est incluse dans cette version. Pour plus d'informations, reportez-vous à la section "Configuration des serveurs NFS Kerberos" à la page 122.

Chiffrement DES avec NFS sécurisé

Les fonctions de chiffrement Data Encryption Standard (DES) utilisent une clé 56 bits pour chiffrer les données. Si deux utilisateurs identifiés ou principaux disposent de la même clé DES, ils peuvent communiquer en privé à l'aide de la clé pour chiffrer et déchiffrer du texte. DES est un mécanisme de chiffrement relativement rapide.

Le risque lié à l'utilisation de la clé DES uniquement est qu'un intrus puisse recueillir suffisamment de messages texte chiffrés avec la même clé pour être en mesure de découvrir la clé et déchiffrer les messages. C'est pour cette raison que les systèmes de sécurité tels que NFS sécurisé doivent changer de clés fréquemment.

Authentification Diffie-Hellman et RPC sécurisé

La méthode Diffie-Hellman (DH) d'authentification des utilisateurs se révèle très difficile à déchiffrer pour un intrus. Le client et le serveur disposent chacun de leur clé privée, qu'ils utilisent avec la clé publique pour créer une clé commune. La clé privée est également appelée *clé secrète*. Le client et le serveur utilisent la clé commune pour communiquer l'un avec l'autre. La clé commune est chiffrée à l'aide d'une fonction de chiffrement convenue, comme par exemple la méthode DES.

L'authentification est basée sur la capacité du système émetteur à utiliser la clé commune pour chiffrer l'heure actuelle. Le système récepteur peut ensuite la déchiffrer et la vérifier par rapport à son heure actuelle. L'heure du client et l'heure du serveur doivent être synchronisées. Le NTP

(Network Time Protocol) peut être utilisée pour synchroniser les horloges. Le logiciel NTP du domaine public de l'Université du Delaware est inclus dans le logiciel Oracle Solaris. La documentation est disponible sur le site Web NTP Documentation.

Les clés publiques et les clés privées sont conservées dans une base de données NIS. NIS stocke les clés dans la carte `publickey`. Ce fichier contient la clé publique et la clé privée pour tous les utilisateurs potentiels.

L'administrateur système est responsable du paramétrage des cartes NIS, et de la génération d'une clé publique et d'une clé privée pour chaque utilisateur. La clé privée est stockée sous forme chiffrée avec le mot de passe de l'utilisateur. Du fait de ce processus, la clé privée n'est connue que de l'utilisateur.

Administration de l'authentification avec RPC sécurisé

En demandant l'authentification pour l'utilisation des systèmes de fichiers monté NFS, vous augmentez la sécurité de votre réseau.

La liste des tâches suivante indique les procédures de configuration du RPC sécurisé pour NIS et NFS.

TABLEAU 10-1 Administration de l'authentification avec RPC sécurisé

Tâche	Description	Pour obtenir des instructions
1. Démarrage du serveur de clés.	Permet de s'assurer que des clés peuvent être créées, de sorte que les utilisateurs peuvent être authentifiés.	“Redémarrage du serveur de clé RPC sécurisé” à la page 215
2. Définition des informations d'identification sur un hôte NIS.	Permet de s'assurer que l'utilisateur `root` sur un hôte peut être authentifié dans un environnement NIS.	“Configuration d'une clé Diffie-Hellman pour un hôte NIS” à la page 216
3. Attribution d'une clé à un utilisateur NIS.	Permet à un utilisateur d'être authentifié dans un environnement NIS.	“Configuration d'une clé Diffie-Hellman pour un utilisateur NIS” à la page 217
4. Partage de fichiers NFS avec authentification.	Permet à un serveur NFS de protéger en toute sécurité des systèmes de fichiers partagés à l'aide de l'authentification.	“Partage de fichiers NFS avec l'authentification Diffie-Hellman” à la page 218

▼ Redémarrage du serveur de clé RPC sécurisé

Avant de commencer

Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. **Vérifiez que le démon `keyserv` est en cours d'exécution.**

```
# svcs \*keyserv\*
STATE    STIME   FMRI
disabled Dec_14  svc:/network/rpc/keyserv
```

2. **Activez le service du serveur de clés si le service n'est pas en ligne.**

```
# svcadm enable network/rpc/keyserv
```

Configuration d'une clé Diffie-Hellman pour un hôte NIS

Suivez cette procédure sur chaque hôte du domaine NIS.

Avant de commencer Vous devez prendre le rôle `root`. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Si le service de noms par défaut n'est pas NIS, ajoutez-y la carte `publickey`.**

   a. **Vérifiez que la valeur de `config/default` pour le service de noms n'est pas `nis`.**

```
# svccfg -s name-service/switch listprop config
config                      application
config/value_authorization  astring       solaris.smf.value.name-service.switch
config/default              astring       files
config/host                 astring       "files nis dns"
config/printer              astring       "user files nis"
```

   Si la valeur de `config/default` est `nis`, vous pouvez arrêter ici.

   b. **Définissez le service de noms pour `publickey` sur `nis`.**

```
# svccfg -s name-service/switch setprop config/publickey = astring: "nis"
# svccfg -s name-service/switch:default refresh
```

   c. **Confirmez la valeur de `publickey`.**

```
# svccfg -s name-service/switch listprop
config                      application
config/value_authorization  astring       solaris.smf.value.name-service.switch
config/default              astring       files
config/host                 astring       "files nis dns"
config/printer              astring       "user files nis"
```

```
config/publickey            astring      nis
```

Sur ce système, la valeur de `publickey` est répertoriée, car elle est différente de la valeur par défaut, `files`.

2. **Créez une nouvelle paire de clés à l'aide de la commande `newkey`.**

```
# newkey -h hostname
```

où *hostname* est le nom du client.

Exemple 10-1 Configuration d'une nouvelle clé pour `root` sur un client NIS

Dans l'exemple suivant, un administrateur avec le profil des droits de sécurité de service des nom paramètre `earth` en tant que client NIS.

```
# newkey -h earth
Adding new key for unix.earth@example.com
New Password: xxxxxxxx
Retype password: xxxxxxxx
Please wait for the database to get updated...
Your new key has been successfully stored away.
#
```

▼ Configuration d'une clé Diffie-Hellman pour un utilisateur NIS

Exécutez cette procédure pour chaque utilisateur du domaine NIS.

Avant de commencer

Vous devez être connecté au serveur maître NIS pour générer une nouvelle clé pour un utilisateur. Le profil des droits de sécurité de service de noms doit vous être attribué. Pour plus d'informations, reportez-vous à la section " A l'aide de vos droits administratifs attribués " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ".

1. **Créez une nouvelle clé pour un utilisateur.**

```
# newkey -u username
```

où *username* correspond au nom de l'utilisateur. Le système vous invite à saisir un mot de passe. Vous pouvez saisir un mot de passe générique. La clé privée est stockée sous forme chiffrée à l'aide du mot de passe générique.

2. **Indiquez à l'utilisateur de se connecter et de saisir la commande `chkey -p`.**

Cette commande permet aux utilisateurs de re-chiffrer leurs clés privées avec un mot de passe uniquement connu de l'utilisateur.

Remarque - La commande `chkey` peut être utilisée pour créer une nouvelle paire de clés pour un utilisateur.

Exemple 10-2 Configuration et chiffrement d'une nouvelle clé utilisateur dans NIS

Dans cet exemple, le superutilisateur définit la clé.

```
# newkey -u jdoe
Adding new key for unix.12345@example.com
New Password: xxxxxxxx
Retype password: xxxxxxxx
Please wait for the database to get updated...
Your new key has been successfully stored away.
#
```

L'utilisateur `jdoe` re-chiffre la clé à l'aide d'un mot de passe privé.

```
% chkey -p
Updating nis publickey database.
Reencrypting key for unix.12345@example.com
Please enter the Secure-RPC password for jdoe: xxxxxxxx
Please enter the login password for jdoe: xxxxxxxx
Sending key change request to centralexample...
```

▼ Partage de fichiers NFS avec l'authentification Diffie-Hellman

Cette procédure protège les systèmes de fichiers partagés sur un serveur NFS en requérant l'authentification pour l'accès.

Avant de commencer L'authentification par clé publique Diffie-Hellman doit être activée sur le réseau. Pour activer l'authentification sur le réseau, effectuez la procédure “Configuration d'une clé Diffie-Hellman pour un hôte NIS” à la page 216.

Vous devez vous connecter en tant qu'administrateur disposant du profil de droits System Management (gestion de système) pour effectuer cette tâche. Pour plus d'informations, reportez-vous à la section “ A l’aide de vos droits administratifs attribués ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

1. **Sur le serveur NFS, partagez un système de fichiers avec l'authentification Diffie-Hellman..**

   ```
   # share -F nfs -o sec=dh /filesystem
   ```

   où *filesystem* est le système de fichiers partagé.

L'option `-o sec=dh` signifie que l'authentification `AUTH_DH` est désormais requise pour accéder au système de fichiers.

2. **Sur un client NFS, montez un système de fichiers avec l'authentification Diffie-Hellman.**

   `# mount -F nfs -o sec=dh` *server:filesystem mount-point*

server	Nom du système qui partage *filesystem*
filesystem	Nom du système de fichiers partagé, tel que `opt`
mount-point	Nom du point de montage, tel que `/opt`

   L'option `-o sec=dh` permet de monter le système de fichiers avec l'authentification `AUTH_DH`.

♦ ♦ ♦ ANNEXE A

Sondes DTrace pour Kerberos

Cette annexe décrit les structures les sondes DTrace et un argument. Pour des exemples de leur utilisation, reportez-vous à la section “Utilisation de DTrace avec le service Kerberos” à la page 201.

Sondes DTrace pour Kerberos

Les *sondes* sont des emplacements ou activités de programme auxquels DTrace peut lier une demande en vue de réaliser un ensemble d'actions. Les sondes sont définies et mises en oeuvre par un *fournisseur*. Les fournisseurs de modules chargeables, qui sont du noyau des sondes afin de permettre à leurs données de trace.

Ces sondes sont affichés qu'à titre USDT définies de manière statique de la fonction de trace (utilisateur). Les sondes USDT Kerberos sont conçus pour l'aide du protocole, reportez-vous à l'utilisateur. Les éléments requis par définies de manière statique aucune fonction de trace du noyau sont fournis.

Vous créez des scripts sondes DTrace enregistrement le cas échéant les informations souhaitées, par exemple, une trace de pile, un horodatage, ou l'évaluation de l'argument d'une fonction. DTrace sondes sont déclenchées, comme les données à partir de la collecte et états qu'il sondes et de vous la retransmettre. Si vous ne spécifiez pas d'actions pour une sonde enregistre chaque fois que l', DTrace et sur les sonde CPU se déclenche.

Les sondes Kerberos DTrace sont modélisées d'après les types de message Kerberos décrits dans RFC4120: The Kerberos Network Authentication Service (V5) (http://www.ietf.org/rfc/rfc4120.txt). Les sondes sont disponibles aux destinataires de `libkrb5/mech_krb5`, notamment ces applications qui utilisent `mech_krb5` via `libgss`. Message les sondes sont et de réception de messages divisée en deux parties, et l'envoi de la création et la réception. Pour plus d'informations concernant `libgss`, reportez-vous à la page de manuel `libgss`(3LIB).

Pour utiliser les sondes, vous indiquez le fournisseur `kerberos`, le nom de la sonde (par exemple `krb_message-recv`) ainsi que les arguments. Pour obtenir des exemples, reportez-vous à la section“Utilisation de DTrace avec le service Kerberos” à la page 201.

Définitions des sondes Kerberos DTrace

Sondes pour `KRB_AP_REP` :

```
kerberos$pid:::krb_ap_rep-make
kerberos$pid:::krb_ap_rep-read

args[0]        krbinfo_t *
args[1]        kaprepinfo_t *
```

Sondes pour `KRB_AP_REQ` :

```
kerberos$pid:::krb_ap_req-make
kerberos$pid:::krb_ap_req-read

args[0]        krbinfo_t *
args[1]        kapreqinfo_t *
args[2]        kticketinfo_t *
args[3]        kauthenticatorinfo_t *
```

Sondes pour `KRB_KDC_REP` :

```
kerberos$pid:::krb_kdc_rep-make
kerberos$pid:::krb_kdc_rep-read

args[0]        krbinfo_t *
args[1]        kdcrepinfo_t *
args[2]        kticketinfo_t *
```

Sondes pour `KRB_KDC_REQ` :

```
kerberos$pid:::krb_kdc_req-make
kerberos$pid:::krb_kdc_req-read

args[0]        krbinfo_t *
args[1]        kdcreqinfo_t *
```

Sondes pour `KRB_CRED` :

```
kerberos$pid:::krb_cred-make
kerberos$pid:::krb_cred-read

args[0]        krbinfo_t *
args[1]        kcredinfo_t *
```

Sondes pour `KRB_ERROR` :

```
kerberos$pid:::krb_error-make
kerberos$pid:::krb_error-read

args[0]        krbinfo_t *
args[1]        kerrorinfo_t *
```

Sondes pour `KRB_PRIV` :

```
kerberos$pid:::krb_priv-make
kerberos$pid:::krb_priv-read

args[0]        krbinfo_t *
args[1]        kprivinfo_t *
```

Sondes pour `KRB_SAFE` :

```
kerberos$pid:::krb_safe-make
kerberos$pid:::krb_safe-read

args[0]        krbinfo_t *
args[1]        ksafeinfo_t *
```

Les éléments requis par l'envoi et la réception des messages

```
kerberos$pid:::krb_message-recv
kerberos$pid:::krb_message-send

args[0]        krbinfo_t *
args[1]        kconninfo_t *
```

Structures d'arguments dans Kerberos

Dans certaines situations, les valeurs de certaines arguments peuvent être défini sur `0` ou peuvent être vides. Les structures d'argument Kerberos sont conçus pour être, en général, cohérents avec RFC4120: The Kerberos Network Authentication Service (V5) (http://www.ietf.org/rfc/rfc4120.txt).

Informations des message Kerberos dans DTrace

```
typedef struct krbinfo {
uint8_t krb_version;                /* protocol version number (5) */
string krb_message_type;            /* Message type (AS_REQ(10), ...) */
uint64_t krb_message_id;            /* message identifier */
uint32_t krb_message_length;        /* message length */
uintptr_t krb_message;              /* raw ASN.1 encoded message */
} krbinfo_t;
```

Remarque - Le protocole ne comporte pas le message Kerberos identificateur. L'identifiant `krb_message_id` est spécifique au fournisseur Kerberos et est conçu pour lier des messages entre les sondes `make/read` et `send/recv`.

Informations de connexion Kerberos dans DTrace

```
typedef struct kconninfo {
string kconn_remote;                /* remote host address */
string kconn_local;                 /* local host address */
uint16_t kconn_localport;           /* local port */
uint16_t kconn_remoteport;          /* remote port */
string kconn_protocol;              /* protocol (ipv4, ipv6) */
string kconn_type;                  /* transport type (udp, tcp) */
} kconninfo_t;
```

Fournisseur d'authentification des informations sur DTrace Kerberos

```
typedef struct kauthenticatorinfo {
string kauth_client;                /* client principal identifier */
string kauth_cksum_type;            /* type of checksum (des-cbc, ...) */
uint32_t kauth_cksum_length;        /* length of checksum */
uintptr_t kauth_cksum_value;        /* raw checksum data */
uint32_t kauth_cusec;               /* client time, microseconds */
uint32_t kauth_ctime;               /* client time in seconds */
string kauth_subkey_type;           /* sub-key type (des3-cbc-sha1, ...) */
uint32_t kauth_subkey_length;       /* sub-key length */
uintptr_t kauth_subkey_value;       /* sub-key data */
uint32_t kauth_seq_number;          /* sequence number */
string kauth_authorization_data;    /* top-level authorization types
(AD-IF-RELEVANT, ... ) */
} kauthenticatorinfo_t;

typedef struct kticketinfo_t {
string kticket_server;              /* service principal identifier */
uint32_t kticket_enc_part_kvno;     /* key version number */
string kticket_enc_part_etype;      /* enc type of encrypted ticket */
string kticket_enc_flags;           /* ticket flags (forwardable, ...) */
string kticket_enc_key_type;        /* key type (des3-cbc-sha1, ...) */
uint32_t kticket_enc_key_length;    /* key length */
uintptr_t kticket_enc_key_value;    /* key data */
string kticket_enc_client;          /* client principal identifier */
string kticket_enc_transited;       /* list of transited Kerberos realms */
string kticket_enc_transited_type;  /* encoding type */
uint32_t kticket_enc_authtime;      /* time of initial authentication */
uint32_t kticket_enc_starttime;     /* ticket start time in seconds */
uint32_t kticket_enc_endtime;       /* ticket end time in seconds */
uint32_t kticket_enc_renew_till;    /* ticket renewal time in seconds */
string kticket_enc_addresses;       /* addresses associated with ticket */
string kticket_enc_authorization_data;  /* list of top-level auth types */
} kticketinfo_t;

typedef struct kdcreqinfo {
```

```
string kdcreq_padata_types;         /* list of pre-auth types */
string kdcreq_kdc_options;          /* requested ticket flags */
string kdcreq_client;               /* client principal identifier */
string kdcreq_server;               /* server principal identifier */
uint32_t kdcreq_from;               /* requested start time in seconds */
uint32_t kdcreq_till;               /* requested end time in seconds */
uint32_t kdcreq_rtime;              /* requested renewal time in seconds */
uint32_t kdcreq_nonce;              /* nonce for replay detection */
string kdcreq_etype;                /* preferred encryption types */
string kdcreq_addresses;            /* list of requested ticket addresses */
string kdcreq_authorization_data;   /* list of top-level auth types */
uint32_t kdcreq_num_additional_tickets; /* number of additional tickets */
} kdcreqinfo_t;

typedef struct kdcrepinfo {
string kdcrep_padata_types;         /* list of pre-auth types */
string kdcrep_client;               /* client principal identifier */
uint32_t kdcrep_enc_part_kvno;      /* key version number */
string kdcrep_enc_part_etype;       /* enc type of encrypted KDC reply */
string kdcrep_enc_key_type;         /* key type (des3-cbc-sha1, ...) */
uint32_t kdcrep_enc_key_length;     /* key length */
uintptr_t kdcrep_enc_key_value;     /* key data */
string kdcrep_enc_last_req;         /* times of last request of principal */
uint32_t kdcrep_enc_nonce;          /* nonce for replay detection */
uint32_t kdcrep_enc_key_expiration; /* expiration time of client's key */
string kdcrep_enc_flags;            /* ticket flags */
uint32_t kdcrep_enc_authtime;       /* time of authentication of ticket */
uint32_t kdcrep_enc_starttime;      /* ticket start time in seconds */
uint32_t kdcrep_enc_endtime;        /* ticket end time in seconds */
uint32_t kdcrep_enc_renew_till;       /* ticket renewal time in seconds*/
string kdcrep_enc_server;           /* server principal identifier */
string kdcrep_enc_caddr;            /* zero or more client addresses */
} kdcrepinfo_t;

typedef struct kapreqinfo {
string kapreq_ap_options;           /* options (use-session-key,... ) */
uint32_t kapreq_authenticator_kvno; /* key version number */
string kapreq_authenticator_etype;  /* enc type of authenticator */
} kapreqinfo_t;

typedef struct kaprepinfo {
uint32_t kaprep_enc_part_kvno;      /* key version number */
string kaprep_enc_part_etype;       /* enc type of encrypted AP reply */
uint32_t kaprep_enc_ctime;          /* client time in seconds */
uint32_t kaprep_enc_cusec;          /* client time, microseconds portion */
string kaprep_enc_subkey_type;      /* sub-key type */
uint32_t kaprep_enc_subkey_length;  /* sub-key length */
uintptr_t kaprep_enc_subkey_value;  /* sub-key data */
uint32_t kaprep_enc_seq_number;     /* sequence number */
} kaprepinfo_t;

typedef struct kerrorinfo {
uint32_t kerror_ctime;              /* client time in seconds */
uint32_t kerror_cusec;              /* client time, microseconds */
```

```
uint32_t kerror_stime;              /* server time in seconds */
uint32_t kerror_susec;              /* server time, microseconds */
string kerror_error_code;           /* error code (KRB_AP_ERR_SKEW, ...) */
string kerror_client;               /* client principal identifier */
string kerror_server;               /* server principal identifier */
string kerror_e_text;               /* additional error text */
string kerror_e_data;               /* additional error data */
} kerrorinfo_t;

typedef struct ksafeinfo {
uintptr_t ksafe_user_data;          /* raw application specific data */
uint32_t ksafe_timestamp;           /* time of sender in seconds */
uint32_t ksafe_usec;                /* time of sender, microseconds */
uint32_t ksafe_seq_number;          /* sequence number */
string ksafe_s_address;             /* sender's address */
string ksafe_r_address;             /* recipient's address */
string ksafe_cksum_type;            /* checksum type (des-cbc, ...) */
uint32_t ksafe_cksum_length;        /* length of checksum */
uintptr_t ksafe_cksum_value;        /* raw checksum data */
} ksafeinfo_t;

typedef struct kprivinfo {
uint32_t kpriv_enc_part_kvno;       /* key version number */
string kpriv_enc_part_etype;        /* enc type of encrypted message */
uintptr_t kpriv_enc_user_data;      /* raw application specific data */
uint32_t kpriv_enc_timestamp;       /* time of sender in seconds */
uint32_t kpriv_enc_usec;            /* time of sender, microseconds */
uint32_t kpriv_enc_seq_number;      /* sequence number */
string kpriv_enc_s_address;         /* sender's address */
string kpriv_enc_r_address;         /* recipient's address */
} kprivinfo_t;

typedef struct kcredinfo {
uint32_t kcred_enc_part_kvno;       /* key version number */
string kcred_enc_part_etype;        /* enc type of encrypted message */
uint32_t kcred_tickets;             /* number of tickets */
uint32_t kcred_enc_nonce;           /* nonce for replay detection */
uint32_t kcred_enc_timestamp;       /* time of sender in seconds */
uint32_t kcred_enc_usec;            /* time of sender, microseconds */
string kcred_enc_s_address;         /* sender's address */
string kcred_enc_r_address;         /* recipient's address */
} kcredinfo_t;
```

Glossaire de la sécurité

AES Standard de chiffrement avancé (Advanced Encryption Standard). Technique de chiffrement de données symétrique par blocs de 128 bits. Le gouvernement des Etats-Unis a adopté la variante Rijndael de l'algorithme comme norme de chiffrement en octobre 2000. AES remplace le chiffrement principal d'utilisateur comme norme administrative.

algorithme Algorithme cryptographique. Il s'agit d'une procédure de calcul récursive établie qui chiffre ou hache une entrée.

algorithme cryptographique Voir algorithme.

allocation de périphériques Protection des périphériques au niveau de l'utilisateur. L'allocation de périphériques met en oeuvre l'utilisation exclusive d'un périphérique par un utilisateur à la fois. Les données des périphériques sont purgées avant toute réutilisation d'un périphérique. Des autorisations peuvent être utilisées pour limiter les utilisateurs autorisés à allouer un périphérique.

amorce Valeur numérique de départ pour la génération de nombres aléatoires. Lorsque cette valeur provient d'une source aléatoire, l'amorce est appelée *amorce aléatoire*.

application privilégiée Application pouvant remplacer les contrôles système. L'application vérifie les attributs de sécurité, tels que des UID spécifiques, des ID de groupe, des autorisations ou des privilèges.

assimilation de message Valeur de hachage calculée à partir d'un message. La valeur de hachage identifie le message de manière presque unique. Une synthèse permet de vérifier l'intégrité d'un fichier.

attributs de sécurité Remplace la stratégie de sécurité qui permet à une commande d'administration de s'exécuter correctement lorsque celle-ci est exécutée par un utilisateur autre que le superutilisateur. Dans le modèle de superutilisateur, les programmes `setuid root` et `setgid` sont des attributs de sécurité. Lorsque ces attributs sont appliqués à une commande, la commande s'exécute correctement, quel que soit l'utilisateur qui l'exécute. Dans le modèle de privilège, les privilèges de noyau et les autres droits remplacent les programmes `setuid root` en tant qu'attributs de sécurité. Le modèle de privilège est compatible avec le modèle de superutilisateur dans la mesure où le modèle de privilège reconnaît également les programmes `setuid` et `setgid` comme des attributs de sécurité.

authentificateur Des authentificateurs sont transmis par des clients lors de la demande de tickets (à un KDC) et de services (à un serveur). Ils contiennent des informations générées par l'utilisation d'une clé

de session connue uniquement du client et du serveur, dont l'origine récente peut être prouvée, indiquant ainsi que la transaction est sécurisée. Lorsqu'un authentificateur est utilisé avec un ticket, il peut permettre d'authentifier un principal d'utilisateur. Un authentificateur inclut le nom du principal de l'utilisateur, l'adresse IP de l'hôte de l'utilisateur, ainsi qu'un horodatage. A la différence d'un ticket, un authentificateur ne peut servir qu'une seule fois, généralement lorsque l'accès à un service est demandé. Un authentificateur est chiffré à l'aide de la clé de session pour ce client et ce serveur.

authentification Processus de vérification de l'identité déclarée d'un principal.

autorisation 1. Dans Kerberos, processus consistant à déterminer si un principal peut utiliser un service et à définir les objets auxquels il peut accéder, ainsi que le type d'accès autorisé pour chaque objet.

2. Dans la gestion des droits des utilisateurs, droit pouvant être affecté à un rôle ou un utilisateur (ou intégré dans un profil de droits) en vue de l'exécution d'une classe d'opérations autrement interdites par la stratégie de sécurité. Les "feux verts" sont mises en oeuvre au niveau de l'application utilisateur, et non dans le noyau.

Blowfish Algorithme de chiffrement par bloc symétrique de longueur de clé variable (entre 32 et 448 bits). Son créateur, Bruce Schneier, affirme que Blowfish est optimisé pour les applications pour lesquelles la clé n'a pas besoin d'être régulièrement modifiée.

cache d'informations d'identification Espace de stockage (généralement un fichier) contenant des informations d'identification reçues de KDC.

champ d'application du service de noms Champ d'application dans lequel un rôle est autorisé à fonctionner, c'est-à-dire un hôte particulier ou tous les hôtes desservis par un service de noms spécifié, tel que NIS LDAP.

chiffrement par clé privée Dans le cas du chiffrement par clé privée, l'expéditeur et le destinataire utilisent la même clé de chiffrement. Voir également chiffrement par clé publique.

chiffrement par clé publique Modèle de chiffrement où chaque utilisateur dispose de deux clés, l'une publique et l'autre privée. Dans le cas du chiffrement par clé publique, l'expéditeur chiffre le message à l'aide de la clé publique du destinataire, et ce dernier se sert d'une clé privée pour le déchiffrer. Le service Kerberos est un système à clé privée. Voir également chiffrement par clé privée.

clé 1. En règle générale, l'un des deux principaux types de clés :

- *clé symétrique* : clé de chiffrement identique à la clé de déchiffrement. Les clés symétriques sont utilisées pour chiffrer des fichiers.
- *clé asymétrique* ou *clé publique* : clé utilisée dans les algorithmes à clé publique, tels que Diffie-Hellman ou RSA. Les clés publiques contiennent une clé privée, connue uniquement d'un utilisateur, une clé publique utilisée par le serveur ou des ressources générales, et une paire de clés publique-privée combinant les deux. Une clé privée est également qualifiée de clé *secrète*. La clé publique est également qualifiée de clé *partagée* ou *commune*.

2. Entrée (nom de principal) dans un fichier keytab. Voir également fichier keytab.

3. Dans Kerberos, clé de chiffrement dont il existe trois types :

- *Clé privée* : clé de chiffrement partagée par un principal et le KDC, et distribuée en dehors des limites du système. Voir également clé privée.
- *Clé de service* : clé remplissant la même fonction que la clé privée, mais utilisée par des serveurs et des services. Voir également clé de service.
- *Clé de session* : clé de chiffrement temporaire utilisée entre deux principaux, avec une durée de vie limitée à la durée d'une seule session de connexion. Voir également clé de session.

clé de service Clé de chiffrement partagée par un principal de service et le KDC, et distribuée en dehors des limites du système. Voir également clé.

clé de session Clé générée par le service d'authentification ou le service d'octroi de ticket. Une clé de session est générée dans le but de sécuriser les transactions entre un client et un service. La durée de vie d'une clé de session est limitée à une seule session de connexion. Voir également clé.

clé privée Chaque principal d'utilisateur reçoit une clé qui n'est connue que du KDC et de l'utilisateur du principal. Pour les principaux d'utilisateur, la clé est basée sur le mot de passe de l'utilisateur. Voir également clé.

clé secrète Reportez-vous à clé privée.

client Au sens strict, il s'agit d'un processus utilisant un service réseau pour le compte d'un utilisateur, par exemple, une application utilisant `rlogin`. Dans certains cas, un serveur peut être lui-même le client d'un autre serveur ou service.

Au sens large, il s'agit d'un hôte qui a) reçoit des informations d'identification Kerberos et b) utilise un service fourni par un serveur.

Dans la pratique, ce terme désigne un principal utilisant un service.

confidentialité Reportez-vous à confidentialité.

confidentialité Un service de sécurité, dans lequel les données transmises sont chiffrées avant leur envoi. La confidentialité inclut également l'intégrité des données et l'authentification de l'utilisateur. Voir également authentification, intégrité, service.

conscient des privilèges Programmes, scripts et commandes qui activent et désactivent l'utilisation des privilèges dans leur code. Dans un environnement de production, les privilèges qui sont activés doivent être fournis au processus, par exemple, en demandant aux utilisateurs du programme d'utiliser un profil de droits qui ajoute les privilèges au programme. Pour une description complète des privilèges, reportez-vous à la page de manuel `privileges`(5).

DES Standard de chiffrement de données (Data Encryption Standard). Méthode de chiffrement à clé symétrique développée en 1975 et standardisée par l'ANSI en 1981 comme ANSI X.3.92. Le DES utilise une clé de 56 bits.

destinataire Dans la fonctionnalité Structure cryptographique d'Oracle Solaris, un consommateur est un utilisateur des services cryptographiques provenant de fournisseurs. Les consommateurs peuvent être des applications, des utilisateurs finaux ou des opérations de noyau. Kerberos, IKE et IPsec sont des exemples de consommateurs. Pour consulter des exemples de fournisseurs, reportez-vous à la section fournisseur.

domaine 1. Réseau logique desservi par une seule base de données Kerberos et un ensemble de centre de distribution des clés (KDC).

2. Troisième partie d'un nom de principal. Pour le nom de principal `jdoe/admin@CORP.EXAMPLE.COM`, le domaine est `CORP.EXAMPLE.COM`. Voir également nom de principal.

droits Alternative au modèle tout ou rien des superutilisateurs. La gestion des droits des utilisateurs et la gestion des droits de processus permettent à une organisation de répartir les privilèges du superutilisateur et que vous les affectez aux utilisateurs ou rôles. Dans Oracle Solaris, les droits sont implémentés sous la forme de privilèges sur le noyau, d'autorisations et de la possibilité d'exécuter un processus en tant que UID ou GID spécifique. Les droits peuvent être collectés dans un profil de droits et un rôle.

DSA Algorithme de signature numérique (Digital Signature Algorithm). Algorithme de clé publique dont la longueur de clé varie de 512 à 4 096 bits. La norme du gouvernement américain, DSS, atteint 1 024 bits. L'algorithme DSA repose sur l'algorithme SHA1 en entrée.

écart d'horloge Ecart maximal toléré entre les horloges système internes de tous les hôtes participant au système d'authentification Kerberos. Si l'écart d'horloge est dépassé entre des hôtes participants, les demandes sont rejetées. L'écart d'horloge est spécifié dans le fichier `krb5.conf`.

ECDSA Algorithme de signature numérique de courbe elliptique. Algorithme de clé publique basé sur une courbe elliptique mathématique. Une clé ECDSA est de beaucoup plus petite taille qu'une clé publique DSA nécessaire pour générer une signature de la même longueur.

escalade de privilèges Accès aux ressources situées en dehors de la plage, les ressources qui y compris les droits qui vous sont affectés qui remplacent les valeurs par défaut, droits y autorisent. Il en résulte qu'un processus peut effectuer des opérations non autorisées.

événement d'audit asynchrone Les événements asynchrones constituent la minorité des événements système. Ces événements ne sont associés à aucun processus, de sorte qu'aucun processus ne peut être bloqué puis réactivé ultérieurement. L'initialisation système initiale et les événements d'entrée et de sortie PROM sont des exemples d'événements asynchrones.

événement d'audit non allouable Evénement d'audit dont l'initiateur ne peut pas être déterminé, tel que l'événement AUE_BOOT.

événement d'audit synchrone Majorité des événements d'audit. Ces événements sont associés à un processus dans le système. Un événement non allouable associé à un processus est un événement synchrone, tel que l'échec d'une connexion.

fichier de ticket Voir cache d'informations d'identification.

fichier keytab Fichier de table de clés contenant une ou plusieurs clés (principaux). Un hôte ou un service utilise un fichier keytab à peu près de la même façon qu'un utilisateur se sert d'un mot de passe.

fichier stash Un fichier stash contient une copie chiffrée de la clé principale pour le KDC. Cette clé principale est utilisée lorsqu'un serveur est réinitialisé pour authentifier automatiquement le KDC avant qu'il ne démarre les processus `kadmind` et `krb5kdc`. Etant donné que le fichier stash inclut la clé principale, ce fichier et toutes ses sauvegardes doivent être sécurisés. Si le chiffrement est compromis, la clé peut être utilisée pour accéder à la base de données KDC ou la modifier.

fichiers d'audit Journaux d'audit binaires. Les fichiers d'audit sont stockés séparément dans un système de fichiers d'audit.

fournisseur Dans la fonctionnalité Structure cryptographique d'Oracle Solaris, un service cryptographique est fourni aux consommateurs. Les bibliothèques PKCS #11, les modules cryptographiques du noyau et les accélérateurs de matériel sont des exemples de fournisseurs. Les fournisseurs se connectent à la structure cryptographique et sont donc également appelés *plug-ins*. Pour consulter des exemples de consommateurs, voir destinataire.

fournisseur de logiciels Dans la fonctionnalité Structure cryptographique d'Oracle Solaris, module logiciel de noyau ou bibliothèque PKCS#11 fournissant des services cryptographiques. Reportez-vous également à fournisseur.

fournisseur de matériel Dans la fonctionnalité Structure cryptographique d'Oracle Solaris, un pilote de périphérique et son accélérateur matériel. Les fournisseurs de matériel déchargent le système informatique d'opérations cryptographiques coûteuses, libérant ainsi les ressources de la CPU pour d'autres utilisations. Reportez-vous également à fournisseur.

FQDN Nom de domaine complet. Par exemple, `central.example.com` (et pas simplement `denver`).

GSS-API Interface de programmation d'application générique de service de sécurité. Couche réseau assurant la prise en charge de différents services de sécurité modulaires, y compris le service Kerberos. GSS-API fournit des services d'authentification, d'intégrité et de confidentialité. Voir également authentification, intégrité, confidentialité.

hôte Système accessible par l'intermédiaire d'un réseau.

image système unique Une image système unique est utilisée dans l'audit d'Oracle Solaris afin de décrire un groupe de systèmes audités utilisant le même service de noms. Ces systèmes envoient leurs enregistrements d'audit à un serveur d'audit central où les enregistrements peuvent être comparés comme s'ils provenaient d'un même système.

informations d'identification Package d'informations comprenant un ticket et une clé de session correspondante. Informations utilisées pour authentifier l'identité d'un principal. Voir également ticket, clé de session.

instance Deuxième partie d'un nom de principal, une instance qualifie le primaire du principal. Dans le cas d'un principal de service, l'instance est requise. L'instance est le nom de domaine complet de l'hôte, comme dans `host/central.example.com`. Pour les principaux d'utilisateur, une instance est facultative. Notez, cependant, que `jdoe` et `jdoe/admin` sont des principaux uniques. Voir également primaire, nom de principal, principal de service, principal d'utilisateur.

intégrité Service de sécurité qui assure, outre l'authentification de l'utilisateur, la validité des données transmises par le biais de sommes de contrôle cryptographiques. Voir également authentification, confidentialité.

jeu autorisé Jeu des privilèges disponibles pour l'utilisation par un processus.

jeu de base Jeu de privilèges affecté à un processus d'utilisateur lors de la connexion. Sur un système non modifié, le jeu héritable initial de chaque utilisateur correspond au jeu de base défini au moment de la connexion.

jeu de privilèges Ensemble de privilèges. Chaque processus comporte quatre jeux de privilèges qui déterminent s'il peut utiliser un privilège particulier. Voir jeu limite, jeu effectif, jeu autorisé et jeu hérité.

De même, le jeu de base de privilèges est l'ensemble des privilèges affectés aux processus d'un utilisateur au moment de la connexion.

jeu effectif Jeu de privilèges actuellement en vigueur sur un processus.

jeu hérité Jeu de privilèges dont un processus peut hériter en appelant `exec`.

jeu limite Limite extérieure des privilèges disponibles pour un processus et ses enfants.

KDC Centre de distribution de clés (Key Distribution Center). Machine disposant de trois composants Kerberos V5.

- Principal et base de données de clés
- Service d'authentification
- Service d'octroi de tickets

Chaque domaine dispose d'un KDC maître et doit avoir un ou plusieurs KDC esclaves.

KDC esclave Copie d'un KDC maître capable de réaliser la plupart des fonctions du maître. Chaque domaine dispose généralement de plusieurs KDC esclaves et d'un seul KDC maître. Voir également KDC, KDC maître.

KDC maître KDC maître dans chaque domaine, comprenant un serveur d'administration Kerberos, `kadmind` et un démon d'authentification et d'octroi de tickets, `krb5kdc`. Chaque domaine doit disposer d'au moins un KDC maître, et peut avoir plusieurs KDC de duplication ou esclaves fournissant des services d'authentification aux clients.

Kerberos Service d'authentification, protocole utilisé par ce service ou code servant à mettre en oeuvre ce service.

Mise en oeuvre Kerberos d'Oracle Solaris étroitement basée sur la mise en oeuvre Kerberos V5.

Bien que techniquement différents, "Kerberos" et "Kerberos V5" sont souvent utilisés de façon interchangeable dans la documentation Kerberos.

Dans la mythologie grecque, Kerberos (en français Cerbère) était un chien féroce tricéphale qui gardait la porte des Enfers.

keystore Un keystore contient des mots de passe, des phrases de passe, des certificats et d'autres objets d'authentification pour l'extraction par des applications. Il peut être spécifique à une technologie, ou à un emplacement utilisé par plusieurs applications.

kvno Numéro de version de la clé. Numéro de série faisant le suivi d'une clé spécifique selon l'ordre dans lequel elle a été générée. Plus le numéro kvno est élevé, plus la clé est récente.

liste de contrôle d'accès (ACL) Une liste de contrôle d'accès (ACL) offre une sécurité des fichiers plus précise que la protection de fichier UNIX conventionnelle. Par exemple, une ACL vous permet d'autoriser l'accès en lecture de groupe à un fichier, tout en autorisant un seul membre de ce groupe à écrire dans le fichier.

MAC 1. Voir MAC.

2. Egalement appelé étiquetage. Dans la terminologie de sécurité gouvernementale, MAC signifie Mandatory Access Control (contrôle d'accès obligatoire). Les étiquettes telles que Top Secret et Confidential sont des exemples de MAC. MAC diffère de DAC (Discretionary Access Control, contrôle d'accès discrétionnaire). Les autorisations UNIX constituent un exemple de DAC.

3. Dans le matériel, il s'agit de l'adresse système unique sur un réseau local (LAN). Si le système est sur un réseau Ethernet, le MAC est l'adresse Ethernet.

MAC MAC garantit l'intégrité des données et authentifie leur origine. MAC ne protège aucunement contre l'écoute frauduleuse des informations échangées.

MD5 Fonction de hachage cryptographique répétitive utilisée pour authentifier les messages, y compris les signatures numériques. Elle a été développée en 1991 par Rivest. Son utilisation a été désapprouvée.

mécanisme 1. Package logiciel spécifiant des techniques cryptographiques pour assurer l'authentification ou la confidentialité des données. Exemples : Kerberos V5, clé publique Diffie-Hellman

2. Dans la fonctionnalité Structure cryptographique d'Oracle Solaris, mise en oeuvre d'un algorithme destiné à un usage particulier. Par exemple, un mécanisme DES appliqué à l'authentification, tel que CKM_DES_MAC, est un mécanisme distinct d'un mécanisme DES appliqué au chiffrement, CKM_DES_CBC_PAD.

mécanisme de sécurité Voir mécanisme.

minimisation Installation du système d'exploitation minimal nécessaire pour l'exécution du serveur. Tout logiciel n'étant pas directement lié au fonctionnement du serveur n'est pas installé, ou est supprimé après l'installation.

modèle de privilège Modèle de sécurité plus strict que le modèle superutilisateur sur un système informatique. Dans le modèle de privilège, les processus nécessitent des privilèges pour s'exécuter. L'administration du système peut être divisée en quatre parties discrètes basées sur les privilèges dont les administrateurs disposent dans leurs processus. Des privilèges peuvent être affectés au processus de connexion d'un administrateur, ou des privilèges peuvent être affectés de manière à ne s'appliquer qu'à certaines commandes.

modèle de superutilisateur Modèle de sécurité UNIX standard sur un système informatique. Dans le modèle de superutilisateur, un administrateur dispose d'un contrôle de type tout ou rien sur le système. En règle générale, pour l'administration de la machine, un utilisateur se connecte en tant que superutilisateur (`root`) et peut effectuer toutes les activités d'administration.

moindre privilège Modèle de sécurité qui octroie à un processus spécifié uniquement un sous-ensemble de pouvoirs de superutilisateur. Le modèle de moindre privilège octroie aux utilisateurs standard les privilèges suffisants pour qu'ils puissent effectuer des tâches d'administration personnelles, telles que le montage de systèmes de fichiers et la modification de l'appartenance des fichiers. Par ailleurs, les processus s'exécutent uniquement avec les privilèges dont ils ont besoin pour terminer la tâche, et non avec tous les pouvoirs du superutilisateur, c'est-à-dire, tous les privilèges. Les dommages causés par les erreurs de programmation, comme les débordements de la mémoire tampon, peuvent être limités à un utilisateur non root, qui n'a pas accès à des fonctions essentielles comme la lecture ou l'écriture de fichiers système protégés ou l'arrêt de la machine.

moteur d'analyse Application tierce, résidant sur un hôte externe, qui examine un fichier à la recherche de virus connus.

nom de principal 1. Le nom d'un principal, au format *primary/instance@REALM*. Voir également instance, primaire, domaine.

2. (RPCSEC_GSS API) Voir principal de client, principal de serveur.

NTP Network Time Protocol. Logiciel de l'Université de l'Etat du Delaware qui vous permet de gérer avec précision la synchronisation de l'heure ou de l'horloge réseau, ou les deux, dans un environnement réseau. Vous pouvez utiliser le protocole NTP pour préserver l'écart d'horloge dans un environnement Kerberos. Voir également écart d'horloge.

objet public Fichier appartenant à l'utilisateur `root` et lisible par tout le monde, tel que n'importe quel fichier du répertoire `/etc`.

PAM Module d'authentification enfichable (Pluggable Authentication Module). Structure permettant d'utiliser plusieurs mécanismes d'authentification sans recompilation des services recourant à ces mécanismes. PAM permet l'initialisation de la session SEAM au moment de son ouverture

phrase de passe Phrase utilisée pour vérifier qu'une clé privée a été créée par l'utilisateur de la phrase de passe. Une bonne phrase de passe contient 10 à 30 caractères alphanumériques et évite les noms et

proses simples. Vous êtes invité à saisir la phrase de passe pour authentifier l'utilisation de la clé privée pour chiffrer et déchiffrer les communications.

primaire Première partie du nom d'un nom de principal. Voir également instance, nom de principal, domaine.

principal 1. Client/utilisateur dont le nom est unique ou instance de serveur/service participant à une communication en réseau. Les transactions Kerberos impliquent des interactions entre des principaux (principaux de service et d'utilisateur) ou entre des principaux et des KDC. En d'autres termes, un principal est une entité unique à laquelle Kerberos peut attribuer des tickets. Voir également nom de principal, principal de service, principal d'utilisateur.

2. (RPCSEC_GSS API) Voir principal de client, principal de serveur.

Principal admin Principal d'utilisateur dont le nom est au format *nom_utilisateur*`/admin` (comme dans `jdoe/admin`). Un principal admin peut disposer de davantage de privilèges (de modification de stratégies par exemple) qu'un principal d'utilisateur standard. Voir également nom de principal et principal d'utilisateur.

principal d'hôte Instance spécifique d'un principal de service dans lequel le principal (indiqué par le nom primaire `host`) est configuré de manière à fournir une gamme de services réseau, comme `ftp`, `rcp` ou `rlogin`. `host/central.example.com@EXAMPLE.COM` est un exemple d'hôte principal. Voir également principal de serveur.

principal d'utilisateur Principal attribué à un utilisateur particulier. Le nom primaire d'un principal d'utilisateur est un nom d'utilisateur et son instance facultative est un nom utilisé pour décrire l'utilisation prévue des informations d'identification correspondantes (`jdoe` ou `jdoe/admin` par exemple). Egalement appelé instance d'utilisateur. Voir également principal de service.

principal de client (RPCSEC_GSS API) Client (utilisateur ou application) utilisant des services réseau sécurisés RPCSEC_GSS. Les noms de principaux client sont stockés sous forme de structures `rpc_gss_principal_t`.

principal de serveur (RPCSEC_GSS API) Principal fournissant un service. Le principal de serveur est stocké sous forme d'une chaîne de caractères ASCII dont le format est *service*`@`*hôte*. Voir également principal de client.

principal de service Principal assurant l'authentification Kerberos pour un ou plusieurs services. Pour les principaux de service, le nom primaire est un nom de service, tel que `ftp`, et son instance est le nom d'hôte complet du système fournissant le service. Voir également principal d'hôte, principal d'utilisateur.

principe du moindre privilège Reportez-vous à moindre privilège.

privilège En règle générale, les d'alimentation 1. capable d'effectuer une opération ou sur un système informatique qui se situe au-delà de la puissance de un utilisateur standard. Les privilèges de

superutilisateur correspondent à l'ensemble desdroits octroyés au superutilisateur. Application privilégiée par un utilisateur doté de privilèges ou est un utilisateur ou application qui a reçu des droits supplémentaires.

2. Droit discret dans le cadre d'un processus dans un système Oracle Solaris. Les privilèges offrent un contrôle des processus plus détaillé que `root`. Les privilèges sont définis et appliqués dans le noyau. Les privilèges sont également appelés *privilèges de processus* ou *privilèges de noyau*. Pour une description complète des privilèges, reportez-vous à la page de manuel `privileges`(5).

profil de droits On parle aussi de profil. Collection de remplacements de sécurité pouvant être affectés à un rôle ou à un utilisateur. Un profil de droits peut se composer d'autorisations, privilèges, de commandes avec des attributs de sécurité et d'autres profils de droits sont appelés les profils supplémentaires.

profil de droits authentifié Un profil de droits qui exige que l'utilisateur ou le rôle affecté saisisse un mot de passe avant d'exécuter une opération à partir du profil. Ce comportement est similaire au comportement `sudo`. La durée de temps pendant laquelle le mot de passe est valide est configurable.

protocole Diffie-Hellman Egalement appelé cryptographie par clé publique. Protocole d'accord de clés cryptographiques asymétriques mis au point par Diffie et Hellman en 1976. Ce protocole permet à deux utilisateurs d'échanger une clé secrète via un moyen non sécurisé sans secrets préalables. Diffie-Hellman est utilisé par Kerberos.

QOP Qualité de la protection. Paramètre servant à sélectionner les algorithmes cryptographiques utilisés en association avec le service d'intégrité ou de confidentialité.

RBAC Contrôle d'accès basé sur les rôles, la fonction de gestion des droits des utilisateurs d'Oracle Solaris. Reportez-vous à droits.

réauthentification Mot de passe qu'il soit nécessaire de fournir une opération. pour effectuer un ordinateur En règle générale, les opérations `sudo` nécessitent une réauthentification. Les profils de droits peuvent contenir authentifiés des commandes qui nécessitent la réauthentification. Reportez-vous à profil de droits authentifié.

relation Variable ou relation de configuration définie dans les fichiers `kdc.conf` ou `krb5.conf`.

renforcement Modification de la configuration par défaut du système d'exploitation pour supprimer les failles de sécurité inhérentes à l'hôte.

rôle Identité spéciale destinée à l'exécution d'applications privilégiées et ne pouvant être prise que par des utilisateurs assignés.

RSA Méthode permettant d'obtenir des signatures numériques et des systèmes de cryptographie par clé publique. Cette méthode datant de 1978 a été décrite par trois développeurs (Rivest, Shamir et Adleman).

SEAM Le nom de produit pour la version initiale de Kerberos sur les systèmes Solaris. Ce produit est à partir de la technologie Kerberos V5 développé au Massachusetts Institute of Technology.

SEAM est maintenant appelé "service Kerberos". Il continue d'être légèrement différent de l'version MIT.

séparation des tâches Composante de la notion de moindre privilège. La séparation des tâches permet d'empêcher un utilisateur d'exécuter ou d'approuver les opérations terminant une transaction. Par exemple, dans RBAC, vous pouvez séparer la création d'un utilisateur de connexion de l'affectation des remplacements de sécurité. Un rôle crée l'utilisateur. Un autre rôle peut affecter les attributs de sécurité, tels que des profils de droits, des rôles et des privilèges aux utilisateurs existants.

serveur Principal fournissant une ressource aux clients réseau. Si vous vous connectez par `ssh` au système `central.example.com` par exemple, ce système est le serveur fournissant le service `ssh`. Voir également principal de service.

serveur d'application Voir serveur d'application réseau.

serveur d'application réseau Serveur fournissant une application réseau, telle que `ftp`. Un domaine peut contenir plusieurs serveurs d'application réseau.

service 1. Ressource fournie aux clients du réseau, souvent par plusieurs serveurs. Si vous vous connectez par `rlogin` à la machine `central.example.com` par exemple, cette machine est le serveur fournissant le service `rlogin`.

2. Service de sécurité (d'intégrité ou de confidentialité) fournissant un niveau de protection supérieur à l'authentification. Voir également intégrité et confidentialité.

service de sécurité Voir service.

SHA1 Secure Hashing Algorithm, algorithme de hachage sécurisé. L'algorithme s'applique à toute longueur d'entrée inférieure à 2^{64} afin d'obtenir une synthèse des messages. L'algorithme SHA1 sert d'entrée à DSA.

shell de profil Dans la gestion des droits, shell permettant à un rôle (ou un utilisateur) d'exécuter à partir de la ligne de commande toutes les applications privilégiées affectées aux profils de droits du rôle. Les versions du shell de profil correspondent aux shells disponibles sur le système, tels que la version `pfbash` de `bash`.

shell sécurisé Un protocole particulier pour une connexion à distance sécurisée et d'autres services de réseau sécurisé via un réseau non sécurisé.

stratégie En règle générale, plan ou ensemble d'actions qui influence ou détermine les décisions et actions. Pour les systèmes informatiques, la stratégie fait généralement référence à la stratégie de sécurité. La stratégie de sécurité de votre site constitue un ensemble de règles qui définissent la sensibilité des informations traitées et les mesures prises pour protéger les informations contre tout accès non autorisé. Par exemple, la stratégie de sécurité peut exiger que les systèmes soient audités, que les périphériques soient alloués pour être utilisés et que les mots de passe soient modifiés toutes les six semaines.

Pour la mise en oeuvre d'une stratégie dans des zones spécifiques du Oracle Solaris (SE), reportez-vous à stratégie d'audit, stratégie dans la structure cryptographique, stratégie de périphériques, stratégie Kerberos, stratégie de mot de passe et stratégie de droits.

stratégie d'audit Paramètres globaux et par utilisateur qui déterminent les événements d'audit enregistrés. Les paramètres globaux s'appliquant au service d'audit déterminent généralement les informations facultatives à inclure dans la piste d'audit. Deux paramètres, `cnt` et `ahlt`, affectent le fonctionnement du système lorsque la file d'attente de l'audit est pleine. Par exemple, la stratégie d'audit peut exiger qu'un numéro de séquence fasse partie de chaque enregistrement d'audit.

stratégie dans la structure cryptographique Dans la fonctionnalité Structure cryptographique d'Oracle Solaris, la stratégie correspond à la désactivation de mécanismes de chiffrement existants. Les mécanismes ne peuvent ensuite plus être utilisés. La stratégie de la structure cryptographique peut empêcher l'utilisation d'un mécanisme particulier tel que `CKM_DES_CBC` provenant d'un fournisseur tel que DES.

stratégie de droits Stratégie de sécurité associée à une commande. Actuellement, `solaris` est la stratégie valide pour Oracle Solaris. La stratégie `solaris` reconnaît les privilèges et la stratégie de privilèges étendue, les autorisations et les attributs de sécurité `setuid`.

stratégie de mot de passe Algorithmes de chiffrement pouvant être utilisés pour générer des mots de passe. Peut également faire référence à des questions plus générales concernant les mots de passe, telles que la fréquence à laquelle le mot de passe doit être modifié, le nombre de tentatives autorisées de saisie d'un mot de passe et d'autres considérations relatives à la sécurité. La stratégie de sécurité requiert des mots de passe. La stratégie de mot de passe peut requérir des mots de passe chiffrés avec l'algorithme AES, et imposer d'autres exigences relatives à la force des mots de passe.

stratégie de périphériques Protection d'un périphérique au niveau du noyau. La stratégie de périphériques repose sur deux jeux de privilèges pour un périphérique. Un jeu de privilèges contrôle l'accès en lecture au périphérique. Le deuxième jeu de privilèges contrôle l'accès en écriture sur le périphérique. Voir également stratégie.

stratégie de sécurité Voir stratégie.

stratégie Kerberos Ensemble de règles régissant l'utilisation des mots de passe dans le service Kerberos. Les stratégies peuvent réguler les accès des principaux ou des paramètres de ticket, tels que la durée de vie.

stratégie pour technologies à clé publique Dans la structure de gestion des clés (KMF), la stratégie correspond à la gestion de l'utilisation des certificats. La base de données de stratégies KMF peut définir des contraintes s'appliquant à l'utilisation des clés et des certificats gérés par la bibliothèque KMF.

stratégie RBAC Reportez-vous à stratégie de droits.

stratégies de réseau Paramètres configurés par les utilitaires réseau pour protéger le trafic réseau. Pour plus d'informations sur la sécurité réseau, reportez-vous à la section " Sécurisation du réseau dans Oracle Solaris 11.2 ".

synthèse Voir assimilation de message.

TGS Service d'octroi de tickets (Ticket-Granting Service) Partie du KDC responsable de l'émission des tickets.

TGT Ticket d'octroi de tickets (Ticket-Granting Ticket) Ticket émis par le KDC, permettant au client de demander des tickets pour d'autres services.

ticket Paquet d'informations servant à transmettre en toute sécurité l'identité d'un utilisateur à un serveur ou un service. Un ticket n'est valable que pour un client et un service particulier sur un serveur spécifique. Il contient le nom de principal du service, le nom de principal de l'utilisateur, l'adresse IP de l'hôte de l'utilisateur, un horodatage et une valeur définissant la durée de vie du ticket. La création d'un ticket s'effectue à l'aide d'une clé de session aléatoire utilisée par le client et le service. Une fois le ticket créé, il peut être réutilisé jusqu'à son expiration. Un ticket sert uniquement à authentifier un client lorsqu'il est présenté avec un nouvel authentificateur. Voir également authentificateur, informations d'identification, service, clé de session.

ticket initial Ticket émis de manière directe, et pas à partir d'un ticket d'octroi de tickets. Certains services, tels que les applications modifiant les mots de passe, peuvent nécessiter des tickets marqués comme étant `initiaux` afin d'assurer que le client peut démontrer qu'il connait sa clé secrète. Cette garantie est importante car un ticket initial indique que le client s'est récemment authentifié et ne dépend pas d'un ticket d'octroi de tickets qui peut exister depuis un certain temps.

ticket non valide Ticket postdaté n'étant pas encore utilisable. Un ticket non valide est rejeté par un serveur d'application jusqu'à ce qu'il soit validé. Pour être validé, un ticket non valide doit être présenté au KDC par le client dans une demande de TGS, avec l'indicateur `VALIDATE`, après l'heure de début. Voir également ticket postdaté.

ticket postdaté Un ticket postdaté ne devient valide qu'un certain temps après sa création. Par exemple, un tel ticket peut être utile avec les tâches exécutées par lots la nuit car le ticket, s'il est volé, ne peut pas être utilisé tant que l'exécution de ces tâches n'a pas eu lieu. Lorsqu'un ticket postdaté est émis, il est émis en tant que `non valide` et le reste jusqu'à ce que a) son heure de début soit dépassée et b) le client demande la validation par le KDC. Un ticket postdaté est normalement valide jusqu'à l'heure d'expiration du ticket d'octroi de tickets. Toutefois, si le ticket postdaté est marqué comme `renouvelable`, sa durée de vie est normalement égale à la durée de vie entière du ticket d'octroi de tickets. Voir également ticket non valide, ticket renouvelable.

ticket renouvelable Etant donné que tout ticket dont la durée de vie est très longue peut présenter un risque pour la sécurité, les tickets peuvent être conçus pour être `renouvelables`. Un ticket renouvelable possède deux moments d'expiration : a) l'heure à laquelle l'instance courante du ticket expire et b) la durée de vie maximale de tout ticket. Si un client souhaite continuer à utiliser un ticket, il

peut le renouveler avant sa première expiration. Par exemple, un ticket peut être valide pendant une heure, et tous les tickets ont une durée de vie maximale de dix heures. Si le client détenant le ticket souhaite le conserver plus d'une heure, il doit le renouveler. Lorsqu'un ticket atteint sa durée de vie maximale, celui-ci expire automatiquement et ne peut pas être renouvelé.

ticket transmissible Ticket pouvant être utilisé par un client pour demander un ticket sur un hôte distant sans devoir se soumettre au processus complet d'authentification sur l'hôte concerné. Par exemple, si l'utilisateur `david` obtient un ticket transmissible sur la machine de l'utilisateur `jennifer`, `david` peut se connecter à sa propre machine sans avoir à demander un nouveau ticket (et donc à s'authentifier de nouveau). Voir également ticket utilisable avec proxy.

ticket utilisable avec proxy Ticket pouvant être utilisé par un service pour le compte d'un client afin d'effectuer une opération pour ce dernier. On dit alors que le service agit en tant que proxy du client. Grâce à ce ticket, le service peut prendre l'identité du client. Le service peut utiliser un tel ticket afin d'obtenir un ticket de service d'un autre service, mais non un ticket d'octroi de tickets. La différence entre un ticket utilisable avec proxy et un ticket transmissible réside dans le fait que le premier n'est valide que pour une seule opération. Voir également ticket transmissible.

trace d'audit Collection de tous les fichiers d'audit provenant de tous les hôtes.

utilisateur privilégié Droits un utilisateur disposant des droits d'utilisateur standard au-delà de la sur un système informatique. Reportez-vous également aux utilisateurs de confiance.

utilisateurs de confiance Que vous avez décidé des utilisateurs avec lesquels vous pouvez effectuer des tâches d'administration à un certain niveau d'approbation. Les connexions aux fins en général, les administrateurs d'abord créer et affecter des utilisateurs de confiance des droits d'administration qui correspondent au niveau de confiance des utilisateurs et leur capacité. Ces utilisateurs aider à configurer et de tenir à jour le système. Egalement appelés *utilisateurs privilégiés*.

variante Historiquement, la *variante de sécurité* et la *variante d'authentification* désignaient la même chose, lorsqu'une variante indiquait un type d'authentification (AUTH_UNIX, AUTH_DES, AUTH_KERB). RPCSEC_GSS est également une variante de sécurité, même s'il permet d'assurer des services d'intégrité et de confidentialité, en plus de l'authentification.

variante de sécurité Reportez-vous à variante.

VPN Réseau privé virtuel assurant une communication sécurisée en utilisant les mécanismes de chiffrement et de mise en tunnel pour connecter les utilisateurs via un réseau public.

Index

A

B

C

D

E

F

G

H

I

P

R

S

T

U

V

W