Bull

AIX 5L Guide de sécurité

AIX

REFERENCE
86 F2 57EM 01

Bull

AIX 5L Guide de sécurité

AIX

Logiciel

Février 2005

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

REFERENCE
86 F2 57EM 01

Imprimé en France

Vos suggestions sur la forme et le fond de ce manuel seront les bienvenues. Une feuille destinée à recevoir vos remarques se trouve à la fin de ce document.

Pour commander d’autres exemplaires de ce manuel ou d’autres publications techniques Bull, veuillez utiliser le bon de commande également fourni en fin de manuel.

Marques déposées

Toutes les marques déposées sont la propriété de leurs titulaires respectifs.

AIX® est une marque déposée d’IBM Corp. et est utilisée sous licence.

UNIX est une marque déposée aux Etats–Unis et dans d’autres pays, licenciée exclusivement par Open Group.

Linux est une marque déposée de Linus Torvalds.

Préface

Le présent guide fournit aux administrateurs système des informations détaillées sur la sécurité liée aux fichiers, aux systèmes et aux réseaux. Ce guide explique comment renforcer un système, modifier les droits d'accès, mettre en oeuvre les méthodes d'authentification et configurer les fonctionnalités d'évaluation de la sécurité par des critères communs (Common Criteria Security Evaluation). Cette publication est également disponible sur le "Hypertext Library for AIX 5.3" CD-ROM fourni avec le système d'exploitation.

Conventions typographiques

Les conventions typographiques utilisées sont les suivantes :

Gras	Identifie les commandes, les sous–programmes, les mots clés, les fichiers, les structures, les répertoires, et les autres éléments dont le nom est défini par le système. Identifie également les objets de l'interface graphique, tels que les boutons, les libellés et les icônes sélectionnés par l'utilisateur.
Italique	Identifier les paramètres dont les noms ou les valeurs doivent être indiqués par l'utilisateur.
`Espacement fixe`	Identifier des exemples de données, des exemples de textes similaires à ceux affichés à l'écran, des parties de code similaires à celui que vous serez susceptible de rédiger, des messages système, ou des informations que vous devez saisir.

Distinction majuscules/minuscules dans AIX

La distinction majuscules/minuscules s'applique à toutes les données entrées dans le système d'exploitation AIX. Par exemple, la commande **ls** afficher la liste des fichiers. Si vous entrez `LS`, le système affiche un message d'erreur indiquant que la commande entrée est introuvable. De la même manière, **FICHEA**, **FiChea** et **fichea** sont trois noms de fichiers distincts, même s'ils se trouvent dans le même répertoire. Pour éviter toute effet inattendu, vérifiez systématiquement que vous utilisez la casse appropriée.

ISO 9000

Des systèmes homologués ISO 9000 ont été utilisés pour le développement et la fabrication de ce produit.

Bibliographie

Les publications suivantes contiennent des informations connexes :

- *AIX 5L Version 5.3 System Management Guide: Operating System and Devices*
- *AIX 5L Version 5.3 System Management Concepts: Operating System and Devices*
- *AIX 5L Version 5.3 System Management Guide: Communications and Networks*
- *AIX 5L Version 5.3 Installation Guide and Reference*
- *AIX 5L Version 5.3 Commands Reference*
- *AIX 5L Version 5.3 Files Reference*
- *AIX 5L Version 5.3 General Programming Concepts: Writing and Debugging Programs*
- *AIX 5L Version 5.3 System User's Guide: Operating System and Devices*
- *AIX 5L Version 5.3 System User's Guide: Communications and Networks*
- *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*
- *AIX 5L Version 5.3 Guide to Printers and Printing*

Table des matières

Partie 1. Sécurité d’un système

La présente partie fournit des informations relatives à la sécurité d’un système, quelle que soit la connectivité réseau. Les chapitres qui suivent décrivent la procédure d’installation de votre système lorsque les options de sécurité sont activées, ainsi que les moyens d’empêcher des utilisateurs non autorisés d’accéder au système en configurant la sécurité d’AIX.

Chapitre 1. Installation et configuration d'un système sécurisé

Ce chapitre fournit des informations sur l'installation et la configuration d'un système sécurisé.

Il contient les sections suivantes :

La base TCB

L'administrateur système doit déterminer le niveau de sécurisation qui peut être accordé à un programme donné. Cette détermination prend en compte la valeur des ressources d'information du système en décidant du niveau de confiance nécessaire à l'installation d'un programme avec privilèges.

La base informatique sécurisée (TCB) est responsable de l'application des règles de sécurité du système. L'installation et l'utilisation de la base TCB permettent de définir l'accès utilisateur via un chemin d'accès sécurisé, assurant ainsi la communication sécurisée entre les utilisateurs et la base TCB. Les fonctions de la base TCB ne peuvent être activées qu'à l'installation du système d'exploitation. Pour installer la base TCB sur un poste déjà installé, il faudra effectuer une installation avec préservation. L'activation de la TCB donne accès au shell sécurisé, aux processus sécurisés et à la clé SAK (Secure Attention Key).

Cette section traite des points suivants :

Installation d'un système avec TCB

La base TCB est responsable de l'application des règles de sécurité du système. Tous les matériels de l'ordinateur sont inclus dans la TCB, mais l'administrateur système doit en premier lieu s'inquiéter des composants logiciels du TCB.

Si vous installez l'option TCB sur un système, vous activez le chemin et le shell sécurisés ainsi que le contrôle d'intégrité du système (commande **tcbck**). Ces fonctions peuvent être activées *uniquement* lors de l'installation du système d'exploitation de base (BOS). Si l'option TCB n'est pas sélectionnée lors de l'installation initiale, la commande **tcbck** est désactivée. Vous pouvez utiliser cette commande uniquement en réinstallant le système avec l'option TCB activée.

Pour définir l'option TCB lors de l'installation a BOS installation, sélectionnez **Options supplémentaires** à partir de l'écran Installation et paramètres. Dans cet écran, la valeur par défaut de **Installation TCB** est **no**. Pour activer la base TCB, tapez `2` puis appuyez sur Entrée.

Toutes les unités faisant partie de la base TCB, elle contrôle chaque fichier du répertoire **/dev**. De plus, la base TCB contrôle automatiquement plus de 600 fichiers supplémentaires et stocke les informations critiques les concernant dans le fichier **/etc/security/sysck.cfg**. Si vous installez la base TCB, sauvegardez ce fichier dès que l'installation est terminée, sur un support amovible comme une bande, un CD ou un disque, puis conservez le support en lieu sûr.

Vérification de la base TCB

Pour lancer l'audit de la sécurité de la base TCB, utilisez la commande **tcbck**. La sécurité du système d'exploitation est compromise dès lors que les fichiers TCB ne sont pas correctement protégés ou que les fichiers de configuration n'ont pas de valeurs sûres. La commande **tcbck** lance un audit du fichier **/etc/security/sysck.cfg** en le lisant. Ce fichier comporte une description de tous les fichiers TCB et de configuration, et de toutes les commandes sécurisées.

Le fichier **/etc/security/sysck.cfg** est en ligne et peut donc être modifié par un pirate informatique. Assurez–vous de créer une copie hors–ligne en lecture seule, après chaque mise à jour de la base TCB. Copiez également ce fichier depuis le support d'archives sur un disque avant d'effectuer tout contrôle.

L'installation de la base TCB et l'utilisation de la commande tcbck ne garantit pas que le système fonctionne dans un mode conforme CAPP (Controlled Access Protection Profile) ou EAL4+ (Evaluation Assurance Level 4+). Pour obtenir des informations sur les options CAPP/EAL4+, reportez–vous à la section CAPP (Controlled Access Protection Profile) et EAL4+ (Evaluation Assurance Level 4+), page 1-7.

Structure du fichier sysck.cfg

La commande **tcbck** lit le fichier **/etc/security/sysck.cfg** pour définir quels fichiers sont à vérifier. Tout programme sécurisé du système est décrit par une strophe du fichier **/etc/security/sysck.cfg**.

Voici les différents attributs de chaque strophe :

acl — Chaîne de texte représentant la liste de contrôle d'accès associée au fichier. Son format doit être le même que celui de la sortie de la commande **aclget**. Si la chaîne ne correspond pas à l'ACL du fichier, **sysck** applique cette valeur avec la commande **aclput**.

Remarque : Les attributs SUID, SGID et SVTX doivent correspondre à ceux spécifiés dans le mode (le cas échéant).

class — Nom d'un groupe de fichiers. Cet attribut permet de vérifier plusieurs fichiers du même nom de classe en indiquant un seul argument à la commande **tcbck**. Vous pouvez indiquer plusieurs classes (séparées par une virgule).

group — ID de groupe ou nom de groupe du fichier. Si la valeur ne correspond pas au propriétaire du fichier, la commande **tcbck** définit l'ID propriétaire sur cette valeur.

links — Liste de chemins d'accès (séparés par des virgules) associés à ce fichier. Le cas échéant, **tcbck** crée le lien de tout chemin d'accès de la liste qui ne serait pas associé au fichier. Sans le paramètre *tree*, la commande **tcbck** imprime un message indiquant la présence de liens supplémentaires (mais sans définir leurs noms). Avec le paramètre *tree*, la commande **tcbck** imprime également tout nom de chemin d'accès supplémentaire associé au fichier.

mode — Liste de valeurs (séparées par des virgules). Les valeurs acceptées sont SUID, SGID, SVTX et TCB. La dernière valeur doit représenter les autorisations du fichier, sous forme octale ou comme chaîne de 9 caractères. Par exemple **755** ou **rwxr–xr–x**. Si la valeur ne correspond pas au mode réel du fichier, **tcbck** applique la valeur correcte.

owner — ID utilisateur ou nom du propriétaire du fichier. Si la valeur ne correspond pas au propriétaire du fichier, la commande **tcbck** définit l'ID propriétaire sur cette valeur.

program	Liste de valeurs (séparées par des virgules). La première valeur est le chemin d'accès d'un programme de vérification. Les valeurs supplémentaires sont passées en arguments au programme lorsqu'il est exécuté. **Remarque :** Le premier argument est soit **–y**, **–n**, **–p** ou **–t**, selon l'indicateur utilisé avec la commande **tcbck**.
source	Nom du fichier source à utiliser pour générer une copie avant la vérification. Si la valeur n'est pas renseignée, et s'il s'agit d'un fichier ordinaire, un répertoire ou un tube nommé, une nouvelle version vide de ce fichier est créée (sauf s'il elle existe déjà). Pour les fichiers d'unités, un nouveau fichier spécial du même type d'unité est créé.
symlinks	Liste liens symboliques vers ce fichier, séparés par des virgules. Le cas échéant, **tcbck** crée le lien symbolique de tout chemin d'accès de la liste qui ne serait pas un lien symbolique. Avec le paramètre *tree*, la commande **tcbck** imprime également tout nom de chemin d'accès supplémentaire représentant un lien symbolique avec le fichier.

Si un ou plusieurs attributs manquent dans la strophe du fichier **/etc/security/sysck.cfg**, la vérification correspondante n'est pas effectuée.

Utilisation de la commande tcbck

La commande **tcbck** permet généralement de :

- Vérifier l'installation des fichiers relatifs à la sécurité
- Vérifier que l'arborescence du système de fichiers ne contient pas de fichiers violant la sécurité
- Mettre à jour, ajouter ou supprimer des fichiers sécurisés

La commande **tcbck** présente trois modes d'utilisation :

- Normal
  - non interactif à l'initialisation du système
  - avec la commande **cron**
- Interactif
  - Vérifier les fichiers individuels et les classes de fichiers
- Paranoïde
  - en enregistrant le fichier **sysck.cfg** hors ligne et en le restaurant périodiquement pour vérifier la machine

La base TCB est dépourvue de protection cryptographique, mais elle utilise la commande **sum** pour vérifier ses totaux de contrôle. La base de données TCB peut être définie manuellement avec une autre commande de somme de contrôle, par exemple la commande **md5sum** fournie dans le module RPM Package Manager **textutils** avec le CD *AIX Toolbox for Linux Applications*.

Vérification des fichiers sécurisés

Pour vérifier tous les fichiers de la base de données **tcbck**, rendre compte et corriger toutes les erreurs, entrez :

```
tcbck –y ALL
```

Cette action lance la commande **tcbck** pour vérifier l'installation de chaque fichier de la base de données **tcbck** décrit dans le fichier **/etc/security/sysck.cfg**.

Pour lancer cette commande automatiquement pendant l'initialisation du système et générer un journal d'erreurs éventuelles, ajoutez la chaîne ci–dessus au fichier **/etc/rc**.

Vérification de l'arborescence du système de fichiers

Si vous avez des doutes sur l'intégrité du système de fichiers, vous pouvez lancer à tout moment la commande **tcbck** pour vérifier l'arborescence :

```
tcbck –t tree
```

Avec le paramètre *tree*, cette commande vérifie l'installation de tous les fichiers du système (ce qui peut prendre un certain temps). Dans tout fichier trouvé risquant de compromettre la sécurité du système, vous pouvez modifier les attributs suspects. En outre, les vérifications suivantes sont effectuées sur tous les autres fichiers du système de fichiers :

- S'il s'agit d'un fichier avec un propriétaire root et avec le bit **SetUID** défini, ce dernier est effacé.
- S'il s'agit d'un fichier exécutable d'un groupe administratif, avec le bit **SetGID** défini, ce dernier est effacé.
- Si l'attribut **tcb** est défini pour le fichier, il est effacé.
- Si le fichier correspond à une unité (fichier spécial caractères ou blocs), il est supprimé.
- Si le fichier est un lien supplémentaire avec un chemin d'accès décrit dans **/etc/security/sysck.cfg**, ce lien est supprimé.
- Si le fichier est un lien symbolique avec un chemin d'accès décrit dans **/etc/security/sysck.cfg**, ce lien est supprimé.

  Remarque : Toutes les entrées d'unités doivent avoir été ajoutées à **/etc/security/sysck.cfg** avant l'exécution de la commande **tcbck** ; sinon, le système devient inutilisable. Pour ajouter des unités sécurisées au fichier **/etc/security/sysck.cfg**, utilisez l'indicateur **–l**.
  Attention : *Ne lancez pas* l'option **tcbck –y tree**. Cette option supprime et désactive les unités qui ne sont pas correctement répertoriées dans la base TCB, et peut donc désactiver votre système.

Ajout d'un programme sécurisé

Pour ajouter un programme spécifique au fichier **/etc/security/sysck.cfg**, entrez :

```
tcbck –a CheminAccès [ attribut = valeur ]
```

Seuls les attributs dont les valeurs ne sont pas déduites de l'état courant du fichier sont à spécifier dans la ligne de commande. Tous les noms d'attribut sont répertoriés dans le fichier **/etc/security/sysck.cfg**.

Par exemple, la commande suivante enregistre un nouveau programme root SetUID appelé **/usr/bin/setgroups**, possédant un lien nommé **/usr/bin/getgroups** :

```
tcbck –a /usr/bin/setgroups links=/usr/bin/getgroups
```

Pour ajouter, lors d'un audit de sécurité du fichier `/usr/bin/abc`, les utilisateurs administrateurs `jfh` et `jsl`, ainsi que le groupe administratif **developers**, entrez :

```
tcbck –a /usr/bin/abc setuids=jfh,jsl setgids=developers
```

Après l'installation d'un programme, vous ne savez pas nécessairement quels nouveaux fichiers sont enregistrés dans le fichier **/etc/security/sysck.cfg**. La commande suivante permet de les repérer et de les ajouter :

```
tcbck –t tree
```

Elle affiche le nom de tout fichier à enregistrer dans **/etc/security/sysck.cfg**.

Suppression d'un programme sécurisé

La suppression d'un fichier nécessite aussi celle de sa description dans le fichier **/etc/security/sysck.cfg** (si elle existe). Par exemple, si vous supprimez le programme **/etc/cvid**, la commande suivante génère l'affichage d'un message d'erreur :

```
tcbck -y ALL
```

Le message d'erreur généré se présente comme suit :

```
3001-020 The file /etc/cvid was not found.
```

La description de ce programme n'a pas été supprimée de **/etc/security/sysck.cfg**. Pour la supprimer, entrez la commande suivante :

```
tcbck -d /etc/cvid
```

Configuration des options supplémentaires de sécurité

Cette section contient des informations sur la configuration des options supplémentaires pour la base TCB.

Restriction d'accès au terminal

Les commandes **getty** et **shell** changent le propriétaire du terminal, pour empêcher l'accès au terminal de programmes non sécurisés. Le système d'exploitation permet de configurer l'accès exclusif au terminal.

Utilisation de la clé SAK

Attention : Soyez prudent avec SAK car cette clé élimine tous les processus qui tentent d'accéder au terminal et les liens associés (par exemple, **/dev/console** peut être lié à **/dev/tty0**).

Un chemin d'accès sécurisé des communications est généré par la séquence de touches SAK (Ctrl–X puis Ctrl–R). Pour sa mise en œuvre, tenez compte des éléments suivants :

- Lors de la connexion au système

  Après activation de la clé SAK :

  – si un nouvel écran de connexion s'affiche, vous disposez d'un chemin d'accès sécurisé.

  – si l'invite du shell sécurisé s'affiche, l'écran initial de connexion était un programme non autorisé dont le but était peut–être de voler votre mot de passe. Déterminez qui utilise actuellement ce terminal à l'aide de la commande **who** puis déconnectez–vous.

- Lorsque vous souhaitez que la commande que vous avez entrée génère un programme sécurisé. Voici quelques exemples :

  – en tant qu'utilisateur root. Ne passez en utilisateur root qu'après avoir établi un chemin d'accès sécurisé de communication. Cela empêchera l'exécution de programmes non sécurisés avec des droits d'accès root.

  – avec les commandes **su**, **passwd** et **newgrp**. N'exécutez ces commandes qu'après établissement d'un accès sécurisé.

Configuration de la clé SAK

Chaque terminal peut être configuré indépendamment pour qu'une pression de la clé SAK sur ce terminal crée un chemin d'accès sécurisé de communications. Ceci est déterminé par l'attribut **sak_enabled** dans le fichier **/etc/security/login.cfg**. Si la valeur de cet attribut est true, la clé SAK est activée.

Si vous utilisez un port de communication (par exemple, avec la commande **uucp**), la strophe de ce port, dans le fichier **/etc/security/login.cfg** comporte la ligne suivante :

```
sak_enabled = false
```

Cette ligne (ou l'absence d'entrée dans cette strophe) désactive la clé SAK de ce terminal.

Pour activer SAK sur un terminal, ajoutez la ligne suivante à sa strophe :

```
sak_enabled = true
```

CAPP (Controlled Access Protection Profile) et EAL4+ (Evaluation Assurance Level 4+)

A partir de la version AIX 5.2, les administrateurs système peuvent installer un système avec l'option CAPP et EAL4+ lors de l'installation du système d'exploitation de base (BOS). Cette option génère des restrictions sur le logiciel installé en même temps que le système d'exploitation de base, ainsi que sur l'accès au réseau.

Cette section traite des points suivants :

Présentation du système conforme CAPP/EAL4+

Un système CAPP a été conçu et configuré pour répondre au protocole CAPP (Controlled Access Protection Profile) sur l'évaluation de la sécurité selon des critères communs. Le CAPP définit des critères de fonctionnement identiques au précédent standard TCSEC C2 (également connu comme *Orange Book*).

Un Common Criteria (CC) Evaluated System est un système évalué selon les critères communs de la norme ISO 15408, relative à l'évaluation de l'assurance des produits informatiques. La configuration du système répondant à ces règles est appelé un *Système CAPP/EAL4+* dans le présent manuel.

L'évaluation d'un système selon les CC (critères communs) n'est valide que pour cette configuration (matérielle et logicielle). Toute modification de sa configuration de sécurité annule l'évaluation du système. Cela ne signifie pas nécessairement que la sécurité du système sera affectée, mais que la configuration n'est plus certifiée. Ni le CAPP ni le CC ne couvrent l'ensemble des options de configuration de sécurité possibles de AIX 5.2. Certaines fonctions, comme IPsec ou les modules de vérification de mots de passe personnalisés, ne sont pas inclues, mais peuvent être utilisées pour améliorer la sécurité du système.

Le système CAPP/EAL4+ AIX 5.2 comprend le système d'exploitation de base sur des processeurs POWER 3 et POWER 4 à 64 bits et :

- Le LVM (gestionnaire de volumes logiques) et le JFS2 (système de fichiers journalisé amélioré)
- Le système X–Windows avec l'interface CDE
- Les fonctions réseau Telnet, FTP, rlogin et rsh/rcp dans le protocole IPv4
- NFS (Network File System)

Un système CAPP/EAL4+ est considéré en état de sécurité dans les conditions suivantes :

- Si l'audit est configuré et que le système est en mode multi–utilisateurs, alors l'audit doit être opérationnel.
- Le système accepte les requêtes de connexion et de services réseau.
- Pour un système distribué, les bases de données administratives sont montées par NFS à partir du serveur maître.

Les interfaces administratives suivantes des fonctions de sécurité sont fournies :

- Mesures d'identification et d'authentification (configuration des utilisateurs, réglages de mot de passe, configuration de connexion, etc.)
- Mesures d'audit (configuration de l'audition en mode BIN, sélection d'événements audités, traitement de suivis d'audit, etc.)
- Contrôle d'accès discrétionnaire (bits d'autorisation et listes de contrôle des accès (ACL) pour les objets de système de fichiers, mécanismes IPC et ports TCP)
- Réglage de l'heure du système
- Exécution du sous–système de diagnostic **diag**
- Exécution de la commande **su** pour devenir un administrateur privilégié (root)

Ceci inclut les fichiers de configuration et les appels système qui peuvent être utilisés pour effectuer les tâches administratives appropriées.

Les interfaces utilisateur suivantes relatives aux fonctions de sécurité sont fournies :

- La commande **passwd** pour modifier le mot de passe d'un utilisateur
- La commande **su** pour modifier l'identité d'un utilisateur
- Les fonctions **at**, **batch** et **crontab** pour la planification du traitement des commandes
- Contrôle d'accès discrétionnaire (bits d'autorisation et listes de contrôle des accès (ACL) pour les objets de système de fichiers et mécanismes IPC)
- Mécanismes de connexion (par exemple, les mécanismes d'identification et d'authentification) pour la console système et les applications réseau prises en charge (comme **telnet** et **ftp**)

Ceci inclut les appels système ayant trait aux paramètres de l'identité utilisateur ou du contrôle d'accès.

Le système CAPP/EAL4+ AIX 5.2 fonctionne sur des plateformes matérielles basées sur les systèmes ESCALA PL SeriesSMP (Symmetric Multiprocessor) utilisant le processeur POWER3–II (ESCALA PL Series 610) avec un et deux processeurs, les systèmes SMP utilisant le processeur RS64 IV (ESCALA PL Series 660) et les systèmes SMP utilisant le processeur POWER4 (ESCALA PL Series 690). Les périphériques pris en charge sont les suivants : les terminaux et les imprimantes, les disques durs et les lecteurs de CD–Rom utilisés comme unités de stockage, ainsi que les lecteurs multimédia et les lecteurs de disquettes utilisés comme unités de sauvegarde. Les connecteurs réseau pris en charge sont de type Ethernet et Token Ring.

A compter de la version 5.2 d'AIX 5L avec le kit de maintenance recommandé 5200–01, la technologie CAPP/EAL4 fonctionne sur des plateformes matérielles dont le processeur POWER4 (ESCALA PL Series 630, ESCALA PL Series 650 et ESCALA PL Series 690) prend en charge la configuration de partitions logiques. Les périphériques pris en charge sont les suivants : les terminaux et les imprimantes, les disques durs et les lecteurs de CD–Rom utilisés comme unités de stockage, ainsi que les lecteurs multimédia et les lecteurs de disquettes utilisés comme unités de sauvegarde. Les connecteurs réseau pris en charge sont de type Ethernet et Token Ring.

Remarque : Les administrateurs doivent informer tous les utilisateurs du système de ne pas utiliser le fichier **$HOME/.rhosts** pour la connexion à distance et l'exécution de commandes.

Installation d'un système CAPP/EAL4+

Pour définir les options CAPP/EAL4+ lors d'une installation du système d'exploitation de base, procédez comme suit :

1. Dans l'écran Installation et paramètres, sélectionnez **Options supplémentaires**.
2. Dans l'écran Options supplémentaires, tapez le numéro correspondant au choix Oui ou Non pour l'option **Activation de la technologie CAPP et EAL4+**.
   Le réglage par défaut de cette option est `No` (Non).

Cette option n'est disponible que sous les conditions suivantes :

- La méthode d'installation est définie sur Installation avec remplacement total.
- L'anglais est sélectionné.
- Le noyau 64 bits est activé.
- Le JFS2 est activé.

Lorsque l'option **Activation de la technologie CAPP et EAL4+** est définie sur **oui**, l'option **Base informatique sécurisée** est également définie sur **oui** et les seuls choix disponibles de **Bureau** sont **NONE** (aucun) ou **CDE**.

Si vous effectuez une installation sans invite à l'aide d'un fichier **bosinst.data**, définissez la zone INSTALL_TYPE sur `CC_EVAL` et les zones suivantes comme suit :

```
control_flow:
         CONSOLE = ???
         PROMPT = yes
         INSTALL_TYPE = CC_EVAL
         INSTALL_METHOD = overwrite
         TCB = yes
         DESKTOP = NONE   or   CDE
         ENABLE_64BIT_KERNEL = yes
         CREATE_JFS2_FS = yes
         ALL_DEVICES_KERNELS = no
         NETSCAPE_BUNDLE = no
         HTTP_SERVER_BUNDLE = no
         KERBEROS_5_BUNDLE = no
         SERVER_BUNDLE = no
         ALT_DISK_INSTALL_BUNDLE = no

 locale:
         CULTURAL_CONVENTION = en_US   or   C
         MESSAGES = en_US   or   C
```

CAPP/EAL4+ et environnement NIM (Network Installation Management)

Des clients de technologie CAPP/EAL4+ peuvent être installés à l'aide de l'environnement NIM (Network Installation Management). Le maître NIM est configuré afin de fournir les ressources nécessaires à l'installation du niveau CAPP/EAL4+ approprié de AIX 5L. Les clients NIM peuvent être installés à l'aide des ressources situées sur le maître NIM. Vous pouvez effectuer une installation NIM du client sans invite en définissant les champs suivants dans la ressource **bosinst_data** :

```
control_flow:
         CONSOLE = ???
         PROMPT = no
         INSTALL_TYPE = CC_EVAL
         INSTALL_METHOD = overwrite
         TCB = yes
         DESKTOP = NONE   or   CDE
         ENABLE_64BIT_KERNEL = yes
         CREATE_JFS2_FS = yes
         ALL_DEVICES_KERNELS = no
         NETSCAPE_BUNDLE = no
         HTTP_SERVER_BUNDLE = no
         KERBEROS_5_BUNDLE = no
         SERVER_BUNDLE = no
         ALT_DISK_INSTALL_BUNDLE = no

 locale:
         CULTURAL_CONVENTION = en_US   or   C
         MESSAGES = en_US   or   C
```

Le maître NIM ne peut pas être configuré comme un système CAPP/EAL4+ et ne peut pas être connecté au même réseau avec d'autres systèmes CAPP/EAL4+. Lors du lancement de l'installation depuis le maître NIM, l'option de menu **Remain NIM client after install SMIT** doit être définie sur `no` (non). Après l'installation d'un client NIM en tant que système CAPP/EAL4+, le client NIM doit être supprimé du réseau du maître NIM et il n'est plus possible d'installer ou de mettre à jour des logiciels supplémentaires à l'aide du maître NIM.

Prenons par exemple deux environnements réseau ; le premier est composé d'un maître NIM et de systèmes non CAPP/EAL4+, le second uniquement de systèmes CAPP/EAL4+. Effectuez l'installation NIM sur le client NIM. Une fois l'installation terminée, déconnectez le système CAPP/EAL4+ nouvellement installé du réseau du maître NIM et connectez le système au réseau évalué.

Prenons ensuite un exemple comprenant un seul réseau. Le maître NIM n'est pas connecté au réseau lorsque d'autres systèmes fonctionnent dans la configuration évaluée et les systèmes CAPP/EAL4+ ne sont pas connectés au réseau pendant l'installation NIM.

Regroupement de logiciels CAPP/EAL4+

Lorsque l'option CAPP/EAL4+ est sélectionnée, les contenus du regroupement de l'installation **/usr/sys/inst.data/sys_bundles/CC_EVAL.BOS.autoi** sont installés.

L'option CAPP/EAL4+ permet d'installer les regroupements de logiciels graphiques et de services de documentation. Si vous sélectionnez les options Logiciels graphiques avec CAPP/EAL4+, le contenu du regroupement **/usr/sys/inst.data/sys_bundles/CC_EVAL.Graphics.bnd** est installé. Si vous sélectionnez les options Logiciels de services de documentation avec CAPP/EAL4+, le contenu du regroupement **/usr/sys/inst.data/sys_bundles/CC_EVAL.DocServices.bnd** est installé.

Dès que les LPP (Programmes sous licence) ont été installés, le système modifie la configuration par défaut pour respecter les règles CAPP/EAL4+. La configuration par défaut est modifiée comme suit :

- Suppression de **/dev/echo** du fichier **/etc/pse.conf**.
- Instanciation d'unités streams.
- L'accès aux supports amovibles est réservé aux utilisateurs root.

- Suppression des entrées non CC du fichier **inetd.conf**.
- Modification des droits d'accès à divers fichiers.
- Enregistrement de liens symboliques dans le fichier **sysck.cfg**.
- Enregistrement d'unités dans le fichier **sysck.cfg**.
- Définition des attributs de port et utilisateur par défaut.
- Configuration de l'application **doc_search** pour navigateur.
- Suppression de **httpdlite** du fichier **inittab**.
- Suppression de **writesrv** du fichier **inittab**.
- Suppression de **mkatmpvc** du fichier **inittab**.
- Suppression de **atmsvcd** du fichier **inittab**.
- Désactivation de **snmpd** dans le fichier **/etc/rc.tcpip**.
- Désactivation de **hostmibd** dans le fichier **/etc/rc.tcpip**.
- Désactivation de **snmpmibd** dans le fichier **/etc/rc.tcpip**.
- Désactivation de **aixmibd** dans le fichier **/etc/rc.tcpip**.
- Désactivation de **muxatmd** dans le fichier **/etc/rc.tcpip**.
- Le port NFS (2049) est doté de privilèges.
- Ajout d'événements manquants dans le fichier **/etc/security/audit/events**.
- Vérification du bon fonctionnement de l'interface de bouclage.
- Création de synonymes pour **/dev/console**.
- Application forcée des droits de connexion par défaut à un serveur X.
- Modification du répertoire **/var/docsearch** pour permettre une lecture universelle de tous les fichiers.
- Ajout de strophes ODM pour définir les droits d'accès à la console.
- Définition des droits 000 pour les ptys de type BSD.
- Désactivation des fichiers **.netrc**.
- Ajout du traitement du répertoire patch.

Interface utilisateur graphique

Le système compatible avec CAPP/EAL4+ inclut l'interface graphique X Windows. X Windows comporte un mécanisme d'affichage des applications clientes graphiques, tels que les horloges, les calculatrices, etc. ainsi que plusieurs sessions de terminal à l'aide de la commande **aixterm**. Le démarrage du système X Windows est effectué avec la commande **xinit** sur la ligne de commande initiale après la connexion de l'utilisateur sur la console de l'hôte.

Pour lancer une session X Windows, entrez :

```
xinit
```

Cette commande lance le serveur X Windows en activant les mécanismes d'accès locaux uniquement pour l'auteur de l'appel. Les clients X Windows dont l'UID est root peuvent accéder au serveur X Windows via le socket de domaine UNIX grâce à la priorité root sur les restrictions d'accès. Les clients X Windows dont l'UID est autre que root ou qui sont lancés par d'autres utilisateurs ne peuvent pas accéder au serveur X Windows. Cette restriction empêche les autres utilisateurs d'un hôte d'accéder au serveur X Windows sans y être autorisés.

Environnement physique d'un système CAPP/EAL4+

Le système CAPP/EAL4+ dispose de règles spécifiques concernant son environnement. Ces règles sont les suivantes :

- L'accès physique aux systèmes doit être restreint afin que les administrateurs autorisés soient les seuls à pouvoir en utiliser les consoles.
- Le Service Processor n'est pas connecté à un modem.
- L'accès physique aux terminaux est limité aux utilisateurs autorisés.
- Le réseau physique est protégé contre l'écoute électronique et les programmes espion (également appelés chevaux de Troie). Lors de communications via des lignes non sécurisées, des mesures de sécurité supplémentaires (comme le chiffrement) sont nécessaires.
- La communication n'est pas autorisée avec des systèmes autres que CAPP/EAL4+ AIX 5.2 ou qui n'utilisent pas le même contrôle de gestion.
- Seul l'IP version 4 sera utilisé pour communiquer avec des systèmes autres que CAPP/EAL4+ (IPv6 n'ayant pas été évalué).
- Les utilisateurs ne doivent pas être autorisés à modifier l'heure du système.
- Les systèmes dans un environnement LPAR ne peuvent pas partager les PHB.

Environnement organisationnel d'un système CAPP/EAL4+

Un système CAPP/EAL4+ doit répondre aux règles de procédure et d'organisation suivantes :

- L'administrateur doit avoir les compétences requises en la matière.
- L'administrateur est considéré comme digne de confiance.
- Seuls les utilisateurs autorisés à travailler avec les informations du système reçoivent des ID utilisateurs.
- Les utilisateurs doivent utiliser des mots de passe compliqués (aussi aléatoires que possible et sans lien direct avec l'utilisateur ou l'entreprise). Pour plus d'informations concernant les règles de définition de mots de passe, reportez–vous à la section Mots de passe, page 2-20.
- Les mots de passe sont personnels et confidentiels.
- Les administrateurs doivent avoir les connaissances suffisantes pour gérer les systèmes dont la sécurité est essentielle.
- Les administrateurs doivent se conformer aux directives fournies dans la documentation du système.
- Les administrateurs doivent se connecter avec leur propre ID et utiliser la commande **su–** pour passer en mode superutilisateur et effectuer l'administration du système.
- Les mots de passe utilisateurs générés par les administrateurs doivent être transmis en toute sécurité.
- Les responsables du système doivent établir et mettre en application les procédures nécessaires au fonctionnement sécurisé des systèmes.
- Les administrateurs doivent s'assurer que l'accès aux ressources du système sécurisé est protégé par les bits d'autorisation et la liste de contrôle des accès (ACL) appropriés.
- Le réseau physique doit être approuvé par l'entreprise pour le transport des données les plus confidentielles.
- Les procédures de maintenance doivent comprendre des diagnostics réguliers.
- Les administrateurs doivent disposer de procédures qui assurent un fonctionnement sécurisé et la reprise après une panne système.

- La variable d'environnement **LIBPATH** ne doit pas être modifiée au risque de permettre à un processus sécurisé de charger une bibliothèque non sécurisée.
- Les logiciels de traçage ou de dérivation (**tcpdump**, **trace**) ne doivent pas être utilisés sur un système opérationnel.
- Les protocoles anonymes tels que HTTP ne serviront que pour des informations publiques (par exemple, la documentation en ligne).
- Vous pouvez uniquement utiliser un NFS basé sur TCP.
- L'accès aux supports amovibles doit être interdit aux utilisateurs. Les fichiers d'unités doivent être protégés par des bits d'autorisation et des ACL appropriés.
- L'administration AIX se fait uniquement avec les droits d'accès root. Ni les fonctions de délégation/d'administration basées sur le rôle et sur le groupe, ni le mécanisme de privilège d'AIX ne sont inclus dans la conformité CAPP/EAL4+.
- Les administrateurs ne doivent pas utiliser le partitionnement dynamique pour allouer et désallouer des ressources. La configuration de la partition doit uniquement être effectuée lorsque aucune partition n'est en cours d'utilisation.

Environnement opérationnel d'un système CAPP/EAL4+

Un système CAPP/EAL4+ doit répondre aux exigences opérationnelles et aux procédures suivantes :

- Lorsqu'une console HMC (Hardware Management Console) est utilisée, elle est située dans un environnement contrôlé physiquement.
- Seul le personnel autorisé peut accéder à l'environnement opérationnel et à la console HMC.
- Lorsqu'une console HMC est utilisée, elle ne peut être employée que pour les tâches suivantes :
  - Configuration initiale des partitions. Une partition ne peut pas être active au cours d'un processus de configuration.
  - Redémarrage des partitions “en suspens”
- La console HMC ne doit pas être utilisée lors de l'exécution du système configuré.
- La fonction “call home” du système doit être désactivée.
- L'accès au système par modem distant doit être désactivé.
- Si AIX fonctionne dans un environnement sur lequel LPAR est activé, l'administrateur doit vérifier dans la documentation LPAR les exigences concernant le fonctionnement EAL4+ de partitions logiques.
- La fonction d'autorité de service doit être désactivé sur les partitions logiques.

Configuration d'un système CAPP/EAL4+

Cette section fournit des informations sur la configuration des sous–systèmes impliqués dans un système CAPP/EAL4+.

Administration

Les administrateurs doivent se connecter avec leur compte personnel et utiliser la commande **su** pour devenir l'utilisateur root afin d'administrer le système. Pour empêcher l'identification du mot de passe du compte root, vous devez uniquement permettre aux administrateurs autorisés d'utiliser la commande **su** sur ce compte. Veuillez procéder comme suit :

1. Ajoutez une entrée à la strophe **root** du fichier **/etc/security/user** :

```
root:
                admin = true
                .
                .
                .
                 sugroups = SUADMIN
```

2. Définissez un groupe dans le fichier **/etc/group** contenant uniquement les ID utilisateur des administrateurs autorisés :

```
system:!:0:root,paul
staff:!:1:invscout,julie
bin:!:2:root,bin
.
.
.
    SUADMIN:!:13:paul
```

Les administrateurs doivent également respecter les procédures suivantes :

- Etablir et mettre en application des procédures pour que les matériels, logiciels et microcodes qui composent le système distribué, soient partagés, installés et configurés en toute sécurité.
- S'assurer que le système est configuré pour que seul l'administrateur puisse y introduire un nouveau logiciel sécurisé.
- Mettre en place des procédures pour s'assurer que les utilisateurs effacent leur écran avant de se déconnecter des unités de connexion en série.

Configuration du port et de l'utilisateur

Les options de configuration AIX des ports et des utilisateurs sont définies pour satisfaire aux règles de l'évaluation. La règle actuelle est que la probabilité de deviner un mot de passe soit au plus 1 sur 1 000 000, et qu'une minute d'essais répétés ne donne qu'une probabilité de 1 sur 100 000.

Le fichier **/etc/security/user** figurant dans l'exemple suivant utilise la liste de dictionnaire **/usr/share/dict/words**. Le fichier **/usr/share/dict/words** figure dans l'ensemble de fichiers **bos.data**. Vous devez installer l'ensemble de fichiers **bos.data** avant de configurer le fichier **/etc/security/user**. Les valeurs recommandées du fichier **/etc/security/user** sont les suivantes :

```
default:
                admin = false
                login = true
                su = true
                daemon = true
                rlogin = true
                sugroups = ALL
                admgroups =
                ttys = ALL
                auth1 = SYSTEM
                auth2 = NONE
                tpath = nosak
                umask = 077
                expires = 0
                SYSTEM = "compat"
                logintimes =
                pwdwarntime = 5
                account_locked = false
                loginretries = 3
                histexpire = 52
                histsize = 20
                minage = 0
                maxage = 8
                maxexpired = 1
                minalpha = 2
                minother = 2
                minlen = 8
                mindiff = 4
                maxrepeats = 2
                dictionlist = /usr/share/dict/words
                pwdchecks =
                dce_export = false

 root:
                rlogin = false
                login = false
```

Les paramètres par défaut du fichier **/etc/security/user** ne doivent pas être écrasés pas des paramètres d'utilisateurs particuliers.

Remarque : Le réglage `login = false` dans la strophe root empêche toute connexion root directe. Seuls les comptes utilisateurs dotés des droits **su** pour le compte root pourront se connecter en tant que root. Cependant, un refus de service contre un système qui envoie des mots de passe utilisateur incorrects peut verrouiller tous les comptes utilisateurs. Cette action peut empêcher tous les utilisateurs (même administrateurs) de se connecter au système. Une fois son compte verrouillé, l'utilisateur ne pourra pas se connecter avant que l'administrateur ne repasse l'attribut `unsuccessful_login_count` correspondant, dans le fichier **/etc/security/lastlog**, à une valeur inférieure à l'attribut `loginretries`. Si tous les comptes administrateurs sont verrouillés, vous devrez redémarrer le système en mode maintenance et lancer la commande **chsec**. Pour plus d'informations sur l'utilisation de la commande **chsec**, reportez–vous à la section Contrôle des comptes utilisateurs, page 2-11.

Les valeurs recommandées du fichier **/etc/security/login.cfg** sont les suivantes :

```
default:
                sak_enabled = false
                logintimes =
                logindisable = 4
                logininterval = 60
                loginreenable = 30
                logindelay = 5
```

Limites de ressources

Lors de la configuration des limites de ressources dans le fichier **/etc/security/limits**, vérifiez que les limites correspondent aux besoins des processus du système. En particulier, les valeurs pour `stack` et `rss` ne doivent *jamais* être définies sur `unlimited`. Une pile illimitée risque d'écraser d'autres segments du processus, et une `rss` illimitée permet à un processus d'utiliser toute la mémoire réelle, pouvant alors créer des problèmes de ressource pour d'autres processus. Les valeurs `stack_hard` et `rss_hard` doivent également être limitées.

Sous–système d'audit

Les procédures suivantes permettent de protéger le sous–système d'audit :

- Configurer le sous–système d'audit pour enregistrer toutes les activités de sécurité des utilisateurs. Définir un système de fichiers dédié aux données d'audits, pour s'assurer que l'espace nécessaire à l'audit est disponible et ne sera pas utilisé par d'autres processus.
- Protéger les enregistrements d'audits (tels que des suivis d'audit, fichiers bin, et toutes les autres données dans **/audit**) contre les utilisateurs non–root.
- Pour le système CAPP/EAL4+, l'audit en mode **bin** doit être défini lorsque le sous–système est utilisé. Pour plus d'informations sur la procédure de définition du sous–système d'audit, reportez–vous à la section Configuration de l'audit, page 4-9.
- Le suivi d'audit requiert au moins 20 % de l'espace disque d'un système.
- Si l'audit est activé, le paramètre *binmode* de la strophe **start** dans **/etc/security/audit/config** doit avoir pour valeur **panic**. Le paramètre *freespace* de la strophe **bin** doit avoir une valeur au minimum égale à 25 % de l'espace disque alloué au stockage des suivis d'audit. Les paramètres *bytethreshold* et *binsize* doivent être définis sur 65536 octets.
- Archiver les enregistrements d'audit depuis le système vers un stockage permanent.

Configuration réseau

La configuration réseau doit utiliser le contrôle d'accès discrétionnaire aux ports Internet (DACinet) pour interdire l'utilisation anonyme des protocoles X (X11) et NFS. Pour plus d'informations sur la commande **dacinet**, reportez–vous à la section Contrôle d'accès aux ports TCP en fonction de l'utilisateur, avec le contrôle d'accès discrétionnaire aux ports Internet, page 10-10.

La commande **dacinet** permet d'éviter les situations suivantes :

- Empêche un utilisateur de prendre le bureau d'un autre avec X11.
- Empêche l'utilisateur d'un poste client de falsifier les requêtes vers un serveur NFS, ce qui lui permettrait de devenir utilisateur root. Généralement, un utilisateur accède à un serveur NFS distant en faisant ses requêtes au Système de fichiers logiques sur l'hôte local, qui transmet ensuite la requête (en tant que root) au serveur distant. En définissant un ACL réservé aux utilisateurs root et obligeant le passage par ce port, il sera impossible d'envoyer directement des requêtes de protocole au serveur NFS distant.

Services système

Le tableau suivant présente les services système standard, actifs sur le système CAPP/EAL4+ (sans carte graphique).

Tableau 1. Services système standard

UID	Commande	Description
root	/etc/init	processus Init
root	/usr/sbin/syncd 60	Démon sync du système de fichiers
root	/usr/sbin/srcmstr	Démon maître SRC
root	/usr/sbin/cron	Fonction CRON avec support AT
root	/usr/ccs/bin/shlap64	Démon support de bibliothèque partagée
root	/usr/sbin/syslogd	Démon Syslog
root	/usr/lib/errdemon	Démon AIX error log
root	/usr/sbin/getty /dev/console	getty / TSM
root	/usr/sbin/portmap	Mappeur de port pour NFS et CDE
root	/usr/sbin/biod 6	Client NFS
root	/usr/sbin/rpc.lockd	Démon de verrou NFS
démon	/usr/sbin/rpc.statd	Démon stat NFS
root	/usr/sbin/rpc.mountd	Démon de montage NFS
root	/usr/sbin/nfsd	Démon serveur NFS
root	/usr/sbin/inetd	Démon maître Inetd
root	/usr/sbin/uprintfd	Démon print du noyau
root	/usr/sbin/qdaemon	Démon Queuing
root	/usr/lpp/diagnostics/bin/diagd	Diagnostics

Mise en route d'un système distribué CAPP/EAL4+

Pour lancer un système distribué conforme au protocole CAPP/EAL4+, tous les utilisateurs doivent avoir des ID identiques sur tous les systèmes. Bien que ceci soit possible grâce à NIS, le niveau de sécurisation est insuffisant pour CAPP/EAL4+. Cette section décrit une configuration distribuée permettant de garantir des ID utilisateurs identiques sur tous les systèmes CAPP/EAL4+.

Le système maître conserve les données d'identification et d'authentification (configurations utilisateur et groupe) pour tout le système distribué. Tous les autres systèmes utilisent NFS pour monter ces données. NFS est protégé par DACinet de sorte que seuls les administrateurs aient accès aux ports NFS sur le poste maître.

Les données d'authentification sont modifiables par tous les administrateurs à l'aide d'outils tels que SMIT, sur n'importe quel système. Les données modifiées sont celles du poste maître.

Toutes les données partagées d'identification et d'authentification proviennent du répertoire **/etc/data.shared**. Les fichiers standard d'identification et d'authentification sont remplacés par des liens symboliques dans le répertoire **/etc/data.shared**.

Fichiers partagés du système distribué

Dans un système distribué, les fichiers suivants sont partagés. Ils proviennent habituellement du répertoire **/etc/security**.

/etc/group Le fichier **/etc/group**

/etc/hosts Le fichier **/etc/hosts**

/etc/passwd Le fichier **/etc/passwd**

/etc/security/.ids L'utilisateur disponible suivant et l'ID groupe

/etc/security/.profile
Le fichier par défaut **.profile** destiné aux nouveaux utilisateurs

/etc/security/acl Le fichier **/etc/security/acl** conserve les définitions ACL de l'ensemble du système des services protégés qui seront réactivés lors du prochain amorçage du système via le fichier **/etc/rc.tcpip**.

/etc/security/audit/bincmds
Commandes d'audit en mode BIN pour cet hôte

/etc/security/audit/config
Configuration d'audit local

/etc/security/audit/events
Liste des formats et des événements d'audit

/etc/security/audit/objects
Liste d'objets audités sur cet hôte

/etc/security/audit/streamcmds
Commandes d'audit en mode STREAM pour cet hôte

/etc/security/environ
Variables d'environnement par utilisateur

/etc/security/group
Informations de groupe étendues depuis le fichier **/etc/security/group**

/etc/security/limits
Limites de ressource par utilisateur

/etc/security/passwd
Mots de passe par utilisateur

/etc/security/priv
Les ports désignés en tant que privilégiés au démarrage du système sont répertoriés dans le fichier **/etc/security/priv**

/etc/security/services
Les ports répertoriés dans le fichier **/etc/security/services** ne sont pas soumis aux contrôles ACL

/etc/security/user
Attributs par utilisateur et de l'utilisateur par défaut

Fichiers non–partagés du système distribué

Les fichiers suivants, du répertoire **/etc/security**, ne doivent pas être partagés sur le système distribué, et doivent rester spécifiques à l'hôte :

/etc/security/failedlogin
Fichier journal pour les échecs de connexion par hôte

/etc/security/lastlog
Informations par utilisateur concernant les dernières connexions réussies et échouées sur cet hôte

/etc/security/login.cfg
: Caractéristiques de connexion spécifique à l'hôte pour le chemin d'accès sécurisé, shells de connexion, et autres informations relatives à la connexion

/etc/security/portlog
: Informations par port concernant les ports verrouillés sur cet hôte

Les fichiers de sauvegarde des fichiers partagés, générés automatiquement, sont également non–partagés. Ces fichiers ont le même nom que le fichier original, avec un `o` minuscule ajouté au début.

Configuration du système distribué (le système maître)

Sur le poste maître, un nouveau volume logique est créé pour contenir le système de fichiers des données d'identification et d'authentification. Ce volume logique est appelé **/dev/hd10sec**. Il est monté sur le système maître en tant que **/etc/data.master**. Pour générer les modifications sur le système maître, lancez la commande **mkCCadmin** avec l'adresse IP et le nom d'hôte du poste maître :

```
mkCCadmin –m –a   ipaddress hostname
```

Configuration du système distribué (tous les systèmes)

Toutes les données à partager sont transférées dans le répertoire **/etc/data.shared**. Au démarrage, tous les systèmes montent le répertoire **/etc/data.master** du poste maître sur le répertoire **/etc/data.shared**. Le maître lui–même utilise un montage de bouclage.

Les systèmes client sont configurés avec la commande suivante :

```
mkCCadmin –a   ipaddress hostname
```

Pour que le poste client utilise un poste maître différent, utilisez la commande **chCCadmin**.

Lorsque qu'un système a été intégré dans le système distribué d'identification et d'authentification, les entrées **inittab** supplémentaires suivantes sont générées :

`isCChost` Initialise le système en mode CAPP/EAL4+.

`rcCC` Efface tous les ACL DACinet et ouvre uniquement les ports nécessaires au mappeur de port et au NFS. Il monte ensuite le répertoire partagé.

`rcdacinet` Charge les ACL DACinet supplémentaires que l'administrateur pourrait avoir définis.

Tenez compte des points suivants lorsque le système distribué est en cours de fonctionnement :

- Les administrateurs doivent s'assurer que les données partagées sont montées avant de modifier les fichiers partagés de configuration, afin de garantir que les données son visibles pour tous les systèmes.
- La modification du mot de passe est la seule action administrative permise lorsque le répertoire partagé n'est pas monté.

Utilisation de la fonction DACinet pour le contrôle d'accès au réseau en fonction des utilisateurs et des ports

La fonction DACinet permet de limiter l'accès des utilisateurs aux ports TCP. Pour plus d'informations sur DACinet, reportez–vous à la section Contrôle d'accès aux ports TCP en fonction de l'utilisateur, avec le contrôle d'accès discrétionnaire aux ports Internet, page 10-10. Par exemple, lorsque vous utilisez la fonction DACinet pour limiter à root l'accès au port TCP/25 entrant, seuls les utilisateurs root des hôtes CAPP/EAL4+ peuvent y accéder.
Cette situation limite les possibilités d'usurpation d'e–mail par des utilisateurs standard, à l'aide d'une connexion **telnet** sur le port TCP/25 de la victime.

Pour activer les ACL pour les connexions TCP à l'amorçage, le script **/etc/rc.dacinet** est lancé à partir de **/etc/inittab**. Il ira lire les définitions dans le fichier **/etc/security/acl** puis chargera les ACL dans le noyau. Les ports qui ne doivent pas faire l'objet d'une protection par ACL doivent figurer dans le fichier **/etc/security/services** qui utilise le même format que le fichier **/etc/services**.

Par exemple, pour un sous–réseau 10.1.1.0/24 pour tous les systèmes connectés, les entrées ACL destinées à limiter l'accès aux utilisateurs root pour X (TCP/6000) dans le fichier **/etc/security/acl** seraient :

```
6000    10.1.1.0/24 u:root
```

Installation de logiciels supplémentaires sur un système CAPP/EAL4+

L'administrateur peut installer des logiciels supplémentaires sur le système CAPP/EAL4+. Si le logiciel n'est pas exécuté par l'utilisateur root ou avec les privilèges de l'utilisateur root, le système reste néanmoins conforme au protocole CAPP/EAL4+. C'est par exemple le cas pour des applications de bureautique, exécutées uniquement par des utilisateurs courants, sans composants SUID.

Par contre, les logiciels installés fonctionnant avec des privilèges root annulent la conformité CAPP/EAL4+. Cela signifie par exemple que les pilotes de l'ancien JFS (système de fichiers journalisé) ne doivent pas être installés puisqu'ils fonctionnent en mode noyau. D'autres démons fonctionnant en root (par exemple, un démon SNMP) annuleront également la conformité CAPP/EAL4+.

Un système CAPP/EAL4+ est rarement utilisé dans la configuration qui a été évaluée, particulièrement en environnement commercial. Des services supplémentaires sont généralement nécessaires, et le système de production, bien que basé sur un système évalué, ne répond plus exactement à ses spécifications.

Fenêtre de connexion

Les pirates informatiques peuvent obtenir des informations importantes à partir de l'écran de connexion AIX par défaut, comme le nom de l'hôte et la version du système d'exploitation. Ces informations peuvent leur permettre de déterminer les méthodes d'intrusion à essayer. Vous modifierez donc les paramètres par défaut de l'écran de connexion, le plus rapidement possible après une installation système. Cette section traite des points suivants :

Les bureaux KDE et GNOME partagent quelques règles identiques de sécurité. Pour plus d'information concernant KDE et GNOME, reportez–vous au manuel *AIX5L Version 5.3 Références et guide d'installation*.

Pour plus d'informations sur les utilisateurs, groupes et mots de passe, voir Utilisateurs, rôles et mots de passe, page 2-1.

Configuration de la fenêtre de connexion

Pour renforcer la protection du système face aux attaques par identification du mot de passe, vous pouvez configurer la fenêtre de connexion dans le fichier **/etc/security/login.cfg** de la manière suivante :

Tableau 2. Attributs de configuration de la fenêtre de connexion et valeurs recommandées

Attribut	S'applique aux PtY (réseau)	S'applique aux TTY	Valeur recommandée	Remarques
sak_enabled	Y	Y	false	La clé SAK est rarement nécessaire. Reportez–vous à la section Utilisation de la clé SAK (Secure Attention Key), page 1-6.
logintimes	N	Y		Indiquer ici les heures de connexions autorisées.
logindisable	N	Y	4	Désactiver la connexion sur ce terminal après 4 échecs de connexion consécutifs.
logininterval	N	Y	60	Le terminal sera désactivé lorsque que le nombre spécifié de tentatives aura été effectué en l'espace de 60 secondes.

Attribut	S'applique aux PtY (réseau)	S'applique aux TTY	Valeur recommandée	Remarques
loginreenable	N	Y	30	Le terminal sera réactivé 30 minutes après une désactivation automatique.
logindelay	Y	Y	5	Temps en seconde entre les invites de connexion. Il sera multiplié par le nombre d'échecs de connexion, par exemple 5, 10, 15, 20 secondes quand 5 est la valeur initiale.

Vous devez noter que ces restrictions de port fonctionnent généralement sur des terminaux de série reliés directement, et non sur des pseudo–terminaux utilisés via des connexions réseau. Vous pouvez indiquer des terminaux explicites dans ce fichier, par exemple :

```
/dev/tty0:
         logintimes = 0600-2200
         logindisable = 5
         logininterval = 80
         loginreenable = 20
```

Modification du message d'accueil de l'écran de connexion

Pour éviter d'afficher certaines informations sur les écrans de connexion, modifiez le paramètre *herald* dans le fichier **/etc/security/login.cfg**. Par défaut, *herald* contient le message d'accueil qui s'affiche avec l'invite de connexion. Pour le modifier, utilisez la commande **chsec** ou modifiez directement le fichier.

La commande **chsec** est utilisée dans l'exemple suivant pour modifier le paramètre par défaut *herald* :

```
# chsec –f /etc/security/login.cfg –a default –herald
 ″L'accès non autorisé est interdit.\n\nlogin: ″ ″
```

Pour plus d'informations sur la commande **chsec**, reportez–vous au manuel *AIX 5L Version 5.3 Commands Reference, Volume 1*.

Pour une modification directe du fichier, ouvrez le fichier **/etc/security/login.cfg** puis modifiez le paramètre *herald* de cette manière :

```
default:
 herald =″L'accès non autorisé est interdit\n\nlogin:″
  sak_enable = false
  logintimes =
  logindisable = 0
  logininterval = 0
  loginreenable = 0
  logindelay = 0
```

Remarque : Pour renforcer la sécurité du système, définissez les variables **logindisable** et **logindelay** sur une valeur supérieure à `0` (`# > 0`).

Modification de l'écran de connexion dans l'environnement Common Desktop

Ce problème concerne également les utilisateurs de l'environnement Common Desktop (CDE). L'écran de connexion CDE affiche également par défaut, le nom d'hôte et la version du système d'exploitation. Pour éviter d'afficher ces informations, modifiez le fichier **/usr/dt/config/$LANG/Xresources**, où **$LANG** se réfère à la langue locale installée sur votre poste.

Dans notre exemple, supposons que **$LANG** soit défini sur **C**, copiez ce fichier dans le répertoire **/etc/dt/config/C/Xresources**. Ensuite, ouvrez le fichier

/usr/dt/config/C/Xresources puis modifiez–le en supprimant les messages d'accueil qui comportent le nom d'hôte et la version du système d'exploitation.

Pour plus d'informations sur les questions de sécurité du CDE,
reportez–vous à la section Gestion des problèmes sous X11 et CDE, page 1-24.

Désactivation de l'affichage du nom d'utilisateur et de l'invite de modification du mot de passe

Dans un environnement sécurisé, il peut être utile de masquer le nom d'utilisateur dans la fenêtre de connexion et de personnaliser l'invite du mot de passe. Par défaut, l'invite de connexion et de demande du mot de passe est la suivante :

```
login: foo
foo's Password:
```

Pour désactiver l'affichage du nom d'utilisateur des invites et des messages d'erreur système, modifiez le paramètre *usernameecho* dans le fichier **/etc/security/login.cfg**. La valeur par défaut de *usernameecho* étant "true", le nom d'utilisateur s'affiche. Pour modifier ce paramètre, utilisez la commande **chsec** ou modifiez directement le fichier.

Dans l'exemple suivant, la commande **chsec** modifie le paramètre par défaut *usernameecho* :

```
# chsec –f /etc/security/login.cfg –s default –a usernameecho=false
```

Pour plus d'informations sur la commande **chsec**, reportez–vous au manuel *AIX 5L Version 5.3 Commands Reference, Volume 1*.

Pour modifier le paramètre directement dans le fichier **/etc/security/login.cfg**, ouvrez ce fichier puis modifiez le paramètre *usernameecho* de la manière suivante :

```
default:
  usernamecho = false
```

Si vous attribuez la valeur "false" au paramètre *usernameecho*, le nom d'utilisateur ne sera pas affiché à l'invite de connexion. Le nom d'utilisateur sera masqué par des caractères '*' sur les invites et les messages d'erreur du système, comme dans l'exemple suivant :

```
login:
 ***'s Password:
```

Vous pouvez également personnaliser l'invite du mot de passe en configurant le paramètre *pwdprompt* dans le fichier **/etc/security/login.cfg**. La valeur par défaut est " *user* 's Password: " où *user* est remplacé par le nom de l'utilisateur.

Pour modifier ce paramètre, utilisez la commande **chsec** ou modifiez directement le fichier.

Dans l'exemple suivant, la commande **chsec** modifie le paramètre par défaut *pwdprompt* afin d'obtenir "Password: ":

```
# chsec –f /etc/security/login.cfg –s default –a pwdprompt="Password: "
```

Pour modifier le paramètre directement dans le fichier **/etc/security/login.cfg**, ouvrez ce fichier puis modifiez le paramètre *pwdprompt* de la manière suivante :

```
default:
pwdprompt = "Password: "
```

Si vous attribuez au paramètre *pwdprompt* la valeur "Password: ", la valeur définie sera affichée dans la fenêtre de connexion mais également dans les applications qui utilisent l'invite du mot de passe système. Après la définition d'une invite personnalisée, l'invite de connexion fonctionne de la manière suivante :

```
login: foo
Password:
```

Configuration des paramètres de connexion par défaut

Editez le fichier **/etc/security/login.cfg** pour configurer les paramètres de connexion par défaut, comme ceux que vous pourriez définir pour un nouvel utilisateur (nombre de tentatives de connexion, réactivation de la connexion et connexion interne).

Protection de terminaux sans surveillance

Tous les systèmes sont vulnérables si les terminaux connectés sont laissés sans surveillance. Le plus grave étant un administrateur qui abandonne son terminal, connecté sous l'identité de l'utilisateur root. En règle générale, les utilisateurs doivent se déconnecter avant de quitter leur terminal. Un terminal connecté sans surveillance est un risque majeur. La commande **lock** permet de verrouiller votre terminal. Avec l'interface AIXwindows, utilisez la commande **xlock**.

Application de la déconnexion automatique

Un autre sérieux problème de sécurité peut survenir lorsque des utilisateurs laissent leurs comptes sans surveillance pendant une longue période. Un intrus peut profiter de cette situation pour prendre le contrôle d'un terminal utilisateur, compromettant ainsi la sécurité du système.

Pour éviter ce type de risque, vous pouvez activer une déconnexion automatique du système. Pour ce faire, éditez le fichier **/etc/security/.profile** pour y intégrer une valeur de déconnexion automatique pour *tous* les utilisateurs, comme dans l'exemple suivant :

```
TMOUT=600 ; TIMEOUT=600 ; export readonly TMOUT TIMEOUT
```

Dans cet exemple, 600 est en secondes, soit 10 minutes. Cependant, cette méthode ne fonctionnera qu'à partir du shell.

Même si cette action permet d'appliquer une politique de déconnexion automatique à tous les utilisateurs, il peuvent néanmoins contourner certaines restrictions en modifiant leurs propres fichiers **.profile**. Pour mettre en œuvre une politique de déconnexion automatique complète, fournissez aux utilisateurs des fichiers **.profile** appropriés, dépourvus de droits d'écriture.

Gestion des problèmes sous X11 et CDE

Cette section traite des failles dans la sécurité du serveur X11 X et de l'environnement Common Desktop (CDE).

Suppression du fichier /etc/rc.dt

Bien que l'interface CDE soit pratique, elle apporte son lot de problèmes de sécurité. Pour cette raison, n'exécutez pas le CDE sur des serveurs qui exigent un niveau élevé de sécurité. Le meilleur moyen est de ne pas installer les ensembles de fichiers CDE (dt). Si vous les avez installés sur votre système, nous vous conseillons de les désinstaller, et tout particulièrement le script **/etc/rc.dt**, qui démarre le CDE.

Pour plus d'informations sur les instructions associées aux tâches, reportez–vous au document *AIX 5L Version 5.3 – Guide de gestion du système : Système d'exploitation et unités*.

Précautions à prendre pour éviter le contrôle non autorisé des serveurs X distants

Un problème important de sécurité associé au serveur X11 est une écoute discrète et non autorisée d'un serveur distant. Les commandes **xwd** et **xwud** permettent de contrôler l'activité d'un serveur X car elles peuvent capturer la frappe, ce qui permet de connaître les mots de passe et autres données confidentielles. Pour résoudre ce problème, supprimez ces exécutables s'ils ne sont pas indispensables à votre configuration, ou bien limitez ces commandes à un accès root.

Les commandes **xwd** et **xwud** se trouvent dans l'ensemble de fichiers **X11.apps.clients**.

Si vous devez les conserver, utilisez OpenSSH ou MIT Magic Cookies. Ces applications tierces facilitent la prévention des risques créés par l'exécution des commandes **xwd** et **xwud**.

Pour plus d'informations sur OpenSSH et MIT Magic Cookies, reportez–vous à leurs documentations respectives.

Activation et désactivation du contrôle d'accès

Le serveur X permet aux hôtes distants d'utiliser la commande **xhost +** pour se connecter à votre système. Assurez–vous d'indiquer un nom d'hôte à la commande **xhost +**, car elle désactive le contrôle d'accès du serveur X. Cela permet d'autoriser l'accès à des hôtes spécifiques et facilite le contrôle des attaques potentielles du serveur X. Pour ce faire, lancez la commande **xhost** :

```
# xhost +   hostname
```

Si vous n'indiquez aucun nom d'hôte, l'accès sera accordé à tous les hôtes.

Pour plus d'informations sur la commande **xhost**, reportez–vous à la section *AIX Commands Reference, Volume 6.*

Désactivation des droits utilisateurs sur la commande xhost

Une autre manière de s'assurer de l'utilisation correcte de la commande **xhost** est de la limiter à l'utilisateur root. Pour ce faire, utilisez la commande **chmod** pour modifier les droits d'accès de **/usr/bin/X11/xhost** sur 744 :

```
chmod 744/usr/bin/X11/xhost
```

Chapitre 2. Utilisateurs, rôles et mots de passe

Ce chapitre décrit les méthodes de gestion des rôles et utilisateurs AIX.
Il traite des points suivants :

Compte root

Le compte root dispose d'un accès illimité à tous les programmes, fichiers et ressources d'un système. Le compte root est l'utilisateur spécial dans le fichier **/etc/passwd** avec l'ID utilisateur (UID) `0`. On lui attribue généralement le nom d'utilisateur *root* (root). Ce n'est donc pas le nom d'utilisateur qui rend le compte root si spécial, mais la valeur de son UID égale à `0`. Cela signifie que tout utilisateur possédant un UID égal à `0` possède également les mêmes droits que l'utilisateur root. Par ailleurs, le compte root est toujours authentifié au moyen des fichiers locaux de sécurité.

Le compte root doit toujours posséder un mot de passe, lequel ne doit jamais être partagé. Un mot de passe doit être attribué au compte root immédiatement après l'installation du système. Seul l'administrateur système doit connaître le mot de passe root. Les administrateurs système ne doivent utiliser leurs droits d'utilisateur root uniquement pour effectuer des fonctions d'administration exigeant de tels droits. Pour toutes les autres opérations, ils doivent reprendre leur compte utilisateur normal.

Attention : L'utilisation continue des droits d'utilisateur root peut endommager le système car le compte root s'affranchit de nombreuses sécurités dans le système.

Désactivation de la connexion root directe

L'une des méthodes courantes d'attaque employée par des pirates informatiques est d'obtenir le mot de passe de l'utilisateur root.

Pour éviter ce type d'attaque, vous pouvez désactiver l'accès direct à votre ID root et exiger de vos administrateurs système qu'ils obtiennent des droits d'accès root à l'aide de la commande **su –**. Restreindre l'accès root direct vous permet non seulement de supprimer l'utilisateur root comme cible d'attaque, mais également de contrôler quel utilisateur a obtenu un accès root, ainsi que la durée de son action. Pour ce faire, vous pouvez consulter le fichier **/var/adm/sulog**. Vous pouvez aussi activer la fonction d'audit du système, qui rendra compte de ce type d'activité.

Pour désactiver l'accès distant à la connexion pour votre utilisateur root, modifiez le fichier **/etc/security/user**. Indiquez **false** pour rlogin dans l'entrée de root.

Avant de désactiver la connexion root distante, examinez et prévoyez les situations qui empêcheraient un administrateur système de se connecter via un ID utilisateur non root. Par exemple, si un système de fichiers home d'un utilisateur est complet, l'utilisateur ne peut pas se connecter. Si la connexion root distante est désactivée et qu'un utilisateur pouvant obtenir l'accès à des droits root à l'aide de la commande **su –** dispose d'un système de fichiers home complet, l'utilisateur root ne pourra plus prendre le contrôle du système. Pour éviter ce problème, les administrateurs système peuvent se créer leurs propres systèmes de fichiers principaux disposant d'une capacité supérieure au système de fichiers standard.

Pour plus d'informations sur la gestion de la connexion root, reportez–vous à la section Configuration d'un système CAPP/EAL4+, page 1-14.

Rôles administratifs

Vous pouvez attribuer certains des droits d'accès root à des utilisateurs non–root. Les diverses tâches root disposent d'autorisations distinctes, groupées par rôles. Ce sont ces rôles qui peuvent être attribués à divers utilisateurs.

Cette section traite des points suivants :

Présentation des rôles

Les rôles sont des autorisations qui permettent à un utilisateur d'exécuter des fonctions qui, normalement, exigent des droits d'accès root. Voici la liste des rôles possibles :

Ajout et suppression d'utilisateurs	Permet à tout utilisateur de tenir ce rôle root. Il peut ajouter et supprimer des utilisateurs, modifier les informations ou classes d'audit, gérer des groupes et modifier des mots de passe. Toute personne qui administre les utilisateurs doit être dans le groupe **security**.
Modification du mot de passe des utilisateurs	Permet à un utilisateur de modifier les mots de passe.
Gestion des rôles	Permet à un utilisateur de créer, modifier, supprimer et répertorier les rôles. L'utilisateur doit être dans le groupe **security**.
Sauvegarde et restauration	Permet à un utilisateur de sauvegarder et restaurer des systèmes de fichiers et des répertoires. Ce rôle ne permet pas de sauvegarder ni de restaurer le système à l'aide de la commande mksysb et exige des autorisations appropriées.
Sauvegarde uniquement	Ne permet à un utilisateur que de sauvegarder des systèmes de fichiers et des répertoires. Cet utilisateur doit posséder l'autorisation adéquate pour pouvoir activer la sauvegarde système.
Exécution de diagnostics	Permet à un utilisateur ou un technicien de maintenance d'exécuter des diagnostics et de diagnostiquer des tâches. L'utilisateur doit avoir l'option **system** comme groupe principal, ainsi qu'un ensemble de groupes comprenant **shutdown**. **Remarque :** Les utilisateurs du rôle Exécution de diagnostics peuvent modifier la configuration du système, mettre à jour le microcode, etc. Les utilisateurs de ce rôle doivent bien mesurer le niveau de responsabilité de ce rôle.
Procédure d'arrêt du système	Permet à un utilisateur d'arrêter, de redémarrer ou de suspendre un système. Tout utilisateur qui utilise ce rôle doit être associé à un groupe contenant **shutdown**.

Configuration et maintenance des rôles à l'aide de SMIT

Les raccourcis SMIT suivants sont disponibles pour l'implémentation et la maintenance des rôles :

Tableau 3. Configuration et maintenance des rôles

Tâche	**Raccourci SMIT**
Ajout d'un rôle	**smit mkrole**
Modification des caractéristiques d'un rôle	**smit chrole**
Affichage des caractéristiques d'un rôle	**smit lsrole**
Retrait d'un rôle	**smit rmrole**
Liste des rôles	**smit lsrole**

Comprendre les autorisations

Les autorisations sont les attributs des droits d'accès d'un utilisateur. Elles permettent à un utilisateur d'exécuter certaines tâches. Les différents types d'autorisation sont :

Autorisation principale
: Permet d'exécuter une commande spécifique. Par exemple, l'autorisation RoleAdmin est une autorisation principale permettant à l'administrateur d'exécuter la commande **chrole**. Sans cette autorisation, la commande s'interrompt sans avoir modifié les définitions du rôle.

Modificateur d'autorisation
: Augmente les droits d'un utilisateur. Par exemple, l'autorisation UserAdmin augmente les capacités d'un administrateur appartenant au groupe **security**. Sans cette autorisation, la commande **mkuser** ne crée que des utilisateurs non administratifs. Avec cette autorisation, la commande **mkuser** crée également des utilisateurs administratifs.

Correspondance entre les autorisations et les fonctions :

Backup
: Effectue une sauvegarde système. La commande suivante utilise l'autorisation Backup :

  Backup — Sauvegarde des fichiers et des systèmes de fichiers. L'administrateur doit posséder l'autorisation Backup.

Diagnostics
: Permet à un utilisateur d'exécuter des diagnostics. Ce droit d'accès est également obligatoire pour exécuter des tâches de diagnostic directement depuis la ligne de commande. La commande suivante utilise l'autorisation Diagnostics :

  diag — Exécute des diagnostics sur des ressources sélectionnées. Si l'administrateur ne possède pas de droits d'accès Diagnostics, la commande ne s'exécute pas.

GroupAdmin
: Exécute les fonctions de l'utilisateur root sur les données du groupe. Les commandes suivantes utilisent l'autorisation GroupAdmin :

  chgroup — Modifie les informations d'un groupe. Si l'utilisateur ne possède pas d'autorisation GroupAdmin, il ne peut modifier que les informations d'un groupe non administratif.

  chgrpmem — Administre tous les groupes. Si l'administrateur d'un groupe ne possède pas d'autorisation GroupAdmin, il ne peut modifier que l'appartenance au groupe qu'il gère ou définit un utilisateur du groupe security pour gérer un groupe non administratif.

  chsec — Modifie les données d'un groupe administratif dans les fichiers **/etc/group** et **/etc/security/group**. L'utilisateur peut également modifier les **valeurs de strophe** par défaut. Si l'utilisateur ne possède pas d'autorisation GroupAdmin, il ne peut modifier que les données d'un groupe non administratif dans les fichiers **/etc/group** et **/etc/security/group**.

  mkgroup — Crée un groupe. Si l'utilisateur ne possède pas d'autorisation GroupAdmin, il ne peut créer que des groupes non administratifs.

  rmgroup — Supprime un groupe. Si l'utilisateur ne possède pas d'autorisation GroupAdmin, il ne peut supprimer que des groupes non administratifs.

ListAuditClasses
: Affiche la liste des classes d'audit valides. L'administrateur utilisant cette autorisation n'a pas besoin d'être l'utilisateur root ou de faire partie du groupe **audit**.

Utilisez le raccourci **smit mkuser** ou **smit chuser** pour afficher la liste des classes d'audit disponibles pour créer ou modifier un utilisateur. Entrez la liste des classes d'audit dans la zone **Classes d'AUDIT**.

PasswdAdmin
: Exécute les fonctions de l'utilisateur root sur les données des mots de passe. Les commandes suivantes utilisent l'autorisation PasswdAdmin :

  chsec
  : Modifie les attributs **lastupdate** et **flags** de tous les utilisateurs. Sans l'autorisation PasswdAdmin, la commande **chsec** ne permet à l'administrateur que de modifier les attributs **lastupdate** et **flags** des utilisateurs non administratifs.

  lssec
  : Affiche les attributs **lastupdate** et **flags** de tous les utilisateurs. Sans l'autorisation PasswdAdmin, la commande **lssec** ne permet à l'administrateur que d'afficher les attributs **lastupdate** et **flags** des utilisateurs non administratifs.

  pwdadm
  : Modifie le mot de passe de tous les utilisateurs. L'utilisateur doit être dans le groupe **security**.

PasswdManage
: Exécute l'administration des mots de passe des utilisateurs non administratifs. La commande suivante utilise l'autorisation PasswdManage :

  pwdadm
  : Modifie le mot de passe d'un utilisateur non administratif. L'administrateur doit être dans le groupe **security** ou posséder l'autorisation PasswdManage.

UserAdmin
: Exécute les fonctions de l'utilisateur root sur les données de l'utilisateur. Seuls les utilisateurs disposant de l'autorisation UserAdmin peuvent modifier les informations du rôle d'un utilisateur. Cette autorisation ne permet pas d'accéder ou de modifier les informations d'audit d'un utilisateur. Les commandes suivantes utilisent l'autorisation UserAdmin :

  chfn
  : Modifie la zone informations générales (gecos) d'un utilisateur. Si l'utilisateur ne possède pas d'autorisation UserAdmin mais figure dans le groupe **security**, il peut modifier la zone gecos de tous les utilisateurs non administratifs. Par ailleurs, les utilisateurs ne peuvent modifier que leur propre zone gecos.

  chsec
  : Modifie les données des utilisateurs administratifs dans les fichiers **/etc/passwd**, **/etc/security/environ**, **/etc/security/lastlog**, **/etc/security/limits** et **/etc/security/user** contenant l'attribut des rôles. L'administrateur peut également modifier les valeurs de strophe par défaut et le fichier **/usr/lib/security/mkuser.default**, à l'exception des attributs auditclasses.

  chuser
  : Modifie les informations des utilisateurs, sauf l'attribut auditclasses. Si l'utilisateur ne possède pas d'autorisation UserAdmin, il ne peut modifier que les informations d'un utilisateur non administratif, à l'exception des attributs rôles et auditclasses.

  mkuser
  : Crée un utilisateur, excepté pour l'attribut auditclasses. Si l'utilisateur ne possède pas d'autorisation UserAdmin, il ne peut créer que des utilisateurs non administratifs, excepté pour les attributs rôles et auditclasses.

  rmuser
  : Supprime un utilisateur. Si l'administrateur ne possède pas d'autorisation UserAdmin, il ne peut créer que des utilisateurs non administratifs.

UserAudit — Permet à l'utilisateur de modifier des informations d'audit.
Les commandes suivantes utilisent l'autorisation UserAudit :

- chsec — Modifie l'attribut auditclasses du fichier **mkuser.default** pour les utilisateurs non administratifs. Si l'utilisateur possède une autorisation UserAdmin, il peut également modifier l'attribut auditclasses du fichier **mkuser.default**, pour les utilisateurs administratifs et non administratifs.
- chuser — Modifie l'attribut auditclasses d'un utilisateur non administratif. Si l'administrateur possède une autorisation UserAdmin, il peut également modifier l'attribut auditclasses de tous les utilisateurs.
- lsuser — Affiche l'attribut auditclasses d'un utilisateur non administratif s'il s'agit de l'utilisateur root ou s'il fait partie du groupe **security**. Si l'utilisateur possède une autorisation UserAdmin, il peut également afficher l'attribut auditclasses de tous les utilisateurs.
- mkuser — Crée un nouvel utilisateur et permet à l'administrateur d'attribuer l'attribut auditclasses d'un utilisateur non administratif. Si l'utilisateur possède une autorisation UserAdmin, il peut également modifier l'attribut auditclasses de tous les utilisateurs.

RoleAdmin — Exécute les fonctions de l'utilisateur root sur les données du rôle.
Les commandes suivantes utilisent l'autorisation RoleAdmin :

- chrole — Modifie un rôle. Si l'administrateur ne possède pas d'autorisation RoleAdmin, la commande se termine.
- **lsrole** — Affiche un rôle.
- **mkrole** — Crée un rôle. Si l'administrateur ne possède pas d'autorisation RoleAdmin, la commande se termine.
- **rmrole** — Supprime un rôle. Si l'administrateur ne possède pas d'autorisation RoleAdmin, la commande se termine.

Restore — Exécute une restauration du système. La commande suivante utilise l'autorisation Restore :

- **Restore** — Restaure des fichiers sauvegardés. L'administrateur doit posséder une autorisation Restore.

Liste des commandes d'autorisation

Voici la liste des commandes et des autorisations dont elles font usage.

Commande	Permissions	Autorisations
chfn	2555 root.security	UserAdmin
chuser	4550 root.security	UserAdmin, UserAudit
diag	0550 root.system	Diagnostics
lsuser	4555 root.security	UserAudit, UserAdmin
mkuser	4550 root.security	UserAdmin, UserAudit
rmuser	4550 root.security	UserAdmin
chgroup	4550 root.security	GroupAdmin
lsgroup	0555 root.security	GroupAdmin
mkgroup	4550 root.security	GroupAdmin
rmgroup	4550 root.security	GroupAdmin
chgrpmem	2555 root.security	GroupAdmin
pwdadm	4555 root.security	PasswdManage, PasswdAdmin
passwd	4555 root.security	PasswdManage, PasswdAdmin
chsec	4550 root.security	UserAdmin, GroupAdmin, PasswdAdmin, UserAudit
lssec	0550 root.security	PasswdAdmin
chrole	4550 root.security	RoleAdmin
lsrole	0550 root.security	RoleAdmin
mkrole	4550 root.security	RoleAdmin
rmrole	4550 root.security	RoleAdmin
backup	4555 root.system	Backup
restore	4555 root.system	Restore

Comptes utilisateur

Attributs utilisateur recommandés

La gestion des utilisateurs consiste à créer des utilisateurs et des groupes, et à définir leurs attributs. L'un des principaux attributs est la méthode d'authentification. Les utilisateurs sont les principaux agents du système. Leurs attributs contrôlent leurs droits d'accès, leur environnement, leur méthode d'authentification de même que la façon, le moment et l'emplacement où leurs comptes sont accessibles.

Les groupes sont des ensembles d'utilisateurs pouvant partager les mêmes droits d'accès à des ressources protégées. Un groupe, défini par un ID, est composé de membres et d'administrateurs. Le créateur du groupe est généralement le premier administrateur.

De nombreux attributs peuvent être définis pour chaque compte utilisateur, y compris les attributs de mot de passe et de connexion. Pour obtenir une liste des attributs pouvant être définis, reportez–vous à la section Présentation du système de quotas de disque, page 2-29. Les attributs suivants sont recommandés :

- Chaque utilisateur doit disposer d'un ID utilisateur unique. Les outils de gestion de comptes et mesures de sécurité ne fonctionnent que si chaque utilisateur dispose de son propre ID.
- Attribuez aux utilisateurs des noms permettant de les identifier facilement sur le système. L'idéal est de choisir leurs noms réels, car la plupart des systèmes de messagerie utilisent l'ID utilisateur pour référencer le message entrant.
- Ajoutez, modifiez et supprimez des utilisateurs à l'aide du Web–based System Manager ou de l'interface SMIT. Bien que vous puissiez exécuter toutes ces tâches depuis la ligne de commande, ces interfaces permettent d'éviter certaines erreurs.
- N'attribuez pas de mot de passe à un compte utilisateur avant que l'utilisateur ne soit prêt à se connecter au système. Si le champ mot de passe a pour valeur un astérisque (*) dans le fichier **/etc/passwd**, les informations sur les comptes sont conservées, mais il est impossible de se connecter à ce compte.
- Ne modifiez pas les ID utilisateur définis par le système, nécessaires à son bon fonctionnement. Ces ID utilisateur sont énumérés dans le fichier **/etc/passwd**.
- En général, n'attribuez au paramètre **admin** la valeur **true** pour aucun ID utilisateur. Seul l'utilisateur root peut modifier les attributs des utilisateurs, et possède **admin=true** dans le fichier **/etc/security/user**.

Le système d'exploitation prend en charge les attributs utilisateur standard habituels des fichiers **/etc/passwd** et **/etc/group**, tels que :

Informations d'authentification	Définit le mot de passe
Données d'identification	Indique l'identifiant de l'utilisateur et les ID du groupe principal et des groupes complémentaires
Environnement	Indique l'environnement de shell et d'accueil.

Contrôle des comptes utilisateur

Un ensemble d'attributs est associé à chaque compte utilisateur. Ces attributs sont créés avec des valeurs par défaut lorsqu'un utilisateur est créé avec la commande **mkuser**. Les attributs peuvent être modifiés par la commande **chuser**. Voici la liste des attributs de gestion de la connexion non liés à la qualité du mot de passe :

account_locked	Pour qu'un compte soit explicitement verrouillé, attribuez la valeur *true* à cet attribut. Sa valeur par défaut est *false*.
admin	Avec la valeur true, cet utilisateur ne peut pas modifier son mot de passe. Seul l'administrateur peut le modifier.
admgroups	Répertorie les groupes pour lesquels l'utilisateur a des droits d'administrateur. Il peut ajouter ou supprimer des membres de ces groupes.
auth1	Méthode d'authentification utilisée pour permettre l'accès des utilisateurs. Généralement, il a pour valeur *SYSTEM*, qui utilisera alors les méthodes les plus récentes.
auth2	Méthode utilisée une fois l'utilisateur authentifié par la méthode spécifiée dans **auth1**. Elle ne peut pas bloquer l'accès au système. Généralement, sa valeur est *NONE*.
daemon	Ce paramètre booléen indique si l'utilisateur est autorisé à lancer des démons ou sous–systèmes à l'aide de la commande **startsrc**. Limite également l'utilisation de **cron** et **at**.
login	Indique si l'utilisateur a la possibilité d'établir une connexion.
logintimes	Applique une restriction lorsqu'un utilisateur se connecte. Par exemple, un utilisateur peut avoir un accès restreint au système, uniquement pendant les heures de bureau normales.
registry	Indique le registre des utilisateurs. Il peut servir à indiquer au système des informations sur les utilisateurs contenues dans les différents registres, tels que NIS, LDAP ou Kerberos.
rlogin	Indique si l'utilisateur a la possibilité d'établir une connexion via **rlogin** ou **telnet**.
su	Indique si d'autres utilisateurs peuvent passer sous cet ID à l'aide de la commande **su**.
sugroups	Indique les groupes autorisés à passer sous cet ID utilisateur.
ttys	Limite certains comptes à des zones physiques sécurisées.
expires	Gère les comptes d'étudiants ou d'invités ; peut également être utilisé pour désactiver des comptes temporairement.
loginretries	Indique le nombre maximum d'échecs de connexion successifs avant le verrouillage de l'ID utilisateur par le système. Les échecs de connexion sont consignés dans le fichier **/etc/security/lastlog**.
umask	Indique l'**umask** initial de l'utilisateur.
rcmds	Indique si le compte utilisateur est accessible avec les commandes **rsh** ou **exec**. La valeur `allow` indique que le compte est accessible avec **rsh** et **rexec**. La valeur `deny` indique que le compte n'est pas accessible avec **rsh** et **rexec**. La valeur `hostlogincontrol` indique que l'accès au compte est contrôlé par les attributs **hostallowedlogin** et **hostsdeniedlogin**.
hostallowedlogin	Indique les hôtes qui permettent à l'utilisateur de se connecter. Cet attribut est destiné à être utilisé dans un environnement réseau où les attributs utilisateur sont partagés par plusieurs hôtes.

hostsdeniedlogin Indique les hôtes qui ne permettent à l'utilisateur de se connecter. Cet attribut est destiné à être utilisé dans un environnement réseau où les attributs utilisateur sont partagés par plusieurs hôtes.

maxulogs Indique le nombre maximal de connexions par utilisateur. Lorsque l'utilisateur atteint le nombre maximal de connexions autorisé, la connexion est refusée.

L'ensemble des attributs utilisateur est défini dans les fichiers **/etc/security/user**, **/etc/security/limits**, **/etc/security/audit/config** et **/etc/security/lastlog**. La valeur utilisée par défaut lors de la création d'un utilisateur à l'aide de la commande **mkuser** est indiquée dans le fichier **/usr/lib/security/mkuser.default**. Seules les options qui remplacent les valeurs par défaut dans les strophes **default** des fichiers **/etc/security/user** et **/etc/securtiy/limits**, ainsi que les classes d'audit, doivent être spécifiées dans le fichier **mkuser.default**. Certains de ces attributs contrôlent la façon dont un utilisateur peut se connecter, et peuvent être configurés pour verrouiller automatiquement le compte utilisateur (empêcher les connexions suivantes) dans certaines conditions.

Une fois le compte utilisateur verrouillé par le système après le nombre de tentatives de connexion autorisé, l'utilisateur ne peut plus se connecter tant que l'administrateur système n'a pas réinitialisé son attribut **unsuccessful_login_count** dans le fichier **/etc/security/lastlog**, avec une valeur inférieure au nombre de tentatives de connexion. Pour ce faire, il faut procéder comme suit, à l'aide de la commande **chsec** :

```
chsec -f /etc/security/lastlog -s username -a
 unsuccessful_login_count=0
```

La commande **chsec** permettra de modifier les valeurs par défaut dans la strophe *default* du fichier de sécurité approprié, tel que **/etc/security/user** ou **/etc/security/limits**. De nombreuses valeurs par défaut sont définies comme comportement standard. Pour spécifier explicitement des attributs définis à chaque création d'utilisateur, modifiez l'entrée *user* dans **/usr/lib/security/mkuser.default**.

Pour plus d'informations sur les attributs de mot de passe étendus, reportez–vous à la section Mots de passe, page 2-20.

ID de connexion

Le système d'exploitation identifie également les utilisateurs par leur ID de connexion. Cet ID permet au système de suivre toutes les actions d'un utilisateur. Après la connexion d'un utilisateur, mais avant l'exécution de son programme initial, le système affecte l'ID de connexion du processus à l'ID utilisateur trouvé dans la base de données. Tous les process suivants qui se produisent pendant la session de connexion, sont balisés par cet ID. Ces balises fournissent un suivi de toutes les activités effectuées sous l'ID utilisateur de connexion. L'utilisateur peut, au cours de la session, réinitialiser l'ID utilisateur effectif, l'ID utilisateur réel, l'ID groupe effectif, l'ID groupe réel, mais ne peut modifier l'ID de connexion.

Sécurisation à l'aide de listes de contrôle des accès (ACL)

Pour atteindre un niveau de sécurité système convenable, élaborez une politique de sécurité cohérente pour la gestion de tous les comptes utilisateur. La liste de contrôle des accès (ACL) est la méthode de sécurité la plus courante. Pour plus d'informations sur les listes de contrôle des accès et la mise en place d'une politique de sécurité, reportez–vous à la section Liste de contrôle des accès, page 3-1.

Variable d'environnement PATH

La variable d'environnement **PATH** est importante pour la sécurité. Elle indique les répertoires dans lesquels une commande doit être recherchée. Pour l'ensemble du système, la valeur par défaut de **PATH** est indiquée dans le fichier **/etc/profile**. Chaque utilisateur dispose normalement d'une valeur **PATH** dans son fichier **$HOME/.profile**. La valeur **PATH** du fichier **.profile** remplace la valeur **PATH** du système ou lui ajoute des répertoires.

Les modifications non autorisées à la variable d'environnement **PATH** peuvent permettre à un utilisateur connecté de tromper d'autres utilisateurs (y compris les utilisateurs root). Les programmes *espion* (ou programmes *Cheval de Troie*) remplacent les commandes du système, puis interceptent les informations destinées à cette commande, telles que les mots de passe utilisateur.

Par exemple, si un utilisateur modifie la valeur **PATH** de façon à ce que le système commence par rechercher le répertoire **/tmp** lorsqu'une commande est lancée, et qu'il place dans le répertoire **/tmp** un programme nommé **su** qui demande le mot de passe root comme le fait la commande **su**, alors le programme **/tmp/su** envoie le mot de passe root à cet utilisateur et appelle la commande **su** réelle avant de se fermer. Dans ce cas, tout utilisateur root ayant utilisé la commande **su** révélerait le mot de passe root sans même s'en rendre compte.

Afin d'éviter tout problème avec la variable d'environnement **PATH** pour les utilisateurs et administrateurs du système, procédez comme suit :

- Lorsque vous avez un doute, appelez une commande par le nom de chemin complet. Lorsque vous entrez un nom de chemin entier, la variable d'environnement **PATH** est ignorée.
- Ne placez jamais le répertoire actuel (spécifié par un **.** (point)) dans la valeur **PATH** spécifiée pour l'utilisateur root. Ne permettez jamais la spécification du répertoire en cours dans **/etc/profile**.
- L'utilisateur root doit avoir sa propre spécification **PATH** dans son fichier privé **.profile**. En règle générale, la spécification dans **/etc/profile** indique les valeurs par défaut minimales pour tous les utilisateurs. Quant à l'utilisateur root, il peut avoir besoin d'un nombre de répertoires différent de la valeur par défaut.
- Prévenez les autres utilisateurs de ne pas modifier leurs fichiers **.profile** sans avoir consulté l'administrateur système. Autrement, un utilisateur pourrait apporter des modifications qui permettraient un accès non souhaité. Les droits d'un fichier **.profile** d'utilisateur doivent être définis sur 740.
- Les administrateurs système ne doivent pas utiliser la commande **su** pour obtenir le privilège root depuis une session utilisateur, car la valeur **PATH** de l'utilisateur indiquée dans le fichier **.profile** est alors effective. Les utilisateurs peuvent définir leurs propres fichiers **.profile**. Les administrateurs système doivent se connecter au poste de l'utilisateur en tant que root, ou mieux, avec leur propre ID, et utiliser la commande suivante :

```
/usr/bin/su - root
```

  Ceci assure d'utiliser l'environnement root au cours de la session. Si un administrateur système travaille en tant que root dans une autre session utilisateur, il doit alors indiquer des chemins d'accès complets au cours de la session.

- Protégez la variable d'environnement **IFS** (input field separator) contre les modifications dans le fichier **/etc/profile**. La variable d'environnement **IFS** dans le fichier **.profile** peut être utilisée pour modifier la valeur **PATH**.

Configuration d'un accès FTP anonyme avec un compte utilisateur sécurisé

Eléments à prendre en considération

Les informations contenues dans ces instructions ont été testées avec AIX 5.2. Les résultats peuvent être sensiblement différents si vous utilisez une autre version ou niveau de AIX.

Le présent scénario configure un accès **ftp** anonyme avec un compte utilisateur sécurisé, à l'aide de l'interface de ligne de commandes et d'un script.

Remarque : Ce scénario ne peut pas être utilisé sur un système équipé de la fonction CAPP (Controlled Access Protection Profile) et EAL4+ (Evaluation Assurance Level 4+).

1. Vérifiez que l'ensemble de fichiers **bos.net.tcp.client** est installé sur votre système, à l'aide de la commande :

```
lslpp -L | grep bos.net.tcp.client
```

   Si vous ne recevez aucune sortie, l'ensemble de fichiers n'est pas installé. Pour plus d'informations sur son installation, reportez–vous au manuel *AIX 5L Version 5.3 – Références et guide d'installation.*

2. Vérifiez que vous disposez d'au moins 8 Mo d'espace libre dans le répertoire **/home** du système :

```
df -k /home
```

   Le script de l'étape 4 nécessite au moins 8 Mo d'espace libre dans le répertoire **/home** pour installer les fichiers et les répertoires nécessaires. Si vous devez augmenter l'espace disponible, reportez–vous au manuel *AIX 5L Version 5.3 – Guide de gestion du système : Système d'exploitation et unités*.

3. Avec les droits d'accès root, allez dans le répertoire **/usr/samples/tcpip**. Par exemple :

```
cd /usr/samples/tcpip
```

4. Pour configurer le compte, exécutez le script suivant :

```
./anon.ftp
```

5. Lorsque s'affiche l'invite vous demandant si vous souhaitez modifier /home/ftp (`Are you sure you want to modify /home/ftp ?`), entrez **oui** (yes). Le résultat affiché sera semblable au suivant :

```
Added user anonymous.
Made /home/ftp/bin directory.
Made /home/ftp/etc directory.
Made /home/ftp/pub directory.
Made /home/ftp/lib directory.
Made /home/ftp/dev/null entry.
Made /home/ftp/usr/lpp/msg/en_US directory.
```

6. Passez dans le répertoire **/home/ftp**. Par exemple :

```
cd /home/ftp
```

7. Créez un sous–répertoire **home**, en entrant :

```
mkdir home
```

8. Changez les droits d'accès du répertoire **/home/ftp/home** en `drwxr-xr-x`, en entrant :

```
chmod 755 home
```

9. Passez dans le répertoire **/home/ftp/etc**, en entrant :

```
cd /home/ftp/etc
```

10. Créez le sous–répertoire **objrepos**, en entrant :

```
mkdir objrepos
```

11. Changez les droits d'accès du répertoire **/home/ftp/etc/objrepos** en `drwxrwxr-x`, en entrant :

```
chmod 775 objrepos
```

12. Changez le propriétaire et le groupe du répertoire **/home/ftp/etc/objrepos** en utilisateur root et groupe system, en entrant :

```
chown root:system objrepos
```

13. Créez un sous–répertoire **security**, en entrant :

```
mkdir security
```

14. Changez les droits d'accès du répertoire **/home/ftp/etc/security** en `drwxr-x---`, en entrant :

```
chmod 750 security
```

15. Changez le propriétaire et le groupe du répertoire **/home/ftp/etc/security** en utilisateur root et groupe de sécurité, en entrant :

```
chown root:security security
```

16. Passez dans le répertoire **/ home/ftp/etc/security**, en entrant :

```
cd security
```

17. Ajoutez un utilisateur à l'aide du raccourci SMIT suivant :

```
smit mkuser
```

    Dans le présent scénario, nous ajoutons l'utilisateur `test`.

18. Dans les zones SMIT, entrez les valeurs suivantes :

```
User NAME                                          [test]
 ADMINISTRATIVE USER?                               true
 Primary GROUP                                     [staff]
 Group SET                                         [staff]
 Another user can SU TO USER?                       true
 HOME directory                                    [/home/test]
```

    Après avoir entré vos modifications, appuyez sur Entrée pour créer l'utilisateur. A la fin du processus SMIT, quittez SMIT.

19. Créez un mot de passe pour l'utilisateur à l'aide de la commande suivante :

```
passwd test
```

    A l'invite, entrez le mot de passe voulu. Vous devez le saisir une deuxième fois pour confirmation.

20. Passez dans le répertoire **/home/ftp/etc**, en entrant :

```
cd /home/ftp/etc
```

21. Copiez le fichier **/etc/passwd** en **/home/ftp/etc/passwd** à l'aide de la commande suivante :

```
cp /etc/passwd /home/ftp/etc/passwd
```

22. Avec l'éditeur de votre choix, éditez le fichier **/home/ftp/etc/passwd**. Par exemple :

```
vi passwd
```

23. Supprimez toutes les lignes du contenu copié, sauf celles concernant les utilisateurs root, ftp et test. Après modification, le contenu du fichier doit être semblable à ce qui suit :

```
root:!:0:0::/:/bin/ksh
 ftp:*:226:1::/home/ftp:/usr/bin/ksh
 test:!:228:1::/home/test:/usr/bin/ksh
```

24. Enregistrez vos modifications et quittez l'éditeur.

25. Changez les droits d'accès du fichier **/home/ftp/etc/passwd** en `-rw-r--r--`, en entrant :

```
chmod 644 passwd
```

26. Changez le propriétaire et le groupe du répertoire **/home/ftp/etc/passwd** en utilisateur root et groupe de sécurité, en entrant :

```
chown root:security passwd
```

27. Copiez le contenu du fichier **/etc/security/passwd** vers **/home/ftp/etc/security/passwd**, à l'aide de la commande suivante :

```
cp /etc/security/passwd /home/ftp/etc/security/passwd
```

28. Avec l'éditeur de votre choix, éditez le fichier **/home/ftp/etc/security/passwd**. Par exemple :

```
vi ./security/passwd
```

29. Supprimez toutes les strophes du contenu copié, sauf celle concernant l'utilisateur test.

30. Supprimez la ligne `flags = ADMCHG` de la strophe de l'utilisateur test. Après modification, le contenu du fichier doit être semblable à ce qui suit :

```
test:
         password = 2HaAYgpDZX3Tw
         lastupdate = 990633278
```

31. Enregistrez vos modifications et quittez l'éditeur.

32. Changez les droits d'accès du fichier **/home/ftp/etc/security/passwd** en `-rw-------`, en entrant :

```
chmod 600 ./security/passwd
```

33. Changez le propriétaire et le groupe du répertoire **/home/ftp/etc/ security/passwd** en utilisateur root et groupe de sécurité, en entrant :

```
chown root:security ./security/passwd
```

34. Avec l'éditeur de votre choix, éditez le fichier **/home/ftp/etc/security/group**. Par exemple :

```
vi ./security/group
```

35. Ajoutez les lignes suivantes au fichier :

```
system:*:0:
 staff:*:1:test
```

36. Enregistrez vos modifications et quittez l'éditeur.

37. A l'aide des commandes suivantes, copiez le contenu approprié dans le répertoire **/home/ftp/etc/objrepos** :

```
cp /etc/objrepos/CuAt ./objrepos
cp /etc/objrepos/CuAt.vc ./objrepos
cp /etc/objrepos/CuDep ./objrepos
cp /etc/objrepos/CuDv ./objrepos
cp /etc/objrepos/CuDvDr ./objrepos
cp /etc/objrepos/CuVPD ./objrepos
cp /etc/objrepos/Pd* ./objrepos
```

38. Passez dans le répertoire **/home/ftp/home**, en entrant :

```
cd ../home
```

39. Créez un nouveau répertoire personnel pour votre utilisateur, en entrant :

```
mkdir test
```

Il s'agira du répertoire de travail du nouvel utilisateur ftp.

40. Changez le propriétaire et le groupe du répertoire **/home/ftp/home/test** en utilisateur `test` et en groupe staff, en entrant :

```
chown test:staff test
```

41. Changez les droits d'accès du fichier **/home/ftp/home/test** en `-rwx------`, en entrant :

```
chmod 700 test
```

A ce stade, une sous–connexion ftp est configurée sur votre machine.
Vous pouvez la tester grâce à la procédure ci–après :

1. Sous ftp, connectez–vous à l'hôte sur lequel vous avez créé l'utilisateur `test`. Par exemple :

```
ftp MyHost
```

2. Connectez–vous en tant qu'utilisateur `anonymous`.
   Lorsque vous êtes invité à spécifier un mot de passe, appuyez sur Entrée.
3. Basculez sur le nouvel utilisateur `test` à l'aide de la commande suivante :

```
user test
```

   A l'invite de mot de passe, utilisez celui que vous avez créé à l'étape 19.

4. Servez–vous de la commande pwd pour vérifier que le répertoire personnel de l'utilisateur existe. Par exemple :

```
ftp>   pwd
       /home/test
```

   /home/test apparaît alors comme sous–répertoire **ftp**. En fait, le nom du chemin d'accès complet sur l'hôte est **/home/ftp/home/test**.

Comptes utilisateurs spécifiques au système

AIX fournit un ensemble par défaut de comptes utilisateur spécifiques au système, qui évite aux comptes utilisateur root et système de détenir tous les fichiers et systèmes de fichiers du système d'exploitation.

Attention : Soyez prudent lors de la suppression d'un compte utilisateur spécifique au système. Vous pouvez désactiver un compte en insérant un astérisque (*) au début de la ligne correspondante du fichier **/etc/security/passwd**. Faites attention à ne pas désactiver le compte utilisateur root. Si vous supprimez des comptes utilisateur spécifiques au système ou désactivez le compte root, le système d'exploitation ne fonctionnera pas.

Les comptes suivants sont prédéfinis dans le système d'exploitation :

adm Le compte utilisateur adm détient deux fonctions système de base :

- Diagnostics, dont les outils sont stockés dans le répertoire **/usr/sbin/perf/diag_tool**.
- Comptabilité, dont les outils sont stockés dans les répertoires suivants :
  - **/usr/sbin/acct**
  - **/usr/lib/acct**
  - **/var/adm**
  - **/var/adm/acct/fiscal**
  - **/var/adm/acct/nite**
  - **/var/adm/acct/sum**

bin Le compte utilisateur bin détient généralement les fichiers exécutables pour la plupart des commandes utilisateur. Son rôle principal est d'aider à répartir la propriété des répertoires et fichiers système importants, afin que tout ne soit pas détenu uniquement par les comptes utilisateur root et système.

daemon Ce compte utilisateur sert uniquement à détenir et exécuter les processus serveur du système et les fichiers associés. Ce compte garantit que ces processus s'exécutent avec les droits appropriés d'accès aux fichiers.

nobody Le compte utilisateur nobody est utilisé par le NFS (Network File System) pour permettre les impressions à distance. Ce compte permet à un programme d'accorder un accès root temporaire aux utilisateurs root. Par exemple, avant d'activer RPC sécurisé ou NFS sécurisé, contrôlez la clé **/etc/public** sur le serveur NIS maître pour vérifier si un utilisateur ne s'est pas vu attribuer une clé publique et une clé secrète. En tant qu'utilisateur root, vous pouvez créer une entrée dans la base de données pour chaque utilisateur sans affectation, en entrant :

```
newkey -u   username
```

Vous pouvez aussi créer une entrée dans la base de données pour le compte utilisateur nobody, chaque utilisateur pourra alors exécuter le programme **chkey** pour créer ses propres entrées dans la base de données, sans se connecter en tant que root.

root Le compte utilisateur root, UID 0, qui permet d'exécuter les tâches de maintenance et de résoudre les problèmes du système.

sys L'utilisateur sys détient le point de montage par défaut du cache DFS (Distributed File Service), qui doit exister préalablement à l'installation et à la configuration du DFS sur un client. Le répertoire **/usr/sys** sert également à stocker les images d'installation.

system Le groupe system est un groupe défini par le système pour les administrateurs. Les utilisateurs appartenant à ce groupe ont le droit d'effectuer certaines tâches de maintenance système sans disposer des droits d'accès root.

Suppression de comptes utilisateur par défaut inutiles

Lors de l'installation du système d'exploitation, plusieurs ID utilisateur et de groupe sont créés. Selon les applications exécutées sur votre système et selon l'emplacement de votre système sur le réseau, certains de ces ID utilisateur et de groupe peuvent devenir des éléments vulnérables qui nuisent à la sécurité. Si ces ID utilisateur et de groupe ne sont pas nécessaires, vous pouvez les retirer pour minimiser les risques.

Le tableau suivant répertorie les ID utilisateur par défaut que vous pouvez le plus souvent supprimer :

Tableau 4. ID utilisateur par défaut que vous pourriez supprimer.

ID utilisateur	Description
uucp, nuucp	Détenteur de fichiers cachés utilisés par le protocole uucp. Le compte utilisateur uucp est utilisé pour le programme de copie UNIX–to–UNIX, lequel est un groupe de commandes, programmes et fichiers, présent sur la plupart des systèmes AIX, et permettant de communiquer avec un autre système AIX via une ligne dédiée ou une ligne téléphonique.
lpd	Détenteur de fichiers utilisés par le sous–système d'impression
guest	Permet l'accès aux utilisateurs qui n'ont normalement pas accès aux comptes

Le tableau suivant répertorie les ID de groupe que vous pouvez peut–être supprimer :

Tableau 5. ID de groupe qui ne seront peut–être pas nécessaires.

ID du groupe	Description
uucp	Groupe auquel appartiennent les utilisateurs uucp et nuucp
printq	Groupe auquel appartient l'utilisateur lpd

Analysez votre système pour déterminer quels ID ne sont pas nécessaires. Il se peut que d'autres ID utilisateur ou de groupe ne soient pas nécessaires. Avant de commencer à exploiter votre système, effectuez une évaluation complète des ID disponibles.

Mots de passe

Deviner les mots de passe est l'une des méthodes d'attaque les plus répandues. Il est donc essentiel de contrôler et surveiller votre politique de gestion des mots de passe. AIX intègre des mécanismes qui vous aideront à appliquer une politique de mots de passe plus efficace, telle que la définition de valeurs pour les données suivantes :

- Nombres minimum et maximum de semaines écoulées avant et après la modification d'un mot de passe
- Longueur minimum d'un mot de passe
- Nombre minimum de caractères alphabétiques pouvant être utilisés lors de la sélection d'un mot de passe

Cette section traite la façon dont AIX conserve et gère les mots de passe, et indique comment mettre en place une politique de mots de passe efficace. Les sujets traités dans cette section sont :

Etablissement de mots de passe efficaces

Les mots de passe sont la première ligne de défense contre l'accès non autorisé aux systèmes. Pour être efficaces, ils doivent répondent aux critères suivants :

- Une combinaison de lettres en minuscules et majuscules.
- Une combinaison de caractères alphabétiques, numériques ou de ponctuation. Les mots de passe peuvent aussi comporter des caractères spéciaux tels que `~!@#$%^&*()-_=+[]{}|\;:'",.?/<espace>`
- Ne sont écrits nulle part.
- Sont composés de 7 à 8 caractères, en cas d'utilisation du fichier **/etc/security/passwd** (d'autres règles d'authentification utilisant des registres, comme LDAP, autorisent des mots de passe plus longs).
- Ne figurent pas dans les dictionnaires.
- Ne sont pas des suites de lettres du clavier, telles que *azerty*.
- Ne sont pas des mots réels ou suites de lettres connus, épelés à l'envers.
- Ne contiennent aucune information sur vous–même, votre famille ou vos amis.
- Ne ressemblent pas au précédent mot de passe
- Peuvent être saisis rapidement, pour ne pas être identifiés par une personne à côté de vous.

Vous pouvez également définir des règles plus strictes, interdisant l'utilisation dans les mots de passe de termes UNIX standard pouvant être devinés. Cette fonction utilise **dictionlist**, qui nécessite l'installation préalable des ensembles de fichiers **bos.data** et **bos.txt**.

Pour mettre en place **dictionlist**, modifiez la ligne qui suit dans le fichier **/etc/security/users** :

```
dictionlist = /usr/share/dict/words
```

Le fichier **/usr/share/dict/words** fait appel à **dictionlist** pour empêcher l'utilisation des termes UNIX standard en tant que mot de passe.

Utilisation du fichier /etc/passwd

En règle générale, le fichier **/etc/passwd** est utilisé pour garder la trace de tous les utilisateurs enregistrés ayant accès au système. Le fichier **/etc/passwd** utilise le caractère ":" comme séparateur. Il contient les informations suivantes :

- Nom d'utilisateur
- Mot de passe chiffré
- ID utilisateur (UID)
- ID du groupe de l'utilisateur (GID)
- Nom complet de l'utilisateur (GECOS)
- Répertoire home de l'utilisateur
- Shell de connexion

Voici un exemple de fichier **/etc/passwd** :

```
root:!:0:0::/:/usr/bin/ksh
daemon:!:1:1::/etc:
bin:!:2:2::/bin:
sys:!:3:3::/usr/sys:
adm:!:4:4::/var/adm:
uucp:!:5:5::/usr/lib/uucp:
guest:!:100:100::/home/guest:
nobody:!:4294967294:4294967294::/:
lpd:!:9:4294967294::/:
lp:*:11:11::/var/spool/lp:/bin/false
invscout:*:200:1::/var/adm/invscout:/usr/bin/ksh
nuucp:*:6:5:uucp login user:/var/spool/uucppublic:/usr/sbin/uucp/uucico
paul:!:201:1::/home/paul:/usr/bin/ksh
jdoe:*:202:1:John Doe:/home/jdoe:/usr/bin/ksh
```

Contrairement à UNIX, qui conserve les mots de passe chiffrés dans le fichier **/etc/password**, AIX les enregistre par défaut dans le fichier **/etc/security/password**, lisible uniquement par l'utilisateur root. AIX utilise le mot de passe classé dans **/etc/passwd** pour déterminer si un mot de passe existe ou si le compte est bloqué.

Le fichier **/etc/passwd** doit être accessible en lecture par tous les utilisateurs, et accessible en écriture uniquement par l'utilisateur root (qui en est le propriétaire). Ceci est déterminé par `-rw-r--r--`. Si un ID utilisateur dispose d'un mot de passe, le champ du mot de passe contiendra un `!` (point d'exclamation). Dans le cas contraire, le champ du mot de passe contiendra un `*` (astérisque). Les mots de passe chiffrés sont conservés dans le fichier **/etc/security/passwd**. L'exemple suivant montre les quatre dernières entrées du fichier **/etc/security/passwd**, correspondant aux entrées du fichier **/etc/passwd** mentionné ci–dessus.

```
guest:
         password = *

nobody:
         password = *

lpd:
         password = *

paul:
         password = eacVScDKri4s6
         lastupdate = 1026394230
         flags = ADMCHG
```

Notez qu'aucune entrée ne figure dans le fichier **/etc/security/passwd** pour l'ID utilisateur `jdoe`, parce que aucun mot de passe n'est défini dans le fichier **/etc/passwd**.

Vous pouvez contrôler la cohérence du fichier **/etc/passwd** à l'aide de la commande **pwdck**. Cette commande vérifie l'exactitude des informations relatives aux mots de passe dans les fichiers de la base de données utilisateurs, en contrôlant les définitions de tous les utilisateurs ou des utilisateurs indiqués.

Utilisation du fichier /etc/passwd et des environnements réseau

Dans un environnement réseau standard, chaque utilisateur doit posséder un compte sur chacun des systèmes pour y avoir accès. En principe, l'utilisateur a donc une entrée dans le fichier **/etc/passwd** de chaque système. Toutefois, dans un environnement distribué, il n'est pas facile de vérifier que chaque système a le même fichier **/etc/passwd**. Pour résoudre ce problème, plusieurs méthodes mettent les informations à disposition dans le fichier **/etc/passwd** via le réseau, y compris NIS (Network Information System) et NIS +.

Pour plus d'informations sur NIS et NIS+, reportez–vous à la section Sécurité NIS (Network Information Services) et NIS+, page 13-1.

Dissimulation des noms d'utilisateur et mots de passe

Pour atteindre un niveau de sécurité supérieur, assurez–vous que les ID utilisateur et mots de passe ne sont pas visibles sur le système. Les fichiers **.netrc** contiennent les ID utilisateurs et mots de passe. Ces fichiers ne sont pas protégés par chiffrement ou codage : leur contenu est affiché en clair. Pour trouver ces fichiers, lancez la commande suivante :

```
# find `awk –F: '{print $6}' /etc/passwd` –name .netrc –ls
```

Après avoir localisé les fichiers, supprimez–les. Un moyen plus efficace d'enregistrer des mots de passe consiste à installer Kerberos. Pour plus d'informations sur Kerberos, reportez–vous à la section Kerberos, page 16-1.

Paramétrage des options de mot de passe recommandées

Seule la formation des utilisateurs peut permettre une gestion efficace des mots de passe. Pour une meilleure sécurité, le système d'exploitation propose des fonctions configurables de restriction des mots de passe. Elles permettent à l'administrateur de contrôler les mots de passe choisis par les utilisateurs, et d'imposer leur modification régulière. Les options de mots de passe et les attributs utilisateur étendus se trouvent dans **/etc/security/user**, un fichier ASCII contenant les strophes d'attributs pour les utilisateurs. Ces restrictions sont appliquées à chaque définition d'un nouveau mot de passe utilisateur. Toutes les restrictions de mot de passe sont définies au cas par cas. En conservant les restrictions dans la strophe par défaut du fichier **/etc/security/user**, les mêmes restrictions sont appliquées à tous les utilisateurs. Pour maintenir une sécurité efficace, tous les mots de passe doivent être protégés de la même façon.

Les administrateurs peuvent également étendre les restrictions de mots de passe. L'attribut **pwdchecks** du fichier **/etc/security/user** permet à un administrateur d'ajouter de nouvelles sous–routines (ou *méthodes*) au code de restriction de mot de passe. Des politiques de site peuvent donc être ajoutées et appliquées par le système d'exploitation. Pour plus d'informations, reportez–vous à la section Extension des restrictions de mot de passe, page 2-26.

Appliquez les restrictions de mot de passe de manière rationnelle. La mise en place de mesures trop restrictives peut diminuer la sécurité des mots de passe : un mot de passe contenant un nombre de caractères trop limité permet de le deviner plus facilement, et imposer des mots de passe compliqués difficiles à mémoriser incitera les utilisateurs à les noter. Enfin, la sécurité des mots de passe dépend des utilisateurs. La meilleure stratégie consiste à imposer des restrictions simples de mots de passe, à donner des directives claires et à contrôler régulièrement que les mots de passe actuels sont uniques.

Le tableau suivant indique les valeurs recommandées pour certains attributs de sécurité liés aux mots de passe utilisateur, dans le fichier **/etc/security/user**.

Tableau 6. Valeurs des attributs de sécurité recommandées pour les mots de passe utilisateur.

Attribut	Description	Valeur recommandée	Valeur par défaut	Valeur maximale
dictionlist	Vérifie que les mots de passe n'incluent pas de termes UNIX.	**/usr/share/dict/words**	Ne s'applique pas	Ne s'applique pas
histexpire	Nombre de semaines avant de pouvoir réutiliser le mot de passe.	26	0	260 Remarque 1
histsize	Nombre de répétitions permises du mot de passe.	20	0	50
maxage	Nombre maximum de semaines avant de devoir modifier le mot de passe.	8	0	52
maxexpired	Nombre maximum de semaines après *maxage* pendant lesquelles un utilisateur peut modifier son mot de passe expiré. (L'utilisateur root en est dispensé.)	2	–1	52
maxrepeats	Nombre maximum de caractères pouvant être répétés dans les mots de passe.	2	8	8

minage	Nombre minimum de semaines avant de pouvoir modifier un mot de passe. Cette valeur doit être égale à zéro, à moins que les adminis–trateurs ne puissent être joints à tout moment pour réinitialiser un mot de passe récemment modifié et compromis par accident.	0	0	52
minalpha	Nombre minimum de caractères alphabétiques dans un mot de passe.	2	0	8
mindiff	Nombre minimum de caractères uniques devant figurer dans un mot de passe.	4	0	8
minlen	Longueur minimum du mot de passe.	6 (8 pour l'utilisateur root)	0	8
minother	Nombre minimum de caractères non–alphabétiques devant figurer dans un mot de passe.	2	0	8

pwdwarntime	Nombre de jours avant que le système n'émette un avertissement indiquant que la modification du mot de passe est nécessaire.	5	Ne s'applique pas	Ne s'applique pas
pwdchecks	Cette entrée ajoute à la commande **passwd** un code personnalisé chargé du contrôle de la qualité du mot de passe.	Pour plus d'informations, reportez–vous à la section Extension des restrictions de mot de passe, page 2-26.	Ne s'applique pas	Ne s'applique pas

Remarques :

1. Un maximum de 50 mots de passe sont mémorisés.

Pour un système CAPP et EAL4+ (Controlled Access Protection Profile et Evaluation Assurance Level 4+), appliquez les valeurs recommandées à la section Configuration du port et de l'utilisateur, page 1-14.

Si un traitement de texte est installé sur le système, l'administrateur peut utiliser le fichier **/usr/share/dict/words** comme fichier dictionnaire **dictionlist**. Dans ce cas, il définira l'attribut **minother** sur `0`. En effet, la plupart des mots du dictionnaire ne remplissent pas la condition définie par l'attribut **minother**, le fait de définir cet attribut sur `1` ou plus élimine la grande majorité des mots contenus dans ce dictionnaire.

La longueur minimale d'un mot de passe du système est définie par le maximum de la somme de l'attribut **minother** avec l'attribut **minlen** ou l'attribut **minalpha**. Le mot de passe ne doit pas dépasser 8 caractères. La valeur de l'attribut **minalpha** ajoutée à la valeur de l'attribut **minother** ne doit jamais être supérieure à 8. Si la valeur de l'attribut **minalpha** ajoutée à la valeur de l'attribut **minother** est supérieure à 8, la valeur de l'attribut **minother** est réduite à 8 moins la valeur de l'attribut **minalpha**.

Si les attributs **histexpire** et **histsize** sont définis, le système mémorise le nombre de mots de passe requis pour remplir les deux conditions, dans la limite système de 50 mots de passe par utilisateur. Les mots de passe vides ne sont pas mémorisés.

Vous pouvez modifier le fichier **/etc/security/user** pour inclure toute valeur par défaut à utiliser pour administrer les mots de passe utilisateur. Une autre solution consiste à modifier les valeurs d'attribut à l'aide de la commande **chuser**.

Avec ce fichier, vous pouvez également utiliser les commandes **mkuser**, **lsuser** et **rmuser**. La commande **mkuser** crée dans le fichier **/etc/security/user** une entrée pour chaque nouvel utilisateur et initialise ses attributs avec ceux qui ont été définis dans le fichier **/usr/lib/security/mkuser.default**. Pour afficher les attributs et leurs valeurs, utilisez la commande **lsuser**. Pour supprimer un utilisateur, utilisez la commande **rmuser**.

Extension des restrictions de mot de passe

Les administrateurs système peuvent étendre les règles utilisées par le programme de mots de passe pour accepter et rejeter les mots de passe (restrictions de composition de mot de passe) afin de les adapter à chaque site. Les restrictions sont étendues grâce à l'ajout de méthodes, qui sont appelées lors d'une modification de mot de passe. L'attribut **pwdchecks** du fichier **/etc/security/user** indique les méthodes appelées.

Le guide *AIX 5L Version 5.3 Technical Reference* décrit l'interface de sous–routine **pwdrestrict_method**, à laquelle les méthodes de restriction de mot de passe spécifiées doivent se conformer. Pour étendre correctement les restrictions de composition de mot de passe, l'administrateur système doit programmer cette interface lorsqu'il écrit une méthode de restriction de mots de passe. Soyez prudent lorsque vous étendez des restrictions de composition de mot de passe. Ces extensions affectent directement les commandes **login**, **passwd** et **su**, ainsi que d'autres programmes. Un code malveillant ou défectueux peut facilement nuire à la sécurité d'un système.

Authentification de l'utilisateur

Identification et authentification déterminent l'identité d'un utilisateur. Chaque utilisateur doit se connecter au système. Il indique le nom d'utilisateur d'un compte et éventuellement un mot de passe (sur un système sécurisé, tous les comptes non affectés d'un mot de passe doivent être invalidés). Si le mot de passe est correct, l'utilisateur accède au compte correspondant et dispose des droits et des privilèges associés. Les fichiers **/etc/passwd** et **/etc/security/passwd** gèrent les mots de passe utilisateur.

Par défaut, les utilisateurs sont définis dans le registre de fichiers. Cela signifie que les données du compte utilisateur et du groupe sont stockées dans des fichiers ASCII ordinaires. Avec l'introduction de modules de chargement complémentaires, les utilisateurs peuvent également être définis dans d'autres registres. Par exemple, si le module complémentaire LDAP est utilisé pour l'administration des utilisateurs, alors les définitions des utilisateurs sont stockées dans le répertoire LDAP. Dans ce cas, le fichier **/etc/security/user** ne contient aucune entrée relative aux utilisateurs (à l'exception des attributs utilisateur **SYSTEM** et **registry**). Lorsqu'un module de chargement composé (c'est–à–dire les modules de chargement contenant une partie authentification et une partie base de données) est utilisé pour l'administration des utilisateurs, la partie base de données détermine la méthode d'administration des informations relatives aux comptes des utilisateurs AIX, tandis que la partie authentification décrit l'administration liée à l'authentification et aux mots de passe. La partie authentification peut également décrire les attributs d'administration des comptes utilisateurs spécifiques à l'authentification via la mise en oeuvre d'interfaces de module de chargement (newuser, getentry, putentry, etc.).

D'autres méthodes d'authentification sont intégrées au système au moyen de l'attribut SYSTEM qui figure dans le fichier **/etc/security/user**. L'attribut **SYSTEM** permet à l'administrateur système de définir avec un niveau de granularité élevé la ou les méthodes d'authentification à appliquer pour permettre aux utilisateurs d'accéder au système. Par exemple, dans l'environnement DCE (Distributed Computing Environment) l'authentification par mot de passe et requise mais la méthode de validation des mots de passe est différente de celle du modèle de chiffrement utilisé dans les commandes **/etc/passwd** et **/etc/security/passwd**.

La valeur de l'attribut **SYSTEM** est définie via une grammaire. L'application d'une grammaire permet à l'administrateur système de combiner une ou plusieurs méthodes pour authentifier un utilisateur particulier sur le système. Les jetons utilisés dans les méthodes courantes sont `compat`, `DCE`, `files` et `NONE`.

La valeur par défaut du système est `compat`. La valeur par défaut `SYSTEM=compat` indique au système d'utiliser la base de données locale pour l'authentification. En l'absence de résolution, le système consulte la base de données NIS (Network Information Services). Le jeton `files` indique que seuls les fichiers locaux doivent être utilisés lors de l'authentification, tandis que le résultat de `SYSTEM=DCE` est un flux d'authentification `DCE`.

Le jeton `NONE` désactive l'authentification par méthode. Pour désactiver toutes les authentifications, `NONE` doit apparaître dans les lignes **SYSTEM** et **auth1** de la strophe de l'utilisateur.

Vous pouvez indiquer deux méthodes ou plus et les combiner avec les opérateurs logiques `AND` et `OR`. Par exemple, `SYSTEM=DCE` OR `compat` indique que l'utilisateur est autorisé à se connecter en cas de succès de l'une des méthodes, `DCE` ou l'authentification locale (crypt()), qui sont appliquées dans l'ordre indiqué.

De façon similaire, un administrateur système peut utiliser les noms de module de chargement d'authentification pour l'attribut **SYSTEM**. Par exemple, si la valeur de l'attribut **SYSTEM** est `SYSTEM=KRB5files` OR `compat`, l'hôte AIX tente l'authentification Kerberos, puis en cas d'échec, il tente l'authentification AIX standard.

Les attributs **SYSTEM** et **registry** sont toujours stockés sur le système de fichiers local, dans le fichier **/etc/security/user**. Si un utilisateur AIX est défini sous LDAP et les attributs **SYSTEM** et **registry** sont définis de façon appropriée, une entrée du fichier **/etc/security/user** sera associée à l'utilisateur.

Les attributs **SYSTEM** et **registry** d'un utilisateur peuvent être modifiés à l'aide de la commande **chuser**.

Vous pouvez définir des jetons valides pour l'attribut **SYSTEM** dans le fichier **/usr/lib/security/methods.cfg**.

Remarque : L'utilisateur root est toujours authentifié au moyen d'un fichier sécurité du système local. La valeur de l'attribut **SYSTEM** de l'utilisateur root est `SYSTEM = "compat"` dans le fichier **/etc/security/user**.

Pour obtenir des informations sur la protection par mot de passe, voir *AIX 5L Version 5.3 System User's Guide : Operating System and Devices*.

D'autres méthodes d'authentification sont intégrées au système au moyen de l'attribut **SYSTEM** qui figure dans **/etc/security/user**. Par exemple, l'environnement DCE (Distributed Computing Environment) exige une authentification par mot de passe mais valide les mots de passe d'une façon différente du modèle utilisé dans / **etc/passwd** et **/etc/security/passwd**. Les utilisateurs qui s'authentifient via DCE doivent avoir leur strophe du fichier **/etc/security/user** définie sur `SYSTEM=DCE`.

Les autres valeurs de l'attribut **SYSTEM** sont **compat**, **files** et **NONE**. **compat** est utilisé lorsque la résolution de nom (et l'authentification qui en résulte) suit la base de données locale. Si aucune résolution n'est trouvée, une tentative est effectuée sur la base de données NIS (Network Information Services). **files** indique que seuls les fichiers locaux doivent être utilisés lors de l'authentification. Enfin, **NONE** désactive l'authentification par méthode. Pour désactiver toutes les authentifications, **NONE** doit apparaître dans les lignes **SYSTEM** et **auth1** de la strophe de l'utilisateur.

Vous pouvez définir d'autres valeurs valides de l'attribut **SYSTEM** dans **/usr/lib/security/methods.cfg**.

Remarque : L'utilisateur root est toujours authentifié au moyen d'un fichier sécurité du système local. L'attribut **SYSTEM** de l'utilisateur root est défini sur **`SYSTEM = "compat"`** dans **/etc/security/user**.

Pour de plus amples informations concernant la protection des mots de passe, reportez–vous au manuel *AIX 5L Version 5.3 System User's Guide : Operating System and Devices*.

ID de connexion

Tous les événements d'audit enregistrés pour cet utilisateur portent cet ID. Vous pouvez les examiner lorsque vous générez des enregistrements d'audit. Pour plus d'informations concernant les ID de connexion, reportez–vous au manuel *AIX 5L Version 5.3 System User's Guide : Operating System and Devices*.

Présentation du système de quotas de disque

Le système de quotas de disque permet aux administrateurs de contrôler le nombre de fichiers et de blocs de données pouvant être alloués à des utilisateurs ou groupes. Les sections qui suivent apportent des informations complémentaires sur le système de quotas de disque, son implémentation et son utilisation :

Présentation du système de quotas de disque

Le système de quotas de disque, basé sur le système Berkeley, constitue un moyen efficace de contrôler l'utilisation de l'espace disque. Le système de quotas peut être défini pour des utilisateurs individuels ou des groupes. Il est géré pour chaque système de fichiers journalisé (JFS et JFS2).

Le système de quotas de disque fixe les limites en fonction des paramètres suivants, modifiables via la commande **edquota** pour les systèmes de fichiers JFS et via la commande **j2edlimit** pour les systèmes de fichiers JFS2 :

- Seuils d'avertissement d'un utilisateur ou d'un groupe
- Seuils obligatoires d'un utilisateur ou d'un groupe
- Période de tolérance du quota

Le *seuil d'avertissement* définit le nombre de blocs de disques de 1 Ko ou de fichiers que l'utilisateur ne doit pas dépasser. Le *seuil obligatoire* définit le nombre maximum de blocs de disques ou de fichiers que l'utilisateur peut cumuler dans la limite des quotas disque établis. La *période de tolérance de quota* permet à l'utilisateur de dépasser le seuil d'avertissement pendant une courte période (la valeur par défaut est d'une semaine). Si l'utilisateur ne parvient pas à réduire son utilisation sous le seuil d'avertissement pendant la période indiquée, le système interprétera le seuil d'avertissement comme l'allocation maximum permise, et aucun espace disque supplémentaire ne sera alloué à l'utilisateur. L'utilisateur peut supprimer cette condition en supprimant un nombre suffisant de fichiers pour passer sous le seuil d'avertissement.

Le système de quotas de disque enregistre le suivi des quotas utilisateur et de groupe dans les fichiers **quota.user** et **quota.group**, dans les répertoires root des systèmes de fichiers soumis aux quotas. Ces fichiers sont créés avec les commandes **quotacheck** et **edquota** et peuvent être lus avec les commandes de quota.

Reprise après un dépassement de quota

Vous pouvez appliquer les méthodes qui suivent pour réduire l'utilisation des systèmes de fichiers lorsque vous avez dépassé les seuils de quota :

- Arrêtez le processus en cours à l'origine du dépassement de seuil, supprimez les fichiers inutiles pour passer sous le seuil autorisé et relancez le programme qui a échoué.
- Si vous exécutez un éditeur tel que vi, utilisez la séquence d'échappement du shell pour contrôler l'espace réservé aux fichiers, supprimez les fichiers inutiles et reprenez la tâche sans perdre le fichier modifié. Si vous utilisez les shells C ou Korn, une autre méthode consiste à arrêter temporairement l'éditeur à l'aide de la séquence de touches Ctrl–Z, à exécuter les commandes sur le système de fichiers, et à revenir en exécutant la commande d'avant–plan **fg**.
- Enregistrez temporairement le fichier dans un système de fichiers n'ayant pas dépassé le quota autorisé, supprimez les fichiers inutiles et replacez le fichier dans le système d'origine.

Configuration du système de quotas de disque

En principe, seuls les systèmes de fichiers qui contiennent des fichiers et répertoires personnels sont soumis aux quotas de disque. Vous devez envisager d'utiliser un système de quotas de disque dans les conditions suivantes :

- L'espace disque de votre système est limité.
- Vous souhaitez que vos systèmes de fichiers bénéficient d'un niveau de sécurité supérieur.
- Vous utilisez le disque de façon intensive, à l'exemple des universités.

Si ces conditions ne s'appliquent pas à votre environnement, vous ne souhaiterez peut–être pas limiter l'utilisation du disque via un système de quotas.

Les quotas ne sont applicables qu'aux systèmes de fichiers journalisés.

Remarque : N'établissez pas de quotas de disque pour le système de fichiers **/tmp**.

Pour paramétrer le système de quotas de disque, appliquez la procédure suivante :

1. Connectez–vous en tant qu'utilisateur root.
2. Choisissez les systèmes de fichiers auxquels appliquer les quotas.

   Remarque : N'appliquez pas de quota au système de fichiers **/tmp**, nombre d'éditeurs et d'utilitaires système créant des fichiers temporaires dans /tmp.
3. Avec la commande **chfs**, ajoutez les attributs de configuration **userquota** et **groupquota** au fichier **/etc/filesystems**. L'exemple suivant utilise la commande **chfs** pour activer les quotas utilisateur sur le système de fichiers **/home** :

```
chfs –a "quota = userquota" /home
```

   Pour activer les quotas utilisateur et groupe dans le système de fichiers **/home**, entrez :

```
chfs –a "quota = userquota,groupquota" /home
```

   Dans le fichier **/etc/filesystems**, l'entrée correspondante ressemble à :

```
/home:
dev         = /dev/hd1
vfs         = jfs
log         = /dev/hd8
mount       = true
check       = true
quota       = userquota,groupquota
options     = rw
```

4. La désignation d'autres noms de fichier de quotas disque est facultative. Les noms de fichiers **quota.user** et **quota.group** sont les noms par défaut situés dans les répertoires root des systèmes de fichiers activés avec des quotas. En outre, vous pouvez spécifier d'autres noms ou répertoires pour ces fichiers de quotas avec les attributs **userquota** et **groupquota** du fichier **/etc/filesystems**.

   L'exemple suivant utilise la commande **chfs** pour établir des quotas utilisateur et groupe pour le système de fichiers **/home** et nomme les fichiers de quota **myquota.user** et **myquota.group** :

```
chfs –a "userquota = /home/myquota.user" –a "groupquota = /home
         /myquota.group" /home
```

   Dans le fichier **/etc/filesystems**, l'entrée correspondante ressemble à :

```
/home:
dev         = /dev/hd1
vfs         = jfs
log         = /dev/hd8
mount       = true
check       = true
quota       = userquota,groupquota
userquota   = /home/myquota.user
groupquota  = /home/myquota.group
options     = rw
```

5. Si ce n'est pas déjà fait, montez les systèmes de fichiers spécifiés.
6. Définissez les limites de quota souhaitées pour chaque utilisateur ou groupe. Utilisez la commande **edquota** pour créer le seuil d'avertissement et le seuil obligatoire de chaque utilisateur ou groupe, ainsi que l'espace disque autorisé et le nombre maximum de fichiers.

   L'exemple suivant illustre les limites de quota pour l'utilisateur *davec* :

```
Quotas for user davec:
 /home: blocks in use: 30, limits (soft = 100, hard = 150)
         inodes in use: 73, limits (soft = 200, hard = 250)
```

   davec a utilisé 30 Ko sur les 100 Ko autorisés. Sur un maximum de 200 fichiers, *davec* en a créé 73. Cet utilisateur a des mémoires tampon de 50 Ko d'espace disque et 50 fichiers qui peuvent être dédiés au stockage temporaire.

   Quand vous définissez des quotas disque pour plusieurs utilisateurs, utilisez l'indicateur **–p** avec la commande **edquota** pour dupliquer les quotas pour un autre utilisateur.

   Pour dupliquer les quotas attribués à l'utilisateur *davec* au profit de l'utilisateur *nanc*, entrez :

```
edquota –p davec nanc
```

7. Activez le système de quotas avec la commande **quotaon**. Accompagnée de l'indicateur **–a**, la commande **quotaon** active les quotas pour le système de fichiers précisé, ou pour tous les systèmes de fichiers soumis aux quotas (comme indiqué dans le fichier **/etc/filesystems**).
8. Utilisez la commande **quotacheck** pour vérifier la cohérence entre les fichiers de quotas et l'utilisation en cours du disque.

   Remarque : Cette vérification est recommandée pour toute activation initiale de quotas sur un système de fichiers et après toute réinitialisation du système.

   Pour activer la vérification et les quotas au démarrage du système, ajoutez les lignes suivantes à la fin du fichier **/etc/rc** :

```
echo " Enabling filesystem quotas "
/usr/sbin/quotacheck –a
/usr/sbin/quotaon –a
```

Chapitre 3. Listes de contrôle des accès (ACL)

Une liste de contrôle des accès (ACL) est une entité associée à un objet qui permet au système d'exploitation d'appliquer le contrôle des accès à cet objet. En général, une liste ACL est constituée d'entrées de contrôle d'accès appelées entrées ACE (Access Control Entry). Chaque entrée ACE définit les droits d'accès d'un utilisateur par rapport à l'objet. Lors d'une tentative d'accès, le système d'exploitation utilise la liste ACL associée à l'objet pour déterminer si l'utilisateur dispose des droits d'accès appropriés. Les listes de contrôle des accès (ACL) et les contrôles d'accès correspondants constituent la base du mécanisme de contrôle des accès discrétionnaire (DAC, Discretionary Access Control) pris en charge par AIX.

Le système d'exploitation prend en charge différents types d'objet système qui permettent aux processus utilisateur de stocker ou de transmettre des données. Les principaux types d'objet concernés par le contrôle des accès sont les suivants :

- fichiers et répertoires,
- tubes nommés,
- objets IPC tels que les files d'attente de messages, les segments de mémoire partagée et les sémaphores.

Tous les contrôles d'accès pour ces objets sont effectués au niveau de l'appel système, au moment du premier accès aux objets. Dans la mesure où l'accès aux objets SVIPC n'est pas nominatif, les contrôles sont effectués à chaque accès. Pour les objets avec noms de systèmes de fichiers, il est nécessaire de pouvoir résoudre le nom de l'objet réel. Les noms sont résolus de façon relative (au répertoire de travail du processus) ou absolue (par rapport au répertoire root du processus). Toute résolution de nom commence par la recherche de l'un de ces répertoires.

Le mécanisme de contrôle d'accès discrétionnaire assure un contrôle effectif de l'accès aux ressources ainsi qu'une protection distincte de la confidentialité et de l'intégrité des informations. Les mécanismes de contrôle gérés par le propriétaire ne sont effectifs que s'ils sont définis par les utilisateurs. Tous les utilisateurs doivent maîtriser le mécanisme d'octroi et de refus de droits d'accès.

Par exemple, une liste ACL associée à un objet de système de fichiers (fichier ou répertoire) peut contrôler les droits d'accès de plusieurs utilisateurs sur l'objet. Ce type de liste ACL peut contrôler les droits d'accès de différents utilisateurs à différents niveaux, par exemple en lecture ou en écriture.

En général, chaque objet est associé à un propriétaire et dans certains cas, à un groupe principal. Le propriétaire d'un objet spécifique contrôle ses attributs d'accès discrétionnaire. Les attributs du propriétaire sont définis pour l'ID utilisateur effectif du processus créé.

Le tableau suivant répertorie les attributs de contrôle d'accès direct des différents types d'objet :

Propriétaire
: Pour les objets SVIPC (System V InterProcess Communication), le propriétaire peut être modifié par le créateur ou par le propriétaire. Le créateur associé à ces objets possède les mêmes droits que le propriétaire (y compris l'autorisation d'accès). Le créateur ne peut être modifié, même avec les droits d'accès root.

  Les objets SVIPC sont initialisés à l'ID groupe effectif du processus créé. Pour les objets système de fichiers, les attributs de contrôle d'accès direct sont initialisés à l'ID groupe effectif du processus créé ou à l'ID groupe du répertoire parent (ceci est défini par l'indicateur héritage du groupe du répertoire parent).

Groupe
: Le propriétaire d'un objet peut modifier le groupe. Le nouveau groupe doit être l'ID de groupe effectif du processus créé ou bien l'ID de groupe du répertoire parent. Comme indiqué plus haut, des objets SVIPC sont associés à un groupe qui ne peut pas être modifié et partagent l'autorisation d'accès du groupe d'objets.

Mode
: La commande **chmod** en mode numérique (avec notations octales) peut définir des droits d'accès et des attributs de base. La sous–routine **chmod** appelée par la commande désactive les droits d'accès étendus. Donc, si vous utilisez le mode numérique de la commande **chmod** sur un fichier doté d'une liste ACL, les droits étendus sont désactivés. Le mode symbolique de la commande **chmod** ne désactive pas ces droits. Pour obtenir des informations sur les modes numérique et symbolique, voir la description de la commande **chmod**.

De nombreux objets du système d'exploitation, tels que les sockets et les objets de système de fichiers, sont associés à des listes ACL relatives à différents sujets. Les caractéristiques des listes ACL associées à ces différents types d'objet peuvent varier.

A l'origine, AIX prenait en charge les bits de mode pour le contrôle des accès sur les objets de système de fichiers. AIX prenait également en charge une forme unique de liste ACL autour des bits de mode. Cette liste ACL constituée de bits de mode de base permettait également de définir plusieurs entrées ACE, chaque entrée ACE définissant les droits d'accès d'un utilisateur ou d'un groupe autour des bits de mode. Ce type de fonctionnement classique de liste ACL existait avant AIX 5.3 et continue d'être pris en charge. Ce type de liste ACL a été nommé type ACL AIXC.

Notez que la prise en charge d'une liste ACL sur des objets de système de fichiers dépend du système de fichiers physique (PFS) sous–jacent. Le système de fichiers physique doit être capable de comprendre les données de la liste ACL et de stocker, extraire et appliquer les droits d'accès des différents utilisateurs. Certains systèmes de fichiers physiques ne prennent en charge aucun type de liste ACL (uniquement des bits de mode de base), alors que d'autres prennent en charge différents types de liste ACL. Depuis AIX 5.3, quelques systèmes de fichiers sous AIX ont évolués pour prendre en charge différents types de liste ACL. JFS2 et GPFS peuvent également prendre en charge le type de liste ACL reposant sur le protocole NFS version 4. Sous AIX, ce type de liste ACL a été nommé type ACL NFS4. Ce type de liste ACL adhère à la plupart des définitions ACL des spécifications du protocole NFS version 4. Il prend également en charge une plus grande granularité de contrôle d'accès que le type ACL AIXC et assure des fonctions telles que l'héritage.

Les sections suivantes décrivent la prise en charge de l'infrastructure de listes ACL sous AIX, en expliquant notamment la gestion des listes ACL.

Prise en charge d'une structure contenant plusieurs types ACL

Depuis la version 5.3.0, AIX prend en charge une infrastructure acceptant différents types ACL pour différents objets de système de fichiers d'un système d'exploitation. Cette infrastructure permet de gérer les listes ACL de façon uniforme, indépendamment du type ACL associé à l'objet. La structure comprend les éléments suivants :

Commandes d'administration ACL
: Il s'agit de commandes telles que **aclget**, **aclput**, **acledit**, **aclconvert** et **aclgetttypes**. Ces commandes appellent les interfaces de bibliothèque qui appellent les modules spécifiques aux différents types de liste ACL.

Interfaces de bibliothèque ACL
: Une interface de bibliothèque ACL agit comme une interface frontale pour les applications qui accèdent aux listes ACL.

Modules ACL de chargement dynamique spécifiques au type ACL
: AIX fournit un ensemble de modules spécifiques au type ACL pour les listes ACL AIX classiques (**AIXC**) et les listes ACL NFS4 (**nfs4**).

Compatibilité binaire

Le fonctionnement des applications sur les systèmes de fichiers JFS2 existants ne présente aucun problème de compatibilité, avec ou sans utilisation des listes ACL AIX existantes. Cependant, l'accès des applications aux fichiers peut échouer si l'objet de système de fichiers demandé est associé à une liste ACL plus stricte (par exemple NFS4). Une simple vérification dans le but de déterminer si le fichier existe nécessite un droit d'accès en écriture dans la liste ACL NFS4.

Types de liste ACL pris en charge sous AIX

Actuellement, AIX prend en charge les listes ACL de type AIXC et NFS4. Comme indiqué plus haut, il prend également en charge une infrastructure permettant d'ajouter tout autre type de liste ACL pris en charge par le système de fichiers physique sous–jacent. Notez que le système de fichiers physique JFS2 prend en charge le type ACL NFS4 de façon native si l'instance du système de fichiers est créée avec la fonction d'attributs étendus Extended Attributes Version 2.

Listes ACL AIXC

Le type ACL AIXC correspond au type ACL pris en charge dans les versions AIX antérieures à 5.3.0. Les listes ACL AIXC comportent des droits d'accès de base et des droits d'accès étendus. Avec le système de fichiers JFS2, la taille des listes ACL AIXC peut atteindre 4 ko.

Droits d'accès de base

Les droits d'accès de base sont constitués des droits d'accès attribués normalement au propriétaire du fichier, au groupe associé et aux autres utilisateurs. Il s'agit des droits d'accès : en lecture (r), en écriture (w) et en exécution/recherche (x).

Dans une liste de contrôle des accès, les droits d'accès de base sont au format suivant, le paramètre *Mode* étant exprimé sous la forme rwx (un tiret indiquant l'absence de droit) :

```
base permissions:
    owner(name):   Mode
    group(group):   Mode
    others:   Mode
```

Attributs

Les attributs suivants peuvent être ajoutés à une liste de contrôle des accès AIXC :

setuid (SUID) Le bit de mode Set–user–ID. Cet attribut définit les ID utilisateur enregistré et effectif du processus, avec l'ID du propriétaire du fichier exécuté.

setgid (SGID) Le bit de mode Set–group–ID. Cet attribut définit les ID groupe enregistrés et effectifs du processus, avec l'ID du groupe du fichier exécuté.

savetext (SVTX)
Indique que seuls les propriétaires de fichiers peuvent créer ou supprimer des liens entre fichiers, dans le répertoire indiqué.

Ces attributs sont ajoutés au format suivant :

```
attributes: SUID, SGID, SVTX
```

Droits d'accès étendus

Les droits d'accès étendus sont un moyen pour le propriétaire d'un fichier d'affiner les droits d'accès à ce fichier. Ils permettent de modifier les droits de base (utilisateur, groupe et autres) en accordant, en supprimant ou en spécifiant des droits spécifiques pour des individus, des groupes ou des combinaisons de groupes. Les droits d'accès sont modifiés à l'aide de mots clés.

Les mots clés **permit**, **deny** et **specify** sont définis comme suit :

permit	Accorde à l'utilisateur ou au groupe l'accès au fichier spécifié
deny	Retire à l'utilisateur ou au groupe l'accès au fichier spécifié
specify	Définit précisément l'accès au fichier de l'utilisateur ou du groupe

Si un droit est refusé à un utilisateur par le biais de **deny** ou de **specify**, ce refus ne peut être rétabli par aucune autre entrée.

La liste de contrôle des accès (ACL) doit être activée (mot clé **enabled**) pour que les droits étendus prennent effet. La valeur par défaut est **disabled**.

Dans une liste ACL, les droits étendus apparaissent sous la forme :

```
extended permissions:
   enabled | disabled
     permit    Mode  UserInfo...
     deny      Mode  UserInfo...
     specify   Mode  UserInfo...
```

Placez chacune des entrées **permit**, **deny** et **specify** sur une ligne distincte. Le paramètre *Mode* est sous la forme **rwx** (un tiret indiquant l'absence de droit). Le paramètre *UserInfo* est sous la forme `u:UserName` ou `g:GroupName`, ou encore par une combinaison séparée par une virgule de `u:UserName` et `g:GroupName`.

Remarque : Comme un processus est associé à un seul ID utilisateur, si vous indiquez plusieurs noms d'utilisateur dans une entrée, celle–ci ne peut pas être utilisée dans une décision de contrôle d'accès.

Représentation textuelle

La strophe suivante contient une représentation textuelle d'une liste ACL AIXC :

```
Attributes:  { SUID | SGID | SVTX }
Base Permissions:
    owner(name):   Mode
    group(group):   Mode
    others:   Mode
Extended Permissions:
    enabled | disabled
      permit    Mode  UserInfo...
      deny      Mode  UserInfo...
      specify   Mode  UserInfo...
```

Format binaire

Le format binaire des listes ACL AIXC est défini dans **/usr/include/sys/acl.h** et il est mis en oeuvre dans la version AIX actuelle.

Exemple de liste ACL AIXC

Voici un exemple de liste ACL pour AIXC :

```
attributes: SUID
base permissions:
       owner(frank):  rw-
       group(system): r-x
    others: ---
extended permissions:
       enabled
         permit  rw-  u:dhs
         deny    r--  u:chas, g:system
         specify r--  u:john, g:gateway, g:mail
         permit  rw-  g:account, g:finance
```

Description des entrées de la liste ACL :

- La première ligne indique que le bit **setuid** est activé.
- La deuxième ligne présente les droits d'accès de base (elle est facultative).
- Les trois lignes suivantes précisent ces droits. Les noms du propriétaire et du groupe (entre parenthèses) sont donnés à titre d'information : La modification de ces noms n'a pas d'incidence sur le propriétaire réel du fichier, pas plus que sur le groupe auquel appartient le fichier. Seules les commandes **chown** et **chgrp** permettent de modifier ces attributs.
- La ligne suivante, qui présente les droits d'accès étendus, est facultative.
- La ligne suivante indique que les droits d'accès étendus qui suivent sont activés.
- Les quatre dernières lignes correspondent aux entrées étendues. La première entrée étendue accorde à l'utilisateur `dhs` l'accès en lecture (r) et en écriture (w) au fichier.
- La deuxième interdit l'accès en lecture (r) à l'utilisateur `chas` lorsqu'il est membre du groupe `system`.
- La troisième accorde à l'utilisateur `john` l'accès en lecture (r) tant qu'il est membre des deux groupes `gateway` et `mail`. Si l'utilisateur `john` n'appartient pas à ces deux groupes, l'accès lui est refusé.
- La dernière ligne, enfin, accorde à tout utilisateur membre des *deux* groupes `account` et `finance` l'accès en lecture (r) et en écriture (w).

  Remarque : Plusieurs entrées étendues peuvent s'appliquer à un processus qui demande l'accès à un objet contrôlé, les entrées restrictives ont la priorité sur les modes permissifs.

  Pour obtenir des détails sur la syntaxe, reportez–vous à la description de la commande **acledit** dans le manuel *AIX 5L Version 5.3 Commands Reference*.

Listes ACL NFS4

AIX prend également en charge les listes ACL de type NFS4. Le type de liste ACL NFS4 met en oeuvre le contrôle des accès décrit dans *RFC 3530 du protocole NFS (Network File System) version 4*. Avec le système de fichiers JFS2, la taille des listes ACL NFS4 peut atteindre 64 ko.

Représentation textuelle

Une liste ACL NFS V4 textuelle est une liste d'entrées ACE (Access Control Entries) contenant une entrée par ligne. Une entrée ACE comporte quatre éléments au format suivant :

```
IDENTITY    ACE_TYPE      ACE_MASK     ACE_FLAGS

where:
IDENTITY => Has format of 'IDENTITY_type:(IDENTITY_name or IDENTITY_ID or
IDENTITY_who):'
          where:
          IDENTITY_type => One of the following Identity type:
                  u : user
                  g : group
                  s : special who string (IDENTITY_who must be a special
who)
                          IDENTITY_name=> user/group name
                          IDENTITY_ID   => user/group ID
                          IDENTITY_who  => special who string (e.g. OWNER@,
GROUP@, EVERYONE@)
ACE_TYPE => One of the following ACE Type:
                 a : allow
                 d : deny
                 l : alarm
                  u : audit
ACE MASK => One or more of the following Mask value Key without separator:
                 r : READ_DATA                    or LIST_DIRECTORY
                 w : WRITE_DATA      or ADD_FILE
                 p : APPEND_DATA     or ADD_SUBDIRECTORY
                 R : READ_NAMED_ATTRS
                 W : WRITE_NAMED_ATTRS
                 x : EXECUTE         or SEARCH_DIRECTORY
                 D : DELETE_CHILD
                 a : READ_ATTRIBUTES
                 A : WRITE_ATTRIBUTES
                 d : DELETE
                 c : READ_ACL
                 C : WRITE_ACL
                 o : WRITE_OWNER
                 s : SYNCHRONIZE
ACE_FLAGS (Optional) => One or more of the following Attribute Key without
separater:
                 fi : FILE_INHERIT
                 di : DIRECTORY_INHERIT
                 oi : INHERIT_ONLY
                 ni : NO_PROPAGATE_INHERIT
                 sf : SUCCESSFUL_ACCESS_ACE_FLAG
                 ff : FAILED_ACCESS_ACE_FLAG

Example:
         u:user1(aa@ibm.com):     a    rwp    fidi
         *s:(OWNER@):             d    x      dini          * This line is
a comment
         g:staff(jj@jj.com):      a    rx
         s:(GROUP@):              a    rwpx   fioi
         u:2:                     d    r     di             * This line
shows user bin (uid=2)
         g:7:                     a    ac    fi             * This line
shows group security (gid=7)
         s:(EVERYONE@):           a    rca   ni
```

Format binaire

Le format binaire des listes ACL NFS4 est défini dans **/usr/include/sys/acl.h** et il est mis en oeuvre dans la version AIX actuelle.

Exemple de liste ACL NFS4

L'exemple suivant représente une liste ACL NFS4 appliquée à un répertoire (tel que **j2eav2/d0**) :

```
s:(OWNER@):          a          rwpRWxDdo          difi          * 1st   ACE
s:(OWNER@):          d          D                  difi          * 2nd  ACE
s:(GROUP@):          d          x                  ni            * 3rd   ACE
s:(GROUP@):          a          rx                 difi          * 4th   ACE
s:(EVERYONE@):       a          c                  difi          * 5th   ACE
s:(EVERYONE@):       d          C                  difi          * 6th   ACE
u:user1:             a          wp                 oi            * 7th   ACE
g:grp1:              d          wp                               *  8th  ACE
u:101:               a          C                                * 9th   ACE
g:100:               d          c                                * 10th  ACE
```

Description des entrées de la liste ACL :

- La première entrée ACE indique que le propriétaire dispose des droits d'accès suivants sur **/j2eav2/d0** et sur tous les enfants créés après l'application de cette liste ACL :
  - **READ_DATA (= LIST_DIRECTORY)**
  - **WRITE_DATA (=ADD_FILE)**
  - **APPEND_DATA (= ADD_SUBDIRECTORY)**
  - **READ_NAMED_ATTR**
  - **WRITE_NAMED_ATTR**
  - **EXECUTE (=SEARCH_DIRECTORY)**
  - **DELETE_CHILD**
  - **DELETE**
  - **WRITE_OWNER**
- La seconde entrée ACE indique que le propriétaire ne dispose pas du droit **DELETE_CHILD** (suppression de fichiers ou de sous–répertoires créés dans **/j2eav2**), mais il peut les supprimer car la première entrée ACE lui donne le droit **DELETE_CHILD**.
- La troisième entrée ACE indique que tous les membres du groupe de l'objet (**/j2eav2/d0**) ne disposent pas du droit **EXECUTE (=SEARCH_DIRECTORY)**, mais la première entrée ACE accorde ce droit au propriétaire. Cette entrée ACE ne peut pas être propagée à tous les enfants, car l'indicateur **NO_PROPAGATE_INHERIT** est défini. Cette entrée ACE est appliquée uniquement au répertoire **/j2eav2/d0** ainsi qu'à ses fichiers et sous–répertoires enfants directs.
- La quatrième entrée ACE indique que chaque membre du groupe de l'objet (**/j2eav2/d0**) dispose du droit **READ_DATA (= LIST_DIRECTORY)** et **EXECUTE (=SEARCH_DIRECTORY)** sur le répertoire /j2eav2/d0 et tous ses enfants. Cependant, selon la troisième entrée ACE, les membres du groupe (excepté le propriétaire) ne disposent pas du droit **EXECUTE (=SEARCH_DIRECTORY)** sur le répertoire **/j2eav2/d0** et tous les fichiers et répertoires enfants directs.
- La cinquième entrée ACE indique que tous disposent du droit d'accès **READ_ACL** sur le répertoire **/j2eav2/d0** et sur tout enfant créé après l'application de cette liste ACL.
- La sixième entrée ACE indique qu'aucun ne dispose du droit d'accès **WRITE_ACL** sur le répertoire **/j2eav2/d0** et tous les enfants. Le propriétaire dispose toujours du droit **WRITE_ACL** sur les fichiers et répertoires associés à une ACL NFS4.
- La septième entrée ACE indique que l'utilisateur 1 dispose des droits **WRITE_DATA (=ADD_FILE)** et **APPEND_DATA (= ADD_SUBDIRECTORY)** sur tous les enfants du répertoire **/j2eav2/d0** mais pas sur le répertoire **/j2eav2/d0** lui–même.

- La huitième entrée ACE indique qu'aucun membre du groupe 1 ne dispose des droits **WRITE_DATA (=ADD_FILE)** et **APPEND_DATA (= ADD_SUBDIRECTORY)**. Cette liste ACE ne s'applique pas au propriétaire même s'il appartient au groupe 1, à cause de la première entrée ACE.
- La neuvième entrée ACE indique que l'utilisateur identifié par l'**UID 101** dispose du droit **WRITE_ACL**, mais aucun, excepté le propriétaire, ne dispose du droit **WRITE_ACL**, à cause de la sixième entrée ACE.
- La dixième entrée ACE indique que le droit **READ_ACL** est refusé à tous les membres du groupe identifié par le **GID 100**, mais ceux–ci disposent de ce droit grâce à la cinquième entrée ACE.

Gestion des listes ACL

Cette section décrit les méthodes de gestion des listes ACL. Pour visualiser et définir les listes ACL, les utilisateurs AIX peuvent utiliser Web–based System Manager ou bien exécuter des commandes. Les programmeurs d'applications et les développeurs de sous–systèmes peuvent utiliser les interfaces de bibliothèque ACL et les routines de conversion ACL décrites dans cette section.

Commandes d'administration des listes ACL

Pour gérer et utiliser les listes ACL d'un objet de système de fichiers, vous pouvez utiliser les commandes suivantes.

aclget Ecrit la liste ACL de l'objet fichier nommé **FileObject** dans un format standard lisible ou bien dans un fichier de sortie nommé **outAclFile**.

aclput Définit la liste ACL de l'objet fichier **FileObject** sur le système de fichiers selon l'entrée standard définie ou **inAclFile**.

acledit Ouvre un éditeur pour modifier la liste ACL de l'objet fichier **FileObject** défini.

aclconvert Convertit une liste ACL d'un type vers un autre. Cette commande échoue si la conversion n'est pas prise en charge.

aclgettypes Obtention de tous les types ACL pris en charge par un chemin de système de fichiers.

Interfaces de bibliothèque ACL

Une interface de bibliothèque ACL agit comme une interface frontale pour les applications qui accèdent aux listes ACL. Les applications (y compris les commandes génériques d'administration des listes ACL indiquées ci–dessus) n'appellent pas directement les commandes syscall ACL non documentées : elles accèdent aux commandes syscall génériques et aux modules spécifiques aux types via les interfaces de bibliothèque. Cela évite aux programmeurs d'applications d'utiliser des modules complexes à gérer et réduit les problèmes de compatibilité binaire amont avec les futures versions d'AIX.

Les interfaces de bibliothèque suivantes appellent des commandes syscall.

aclx_fget et aclx_get
: Les fonctions **aclx_get** et **aclx_fget** récupèrent les informations de contrôle des accès relatives à un objet de système de fichiers et les placent dans la région mémoire définie par **acl**. La taille et le type de **acl** sont stockés dans ***acl_sz** et ***acl_type**.

aclx_fput et aclx_put
: Les fonctions **aclx_put** et **aclx_fput** stockent les informations de contrôle d'accès définies par **acl** pour un objet fichier d'entrée. Ces fonctions n'effectuent pas de conversion de type ACL. Pour effectuer cette conversion, l'appelant doit appeler explicitement la fonction **aclx_convert**.

aclx_gettypes La fonction **aclx_gettypes** récupère la liste des types ACL pris en charge sur le système de fichiers particulier. Un type de système de fichiers peut prendre en charge plusieurs types ACL simultanément. Chaque objet de système de fichiers est associé à un type ACL unique inclus à la liste des types d'ACL pris en charge par le système de fichiers.

aclx_gettypeinfo
La fonction **aclx_gettypeinfo** récupère les caractéristiques et les fonctionnalités d'un type ACL du système de fichiers défini par le chemin. Notez que les caractéristiques ACL appartiennent en général à un type de structure de données spécifique à chaque type ACL particulier. Les structures de données utilisées pour les listes ACL AIXC et NFS4 sont décrites dans un autre document.

aclx_print et aclx_printStr
Ces deux fonctions convertissent la liste ACL au format binaire pour obtenir une représentation textuelle. Ces fonctions sont appelées par les commandes **aclget** et **acledit**.

aclx_scan et aclx_scanStr
Ces deux fonctions convertissent la représentation textuelle de la liste ACL en format binaire.

aclx_convert Convertit une liste ACL d'un type en un autre type. Cette fonction permet la conversion implicite par des commandes telles que **cp**, **mv** ou **tar**.

Conversion de listes ACL

La conversion de liste ACL permet de convertir un type ACL en un autre type. La prise en charge de plusieurs types ACL dépend des types ACL pris en charge sur un système de fichiers physique particulier. Tous les types ACL ne sont pas pris en charge sur tous les systèmes de fichiers. Par exemple, un système de fichiers peut prendre en charge uniquement les types de liste ACL AIXC et un autre système de fichiers peut prendre en charge les types de liste ACL AIXC et NFS4. Vous pouvez copier les listes ACL AIXC entre les deux systèmes de fichiers, mais vous devez utiliser la conversion de liste ACL pour copier des listes ACL NFS du second système de fichiers vers le premier. La conversion de liste ACL conserve autant que possible les informations de contrôle des accès.

Remarque : Le processus de conversion est approximatif et peut entraîner la perte d'informations de contrôle d'accès. Tenez compte de cet aspect lors de la planification de conversions de listes ACL.

La conversion de listes ACL sous AIX est prise en charge avec l'infrastructure suivante :

Routines de bibliothèque
Ces routines et la structure des listes ACL au niveau utilisateur permettent la conversion des listes ACL d'un type vers un autre.

Commande **aclconvert**
Cette commande convertit les listes ACL.

Commandes **aclput** et **acledit**
Ces commandes permettent de modifier les types de liste ACL.

Commandes **cp** et **mv**
Ces commandes permettent de gérer plusieurs types de liste ACL et d'effectuer la conversion interne de liste ACL, si nécessaire.

Commande **backup**
Cette commande convertit les informations des listes ACL vers un type et un format connus (type ACL AIXC) lorsque la sauvegarde doit être effectuée au format d'origine. Pour récupérer la liste ACL au format natif, définissez l'option **–U**. Pour en savoir plus, voir la description de la commande backup.

Chaque type ACL est unique et l'affinement des masques de contrôle d'accès varie beaucoup d'un type ACL à un autre. Les algorithmes de conversion sont approximatifs et ne sont pas équivalents à la conversion manuelle d'une liste ACL. Dans certains cas, la conversion est inexacte. Par exemple, les listes ACL NFS4 ne peuvent pas être converties de façon exacte en liste ACL AIXC car les listes ACL NFS4 contiennent jusqu'à 16 masques d'accès et des fonctions d'héritage non prises en charge par le type ACL AIXC. Si vous utilisez les fonctions et les interfaces de conversion ACL, tenez compte de la perte d'informations de contrôle d'accès.

Remarque : Les algorithmes de conversion ACL sont soumis à des droits de propriété et peuvent être modifiés.

Bits S et listes ACL

Utilisation des programmes setuid et setgid

Le mécanisme de bits d'autorisation permet le contrôle d'accès effectif des ressources dans la plupart des situations. Pour un contrôle plus précis, le système d'exploitation propose les programmes **setuid** et **setgid**.

AIX définit l'identité uniquement sous la forme UID et GID. Les types ACL qui ne définissent pas l'identité sous la forme UID ou GID sont mappés sur le modèle d'identité AIX. Par exemple, le type ACL NFS4 définit l'identité de l'utilisateur sous forme de chaîne de type utilisateur@domaine, laquelle est mappée sur les UID et GID numériques.

La plupart des programmes sont exécutés avec les droits d'accès utilisateur et groupe d'utilisateur de l'utilisateur qui les a appelés. Les propriétaires de programmes peuvent associer les droits d'accès de l'utilisateur qui a appelé ces programmes en transformant ces derniers en programmes **setuid** ou **setgid**, c'est–à–dire en définissant dans leur zone d'autorisation le bit setuid ou setgid. Quand le processus exécute le programme, il obtient les droits d'accès du propriétaire du programme. Un programme **setuid** s'exécute avec les droits d'accès de son propriétaire, tandis qu'un programme **setgid** a les droits d'accès de son groupe ; les deux bits peuvent être définis en fonction du mécanisme d'autorisation.

Bien que les droits d'accès supplémentaires soient attribués au processus, ils sont contrôlés par le programme qui les possède. Ainsi, **setuid** et **setgid** permettent les contrôles d'accès programmés par l'utilisateur, dans lesquels les droits d'accès sont attribués indirectement. Le programme fonctionne comme un sous–système sécurisé, surveillant les droits d'accès utilisateur.

Les programmes setuid et setgid sont très efficaces, mais ils peuvent mettre la sécurité en danger s'ils ne sont pas soigneusement mis en œuvre. Notamment, le programme ne doit jamais renvoyer le contrôle à l'utilisateur s'il détient toujours les droits d'accès de son propriétaire, ce qui permettrait à l'utilisateur d'utiliser sans restriction les droits du propriétaire.

Remarque : Pour des raisons de sécurité, le système d'exploitation interdit les appels **setuid** ou **setgid** depuis un script shell.

Application des bits S aux listes ACL

Notez que les listes ACL telles que NFS4 n'ont pas de lien direct avec les bits S. Une liste ACL NFS4 ne définit pas la manière dont ces bits sont insérés dans la liste ACL. Dans l'approche de ce problème sous AIX, les bits S sont utilisés lors des contrôles d'accès et complètent tous les contrôles d'accès liés aux listes ACL NFS4. La commande **chmod** sous AIX peut être utilisée pour définir ou réinitialiser les bits S sur les objets de système de fichiers avec les listes ACL telles que NFS4.

Droits d'accès d'administration

Le système d'exploitation fournit des droits d'accès privilégiés pour l'administration du système. Les privilèges système sont fondés sur les ID utilisateur et groupe. Sont reconnus privilégiés les utilisateurs dont l'ID utilisateur ou groupe effectif est défini à 0.

Les processus avec un ID utilisateur effectif à 0 sont des processus réputés utilisateur root qui peuvent :

- Lire ou écrire tout objet
- Appeler toute fonction système
- Effectuer certains contrôles de sous–système avec les programmes **setuidroot**.

Il existe deux types de privilèges pour l'administration du système : le privilège de la commande **su** et celui du programme **setuid–root**. La commande **su** permet à tous les programmes que vous appelez de fonctionner en tant que processus réputés utilisateur root. La commande **su** offre une méthode flexible d'administration du système mais n'est pas très sûre.

Passer un programme en programme **setuid–root** signifie que le programme appartient à un utilisateur root avec le bit setuid défini. Un programme **setuid–root** offre des fonctions administratives à tout utilisateur sans compromettre la sécurité ; le privilège n'est pas directement accordé à l'utilisateur, il est encapsulé dans le programme. Encapsuler toutes les fonctions administratives nécessaires dans des programmes **setuid–root** n'est pas un procédé particulièrement simple, mais il est sûr.

Autorisations d'accès

Lorsqu'un utilisateur se connecte à un compte (via une commande **login** ou **su**), les ID utilisateur et groupe attribués au compte sont associés aux processus de cet utilisateur et en déterminent les droits d'accès. Ces ID déterminent les droits d'accès du processus.

Un processus doté de l'ID utilisateur 0 est dit *processus utilisateur root*. Ces processus ont généralement tous les droits d'accès. Cependant, si un processus utilisateur root demande l'autorisation d'exécution pour un programme, l'accès ne lui est accordé que si la permission d'exécution est accordée à un utilisateur au moins.

Autorisation d'accès pour les listes ACL AIXC

Le propriétaire de la ressource d'informations est responsable de la gestion des droits d'accès. Les ressources sont protégées par des *bits d'autorisation*, intégrés au mode de l'objet. Ces bits définissent les droits du propriétaire de l'objet, ceux du groupe correspondant et ceux de la classe par défaut `others` (autres). Le système d'exploitation gère trois droits d'accès (lecture, écriture et exécution), qui peuvent être accordés séparément.

Pour les fichiers, les répertoires, les tubes nommés, et les unités (fichiers spéciaux), les accès sont autorisés comme suit :

- Pour chaque entrée (ACE) de la liste de contrôle des accès (ACL), la liste des identificateurs est comparée aux identificateurs du processus. Si elles sont identiques, le processus se voit attribuer les autorisations et les restrictions définies pour cette entrée. Les correspondances logiques pour les autorisations comme pour les restrictions sont calculées pour chaque entrée concernée de l'ACL. Si le processus demandeur ne correspond à aucune entrée de l'ACL, il se voit attribuer les autorisations et les restrictions de l'entrée par défaut.

- L'accès est accordé si le droit d'accès demandé est autorisé (inclus dans l'union des autorisations) et non restreint (inclus dans l'union des restrictions). Sinon, il est refusé.

La liste des identificateurs d'une ACL correspond à un processus si tous ses identificateurs correspondent à l'identificateur effectif – de même type – du processus demandeur. Un identificateur de type USER correspond à un processus s'il est identique à l'ID utilisateur effectif du processus. Un identificateur de type GROUP correspond s'il est identique à l'ID groupe effectif du processus ou à l'un des ID des groupes complémentaires. Ainsi, une liste ACE avec la liste d'identificateurs suivante :

```
USER:fred, GROUP:philosophers, GROUP:software_programmer
```

correspondra au processus dont l'ID utilisateur est `fred` et dont l'ensemble de groupes est :

```
philosophers, philanthropists, software_programmer, doc_design
```

mais pas au processus dont l'ID utilisateur est `fred` et dont l'ensemble de groupes est :

```
philosophers, iconoclasts, hardware_developer, graphic_design
```

Notez qu'une ACE avec la liste suivante correspond aux deux processus :

```
USER:fred, GROUP:philosophers
```

En d'autres termes, la liste d'identificateurs de l'ACE fonctionne comme un ensemble de conditions, à respecter pour que l'accès spécifié soit accordé.

Tous les contrôles d'accès pour ces objets sont effectués au niveau de l'appel système, au moment du premier accès aux objets. Dans la mesure où l'accès aux objets SVIPC n'est pas nominatif, les contrôles sont effectués à chaque accès. Pour les objets avec noms de systèmes de fichiers, il est nécessaire de pouvoir résoudre le nom de l'objet réel. Les noms sont résolus de façon relative (au répertoire de travail du processus) ou absolue (par rapport au répertoire root du processus). Toute résolution de nom commence par la recherche de l'un de ces répertoires.

Le mécanisme de contrôle d'accès discrétionnaire assure un contrôle effectif de l'accès aux ressources ainsi qu'une protection distincte de la confidentialité et de l'intégrité des informations. Les mécanismes de contrôle gérés par le propriétaire ne sont effectifs que s'ils sont définis par les utilisateurs. Tous les utilisateurs doivent maîtriser le mécanisme d'octroi et de refus de droits d'accès.

Autorisation d'accès pour les listes ACL NFS4

Tout utilisateur disposant du droit **WRITE_ACL** peut gérer les droits d'accès. Le propriétaire de la source d'informations a toujours le droit **WRITE_ACL**. Pour les fichiers et les répertoires associés à des listes ACL NFS4, les autorisations d'accès sont définies comme suit :

- La liste des entrées ACE est traitée dans l'ordre et seules les entrées ACE pour lesquelles "who" (Identity) correspond au demandeur peuvent être traitées. Les données d'identification du demandeur ne sont pas vérifiées lors du traitement de l'entrée ACE pour laquelle "special who" correspond à EVERYONE@.
- Chaque entrée ACE est traitée jusqu'à ce que tous les bits de l'accès du demandeur aient été autorisés. Après avoir été autorisé, le bit ne fait plus partie du traitement des autres entrées ACE.
- Si un bit correspondant à l'accès du demandeur est refusé, l'accès est refusé et les autres entrées ACE ne sont pas traitées.
- Si tous les bits de l'accès du demandeur n'ont pas été autorisés et s'il ne reste plus aucune entrée ACE à traiter, l'accès est refusé.

Si l'accès demandé est refusé par les entrées ACE et si l'utilisateur demandeur est un superutilisateur ou un utilisateur root, l'accès est en général autorisé. Notez que le propriétaire de l'objet dispose toujours des droits **READ_ACL**, **WRITE_ACL**, **READ_ATTRIBUTES** et **WRITE_ATTRIBUTES**. Pour plus d'informations sur l'algorithme d'autorisation d'accès, voir l'Exemple de liste ACL NFS4, page 3-7.

Résolution des erreurs liées aux listes ACL

Application d'une liste ACL NFS4 sur un objet en échec

Vous pouvez utiliser le code de retour ou la fonction de suivi pour résoudre les erreurs liées à la configuration d'une liste ACL NFS4 sur un objet, tel qu'un fichier ou un répertoire. Dans les deux méthodes, les commandes **aclput** et **acledit** sont utilisées pour identifier l'origine de l'erreur.

Résolution d'erreur à l'aide du code de retour

Pour afficher le code de retour, utilisez la commande `echo $?` après l'exécution de la commande **aclput**. Les listes suivantes répertorient les codes de retour et leur description :

22 (EINVAL, défini dans **/usr/include/sys/errno.h**)

Les causes possibles de cette erreur sont :

- Format textuel non valide dans l'une des 4 zones.
- La taille de la liste ACL NFS4 d'entrée est supérieure à 64 Ko.
- La liste ACL est appliquée sur un fichier qui contient au moins une entrée ACE dont le masque ACE est défini sur **w** (**WRITE_DATA**) mais pas sur **p** (**APPEND_DATA**), ou est défini sur **p** (**APPEND_DATA**) mais pas sur **w** (**WRITE_DATA**).
- La liste ACL est appliquée sur un répertoire qui contient au moins une entrée ACE dont le masque ACE est défini sur **w** (**WRITE_DATA**) mais pas sur **p** (**APPEND_DATA**), ou bien sur **p** (**APPEND_DATA**) mais pas sur **w** (**WRITE_DATA**), et l'indicateur ACE **fi** (**FILE_INHERIT**).
- Dans au moins une entrée ACE, la valeur de **special who** (**Identity**) est OWNER@ et un ou plusieurs masques ACE **c** (**READ_ACL**), **C** (**WRITE_ACL**), **a** (**READ_ATTRIBUTE**) et **A** (**WRITE_ATTRIBUTE**) sont refusés par le type d'entrée ACE **d**.

124 (ENOTSUP, défini dans **/usr/include/sys/errno.h**)

Les causes possibles de cette erreur sont :

- Dans l'une des entrées ACE, la valeur de "special who" n'est peut–être pas l'une des trois valeurs possibles (**OWNER@**, **GROUP@** ou **EVERYONE@**).
- Dans au moins une entrée ACE, le type ACE est **u** (**AUDIT**) ou **l** (**ALARM**).

13 (EACCES, défini dans **/usr/include/sys/errno.h**)

Les causes possibles de cette erreur sont :

- Vous n'avez pas l'autorisation de lire le fichier d'entrée contenant les entrées ACE NFS4.
- Vous n'êtes pas autorisé à effectuer une recherche dans le répertoire parent de l'objet cible car vous ne disposez pas du droit **x** (**EXECUTE**) sur ce répertoire.
- Vous n'êtes pas autorisé à écrire ou à modifier la liste ACL. Si l'objet est déjà associé à une liste ACL NFS4, vérifiez que vous disposez du droit **C** (**WRITE_ACL**) pour le masque ACE.

Résolution d'erreur à l'aide de la fonction de suivi

Vous pouvez également rechercher la cause d'une erreur en générant un rapport de suivi. Le scénario suivant montre comment utiliser les informations de suivi pour identifier l'origine d'un incident dans le cas d'une liste ACL NFS4. Par exemple, vous avez le fichier **/j2v2/file1** avec la liste ACL NFS4 suivante :

```
 s:(EVERYONE@):  a       acC
```

et la liste ACL suivante est contenue dans le fichier d'entrée **input_acl_file** :

```
s:(EVERYONE@):  a       rwxacC
```

Pour résoudre l'incident à l'aide de la fonction de suivi, procédez comme suit :

1. Lancez le suivi **aclput** et **trcrpt** à l'aide des commandes suivantes :

```
$ trace -j  478 -o trc.raw
$->!aclput -i input_acl_file -t NFS4 /j2v2/file1
$ ->quit
$ trcrpt trc.raw > trc.rpt
```

2. Analysez le rapport de suivi. Si la liste ACL est appliquée à un fichier ou un répertoire, elle vérifie l'accès en écriture ou en modification de la liste ACL, puis applique la liste ACL. Le fichier contient des lignes du type suivant :

```
478 xxx  xxx  ACL ENGINE: chk_access entry: type=NFS4 obj_mode=33587200
size=68 ops=16384 uid=100

478 xxx  xxx  ACL ENGINE: chk_access exit: type=NFS4 rc=0 ops=16384 priv=0
against=0
478 xxx  xxx  ACL ENGINE: set_acl entry: type=NFS4 ctl_flg=2
obj_mode=33587200 mode=0 size=48

478 xxx  xxx  ACL ENGINE: validate_acl: type=NFS4 rc=22 ace_cnt=1
acl_len=48 size=12
478 xxx  xxx  ACL ENGINE: set_acl exit: type=NFS4 rc=22 obj_mode=33587200
size=68 cmd=536878912
```

   La seconde ligne, qui contient **chk_access exit**, indique que l'accès en écriture sur l'ACL est autorisé (**rc = 0**). La quatrième ligne, qui contient **validate_acl**, et la cinquième ligne, qui contient **set_acl exit**, indiquent que la liste ACL n'est pas appliquée avec succès (**rc=22** indique **EINVAL**). La quatrième ligne, qui contient **validate_acl**, indique la présence d'une erreur dans la première ligne de l'entrée ACE (**ace_cnt=1**). La première entrée ACE, `s:(EVERYONE@): a rwxacC)`, ne contient aucun masque d'accès **p**. **p** est nécessaire en plus de **w** pour appliquer la liste ACL.

Interdictions d’accès

Une opération de système de fichiers (par exemple, lecture ou écriture) peut échouer sur un objet associé à une liste ACL NFS4. En général, un message d’erreur s’affiche, mais celui–ci ne contient pas toujours les informations nécessaires à déterminer l’origine de l’erreur d’accès. La fonction de suivi peut vous aider à identifier la cause de l’erreur d’accès. Par exemple, vous avez le fichier **/j2v2/file2** avec la liste ACL NFS4 suivante :

```
s:(EVERYONE@):  a       rwpx
```

Le résultat de la commande suivante est **Permission denied** :

```
ls –l /j2v2/file2
```

Pour résoudre cette erreur, procédez comme suit :

1. Lancez le suivi **ls –l /j2v2/file2** et **trcrpt** à l’aide des commandes suivantes :

```
$ trace –j  478 –o trc.raw
$–>!ls –l /j2v2/file2
$ –>quit
$ trcrpt trc.raw > trc.rpt
```

2. Analysez le rapport de suivi. Le fichier contient des lignes du type suivant :

```
478  xxx  xxx   ACL ENGINE: chk_access entry: type=NFS4 obj_mode=33587711
size=68 ops=1024 uid=100
478  xxx  xxx   ACL ENGINE: nfs4_chk_access_self: type=NFS4 aceN=1 aceCnt=1
req=128 deny=0
478  xxx  xxx   ACL ENGINE: nfs4_mask_privcheck: type=NFS4 deny=128
priv=128
478  xxx  xxx   ACL ENGINE: chk_access exit: type=NFS4 rc=13 ops=1024
priv=0 against=0
```

   La troisième ligne indique que l’accès est refusé pour le **masque d’accès = 128** (**0x80**) qui est en lecture uniquement (**READ_ATTRIBUTES**, voir le fichier **/usr/include/sys/acl.h**).

Chapitre 4. Audit

Le sous–système d'audit permet à l'administrateur système d'enregistrer des informations de sécurité, qui peuvent être analysées pour détecter des violations potentielles ou effectives de la politique de sécurité.

Cette section traite des points suivants :

Sous–système d'audit

Les fonctions du sous–système d'audit sont les suivantes :

L'administrateur système peut utiliser l'une de ces fonctions pour la configuration.

Détection des événements

La détection des événements est répartie au sein de la base TCB (Trusted Computing Base), dans le noyau (code d'état de superviseur) et dans les programmes sécurisés (code d'état d'utilisateur). Un événement auditable correspond à toute occurrence relative à la sécurité dans le système. Une occurrence relative à la sécurité correspond à toute modification de l'état de sécurité du système, toute tentative de violation ou violation réelle du contrôle d'accès du système ou des politiques de sécurité de gestion de compte, ou des deux. Les programmes et les modules de noyau qui détectent des événements auditables sont responsables de leur enregistrement dans le journal d'audit du système, qui s'exécute dans le noyau et est accessible via une sous–routine (pour l'audit des programmes sécurisés) ou un appel de procédure de noyau (pour l'audit d'état du superviseur). Les informations recueillies comprennent le nom de l'événement auditable, le succès ou l'échec de l'événement, et toute autre information spécifique à l'événement et relative à l'audit de sécurité.

La configuration de la détection des événements consiste à activer et désactiver la détection, et à indiquer les événements à auditer et les utilisateurs concernés. Pour activer la détection, utilisez la commande **audit**, qui active ou désactive le sous–système d'audit. Le fichier **/etc/security/audit/config** contient les événements et utilisateurs traités par le sous–système.

Collecte d'informations sur les événements

Le recueil d'informations comprend la journalisation des événements auditables sélectionnés. Cette fonction est exécutée par le journal d'audit du noyau, qui fournit un appel système et une interface d'appel de procédure interne au noyau qui enregistre les événements auditables.

Le journal d'audit est responsable de l'élaboration de l'enregistrement d'audit complet, composé de l'en–tête, qui contient des informations communes à tous les événements (nom de l'événement, utilisateur responsable, heure et état de retour), et du suivi, qui contient les informations spécifiques à l'événement. Le journal d'audit ajoute chaque enregistrement au suivi d'audit du noyau, qui peut être écrit dans les modes suivants :

Mode BIN Le suivi est écrit sous forme de fichiers alternés, dans un but de sécurité et de stockage à long terme.

Mode STREAM Le suivi est écrit dans un tampon circulaire lu de façon synchrone par une pseudo–unité d'audit. Le mode STREAM assure une réponse immédiate.

La collecte des informations peut être configurée du côté de l'enregistrement des événements comme pour le traitement du suivi. L'enregistrement des événements peut être sélectionné par utilisateur. A chaque utilisateur correspond un ensemble d'événements d'audit, consignés dans le suivi lorsqu'ils se produisent. Côté traitement, les modes sont configurables individuellement, afin que l'administrateur puisse choisir le traitement le mieux adapté à l'environnement. De plus, l'audit en mode BIN peut être configuré pour générer une alerte au cas où l'espace du système de fichiers disponible pour le suivi devienne insuffisant.

Traitement des informations sur le suivi d'audit

Le système d'exploitation fournit plusieurs options de traitement du suivi d'audit du noyau. Le suivi en mode BIN peut être compressé, filtré, et/ou formaté avant d'être archivé, le cas échéant. La compression se fait par codage Huffman. Le filtrage se fait par une sélection des enregistrements d'audit de type SQL, à l'aide de la commande **auditselect**, ce qui permet un affichage et une rétention sélectifs du suivi d'audit. Le formatage des enregistrements du suivi d'audit permet d'examiner le suivi, de générer des rapports de sécurité réguliers, et d'imprimer un document de suivi d'audit.

Le suivi d'audit en mode STREAM peut être contrôlé en temps réel, pour détecter immédiatement les menaces. La configuration de ces options se fait à l'aide de programmes distincts pouvant être appelés en tant que processus démons pour filtrer les suivis en mode BIN ou STREAM, bien que certains des programmes de filtrage soient mieux adaptés à un mode donné.

Sélection des événements

L'ensemble des événements auditables du système définit les occurrences pouvant faire l'objet d'un audit, ainsi que la finesse de l'audit fourni. Les événements auditables doivent regrouper les événements de sécurité du système, tels que définis précédemment.
Le niveau de détail utilisé dans la définition des événements auditables doit assurer l'équilibre entre un manque de détail, qui complique pour l'administrateur la compréhension des informations sélectionnées, et un excès de détails, qui entraîne le recueil de nombreuses informations inutiles. La définition des événements exploite les similitudes entre événements détectés. Un *événement détecté* correspond à une instance d'événement auditable. Par exemple, un événement donné peut être détecté à plusieurs emplacements. Le principe est que les événements détectés ayant les mêmes propriétés de sécurité sont sélectionnés comme un même événement auditable. La liste suivante répertorie les événements de politique de sécurité :

- Evénements sujet
  - Création de processus
  - Suppression de processus
  - Définition des attributs de sécurité des sujets : ID utilisateur, ID de groupe
  - Groupe de processus, terminal de contrôle
- Evénements objet
  - Création d'objets
  - Suppression d'objets
  - Ouverture d'objet (y compris les processus comme objets)
  - Fermeture d'objet (y compris les processus comme objets)
  - Définition des attributs de sécurité des objets : propriétaire, groupe, ACL
- Evénements import/export
  - Import ou export d'un objet
- Evénements de gestion de comptes
  - Ajout d'un utilisateur, modification des attributs des utilisateurs dans la base de mots de passe
  - Ajout d'un groupe, modification des attributs de groupes dans la base de groupes
  - Connexion utilisateur
  - Déconnexion utilisateur
  - Modification des informations d'authentification de l'utilisateur
  - Configuration du terminal de chemins d'accès sécurisés
  - Configuration de l'authentification
  - Gestion des audits : sélection des événements de suivi d'audit, activation, désactivation, définition des classes d'audit des utilisateurs
- Evénements généraux de gestion système
  - Utilisation de privilèges
  - Configuration du système de fichiers
  - Définition et configuration des unités
  - Définition des paramètres de configuration du système

  - IPL (chargement initial) et fermeture corrects du système
  - Configuration RAS
  - Autres configurations du système
- Violations (potentielles) de sécurité
  - Refus de droits d'accès
  - Echecs de privilèges
  - Pannes et erreurs système détectées par diagnostic
  - Tentatives de modification de la TCB

Configuration du sous–système d'audit

Le sous–système d'audit possède une variable d'état global qui indique s'il est activé. De plus, chaque processus possède une variable d'état local qui indique si le sous–système d'audit doit enregistrer des informations sur ce processus. Ces deux variables déterminent si des événements sont détectés par les modules et programmes de la base TCB (Trusted Computing Base). La désactivation de l'audit de la base TCB pour un processus donné permet à ce processus de réaliser son propre audit en respectant la politique de gestion de comptes du système. La fait de permettre à un programme sécurisé de s'auto–auditer assure un recueil d'informations plus efficace.

Collecte d'informations sur le sous–système d'audit

Le recueil d'informations concerne les modes **sélection des événements** et **suivi d'audit du noyau**. Il est effectué par une routine du noyau qui fournit les interfaces nécessaires à la journalisation des informations, utilisées par les composants de la base TCB qui détectent les événements auditables, et les interfaces de configuration, utilisées par le sous–système d'audit pour contrôler la routine de journalisation des audits.

Journalisation des audits

Les événements auditables sont journalisés par les interfaces suivantes : l'état utilisateur et l'état superviseur. La partie état utilisateur de la TCB utilise les sous–routines **auditlog** ou **auditwrite**, tandis que la partie état superviseur de la TCB utilise un ensemble d'appels de procédures du noyau.

Pour chaque enregistrement, le journal des événements d'audit place un en–tête d'audit devant les informations spécifiques à l'événement. Cet en–tête indique l'utilisateur et le processus pour lesquels l'événement est audité, ainsi que l'heure de l'événement. Le code qui détecte l'événement indique le type d'événement et le code de retour ou l'état, et peut fournir des informations spécifiques à l'événement (le suivi). Les informations spécifiques à l'événement comprennent les noms d'objets (par exemple, les fichiers dont l'accès est refusé ou les tty utilisés lors des échecs de connexion), les paramètres de sous–routines, et les autres informations modifiées.

Les événements sont définis par symboles plutôt que par numéros. Ceci réduit les risques de noms identiques, sans utiliser de méthode d'enregistrement des événements. Comme les sous–routines sont auditables et que la définition du noyau extensible n'a pas de numéro SVC (switched virtual circuit) fixe, il est difficile d'enregistrer des événements par numéro. Il faudrait pour cela corriger ces numéros à chaque extension ou redéfinition de l'interface du noyau.

Format des enregistrements d'audits

Les enregistrements d'audit comprennent un en–tête commun, qui précède les suivis d'audit spécifiques à l'événement de l'enregistrement. Les structures des en–têtes sont définies dans le fichier **/usr/include/sys/audit.h**. Le format des informations dans le suivi d'audit est spécifique à chaque événement, et indiqué dans le fichier **/etc/security/audit/events**.

Les informations de l'en–tête sont généralement recueillies par la routine de journalisation, pour garantir qu'elles correspondent aux informations des suivis d'audit fournies par le code qui détecte l'événement. Le journal d'audit ne connaît absolument pas la structure ni la sémantique des suivis d'audit. Par exemple, lorsque la commande **login** détecte un échec de connexion, elle enregistre cet événement avec le terminal sur lequel il s'est produit, et l'écrit dans le suivi d'audit à l'aide de la sous–routine **auditlog**. Le composant noyau du journal d'audit enregistre des informations spécifiques (ID utilisateur, ID de processus, heure) dans un en–tête qu'il ajoute aux autres informations. L'appelant renseigne uniquement les champs nom de l'événement et résultat de l'en–tête.

Configuration de la journalisation d'audit

Le journal d'audit est responsable de l'élaboration de l'enregistrement complet de l'audit. Vous devez sélectionner les événements d'audit à journaliser.

Sélection des événements audités

La sélection des événements audités comprend les types suivants :

Audit par processus

Pour définir efficacement les événements de processus, l'administrateur système peut définir des classes d'audit. Une classe d'audit est un sous–ensemble des événements d'audit du système. Ces classes d'audit regroupent de manière logique et pratique des événements d'audit de base.

Pour chaque utilisateur du système, l'administrateur définit un ensemble de classes d'audit qui déterminent les événements de base à enregistrer. Chaque processus exécuté par un utilisateur est référencé par ses classes d'audit.

Audit par objet

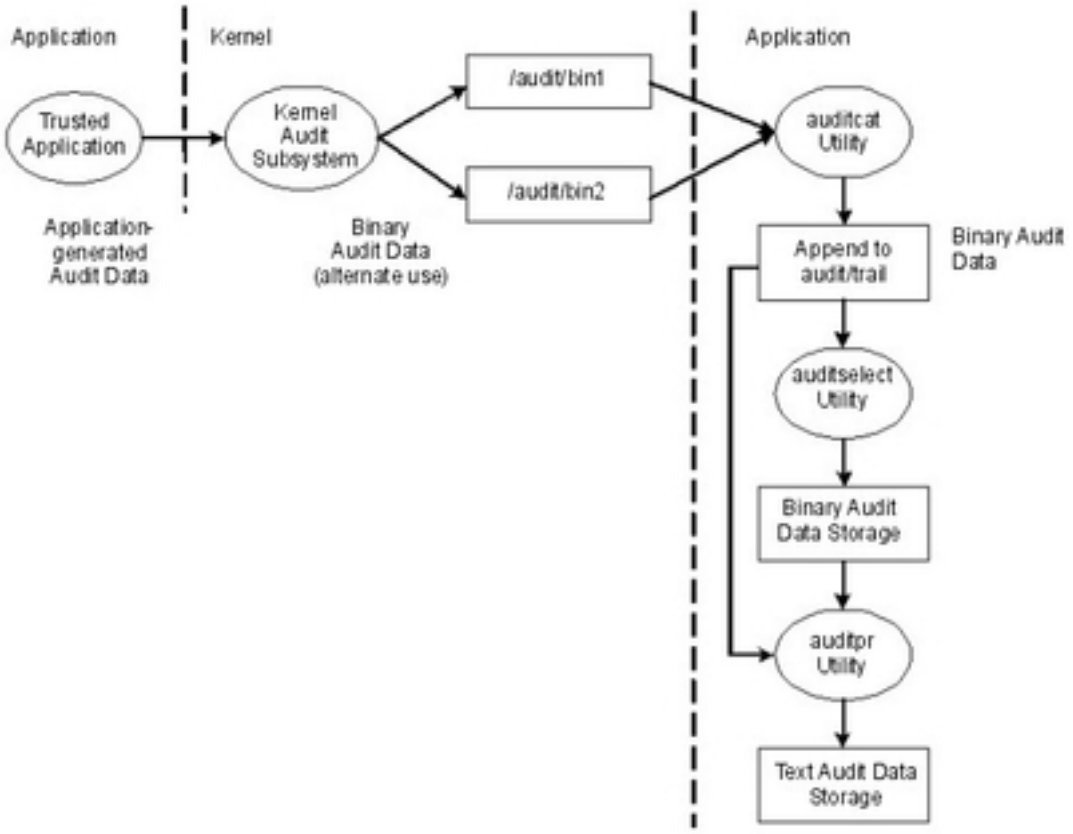
Le système d'exploitation assure l'audit des accès aux objets par nom, c'est à dire l'audit d'objets spécifiques (normalement des fichiers). L'audit d'objets par nom évite de devoir auditer tous les accès aux objets pour couvrir ceux qui sont pertinents. De plus, le mode d'audit peut être spécifié, afin que seuls les accès du mode indiqué (read/write/execute) soient enregistrés.

Modes de suivi d'audit du noyau

La journalisation du noyau peut utiliser les modes BIN ou STREAM pour définir l'emplacement d'écriture du suivi d'audit. En mode BIN, le journal d'audit du noyau doit disposer (avant le début de l'audit) d'au moins un descripteur de ficher où les enregistrements seront placés.

Le mode BIN écrit les enregistrements dans des fichiers alternés. Au début de l'audit, le noyau reçoit deux descripteurs de fichiers et une taille bin maximale recommandée. Il suspend le processus d'appel et lance l'écriture des enregistrements dans le premier descripteur de fichier. Lorsque la taille du premier fichier atteint son maximum, et si le deuxième descripteur de fichier est valide, il passe sur le deuxième fichier et réactive le processus d'appel. Le noyau poursuit l'écriture dans le deuxième fichier jusqu'à ce qu'il reçoive un nouvel appel avec un descripteur de fichier valide. Si à ce moment le deuxième fichier est plein, il retourne sur le premier, et le processus d'appel repart immédiatement. Sinon, le processus d'appel est suspendu, et le noyau poursuit l'écriture d'enregistrements dans le deuxième fichier jusqu'à ce qu'il soit plein. Le traitement se poursuit de la même manière jusqu'à la désactivation de l'audit. Reportez–vous à la figure suivante illustrant le mode BIN :

Figure 1. Fonctionnement du mode d'audit BIN. Cette illustration présente le fonctionnement du mode d'audit BIN.

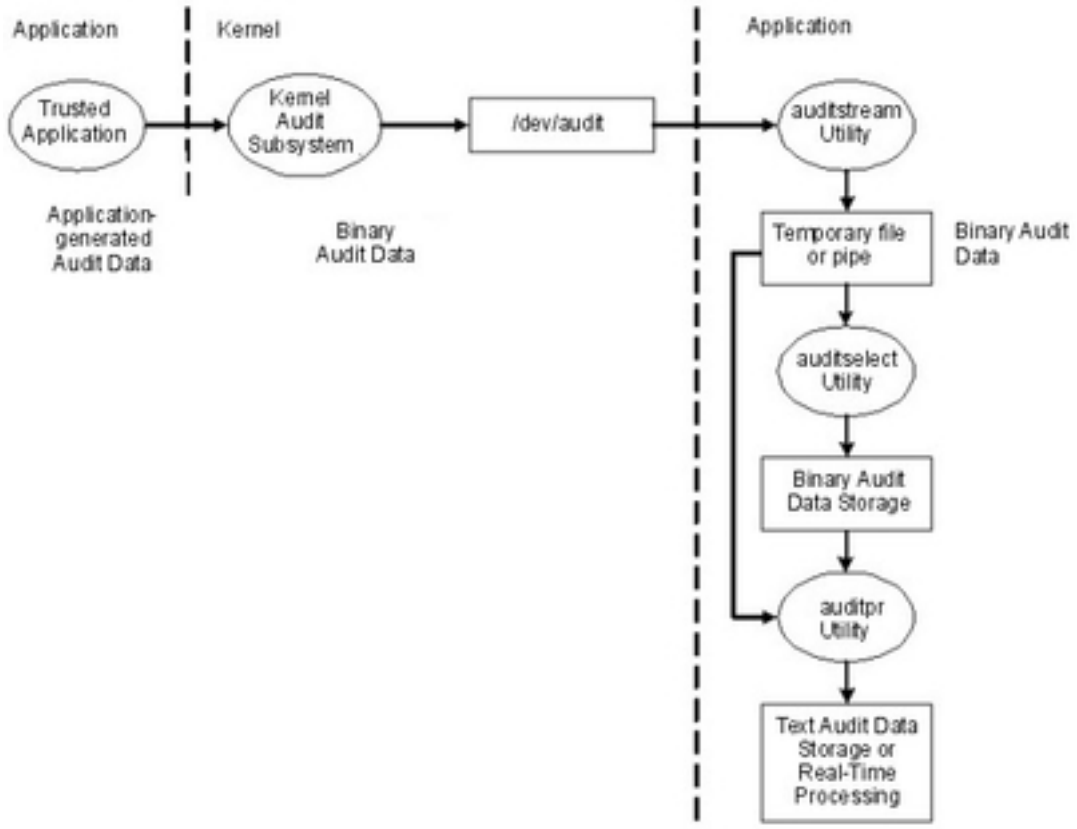


Le mécanisme de casiers (ou fichiers) alternés permet de garantir que le sous–système d'audit dispose toujours d'un emplacement d'écriture lors du traitement des enregistrements d'audit. Lorsque le sous–système d'audit change de casier, il vide le précédent dans le fichier **trace**. Lorsqu'il faut de nouveau changer de casier, le premier est disponible. Ce mode dissocie le stockage et l'analyse des données de leur génération. Généralement, le programme **auditcat** sert à lire les données du casier dans lequel le noyau n'est pas en train d'écrire. Pour s'assurer que le système dispose toujours d'espace disponible pour le suivi d'audit (le résultat du programme **auditcat**), le paramètre **freespace** peut être indiqué dans le fichier **/etc/security/audit/config**. Si le système dispose de moins d'espace que le nombre de blocs de 512 octets spécifiés, il génère un message **syslog**.

Si l'audit est activé, la paramètre **binmode** de la strophe **start** dans **/etc/security/audit/config** doit avoir pour valeur **panic**. Le paramètre **freespace** de la strophe **bin** doit avoir une valeur au minimum égale à 25 % de l'espace disque dédié au stockage des suivis d'audit. Les paramètres **bytethreshold** et **binsize** doivent tous deux être définis sur 65536 octets.

En mode STREAM, le noyau écrit les enregistrements dans un tampon circulaire. Une fois que le noyau atteint la fin du tampon, il continue simplement au début. Les processus lisent les informations via une pseudo–unité nommée **/dev/audit**. Lorsqu'un processus ouvre cette unité, un nouveau canal est créé pour ce processus. Eventuellement, les événements à lire sur le canal peuvent être indiqués comme liste des classes d'audit. Reportez–vous à la figure suivante illustrant le mode STREAM :

Figure 2. Fonctionnement du mode d'audit STREAM. Cette illustration présente le fonctionnement du mode d'audit STREAM.



Le but principal du mode STREAM est de permettre la lecture permanente du suivi d'audit, très utile pour contrôler les menaces en temps réel. Il sert également à créer un suivi écrit immédiatement, évitant toute altération qui pourrait se produire en cas de stockage sur un support inscriptible.

Une autre méthode d'utilisation du mode STREAM est d'écrire les données d'audit dans un programme qui stocke les informations d'audit sur un système distant, permettant un traitement central en temps quasi–réel, tout en protégeant les informations d'audit contre une altération sur l'hôte qui les a émises.

Traitement des enregistrements d'audit

Les commandes **auditselect**, **auditpr** et **auditmerge** permettent de traiter les enregistrements d'audit en mode BIN ou STREAM. Elles fonctionnent comme des filtres, et peuvent donc être utilisés avec des pipes, ce qui est particulièrement pratique pour les audits en mode STREAM.

auditselect — Permet de sélectionner des enregistrements d'audits spécifiques avec des instructions de type SQL. Par exemple, pour ne sélectionner que des événements **exec()** générés par l'utilisateur **afx**, saisissez la commande suivante :

```
auditselect -e "login==afx && event==PROC_Execute"
```

auditpr — Permet de convertir les enregistrements d'audit binaires en un format plus lisible. La quantité d'informations affichée dépend des indicateurs spécifiés sur la ligne de commandes. Pour obtenir toutes les informations disponibles, utilisez **auditpr** de la façon suivante :

```
auditpr -v -hhelrtRpPTc
```

Lorsque l'indicateur **–v** est indiqué, le suivi d'audit, qui est une chaîne spécifique à l'événement (voir le fichier **/etc/security/audit/events**), s'affiche en sus des informations d'audit standard fournies par le noyau pour chaque événement.

auditmerge — Utilisé pour fusionner des suivis d'audit binaires. Cela est particulièrement utile pour combiner des suivis d'audit de plusieurs systèmes. La commande **auditmerge** prend les noms des suivis de la ligne de commande et envoie le suivi binaire fusionné à la sortie standard. Il faut donc encore utiliser la commande **auditpr** pour le rendre lisible. Par exemple, les commandes **auditmerge** et **auditptr** peuvent être utilisées de la façon suivante :

```
auditmerge trail.system1 trail.system2 | auditpr -v -hhelrRtpc
```

Utilisation du sous–système d'audit pour un rapide contrôle de sécurité

La commande **watch** permet de contrôler un programme suspect sans configurer le sous–système d'audit. Elle enregistre l'événement requis ou tous les événements générés par ce programme. Par exemple, utilisez la commande suivante pour voir tous les événements **FILE_Open** lors de l'exécution de **vi /etc/hosts** :

```
watch -eFILE_Open -o /tmp/vi.watch vi /etc/hosts
```

Le fichier **/tmp/vi.watch** affiche tous les événements **FILE_Open** pour la session d'édition.

Configuration de l'audit

La procédure suivante montre comment établir un sous–système d'audit.
Pour plus d'informations, reportez–vous aux fichiers de configuration mentionnés pour ces étapes.

1. Sélectionnez des activités système (événements) à auditer depuis la liste du fichier **/etc/security/audit/events**. Si vous avez ajouté des événements d'audit aux applications ou extensions du noyau, vous devez ajouter les nouveaux événements au fichier.
   - Vous pouvez ajouter un événement dans ce fichier si votre programme comporte le code d'enregistrement associé (au moyen des sous–routines **auditwrite** ou **auditlog**), ou si ce code est dans une extension de noyau (par le biais des services noyau **audit_svcstart**, **audit_svcbcopy** et **audit_svcfinis**).
   - Assurez–vous que les instructions de formatage des nouveaux événements d'audit figurent dans le fichier **/etc/security/audit/events**. Ces indications permettent à la commande **auditpr** d'écrire un suivi d'audit au moment du formatage des enregistrements d'audit.
2. Regroupez les événements d'audit sélectionnés en ensembles d'éléments similaires appelés *classes d'audit*. Définissez ces classes d'audit dans la strophe des classes du fichier **/etc/security/audit/config**.
3. Affectez les classes d'audit aux utilisateurs et les événements d'audit aux fichiers (objets) à auditer, comme suit :
   - Pour attribuer des classes d'audit à un utilisateur donné, ajoutez une ligne dans la strophe des utilisateurs dans **/etc/security/audit/config**. Pour affecter des classes d'audit à un utilisateur, vous pouvez utiliser la commande **chuser**.
   - Pour affecter des événements d'audit à un objet (données ou fichier exécutable), ajoutez une strophe à ce fichier dans **/etc/security/audit/objects**.
   - Vous pouvez aussi définir des classes d'audit par défaut pour de nouveaux utilisateurs en modifiant **/usr/lib/security/mkuser.default**. Ce fichier contient les attributs utilisateur servant à générer les nouveaux ID utilisateur. Par exemple, utilisez la classe d'audit **general** pour tous les nouveaux ID utilisateur, comme suit :

     ```
     user:
           auditclasses = general
          pgrp = staff
          groups = staff
          shell = /usr/bin/ksh
          home = /home/$USER
     ```

     Pour obtenir tous les événements d'audit, spécifiez la classe **ALL**. Cela générera une immense quantité de données même sur un système dont l'activité est modérée. Il est généralement plus pratique de limiter le nombre d'événements enregistrés.

4. Dans le fichier **/etc/security/audit/config**, configurez le type de collecte de données souhaité : BIN et/ou STREAM. Assurez–vous que les données d'audit n'entrent pas en concurrence avec d'autres données en termes d'espace de fichiers, en utilisant pour elles un système de fichiers séparé. Vous serez ainsi assuré de disposer de suffisamment d'espace pour les données d'audit. Configurez comme suit le type de collecte de données :
   - Pour configurer la méthode BIN :
     a. Définissez `binmode = on` dans la strophe **start**.
     b. Modifiez la strophe **binmode** pour configurer les casiers et le suivi, et indiquez le chemin d'accès au fichier qui contient les commandes de traitement binmode. Le fichier par défaut des commandes de traitement est **/etc/security/audit/bincmds**.
     c. Vérifiez que les casiers d'audit sont suffisamment grands et définissez **freespace** en conséquence pour obtenir une alerte si le système de fichiers se remplit.
     d. Ajoutez les commandes shell de traitement des casiers d'audit dans un tube d'audit du fichier **/etc/security/audit/bincmds**.
   - Pour utiliser la méthode STREAM
     a. Définissez `streammode = on` dans la strophe **start**.
     b. Modifiez la strophe streammode pour indiquer le chemin d'accès au fichier qui contient les commandes de traitement streammode. Le fichier par défaut qui contient ces informations est **/etc/security/audit/streamcmds**.
     c. Ajoutez les commandes shell de traitement des enregistrements continus dans un tube d'audit du fichier **/etc/security/audit/streamcmds**.
5. Une fois les modifications nécessaires apportées aux fichiers de configuration, activez le sous–système d'audit au moyen de la commande **audit start**.
6. Utilisez la commande **audit query** pour afficher les événements et objets audités.
7. Utilisez la commande **audit shutdown** pour désactiver à nouveau le sous–système d'audit.

Sélection des événements audités

L'objectif d'un audit est de détecter des activités risquant de compromettre la sécurité du système. Les opérations suivantes, effectuées par un utilisateur non autorisé, constituent une violation de la sécurité du système et sont auditables :

- Valider des opérations dans la base TCB
- Authentifier des utilisateurs
- Accéder au système
- Modifier la configuration du système
- Circonvenir le système d'audit
- Initialiser le système
- Installer des programmes
- Modifier des comptes
- Transférer des informations

Le système d'audit ne dispose pas d'ensemble par défaut des événements à auditer. Vous devez sélectionner des événements ou classes d'événements en fonction de vos besoins.

Pour auditer une activité, il est indispensable d'identifier la commande ou le processus à l'origine de l'événement et de s'assurer que cet événement est répertorié dans **/etc/security/audit/events**. Vous devez ensuite inclure l'événement dans la classe appropriée du fichier **/etc/security/audit/config** ou dans la strophe d'objets de **/etc/security/audit/objects**. Reportez–vous au fichier **/etc/security/audit/events** de votre système pour la liste des événements d'audit et les instructions de formatage du suivi. Reportez–vous à la commande **auditpr** pour une description des formats d'événements d'audit (écriture et exploitation).

Une fois les événements à auditer sélectionnés, vous devez regrouper les événements similaires en classes d'audit. Ensuite, les classes d'audit sont affectées aux utilisateurs.

Sélection des classes d'audit

Vous pouvez simplifier l'affectation des événements d'audit aux utilisateurs en regroupant les événements identiques en classes d'audit. Ces classes sont définies dans la strophe des classes du fichier **/etc/security/audit/config**.

Voici quelques classes courantes :

general	Evénements d'ordre général modifiant l'état du système et l'authentification des utilisateurs. Audit des tentatives de circonvenir les contrôles d'accès au système.
objects	Accès en écriture aux fichiers de configuration de la sécurité.
kernel	Les événements de la classe noyau sont générés par les fonctions de gestion des processus du noyau.

Voici un exemple de strophe du fichier **/etc/security/audit/config** :

```
classes:
     general = USER_SU,PASSWORD_Change,FILE_Unlink,FILE_Link,FILE_Rename
     system = USER_Change,GROUP_Change,USER_Create,GROUP_Create
     init = USER_Login,USER_Logout
```

Sélection du mode de collecte des données d'audit

La méthode à retenir dépend de la façon dont vous exploiterez les données d'audit. Si vous souhaitez stocker sur le long terme un volume important des données collectées, le mode BIN est préconisé. Si vous traitez les données au fur et à mesure de leur collecte, utilisez le mode STREAM. Vous pouvez aussi sélectionner les deux méthodes.

Collecte Bin	Stockage à long terme d'un important suivi d'audit. Les enregistrements d'audit sont écrits dans un fichier servant de casier temporaire. Quand ce fichier est saturé, le démon **auditbin** traite les données tandis que le sous–système d'audit écrit vers un autre fichier casier, puis les enregistrements sont stockés dans un fichier de suivi d'audit.
Collecte Stream	Permet le traitement des données d'audit en même temps qu'elles sont recueillies. Les enregistrements d'audit sont inscrits dans un tampon circulaire du noyau et récupérés en lisant **/dev/audit**. Ces enregistrements peuvent être affichés, imprimés et/ou convertis pour être compatibles avec le mode bin (avec la commande **auditcat**).

Exemple de contrôle en temps réel des modifications de fichiers

L'exemple suivant permet de contrôler en temps réel l'accès à des fichiers critiques :

1. Dressez la liste des fichiers dont les modifications sont à contrôler, par exemple tous les fichiers dans **/etc** et configurez–les pour les événements **FILE_Write** dans le fichier **objects** :

```
find /etc –type f | awk '{printf("%s:\n\tw = FILE_Write\n\n",$1)}' >>
/etc/security/audit/objects
```

2. Choisissez le mode Stream pour établir la liste de toutes les écritures sur fichiers. Cet exemple répertorie toutes les écritures sur fichiers sur la console. En environnement de production, vous souhaiterez peut–être disposer d'un back–end qui envoie les événements vers un système de détection des intrusions. Le fichier **/etc/security/audit/streamcmds** est similaire à l'exemple suivant :

```
/usr/sbin/auditstream | /usr/sbin/auditselect –e "event == FILE_Write" |
 auditpr  –hhelpPRtTc –v > /dev/console &
```

3. Choisissez le mode STREAM dans **/etc/security/audit/config**, ajoutez une classe pour les événements d'écriture sur fichiers et configurez tous les utilisateurs à auditer avec cette classe :

```
start:
         binmode = off
         streammode = on

 stream:
         cmds = /etc/security/audit/streamcmds

 classes:
         filemon = FILE_write

 users:
         root = filemon
         afx = filemon
         ...
```

4. Exécutez maintenant **audit start**. Tous les événements **FILE_Write** apparaissent sur la console.

Exemple de scénario de journal d'audit générique

Dans cet exemple, on considère qu'un administrateur système veut utiliser le sous–système d'audit pour contrôler un vaste système multi–utilisateur. Aucune intégration directe vers un IDS n'est effectuée. Tous les enregistrements d'audits seront contrôlés manuellement. Seuls quelques événements d'audit essentiels sont enregistrés, afin que la quantité de données générée reste possible à traiter.

Les événements d'audit pris en compte pour la sélection d'événements sont les suivants :

FILE_Write	Pour analyser toutes les écritures sur les fichiers de configuration. Cet événement sera utilisé avec tous les fichiers de l'arborescence **/etc**.
PROC_SetUserIDs	Toutes les modifications d'ID utilisateur
AUD_Bin_Def	Configuration des casiers d'audit
USER_SU	Commande **su**
PASSWORD_Change	Commande **passwd**
AUD_Lost_Rec	Notification en cas de perte d'enregistrements
CRON_JobAdd	Nouvelle tâche **cron**
AT_JobAdd	Nouvelle tâche **at**
USER_Login	Toutes les connexions
PORT_Locked	Tous les verrouillages de terminaux pour cause de nombreux échecs

L'exemple suivant montre comment générer un journal d'audit générique :

1. Dressez la liste des fichiers critiques dont les modifications sont à contrôler, par exemple tous les fichiers dans **/etc** et configurez–les pour les événements **FILE_Write** dans le fichier **objects** :

```
find /etc –type f | awk '{printf("%s:\n\tw = FILE_Write\n\n",$1)}' >>
/etc/security/audit/objects
```

2. Utilisez la commande **auditcat** pour configurer l'audit en mode BIN. Le fichier **/etc/security/audit/bincmds** est similaire à l'exemple suivant :

```
/usr/sbin/auditcat –p –o $trail $bin
```

3. Modifiez le fichier **/etc/security/audit/config** et ajoutez une classe pour les événements intéressants. Dressez la liste des utilisateurs et affectez–leur la classe **custom**.

```
start:
         binmode = on
         streammode = off

bin:
         cmds = /etc/security/audit/bincmds
         trail = /audit/trail
         bin1 = /audit/bin1
         bin2 = /audit/bin2
         binsize = 100000
         freespace = 100000

 classes:
         custom = FILE_Write,PROC_SetUser,AUD_Bin_Def,AUD_Lost_Rec,USER_SU,
\

PASSWORD_Change,CRON_JobAdd,AT_JobAdd,USER_Login,PORT_Locked

 users:
         root = custom
         afx = custom
         ...
```

4. Ajoutez la classe d'audit **custom** au fichier **/usr/lib/security/mkuser.default**, afin que les nouveaux ID aient automatiquement le bon appel d'audit :

```
user:
      auditclasses = custom
     pgrp = staff
     groups = staff
     shell = /usr/bin/ksh
     home = /home/$USER
```

5. Créez un nouveau système de fichiers nommé **/audit** à l'aide de SMIT ou de la commande **crfs**. Il doit être assez grand pour contenir les deux casiers et un grand suivi d'audit.

6. Exécutez l'option de commande **audit start** et examinez le fichier **/audit**. Vous devez d'abord voir apparaître deux fichiers casiers et un fichier de suivi **trail** vide. Après utilisation du système, le fichier **trail** doit contenir des enregistrements d'audit, lisibles à l'aide de la commande

```
auditpr  -hhelpPRtTc -v | more
```

Cet exemple n'utilise que peu d'événements. Pour tous les utiliser, spécifiez le nom de classe **ALL** pour tous les utilisateurs. Vous obtiendrez de grandes quantités de données. Vous souhaiterez peut–être ajouter tous les événements liés aux modifications d'utilisateurs et de droits à votre classe **custom**.

Chapitre 5. Protocole LDAP – Généralités

Le protocole LDAP (Light Directory Access Protocol) définit une méthode standard pour accéder à des informations dans un répertoire (une base de données) et de les mettre à jour, localement ou à distance, dans un modèle client–serveur. Le protocole est optimisé pour lire, parcourir et rechercher des répertoires. A l'origine, il a été conçu comme frontal léger pour le protocole d'accès au répertoire X.500. La méthode LDAP est utilisée par un cluster d'hôtes pour permettre une authentification centralisée de la sécurité ainsi que l'accès à des informations sur les utilisateurs et les groupes. Dans un environnement de cluster, cette fonctionnalité permet d'utiliser les mêmes informations d'authentification, d'utilisateur et de groupe dans tout l'environnement.

Les objets dans LDAP sont stockés dans une structure hiérarchique connue sous le nom d'arborescence d'information de répertoire (DIT). Un répertoire conforme débute avec la structure de l'arborescence d'information de répertoire (DIT). L'arborescence d'information de répertoire doit être conçue avant l'implémentation du protocole LDAP comme moyen d'authentification.

Module de chargement d'authentification LDAP

L'exploitation LDAP du sous–système de sécurité est implémentée en tant que module de chargement d'authentification LDAP. Sa conception est similaire aux autres modules de chargement tels que NIS, DCE et KRB5. Les modules de chargement sont définis dans le fichier **/usr/lib/security/methods.cfg**. Le module de chargement LDAP fournit une authentification utilisateur et une fonction centralisée de gestion utilisateur et groupe via le protocole LDAP. Un utilisateur défini sur un serveur LDAP peut être configuré pour se connecter à un client LDAP, même si cet utilisateur n'est pas défini localement.

Le module de chargement LDAP AIX est complètement intégré dans le système d'exploitation AIX. Une fois que ce module de chargement d'authentification LDAP est activé pour donner des informations sur les utilisateurs et sur les groupes, les API de haut niveau, les commandes et les outils de gestion de systèmes fonctionnent de manière habituelle. L'indicateur **–R** sert pour que la plupart des commandes de haut niveau puissent utiliser différents modules de chargement. Par exemple, la commande suivante créera depuis un poste client un utilisateur LDAP nommé `joe` :

```
mkuser –R LDAP joe
```

Remarque : Dans une infrastructure LDAP, le nombre d'utilisateurs par groupe est en principe illimité. Un test a été effectué en créant 25 000 utilisateurs dans un groupe et en effectuant diverses opérations sur ce groupe. Notez qu'avec certaines interfaces POSIX traditionnelles, il se peut que les données renvoyées pour un tel groupe soient incomplètes. Pour plus d'informations sur ce type de limitation, veuillez consulter la documentation de chaque interface API.

Authentification basée sur LDAP

Cette section décrit les limites des différentes entités de l'authentification basée sur LDAP sous AIX. Notez que l'infrastructure LDAP elle–même n'impose aucune limite sur le contenu de la base de données. Cependant, cette section fournit les résultats en termes de limites obtenus avec des configurations de test. Voici les limites observées lors d'une authentification basée sur LDAP en environnement AIX :

Nombre total d'utilisateurs : %%1$d Création de 500 000 utilisateurs sur un système et test de centaines d'authentifications d'utilisateur simultanées.

Nombre total de groupes : Création et test de 500 groupes sur un système.

Nombre maximal d'utilisateurs par groupe : Création de 25 000 utilisateurs dans un groupe et test de diverses opérations sur ce groupe.

Notez qu'avec certaines interfaces POSIX traditionnelles, il se peut que les données renvoyées pour un tel groupe soient incomplètes. Pour plus d'informations sur ce type de limitation, veuillez consulter la documentation de chaque interface API. Notez également que les valeurs ci–dessus proviennent des test effectués. Cela n'exclut pas la possibilité de configurer des systèmes avec davantage d'utilisateurs et de groupes, si les ressources le permettent.

Configuration d'un serveur d'informations de sécurité LDAP

Pour configurer un système comme serveur d'informations de sécurité LDAP pour l'authentification, il est nécessaire d'installer au préalable les utilisateurs et les groupes, ainsi que les modules LDAP serveur et client. Si Secure Socket Layer (SSL) est nécessaire, le **GSKit** doit être installé. L'administrateur du système doit créer une clé à l'aide de la commande **ikeyman**. Pour plus d'informations sur la configuration du serveur afin d'utiliser SSL, reportez–vous à la section Communication sécurisée avec SSL.

Pour simplifier la configuration du serveur, AIX a créé la commande **mksecldap**. La commande **mksecldap** peut être utilisée pour configurer un serveur d'informations de sécurité LDAP. Elle configure une base de données appelée **ldapdb2**, y écrit les informations sur les utilisateurs et sur les groupes obtenues de l'hôte local, et définit le nom spécifique et le mot de passe de l'administrateur du serveur LDAP. Elle peut également configurer SSL pour la communication client/serveur. La commande **mksecldap** ajoute également une entrée au fichier **/etc/inittab** pour lancer le serveur LDAP à chaque redémarrage. L'ensemble de la configuration du serveur LDAP est effectué via la commande **mksecldap**, qui met à jour le fichier **ibmslapd.conf** (Tivoli Directory Server Version 5.1 et versions ultérieures), le fichier **slapd.conf** (SecureWay® Directory Versions 3.2 et 4.1) ou le fichier **slapd32.conf** (SecureWay Directory Version 3.2).

A moins que l'option de commande **–u NONE** de la commande **mksecldap** ne soit utilisée, tous les utilisateurs et groupes du système local sont exportés vers le serveur LDAP pendant la configuration. Pour cette étape, choisissez l'un des schémas LDAP suivants :

Schéma AIX spécifique
: Comprend les classes d'objets **aixAccount** et **aixAccessGroup**. Ce schéma apporte un ensemble complet d'attributs pour les utilisateurs et les groupes AIX.

Schéma NIS (RFC 2307)
: Comprend les classes d'objet **posixAccount**, **shadowAccount** et **posixGroup**. Il est utilisé par les répertoires de plusieurs éditeurs. Ce schéma ne définit qu'une petite partie des attributs utilisés par AIX.

Schéma NIS avec support AIX complet
: Comprend les classes d'objet **posixAccount**, **shadowAccount** et **posixGroup**, avec en plus les classes d'objet **aixAusAccount** et **aixAusGroup**. Les classes d'objet **aixAusAccount** et **aixAuxGroup** fournissent les attributs utilisés par AIX qui ne sont pas définis par le schéma NIS. Il est recommandé de configurer le serveur LDAP à l'aide du schéma NIS avec support AIX complet, sauf si vous devez utiliser un schéma LDAP particulier pour des raisons de compatibilité avec d'autres serveurs LDAP.

Toutes les informations sur les utilisateurs et les groupes sont stockées dans une même arborescence (suffixe) AIX. Le suffixe par défaut est "cn=aixdata". La commande **mksecldap** accepte un suffixe fourni par l'utilisateur avec l'indicateur **–d**. Le nom des sous–arbres à créer pour l'utilisateur, le groupe, l'ID, etc. est commandé par le fichier de configuration **sectoldif.cfg**. Reportez–vous au fichier **sectoldif.cfg** pour plus d'informations.

Cette arborescence AIX est protégée par une liste de contrôle d'accès (ACL). La liste de contrôle d'accès (ACL) par défaut attribue des privilèges administratifs uniquement à l'entité définie comme l'administrateur à l'aide de l'option de commande **–a**. Des privilèges supplémentaires peuvent être accordés à une identité proxy si les options de commande **–x** et **–X** sont utilisées. L'utilisation de ces options permet de créer l'identité proxy et de configurer les droits d'accès comme défini dans le fichier **/etc/security/ldap/proxy.ldif.template**. La création d'une identité proxy permet aux clients LDAP de créer des liens vers le serveur sans avoir à recourir à l'identité de l'administrateur, limitant ainsi les privilèges administrateur client sur le serveur LDAP.

La commande **mksecldap** fonctionne même si un serveur LDAP a été configuré pour d'autres raisons (pour des informations de recherche d'ID utilisateur, par exemple). Dans ce cas, **mksecldap** ajoute l'arborescence AIX et la remplit avec les informations sur la sécurité AIX dans la base de données existante. Cette arborescence est spécifiquement protégée par une liste de contrôle des accès ACL. Le serveur LDAP fonctionne normalement, et sert également de serveur de sécurité LDAP AIX.

Remarque : Il est conseillé de sauvegarder la base de données avant d'exécuter la commande **mdsecldap** et de configurer le serveur de sécurité pour l'utilisation partagée de la base de données.

Une fois le serveur d'informations de sécurité LDAP configuré, il peut également être configuré en tant que client pour permettre la gestion des utilisateurs et des groupes LDAP ainsi que la connexion au serveur des utilisateurs LDAP.

Si la configuration du serveur d'informations de sécurité LDAP échoue, vous pouvez l'annuler par la commande **mksecldap** avec l'indicateur **–U**. Cela permet de restaurer le fichier **ibmslapd.conf** (ou **slapd.conf** ou **slapd32.conf**) tel qu'il était avant la configuration. Vous utiliserez la commande **mksecldap** avec l'indicateur **–U** après une tentative infructueuse de configuration, et avant d'essayer à nouveau la commande **mksecldap**. Sinon, des informations peuvent rester dans le fichier de configuration et faire échouer la configuration suivante. Pour des raisons de sécurité, l'option annulation ne modifie ni la base de données ni ses données, car cette base peut avoir existé avant l'exécution de la commande **mksecldap**. Vous devez supprimer manuellement toute base de données créée par la commande **mksecldap**. Si des données ont été ajoutées à une base de données préexistante par la commande **mksecldap**, vous devez décider des mesures à prendre pour corriger la tentative de configuration.

Pour plus d'informations sur la configuration d'un serveur d'informations de sécurité LDAP, reportez–vous à la description de la commande **mksecldap**.

Configuration d'un client LDAP

Avant de configurer un client pour qu'il utilise LDAP pour l'authentification et les informations d'un utilisateur/groupe, assurez–vous que le module client LDAP est installé sur chaque client. Si SSL est nécessaire, le GSKit doit être installé, une clé doit être créée et le certificat de la clé SSL du serveur LDAP doit être ajouté à cette clé.

De la même manière que pour la configuration d'un serveur LDAP, vous pouvez configurer un client à l'aide de la commande **mksecldap**. Pour que ce client contacte le serveur d'informations de sécurité LDAP, le nom du serveur doit être indiqué pendant la configuration. Le nom distinctif BIND du serveur et le mot de passe sont également nécessaires pour que le client accède à l'arborescence AIX sur le serveur. La commande **mksecldap** enregistre le nom distinctif BIND du serveur, le mot de passe, le nom du serveur, le nom distinctif de l'arborescence AIX sur le serveur, le chemin et le mot de passe de la clé SSL, ainsi que d'autres attributs de configuration, dans le fichier **/etc/security/ldap/ldap.cfg**.

Plusieurs serveurs peuvent être fournis à la commande **mksecldap** pendant la configuration du client. Dans ce cas, le client contacte les serveurs dans l'ordre indiqué et se connecte au premier serveur avec lequel il réussit à établir une connexion. Si une erreur de connexion entre le client et le serveur se produit, une requête de reconnexion est tentée à l'aide de la même logique. Le modèle d'exploitation LDAP de sécurité ne prend pas en charge les renvois. Il est important que les serveurs dupliqués restent synchronisés.

Le client communique avec le serveur d'informations de sécurité LDAP par le biais d'un démon côté client (**secldapclntd**). Si le module de chargement LDAP est activé sur le client, les commandes de haut niveau finissent par trouver ce démon par le biais des API pour les utilisateurs définis dans LDAP. Le démon permet de conserver un cache des entrées LDAP demandées. Si une requête n'est pas satisfaite à partir du cache, le démon interroge le serveur, met à jour le cache et renvoie l'information à l'appelant.

D'autres options d'optimisation peuvent être passées à la commande **mksecldap** pendant la configuration du client, comme le nombre de routines utilisées par le démon, la capacité de l'entrée cache et le délai d'expiration de cache. Ces options sont réservées aux utilisateurs expérimentés. Pour la plupart des environnements, les valeurs par défaut sont suffisantes.

Dans les étapes finales de la configuration du client, la commande **mksecldap** lance le démon côté client et ajoute une entrée dans le fichier **/etc/inittab** pour que le démon soit lancé à chaque redémarrage. Vous pouvez vérifier si la configuration est réussie en consultant le process **secldapclntd** via la commande **ls–secldapclntd**. Du moment que le serveur d'informations de sécurité LDAP est configuré et fonctionne, ce démon s'exécutera si la configuration a réussi.

Gestion des utilisateurs LDAP

La gestion des utilisateurs et des groupes sur un serveur d'informations de sécurité LDAP peut être effectuée depuis tout client LDAP à l'aide des commandes de haut niveau. Un indicateur **–R** ajouté à la plupart des commandes de haut niveau permet de gérer des utilisateurs et des groupes utilisant LDAP de même que d'autres modules de chargement d'authentification tels que DCE, NIS et KRB5. Pour plus d'informations sur l'utilisation de l'indicateur **–R**, reportez–vous à la description de chacune des commandes de gestion utilisateur ou groupe.

Pour permettre à un utilisateur de s'authentifier via LDAP, exécutez la commande **chuser** afin de définir l'attribut **SYSTEM** de l'utilisateur sur LDAP. En définissant l'attribut **SYSTEM** selon la syntaxe définie, un utilisateur peut être authentifié par plusieurs modules de chargement (compat et LDAP par exemple). Pour plus d'informations sur la configuration des méthodes d'authentification des utilisateurs, reportez–vous à la section Authentification de l'utilisateur, page 2-27 et à la syntaxe de l'attribut **SYSTEM** définie dans le fichier **/etc/security/user**.

Un utilisateur peut devenir un utilisateur LDAP au moment de la configuration du client en exécutant la commande **mksecldap** avec l'indicateur **–u** sous l'une des formes suivantes :

1. Exécutez `mksecldap –c –u` *`user1,user2,...`*, où **user1,user2,...** est une liste d'utilisateurs. Les utilisateurs contenus dans cette liste peuvent être définis soit localement, soit à distance via LDAP. L'attribut **SYSTEM** a pour valeur LDAP dans chacune des strophes d'utilisateurs ci–dessus dans le fichier **/etc/security/user**. Ces utilisateurs ne seront authentifiés que par LDAP. Les utilisateurs de cette liste doivent exister sur le serveur d'informations de sécurité LDAP, sinon ils ne peuvent pas se connecter à partir de cet hôte. Exécutez la commande **chuser** pour modifier l'attribut **SYSTEM** et pour permettre l'authentification par plusieurs méthodes (locale et LDAP par exemple).

2. Exécutez “mksecldap –c –u ALL”. Cette commande donne à l’attribut **SYSTEM** la valeur LDAP dans chaque strophe utilisateur du fichier **/etc/security/user**, pour tous les utilisateurs définis localement. Tous ces utilisateurs ne s’authentifient que par LDAP. Les utilisateurs définis localement doivent exister sur le serveur d’informations de sécurité LDAP, sinon ils ne peuvent pas se connecter à partir de cet hôte. Un utilisateur défini sur le serveur LDAP, mais pas localement, ne peut pas se connecter à partir de cet hôte. Pour permettre à un utilisateur défini à distance via LDAP de se connecter à partir de cet hôte, lancez la commande **chuser** pour donner à l’attribut **SYSTEM** de cet utilisateur la valeur LDAP.

Vous pouvez également permettre à tous les utilisateurs LDAP, qu’ils soient définis localement ou pas, de s’authentifier par le biais de LDAP sur un hôte local en modifiant la strophe ”par défaut” du fichier **/etc/security/user** et utiliser la valeur ”LDAP”. Tous les utilisateurs dont la valeur de l’attribut **SYSTEM** n’est pas définie doivent suivre la définition présente dans la strophe par défaut. Par exemple, si la strophe par défaut est ”SYSTEM = “compat””, passer à “SYSTEM = ”compat OR LDAP”” authentifiera ces utilisateurs par AIX ou LDAP. La modification de la strophe par défaut en “SYSTEM = ”LDAP”” impose à ces utilisateurs de s’authentifier par LDAP. Les utilisateurs pour lesquels l’attribut **SYSTEM** est défini ne sont pas affectés par la strophe par défaut.

Contrôle d’accès par LDAP

AIX fournit à un système un contrôle d’accès à un hôte (connexion) au niveau utilisateur. Les administrateurs peuvent configurer les utilisateurs LDAP afin qu’ils se connectent à un système AIX en définissant leur attribut **SYSTEM** sur LDAP. L’attribut **SYSTEM** se trouve dans le fichier **/etc/security/user**. La commande **chuser** peut être utilisée pour définir sa valeur, comme dans l’exemple suivant :

```
# chuser –R LDAP SYSTEM=LDAP registry=LDAP foo
```

Remarque : Avec ce type de contrôle, ne définissez pas la valeur par défaut de l’attribut **SYSTEM** sur LDAP, car cela autoriserait tous les utilisateurs LDAP à se connecter au système.

L’attribut LDAP ainsi défini autorise l’utilisateur `foo` à se connecter au système. Le registre est également défini sur LDAP, ce qui permet au processus de connexion de consigner toutes les tentatives de connexion à LDAP de `foo`, et autorise également toutes les tâches de gestion des utilisateurs effectuées via LDAP.

Pour autoriser la connexion en fonction des utilisateurs, l’administrateur doit effectuer cette configuration sur chacun des systèmes clients.

A partir de AIX 5.2, l’accès des utilisateurs LDAP peut désormais être limité à certains systèmes clients LDAP. Cette fonction permet la gestion centralisée des contrôles d’accès. Les administrateurs peuvent spécifier deux listes de contrôle d’accès hôte pour un compte utilisateur : une liste d’autorisation et une liste de refus. Ces deux attributs sont enregistrés avec le compte utilisateur, sur le serveur LDAP. L’utilisateur est donc autorisé à accéder aux systèmes ou réseaux répertoriés dans la liste d’accès accordés, tandis que l’accès aux systèmes et réseaux spécifiés dans la liste d’accès refusés lui est interdit. Si un système apparaît dans les deux listes, son accès est refusé à l’utilisateur. Il existe deux procédures permettant de spécifier les listes d’accès pour un utilisateur : à l’aide de la commande **mkuser** lors de la création de l’utilisateur ou à l’aide de la commande **chuser** pour un utilisateur existant. Pour des raisons de compatibilité, si aucune de ces listes n’existe pour un utilisateur, l’accès à tous les systèmes client LDAP lui est accordé par défaut. Cette fonction de contrôle d’accès hôte est disponible à partir de la version AIX 5.2.

Voici des exemples de définition de listes d’accès accordés et refusés :

```
# mkuser –R LDAP hostsallowedlogin=host1,host2 foo
```

L’utilisateur `foo` est créé, et il n’est autorisé à se connecter qu’à `host1` et `host2`.

```
# mkuser –R LDAP hostsdeniedlogin=host2 foo
```

L’utilisateur `foo` est créé ; il est autorisé à se connecter à tous les systèmes clients LDAP, hormis `host2`.

```
# chuser -R LDAP hostsallowedlogin=192.9.200.1 foo
```

L'utilisateur `foo` est ici autorisé à se connecter au système client dont l'adresse est `192.9.200.1`.

```
# chuser -R LDAP hostsallowedlogin=192.9.200/24
hostsdeniedlogin=192.9.200.1 foo
```

L'utilisateur `foo` est autorisé à se connecter à tout client appartenant au sous–réseau `192.9.200/24`, excepté celui qui a pour adresse `192.9.200.1`.

Pour plus d'informations, reportez–vous à la commande **chuser** .

Communication sécurisée avec SSL

En fonction du type d'authentification utilisé entre le client et le serveur LDAP, les mots de passe sont transmis soit au format crypté (**unix_auth**), soit en texte clair (**ldap_auth**). Utilisez le protocole SSL (Secure Socket Layer) pour protéger contre les menaces à la sécurité l'envoi des mots de passe, même chiffrés, via le réseau, ou via Internet dans certains cas. AIX offre des modules pour SSL qui peuvent fournir une communication sécurisée entre les clients et serveurs d'annuaires.

Afin de configurer le protocole SLL sur le serveur LDAP, installez les ensembles de fichiers **ldap.max_crypto_server** et **GSKit** pour activer la prise en charge du chiffrement du serveur. Ces ensembles de fichiers sont disponibles sur le CD–Rom AIX Expansion Pack. Suivez ensuite cette procédure pour activer la prise en charge SSL pour l'authentification du serveur Directory.

1. Installez le progiciel **GSKit** fourni avec Directory si ce n'est déjà fait.
2. Générez la clé privée de serveur Directory et le certificat de serveur à l'aide de l'utilitaire **gsk7ikm** (installé avec **GSKit**). Le certificat de serveur peut être signée par une autorité de certification (CA) privée comme VeriSign, ou peut être autosignée au moyen de l'outil **gsk7ikm**. Le certificat public de l'autorité de certification (CA) doit également être transmis au fichier de base de données de clé de l'application client.
3. Enregistrez sur le serveur le fichier de base de données de clés de serveur et le fichier cachette de mots de passe associé. Le chemin par défaut pour la base de données de clés, à savoir le répertoire **/usr/ldap/etc**, est un emplacement type.
4. Exécutez la commande suivante pour une première installation de serveur :

```
# mksecldap -s -a cn=admin -p pwd -S rfc2307aix -k /usr/ldap/etc/mykey.kdb
-w   keypwd
```

où **mykey.kdb** est la base de données de clés et *keypwd* le mot de passe de la base de données de clés. Pour configurer un serveur déjà configuré et en cours de fonctionnement :

```
# mksecldap -s -a cn=admin -p pwd -S rfc2307aix -u NONE -k
/usr/ldap/etc/mykey.kdb -w   keypwd
```

Pour utiliser SSL sur un client LDAP, installez les ensembles de fichiers **ldap.max_crypto_client** et **GSKit** présents sur le CD–Rom Expansion Pack AIX. Effectuez ensuite les étapes suivantes pour activer la prise en charge SSL pour LDAP après l'activation du serveur pour SSL.

1. Exécutez **gsk7ikm** pour générer la base de données de clés sur chaque client.
2. Copiez le certificat du serveur sur chacun des clients. Si le serveur SSL utilise un certificat autosigné, le certificat doit être exporté en premier.
3. Sur chaque système client, exécutez **gsk7ikm** pour importer le certificat de serveur vers la base de données de clés.
4. Activez SSL pour chaque client :

```
# mksecldap -c -h servername -a adminDN -p pwd -k /usr/ldap/etc/mykey.kdb
-p   keypwd
```

où **/usr/ldap/etc/mykey.kdb** est le chemin complet vers la base de données de clés et *keypwd* le mot de passe de la clé. Si vous n'entrez pas le mot de passe de clé à partir de la ligne de commande, un fichier de mots de passe caché du même répertoire est utilisé. Le fichier caché doit porter le même nom que la base de données de clés suivi de l'extension **.sth** (par exemple **mykey.sth**).

Liaison Kerberos

En plus de la simple liaison utilisant un nom DN de liaison et un mot de passe de liaison, **secldapclntd** prend également en charge une liaison utilisant les données d'identification Kerberos V. Les clés du principal de liaison sont stockées dans un fichier keytab et doivent être accessibles pour le démon **secldapclntd** afin de permettre l'utilisation de la liaison Kerberos. Lorsque la liaison Kerberos est activée, le démon **secldapclntd** effectue l'authentification Kerberos sur le serveur LDAP à l'aide du nom du principal et du fichier keytab indiqué dans le fichier de configuration client **/etc/security/ldap/ldap.cfg**. Lorsque la liaison Kerberos est utilisée, le démon **secldapclntd** ignore le DN et le mot de passe de liaison indiqués dans le fichier **/etc/security/ldap/ldap.cfg**.

Lorsque l'authentification Kerberos aboutit, le démon **secldapclntd** enregistre les données d'identification de la liaison dans le répertoire **/etc/security/ldap/krb5cc_secldapclntd**. Les données d'identification enregistrées sont ensuite utilisées pour les liaisons ultérieures. Lorsque le démon **secldapclntd** tente d'établir à nouveau la liaison vers un serveur LDAP, si les données d'identification datent de plus d'une heure, le démon **secldapclntd** est réinitialisé avec de nouvelles données d'identification.

Pour configurer le système client LDAP de façon à utiliser la liaison Kerberos, vous devez configurer le client à l'aide de la commande **mksecldap** en utilisant un DN et un mot de passe de liaison. Une fois la configuration effectuée, modifiez le fichier **/etc/security/ldap/ldap.cfg** en indiquant les valeurs des attributs relatifs à Kerberos. Le démon **secldapclntd** utilise la liaison Kerberos au redémarrage. Après une configuration réussie, le DN et le mot de passe de liaison ne sont plus utilisés. Vous pouvez les supprimer sans risque du fichier **/etc/security/ldap/ldap.cfg** ou les mettre en commentaires.

Création d'un principal Kerberos

Pour assurer la prise en charge de la liaison Kerberos, il est nécessaire de créer au moins deux principaux sur le KDC (Key Distribution Center) qui seront utilisés par le serveur et le client IDS. Le premier principal est un principal de serveur LDAP et le second est le principal utilisé par les systèmes client pour établir la liaison au serveur.

Pour utiliser les clés de principal afin de lancer le processus serveur ou le processus démon client, placez chacune d'elle dans un fichier keytab.

L'exemple suivant concerne le service NAS (Network Authentication Service). Si vous installez les logiciels Kerberos d'autres sources, les commandes peuvent différer de celles qui apparaissent ici.

- Lancez l'outil kadmin sur le serveur KDC, en tant qu'utilisateur root.

```
#/usr/krb5/sbin/kadmin.local
kadmin.local:
```

- Créez le principal **ldap/** ***nom_hôte_serveur*** pour le serveur LDAP. *nom_hôte_serveur* est le nom DNS complet de l'hôte qui va exécuter le serveur LDAP.

```
kadmin.local: addprinc ldap/plankton.austin.ibm.com
WARNING: no policy specified for
"ldap/plankton.austin.ibm.com@ud3a.austin.ibm.com":
Re-enter password for principal
"ldap/plankton.austin.ibm.com@ud3a.austin.ibm.com":
Principal "ldap/plankton.austin.ibm.com@ud3a.austin.ibm.com" created.
kadmin.local:
```

- Créez un fichier keytab pour le principal serveur créé. Cette clé sera utilisée par le serveur LDAP lors du démarrage du serveur. Pour créer un fichier keytab appelé **slapd_krb5.keytab** :

```
kadmin.local: ktadd -k /etc/security/slapd_krb5.keytab
ldap/plankton.austin.ibm.com
Entry for principal ldap/plankton.austin.ibm.com with kvno 2,
encryption type Triple DES cbc mode with HMAC/sha1 added to keytab
WRFILE:/etc/security/slapd_krb5.keytab.
Entry for principal ldap/plankton.austin.ibm.com with kvno 2,
encryption type ArcFour with HMAC/md5 added to keytab
WRFILE:/etc/security/slapd_krb5.keytab.
Entry for principal ldap/plankton.austin.ibm.com with kvno 2,
encryption type AES-256 CTS mode with 96-bit SHA-1 HMAC added to keytab
WRFILE:/etc/security/slapd_krb5.keytab.
Entry for principal ldap/plankton.austin.ibm.com with kvno 2,
encryption type DES cbc mode with RSA-MD5 added to keytab
WRFILE:/etc/security/slapd_krb5.keytab.
kadmin.local:
```

- Créez un principal appelé **ldapadmin** pour l'administrateur IDS.

```
kadmin.local: addprinc ldapadmin
WARNING: no policy specified for ldapadmin@ud3a.austin.ibm.com; defaulting
to no policy.
Note that policy may be overridden by ACL restrictions.
Enter password for principal "ldapadmin@ud3a.austin.ibm.com":
Re-enter password for principal "ldapadmin@ud3a.austin.ibm.com":
Principal "ldapadmin@ud3a.austin.ibm.com" created.
kadmin.local:
```

- Créez un fichier keytab pour le principal de liaison **kdapadmin.keytab**. Cette clé peut être utilisée par le démon client **secldapclntd**.

```
kadmin.local: ktadd -k /etc/security/ldapadmin.keytab ldapadmin
Entry for principal ldapadmin with kvno 2, encryption type
Triple DES cbc mode with HMCA/sha1 added to keytab
WRFILE:/etc/security/ldapadmin.keytab.
Entry for principal ldapadmin with kvno 2, encryption type
ArcFour with HMAC/md5 added to keytab
WRFILE:/etc/security/ldapadmin.keytab.
Entry for principal ldapadmin with kvno 2, encryption type
AES-256 CTS mode with 96-bit SHA-1 HMAC added to keytab
WRFILE:/etc/security/ldapadmin.keytab.
Entry for principal ldapadmin with kvno 2, encryption type
DES cbc mode with RSA-MD5 added to keytab
WRFILE:/etc/security/ldapadmin.keytab.
kadmin.local
```

- Créez un principal appelé **ldapproxy** pour les clients à relier au serveur LDAP.

```
kadmin.local: addprinc ldapproxy
WARNING: no policy specified for ldapproxy @ud3a.austin.ibm.com; defaulting
to no policy.
Note that policy may be overridden by ACL restriction
Enter password for principal "ldapproxy@ud3a.austin.ibm.com":
Re-enter password for principal "ldapproxy@ud3a.austin.ibm.com":
Principal "ldapproxy@ud3a.austin.ibm.com" created.
kadmin.local:
```

- Créez un fichier keytab appelé **ldapproxy.keytab** pour le principal de liaison **ldapproxy**. Cette clé peut être utilisée par le démon client **secldapclntd**.

```
kadmin.local: ktadd -k /etc/security/ldapproxy.keytab ldapproxy
Entry for principal ldapproxy with kvno 2, encryption type
Triple DES cbc mode with HMAC/sh1 added to keytab
WRFILE:/etc/security/ldapproxy.keytab.
Entry for principal ldapproxy with kvno 2, encryption type
ArcFour with HMAC/md5 added to keytab WRFILE:/etc/security/ldapproxy.keytab
Entry for principal ldapproxy with kvno 2, encryption type
AES-256 CTS mode with 96-bit SHA-1 HMAC added to keytab
WRFILE:/etc/security/ldapproxy.keytab
Entry for principal ldapproxy with kvno 2,
encryption type DES cbc mode with RSA-MD5 added to keytab
WRFILE:/etc/security/ldapproxy.keytab.
kadmin.local:
```

Activation de la liaison Kerberos sur un serveur IDS

L'exemple ci–dessous montre comment configurer un serveur IDS pour activer la liaison Kerberos.

Cet exemple a été testé avec IDS v 5.1

1. Installez l'ensemble de fichiers **krb5.client**.
2. Vérifiez que le fichier **/etc/krb5/krb5.conf** existe et qu'il est correctement configuré. Pour le configurer, lancez la commande **/usr/sbin/config.krb5**.

```
# config.krb5 -r ud3a.austin.ibm.com -d austin.ibm.com -c KDC -s
alyssa.austin.ibm.com
Initializing configuration...
Creating /etc/krb5/krb5_cfg_type...
Creating /etc/krb5/krb5.conf...
The command completed successfully.
# cat /etc/krb5/krb5.conf
[libdefaults]
      default_realm = ud3a.austin.ibm.com
      default_keytab_name=FILE:/etc/krb5/krb5.keytab
      default_tkt_enctypes = des3-cbc-sha1 arcfour-hmac aes256-cts
des-cbc-md5 des-cbc-crc
      defaut_tgs_enctypes = des3-cbc-sha11 arcfour-hmac aes256-cts
des-cbc-md5 des-cbc-crc
[realms]
   ud3a.austin.ibm.com = {
      kdc = alyssa.austin.ibm.com:88
      admin_server = alyssa.austin.ibm.com:749
      default_domain = austin.ibm.com
   }

[domain_realm]
      .austin.ibm.com = ud3a.austin.ibm.com
      alyssa.austin.ibm.com = ud3a.austin.ibm.com

[logging]
      kdc = FILE:/var/krb5/log/krb5
      admin_server = FILE:/var/krb5/log/kadmin.log
      default = FILE:/var/krb5/log/krb5lib.log
```

3. Récupérez le fichier keytab du principal **ldap:/** ***nom_hôte_serveur*** et placez–le dans le répertoire **/usr/ldap/etc**. Par exemple : **/usr/ldap/etc/slapd_krb5.keytab**.
4. Définissez les droits d'accès afin d'autoriser le processus serveur à accéder au fichier.

```
# chown ldap:ldap/usr/ldap/etc/slapd_krb5.keytab #
```

5. Pour activer la liaison Kerberos sur le serveur IDS, modifiez le fichier **/etc/ibmslapd.conf** et ajoutez l'entrée suivante :

```
dn: cn=Kerberos, cn-Configuration
cn: Kerberos
ibm-slapdKrbAdminDN: ldapadmin
ibm-slapdKrbEnable: true
ibm-slapdKrbIdentityMap: true
ibm-slapdKrbKeyTab: /usr/ldap/etc/slapd_krb5.keytab
ibm-slapdKrbRealm: ud3a.austin.ibm.com
objectclass: ibm-slapdKerberos
objectclass: ibm-slapdconfigEntry
objectclass: top
```

6. Mappez le principal **ldapproxy** au DN de liaison nommé *cn–proxyuser,cn=aixdata*.

   a. Si l'entrée DN de laison existe, créez un fichier nommé **ldapproxy.ldif** avec le contenu suivant :

   ```
   dn: cn=proxyuser,cn=aixdata
    changetype: modify
    add: objectclass
    objectclass: ibm-securityidentities
    -
    add:altsecurityidentities
    alsecurityidentities: Kerberos:ldapproxy@ud3a.austin.ibm.com
   ```

   OU

   b. Si l'entrée DN de laison n'a pas été ajoutée au serveur, créez un fichier nommé *proxyuser.ldif* avec le contenu suivant :

   Remarque : Remplacez *proxyuserpwd* par votre mot de passe

   ```
   dn: cn=proxyuser,cn=mytest
   cn: proxyuser
   sn: proxyuser
   userpassword: proxyuserpwd
   objectclass: person
   objectclass: top
   objectclass: ibm-securityidentities
   altsecurityidentities: Kerberos:ldapproxy@ud3a.austin.ibm.com
   Ajoutez l'entrée DN de liaison créée sur le serveur IDS, à l'aide de
   la commande ldapmodify.
   ```

```
# ldapmodify –D cn–admin –w adminPwd –f /tmp/proxyuser.ldif modifying entry
cn=proxyuser,cn=mytest #
```

7. Redémarrez le serveur IDS.

Activation de la liaison Kerberos sur le client LDAP AIX

Cette section explique comment configurer un système client LDAP AIX de façon à utiliser Kerberos lors de la liaison initiale vers le serveur LDAP.

Le serveur IDS doit être configuré de cette façon afin que l'hôte serveur soit client de lui–même.

Cet exemple a été testé avec IDS v 5.1.

1. Installez l'ensemble de fichiers **krb5.client**.
2. Vérifiez que le fichier **/etc/krb.conf** existe et qu'il est correctement configuré. Pour le configurer, lancez la commande **/usr/sbin/config.krb5**.
3. Récupérez le fichier keytab du principal de liaison et placez–le dans le répertoire **/etc/security/ldap**.
4. Sélectionnez le droit d'accès 600.
5. Configurez le client à l'aide de la commande **mksecldap** en utilisant le DN et le mot de passe de liaison. Vérifiez que les commandes AIX fonctionnent sur les utilisateurs LDAP.

6. Modifiez le fichier **/etc/security/ldap/ldap.cfg** afin de définir les attributs relatifs à Kerberos. Dans l'exemple suivant, le principal de liaison est *ldapproxy* et le fichier keytab est *ldapproxy.keytab*. Si vous souhaitez avoir des droits d'accès d'administrateur du serveur IDS, remplacez *ldapproxy* par *ldapadmin* et *ldapproxy.keytab* par *ldapadmin.keytab*.

```
useKRB5:yes
krbprincipal:ldapproxy
krbkeypath:/etc/security/ldap/ldapproxy.keytab
krbcmddir:/usr/krb5/bin/
```

   Vous pouvez maintenant supprimer le DN et le mot de passe de liaison du **ldap.cfg** ou les mettre en commentaires, car le démon **secldapclntd** utilise la liaison Kerberos.

7. Relancez le démon **secldapclntd**.
8. Le fichier **/etc/security/ldap/ldap.cfg** peut alors être diffusé aux autres systèmes clients.

Audit du serveur d'informations de sécurité LDAP

SecureWay Directory version 3.2 (et versions ultérieures) offre une fonctionnalité d'enregistrement des audits serveur par défaut. Une fois activé, ce module complémentaire d'audit enregistre les activités du serveur LDAP dans un fichier journal. Pour plus d'informations sur ce module complémentaire, reportez–vous au manuel *Packaging Guide for LPP* dans la documentation LDAP.

Une fonction d'audit de serveur d'informations de sécurité LDAP a été mise en place dans AIX 5.1 et dans les versions ultérieures ; il s'agit du *plug–in d'audit de sécurité LDAP*. Elle est indépendante du service d'audit par défaut de SecureWay Directory. Il est donc possible d'activer l'un ou l'autre sous–système d'audit, ou les deux. Le plug–in d'audit d'AIX n'enregistre que les événements qui mettent à jour ou demandent des informations sur la sécurité d'AIX sur un serveur LDAP. Il fonctionne dans le cadre d'audit du système AIX.

Pour accepter LDAP, les événements audit suivants sont compris dans le fichier **/etc/security/audit/event** :

- **LDAP_Bind**
- **LDAP_Unbind**
- **LDAP_Add**
- **LDAP_Delete**
- **LDAP_Modify**
- **LDAP_Modifydn**
- **LDAP_Search**

La définition de classe d'audit **ldapserver** est également créée dans le fichier **/etc/security/audit/config** qui contient tous les événements cités ci–dessus.

Pour que le serveur d'informations de sécurité LDAP subisse un audit, ajoutez la ligne suivante à chaque strophe d'utilisateur dans le fichier **/etc/security/audit/config** :

```
ldap = ldapserver
```

Comme le plug–in d'audit du serveur d'informations de sécurité LDAP est implémenté dans le cadre de l'audit du système AIX, il fait partie de ce sous–système. Vous pouvez activer ou désactiver l'audit du serveur d'informations de sécurité LDAP à l'aide de commandes audit du système, comme **audit start** (démarrer l'audit) ou **audit shutdown** (arrêter l'audit). Tous les enregistrements d'audit sont ajoutés aux traces d'audit du système, qui peuvent être consultées à l'aide de la commande **auditpr**. Pour plus d'informations, reportez–vous à la section Audit, page 4-1.

Commandes LDAP

Commande mksecldap

La commande **mksecldap** peut servir à configurer des serveurs et des clients SecureWay Directory pour l'authentification de sécurité et la gestion des données. Elle doit être exécutée sur le serveur, ainsi que sur tous les clients.

Démon secldapclntd

Le démon **secldapclntd** reçoit des requêtes de la part des modules de chargement LDAP, les transmet au serveur d'informations de sécurité LDAP, puis renvoie les résultats au module de chargement LDAP.

Commandes de gestion LDAP

Commande start–secldapclntd

La commande **start–secldapclntd** lance le démon **secldapclntd** s'il n'est pas déjà lancé.

Commande stop–secldapclntd

La commande **stop–secldapclntd** termine le processus démon **secldapclntd** en cours d'exécution.

Commande restart–secldapclntd

Le script **restart–secldapclntd** arrête le démon **secldapclntd** s'il est en cours d'exécution, puis le redémarre. Si le démon **secldapclntd** n'est pas en cours d'exécution, le script le lance directement.

Commande ls–secldapclntd

La commande **ls–secldapclntd** affiche l'état du démon **secldapclntd**.

Commande flush–secldapclntd

La commande **flush–secldapclntd** nettoie le cache pour le processus du démon **secldapclntd**.

Commande sectoldif

La commande **sectoldif** permet de récupérer le nom des utilisateurs et des groupes définis localement et d'envoyer le résultat au format **ldif** vers la sortie standard.

Le format de fichier ldap.cfg

Le fichier **/etc/security/ldap/ldap.cfg** contient les informations nécessaires au démarrage et au bon fonctionnement du démon **secldapclntd**, ainsi que des informations permettant d'ajuster ses performances. Il est mis à jour par la commande **mksecldap** lors de la configuration du client.

Pour plus d'informations sur le fichier **/etc/security/ldap/ldap.cfg**, reportez–vous à la description de **/etc/security/ldap/ldap.cfg** dans le manuel *AIX 5L Version 5.3 Files Reference*.

Mappage de format de fichier pour les attributs LDAP

Ces fichiers de mappage sont utilisés par le module **/usr/lib/security/LDAP** et le démon **secldapclntd** pour convertir les noms des attributs AIX en noms d'attributs LDAP. Chaque entrée dans le fichier de mappage correspond à la traduction d'un attribut. Chaque entrée comporte quatre champs séparés par des espaces :

```
AIX_Attribute_Name AIX_Attribute_Type LDAP_Attribute_Name LDAP_Value_Type
```

AIX_Attribute_Name	Indique le nom de l'attribut AIX.
AIX_Attribute_Type	Indique le type de l'attribut AIX. Les valeurs admises sont SEC_CHAR, SEC_INT, SEC_LIST et SEC_BOOL.
LDAP_Attribute_Name	Indique le nom de l'attribut LDAP.
LDAP_Value_Type	Indique le type de valeur LDAP. **s** indique une valeur unique et **m** des valeurs multiples.

Pour plus d'informations sur le format de fichier de mappage des attributs, reportez–vous à la section **Format de fichier de mappage des attributs LDAP** dans le manuel *AIX 5L Version 5.3 Files Reference*.

Informations connexes

Commandes **mksecldap**, **start–secldapclntd**, **stop–secldapclntd**, **restart–secldapclntd**, **ls–secldapclntd**, **sectoldif** et **flush–secldapclntd**.

Démon **secldapclntd**.

Fichier **/etc/security/ldap/ldap.cfg**.

Format de fichier de mappage des attributs LDAP.

Chapitre 6. PKCS #11

Le sous–système PKCS #11 fournit des applications offrant une méthode pour accéder aux unités matérielles (jetons), indépendamment du type d'unité. Le contenu de ce chapitre est conforme à la Version 2.01 de la norme PKCS #11.

Le sous–système PKCS #11 a été implémenté à l'aide des composants suivants :

- Un démon gestionnaire de slot (**pkcsslotd**), qui fournit au sous–système des informations sur l'état des unités matérielles disponibles. Ce démon est démarré automatiquement lors de l'installation ou du redémarrage du système.
- Un objet API partagé (**/usr/lib/pkcs11/pkcs11_API.so**) est fourni comme interface générique des cartes pour lesquelles PKCS #11 a été implémentée.
- Une bibliothèque pour chaque carte, qui fournit le support PKCS #11 spécifique. Cette conception étagée permet à l'utilisateur d'utiliser de nouvelles unités PKCS #11 lorsqu'elles sont disponibles sans avoir à recompiler des applications existantes.

Ce chapitre traite des points suivants :

Coprocesseur de chiffrement 4758 Model 2

Le coprocesseur de chiffrement 4758 Model 2 sécurise l'environnement informatique. Avant de tenter de configurer le sous-système PKCS #11, vérifiez que la carte a été correctement configurée avec un microcode compatible.

Vérification du coprocesseur de chiffrement 4758 Model 2 pour une utilisation avec le sous-système PKCS #11

Le sous–système PKCS #11 détecte automatiquement lors de l'installation et du redémarrage les cartes acceptant les appels PKCS #11. C'est pourquoi un coprocesseur de chiffrement 4758 Model 2 qui n'est pas correctement configuré ne sera pas accessible depuis l'interface PKCS #11 et les appels envoyés échoueront. Procédez comme suit pour vérifier si votre carte est correctement configurée :

1. Assurez–vous que le logiciel de la carte est correctement installé à l'aide de la commande suivante :

```
lsdev -Cc adapter | grep crypt
```

   Si le coprocesseur de chiffrement 4758 Model 2 ne s'affiche pas dans la liste des résultats, vérifiez que la carte est correctement insérée et que le logiciel de prise en charge est correctement installé.

2. Vérifiez que le microcode approprié a été chargé sur la carte en entrant la commande suivante :

```
csufclu /tmp/l ST   device_number_minor
```

   Vérifiez que l'Image Segment 3 montre l'application PKCS #11 chargée. Si ce n'est pas le cas, consultez la documentation de la carte pour obtenir les dernières informations relatives au microcode et à l'installation.

   Remarque : Si cet utilitaire n'est pas disponible, il faudra installer le logiciel de prise en charge.

Configuration du sous-système PKCS #11

Le sous-système PKCS #11 détecte automatiquement les unités compatibles. Toutefois, pour utiliser ces unités, certaines applications ont besoin d'une configuration initiale. Ces tâches sont les suivantes :

- Initialisation du jeton, page 6-2
- Configuration du PIN de l'agent de sécurité, page 6-2
- Initialisation du PIN utilisateur, page 6-3

Vous pouvez effectuer ces tâches via l'API (en écrivant une application PKCS #11) ou à l'aide de l'interface SMIT. Les options SMIT de PKCS #11 sont accessibles via le **sous-système Gestion du PKCS11** depuis le menu principal SMIT, ou à l'aide du raccourci **smit pkcs11**.

Initialisation du jeton

Chaque carte ou jeton PKCS #11 doit être initialisé avant de pouvoir être utilisé. Cette procédure implique la définition d'un libellé unique pour le jeton. Ce libellé permet aux applications d'identifier le jeton par un numéro d'ordre unique. Par conséquent, les libellés ne doivent pas être répétés. Toutefois, l'API ne vérifie pas si les libellés sont réutilisés ou non. Cette initialisation peut se faire par une application PKCS #11, ou par l'administrateur du système avec l'interface SMIT. Si votre jeton dispose d'un PIN du responsable de sécurité, la valeur par défaut est 87654321. Pour garantir la sécurité du sous-système PKCS #11, cette valeur doit être modifiée après l'initialisation.

Pour initialiser le jeton :

1. Affichez l'écran de gestion du jeton en tapant `smit pkcs11`.
2. Sélectionnez **Initialiser un jeton**.
3. Sélectionnez une carte PKCS #11 dans la liste de celles prises en charge.
4. Appuyez sur Entrée pour confirmer votre choix.

   Remarque : Cette action effacera toutes informations figurant sur le jeton.
5. Entrez le PIN du responsable de sécurité (SO PIN) et un libellé unique du jeton.

Si le PIN correct est entré, la carte sera initialisée ou réinitialisée une fois l'exécution de la commande terminée.

Configuration du PIN du responsable de sécurité

Si votre jeton dispose d'un SO PIN, vous pouvez en modifier la valeur par défaut, comme suit :

1. Tapez `smit pkcs11`.
2. Sélectionnez **Configurer le PIN du responsable de sécurité**.
3. Sélectionnez la carte initialisée pour laquelle vous voulez configurer le SO PIN.
4. Entrez le SO PIN courant et un nouveau PIN.
5. Vérifiez le nouveau PIN.

Initialisation du PIN utilisateur

Une fois le jeton initialisé, vous devrez peut–être configurer le PIN utilisateur pour permettre aux applications d'accéder aux objets du jeton. Consultez la documentation spécifique à votre unité pour déterminer si elle nécessite la connexion d'un utilisateur avant de pouvoir accéder aux objets.

Pour initialiser le PIN utilisateur :

1. Affichez l'écran de gestion du jeton en tapant `smit pkcs11`.
2. Sélectionnez **Initialiser le PIN utilisateur**.
3. Sélectionnez une carte PKCS #11 dans la liste.
4. Entrez le SO PIN et le PIN utilisateur.
5. Vérifiez le PIN utilisateur.
6. Après vérification, le PIN utilisateur sera modifié.

Reconfiguration du PIN utilisateur

Pour reconfigurer le PIN utilisateur, vous pouvez réinitialiser le PIN à l'aide du SO PIN ou définir le PIN utilisateur à l'aide de celui existant. Pour ce faire :

1. Affichez l'écran de gestion du jeton en tapant `smit pkcs11`.
2. Sélectionnez **Définir le PIN utilisateur**.
3. Sélectionnez la carte initialisée pour laquelle vous voulez configurer le PIN utilisateur.
4. Entrez le PIN utilisateur courant et un nouveau PIN.
5. Vérifiez le nouveau PIN utilisateur.

Utilisation de PKCS #11

Pour qu'une application puisse utiliser le sous-système PKCS #11, le démon gestionnaire d'emplacements du sous- doit être actif, et l'application doit charger l'objet d'API partagé.

Le gestionnaire d'emplacements est généralement lancé au démarrage par **inittab**, avec le script **/etc/rc.pkcs11**. Ce script vérifie les cartes du système avant de lancer le démon de gestionnaire d'emplacements. Par conséquent, ce démon n'est pas disponible avant que l'utilisateur ne soit connecté au système. Une fois le démon lancé, le sous-système intègre toutes les modifications apportées au nombre et aux types de cartes prises en charge, sans aucune intervention de l'administrateur système.

L'API peut être chargée en faisant un link de l'objet au moment de l'exécution ou en utilisant une résolution différée de symboles. Par exemple, une application peut obtenir la liste des fonctions PKCS #11 de la manière suivante :

```
d CK_RV (*pf_init)();
 void *d;
 CK_FUNCTION_LIST *functs;

 d = dlopen(e, RTLD_NOW);
 if ( d == NULL ) {
    return FALSE;
 }

 pfoo = (CK_RV (*)())dlsym(d, "C_GetFunctionList");
 if (pfoo == NULL) {
    return FALSE;
 }

 rc = pf_init(&functs);
```

Chapitre 7. Service d'authentification de certificats X.509 et infrastructure à clé publique

Le Service d'authentification des certificats permet au système d'exploitation AIX 5.2 d'authentifier les utilisateurs à l'aide de certificats X.509 d'infrastructure à clef publique (PKI, Public Key Infrastructure), et d'associer ces certificats à des processus pour confirmer l'identité d'un utilisateur. Pour ce faire, il utilise LAMF (Loadable Authentication Module Framework), le même mécanisme d'extension AIX utilisé pour les méthodes d'authentification DCE, Kerberos et autres.

Cette section traite des points suivants :

Présentation du service d'authentification de certificats

Chaque compte utilisateur participant à une authentification PKI possède un certificat PKI unique. En association avec un mot de passe, ce certificat authentifie l'utilisateur lors de sa connexion. Les certificats PKI reposent sur le système de clef publique/clef privée. Ce système utilise deux clefs non symétriques pour chiffrer et déchiffrer les données. Les données chiffrées à l'aide d'une clef ne peuvent être déchiffrées qu'avec l'autre clef. L'utilisateur conserve la clef privée et l'enregistre dans un magasin de clefs privé, et publie l'autre clef, la clef publique, sous la forme d'un certificat. Les certificats sont généralement conservés sur un serveur LDAP (Lightweight Directory Access Protocol), soit pour une utilisation interne à l'entreprise, soit sur Internet pour une utilisation dans le monde entier.

Pour que John puisse envoyer des données déchiffrables uniquement par Kathy, le certificat publié pour Kathy fournira à John la clef publique nécessaire pour chiffrer les données, qui pourront alors être envoyées à Kathy. Kathy déchiffrera les données envoyées par John à l'aide de sa clef privée, conservée dans son magasin de clefs privé.

Ce système est également utilisé pour les signatures numériques. Pour envoyer à John des données validées par sa signature numérique, Kathy devra utiliser sa clef privée. John utilisera la clef publique contenue dans le certificat publié de Kathy pour vérifier la signature numérique attachée aux données.

Dans les deux cas, la clef privée de Kathy est conservée dans un magasin de clefs privé. Il existe divers types de magasins de clefs privés, par exemple des fichiers ou des smart cards, mais tous protègent toutes les clefs privées au moyen de mots de passe ou de numéros PIN (Personal Identification Numbers). Ils assurent généralement le stockage de plusieurs clefs privées ainsi que de certificats et d'objets PKI. Les utilisateurs disposent habituellement de leurs propres magasins de clefs.

Le service d'authentification de certificats utilise les signatures numériques pour authentifier un utilisateur lors de la connexion. Il repère le magasin de clefs et le certificat de l'utilisateur en fonction de son nom de compte, utilise le mot de passe de l'utilisateur pour trouver dans le magasin la clef privée correspondant au certificat, signe les données avec cette clef privée, et utilise la clef publique de l'utilisateur contenu dans le certificat pour vérifier la signature. Une fois l'utilisateur authentifié, le système enregistre le certificat en mémoire protégée et l'associe à chaque processus créé par l'utilisateur. Cette association en mémoire permet l'accès rapide au certificat de l'utilisateur pour tous les processus qu'il possède, ainsi que pour ceux détenus par le noyau du système d'exploitation.

Certificats

Pour comprendre le service d'authentification des certificats, vous devez posséder quelques connaissances sur les certificats, leurs formats et la gestion de leur cycle de vie. Les certificats sont des objets standardisés conformes à la norme X.509, dont X.509v3 est la toute dernière version. Les certificats sont créés, signés et émis par une autorité de certification (CA), la plupart du temps un logiciel qui accepte et traite les demandes de certificats. Il existe plusieurs attributs de certificats. Certains d'entre eux sont obligatoires, d'autres sont facultatifs. Voici les attributs de certificats les plus couramment utilisés et traités dans ce document :

- Version des certificats – Le numéro de version X.509 (1, 2 ou 3).
- Numéro de série – Un numéro unique, qui différencie un certificat parmi tous ceux qui ont été émis par la même autorité de certification.
- Nom de l'émetteur – Le nom de l'autorité de certification qui a émis le certificat.
- Période de validité – La date d'activation et d'expiration du certificat.
- Clef publique – La clef publique.
- Utilisateur DN – Nom du propriétaire du certificat.
- E–mail du nom d'utilisateur supplémentaire – Adresse e–mail du propriétaire.
- URI du nom d'utilisateur supplémentaire – L'URI/URL du site Web du propriétaire.

Chaque certificat possède un numéro de version, qui indique à quelle version de la norme X.509 il est conforme. Chaque certificat possède un numéro de série unique qui le différencie de tous les autres certificats émis par la même autorité de certification. Le numéro de série n'est unique que pour l'autorité de certification émettrice. Le nom d'émetteur du certificat identifie l'autorité de certification émettrice.

Les certificats ne sont valides qu'entre deux dates spécifiées : la date " Pas avant " et la date " Pas après ". Les certificats peuvent donc être créés avant leur date de début de validité. La durée de vie d'un certificat est en moyenne de 3 mois à 5 ans.

L'utilisateur DN indique le propriétaire du certificat, dans un format de dénomination spécial, le Nom spécifique (Distinguished Name, DN). Un DN peut indiquer le pays, l'entreprise, la ville, l'état, le nom du propriétaire et d'autres attributs associés au demandeur (généralement une personne, mais pas seulement). L'e–mail du nom d'utilisateur supplémentaire indique l'adresse e–mail du propriétaire, et l'URI du nom d'utilisateur supplémentaire permet d'indiquer l'URI/URL du site Web du propriétaire.

Autorités de certification et certificats

Les autorités de certification émettent, enregistrent et généralement publient les certificats. La publication des certificats se fait souvent sur un serveur LDAP, puisque LDAP offre à tous un accès facile aux données d'une communauté.

Les autorités de certification assurent également la révocation des certificats et la gestion des listes de révocation de certificats (CRL). La révocation d'un certificat consiste à publier le fait qu'il n'est plus valide, pour des raisons autres que l'expiration de sa période de validité. Comme de nombreuses ces copies des certificats peuvent être conservées et utilisées indépendamment du contrôle de l'autorité de certification émettrice, les autorités de certification publient une liste des certificats révoqués dans une CRL, de sorte que des entités extérieures puisse interroger la liste. Il est alors de leur responsabilité de s'assurer que le certificat est valide, en comparant la copie du certificat à la CRL de l'autorité de certification émettrice. Une autorité de certification ne peut révoquer que des certificats qu'elle crée ou émet. Elle ne peut pas révoquer des certificats émis par d'autres autorités de certification.

Voici quelques raisons administratives justifiant la révocation d'un certificat :

- Sécurité compromise de la clef privée du certificat.
- Le propriétaire du certificat a quitté l'entreprise.
- Sécurité compromise de l'autorité de certification.

Les autorités de certification disposent également de leur propre certificat d'identification. Elles peuvent ainsi s'identifier entre elles, par exemple dans une communication pair à pair (chaînes de sécurité, etc.).

De nombreuses autorités de certification utilisent le CMP (Certificate Management Protocol) pour demander et révoquer des certificats. Ce protocole dispose de plusieurs méthodes pour établir une connexion sécurisée entre un client (également appelé Entité de fin) et l'autorité de certification, mais tous les clients et autorités n'acceptent pas toutes les méthodes. L'une des méthodes courantes nécessite l'utilisation d'un numéro de référence et d'un mot de passe reconnus par l'autorité de certification pour chaque création de certificat ou demande de révocation. D'autres informations telles qu'un certificat spécial reconnu par l'autorité de certification peuvent également être requises. Les demandes de révocation peuvent nécessiter la clef privée associée au certificat à révoquer.

Même si le CMP permet de créer des certificats et d'effectuer des demandes de révocation, il ne prend pas en charge les requêtes CRL. Les CRL sont généralement accessibles via des méthodes hors bande. Les CRL sont fréquemment publiées sur des serveurs LDAP, ce qui permet aux applications de les y récupérer pour les examiner. Une autre nouvelle méthode est le protocole OCSP (Online Certification Status Protocol), mais il n'est pas encore accepté par toutes les autorités de certification.

Les autorités de certification sont généralement contrôlées par des organisations gouvernementales ou des entreprises privées de confiance qui visent à assurer la correspondance entre le certificat émis et la personne qui a demandé l'émission. L'expression *émission d'un certificat* implique la création d'un certificat, elle diffère donc d'une demande de copie d'un certificat publié.

Format de stockage des certificats

Le format de stockage le plus courant est ASN.1 (Abstract Syntax Notation version 1) ; il utilise les DER (Distinguished Encoding Rules). Il est donc appelé *format DER*.

Magasins de clefs

Un magasin de clefs (parfois appelé *ensemble de clefs*) contient les clefs privées d'un utilisateur correspondant aux clefs publiques de leurs certificats. Un libellé unique est attribué à chaque clef privée, généralement par l'utilisateur, pour faciliter l'identification. Les magasins de clefs sont protégés par un mot de passe, qui doit être saisi avant de pouvoir accéder aux clefs ou en ajouter. Les utilisateurs possèdent généralement leurs propres magasins de clefs. Il existe plusieurs formes de magasins de clefs, par exemple : smart cards, magasin sous LDAP, magasin sur fichier, etc. Les méthodes utilisées pour y accéder varient également, ainsi que les formats utilisés pour enregistrer les clefs privées. Le service d'authentification des certificats n'accepte que les magasins de clefs sur fichier.

Mise en œuvre du service d'authentification de certificats

Le service d'authentification de certificats fonctionne dans un modèle client/serveur. Le côté serveur consiste en une autorité de certification (CA) pour créer et conserver des certificats X.509 version 3 et des listes de révocation de certificats (CRL). (Une seule autorité de certification est généralement utilisée pour l'ensemble de l'organisation.) Le côté client contient le logiciel (commandes, bibliothèques, modules de chargement et fichiers de configuration) requis par chaque système participant à une authentification PKI. Les modules d'installation du serveur sont **cas.server**, ceux du client sont **cas.client**.

Création de comptes utilisateur PKI

Pour créer un compte utilisateur PKI, utilisez la commande AIX **mkuser**. Une fois créé, chaque compte dispose d'un certificat et d'un magasin de clefs privé. (vous pouvez également transformer des comptes existants en comptes PKI, mais des étapes supplémentaires sont nécessaires.) L'administrateur fournit à chaque nouvel utilisateur le mot de passe du magasin de clefs, afin qu'il puisse se connecter au système et modifier le mot de passe de son magasin de clefs.

Flux de données d'authentification utilisateur

Cette section décrit comment authentifier un utilisateur PKI. Les utilisateurs peuvent associer plusieurs certificats à leurs comptes. Une valeur d'indicateur unique définie par l'utilisateur facilite l'identification de chaque certificat, mais il n'est possible de spécifier qu'un seul certificat d'authentification. Le service d'authentification de certificats utilise l'attribut **auth_cert** pour connaître le certificat d'authentification de chaque utilisateur. La valeur de l'attribut **auth_cert** correspond à la valeur de l'indicateur du certificat.

Les certificats, indicateurs, emplacements de magasin de clefs correspondants, libellés de clef correspondants et autres informations connexes sont conservés sous LDAP pour chaque utilisateur. La combinaison de l'indicateur et du nom de l'utilisateur permet au service d'authentification de certificats de trouver le certificat sur le serveur LDAP. Pour plus d'informations sur la couche LDAP PKI, consultez Couche LDAP PKI (stockage des certificats), page 7-8.

Lors de la connexion, les utilisateurs donnent un nom d'utilisateur et un mot de passe. A partir du nom, le système récupère l'attribut **auth_cert** de l'utilisateur, puis l'indicateur du certificat d'authentification. En associant le nom d'utilisateur et l'indicateur, le système récupère sur le serveur LDAP le certificat, l'emplacement du magasin de clefs et le libellé de clef correspondant de l'utilisateur. Il vérifie la période de validité indiquée dans le certificat pour déterminer s'il a expiré ou n'a pas encore atteint sa date d'activation. Le système extrait ensuite la clef privée de l'utilisateur à l'aide de l'emplacement du magasin de clefs, du libellé de clef et du mot de passe fourni. Une fois la clef privée récupérée, le système vérifie que la clef privée et le certificat correspondent à l'aide d'un processus de signature interne. Si c'est le cas, l'utilisateur a réussi l'étape d'authentification PKI de la procédure de connexion. (ce qui ne signifie pas qu'il est connecté. Plusieurs autres vérifications sont effectuées par le système AIX sur un compte utilisateur avant d'autoriser l'accès au système.)

Pour utiliser un certificat comme certificat d'authentification, vous devez le signer à l'aide d'une clef de signature sécurisée. La signature est enregistrée sous LDAP avec le certificat, pour référence ultérieure. Cette mise en œuvre exige qu'un certificat possède une signature avant de pouvoir attribuer l'indicateur à **auth_cert**.

Le processus d'authentification ne compare pas un certificat à une CRL. Cela s'explique par des raisons de performance (il faut du temps pour obtenir et examiner les CRL qui peuvent être temporairement indisponibles), mais également par les délais de publication des CRL (les autorités de certification peuvent nécessiter une heure ou plus pour indiquer la révocation d'un certificat révoqué dans une CRL, ce qui fait de la révocation de certificats une alternative limitée à la désactivation d'un compte utilisateur).

Noter également que l'authentification ne nécessite pas d'autorité de certification. La majorité du travail est effectuée en local par le service d'authentification de certificats, à l'exception de l'extraction des données enregistrées sur le serveur LDAP.

Implémentation du serveur

Le côté serveur du service d'authentification de certificats implémente une autorité de certification écrite en Java, et contient une autorité d'inscription (Registration Authority, RA) ainsi que des fonctionnalités d'auto–audit. Il publie des certificats et des CRL qu'il crée sur un serveur LDAP. Vous pouvez configurer cette autorité de certification via un ensemble de fichiers de configuration (fichiers de propriété Java). Il contient l'application administrative **runpki**, dont les sous-commandes permettent entre autres de démarrer et d'arrêter le serveur, et il prend également en charge le CMP pour créer et révoquer des certificats. L'autorité de certification a besoin de Java 1.3.1, de DB2 7.1 et de Directory 4.1. En raison des exigences de DB2, l'autorité de certification doit fonctionner sous un compte utilisateur différent de l'utilisateur root.

Les commandes serveur suivantes permettent d'installer et de gérer le composant **cas.server** :

mksecpki — Utilisée lors de l'installation pour configurer les composants du serveur PKI AIX. Entre autres tâches, elle crée un compte utilisateur pour l'autorité de certification.

runpki — Permet à l'administrateur système de démarrer le serveur. Si les démons JavaPKI sont en cours de fonctionnement, ils doivent tout d'abord être arrêtés. La commande **runpki** lance le démon en arrière-plan à l'aide de la combinaison des indicateurs **lb**. Si les démons doivent être démarrés en mode interactif, l'administrateur peut modifier la commande **runpki** et utiliser l'indicateur **l** au lieu des indicateurs **lb**.

runpki doit être lancée après l'exécution d'une opération **su –** sur le compte utilisateur sous lequel l'autorité de certification est en cours de fonctionnement. La commande est située dans le répertoire **javapki** sous le répertoire principal du compte utilisateur de l'autorité de certification. (La commande **mksecpki** crée le compte utilisateur de l'autorité de certification.)

Par exemple, si le compte utilisateur de l'autorité de certification est **pkiinst**, entrez les commandes suivantes avec les droits root :

1. `su – pkiinst`
2. `cd javapki`
3. `runpki`

Implémentation du client

Le côté client du service d'authentification de certificats implémente les fonctions d'authentification, d'administration et de gestion des certificats de l'utilisateur. Une fois installé et configuré sur un système, le service d'authentification de certificats s'intègre aux fonctions existantes d'administration et d'authentification de l'utilisateur (comme les commandes **mkuser**, **chuser**, **passwd** et **login**) à l'aide du LAMF AIX (Loadable Authentication Module Framework). Il apporte également plusieurs commandes, bibliothèques et fichiers de configuration pour aider à gérer les magasins de clefs et les certificats de l'utilisateur.

Le service d'authentification de certificats peut être utilisé en association avec la base de données LDAP AIX ou bien la base de données composée de fichiers pour enregistrer des attributs AIX standard. Le service d'authentification de certificats utilise toujours un serveur LDAP pour conserver les certificats utilisateur, même en cas d'utilisation d'une base de données composée de fichiers. Pour connaître les limitations d'utilisation d'une base de données composée de fichiers, consultez Planification du service d'authentification de certificats, page 7-15.

Le côté client du service d'authentification de certificats contient le logiciel le plus orienté utilisateur. Pour cette raison, les sections suivantes décrivent comment le service d'authentification de certificats conserve et utilise les données nécessaires à l'authentification PKI.

Fonctionnalités générales du client

La liste suivante décrit certaines des fonctionnalités générales du service d'authentification de certificats :

- Authentification utilisateur via les certificats PKI
- Commandes permettant de gérer les magasins de clefs et les certificats de l'utilisateur
- Prise en charge de plusieurs certificats par utilisateur
- Prise en charge simultané de plusieurs autorités de certification
- Intégration aux commandes existantes d'administration et authentification AIX (par exemple, **login**, **passwd**, **mkuser**)
- Génération de certificats au moment de la création de l'utilisateur ou ajout de certificats après la création de l'utilisateur
- Travail avec une base de données utilisateur LDAP ou la base de données utilisateur sur fichiers standard AIX
- Algorithmes et tailles de clef configurables
- Association de certificats avec les PAG (Process Authentication Groups).

Architecture générale du client

L'architecture client du service d'authentification de certificats utilise une approche en couches et comprend les composants suivants :

Démon Java

Le côté client s'appuie sur un démon Java utilisant le logiciels de sécurité JCE. Ce démon gère les magasins de clefs utilisateur, crée des paires de clef, effectue les communications CMP et propose toutes les fonctions de hachage et de chiffrement. Les API des modules PKI de prestataires de service n'étant pas standardisées pour les applications C, une API appelée Couche de gestion de service (Service Management Layer, SML) propose aux démons et programmes une interface normalisée.

Couche de gestion de service (SML)

Le service SML du démon Java s'appelle **/usr/lib/security/pki/JSML.sml**. SML assure la création de certificats, ainsi que la création et la gestion de magasins de clefs, mais ne gère pas le stockage des certificats, lequel est assuré par le couche LDAP PKI.

Stockage de clef privée via SML

Le démon Java utilise les fichiers du magasin de clefs au format PKCS#12 pour stocker les clefs des utilisateurs. Ces magasins sont protégés par un seul mot de passe, utilisé pour chiffrer toutes les clefs du magasin. L'emplacement d'un magasin de clefs est indiqué comme une URI. Par défaut, le service d'authentification de certificats conserve les fichiers du magasin de clefs dans le répertoire **/var/pki/security/keys**.

Les magasins de clefs et leurs fichiers sont généralement limités en taille. La Couche SML fournit l'API permettant de gérer les magasins de clefs.

Le service d'authentification de certificats ne gère que les magasins de clefs sur fichiers. Il n'accepte ni les smart cards ni les magasins de clefs LDAP. Vous pouvez accepter des visiteurs en plaçant les magasins de clefs sur fichiers dans un système de fichiers partagé, sur le même point de montage pour tous les systèmes.

Couche LDAP PKI (stockage de certificats)

Le service d'authentification de certificats stocke les certificats et autres informations relatives dans LDAP via la Couche LDAP PKI, pour chaque utilisateur. Ce service conserve les associations de certificats sur un serveur LDAP, pour chaque utilisateur. Plusieurs certificats peuvent être associés à un même compte utilisateur. Chaque association possède un indicateur unique spécifié par l'utilisateur pour faciliter l'identification et la recherche. Le service d'authentification de certificats utilise la combinaison du nom de l'utilisateur et de l'indicateur pour repérer l'association de certificats d'un utilisateur dans LDAP.

Pour optimiser les performances et l'espace disque, le service d'authentification de certificats peut sauvegarder la totalité du certificat sous LDAP ou simplement une référence URI de ce certificat. Si vous utilisez une référence URI au lieu d'un certificat, le service d'authentification de certificats demande à la référence de fournir le certificat concerné. Les références sont le plus souvent utilisées en association avec une autorité de certification qui publie ses certificats sur un serveur LDAP. Les types de référence URI couramment acceptées par le service d'authentification de certificats sont des références LDAP. Le service d'authentification de certificats stocke les certificats au format DER et demande aux références URI de se reporter aux certificats au format DER.

Le service d'authentification conserve également le type et l'emplacement du magasin de clefs et libellé de clef de chaque certificat, dans le même enregistrement que celui de l'association de certificats sur le serveur LDAP. Les utilisateurs peuvent ainsi disposer de plusieurs magasins de clefs, et le service d'authentification de certificats peut détecter plus rapidement la clef privée associée à un certificat. Pour gérer des visiteurs, il faut qu'un magasin de clefs se trouve au même endroit sur tous les systèmes.

Le service d'authentification de certificats gère l'attribut **auth_cert** dans LDAP pour chaque utilisateur. Cet attribut spécifie l'indicateur du certificat utilisé pour l'authentification.

Toutes les informations LDAP peuvent être consultées par les utilisateurs, à l'exception de l'attribut **auth_cert** qui est limité au compte LDAP **ldappkiadmin**. Puisque l'utilisateur root a accès au mot de passe LDAP **ldappkiadmin** via le fichier **acct.cfg**, les applications fonctionnant avec l'UID de root peuvent accéder à l'attribut **auth_cert**. (ceci s'applique à l'accès à l'URI de référence, pas aux données référencées par la valeur de cette URI. En général, ces données sont publiques.) L'API assurant le gestion du stockage des certificats figure dans la bibliothèque **libpki.a**.

La bibliothèque libpki.a

Outre sa fonction de centralisation des API SML et de Couche LDAP PKI, le bibliothèque **libpki.a** héberge plusieurs sous-routines. Les API contenues dans la bibliothèque ont les actions suivantes :

- Gestion des nouveaux fichiers de configuration
- Accès aux attributs spécifiques des certificats
- Association de plusieurs fonctions de couche basse en fonctions de niveau supérieur
- Sont communes aux services SML

Remarque : Les API ne sont pas publiées.

Couche LAMF (Loadable Authentication Module Framework)

Outre les API SML et LDAP PKI, la couche LAMF (Loadable Authentication Module Framework) figure dans la bibliothèque. LAMF fournit des applications d'administration utilisateur et d'authentification AIX avec les API correspondantes, indépendamment du mécanisme sous-jacent (par exemple, Kerberos, LDAP, DCE, fichiers). LAMF utilise les API SML et LDAP PKI pour implémenter l'authentification PKI.

Pour ce faire, il utilise des modules de chargement qui mappent l'API du LAMF vers différentes technologies d'authentification/base de données. Les commandes telles que **login**, **telnet**, **passwd**, **mkuser** et bien d'autres utilisent l'API du LAMF pour implémenter leurs fonctions ; par conséquent, elles acceptent automatiquement de nouvelles technologies d'authentification et de base de données dès l'ajout des nouveaux modules de chargement correspondants.

Le service d'authentification de certificats ajoute au système un nouveau module de chargement LAMF appelé **/usr/lib/security/PKI**. Ce module doit être ajouté par l'administrateur système dans le fichier **/usr/lib/security/methods.cfg**, avant d'utiliser PKI pour l'authentification. Il doit également être associé à un type de base de données (par exemple, LDAP) dans le fichier **methods.cfg** avant de pouvoir être utilisé pour l'authentification. Vous trouverez un exemple du fichier **methods.cfg** contenant le module LAMF et la définition de la base de données à la section Le fichier methods.cfg, page 7-30.

Une fois les définitions ajoutées à **methods.cfg**, l'administrateur peut configurer les attributs utilisateur **registry** et **SYSTEM** (définis dans le fichier **/etc/security/user**) sur la ou les nouvelles valeurs de strophe pour l'authentification PKI.

Commandes du client

Les commandes sont situées au–dessus de toutes les couches d'API (LAMF, LDAP PKI et SML). Outre les commandes AIX standard d'authentification et d'administration utilisateur pour le service d'authentification de certificats (via LAMF), il existe plusieurs commandes spécifiques à ce service. Ces commandes aident l'utilisateur à gérer les certificats et les magasins de clefs. Vous trouverez ci-dessous une liste de ces commandes accompagnée d'une brève description.

certadd	Ajoute un certificat au compte utilisateur dans LDAP et vérifie si le certificat est révoqué.
certcreate	Crée un certificat.
certdelete	Supprime un certificat du compte de l'utilisateur (c–à–d, de LDAP).
certget	Extrait un certificat du compte de l'utilisateur (c–à–d, de LDAP).
certlink	Ajoute dans LDAP un lien vers un certificat existant dans un référentiel distant du compte de l'utilisateur, et vérifie si le certificat est révoqué.
certlist	Affiche la liste des certificats associés au compte de l'utilisateur figurant dans LDAP.
certrevoke	Révoque un certificat.
certverify	Vérifie que la clef privée correspond au certificat et effectue une signature sécurisée.
keyadd	Ajoute un objet à un magasin de clefs.
keydelete	Supprime un objet d'un magasin de clefs.
keylist	Affiche la liste des objets d'un magasin de clefs.
keypasswd	Modifie le mot de passe d'un magasin de clefs.

Pour de plus amples informations sur ces commandes, consultez le manuel *AIX 5L Version 5.3 Commands Reference*.

Commandes PAG (Process Authentication Group)

Les commandes PAG (Process Authentication Group) sont nouvelles dans AIX. Les commandes PAG sont des éléments de données qui associent des données d'authentification utilisateur à des processus. Pour le service d'authentification de certificats, si le mécanisme PAG est activé, le certificat d'authentification de l'utilisateur est associé à son shell de connexion. Le PAG se propage à chaque processus fils créé par le shell.

Pour pouvoir proposer cette fonctionnalité, le PAG nécessite l'activation du démon **/usr/sbin/certdaemon**. Par défaut, ce mécanisme n'est pas activé. Son activation n'est pas nécessaire au service d'authentification de certificats, mais ce dernier l'utilise s'il est activé.

Pour activer le démon **certdaemon**, ajoutez la ligne suivante au fichier **/etc/inittab** :

```
certdaemon:2:wait:/usr/sbin/certdaemon
```

Vous trouverez ci-dessous une liste des commandes PAG accompagnée d'une brève description :

paginit — Authentifie un utilisateur et crée une association PAG.

paglist — Affiche une liste des informations d'authentification associées au processus courant.

pagdel — Supprime les associations PAG existantes dans les références du processus courant.

Pour de plus amples informations sur ces commandes, consultez le manuel *AIX 5L Version 5.3 Commands Reference*.

Commandes d'administration utilisateur

Comme pour l'authentification utilisateur, le service d'authentification de certificats s'intègre aux fonctions AIX d'administration utilisateur via le LAMF AIX. Les commandes telles que **chuser**, **lsuser**, **mkuser** et **passwd** utilisent l'API du LAMF. Par conséquent, elles acceptent automatiquement de nouvelles technologies d'authentification et de base de données dès l'ajout au système de nouveaux modules de chargement.

Les sous-sections suivantes décrivent plus en détail la façon dont l'authentification PKI affecte les commandes d'administration utilisateur.

Les commandes suivantes sont concernées par le processus d'authentification PKI :

chuser — Permet à l'administrateur de modifier l'attribut utilisateur **auth_cert**. Cet attribut spécifie la valeur de l'indicateur du certificat utilisé pour l'authentification. Pour pouvoir utiliser le certificat comme certificat d'authentification, vous devez le signer à l'aide de la clef de signature sécurisée. (Les attributs de certificat, les attributs de stockage de certificat et les attributs de magasin de clefs ne sont pas accessibles via cette commande.)

lsuser — Affiche la valeur de l'attribut **auth_cert** de l'utilisateur, ainsi que les attributs de certificats répertoriés ci-dessous. L'attribut **auth_cert** indique la valeur de l'indicateur du certificat utilisé pour l'authentification. (Les autres attributs de certificat, attributs de stockage de certificat et attributs de magasin de clefs ne sont pas accessibles via cette commande.)

Les attributs de certificat répertoriés par la commande **lsuser** sont les suivants :

subject–DN — Le nom spécifique de l'utilisateur.

subject–alt–name — L'e–mail du nom supplémentaire de l'utilisateur.

valid–after — La date à laquelle le certificat de l'utilisateur devient valide.

valid–until — La date à laquelle le certificat de l'utilisateur n'est plus valide.

issuer — Le nom spécifique de l'émetteur.

mkuser Génère un certificat au moment de la création de l'utilisateur. Un administrateur peut utiliser **mkuser** pour générer un certificat lors de la création d'utilisateurs ne possédant pas encore de certificat d'authentification. Eventuellement, si un utilisateur possède déjà un certificat d'authentification, mais pas de compte utilisateur, l'administrateur peut créer le compte sans générer de certificat et l'ajouter (ainsi que le magasin de clefs) ultérieurement. La valeur par défaut de cette option est indiquée par l'attribut **cert** de la strophe newuser du fichier **/usr/lib/security/pki/policy.cfg**.

Plusieurs valeurs par défaut sont nécessaires lors de la génération automatique d'un certificat d'authentification pour un utilisateur à l'aide la commande **mkuser**. La plupart sont indiquées dans la strophe **newuser** du fichier **/usr/lib/security/pki/policy.cfg**. La strophe **newuser** assure un contrôle administratif de ces valeurs par défaut. Voici quelques-unes de ces valeurs par défaut :

- CA
- Valeur de l'attribut **auth_cert**
- Emplacement du magasin de clefs
- Mot de passe du magasin de clefs
- Libellé de clef privée
- Nom de domaine du champ E–mail du nom d'utilisateur supplémentaire

Il existe une différence entre créer un compte utilisateur PKI ou non–PKI. Pour créer un compte utilisateur PKI, il vous faut un mot de passe pour chiffrer la clef privée, si la commande **mkuser** génère un certificat d'authentification pour ce compte. Mais la commande **mkuser** n'étant pas interactive, elle obtient donc le mot de passe du fichier **policy.cfg** et l'utilise comme mot de passe du magasin de clefs (le mot de passe de la clef privée) ; vous pouvez donc accéder au compte immédiatement après sa création. Lorsque vous créez un compte utilisateur non–PKI, la commande **mkuser** configure le mot de passe sur une valeur non valide, ce qui empêche l'accès.

passwd Cette commande modifie le mot de passe du magasin de clefs de l'utilisateur lorsqu'il est utilisé sur un compte utilisateur PKI. Elle oblige à utiliser les règles de restriction du mot de passe indiquées dans le fichier **/etc/security/user**, l'attribut des indicateurs figurant dans le fichier **/etc/security/passwd** ainsi que toutes les règles requises par le prestataire de service PKI.

Les magasins de clefs sur fichiers chiffrant leurs clefs privées à l'aide du mot de passe de l'utilisateur, l'utilisateur **root** ne peut pas réinitialiser le mot de passe d'un magasin de clefs sur fichiers s'il ne connaît pas le mot de passe courant du magasin. En cas d'oubli du mot de passe par un utilisateur, l'utilisateur root ne pourra pas le réinitialiser sauf si root le connaît. Si le mot de passe est inconnu, il faudra probablement créer un nouveau magasin de clefs et de nouveaux certificats pour l'utilisateur.

Fichiers de configuration

Le service d'authentification de certificats utilise des fichiers de configuration pour configurer le côté client : **acct.cfg**, **ca.cfg** et **policy.cfg**. L'interface SMIT gère ces fichiers de configuration. Vous trouverez dans les sections suivantes des informations sur les fichiers de configuration.

Le fichier acct.cfg

Le fichier **acct.cfg** contient des strophes CA et LDAP. Les strophes CA contiennent des informations confidentielles ne devant pas figurer dans le fichier **ca.cfg** accessible au public, comme par exemple, des mots de passe et des numéros de référence du CMP. Les strophes LDAP contiennent des informations LDAP confidentielles interdites au public, comme par exemple, des mots de passe et des noms administratifs LDAP PKI.

Pour chaque strophe CA du fichier **ca.cfg**, le fichier **acct.cfg** doit contenir une strophe CA de nom identique, chaque nom de strophe CA doit être unique. Les strophes LDAP sont toutes appelées **ldap**, ce qui explique pourquoi une strophe CA ne peut pas s'appeler **ldap**. De même, aucune strophe ne peut s'appeler **default**. Il doit y avoir une strophe LDAP, et également au moins une strophe CA appelée **local**.

Les strophes CA contiennent les attributs suivants :

capasswd	Indique le mot de passe CMP de CA. La longueur du mot de passe est définie par la CA.
carefnum	Indique le numéro de référence CMP de la CA.
keylabel	Indique le libellé de la clef privée dans le magasin de clefs sécurisé utilisé pour signer les demandes de certificats.
keypasswd	Indique le mot de passe du magasin de clefs sécurisé.
rvpasswd	Indique le mot de passe de révocation utilisé pour le CMP. La longueur du mot de passe est définie par la CA
rvrefnum	Indique le numéro de référence de la révocation utilisé pour le CMP.

La strophe LDAP contient les attributs suivants :

ldappkiadmin	Indique le nom de compte du serveur LDAP affiché dans la liste **ldapservers**.
ldappkiadmpwd	Indique le mot de passe du compte du serveur LDAP.
ldapservers	Indique le nom du serveur LDAP.
ldapsuffix	Indique les attributs DN ajoutés au certificat DN d'un utilisateur par la commande **mkuser**.

Voici un exemple de fichier **acct.cfg** :

```
local:
     carefnum = 12345678
     capasswd = password1234
     rvrefnum = 9478371
     rvpasswd = password4321
     keylabel = "Trusted Key"
     keypasswd = joshua

 ldap:
     ldappkiadmin = "cn=admin"
     ldappkiadmpwd = secret
     ldapservers = "ldap.server.austin.ibm.com"
     ldapsuffix = "ou=aix,cn=us"
```

Pour de plus amples informations, consultez le manuel *AIX 5L Version 5.3 Files Reference*.

Le fichier ca.cfg

Le fichier **ca.cfg** est constitué de strophes CA. Elles contiennent des informations CA publiques utilisées par le service d'authentification de certificats pour générer des demandes de certificats et des demandes de révocation de certificats.

Pour chaque strophe CA du fichier **ca.cfg**, le fichier **acct.cfg** doit contenir une strophe CA du même nom. Chaque nom de strophe CA figurant dans le fichier **ca.cfg** doit être unique. Il doit y avoir au moins une strophe appelée **local**. Il ne doit y avoir aucune strophe appelée **ldap** ou **default**.

Les strophes CA contiennent les attributs suivants :

algorithm	Indique l'algorithme de clef publique (par exemple, RSA).
crl	Indique l'URI de la CRL pour la CA.
dn	Indique le DN de base utilisé lors de la création des certificats.
keysize	Indique la taille de clef minimale en bits.
program	Indique le nom de fichier du module de service PKI.
retries	Indique le nombre de tentatives de contact avec la CA.
server	Indique l'URI de la CA.
signinghash	Indique l'algorithme de hachage utilisé pour signer les certificats (par exemple, MD5).
trustedkey	Indique le magasin de clefs sécurisé contenant la clef de signature sécurisée utilisée pour signer les certificats d'authentification.
url	Indique la valeur par défaut de l'URI du nom d'utilisateur supplémentaire.

La strophe CA par défaut est appelée local. Voici un exemple de fichier **ca.cfg** :

```
local:
     program = /usr/lib/security/pki/JSML.sml
     trustedkey = file:/usr/lib/security/pki/trusted.p15
     server = "cmp://9.53.230.186:1077"
     crl = "ldap://dracula.austin.ibm.com/o=aix,c=us"
     dn = "o=aix,c=us"
     url = "http://www.ibm.com/"
     algorithm = RSA
     keysize = 512
     retries = 5
     signinghash = MD5
```

Pour de plus amples informations, consultez le manuel *AIX 5L Version 5.3 Files Reference*.

Le fichier policy.cfg

Le fichier **policy.cfg** est composé de quatre strophes : **newuser**, **storage**, **crl** et **comm**. Ces strophes modifient le comportement de certaines commandes d'administration du système. La commande **mkuser** utilise la strophe **newuser**. La commande **certlink** utilise la strophe **storage**. Les commandes **certadd** et **certlink** utilisent les strophes **comm** et **crl**.

La strophe **newuser** contient les attributs suivants :

ca	Indique la CA utilisée par la commande **mkuser** lors de la génération d'un certificat.
cert	Indique si la commande **mkuser** va par défaut générer un nouveau (new) certificat ou non (get).
domain	Indique la partie domaine de la valeur E–mail du nom d'utilisateur supplémentaire du certificat utilisée par la commande **mkuser** lors de la génération d'un certificat.
keysize	Indique la taille minimale en bits de la clef de chiffrement utilisée par la commande **mkuser** lors de la génération d'un certificat.
keystore	Indique l'URI du magasin de clefs utilisée par la commande **mkuser** lors de la génération d'un certificat.
keyusage	Indique la valeur d'usage de la clef du certificat utilisée par la commande **mkuser** lors de la génération d'un certificat.
label	Indique le libellé de la clef privée utilisée par la commande **mkuser** lors de la génération d'un certificat.

passwd — Indique le mot de passe du magasin de clefs utilisé par la commande **mkuser** lors de la génération d'un certificat.

subalturi — Indique la valeur de l'URI du nom d'utilisateur supplémentaire utilisée par la commande **mkuser** lors de la génération d'un certificat.

tag — Indique la valeur de l'indicateur **auth_cert** utilisée par la commande **mkuser** lors de la création d'un utilisateur si cert = new.

validity — Indique la valeur de la période de validité du certificat utilisée par la commande **mkuser** lors de la génération d'un certificat.

version — Indique le numéro de version du certificat à créer. La valeur 3 est la seule valeur prise en charge.

La strophe **storage** contient les attributs suivants :

replicate — Indique si la commande **certlink** sauvegarde une copie du certificat (**yes**) ou simplement le lien (**no**).

La strophe **crl** contient l'attribut **check**, qui indique si les commandes **certadd** et **certlink** doivent vérifier la CRL (**yes**) ou non (**no**).

La strophe **comm** contient l'attribut **timeout** qui indique le délai d'attente en secondes utilisé par **certadd** et **certlink** lors d'une demande d'informations sur un certificat à l'aide du protocole HTTP (par exemple, extraction de CRL).

Voici un exemple de fichier **policy.cfg** :

```
newuser:
     cert = new
     ca = local
     passwd = pki
     version = "3"
     keysize = 512
     keystore = "file:/var/pki/security/keys"
     validity = 86400

 storage:
     replicate = no

 crl:
     check = yes

 comm:
     timeout = 10
```

Pour de plus amples informations, consultez le manuel *AIX 5L Version 5.3 Files Reference*.

Evénements du journal d'audit

Le client du service d'authentification de certificats génère les événements du journal d'audit suivants :

- CERT_Create
- CERT_Add
- CERT_Link
- CERT_Delete
- CERT_Get
- CERT_List
- CERT_Revoke
- CERT_Verify
- KEY_Password
- KEY_List

- KEY_Add
- KEY_Delete

Evénements de trace

Le client du service d'authentification de certificats génère plusieurs nouveaux événements de trace, dans la plage 3B7 – 3B8.

Planification du service d'authentification de certificats

Le Service d'authentification des certificats est disponible à partir d'AIX 5.2. Les configurations logicielles minimales requises sont un serveur DB2, un serveur Directory et un serveur de service d'authentification de certificats. Vous pouvez les installer sur un seul système ou sur plusieurs. Chaque entreprise doit choisir l'environnement qui lui convient le mieux.

Cette section présente des informations sur la planification du service d'authentification de certificats, notamment :

Remarques sur les certificats

Le service d'authentification de certificats gère les certificats X.509 version 3. Il accepte également plusieurs attributs de certificat de la version 3, mais pas tous. Pour connaître la liste des attributs de certificats pris en charge, reportez-vous à la commande **certcreate** et au fichier **ca.cfg**. Le service d'authentification de certificats gère une partie de l'ensemble de caractères Teletex. Il n'accepte que le Teletex 7 bits (sous-ensemble ASCII).

Remarques sur les magasins de clefs

Le service d'authentification des certificats gère les fichiers de magasin de clefs.
Les types de magasins de clefs smart cards, LDAP et autres ne sont pas pris en charge.

Par défaut, les magasins de clefs utilisateur sont conservés dans le système de fichiers local sous le répertoire **/var/pki/security/keys**. Les magasins de clefs étant en local sur le système, ils ne sont pas accessibles aux autres systèmes ; par conséquent, l'authentification utilisateur sera limitée au système sur lequel figure le magasin de clefs de l'utilisateur. Pour gérer des visiteurs, vous pouvez copier le magasin de clefs de l'utilisateur vers le même emplacement local et avec le même nom de magasin de clefs sur les autres systèmes, ou bien placer les magasins de clefs sur un système de fichiers distribué.

Remarque : Vous devez faire attention à ce que les droits d'accès au magasin de clefs de l'utilisateur ne soient pas modifiés. (Dans AIX, chaque certificat sur LDAP contient le chemin d'accès vers le magasin de clefs privé contenant la clef privée du certificat. Le magasin de clefs doit figurer à l'endroit du chemin d'accès indiqué dans LDAP pour pouvoir servir à l'authentification.)

Remarques sur le registre des utilisateurs

Le service d'authentification des certificats utilise un registre LDAP des utilisateurs. LDAP est également le type de registre utilisateur conseillé pour une utilisation avec le service d'authentification de certificats.

Le service d'authentification des certificats gère également un registre des utilisateurs sur fichier. L'administrateur doit imposer certaines restrictions pour que la PKI sur fichiers puisse fonctionner. Les comptes utilisateur possédant le même nom sur différents systèmes participant à l'authentification PKI doivent faire référence au même compte.

Par exemple, l'utilisateur *Bob* sur le *système A* et l'utilisateur *Bob* sur le *système B* doivent faire référence au même utilisateur *Bob*. Cela s'explique du fait que le service d'authentification de certificats utilise LDAP pour stocker les informations sur les certificats pour chaque utilisateur. Le nom d'utilisateur est utilisé comme clef d'indexation pour accéder à ces informations. Les registres sur fichier étant en local sur chaque système et LDAP commun à tous les systèmes, les noms d'utilisateur sur tous les systèmes participant à l'authentification PKI doivent correspondre aux noms d'utilisateur uniques dans l'espace de nom LDAP. Si l'utilisateur *Bob* du *système A* est différent de l'utilisateur *Bob* du *système B*, soit un seul des utilisateurs *Bob* peut participer à l'authentification PKI, ou bien chaque compte *Bob* doit utiliser un serveur/espace de nom LDAP différent.

Remarques sur la configuration

Pour simplifier la configuration, vous pourrez conserver les trois fichiers de configuration (**acct.cfg**, **ca.cfg** et **policy.cfg**) sur un système de fichiers distribué à l'aide de liens symboliques, pour ne pas avoir à les modifier sur chaque système. Conservez les paramètres de contrôle d'accès appropriés sur ces fichiers. C'est une situation qui peut diminuer votre sécurité, car les informations contenues par ces fichiers vont circuler sur le réseau.

Remarques sur la sécurité

Le fichier acct.cfg

Le fichier **acct.cfg** contient les numéros de référence et les mots de passe de chaque CA (consultez les descriptions des attributs **carefnum**, **capasswd**, **rvrefnum** et **rvpasswd** pour **acct.cfg**). Ces valeurs sont utilisées uniquement pour les communications CMP avec la CA lors de la création ou de la révocation d'un certificat. En cas de sécurité compromise, l'individu à l'origine de cette situation pourrait créer ou révoquer des certificats comme bon lui semble.

Pour limiter ce risque, vous ne devez appliquer la restriction de création ou de révocation de certificats qu'à un petit nombre de systèmes. Les valeurs des attributs **carefnum** et **capasswd** ne sont requises que sur les systèmes sur lesquels les certificats sont créés (via les commandes **certcreate** ou **mkuser**). Ce qui peut imposer de limiter de la création de compte utilisateur au même ensemble de systèmes.

Remarque : La commande **mkuser** peut être configurée pour créer automatiquement un certificat lors de la création d'un utilisateur ou bien créer un compte sans lui associer de certificat, auquel cas l'administrateur devra créer et ajouter ce certificat ultérieurement. De même, les valeurs des attributs **rvrefnum** et **rvpasswd** ne sont requises que sur les systèmes sur lesquels les certificats doivent être révoqués (via la commande **certrevoke**).

Le fichier **acct.cfg** contient également des informations sur chacune des clefs de signature sécurisée (consultez les descriptions des attributs **keylabel** et **keypasswd** pour le fichier **acct.cfg**). Ces valeurs sont utilisées uniquement dans le cadre d'opérations spéciales de vérification des certificats. En cas de sécurité compromise, l'individu à l'origine de cette situation pourrait créer des certificats vérifiés.

Pour limiter ce risque, vous ne devez accorder la vérification de certificats qu'à un petit nombre de systèmes. Les valeurs des attributs **keylabel** et **keypasswd** du fichier **acct.cfg** ainsi que la valeur de l'attribut **trustedkey** du fichier **ca.cfg** ne sont requises que sur les systèmes sur lesquels la vérification de certificats est nécessaire. Autrement dit, sur les systèmes sur lesquels les commandes **mkuser** (avec la fonction de création automatique de certificats activée) et **certverify** sont nécessaires.

Nouveaux comptes actifs

Lorsque vous créez un compte utilisateur PKI, si l'attribut **cert** de la strophe **newuser** contenue dans le fichier **policy.cfg** est défini sur **new**, la commande **mkuser** crée un compte PKI actif ainsi qu'un mot de passe et un certificat. Le mot de passe de ce compte est indiqué par l'attribut **passwd** dans la strophe **newuser**. Les magasins de clefs exigent en effet un mot de passe pour stocker les clefs privées. Cette méthode est différente des autres types de création de compte utilisateur pour lesquels l'administrateur doit d'abord créer le compte, puis définir le mot de passe avant de pouvoir activer le compte.

L'utilisateur root et les mots de passe des magasins de clefs

Contrairement à d'autres types de compte pour lesquels l'utilisateur **root** peut modifier le mot de passe d'un compte sans le connaître, les comptes PKI ne le permettent pas. En effet, les mots de passe des comptes servent à chiffrer les magasins de clefs et ils sont nécessaires pour pouvoir les déchiffrer. Si les utilisateurs oublient leurs mots de passe, vous devez émettre de nouveaux certificats et créer de nouveaux magasins de clefs.

Autres remarques sur le service d'authentification de certificats

Voici quelques remarques à prendre en compte lors de la planification du service d'authentification de certificats :

- Le service d'authentification de certificats contient sa propre autorité de certification (CA). Il ne prend pas en charge d'autres autorités de certification.
- Plus la taille de la clef est importante, plus il faut de temps pour générer des paires de clefs et chiffrer les données. Le chiffrement matériel n'est pas pris en charge.
- Le service d'authentification de certificats utilise Directory for LDAP. Il ne gère pas d'autres implémentations LDAP.
- Le service d'authentification de certificats utilise DB2 comme base de données. Il n'accepte pas d'autres bases de données.
- Le service d'authentification de certificats impose que toutes les commandes, bibliothèques et démons soient dans un environnement Unicode.

Modules du service d'authentification de certificats

Les composants du module de Service d'authentification de certificats sont :

Tableau 7. Modules du service d'authentification de certificats

Nom du module	Ensemble de fichiers	Contenu	Dépendances	Installation
cas.server	cas.server.rte	Autorité de certification (AC)	• AIX 5.2 • Java131 (livré avec les supports de base AIX) • Extensions de sécurité Java131 (livré avec Expansion Pack) • Serveur Directory (LDAP) • DB2 7.1	Manuelle
cas.client	cas.client.rte	• Commandes Cert • Module de chargement d'authentification PKI • libpki.a • Module SML • Fichiers de configuration • Démon Java	• AIX 5.2 • Java131 (livré avec les supports de base AIX) • Extensions de sécurité Java131 (livré avec Expansion Pack) • Client Directory (LDAP) • PAG (implicite)	Manuelle
cas.msg	cas.msg.[lang].client	Catalogues de messages	cas.client	Manuelle
bos	bos.security.rte	Démon et commandes PAG	Non applicable	Installé avec le noyau

Le module **cas.server** contient la CA et s'installe dans les répertoires **/usr/cas/server** et **/usr/cas/client**. Une entreprise n'utilise généralement qu'une seule CA, et par ailleurs, ce module est installé manuellement. Ce module exige l'installation préalable du serveur Directory, **db2_07_01.client**, **Java131.rte** et **Java131.ext.security**. Le **module Java131.rte** est installé par défaut lorsque le système d'exploitation AIX 5.2 est installé, mais les autres modules sont installés manuellement.

Pour que le module **db2_07_01.client** fonctionne, le module **db2_07_01.server** doit être installé sur un système du réseau.

Le module **cas.client** contient les fichiers requis pour tous les systèmes clients prenant en charge le service d'authentification de certificats. Sans ce module, un système ne peut pas participer à l'authentification PKI AIX.

Installation et configuration du service d'authentification de certificats

L'installation du service d'authentification de certificats consiste à exécuter les procédures suivantes :

- Installation et configuration du serveur LDAP, page 7-19
- Installation et configuration du serveur pour le service d'authentification de certificats, page 7-22
- Configuration LDAP du serveur pour le service d'authentification de certificats, page 7-23
- Configuration du client pour le service d'authentification de certificats, page 7-25
- Exemples de configuration de l'administration, page 7-30

Installation et configuration du serveur LDAP

Les scénarios possible pour l'installation et la configuration de LDAP pour les données des certificats de l'utilisateur PKI sont les suivants.

- Si le serveur LDAP n'est pas installé, exécutez les procédures suivantes :
  1. Installation du serveur LDAP, page 7-19
  2. Configuration du serveur LDAP, page 7-20
  3. Configuration du serveur LDAP pour PKI, page 7-21
- Si le serveur LDAP est installé et configuré, mais pas pour PKI, effectuez la Configuration du serveur LDAP pour PKI, page 7-21.

Installation du serveur LDAP

Vous trouverez des instructions détaillées relatives à l'installation du serveur Directory dans la documentation du produit contenue dans l'ensemble de fichiers **ldap.html.en_US.config**. Une fois l'ensemble de fichiers **ldap.html.en_US.config** installé, vous pouvez afficher la documentation à l'aide d'un navigateur Web à l'URL suivante : **file:/usr/ldap/web/C/getting_started.htm**.

La procédure d'installation du serveur LDAP est la suivante :

1. Connectez-vous en tant qu'utilisateur **root**.
2. Insérez le volume 1 des CD du système d'exploitation de base d'AIX dans le lecteur de CD–ROM.
3. Entrez **smitty install_latest** sur la ligne de commandes et appuyez sur Entrée
4. Sélectionnez **Installer logiciels**
5. Sélectionnez l'unité ou le répertoire contenant le logiciel serveur Directory puis appuyez sur Entrée.
6. Utilisez la touche **F4** pour afficher la liste des modules dans la zone **Logiciels à installer**.
7. Sélectionnez le module LDAP server puis appuyez sur Entrée.
8. Vérifiez que l'option **Installation AUTOMATIQUE des logiciels dépendants** est définie sur **YES**, puis appuyez sur Entrée. Cette option installera les ensembles de fichiers du client et du serveur LDAP ainsi que les ensembles de fichiers de la base de données DB2.

Les ensembles de fichiers installés sont les suivants :

- LDAP client.**adt** (SDK du client Directory)
- LDAP client.**dmt** (DMT du client Directory)
- LDAP client.**java** (Java du client Directory)
- LDAP client.**rte** (Environnement d'exécution du client Directory)
- LDAP server.**rte** (Environnement d'exécution du serveur Directory)
- LDAP server.**admin** (Serveur Directory)
- LDAP server.**cfg** (Configuration du serveur Directory)
- LDAP server.**com** (Cadre du serveur Directory)
- **db2_07_01.*** (Environnement d'exécution DB2 et ensembles de fichiers associés)

Le module DB2, **db2_07_01.jdbc**, doit également être installé. Le module DB2 **db2_07_01.jdbc** se trouve sur le CD Expansion Pack. Utilisez la procédure d'installation affichée dans la liste ci-dessus pour installer le module **db2_07_01.jdbc**.

Configuration du serveur LDAP

Une fois les ensembles de fichiers LDAP et DB2 installés, vous devez configurer le serveur LDAP. Même si vous pouvez effectuer la configuration via la ligne de commandes et l'édition de fichiers, il est conseillé d'utiliser l'administrateur web LDAP. Cet outil nécessite un serveur web.

Le serveur web Apache se trouve sur le CD AIX Toolbox for LINUX Applications. Utilisez l'interface SMIT ou la commande **geninstall** pour installer le serveur web Apache. Vous pouvez également utiliser d'autres serveurs web, consultez la documentation LDAP pour plus de détails.

Vous trouverez des instructions détaillées relatives à la configuration LDAP dans la documentation HTML du produit. Voici une description succincte des étapes de configuration :

1. Utilisez **ldapcfg** pour définir le mot de passe et le DN de l'administrateur pour la base de données LDAP. L'administrateur est l'utilisateur **root** de la base de données LDAP. Pour configurer un DN administrateur de **cn=admin** avec le mot de passe **secret**, entrez :

```
# ldapcfg -u cn=admin -p secret
```

   Le mot de passe et le DN vous seront demandés ultérieurement lors de la configuration de chaque client. Le DN et le mot de passe seront utilisés comme attributs **ldappkiadmin** et **ldappkiadmpwd** pour une strophe **ldap** dans le ficher **acct.cfg**.

2. Configurez l'outil de l'administrateur web en utilisant l'emplacement du fichier de configuration du serveur web :

```
# ldapcfg -s apache -f /etc/apache/httpd.conf
```

3. Redémarrez le serveur web. Pour le serveur Apache, utilisez la commande :

```
# /usr/local/bin/apachectl restart
```

4. Accédez à l'administrateur web à l'aide de l'URL **http:// hostname/ldap**. Connectez-vous ensuite à l'aide du mot de passe et du DN de l'administrateur LDAP configurés à l'étape 2.
5. A l'aide de l'outil de l'administrateur web, suivez les directives pour configurer la base de données DB2 et redémarrez le serveur LDAP.

Configuration du serveur LDAP pour PKI

Le service d'authentification de certificats exige deux arborescences distinctes du répertoire LDAP. L'une est utilisée par l'autorité de certification pour publier des certificats et des CRL. L'autre est utilisée par les clients pour stocker et extraire des données PKI pour chaque utilisateur. Les étapes suivantes configurent l'arborescence du répertoire LDAP utilisée pour le stockage et l'extraction des données PKI de chaque utilisateur.

1. **Ajout d'une entrée de suffixe pour la configuration LDAP**. Le suffixe par défaut des données PKI est **cn=aixdata**. Il place les données du certificat PKI en-dessous du suffixe par défaut de toutes les données AIX. Le répertoire root par défaut pour les données PKI est **ou=pkidata,cn=aixdata**. Toutes les données PKI sont placées à cet endroit.

   – **Suffixe des données PKI**

cn=aixdata	Suffixe commun pour toutes les données AIX. Il existe peut–être déjà si le serveur LDAP est utilisé pour d'autres données AIX. Vous pouvez ajouter l'entrée de configuration du suffixe via l'outil de l'administrateur web, ou en éditant directement le fichier de configuration du serveur LDAP.

   Pour ajouter l'entrée de configuration du suffixe à l'aide de l'administrateur Web, procédez comme suit :

   a. Sélectionnez **Paramètres** dans le menu de gauche.

   b. Sélectionnez **Suffixes**.

   c. Entrez le suffixe nécessaire pour les données PKI, puis cliquez sur le bouton **Mettre à jour**.

   d. Une fois le suffixe ajouté, redémarrez le serveur LDAP.

   Pour ajouter l'entrée de configuration du suffixe en éditant le fichier de configuration du serveur LDAP, procédez comme suit :

   a. Dans le fichier **/usr/ldap/etc/slapd32.conf**, repérez la ligne contenant

   ```
   ibm-slapdSuffix: cn=localhost
    Il s'agit du suffixe système par défaut.
   ```

   b. Ajoutez l'entrée `ibm-slapdSuffix` nécessaire pour les données PKI. Par exemple, vous pouvez ajouter une entrée de suffixe comme celle qui suit :

   ```
   ibm-slapdSuffix: cn=aixdata
   ```

   c. Sauvegardez les modifications apportées au fichier de configuration.

   d. Redémarrez le serveur LDAP.

2. **Ajout des Entrées suffixe, répertoire Root et base de données ACL pour les données PKI**. Le répertoire root des données est le point dans la structure du répertoire LDAP sous lequel se trouvent toutes les données PKI. L'ACL est la Liste de contrôle d'accès du répertoire Root des données qui définit les règles d'accès pour toutes les données PKI. Le fichier **pkiconfig.ldif** fourni permet d'ajouter les entrées suffix, root et ACL à la base de données. Ajoutez tout d'abord les entrées suffixe et base de données root, ainsi que le mot de passe de l'administrateur des données PKI. La première partie du fichier ajoute les entrées suffixe par défaut à la base de données et définit le mot de passe :

```
dn: cn=aixdata
 objectclass: top
 objectclass: container
 cn: aixdata

 dn: ou=pkidata,cn=aixdata
 objectclass: organizationalUnit
 ou: cert
 userPassword: <<password>>
```

Modifiez le fichier **pkiconfig.ldif** et remplacez la chaîne <<password>> après l'attribut **userPassword** par le mot de passe pour votre administrateur de données PKI.

Les valeurs de DN et de **userPassword** seront requises ultérieurement lors de la configuration de chaque client. Le DN (`ou=pkidata,cn=aixdata`) et la valeur du *password* seront utilisés pour les attributs **ldappkiadmin** et **ldappkiadmpwd** dans une strophe **ldap** du fichier **acct.cfg**.

La seconde partie du fichier modifie le propriétaire et ajoute l'ACL des données PKI :

```
dn: ou=pkidata,cn=aixdata
 changetype: modify
 add: entryOwner
 entryOwner: access-id:ou=pkidata,cn=aixdata
 ownerPropagate: true

 dn: ou=pkidata,cn=aixdata
 changetype: modify
 add: aclEntry
 aclEntry: group:cn=anybody:normal:grant:rsc:normal:deny:w
 aclEntry: group:cn=anybody:sensitive:grant:rsc:sensitive:deny:w
 aclEntry: group:cn=anybody:critical:grant:rsc:critical:deny:w
 aclEntry: group:cn=anybody:object:deny:ad aclPropagate: true
```

Remarque : Pour ne pas endommager l'intégrité de votre implémentation PKI, n'effectuez *aucun* changement sur les réglages ACL.

Vous pouvez modifier le fichier **pkiconfig.ldif** pour utiliser un suffixe autre que celui par défaut, ce qui n'est toutefois conseillé qu'aux administrateurs LDAP expérimentés. Vous pouvez alors appliquer le fichier **ldif** à la base de données à l'aide de la commande **ldapadd** suivante. Remplacez les valeurs des options **–D** et **–w** par le mot de passe et le DN de votre administrateur LDAP local, comme suit :

```
# ldapadd -c -D cn=admin -w secret -f pkiconfig.ldif
```

3. **Redémarrage du serveur LDAP**. Redémarrez le serveur LDAP à l'aide de l'outil de l'administrateur web, ou en utilisant la commande kill et en redémarrant le processus **slapd**.

Installation et configuration du serveur pour le service d'authentification de certificats

Pour installer et configurer le service d'authentification de certificats, procédez comme suit :

1. Installez les ensembles de fichiers de sécurité Java (**Java131.ext.security.***) à partir du CD Expansion Pack. Les modules requis sont les suivants :
   - **Java131.ext.security.cmp–us** (Gestion des certificats Java)
   - **Java131.ext.security.jce–us** (Extension du chiffrement Java)
   - **Java131.ext.security.jsse–us** (Extension de socket Java sécurisée)
   - **Java131.ext.security.pkcs–us** (Chiffrement à clef publique Java)

2. Déplacez le fichier **ibmjcaprovider.jar** dans un autre répertoire que **/usr/java131/jre/lib/ext**. Ce fichier est incompatible avec les ensembles de fichier de sécurité Java et doit être déplacé pour que le service d'authentification de certificats puisse fonctionner correctement.
3. Installez l'ensemble de fichiers du serveur pour le service d'authentification de certificats (**cas.server.rte**) à partir du CD Expansion Pack.

Configuration LDAP du serveur pour le service d'authentification de certificats

Pour configurer le serveur du service d'authentification de certificats de sorte qu'il travaille avec LDAP, effectuez les étapes suivantes :

1. Si vous ne l'avez pas déjà installé, installez le module client Directory sur le système prenant en charge le module **cas.server**.
2. Si vous ne l'avez pas déjà configuré, configurez le client Directory client comme suit :

```
# ldapcfg -l /home/ldapdb2 -u "cn=admin" -p secret -s apache \
          -f /usr/local/apache/conf/httpd.conf
```

   La commande de configuration ci-dessus considère que le serveur Web est le serveur Apache.

3. Ajoutez le suffixe suivant au fichier **slapd.conf** :

```
ibm-slapdSuffix: o=aix,c=us
```

   Vous pouvez indiquer un nom spécifique différent à la place de `o=aix,c=us`.

4. Exécutez la commande **slapd** :

```
# /usr/bin/slapd -f /etc/slapd32.conf
```

5. Ajoutez les classes d'objet, comme suit :

```
# ldapmodify -D cn=admin -w secret -f setup.ldif
 où  setup.ldif contient les éléments suivants :

dn: cn=schema
changetype: modify
add: objectClasses
objectClasses: ( 2.5.6.21 NAME 'pkiuser' DESC 'auxiliary class for non-CA
certificate owners'
  SUP top AUXILIARY MAY userCertificate )

dn: cn=schema
changetype: modify
add: objectClasses
objectClasses: ( 2.5.6.22 NAME 'pkiCA' DESC 'class for Cartification
Authorities' SUP top
   AUXILIARY MAY ( authorityRevocationList $ caCertificate $
certificateRevocationList $
   crossCertificatePair ) )

dn:cn=schema
changetype: modify
replace: attributetypes
attributetypes: ( 2.5.4.39 NAME ( 'certificateRevocationList'
   'certificateRevocationList;binary' ) DESC '  ' SYNTAX
1.3.6.1.4.1.1466.115.121.1.5
   SINGLE-VALUE )

replace:ibmattributetypes
ibmattributetypes:( 2.5.4.39 DBNAME ( 'certRevocationLst'
'certRevocationLst' )
   ACCESS-CLASS NORMAL)
```

6. Ajoutez les entrées :

```
# ldapadd -D cn=admin -w secret -f addentries.ldif
 où  addentries.ldif contient les éléments suivants :
```

```
dn: o=aix,c=us
changetype: add
objectclass: organization
objectclass: top
objectclass: pkiCA
o: aix
```

Remarque : Les modèles de fichiers **addentries.ldif** et **setup.ldif** sont fournis dans le module **cas.server**.

7. Arrêtez puis démarrez le démon **slapd**.

Création d'une autorité de certification

Créez l'autorité de certification comme suit :

1. Créez un fichier de référence. Le fichier de référence contient une ou plusieurs paires de mots de passe et de numéros de référence pour la création de certificat. Chaque paire indique les informations d'authentification acceptées par le serveur du service d'authentification de certificats lorsqu'un client de ce service tente de s'authentifier auprès du serveur lors de la création d'un certificat (généralement à l'aide du protocole CMP). Le format du fichier est un numéro de référence suivi d'un mot de passe, tous deux sur des lignes distinctes. Par exemple :

```
12345678
password1234
87654321
password4321
```

   où `12345678` et `87654321` sont les numéros de référence, et `password1234` et `password4321` sont leurs mots de passe respectifs. Les lignes vides ne sont pas autorisées. Les caractères espace ne doivent pas précéder ou suivre des mots de passe ou des numéros de référence. Le fichier doit contenir au moins un mot de passe et un numéro de référence. Vous trouverez un exemple de fichier dans **/usr/cas/server/iafile**. Il vous faudra faire référence à ces valeurs à chaque configuration de client.

2. Configurez l'autorité de certification à l'aide de la commande **mksecpki** comme suit :

```
# mksecpki –u pkiuser –f /usr/cas/server/iafile –p 1077 –H
ldap.cert.mydomain.com \
            –D cn=admin –w secret –i o=aix,c=us
```

   Vous trouverez ci-dessous des informations sur les indicateurs **mksecpki** :

–u	Indique un nom de compte utilisateur sur lequel le serveur du service d'authentification de certificats sera installé.
–f	Indique le fichier de référence créé dans l'étape précédente.
–p	Indique un numéro de port pour le serveur LDAP.
–H	Indique le nom d'hôte ou l'adresse IP du serveur LDAP.
–D	Indique le nom commun de l'administrateur LDAP.
–w	Indique le mot de passe de l'administration LDAP.
–i	Indique la branche LDAP sur laquelle se trouve les données des certificats utilisateur.

   La commande **mksecpki** génère automatiquement la clef de signature sécurisée avec le libellé de clef **TrustedKey**, le mot de passe du compte utilisateur de l'autorité de certification, et la place dans le fichier de magasin de clefs **/usr/lib/security/pki/trusted.pkcs12**. Vous n'avez pas besoin d'exécuter les étapes de Création de la clef de signature sécurisée, page 7-25, à moins d'avoir à générer plusieurs clefs ou de vouloir une clef de signature sécurisée avec un libellé de clef et/ou un mot de passe différent.

Création de la clef de signature sécurisée

La commande **mksecpki** génère automatiquement une clef de signature sécurisée avec le libellé de clef de **TrustedKey**, le mot de passe du compte utilisateur de l'autorité de certification, et la place dans le fichier de magasin de clefs **/usr/lib/security/pki/trusted.pkcs12**. Si vous devez générer une ou plusieurs nouvelles clefs de signature sécurisée, vous trouverez dans cette section les étapes adéquates pour une telle opération.

Tous les clients du service d'authentification de certificats sur lesquels la création ou la révocation de certificats sont autorisées exigent une clef de signature sécurisée pour signer le certificat d'authentification de l'utilisateur. La clef est sauvegardée dans un magasin de clefs distinct et accessible à tous les systèmes sur lesquels des certificats peuvent être créés. Tous les systèmes peuvent utiliser une clef unique ou, pour une approche plus sécurisée, vous pouvez créer et distribuer plusieurs clefs.

Pour créer une clef sécurisée, utilisez la commande **/usr/java131/bin/keytool**. Utilisez un nom de fichier non utilisé. La commande **keytool** vous invite à entrer le mot de passe d'une clef ou d'un magasin de clefs. Ces deux mots de passe peuvent être identiques pour permettre au service d'authentification de certificats d'accéder à la clef dans le magasin. Exécutez la commande **keytool** comme suit :

```
keytool –genkey –dname 'cn=trusted key' –alias 'TrustedKey' –keyalg RSA \
         –keystore   filename .pkcs12 –storetype pkcs12ks
```

Dans cet exemple, le libellé de clef sécurisée est **TrustedKey** et le mot de passe du magasin de clefs sécurisé est fourni par l'utilisateur. Notez bien ces valeurs car vous en aurez besoin lors de la configuration des clients du service d'authentification de certificats. Lorsque vous configurez un client de service d'authentification de certificats, les attributs **keylabel** et **keypasswd** du fichier **acct.cfg** devront être respectivement définis sur le libellé de clef sécurisé et le mot de passe du magasin de clefs sécurisé.

Pour des raisons de sécurité, assurez-vous que le fichier du magasin de clefs (*filename*`.pkcs12`) est protégé en lecture et en écriture. Seul l'utilisateur root doit avoir accès à ce fichier. La clef sécurisée doit être le seul objet du magasin de clefs.

Configuration du client du service d'authentification de certificats

Il existe de nombreuses options de configuration sur le client du service d'authentification de certificats. Les sections suivantes proposent la procédure de configuration requise pour chaque système participant à l'authentification PKI.

Installation de la clef de signature sécurisée

Copiez le magasin de clefs sécurisé contenant la clef de signature sécurisée sur le système local. Pour plus d'informations sur la création de la clef de signature sécurisée, consultez la section Création de la clef de signature sécurisée, page 7-25. L'emplacement par défaut du magasin de clefs sécurisé est dans le répertoire **/usr/lib/security/pki**.

Pour des raisons de sécurité, assurez-vous que le fichier du magasin de clefs est protégé en lecture et en écriture. Seul l'utilisateur root doit avoir accès à ce fichier.

Edition du fichier acct.cfg

Supprimez toutes les strophes **ldap** pouvant exister dans le fichier **/usr/lib/security/pki/acct.cfg** à l'aide d'un éditeur de texte comme **vi**.

Configuration de l'autorité de certification

Le compte de l'autorité de certification locale doit au moins être configuré. Par défaut, il existe mais vous devez le modifier pour qu'il corresponde à votre environnement.

Le service d'authentification de certificats peut utiliser plusieurs autorités de certification pour un seul système via les fichiers de configuration en strophes. **local**, le nom par défaut de la strophe de l'autorité de certification, est utilisé lorsqu'une autorité de certification n'est pas spécifiée par l'utilisateur ou le logiciel. Tous les systèmes doivent disposer d'une telle strophe **local** valide, dans les fichiers de configuration appropriés du service d'authentification de certificats. Une seule autorité de certification peut disposer du nom de strophe **local**. Toutes les autres autorités de certification doivent posséder un nom de strophe unique. Les noms de strophe de l'autorité de certification ne peuvent pas être **ldap** ou **default**.

Les sections suivantes vous guident dans les écrans de configuration de l'outil SMIT pour configurer l'autorité de certification locale.

Modification / Affichage d'une autorité de certification

1. Lancez l'outil SMIT PKI, comme suit :

```
smitty pki
```

2. Sélectionnez **Modification / Affichage d'une autorité de certification**.
3. Tapez `local` pour Nom de l'autorité de certification puis appuyez sur entrée.
4. Configurez la zone **Nom du module de service** sur **/usr/lib/security/pki/JSML.sml**. Il s'agit de la valeur par défaut du module de chargement SML. Cette zone mappe vers l'attribut **program** du fichier **/usr/lib/security/pki/ca.cfg**.
5. Ignorez la zone **Chemin d'accès du certificat de l'autorité de certification**. Cette zone mappe vers l'attribut **certfile** du fichier **/usr/lib/security/pki/ca.cfg**.
6. Configurez la zone **Chemin d'accès à la clef sécurisée de l'autorité de certification** sur une URI correspondant à l'emplacement du magasin de clefs sécurisé sur le système local. Seuls les magasins de clefs sur fichiers sont pris en charge. L'emplacement habituel du magasin de clefs sécurisé est le répertoire **/usr/lib/security/pki**. (Consultez la section Installation de la clef de signature sécurisée, page 7-25.) Cette zone mappe vers l'attribut **trustedkey** du fichier **/usr/lib/security/pki/ca.cfg**.
7. Configurez la zone **URI du serveur de l'autorité de certification** sur une URI correspondant à l'emplacement de l'autorité de certification (**cmp://myserver:1077**). Cette zone mappe vers l'attribut **server** du fichier **/usr/lib/security/pki/ca.cfg**.
8. Ignorez la zone **Point de distribution du certificat**. Cette zone mappe vers l'attribut **cdp** du fichier **/usr/lib/security/pki/ca.cfg**.
9. Configurez la zone **URI de la liste de révocation des certificats (CRL)**. Cette zone indique l'URI, et doit être définie sur l'emplacement de la liste de révocation des certificats de cette autorité de certification. Il s'agit généralement d'une URI LDAP, par exemple :

   ```
   ldap://crlserver/o=   XYZ ,c=   us
   ```

   Cette zone mappe vers l'attribut **crl** du fichier **/usr/lib/security/pki/ca.cfg**.
10. La zone **Nom spécifique du certificat par défaut** indique le DN de base utilisé lors de la création de certificats (par exemple, `o= XYZ,c= us`). Cette zone *n'est pas* obligatoire. Cette zone mappe vers l'attribut **dn** du fichier **/usr/lib/security/pki/ca.cfg**.
11. La zone **URI du nom d'utilisateur supplémentaire par défaut du certificat** indique l'URI correspondante utilisée lors de la création de certificats si elle n'a pas été fournie au moment de la création. Cette zone *n'est pas* obligatoire. Cette zone mappe vers l'attribut **url** du fichier **/usr/lib/security/pki/ca.cfg**.

12. La zone **Algorithme de clef publique** indique l'algorithme utilisé lors de la création d'un certificat. Au choix, **RSA** ou **DSA**. Si aucun n'est spécifié, le système prend comme valeur par défaut **RSA**. Cette zone mappe vers l'attribut **algorithm** du fichier **/usr/lib/security/pki/ca.cfg**.
13. La zone **Taille de la clef publique (en bits)** indique la taille en bits de l'algorithme de la clef publique. Cette zone est en bits, et non en octets, et cette valeur peut être arrondie à la valeur supérieure par le mécanisme de clef publique en cours afin d'utiliser la taille suivante en octets. (L'arrondi est utilisé pour les nombres de bits qui ne sont pas des multiples entiers de 8). Voici quelques exemples, `512`, `1024` et `2048`. Si cette zone n'est pas renseignée, le système prend comme valeur par défaut `1024` bits. Cette zone mappe vers l'attribut **keysize** du fichier **/usr/lib/security/pki/ca.cfg**.
14. La zone **Nombre MAX. de nouvelle tentative de communication** indique le nombre de tentatives du système pour contacter l'autorité de certification (lors de la création ou la révocation d'un certificat) avant abandon. Le système prend comme valeur par défaut le nombre de 5 tentatives. Cette zone mappe vers l'attribut **retries** du fichier **/usr/lib/security/pki/ca.cfg**.
15. La zone **Algorithme de hachage pour signature** indique l'algorithme de hachage utilisé lors de la signature d'un certificat d'authentification. Au choix, **MD2**, **MD5** ou **SHA1**. Le système prend comme valeur par défaut **MD5**. Cette zone mappe vers l'attribut **signinghash** du fichier **/usr/lib/security/pki/ca.cfg**.
16. Appuyez sur Entrée pour valider les modifications.

Modification / Affichage des comptes d'une autorité de certification

1. Lancez l'outil SMIT PKI, comme suit :

```
smitty pki
```

2. Sélectionnez **Modification / Affichage des comptes d'une autorité de certification**.
3. Tapez `local` dans la zone Nom de l'autorité de certification puis appuyez sur entrée.
4. La zone **Numéro de référence de création d'un certificat** indique le numéro de référence utilisé pour créer un certificat. Le numéro de référence de création doit comporter au moins 7 chiffres. Le numéro de référence est défini par l'autorité de certification. (Consultez la section Création de l'autorité de certification, page 7-24.) Cette zone mappe vers l'attribut **carefnum** du fichier **/usr/lib/security/pki/acct.cfg**.
5. La zone **Mot de passe de création d'un certificat** indique le mot de passe utilisé pour créer un certificat. La longueur de ce mot de passe doit être d'au moins 12 caractères de type alphanumériques ASCII 7 bits. Le mot de passe de création est défini dans l'autorité de certification et il doit être associé au numéro de référence de création mentionné ci-dessus. (Consultez la section Création de l'autorité de certification, page 7-24.) Cette zone mappe vers l'attribut **capasswd** du fichier **/usr/lib/security/pki/acct.cfg**.
6. La zone **Numéro de référence de révocation d'un certificat** indique le numéro de référence utilisé pour révoquer un certificat. Le numéro de référence de création doit comporter au moins 7 chiffres. Le numéro de référence de révocation est envoyé à l'autorité de certification qui l'associe au certificat lors de chaque création. Pour révoquer un certificat, le même numéro de référence de révocation (et mot de passe de révocation) que celui envoyé lors de la création doit être envoyé lors de la révocation du certificat. Cette zone mappe vers l'attribut **rvrefnum** du fichier **/usr/lib/security/pki/acct.cfg**.

7. La zone **Mot de passe de révocation d'un certificat** indique le mot de passe de référence utilisé pour révoquer un certificat. La longueur de ce mot de passe doit être d'au moins 12 caractères de type alphanumériques ASCII 7 bits Le mot de passe de révocation est envoyé à l'autorité de certification qui l'associe au certificat lors de chaque création. Pour révoquer un certificat, le même mot de passe de révocation (et numéro de référence de révocation) que celui envoyé lors de la création doit être envoyé lors de la révocation du certificat. Cette zone mappe vers l'attribut **rvpasswd** du fichier **/usr/lib/security/pki/acct.cfg**.
8. La zone **Libellé de clef sécurisé** indique le libellé (parfois appelé *alias*) de la clef de signature sécurisée située dans le magasin de clefs sécurisé. La valeur du libellé de la clef sécurisé est tirée de la section Création de la clef de signature sécurisée, page 7-25. Cette zone mappe vers l'attribut **keylabel** du fichier **/usr/lib/security/pki/acct.cfg**.
9. La zone **Mot de passe de la clef sécurisée** indique le mot de passe de la clef de signature sécurisée située dans le magasin de clefs sécurisé. La valeur du mot de passe de la clef sécurisé est tirée de la section Création de la clef de signature sécurisée, page 7-25. Cette zone mappe vers l'attribut **keypasswd** du fichier **/usr/lib/security/pki/acct.cfg**.
10. Appuyez sur Entrée pour valider les modifications.

Ajout du compte LDAP pour l'autorité de certification

1. Lancez l'outil SMIT PKI, comme suit :

```
smitty pki
```

2. Sélectionnez **Ajout d'un compte LDAP**.
3. La zone **Nom de l'utilisateur administratif** indique le DN du compte administratif LDAP. Le nom de l'utilisateur administratif du compte LDAP de l'autorité de certification est le même que celui utilisé dans les sections Configuration du serveur LDAP, page 7-20 et Configuration LDAP du serveur pour le service d'authentification de certificats, page 7-23. La valeur doit être `cn=admin`. Elle permet au client de communiquer avec le serveur LDAP lors
   de l'accès aux données LDAP de l'autorité de certification. Cette zone mappe vers l'attribut **ldappkiadmin** du fichier **/usr/lib/security/pki/acct.cfg**.
   Par exemple :

```
ldappkiadmin = "cn=admin"
```

4. La zone **Mot de passe administratif** indique le mot de passe du compte administratif LDAP. Le mot de passe administratif est le même que celui utilisé dans la section Configuration du serveur LDAP, page 7-20 et Configuration LDAP du serveur pour le service d'authentification de certificats, page 7-23. Cette zone mappe vers l'attribut **ldappkiadmpwd** du fichier **/usr/lib/security/pki/acct.cfg**.
   Par exemple :

```
ldappkiadmpwd = secret
```

5. La zone **Nom du serveur** indique le nom du serveur LDAP et doit être définie dans chaque strophe LDAP. La valeur est un nom de serveur LDAP unique.
   Cette zone mappe vers l'attribut **ldapservers** du fichier **/usr/lib/security/pki/acct.cfg**.
   Par exemple :

```
ldapservers = ldapserver.mydomain.com
```

6. La zone **Suffixe** indique le suffixe DN de l'arborescence du répertoire sur lequel se trouvent les données. Le suffixe correspond à la valeur de l'attribut **ibm–slapdSuffix** utilisé dans la section Configuration LDAP du serveur pour le service d'authentification de certificats, page 7-23. Cet attribut doit être défini dans chaque strophe LDAP.
   Cette zone mappe vers l'attribut **ldapsuffix** du fichier **/usr/lib/security/pki/acct.cfg**.
   Par exemple :

```
ldapsuffix = "ou=aix,cn=us"
```

7. Appuyez sur Entrée pour valider les modifications.

Ajout de PKI pour chaque compte utilisateur LDAP

Exécutez les mêmes étapes que décrites dans la section Ajout du compte LDAP de l'autorité de certification, page 7-28, sauf que vous utilisez les valeurs utilisées dans l'étape **Ajout des entrées de base de données ACL et suffixe PKI** de la section Configuration du serveur LDAP
pour PKI, page 7-21. Utilisez les valeurs suivantes :

- Nom de l'utilisateur administratif (`ou=pkidata,cn=aixdata`),
- Mot de passe administratif (*`password`*),
- Nom du serveur (*`spécifique au site`*),
- Suffixe (`ou=pkidata,cn=aixdata`).

Appuyez sur Entrée pour valider les modifications.

Modification / Affichage de la politique

1. Lancez l'outil SMIT PKI, comme suit :

```
smitty pki
```

2. Sélectionnez **Modification / Affichage de la politique**.

- La zone **Création de certificats pour de nouveaux utilisateurs** indique si la commande **mkuser** génère un certificat et un magasin de clefs pour le nouvel utilisateur (**new**), ou si l'administrateur fournit un certificat et un magasin de clefs une fois l'utilisateur créé (**get**). Cette zone mappe vers l'attribut **cert** de la strophe **newuser** du fichier **/usr/lib/security/pki/policy.cfg**.
- La zone **Nom de l'autorité de certification** indique l'autorité de certification utilisée par la commande **mkuser** lors de la génération d'un certificat. La valeur de cette zone doit correspondre au nom d'une strophe figurant dans le fichier **ca.cfg** ; par exemple, **local**. Cette zone mappe vers l'attribut **ca** de la strophe **newuser** du fichier **/usr/lib/security/pki/policy.cfg**.
- La zone **Mot de passe utilisateur initial** indique le mot de passe utilisé par la commande **mkuser** lors de la création du magasin de clefs d'un utilisateur. Cette zone mappe vers l'attribut **passwd** de la strophe **newuser** du fichier **/usr/lib/security/pki/policy.cfg**.
- La zone **Version du certificat** indique la version utilisée par la commande **mkuser** lors de la génération d'un certificat. Actuellement, la seule valeur prise en charge est 3, ce qui correspond à X.509v3. Cette zone mappe vers l'attribut **version** de la strophe **newuser** du fichier **/usr/lib/security/pki/policy.cfg**.
- La zone **Taille de la clef publique** indique la taille (en bits) de la clef publique utilisée par la commande **mkuser** lors de la génération d'un certificat. Cette zone mappe vers l'attribut **keysize** de la strophe **newuser** du fichier **/usr/lib/security/pki/policy.cfg**.
- La zone **Emplacement du magasin de clefs** indique le format URI du répertoire du magasin de clefs utilisé par la commande **mkuser** lors de la création d'un magasin de clefs. Cette zone mappe vers l'attribut **keystore** de la strophe **newuser** du fichier **/usr/lib/security/pki/policy.cfg**.
- La zone **Période de validité** indique la période de validité du certificat utilisée par la commande **mkuser** lors de la génération d'un certificat. La période validité demandée peut ou non être honorée par l'autorité de certification lors de la création du certificat. Cette période peut être indiquée en secondes, jours ou années. Si vous ne proposez qu'un seul nombre, on considère qu'il s'agit de secondes. Si la lettre `d` suit immédiatement le nombre, on considère qu'il s'agit de jours. Si la lettre `y` suit immédiatement le nombre, on considère qu'il s'agit d'années. Voici des exemples :
  - 1y (pour 1 an)
  - 30d (pour 30 jours)

  - 2592000 (pour 30 jours en secondes). Cette zone mappe vers l'attribut **validity** de la strophe **newuser** du fichier **/usr/lib/security/pki/policy.cfg**.

- La zone **Réplication de certificats non locaux** indique si la commande **certlink** sauvegarde une copie d'un certificat (**yes**) ou juste le lien vers ce certificat (**no**). Cette zone mappe vers l'attribut **replicate** de la strophe **storage** du fichier **/usr/lib/security/pki/policy.cfg**.
- La zone **Vérification des listes de révocation de certificats** indique si les commandes **certadd** et **certlink** vérifient (**yes**) ou non (**no**) la CRL avant d'exécuter leurs tâches. Cette zone mappe vers l'attribut **check** de la strophe **crl** du fichier **/usr/lib/security/pki/policy.cfg**.
- La zone **Délai de communication par défaut** indique le délai en secondes utilisée par les commandes **certadd** et **certlink** lors de la demande d'informations sur des certificats à l'aide du HTTP (par exemple, extraction des CRL). Cette zone mappe vers l'attribut **timeout** de la strophe **comm** du fichier **/usr/lib/security/pki/policy.cfg**.

Le fichier methods.cfg

Le fichier **methods.cfg** indique les définitions de la grammaire d'authentification utilisée par les attributs **registry** et **SYSTEM**. C'est là que la grammaire d'authentification de **PKILDAP** (pour PKI utilisant LDAP) et **FPKI** (*fichiers* PKI) doit être définie et ajoutée par l'administrateur système.

Vous trouverez ci-dessous une définition **methods.cfg** type. Les noms des strophes **PKI**, **LDAP** et **PKILDAP** sont des noms arbitraires qui peuvent être modifiés par l'administrateur. Ces noms des strophes sont utilisées tout au long de cette section.

```
PKI:
         program = /usr/lib/security/PKI
         options = authonly

LDAP:
         program = /usr/lib/security/LDAP

PKILDAP:
         options = auth=PKI,db=LDAP
```

Pour prendre en charge des utilisateurs visiteurs, utilisez les mêmes noms de strophe **methods.cfg** et valeurs d'attribut sur tous les systèmes assurant cette option.

Exemples des configuration de l'administration

Création d'un nouveau compte utilisateur PKI

Pour créer un nouveau compte utilisateur PKI, utilisez la commande **mkuser** et le nom de strophe **/usr/lib/security/methods.cfg** approprié (**PKILDAP**). Selon les attributs du fichier **/usr/lib/security/pki/policy.cfg**, la commande **mkuser** peut créer automatiquement un certificat pour l'utilisateur. Voici un exemple de commande **mkuser** qui crée le compte utilisateur `bob` :

```
mkuser –R PKILDAP SYSTEM="PKILDAP" registry=PKILDAP bob
```

Conversion d'un compte utilisateur non–PKI en un compte utilisateur PKI

Il existe deux façons pour convertir un compte utilisateur non–PKI en un compte utilisateur PKI. La première permet à l'administrateur du système d'accéder au magasin de clefs privé de l'utilisateur initial, ce qui peut ne pas être autorisé dans un environnement donné, et c'est la méthode la plus rapide. La seconde exige l'interaction entre l'utilisateur et l'administrateur du système, ce qui peut prendre plus de temps à configurer.

Ces deux exemples utilisent les hypothèses suivantes :

- **cas.server** et **cas.client** sont déjà installés, configurés et en cours de fonctionnement.
- **PKILDAP** est défini dans **methods.cfg** comme indiqué dans la section Le fichier methods.cfg, page 7-30.

Exemple 1 :

Avec l'autorité root, l'administrateur du système peut exécuter les commandes suivantes pour le compte utilisateur `bob` :

```
certcreate -f cert1.der -l auth_lbl1 cn=bob bob  # Créez et sauvegardez
                                                   cert dans cert1.der.
 certadd -f cert1.der -l auth_lbl1 auth_tag1 bob  # Ajoutez cert dans LDAP
                                                    en tant que auth_tag1.
 certverify auth_tag1 bob                         # Vérifiez et signez cert
                                                    dans LDAP.
 chuser SYSTEM="PKILDAP" registry=PKILDAP bob     # Modifiez le type de
                                                    compte en PKILDAP.
 chuser -R PKILDAP auth_cert=auth_tag1 bob        # Configurez le certificat
                                                    auth de l'utilisateur.
```

Modifiez ensuite le mot de passe du magasin de clefs de l'utilisateur `bob` à l'aide de la commande **keypasswd**.

Exemple 2 :

L'utilisateur `bob` doit exécuter les 3 premières commandes de l'exemple 1 ci-dessus (**certcreate**, **certadd**, **certverify**) pour créer son propre certificat et magasin de clefs. L'administrateur du système doit ensuite exécuter les deux dernières commandes **chuser** de l'exemple 1 ci-dessus.

Création et ajout d'un certificat d'authentification

Si un utilisateur PKI demande un nouveau certificat d'authentification, l'utilisateur peut créer un nouveau certificat et demander à l'administrateur du système d'en faire un certificat d'authentification. L'exemple ci-dessous montre la création d'un certificat par l'utilisateur `bob`, puis sa conversion par l'administrateur du système en un certificat d'authentification.

```
# Connexion en tant que compte utilisateur bob :
 certcreate -f cert1.der -l auth_lbl1 cn=bob  # Créez et sauvegardez cert
                                                dans cert1.der.
 certadd -f cert1.der -l auth_lbl1 auth_tag1  # Ajoutez cert dans LDAP en
                                                tant que auth_tag1.
 certverify auth_tag1                         # Vérifiez et signez le cert
                                                dans LDAP.
 # En tant qu'administrateur du système :
 chuser -R PKILDAP auth_cert=auth_tag1 bob    # Configurez le certificat
                                                auth de l'utilisateur.
```

Modification du mot de passe par défaut du nouveau magasin de clefs

Pour modifier le mot de passe utilisé pour créer les magasins de clefs des nouveaux utilisateurs PKI, éditez la valeur de l'attribut **passwd** de la strophe **newuser** dans le fichier **/usr/lib/security/pki/policy.cfg**.

Gestion d'une clef de signature sécurisée compromise

Le fichier qui contient la clef de signature sécurisée doit être remplacé et les certificats d'authentification de l'utilisateur doivent être de nouveau signés.

Gestion d'une clef privée d'utilisateur compromise

Si la clef privée d'un utilisateur est compromise, l'utilisateur ou l'administrateur doit révoquer le certificat au moyen du code de raison approprié, les autres utilisateurs utilisant la clef publique doivent en être informés et, selon l'utilisation de cette clef privée/publique, un nouveau certificat doit être émis. Si le certificat a été utilisé comme certificat d'authentification, un autre certificat (soit le nouveau, soit un certificat existant non promis détenu par l'utilisateur) doit être ajouté comme nouveau certificat d'authentification.

Gestion d'un magasin de clefs ou d'un mot de passe de magasin de clefs compromis

Modifiez le mot de passe du magasin de clefs. Révoquez tous les certificats utilisateur. Créez de nouveaux certificats pour l'utilisateur, y compris un nouveau certificat d'authentification. Les clefs privées compromises peuvent toujours servir à l'utilisateur pour accéder à des données chiffrées auparavant.

Déplacement du magasin de clefs d'un utilisateur ou modification du nom de magasin de clefs d'un utilisateur

Si la clef privée d'un utilisateur est compromise, l'utilisateur ou l'administrateur doit révoquer le certificat au moyen du code de raison approprié, les autres utilisateurs utilisant la clef publique doivent en être informés et, selon la fonction de la clef privée et publique, un nouveau certificat doit être émis. Si le certificat a été utilisé comme certificat d'authentification, un autre certificat (soit le nouveau, soit un certificat existant non promis détenu par l'utilisateur) doit être ajouté comme nouveau certificat d'authentification.

Déplacement du magasin de clefs d'un utilisateur ou modification du nom de magasin de clefs d'un utilisateur

Chaque certificat utilisateur conservé dans LDAP contient l'emplacement du magasin de clefs de la clef privée correspondante. Déplacer le magasin de clefs d'un utilisateur d'un répertoire à un autre ou modifier le nom du magasin de clefs impose de modifier le nom et l'emplacement du magasin de clefs LDAP associé aux certificats de l'utilisateur. Si l'utilisateur possède plusieurs magasins de clefs, vous devez faire particulièrement attention à ne modifier que les informations LDAP des certificats concernés par cette modification.

Pour déplacer un magasin de clefs de **/var/pki/security/keys/user1.p12** vers **/var/pki/security1/keys/user1.p12** :

```
# En tant que root...

cp /var/pki/security/keys/user1.p12 /var/pki/security1/keys/user1.p12

# Récupérez une liste de tous les certificats associés à cet utilisateur.
certlist ALL user1

# Pour chaque certificat associé au magasin de clefs, procédez comme suit :
# A) Récupérez le libellé de la clef privée du certificat et son statut " vérifié ".
# B) Récupérez le certificat dans LDAP.
# C) Remplacez le certificat dans LDAP à l'aide du même libellé de clef privée,
# sans le raccourci du nouveau magasin de clefs.
# D) Si le certificat a été vérifié au préalable, il doit être à nouveau vérifié.
# (l'étape D exige le mot de passe du magasin de clefs.)

# Exemple de modification d'un seul certificat.
# Considérons que :

# nomutilisateur : user1

# cert tag : tag1

# key label : label1

# Récupérez le libellé de la clef privée du certificat.
certlist -a label tag1 user1

# Récupérez le certificat dans LDAP et placez-le dans le fichier cert.der.
certget -f cert.der tag1 user1

# Remplacez le certificat dans LDAP.
certadd -r -f cert.der -p /var/pki/security1/keys/user1.p12 -l label1 tag1 user1

# Faites une nouvelle vérification du certificat précédemment vérifié.
# (vous devez connaître le mot de passe du magasin de clefs.)
certverify tag1 user1
```

Chapitre 8. Modules d'extension d'authentification (PAM)

La structure PAM (pluggable authentification module) permet d'incorporer plusieurs mécanismes d'authentification à un système à l'aide de modules d'extension. Les applications compatibles PAM peuvent bénéficier d'*extensions* sans devoir être modifiées. Cette souplesse permet aux administrateurs d'effectuer les tâches suivantes :

- Sélectionner tout service d'authentification du système pour une application
- Utiliser plusieurs mécanismes d'authentification pour un service donné
- Ajouter de nouveaux modules de services d'authentification sans modifier les applications
- Utiliser un mot de passe entré précédemment pour l'authentification avec plusieurs modules

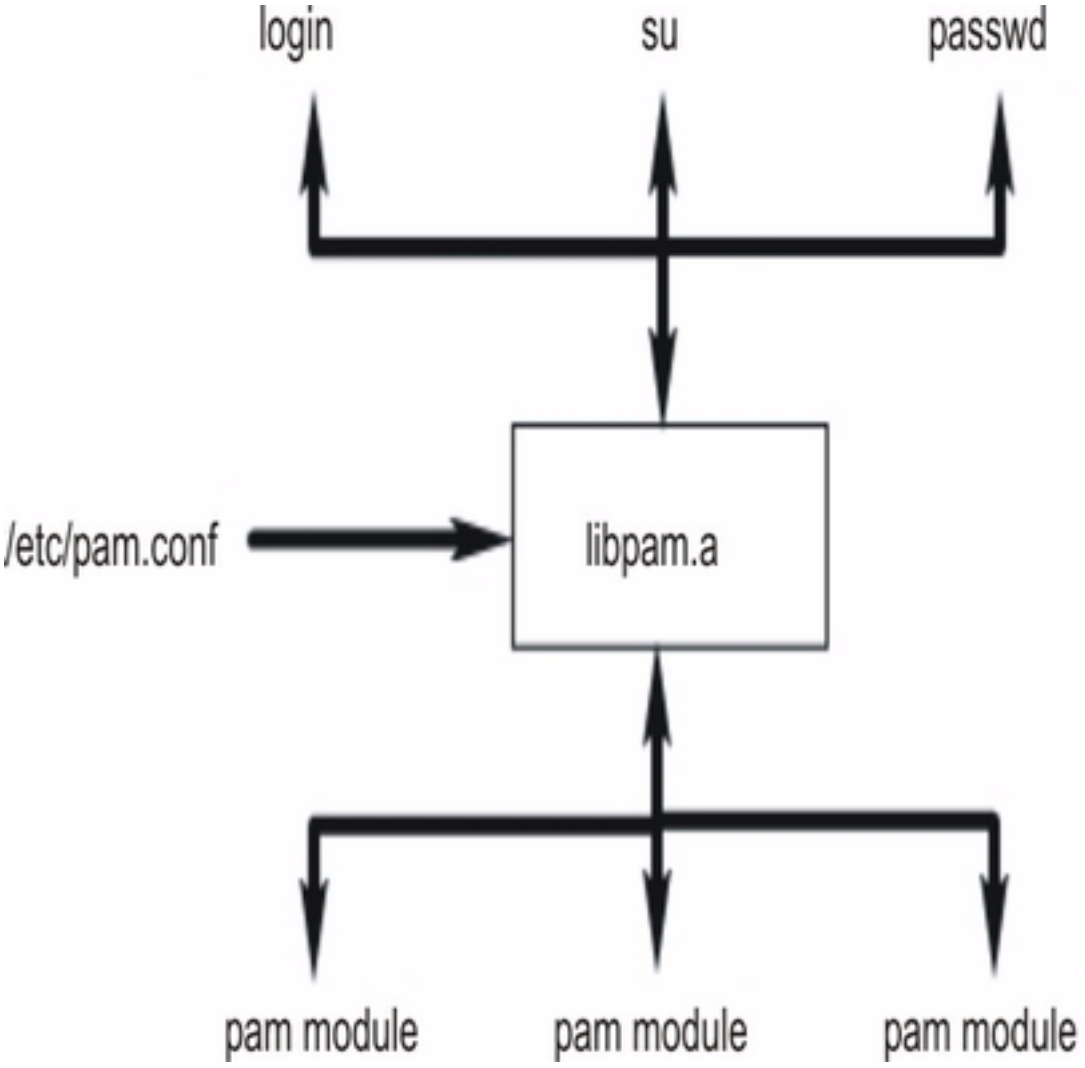
La structure PAM est composée d'une bibliothèque, de modules d'extension et d'un fichier de configuration. La bibliothèque PAM met en œuvre l'interface de programmation d'application (API) PAM et sert à gérer les transactions PAM et à appeler la SPI (service programming interface) PAM définie dans les modules d'extension. Les modules d'extension sont chargés dynamiquement par la bibliothèque, en fonction du service appelant et de son entrée dans le fichier de configuration. Le succès dépend du module d'extension ainsi que du comportement défini pour le service. Le concept de *succession* permet de configurer un service d'authentification utilisant plusieurs méthodes. Le cas échéant, des modules peuvent être configurés pour utiliser un mot de passe entré précédemment plutôt que d'exiger, via une invite, une entrée supplémentaire.

L'administrateur du système peut configurer un système AIX afin qu'il utilise le module d'extension d'authentification (PAM) en modifiant l'attribut **auth_type** dans la strophe **usw** du fichier **/etc/security/login.cfg**. Le réglage **auth_type = PAM_AUTH** permet de configurer les commandes compatibles PAM pour appeler l'API PAM directement en vue de l'authentification plutôt que d'utiliser les routines d'authentification AIX historiques. Cette configuration doit être effectuée en cours d'exécution et ne nécessite pas le redémarrage du système pour prendre effet. Pour plus d'informations sur l'attribut **auth_type**, reportez–vous à la référence du fichier **/etc/security/login.cfg**. Les commandes et applications AIX natives qui suivent ont été modifiées pour reconnaître l'attribut **auth_type** et ont été activées pour l'authentification PAM :

- login
- passwd
- su
- ftp
- telnet
- rlogin
- rexec
- rsh
- snappd
- imapd

Le schéma suivant montre l'interaction entre les applications compatibles PAM, la bibliothèque PAM, le fichier de configuration et les modules PAM sur un système qui a été configuré pour utiliser PAM. Les applications compatibles PAM appellent l'interface API PAM de la bibliothèque PAM. La bibliothèque détermine le module à charger en fonction de l'entrée de l'application dans le fichier de configuration, et appelle la SPI PAM dans le module. La communication entre le module PAM et l'application s'effectue via l'utilisation d'une fonction de conversation implémentée dans l'application. Le succès ou l'échec du module et le comportement défini dans le fichier de configuration déterminent ensuite s'il faut charger un autre module. Si tel est le cas, le processus se poursuit. Sinon, le résultat est retourné à l'application.

Figure 3. Structure et entités PAM Ce schéma montre comment des commandes compatibles PAM utilisent la bibliothèque PAM pour accéder au module PAM approprié.



Bibliothèque PAM

La bibliothèque PAM, **/usr/lib/libpam.a** contient l'API PAM qui sert d'interface commune à toutes les applications PAM et qui permet également de contrôler le chargement des modules. Les modules sont chargés par la bibliothèque PAM en fonction du comportement de succession défini dans le fichier **/etc/pam.conf**.

Les fonctions suivantes de l'API PAM invoquent la SPI PAM correspondante fournie par un module PAM. Par exemple, l'API **pam_authenticate** appelle la SPI **pam_sm_authenticate** dans un module PAM.

- **pam_authenticate**
- **pam_setcred**
- **pam_acct_mgmt**
- **pam_open_session**
- **pam_close_session**
- **pam_chauthtok**

La bibliothèque PAM fournit aussi des fonctions permettant à une application d'invoquer des modules PAM et de leur envoyer des informations. Les API suivantes de la structure PAM sont mises en œuvre dans AIX :

pam_start	Lancement d'une session PAM
pam_end	Fermeture d'une session PAM
pam_get_data	Récupération de données spécifiques au module
pam_set_data	Définition de données spécifiques au module
pam_getenv	Récupération de la valeur d'une variable d'environnement PAM définie
pam_getenvlist	Récupération d'une liste de toutes les variables d'environnement PAM définies et de leurs valeurs
pam_putenv	Réglage d'une variable d'environnement PAM
pam_get_item	Récupération d'informations de PAM communes
pam_set_item	Définition d'informations de PAM communes
pam_get_user	Récupération d'un nom d'utilisateur
pam_strerror	Obtention d'un message d'erreur standard PAM

Modules PAM

Les modules PAM permettent d'utiliser sur un système plusieurs mécanismes d'authentification, collectivement ou indépendamment. Un module PAM donné doit mettre en œuvre au moins l'un des quatre types de modules. Les types de modules sont décrits ci–dessous, accompagnés des SPI PAM requises pour se conformer au type de module.

Modules d'authentification
: Authentifient les utilisateurs et définissent, rafraîchissent ou détruisent les données d'identification. Ces modules identifient l'utilisateur en fonction de son authentification et de ses données d'identification.

Fonctions des modules d'authentification :

- **pam_sm_authenticate**
- **pam_sm_setcred**

Modules de gestion de comptes
: Déterminent la validité du compte utilisateur et des accès suivants, après l'identification par le module d'authentification. Les vérifications effectuées par ces modules incluent généralement des restrictions de mot de passe et d'expiration de compte.

Fonction du module de gestion de comptes :

- **pam_sm_acct_mgmt**

Modules de gestion de sessions
: Lancent et mettent fin aux sessions utilisateur. Vous pouvez aussi bénéficier d'un audit de sessions.

Fonctions du module de gestion de sessions :

- **pam_sm_open_session**
- **pam_sm_close_session**

Modules de gestion des mots de passe
: Ils modifient les mots de passe et gèrent les attributs liés.

Fonction du module de gestion des mots de passe :

- **pam_sm_chauthtok**

Fichier de configuration PAM

Le fichier de configuration **/etc/pam.conf** contient des entrées de service pour chaque type de module PAM et sert à acheminer des services via un chemin d'accès défini. Les entrées du fichier se composent des zones suivantes, délimitées par des espaces :

```
service_name module_type control_flag module_path module_options
```

Où :

service_name	Indique le nom du service. Le mot clé OTHER définit le module par défaut à utiliser pour les applications non spécifiées dans une entrée.
module_type	Désigne le type de module pour le service. Les types de module valides sont **auth**, **account**, **session** ou **password**. Un module donné fournit une prise en charge pour un ou plusieurs types de module.
control_flag	Désigne le comportement de succession du module. Les indicateurs de contrôle pris en charge sont required (obligatoire), requisite (requis), sufficient (suffisant) ou optional (facultatif).
module_path	Désigne le chemin d'accès vers l'objet de la bibliothèque qui met en œuvre la fonctionnalité du service. Les entrées de *module_path* doivent commencer au répertoire root (**/**). Si l'entrée ne commence pas par **/**, alors **/usr/lib/security** est ajouté au début du nom de fichier.
module_options	Spécifie une liste, séparée par des espaces, d'options qui peuvent être transmises aux modules de services. Les valeurs de cette zone dépendent des options prises en charge par le module défini dans la zone *module_path*.

Les entrées incorrectes ou comportant des valeurs non valides pour les zones *module_type* ou *control_flag* sont ignorées par la bibliothèque PAM. Les entrées comportant un dièse (#) au début de la ligne sont également ignorées car mises en commentaire.

PAM prend en charge un concept généralement désigné sous le nom de " succession ", permettant à plusieurs mécanismes d'être utilisés pour chaque service. La succession est mise en œuvre dans le fichier de configuration par la création de plusieurs entrées pour un service avec la même zone *module_type*. Les modules sont appelés selon leur ordre d'apparition dans le fichier pour un service donné, le dernier résultat étant déterminé par le champ *control_flag* indiqué pour chaque entrée. Les valeurs acceptées pour la zone *control_flag* et leur comportement dans la suite sont décrits ci–dessous :

required (obligatoire)	Tous les modules obligatoires d'une suite doivent être présents pour obtenir un résultat positif. Si l'un d'eux échoue, tous les modules obligatoires de la suite seront essayés, mais l'erreur du premier module obligatoire ayant échoué sera retournée.
requisite (requis)	Valeur identique à Obligatoire, hormis le fait que si un module requis échoue, aucun autre module de la succession n'est traité et le premier code d'erreur provenant d'un module obligatoire ou requis est retourné.

sufficient (suffisant)	Si un module marqué ainsi est correct, et si aucun module obligatoire ou suffisant précédent n'a échoué, tous les modules restants dans la suite sont ignorés.
optional (facultatif)	Si aucun module de la suite n'est obligatoire et si aucun module suffisant n'a enregistré un succès, au moins l'un des modules facultatifs doit fournir un succès pour le service. Si un autre module de la suite enregistre un succès, l'échec d'un module facultatif est ignoré.

Le sous–ensemble suivant **/etc/pam.conf** est un exemple de succession dans le type de module **auth** pour le service de connexion.

```
#
# PAM configuration file /etc/pam.conf
#

# Authentication Management
login   auth     required       /usr/lib/security/pam_ckfile
file=/etc/nologin
login   auth     required       /usr/lib/security/pam_aix
login   auth     optional       /usr/lib/security/pam_test
use_first_pass
OTHER   auth     required       /usr/lib/security/pam_prohibit
```

Le fichier de configuration exemple contient trois entrées pour le service de connexion. **pam_ckfile** et **pam_aix** étant tous deux spécifiés comme requis, les deux modules seront exécutés et tous deux devront aboutir avec succès pour garantir le succès du résultat général. La troisième entrée pour le module fictif **pam_test** est facultative et son succès ou son échec n'affecteront pas la capacité de l'utilisateur à se connecter. L'option **use_first_pass** du module **pam_test** exige l'utilisation d'un mot de passe précédemment entré plutôt que d'avoir à en entrer un nouveau.

L'utilisation du mot de passe **OTHER** comme nom de service permet de définir une valeur par défaut pour tout autre service non déclaré explicitement dans le fichier de configuration. La définition d'une valeur par défaut garantit que chaque cas pour un type de module donné sera couvert par au moins un module. Dans cet exemple, tous les services autres que la connexion se solderont par un échec étant donné que le module **pam_prohibit** renvoie un échec PAM pour toutes les invocations.

Ajout d'un module PAM

Procédez comme suit pour ajouter un module PAM :

1. Installez le module dans le répertoire **/usr/lib/security**.
2. Définissez la propriété du fichier sur root et les permissions sur 555.
   La bibliothèque PAM ne charge aucun module n'appartenant pas à l'utilisateur root.
3. Mettez à jour le fichier de configuration **/etc/pam.conf** pour inclure le module dans les entrées des noms de services désirés.
4. Vérifiez que les services concernés fonctionnent correctement.
   Ne vous déconnectez pas du système avant d'avoir effectué un test de connexion.

Modification du fichier /etc/pam.conf

Lorsque vous modifiez le fichier de configuration **/etc/pam.conf**, tenez compte des éléments suivants :

- Le fichier doit toujours appartenir à la sécurité du groupe et de l'utilisateur root. Les droits d'accès au fichier doivent être définis sur 644 pour permettre à tout le monde d'y avoir accès en lecture et pour permettre uniquement à l'utilisateur root de le modifier.
- Pour augmenter la sécurité, pensez à configurer de manière explicite chaque service compatible PAM puis à utiliser le module **pam_prohibit** pour le mot–clé de service OTHER.
- Lisez la documentation fournie avec chaque module choisi, puis déterminez les indicateurs de contrôle et options pris en charge, ainsi que leur effet.
- Classez les modules et indicateurs de contrôle avec soin. Souvenez–vous du comportement des indicateurs de contrôle **required** (obligatoire), **requisite** (requis), **sufficient** (suffisant) et **optional** (facultatif) dans les successions de modules.

Remarque : Une configuration incorrecte du fichier de configuration PAM peut empêcher toute connexion au système, étant donné que la configuration s'applique à tous les utilisateurs, y compris l'utilisateur root. Une fois le fichier modifié, testez systématiquement les applications concernées avant de vous déconnecter du système. Pour récupérer un système auquel il est impossible de se connecter, amorcez–le en mode maintenance et corrigez le fichier de configuration **/etc/pam.conf**.

Activation du débogage PAM

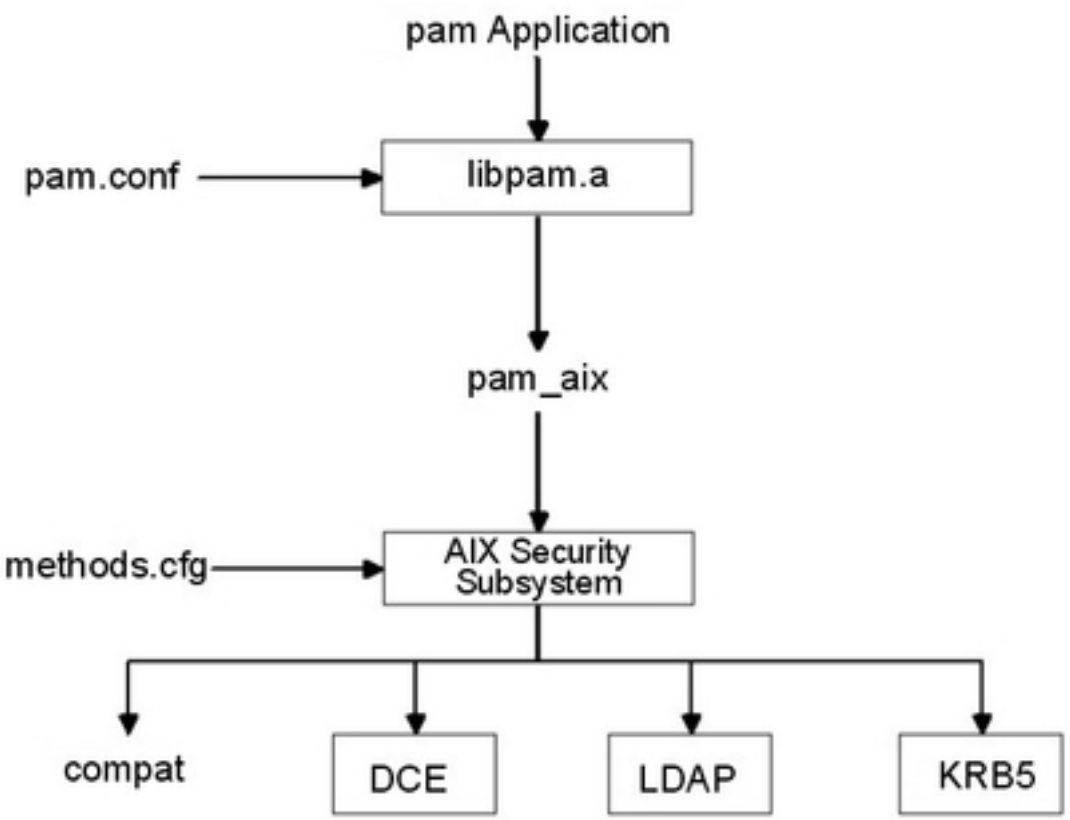
La bibliothèque PAM peut fournir des informations de débogage durant son fonctionnement. Une fois le système activé pour le recueil de sorties de débogage, les informations collectées permettent de suivre les appels à l'API PAM et de localiser les points de panne de la configuration PAM actuelle. Pour activer les sorties de débogage PAM, procédez comme suit :

1. Créez un fichier vide dans **/etc/pam_debug**. La bibliothèque PAM contrôle l'existence du fichier **/etc/pam_debug** et active la sortie syslog.
2. Modifiez le fichier **/etc/syslog.conf** pour inclure les entrées nécessaires pour les niveaux de messages souhaités.
3. Relancez le démon **syslogd** afin que la nouvelle configuration soit reconnue.
4. Une fois l'application PAM redémarrée, les messages de débogage seront recueillis dans le fichier défini dans le fichier de configuration **/etc/syslog.conf**.

Module pam_aix

Le module **pam_aix** est un module PAM qui permet aux applications activées par PAM d'accéder aux services de sécurité AIX en fournissant les interfaces qui appellent les services AIX équivalents lorsqu'ils existent. Ces services sont alors exécutés par un module d'authentification compatible ou par la fonction AIX intégrée, en fonction de la définition de l'utilisateur et de la configuration correspondante dans le fichier **methods.cfg**. Tous les codes d'erreur générés lors de l'exécution d'un service AIX sont convertis en codes PAM correspondants.

Figure 4. Chemin de l'application PAM vers le sous–système de sécurité AIX
Ce schéma montre le chemin suivi par un appel d'API d'une application PAM si le fichier **/etc/pam.conf** est configuré pour utiliser le module **pam_aix**. Comme l'indique l'illustration, l'intégration permet l'authentification des utilisateurs par tout module d'authentification chargeable (DCE, LDAP ou KRB5) ou dans les fichiers AIX (compat).



Le module **pam_aix** est installé dans le répertoire **/usr/lib/security**. L'intégration du module **pam_aix** nécessite la configuration du fichier **/etc/pam.conf**. La succession est toujours disponible mais n'est pas illustrée dans l'exemple suivant de fichier **/etc/pam.conf** :

```
#
# Authentication management
#
OTHER   auth     required       /usr/lib/security/pam_aix

#
# Account management
#
OTHER   account  required       /usr/lib/security/pam_aix

#
# Session management
#
OTHER   session  required       /usr/lib/security/pam_aix

#
# Password management
#
OTHER   password required       /usr/lib/security/pam_aix
```

Le module **pam_aix** prend en charge les fonctions SPI **pam_sm_authenticate**, **pam_sm_chauthok** et **pam_sm_acct_mgmt**. Les SPI **pam_sm_setcred**, **pam_sm_open_session** et **pam_sm_close_session** sont aussi implémentées dans le module **pam_aix**, mais ces fonctions SPI retournent simplement des appels **PAM_SUCCESS**.

L'exemple suivant est un mappage approximatif d'appels SPI PAM vers le sous–système de sécurité AIX :

```
PAM SPI                    AIX
=========                  =====
pam_sm_authenticate   -->  authenticate
pam_sm_chauthtok      -->  passwdexpired, chpass
                           Note: passwdexpired is only checked if the
                           PAM_CHANGE_EXPIRED_AUTHTOK flag is passed in.
pam_sm_acct_mgmt      -->  loginrestrictions, passwdexpired
pam_sm_setcred        -->  No comparable mapping exists, PAM_SUCCESS returned
pam_sm_open_session   -->  No comparable mapping exists, PAM_SUCCESS returned
pam_sm_close_session  -->  No comparable mapping exists, PAM_SUCCESS returned
```

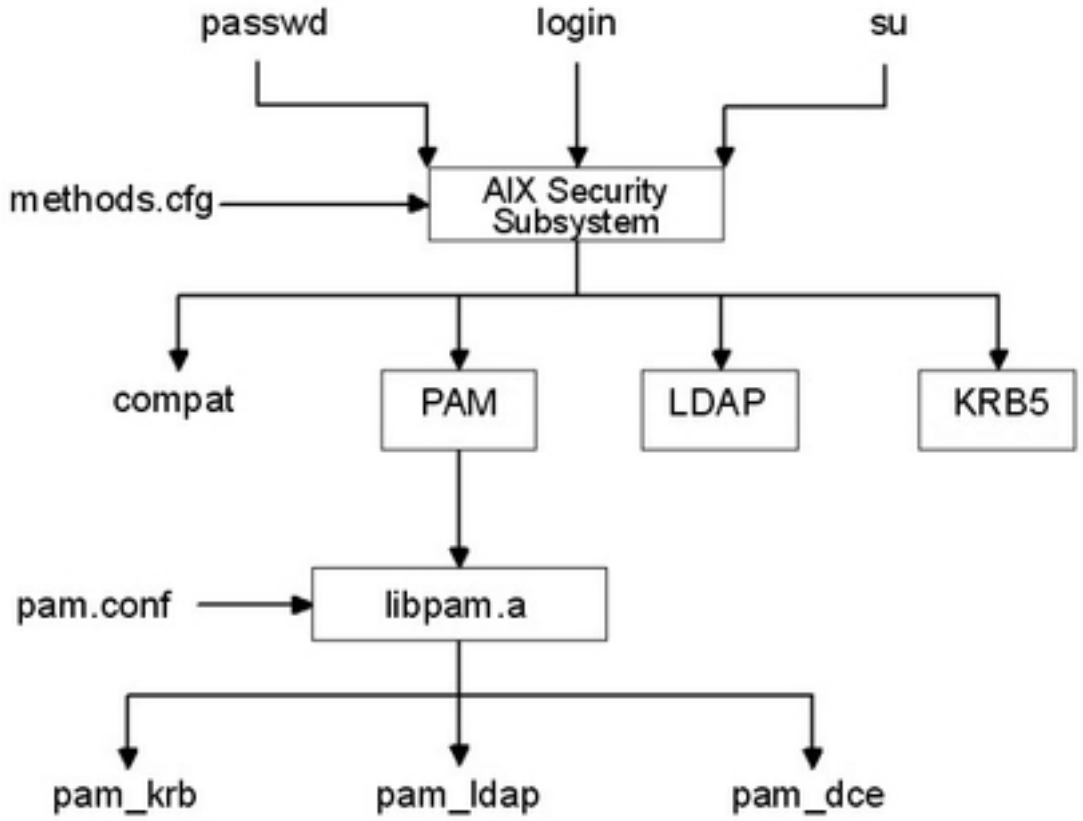
Les données à transmettre au sous–système de sécurité AIX peuvent être définies à l'aide de la fonction **pam_set_item** avant d'utiliser le module, ou à l'aide du module **pam_aix** pour les données si elles n'existent pas encore.

Module d’authentification PAM compatible

Remarque : Dans les versions antérieures à AIX 5.3, un module d’authentification PAM compatible était utilisé pour fournir une authentification PAM aux applications AIX natives. En raison des différences de comportement entre cette solution et une véritable solution PAM, le module d’authentification PAM compatible n’est plus la méthode recommandée pour fournir une authentification PAM aux applications AIX natives. Au lieu de cela, la valeur de l’attribut **auth_type** de la strophe **usw** de **/etc/security/login.cfg** doit être **PAM_AUTH**, afin d’activer l’authentification PAM sous AIX. Pour plus d’informations sur l’attribut **auth_type**, reportez–vous à la description du fichier **/etc/security/login.cfg**. L’utilisation du module d’authentification PAM compatible est toujours prise en charge mais elle n’est pas conseillée. Il est recommandé d’utiliser l’attribut **auth_type** pour activer l’authentification PAM.

Les services de sécurité AIX peuvent être configurés pour appeler les modules PAM à l’aide de la structure de modules d’authentification existante compatible avec AIX. Lorsque le fichier **/usr/lib/security/methods.cfg** est configuré correctement, le module de chargement PAM achemine les services de sécurité AIX (**passwd**, **login**, etc.) vers la bibliothèque PAM. La bibliothèque PAM contrôle le fichier **/etc/pam.conf** pour déterminer quel module PAM utiliser. Elle effectue ensuite l’appel du SPI PAM correspondant. Les valeurs retournées par PAM sont converties en codes d’erreur AIX et retournées au programme appelant.

Figure 5. Chemin du service de sécurité AIX vers le module PAM Ce schéma montre le chemin emprunté par un appel de service de sécurité AIX lorsque PAM est configuré correctement. Les modules PAM indiqués (**pam_krb**, **pam_ldap** et **pam_dce**) sont des exemples de solutions tierces.



Le module de chargement PAM, installé dans le répertoire **/usr/lib/security**, est un module uniquement chargé de l'authentification. Le module PAM doit être associé à une base de données pour former un module de chargement composé. L'exemple suivant indique les strophes pouvant être ajoutées au fichier **methods.cfg** pour former un module PAM composé avec une base de données nommée **files**. Le mot clé **BUILTIN** pour l'attribut **db** désignera la base de données en tant que fichiers UNIX.

```
PAM :
         program = /usr/lib/security/PAM

PAMfiles:
         options = auth=PAM,db=BUILTIN
```

L'option **–R** permet alors de créer et modifier les utilisateurs à l'aide des commandes de gestion, et en définissant l'attribut **SYSTEM** lors de la création d'un utilisateur. Par exemple :

```
mkuser –R PAMfiles SYSTEM=PAMfiles registry=PAMfiles pamuser
```

Cette action permet d'indiquer aux appels suivants aux services de sécurité AIX (**login**, **passwd**, etc.) d'utiliser le module de chargement PAM pour l'authentification. Dans cet exemple, la base de données **files** (fichiers) a été utilisée pour le module composé, mais d'autres bases de données, comme LDAP, peuvent également être utilisées, si elles sont installées. La création d'utilisateurs définie précédemment entraînera le mappage suivant de la sécurité AIX en appels d'API PAM :

```
AIX                     PAM API
=====                   =========
authenticate       -->  pam_authenticate
chpass             -->  pam_chauthtok
passwdexpired      -->  pam_acct_mgmt
passwdrestrictions -->  No comparable mapping exists, success returned
```

La personnalisation du fichier **/etc/pam.conf** permet de diriger les appels d'API PAM vers le module PAM souhaité pour authentification. Il est possible d'implémenter la succession pour affiner encore le mécanisme d'authentification.

Les données appelées par un service de sécurité AIX sont transmises à PAM via la fonction **pam_set_item** car il n'est pas possible de remplir la boîte de dialogue depuis PAM. Les modules PAM écrits pour l'intégration avec le module PAM doivent récupérer toutes les données à l'aide d'appels **pam_get_item** mais ne doivent pas essayer de demander à l'utilisateur d'entrer des données, car le service de sécurité s'en charge.

La détection des boucles permet de repérer de possibles erreurs de configuration qui feraient que le service de sécurité AIX serait acheminé vers PAM, puis qu'un module PAM tenterait d'appeler le service de sécurité AIX pour exécuter l'opération. La détection de cette boucle entraînera l'échec immédiat de l'opération souhaitée.

Remarque : Le fichier **/etc/pam.conf** ne doit *pas* être configuré pour utiliser le module **pam_aix** lors de l'utilisation de l'intégration PAM depuis un service de sécurité AIX vers un module PAM car cela entraînerait la création d'une boucle.

Chapitre 9. Outils OpenSSH

Les outils OpenSSH prennent en charge les protocoles SSH1 et SSH2. Ils offrent des fonctions de shell là où le trafic réseau est chiffré et authentifié. OpenSSH s'appuie sur l'architecture client–serveur. OpenSSH exécute le démon **sshd** sur l'hôte AIX et attend la connexion des clients. Il assure l'authentification et le chiffrement des canaux avec les paires de clés publique et privée pour garantir des connexions réseau sécurisées et une authentification en fonction de l'hôte. Pour plus d'informations sur OpenSSH, y compris sur les pages man, consultez le site Internet suivant :

```
http://www.openssh.org
```

Cette section explique comment installer et configurer OpenSSH sous AIX.

OpenSSH est compris dans l'Expansion Pack AIX 5.3. Cette version de OpenSSH est compilée et regroupée en modules **installp** à l'aide du niveau de code source **openssh–3.8.1p1**. Les modules **installp** comprennent les pages man et les ensembles de fichiers de messages traduits. Le programme OpenSSH contenu sur le CD–ROM Expansion Pack est soumis aux conditions de licence régies par l'IPLA (International Program License Agreement) pour les programmes non garantis.

Avant d'installer les modules OpenSSH **installp**, vous devez installer le logiciel OpenSSL (Open Secure Sockets Layer) qui contient la bibliothèque chiffrée. OpenSSL est disponible en modules RPM sur le *CD AIX Toolbox for Linux Applications*.

Etant donné le contenu cryptographique du module OpenSSl, vous devez vous enregistrer sur le site Internet pour télécharger les modules. Procédez comme suit pour télécharger les modules :

1. Cliquez sur le lien **AIX Toolbox Cryptographic Content** (Contenu cryptographique Toolbox AIX) à droite de la page Internet *AIX Toolbox for Linux Applications* (Toolbox AIX pour applications Linux).
2. Cliquez sur **I have not registered before** (Je ne suis pas encore enregistré).
3. Remplissez les champs requis du formulaire.
4. Lisez les conditions de licence puis cliquez sur **Accept License** (Accepter la licence). Le navigateur vous redirige automatiquement vers l'espace de téléchargement.
5. Faites défiler votre curseur sur la liste de modules de contenu cryptographique jusqu'à l'entrée **openssl–0.9.6m–1.aix4.3.ppc.rpm** sous OpenSSL — SSL Cryptographic Libraries (OpenSSL — Bibliothèques cryptographiques SSL).
6. Cliquez sur le bouton **Download Now!** (Télécharger maintenant) pour **openssl–0.9.6m–1.aix4.3.ppc.rpm**.

Après avoir téléchargé le module OpenSSL, vous pouvez installer OpenSSL et OpenSSH.

1. Installez le module RPM OpenSSL à l'aide de la commande **geninstall** :

```
# geninstall -d/dev/cd0 R:openssl-0.9.6m
```

   Le résultat affiché sera semblable au suivant :

```
SUCCESSES
---------
openssl-0.9.6m-3
```

2. Installez ensuite les modules OpenSSH **installp** à l'aide de la commande **geninstall** :

```
# geninstall -I"Y" -d/dev/cd0 I:openssh.base
```

   Utilisez l'indicateur **Y** pour accepter l'accord de licence OpenSSH après l'avoir lu.

Le résultat affiché sera semblable au suivant :

```
Installation Summary
--------------------
Name                        Level           Part        Event       Result
-------------------------------------------------------------------------------
openssh.base.client         3.8.0.5200      USR         APPLY       SUCCESS
openssh.base.server         3.8.0.5200      USR         APPLY       SUCCESS
openssh.base.client         3.8.0.5200      ROOT        APPLY       SUCCESS
openssh.base.server         3.8.0.5200      ROOT        APPLY       SUCCESS
```

Vous pouvez aussi utiliser le raccourci SMIT **install_software** pour installer OpenSSL et OpenSSH.

La procédure précédente installe les fichiers binaires OpenSSH suivants :

scp	Programme de copie de fichiers similaire à **rcp**
sftp	Programme similaire à **FTP** utilisé par les protocoles SSH1 et SSH2
sftp–server	Sous–système serveur SFTP (lancé automatiquement par le démon **sshd**)
ssh	Similaire aux programmes client **rlogin** et **rsh**
ssh–add	Outil pour ajouter des clés à **ssh–agent**
ssh–agent	Agent pouvant stocker des clés privées
ssh–keygen	Outil de génération de clés
ssh–keyscan	Utilitaire de recueil de clés publiques depuis plusieurs hôtes
ssh–keysign	Utilitaire d'authentification en fonction de l'hôte
ssh–rand–helper	Programme utilisé par OpenSSH pour rassembler des numéros aléatoires. Ce programme est utilisé uniquement sur les postes avec AIX 5.1.
sshd	Démon de connexion

Les informations générales suivantes concernent OpenSSH :

- Le répertoire **/etc/ssh** contient le démon **sshd** et les fichiers de configuration pour la commande de client **ssh**.
- Le répertoire **/usr/openssh** contient le fichier readme et le fichier texte de licence open–source OpenSSH original. Le répertoire contient aussi le protocole **ssh** et le texte de licence Kerberos.
- Le démon **sshd** est contrôlé par AIX SRC. Les commandes suivantes permettent de lancer, arrêter et afficher le statut du démon :

```
startsrc –s sshd   OR startsrc –g ssh  (group)
stopsrc –s sshd    OU stopsrc –g ssh
lssrc –s sshd      OU lssrc –s ssh
```

Les commandes suivantes permettent également de lancer et d'arrêter le démon :

```
/etc/rc.d/rc2.d/Ksshd start
```

OU

```
/etc/rc.d/rc2.d/Sshd start
```

```
/etc/rc.d/rc2.d/Ksshd stop
```

OU

```
/etc/rc.d/rc2.d/Sshd stop
```

- Lorsque l'ensemble de fichiers du serveur OpenSSH est installé, une entrée est ajoutée au répertoire **/etc/rc.d/rc2.d**. Une entrée dans inittab permet d'exécuter des processus de niveau 2 (`l2:2:wait:/etc/rc.d/rc 2`), afin que le démon **sshd** démarre automatiquement à l'amorçage. Pour empêcher le démon de démarrer à l'amorçage, retirez les fichiers **/etc/rc.d/rc2.d/Ksshd** et **/etc/rc.d/rc2.d/Ssshd**.
- OpenSSH consigne des informations dans SYSLOG.
- OpenSSH prend en charge des noms d'utilisateur longs (256 octets) de la même manière que le système d'exploitation de base AIX. Pour plus d'informations sur les noms d'utilisateur longs, reportez–vous à la description de la commande **mkuser**.

Images OpenSSH

Procédez comme suit pour installer les images OpenSSH :

1. Décompressez le module d'images à l'aide de la commande **uncompress** *nom_module*. Par exemple :

```
uncompress openssh361p2_52_nologin.tar.Z
```

2. Décompressez le module tar à l'aide de la commande **tar –xvf** *nom_module*. Par exemple :

```
 tar –xvf openssh361p2_52_nologin.tar
```

3. Lancez **inutoc**.
4. Lancez **smitty install**.
5. Sélectionnez **Install and Update Software**
6. Sélectionnez **Update Installed Software to Latest Level (Update All)**
7. Entrez un point (.) dans la zone **INPUT device / directory for software** et appuyez sur Entrée.
8. Faites défiler la liste jusqu'à **ACCEPT new license agreements** et appuyez sur la touche Tab pour sélectionner **yes**.
9. Appuyez sur la touche Entrée deux fois pour lancer l'installation.

Les images OpenSSH sont des images de niveau de base, et non des PTF. Après l'installation, la totalité du code de la version précédente est remplacé par les images de la nouvelle version.

Configuration de compilation d'OpenSSH

Cette section contient des informations sur la manière dont le code OpenSSH est compilé pour AIX.

Lorsque vous configurez OpenSSH pour AIX 5.1, la sortie est similaire à ce qui suit :

```
OpenSSH has been configured with the following options:
                     User binaries: /usr/bin
                   System binaries: /usr/sbin
               Configuration files: /etc/ssh
                   Askpass program: /usr/sbin/ssh-askpass
                      Manual pages: /usr/man
                          PID file: /etc/ssh
   Privilege separation chroot path: /var/empty
           sshd default user PATH: /usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin

                    Manpage format: man
                       PAM support: no
                  KerberosIV support: no
                   KerberosV support: yes
                   Smartcard support: no
                         AFS support: no
                      S/KEY support: no
                TCP Wrappers support: no
                MD5 password support: no
        IP address in $DISPLAY hack: no
           Use IPv4 by default hack: no
            Translate v4 in v6 hack: no
                   BSD Auth support: no
               Random number source: ssh-rand-helper
      ssh-rand-helper collects from: Command hashing (timeout 200)

               Host: powerpc-ibm-aix5.1.0.0
           Compiler: cc
     Compiler flags: -O -D__STR31__
Preprocessor flags: -I. -I$(srcdir) -I/home/BUILD/test2debug/zlib-1.1.3/ -I/o
pt/freeware/src/packages/SOURCES/openssl-0.9.6m/include -I/usr/include -I/usr/in
clude/gssapi -I/usr/include/ibm_svc -I/usr/local/include $(PATHS) -DHAVE_CONFIG_H
       Linker flags: -L. -Lopenbsd-compat/ -L/opt/freeware/lib/ -L/usr/local/lib
-L/usr/krb5/lib -blibpath:/opt/freeware/lib:/usr/lib:/lib:/usr/local/lib:/usr/kr
b5/lib
          Libraries:   -lz  -lcrypto -lkrb5 -lk5crypto -lcom_err

WARNING: you are using the builtin random number collection
service. Please read WARNING.RNG and request that your OS
vendor includes kernel-based random number collection in
future versions of your OS.
```

Lorsque vous configurez OpenSSH pour AIX 5.2, la sortie est similaire à ce qui suit :

```
OpenSSH has been configured with the following options:
                     User binaries: /usr/bin
                   System binaries: /usr/sbin
               Configuration files: /etc/ssh
                   Askpass program: /usr/sbin/ssh-askpass
                      Manual pages: /usr/man
                          PID file: /etc/ssh
   Privilege separation chroot path: /var/empty
             sshd default user PATH:
/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin

                    Manpage format: man
                       PAM support: no
                  KerberosIV support: no
                   KerberosV support: yes
                   Smartcard support: no
                         AFS support: no
                      S/KEY support: no
                TCP Wrappers support: no
                MD5 password support: no
        IP address in $DISPLAY hack: no
           Use IPv4 by default hack: no
            Translate v4 in v6 hack: no
                   BSD Auth support: no
               Random number source: OpenSSL   internal ONLY

                  Host: powerpc-ibm-aix5.2.0.0
              Compiler: cc
     Compiler flags: -O -D__STR31__
     Preprocessor flags:
-I/opt/freeware/src/packages/BUILD/openssl-0.9.6m/include  -I/usr/local/include
-I/usr/local/include
       Linker flags: -L/opt/freeware/src/packages/BUILD/openssl-0.9.6m
-L/usr/lo
 cal/lib -L/usr/local/lib -blibpath:/usr/lib:/lib:/usr/local/lib:/usr/local/lib
          Libraries:   -lz  -lcrypto -lkrb5 -lk5crypto -lcom_err
```

Lorsque vous configurez OpenSSH pour AIX 5.3, la sortie est similaire à ce qui suit :

```
OpenSSH has been configured with the following options:
                     User binaries: /usr/bin
                   System binaries: /usr/sbin
               Configuration files: /etc/ssh
                   Askpass program: /usr/sbin/ssh-askpass
                      Manual pages: /usr/man
                          PID file: /etc/ssh
   Privilege separation chroot path: /var/empty
             sshd default user PATH: /usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin

                     Manpage format: man                         PAM support: no
                  KerberosIV support: no
                   KerberosV support: yes
                   Smartcard support: no
                         AFS support: no
                      S/KEY support: no
                TCP Wrappers support: no
                MD5 password support: no
        IP address in $DISPLAY hack: no
           Use IPv4 by default hack: no
            Translate v4 in v6 hack: no
                   BSD Auth support: no
              Random number source: OpenSSL   internal ONLY

              Host: powerpc-ibm-aix5.3.0.0
          Compiler: cc
     Compiler flags: -O -D__STR31__
     Preprocessor flags: -I/opt/freeware/src/packages/BUILD/openssl-0.9.6m/include
-I/usr/local/include -I/usr/local/include
       Linker flags: -L/opt/freeware/src/packages/BUILD/openssl-0.9.6m  -L/usr/lo
 cal/lib -L/usr/local/lib -blibpath:/usr/lib:/lib:/usr/local/lib:/usr/local/lib
          Libraries:   -lz  -lcrypto -lkrb5 -lk5crypto -lcom_err
```

OpenSSH et support Kerberos Version 5

Kerberos est un mécanisme d'authentification qui offre un procédé d'authentification sécurisé pour les utilisateurs réseau. Kerberos empêche la transmission de mots de passe en texte clair sur le réseau en cryptant les messages d'authentification entre les clients et les serveurs. De plus, Kerberos offre un système d'autorisation sous forme de jetons administratifs, ou données d'identification.

Pour authentifier un utilisateur utilisant Kerberos, l'utilisateur exécute la commande **kinit** afin d'obtenir les données d'identification initiales depuis un serveur central Kerberos appelé serveur KDC (Key Distribution Center). Le serveur KDC vérifie l'identité de l'utilisateur et transmet à l'utilisateur ses données d'identification, appelées TGT (tickets d'octroi d'autorisations). L'utilisateur peut alors lancer une session de connexion à distance à l'aide d'un service comme un service Telnet " kerbérisé " ou OpenSSH. Kerberos authentifie alors l'utilisateur en récupérant les données d'identification de l'utilisateur à partir du KDC. Kerberos effectue cette authentification sans intervention de la part de l'utilisateur. Par conséquent, l'utilisateur n'a pas besoin d'entrer son mot de passe pour se connecter. La version IBM de Kerberos est désignée par Service d'authentification réseau (Network Authentication Service ou NAS). Le service d'authentification réseau (NAS) peut être installé à partir des CD Expansion Pack AIX. Ce CD est disponible dans les modules **krb5.client.rte** et **krb5.server.rte**. Depuis la sortie d'OpenSSH 3.6 en juillet 2003, OpenSSH prend en charge l'authentification et l'autorisation Kerberos 5 via NAS version 1.3.

OpenSSH version 3.8 et versions ultérieures prennent en charge l'authentification et l'autorisation Kerberos 5 via NAS version 1.4. Toute migration de versions précédentes de NAS (Kerberos) doit être réalisée avant la mise à jour d'OpenSSH. OpenSSH version 3.8.x fonctionne uniquement avec NAS version 1.4 ou versions ultérieures.

AIX a créé OpenSSH avec l'authentification Kerberos comme méthode facultative. Si les bibliothèques Kerberos ne sont pas installées sur le système lors de l'exécution d'OpenSSH, l'authentification Kerberos est ignorée et OpenSSH tente la méthode d'authentification configurée suivante (telle que l'authentification AIX).

Une fois que vous avez installé Kerberos, il est recommandé de lire la documentation Kerberos avant de configurer les serveurs Kerberos. Pour plus d'informations sur la procédure d'installation et la gestion de Kerberos, reportez–vous à *Network Authentication Service Version 1.3 for AIX : Administrator's and User's Guide* dont le chemin d'accès est **/usr/lpp/krb5/doc/html/** *lang* **/ADMINGD.htm**.

Utilisation d'OpenSSH avec Kerberos

Les étapes suivantes contiennent des informations sur la configuration initiale requise pour utiliser OpenSSH avec Kerberos :

1. Le fichier **/etc/krb5.conf** doit être présent sur vos clients et serveurs OpenSSH. Ce fichier indique à Kerberos quel KDC utiliser, la durée de vie à attribuer à chaque ticket, etc. Voici un exemple de fichier **krb5.conf** :

```
[libdefaults]
ticket_lifetime = 600
default_realm = OPENSSH.AUSTIN.XYZ.COM
default_tkt_enctypes = des3-hmac-sha1 des-cbc-crc
default_tgs_enctypes = des3-hmac-sha1 des-cbc-crc

[realms]
OPENSSH.AUSTIN.xyz.COM = {
     kdc = kerberos.austin.xyz.com:88
     kdc = kerberos-1.austin.xyz.com:88
     kdc = kerberos-2.austin.xyz.com:88
     admin_server = kerberos.austin.xyz.com:749
     default_domain = austin.xyz.com
}

[domain_realm]
     .austin.xyz.com = OPENSSH.AUSTIN.XYZ.COM
     kdc.austin.xyz.com = OPENSSH.AUSTIN.XYZ.COM
```

2. Vous devez également ajouter les services Kerberos suivants dans le fichier **/etc/services** de chaque poste client :

```
kerberos      88/udp    kdc    # Kerberos V5 KDC
kerberos      88/tcp    kdc    # Kerberos V5 KDC
kerberos-adm  749/tcp          # Kerberos 5 admin/changepw
kerberos-adm  749/udp          # Kerberos 5 admin/changepw
krb5_prop     754/tcp          # Kerberos slave
                                #  propagation
```

3. Si votre KDC utilise LDAP comme registre pour stocker les informations utilisateur, il est recommandé de lire la section Module de chargement d'authentification LDAP, page 5-1, ainsi que les publications de Kerberos. De plus, assurez–vous que les actions suivantes sont effectuées :
   - KDC exécute le client LDAP. Vous pouvez lancer le démon client LDAP avec la commande **secldapclntd**.
   - Le serveur LDAP exécute le démon serveur LDAP **slapd**.
4. Sur le serveur OpenSSH, modifiez le fichier **/etc/ssh/sshd_config** pour qu'il contienne les lignes :

```
KerberosAuthentication yes
KerberosTicketCleanup yes
GssapiAuthentication yes
GssapiKeyExchange yes
GssapiCleanupCreds yes
UseDNS yes
```

Si la valeur de **UseDNS** est `yes`, le serveur ssh effectue une recherche d'hôte inversée pour trouver le nom du client qui s'est connecté. Cela est nécessaire dans le cas d'une authentification en fonction de l'hôte ou si vous souhaitez afficher dans les dernières informations de connexion le nom d'hôte au lieu de l'adresse IP.

Remarque : Certaines sessions ssh se bloquent lors de la recherche inversée de nom, car le serveur DNS est inaccessible. Dans ce cas, vous pouvez passer la recherche DNS en attribuant à **UseDNS** la valeur `no`. Si **UseDNS** n'est pas explicitement défini dans le fichier **/etc/ssh/sshd_config**, la valeur par défaut est `UseDNS yes`.

5. Sur le serveur SSH, exécutez la commande **startsrc –g ssh** pour lancer le démon serveur ssh.
6. Sur le poste client SSH, exécutez la commande **kinit** pour obtenir les données d'identification initiales (TGT ou ticket d'octroi d'autorisations). Vous pouvez vérifier que vous avez reçu un TGT en exécutant la commande **klist**. Ceci montre que toutes les données d'identification vous appartiennent.
7. Connectez–vous au serveur en exécutant la commande **ssh** *nomutilisateur@nomserveur*.
8. Si Kerberos est correctement configuré pour authentifier l'utilisateur, aucune fenêtre d'invite pour la saisie d'un mot de passe ne s'affiche et l'utilisateur est automatiquement connecté au serveur SSH.

Deuxième partie. Sécurité réseau et Internet

La deuxième partie du présent manuel fournit des informations relatives aux mesures de sécurité réseau et Internet. Ces chapitres décrivent les procédures d'installation et de configuration de la sécurité IP, la procédure d'identification des services réseau obligatoires et facultatifs, l'audit et le contrôle de la sécurité réseau, etc.

Chapitre 10. Sécurité TCP/IP

Si vous avez installé TCP/IP (Transmission Control Protocol/Internet Protocol) et NFS (Network File System), vous pouvez configurer votre système afin de communiquer via un réseau. Ce guide ne décrit pas les concepts TCP/IP de base, mais des sujets liés à la sécurité dans TCP/IP. Pour des informations sur l'installation et la configuration initiale de TCP/IP, consultez le chapitre Transmission Control Protocol/Internet Protocol (TCP/IP) du manuel *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Pour diverses raisons, l'administrateur système doit assurer un certain niveau de protection. Le niveau de sécurité peut relever de la politique d'entreprise. Un système peut aussi devoir accéder à des systèmes publics, et de ce fait doit être étroitement contrôlé. Ces niveaux de sécurité peuvent s'appliquer au réseau, au système d'exploitation, aux applications, et même aux programmes développés par l'administrateur.

Ce chapitre décrit le dispositif de sécurité fourni avec TCP/IP, en mode standard et sécurisé, et développe certaines notions de sécurité propres à l'environnement réseau.

Une fois TCP/IP et NFS installés, utilisez Web-based System Manager ou le raccourci SMIT **tcpip** pour configurer le système.

Ce chapitre traite des points suivants :

Système de protection du système d'exploitation

La plupart des fonctions de protection proposées pour TCP/IP sont calquées sur celles du système d'exploitation. En voici les grandes lignes.

Contrôle d'accès au réseau

Le dispositif de sécurité appliqué au réseau prolonge celui du système d'exploitation :

- L'**authentification de l'utilisateur** s'opère au niveau de l'hôte distant via un nom d'utilisateur un mot de passe et (tout comme lors de la connexion au système local). Les commandes TCP/IP sécurisées, telles que **ftp**, **rexec** et **telnet**, subissent les mêmes contraintes et contrôles que celles du systèmes d'exploitation.
- L'**authentification de connexion** vise à contrôler l'identité et l'adresse IP de l'hôte distant. Ainsi, tout risque d'usurpation d'identité par un hôte distant est évité.
- La **protection des échanges** permet d'importer/exporter des données à un niveau de sécurité spécifique, entre des cartes réseau dotées de droits et de protections identiques. Par exemple des données confidentielles ne peuvent circuler qu'entre cartes du niveau de protection correspondant.

Audit de réseau

L'audit de réseau est réalisé par TCP/IP via le sous–système d'audit qui s'applique aux routines de réseau noyau et aux applications. Il consigne toutes les actions relatives à la sécurité et à l'utilisateur qui les effectue.

L'audit s'applique aux événement suivants :

Evénements au niveau du noyau

- Changement de configuration
- Changement d'ID hôte
- Changement de route
- Connexion
- Création d'une prise (socket)
- Exportation d'objets
- Importation d'objets

Evénements au niveau application

- Accès au réseau
- Changement de configuration
- Changement d'ID hôte
- Changement de route statique
- Configuration du courrier
- Connexion
- Exportation de données
- Importation de données
- Ecriture de courrier dans un fichier

Toute création et suppression d'objets subit un audit de la part du système d'exploitation. Les enregistrements d'audit au niveau application interrompent et relancent l'audit pour éviter leur enregistrement par l'audit du noyau.

Chemin d’accès sécurisé, shell sécurisé et clé SAK

Le système d’exploitation prévoit un *chemin d’accès sécurisé* pour empêcher tout programme non autorisé de lire des données à partir d’un terminal utilisateur. Ce chemin est utilisé pour les communications confidentielles avec le système (par exemple, pour la modification de mots de passe ou l’entrée en communication). Un *shell sécurisé* (**tsh**) est également proposé, qui n’exécute que les programmes sécurisés, testés et contrôlés comme tels. TCP/IP prend tous ces dispositifs en charge, de même que la clé SAK (*Secure Attention Key*) dont le rôle est de mettre en place l’environnement pour une communication sécurisée entre vous et le système. La clé SAK locale est accessible dès l’utilisation de TCP/IP. Par ailleurs, la commande **telnet** donne accès à une clé SAK distante.

La clé SAK locale offre les mêmes fonctions sous **telnet** et sous d’autres programmes du système : elle met fin au processus **telnet** et à tout autre processus associé au terminal qui exécutait **telnet**. Toutefois, sous **telnet**, vous pouvez envoyer une demande de chemin d’accès sécurisé au système distant via la commande **telnet send sak** (en mode commande **telnet**). Vous pouvez aussi définir une seule clé pour l’émission d’une requête SAK à l’aide de la commande **telnet set sak**.

Pour plus d’informations sur la base TCB, consultez le chapitre La base TCB, page 1-2.

Sécurité des commandes TCP/IP

Certaines commandes TCP/IP ont pour but de fournir un environnement sécurisé durant le fonctionnement. Il s’agit de **ftp**, **rexec** et **telnet**. **ftp** concerne les transferts de données. **rexec** s’applique à l’exécution des commandes sur un hôte étranger. **telnet** a trait à la connexion sur un hôte étranger.

Les commandes **ftp**, **rexec** et **telnet** n’assurent la sécurité que pendant leur exécution. C’est–à–dire qu’elles ne définissent pas d’environnement sécurisé pour l’exécution d’autres commandes. Pour protéger votre système lors de l’exécution d’autres opérations, faites appel à la commande **securetcpip**. Cette commande permet de protéger le système en désactivant les applications et démons non sécurisés et en vous permettant d’activer la sécurisation de votre protocole IP.

Les commandes **ftp**, **rexec**, **securetcpip** et **telnet** fournissent les garanties suivantes :

ftp

La commande **ftp** fournit un environnement sécurisé pour le transfert de fichiers. Lorsqu'un utilisateur lance la commande **ftp** vers un hôte étranger, il est invité à fournir un ID de connexion. Un ID de connexion par défaut s'affiche : l'ID de connexion en cours de l'utilisateur sur l'hôte local. L'utilisateur doit fournir un mot de passe pour l'hôte distant.

Le processus de connexion automatique recherche, dans le fichier **$HOME/.netrc** de l'utilisateur local, l'ID et le mot de passe à soumettre à l'hôte étranger. Pour plus de sécurité, les droits d'accès au fichier **$HOME/.netrc** doivent être fixés à 600 (lecture et écriture réservées au propriétaire). A défaut, la connexion automatique échoue.

Remarque : Le fichier **.netrc** impose de stocker les mots de passe dans un fichier non chiffré. C'est pourquoi la connexion automatique par **ftp** n'est pas disponible si le système est configuré avec **securetcpip**. Pour la réactiver, supprimez la commande **ftp** de la strophe tcpip du fichier **/etc/security/config**.

Le transfert de fichiers via **ftp** suppose deux connexions TCP/IP : une pour le protocole et une pour le transfert des données. La connexion au protocole, principale, est une connexion fiable car établie sur des ports de communication fiables. La connexion secondaire, dédiée au transfert des données proprement dit, doit être établie sur les mêmes hôtes local et distant que la première (condition vérifiée sur chacun des hôtes). Faute de quoi, la commande **ftp** émet un message d'erreur indiquant que la connexion n'a pas été authentifiée, puis s'arrête. Ce contrôle vise à éviter qu'un hôte tiers n'intercepte des données qui ne lui sont pas destinées.

rexec

La commande **rexec** s'applique à l'exécution des commandes sur un hôte étranger. L'utilisateur est invité à décliner son ID de connexion et son mot de passe.

Avec le dispositif de connexion automatique, la commande **rexec** recherche, dans le fichier **$HOME/.netrc** de l'utilisateur local, l'ID et le mot de passe à soumettre à l'hôte étranger. Pour plus de sécurité, les droits d'accès au fichier **$HOME/.netrc** doivent être fixés à 600 (lecture et écriture réservées au propriétaire). A défaut, la connexion automatique échoue.

Remarque : Le fichier **.netrc** impose de stocker les mots de passe dans un fichier non chiffré. C'est pourquoi la connexion automatique par **rexec** n'est pas disponible si le système est exploité en mode sécurisé. Pour la réactiver, supprimez l'entrée **rexec** de la strophe tcpip du fichier **/etc/security/config**.

securetcpip

La commande **securetcpip** active le système de protection de TCP/IP. L'accès aux commandes non sécurisées est supprimé du système à l'émission de cette commande. Les commandes suivantes sont supprimées par l'exécution de la commande **securetcpip** :

- **rlogin** et **rlogind**
- **rcp**, **rsh** et **rshd**
- **tftp** et **tftpd**
- **trpt**

La commande **securetcpip** fait passer la sécurité du d'un niveau standard au niveau de maximal. Dès lors, vous n'aurez à relancer **securetcpip** que si vous réinstallez TCP/IP.

telnet ou **tn**

La commande **telnet** (TELNET) fournit un environnement sécurisé à la connexion sur un hôte étranger. L'utilisateur est invité à décliner son ID de connexion et son mot de passe. Le terminal de l'utilisateur est considéré comme directement connecté à l'hôte : l'accès au terminal est contrôlé par des bits d'autorisation. Les autres utilisateurs (groupe et autres) n'ont pas accès en lecture au terminal, mais ils peuvent y écrire des messages si le propriétaire les y autorise. La commande **telnet** donne également accès au shell sécurisé du système distant via la clé SAK (Secure Attention Key). Cette clé, qui peut être définie par la commande **telnet**, doit être différente de celle utilisée pour appeler le chemin d'accès sécurisé local.

Exécution de commandes à distance (/etc/hosts.equiv)

Les utilisateurs répertoriés dans le fichier **/etc/hosts.equiv** peuvent exécuter certaines commandes sur votre système sans fournir de mot de passe. Le tableau suivant fournit des informations sur le classement, l'ajout et la suppression d'hôtes distants à l'aide de Web–based System Manager, SMIT, ou de la ligne de commandes.

Tâches d'accès à distance aux commandes

Tâche	**Raccourci SMIT**	**Commande ou fichier**	Web-based System Manager Management Environment
Afficher la liste des hôtes qui peuvent accéder aux commandes	**smit lshostsequiv**	affichez **/etc/hosts.equiv**	Software —> **Network** —> **TCPIP (IPv4 and IPv6)** —> **TCPIP Protocol Configuration** —> **TCP/IP** —> **Configure TCP/IP** —> **Advanced Methods** —> **Hosts File** —> **Contents of /etc/hosts file**.
Ajouter un hôte distant autorisé à exécuter les commandes	**smit mkhostsequiv**	modifiez **/etc/hosts.equiv** Remarque 1	Software —> **Network** —> **TCPIP (IPv4 and IPv6)** —> **TCPIP Protocol Configuration** —> **TCP/IP** —> **Configure TCP/IP** —> **Advanced Methods** —> **Hosts File**. Dans **Add/Change host entry**, complétez les zones suivantes : **IP Address**, **Host name**, **Alias(es)** et **Comment**. Cliquez sur **Add/Change Entry**, puis cliquez sur **OK**.
Supprimer un hôte distant	**smit rmhostsequiv**	modifiez **/etc/hosts.equiv** Remarque 1	Software —> **Network** —> **TCPIP (IPv4 and IPv6)** —> **TCPIP Protocol Configuration** —> **TCP/IP** —> **Configure TCP/IP** —> **Advanced Methods** —> **Hosts File**. Sélectionnez un hôte dans **Contents of /etc/host file**. Cliquez sur **Delete Entry** —> **OK**.

Remarque : Pour plus d'informations sur ces procédures de fichiers, reportez–vous à la section "hosts.equiv File Format for TCP/IP" dans le manuel *AIX 5L Version 5.3 Files Reference*.

Restrictions d'accès FTP (/etc/ftpusers)

Les utilisateurs répertoriés dans le fichier **/etc/ftpusers** sont protégés contre l'accès FTP à distance. Supposons que l'utilisateur A connecté à un système distant connaît le mot de passe de l'utilisateur B sur votre système. Si l'utilisateur B apparaît dans le fichier **/etc/ftpusers** , l'utilisateur A ne peut transmettre de fichiers par FTP vers ou depuis le compte de l'utilisateur B, bien qu'il connaisse son mot de passe.

Le tableau suivant fournit des informations sur le classement, l'ajout et la suppression d'utilisateurs restreints à l'aide de Web–based System Manager, SMIT, ou de la ligne de commandes.

Opérations relatives aux utilisateurs protégés

Tâche	**Raccourci SMIT**	**Commande ou fichier**	Web-based System Manager Management Environment
Afficher la liste	**smit lsftpusers**	affichez **/etc/ftpusers**	Software —> **Users** —> **All Users**.
Ajouter un utilisateur	**smit mkftpusers**	modifiez **/etc/ftpusers** Remarque 1	Software —> **Users** —> **All Users** —> **Selected** —> **Add this User to Group**. Sélectionnez un groupe, puis cliquez sur **OK**.
Supprimer un utilisateur	**smit rmftpusers**	modifiez **/etc/ftpusers** Remarque 1	Software —> **Users** —> **All Users** —> **Selected** —> **Delete**.

Remarque : Pour plus d'informations sur ces procédures de fichiers, reportez–vous à la section "ftpusers File Format for TCP/IP" dans le manuel *AIX 5L Version 5.3 Files Reference*.

Processus sécurisés

Un processus (ou programme) sécurisé est un script shell, un démon ou un programme conforme aux normes de sécurité établies et révisées par des organismes agrées (aux USA, le ministère de la défense), qui certifient également certains programmes sécurisés.

A ces programmes sont associés différents niveaux de sécurité : A1, B1, B2, B3, C1, C2 et D (A1 étant le niveau maximal) satisfaisant chacun à des critères spécifiques. Par exemple, le niveau C2 intègre les aspects suivants :

intégrité des programmes	Assure le bon fonctionnement du processus.
modularité	Le code des processus est divisé en modules qui ne peuvent pas être directement affectés ou accédés par d'autres modules.
principe du moindre privilège	Les activités utilisateur se déroulent toujours au niveau de privilège le plus faible : un utilisateur habilité à lire un fichier ne peut le modifier par inadvertance.
limitation de la réutilisation d'objets	Un utilisateur ne peut pas, par exemple, trouver une section de mémoire marqué pour écrasement mais non encore effacée, et qui peut contenir des informations importantes.

TCP/IP contient quelques démons sécurisés et de nombreux démons non sécurisés.

On trouve parmi les démons sécurisés :

- **ftpd**
- **rexecd**
- **telnetd**

On trouve parmi les démons non sécurisés :

- **rshd**
- **rlogind**
- **tftpd**

Pour qu'un système soit sécurisé, il doit fonctionner avec une base informatique sécurisée, c'est à dire que pour un hôte unique, le poste doit être sécurisé. Sur un réseau, tous les serveurs de fichiers, passerelles et autres hôtes doivent être sécurisés.

Base NTCB

La base NTCB (Network Trusted Computing Base) associe logiciels et matériels pour protéger le réseau. Cette section définit les différents composants de la base NTCB en relation avec TCP/IP.

Les dispositifs matériels sont fournis par les cartes réseau utilisées avec TCP/IP. Ces cartes contrôlent les données entrantes en ne recevant que des données destinées au local system et diffusent les données pouvant être reçues par tous les systèmes.

Le module logiciel de NTCB est constitué exclusivement de programmes sécurisés. Les programmes et fichiers associés sont indiqués ci–dessous (par répertoires).

Répertoire /etc

Nom	Propriétaire	Groupe	Mode	Droits
gated.conf	root	system	0664	rw–rw–r—
gateways	root	system	0664	rw–rw–r—
hosts	root	system	0664	rw–rw–r—
hosts.equiv	root	system	0664	rw–rw–r—
inetd.conf	root	system	0644	rw–r—r—
named.conf	root	system	0644	rw–r—r—
named.data	root	system	0664	rw–rw–r—
networks	root	system	0664	rw–rw–r—
protocols	root	system	0644	rw–r—r—
rc.tcpip	root	system	0774	rwxrwxr—
resolv.conf	root	system	0644	rw–rw–r—
services	root	system	0644	rw–r—r—
3270.keys	root	system	0664	rw–rw–r—
3270keys.rt	root	system	0664	rw–rw–r—

Répertoire /usr/bin

Nom	Propriétaire	Groupe	Mode	Droits
host	root	system	4555	r–sr–xr–x
hostid	bin	bin	0555	r–xr–xr–x
hostname	bin	bin	0555	r–xr–xr–x
finger	root	system	0755	rwxr–xr–x
ftp	root	system	4555	r–sr–xr–x
netstat	root	bin	4555	r–sr–xr–x
rexec	root	bin	4555	r–sr–xr–x
ruptime	root	system	4555	r–sr–xr–x
rwho	root	system	4555	r–sr–xr–x
talk	bin	bin	0555	r–xr–xr–x
telnet	root	system	4555	r–sr–xr–x

Répertoire /usr/sbin

Nom	Propriétaire	Groupe	Mode	Droits
arp	root	system	4555	r–sr–xr–x
fingerd	root	system	0554	r–xr–xr––
ftpd	root	system	4554	r–sr–xr––
gated	root	system	4554	r–sr–xr––
ifconfig	bin	bin	0555	r–xr–xr–x
inetd	root	system	4554	r–sr–xr––
named	root	system	4554	r–sr–x––
ping	root	system	4555	r–sr–xr–x
rexecd	root	system	4554	r–sr–xr––
route	root	system	4554	r–sr–xr––
routed	root	system	0554	r–xr–x–––
rwhod	root	system	4554	r–sr–xr––
securetcpip	root	system	0554	r–xr–xr––
setclock	root	system	4555	r–sr–xr–x
syslogd	root	system	0554	r–xr–xr––
talkd	root	system	4554	r–sr–xr––
telnetd	root	system	4554	r–sr–xr––

Répertoire /usr/ucb

Nom	Propriétaire	Groupe	Mode	Droits
tn	root	system	4555	r–sr–xr–x

Répertoire /var/spool/rwho

Nom	Propriétaire	Groupe	Mode	Droits
rwho (répertoire)	root	system	0755	drwxr–xr–x

Sécurité des données et protection des informations

Le dispositif de sécurité sous TCP/IP ne chiffre pas les données transmises par le réseau. Il est donc recommandé de prendre des mesures pour prévenir tout risque de défaillance du système de sécurité pouvant révéler des mots de passe ou des informations confidentielles.

L'utilisation de la fonction de sécurité TCP/IP dans un environnement relevant du ministère de la défense (Department of Defense DOD aux États–Unis) requiert la conformité aux normes de sécurité DOD 5200.5 et NCSD–11.

Contrôle d'accès aux ports TCP en fonction de l'utilisateur, avec le contrôle d'accès discrétionnaire aux ports Internet

L'accès discrétionnaire aux ports Internet (DACinet) permet le contrôle de l'accès aux ports TCP pour les communications entre hôtes AIX 5.2. AIX 5.2 peut utiliser un en–tête TCP supplémentaire pour transporter les informations sur les utilisateurs et les groupes. DACinet permet à l'administrateur du système de destination de contrôler l'accès en fonction du port de destination, de l'ID utilisateur d'origine et de l'hôte.

La fonction DACinet lui permet aussi de réserver les ports locaux à l'utilisateur root. Les systèmes UNIX comme AIX traitent les ports en dessous de 1024 comme des ports privilégiés qui ne peuvent être ouverts que par l'utilisateur root. AIX 5.2 permet d'y ajouter des ports au–dessus de 1024, ce qui empêche les autres utilisateurs d'exécuter des serveurs sur des ports connus.

Selon les paramètres, un système non DACinet peut ou non se connecter à un système DACinet. L'accès est refusé dans l'état initial de la fonction DACinet. Une fois la fonction DACinet activée, il est impossible de la désactiver.

La commande **dacinet** accepte des adresses spécifiés sous forme de noms d'hôtes, adresses d'hôtes en notation décimale à point, ou adresses réseau suivies de la longueur du préfixe réseau.

L'exemple suivant spécifie un hôte unique par son nom d'hôte complet *host.domain.org*:

```
host.domain.org
```

L'exemple suivant spécifie un hôte unique par son adresse IP 10.0.0.1 :

```
10.0.0.1
```

L'exemple suivant spécifie le réseau entier dont la valeur des 24 premiers bits (la longueur du préfixe de réseau) est 10.0.0.0 :

```
10.0.0.0/24
```

Ce réseau comprend toutes les adresses IP entre 10.0.0.1 et 10.0.0.254.

Contrôle des accès aux services TCP

DACinet utilise le fichier de démarrage **/etc/rc.dacinet** et les fichiers de configuration **/etc/security/priv**, **/etc/security/services** et **/etc/security/acl**.

Les ports répertoriés dans le fichier **/etc/security/services** ne subissent pas de contrôles ACL Le fichier a le même format que **/etc/services**. La façon la plus facile de l'initialiser est de le copier depuis **/etc** vers **/etc/security** puis de supprimer tous les ports pour lesquels des ACL sont à appliquer. Les ACL sont stockés à deux endroits. Les ACL actives sont stockées dans le noyau et peuvent être lues à l'aide de la commande **dacinet aclls**. Les ACL qui seront réactivées au prochain démarrage par **/etc/rc.tcpip** sont stockées dans **/etc/security/acl**. Le format suivant est utilisé :

```
service host/prefix-length [user|group]
```

Où le service peut être spécifié numériquement ou comme dans la liste dans **/etc/services**, l'hôte peut recevoir un nom d'hôte ou une adresse réseau avec un masque de sous–réseau et l'utilisateur ou le groupe est indiqué à l'aide du préfixe **u:** ou **g:**. En l'absence d'indication d'utilisateur ou de groupe, l'ACL ne prend en compte que l'hôte émetteur. Le préfixe **–** désactive explicitement l'accès au service. Les ACL sont évaluées dans l'ordre de leur mention. Vous pouvez donc spécifier l'accès pour un groupe d'utilisateurs, mais le refuser explicitement pour un utilisateur du groupe en plaçant la règle de cet utilisateur devant la règle du groupe.

Le fichier **/etc/services** comprend deux entrées avec des numéros de ports non pris en charge par AIX 5.2. L'administrateur système doit retirer ces deux lignes du fichier avant d'exécuter la commande **mkCCadmin**. Supprimez les lignes suivantes du fichier **/etc/services** .

```
sco_printer     70000/tcp     sco_spooler    # For System V print IPC
 sco_s5_port     70001/tcp     lpNet_s5_port  # For future use
```

Exemples d'utilisation de DACinet

Lorsque vous utilisez DACinet pour réserver l'accès au port entrant TCP/25 aux utilisateurs root, seuls les utilisateurs root des autres hôtes AIX 5.2 peuvent y accéder, ce qui limite les possibilités des autres utilisateurs d'usurper des courriers électroniques par une commande telnet sur le port TCP/25 de la victime. L'exemple suivant indique comment configurer le protocole X (X11) pour un accès root uniquement. Vérifiez que l'entrée X11 est retirée de **/etc/security/services**, de sorte que les ACL s'appliqueront à ce service.

Par exemple, pour un sous–réseau 10.1.1.0/24 pour tous les systèmes connectés, les entrées ACL pour limiter l'accès aux utilisateurs root uniquement pour X (TCP/6000) dans **/etc/security/acl** seraient :

```
      6000    10.1.1.0/24 u:root
```

Pour limiter le service Telnet aux utilisateurs du groupe **friends**, quel que soit leur système d'origine, utilisez l'ACL suivante après avoir supprimé l'entrée telnet de **/etc/security/services**:

```
telnet    0.0.0.0/0   g:friends
```

Pour empêcher l'utilisateur fred d'accéder au serveur web, mais autoriser tous les autres à y accéder :

```
-80    0.0.0.0/0 u:fred
 80     0.0.0.0/0
```

Ports privilégiés pour l'exécution des services locaux

Normalement, tout utilisateur peut ouvrir chaque port au–dessus du 1024. Par exemple, un utilisateur pourrait placer un serveur sur le port 8080, souvent utilisé pour l'exécution d'une proxy Web, ou sur le 1080, qui héberge généralement un serveur SOCKS. Certains ports peuvent être désignés comme privilégiés afin d'éviter que les tous les utilisateurs puissent y exécuter des serveurs. La commande **dacinet setpriv** permet d'ajouter des ports privilégiés au système exécuté. Les ports désignés en tant que privilégiés au démarrage du système doivent être répertoriés dans le fichier **/etc/security/priv**

Les ports peuvent être placés dans ce fichier sous leur nom symbolique défini dans **/etc/services**, ou en indiquant leur numéro. Les entrées suivantes empêchent les utilisateurs autres que root d'exécuter des serveurs SOCKS ou Lotus Notes sur leurs ports habituels :

```
1080
 lotusnote
```

Remarque : Cette fonction n'empêche pas l'**utilisateur** d'exécuter les programmes. Elle l'empêche seulement d'exécuter les services sur les ports connus où ils sont généralement placés.

Pour plus d'informations sur la commande **dacinet**, consultez le manuel *AIX 5L Version 5.3 Commands Reference*.

Chapitre 11. Services réseau

Ce chapitre apporte des informations sur l'identification et la sécurisation des services réseau avec des ports de communication ouverts.

Correspondance des Services réseau avec les ports de communication ouverts

Les applications client–serveur ouvrent des ports de communication sur le serveur, afin que les applications puissent écouter les requêtes entrantes des clients. Les ports ouverts étant vulnérables aux attaques potentielles, identifiez les applications qui ont des ports ouverts et fermez les ports qui n'ont pas besoin de rester ouverts. Vous pourrez ainsi comprendre quels systèmes sont rendus disponibles à toute personne ayant accès à Internet.

La procédure suivante identifie les ports ouverts :

1. Identifiez les services à l'aide de la commande **netstat** :

   ```
   # netstat –af inet
   ```

   Voici un exemple d'utilisation de cette commande. La dernière colonne indique l'état de chaque service. Les services qui attendent les connexions entrantes sont en état ECOUTE.

Connexion Internet active (y compris serveurs)

Proto	File de réception	File d'émission	Adresse locale	Adresse étrangère	(état)
tcp4	0	0	*.echo	*.*	LISTEN
tcp4	0	0	*.discard	*.*	LISTEN
tcp4	0	0	*.daytime	*.*	LISTEN
tcp	0	0	*.chargen	*.*	LISTEN
tcp	0	0	*.ftp	*.*	LISTEN
tcp4	0	0	*.telnet	*.*	LISTEN
tcp4	0	0	*.smtp	*.*	LISTEN
tcp4	0	0	*.time	*.*	LISTEN
tcp4	0	0	*.www	*.*	LISTEN
tcp4	0	0	*.sunrpc	*.*	LISTEN
tcp	0	0	*.smux	*.*	LISTEN
tcp	0	0	*.exec	*.*	LISTEN
tcp	0	0	*.login	*.*	LISTEN
tcp4	0	0	*.shell	*.*	LISTEN
tcp4	0	0	*.klogin	*.*	LISTEN
udp4	0	0	*.kshell	*.*	LISTEN
udp4	0	0	*.echo	*.*	
udp4	0	0	*.discard	*.*	
udp4	0	0	*.daytime	*.*	
udp4	0	0	*.chargen	*.*	
udp4	0	0	*.time	*.*	

Connexion Internet active (y compris serveurs)

Proto	File de réception	File d'émission	Adresse locale	Adresse étrangère	(état)
udp4	0	0	*.bootpc	*.*	
udp4	0	0	*.sunrpc	*.*	
udp4	0	0	255.255.255.255.ntp	*.*	
udp4	0	0	1.23.123.234.ntp	*.*	
udp4	0	0	localhost.domain.ntp	*.*	
udp4	0	0	name.domain..ntp	*.*	

.....

2. Ouvrez le fichier **/etc/services** et contrôlez les services IANA (Internet Assigned Numbers Authority) pour faire correspondre le service aux numéros de ports du système d'exploitation.

   Voici une partie du fichier **/etc/services** :

```
tcpmux                  1/tcp           # TCP Port Service Multiplexer
tcpmux                  1/tcp           # TCP Port Service Multiplexer
Compressnet             2/tcp           # Management Utility
Compressnet             2/udp           # Management Utility
Compressnet             3/tcp           # Compression Process
Compressnet             3/udp           Compression Process
Echo                    7/tcp
Echo                    7/udp
discard                 9/tcp           sink null
discard                 9/udp           sink null
..............
rfe                     5002/tcp        # Radio Free Ethernet
rfe                     5002/udp        # Radio Free Ethernet
rmonitor_secure         5145/tcp
rmonitor_secure         5145/udp
pad12sim                5236/tcp
pad12sim                5236/udp
sub-process             6111/tcp        # HP SoftBench Sub-Process Cntl.
sub-process             6111/udp        # HP SoftBench Sub-Process Cntl.
xdsxdm                  6558/ucp
xdsxdm                  6558/tcp
afs3-fileserver         7000/tcp        # File Server Itself
afs3-fileserver         7000/udp        # File Server Itself
af3-callback            7001/tcp        # Callbacks to Cache Managers
af3-callback            7001/udp        # Callbacks to Cache Managers
```

3. Fermez les ports non nécessaires en supprimant les services en cours.

Remarque : Le port 657 est utilisé par la fonction de Surveillance et contrôle des ressources (RMC) pour assurer la communication entre les noeuds. Vous ne pouvez ni bloquer ni restreindre ce port.

Identification des sockets TCP et UDP

Identifiez les sockets TCP à l'état LISTEN et les sockets UDP inactifs (idle) en attente de données. Utilisez la commande **lsof**, une variante de la commande **netstat –af**. A partir d'AIX 5.1, la commande **lsof** est sur le CD *AIX Toolbox for Linux Applications*.

Par exemple, pour afficher les sockets TCP à l'état LISTEN et les sockets UDP IDLE, lancez la commande **lsof** :

```
# lsof –i | egrep "COMMAND|LISTEN|UDP"
```

Le résultat de cette commande se présente comme suit :

Commande	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
dtlogin	2122	root	5u	IPv4	0x70053c00	0t0	UDP	*:xdmcp
dtlogin	2122	root	6u	IPv4	0x70054adc	0t0	TCP	*:32768(LISTEN)
syslogd	2730	root	4u	IPv4	0x70053600	0t0	UDP	*:syslog
X	2880	root	6u	IPv4	0x70054adc	0t0	TCP	*:32768(LISTEN)
X	2880	root	8u	IPv4	0x700546dc	0t0	TCP	*:6000(LISTEN)
dtlogin	3882	root	6u	IPv4	0x70054adc	0t0	TCP	*:32768(LISTEN)
glbd	4154	root	4u	IPv4	0x7003f300	0t0	UDP	*:32803
glbd	4154	root	9u	IPv4	0x7003f700	0t0	UDP	*:32805
dtgreet	4656	root	6u	IPv4	0x70054adc	0t0	TCP	*:32768(LISTEN)

Une fois l'ID de processus identifié, vous pouvez obtenir plus d'informations sur le programme à l'aide de la commande suivante :

```
" # ps –fp   PID#  "
```

Le résultat contient le chemin vers le nom de la commande, que vous pouvez utiliser pour accéder à la page man du programme.

Utilisation du port RMC

Chapitre 12. Sécurité IP (Internet Protocol)

Le protocole de sécurité IP permet de sécuriser les communications sur le réseau Internet et les réseaux d'entreprise, en protégeant le flux de données au niveau de la couche IP. Il permet de protéger l'échange de données pour toutes les applications, sans avoir à les modifier. Il sécurise ainsi la transmission de tout type de données, par exemple de messagerie électronique ou d'applications.

Ce chapitre traite des points suivants :

Sécurité IP – Généralités

Cette section traite des points suivants :

Sécurité IP et système d'exploitation

Le système d'exploitation utilise la sécurité IP (IPsec), un standard ouvert développé par l'IETF (Internet Engineering Task Force). IPsec assure la protection par le chiffrement de toutes les données au niveau de la couche de communications IP. Aucune modification des applications n'est nécessaire. IPsec est l'ossature standard de sécurité réseau choisie par l'IETF pour IP versions 4 et 6.

IPsec protège votre trafic de données grâce aux techniques suivantes de chiffrement :

Authentification Processus consistant à vérifier l'identité d'un hôte ou d'un point d'extrémité

Contrôle d'intégrité
: Processus consistant à vérifier qu'aucune modification des données n'est survenue au cours de leur transfert sur le réseau

Chiffrement Processus garantissant la confidentialité par le masquage des données et des adresses IP privées en transit sur le réseau

Les algorithmes d'authentification fournissent la preuve de l'identité de l'expéditeur et de l'intégrité des données, en utilisant une fonction de chiffrement par hachage qui traite un paquet de données (y compris l'en–tête IP fixe) à l'aide d'une clé privée, afin d'en produire un condensé unique. Du côté du destinataire, les données sont traitées à l'aide de la même fonction et de la même clé. Si les données ont subi une altération ou si la clé de l'émetteur est incorrecte, le datagramme est supprimé.

Le chiffrement fait appel à un algorithme et une clé pour modifier et rendre apparemment aléatoires les données, qui se transforment ainsi en *texte chiffré*. Ces données en cours de transfert sont incompréhensibles. Lorsqu'elles arrivent à destination, les données sont rétablies à l'aide du même algorithme et de la même clé (algorithmes de chiffrement symétriques). Le chiffrement doit être utilisé en conjonction avec l'authentification, de manière à vérifier l'intégrité des données chiffrées.

Ces services de base sont implémentés dans IPsec au moyen de l'encapsulation IP ESP (Encapsulating Security Payload) et de l'en–tête d'authentification AH (Authentication Header). ESP assure la confidentialité par le chiffrement du paquet IP original, la création d'un en–tête ESP, et l'insertion des données chiffrées (le texte chiffré) dans le paquet ESP.

Lorsque l'authentification et le contrôle d'intégrité des données sont requis, sans confidentialité, l'en–tête d'authentification (AH) peut être utilisé seul. Avec AH, les zones fixes de l'en–tête IP et les données sont traitées par un algorithme de hachage afin de générer un condensé codé. Le destinataire utilise sa clé pour calculer et comparer le condensé, afin de vérifier que le paquet n'a pas été modifié et que l'identité de l'émetteur ne fait aucun doute.

Fonctions de sécurité IP

La sécurité IP de ce système d'exploitation propose les fonctions suivantes :

- Accélération matérielle avec la carte PCI Ethernet 10/100 Mbits/s type II.
- En–tête d'authentification (AH) de la RFC 2402, encapsulation (ESP) de la RFC 2406.
- Liste de révocation des certificats (CRL) avec extraction via des serveurs HTTP ou LDAP.
- Actualisation automatique des clés à l'aide de tunnels utilisant le protocole IKE (Internet Key Exchange) de l'IETF.
- Certificats numériques X.509 et clés pré–partagées du protocole IKE, lors de la négociation des clés.
- Configuration des tunnels manuels pour garantir la compatibilité avec des systèmes qui ne prennent pas en charge les méthodes automatiques d'actualisation de clés IKE, et pour l'utilisation de tunnels IP v6.
- Utilisation des modes d'encapsulation Tunnel et Transport pour les tunnels hôte ou passerelle.
- Algorithmes d'authentification HMAC (Hashed Message Authentication Code), MD5 (Message Digest 5) et HMAC SHA (Secure Hash Algorithm).
- Algorithmes de chiffrement DES 56 bits (Data Encryption Standard) CBC (Cipher Block Chaining) avec IV 64 bits (initial vector), Triple DES, DES CBC 4 avec IV 32 bits.
- Prise en charge de la double pile IP (IP v4 et v6).
- Le trafic IP v4 et v6 peut être encapsulé et filtré. Les piles IP étant distinctes, la sécurité IP de chaque pile peut être configurée de manière indépendante.
- Les tunnels IKE peuvent être créés à l'aide de fichiers de configuration Linux (AIX 5.1 et plus).
- Filtrage du trafic sécurisé et non sécurisé par un certain nombre de caractéristiques IP, telles que les adresses IP source et destination, l'interface, le protocole, les numéros de port, etc.
- Génération et suppression automatique des règles de filtrage avec la plupart des types de tunnels.
- Utilisation de noms d'hôte pour l'adresse de destination lors de la définition des tunnels et des règles de filtrage. Les noms d'hôte sont convertis automatiquement en adresses IP (si un DNS est disponible).
- Journalisation des événements de sécurité IP dans **syslog**.
- Utilisation des opérations de suivi système et de statistiques pour l'identification des incidents.
- Les opérations par défaut définies par l'utilisateur lui permettent d'indiquer si le trafic doit faire l'objet d'une autorisation d'accès lorsqu'il ne correspond pas aux tunnels définis.

Fonctions IKE (Internet Key Exchange)

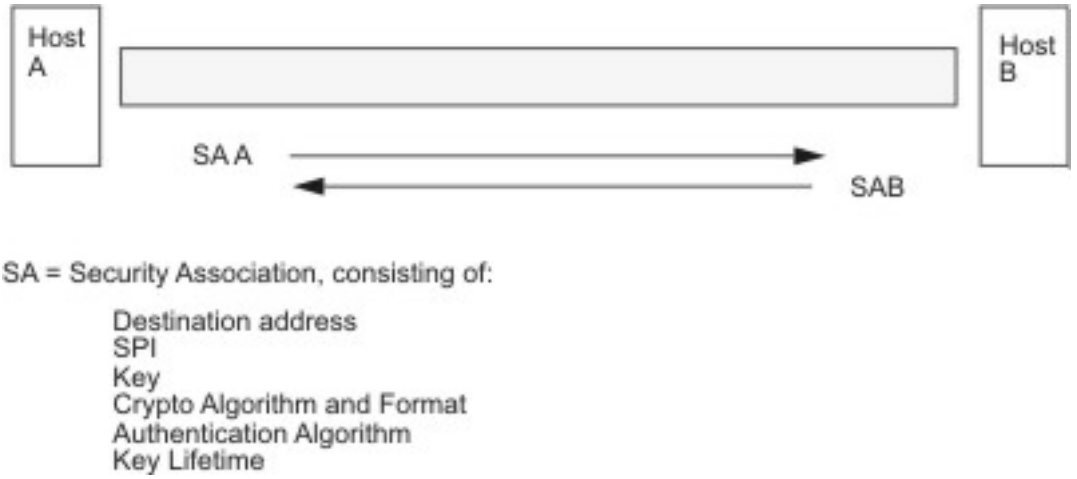
Les fonctions suivantes sont disponibles avec Internet Key Exchange (à partir d'AIX 4.3.2) :

- Authentification avec clés pré–partagées et signatures numériques X.509.
- Utilisation du mode principal (identification du mode de protection) et du mode agressif.
- Prise en charge des groupes Diffie Hellman 1, 2 et 5.
- Chiffrement ESP pour DES, Triple DES, chiffrement Null ; authentification ESP avec HMAC MD5 et HMAC SHA1.
- AH pour HMAC MD5 et HMAC SHA1.
- IP versions 4 et 6.

Liens de sécurité

Le bloc constitutif sur lequel s'appuie la communication sécurisée est un concept connu sous le nom de *lien de sécurité*. Les liens de sécurité associent un ensemble de paramètres de sécurité à un type de trafic. Avec des données protégées par IPsec, chaque direction et chaque type d'en–tête, AH ou ESP, a un lien de sécurité distinct. Le lien de sécurité comprend les informations suivantes : adresses IP des correspondants, identificateur unique SPI (Security Parameters Index), algorithmes sélectionnés et clés d'authentification ou de chiffrement, et durée de vie des clés. La figure suivante montre les liens de sécurité entre les hôtes A et B.

Figure 6. Etablissement d'un tunnel sécurisé entre les hôtes A et B. Ce schéma représente un tunnel virtuel entre l'hôte A et l'hôte B. Le lien de sécurité A est une flèche dirigée de l'hôte A vers l'hôte B. Le lien de sécurité B est une flèche dirigée de l'hôte B vers l'hôte A. Un lien de sécurité se compose de l'adresse de destination, le SPI, la clé, le format et l'algorithme de chiffrement, l'algorithme d'authentification et la durée de vie de la clé.



La gestion des clés a pour objectif de négocier et générer des liens de sécurité pour la protection du trafic IP.

Gestion des clés et tunnels

Pour définir une communication sécurisée entre deux hôtes, les liens de sécurité doivent être négociés et gérés lors de l'utilisation du tunnel. Les types de tunnels suivants sont pris en charge, et utilisent chacun une technique différente de gestion des clés :

- tunnels IKE (clés dynamiques, norme IETF)
- Tunnels manuels (statiques, clés persistantes, norme IETF)

Prise en charge du tunnel IKE

Les tunnels IKE s'appuient sur la norme ISAKMP/Oakley (Internet Security Association and Key Management Protocol) développée par l'IETF. Dans ce protocole, les paramètres de sécurité sont négociés et actualisés, et les clés sont échangées en toute sécurité. Les différents types d'authentification pris en charge sont : clés pré–partagées et signatures de certificats numériques X.509v3.

La négociation s'effectue en deux phases. La première authentifie les correspondants et indique les algorithmes à utiliser pour sécuriser la communication de la deuxième étape. Au cours de la deuxième phase, les paramètres de sécurité IP à utiliser pour le transfert de données sont négociés. Les liens de sécurité et les clés sont créés et échangés.

Le tableau suivant montre les algorithmes d'authentification qui peuvent être utilisés avec les protocoles de sécurité AH et ESP pour les tunnels IKE.

Algorithme	AH IP version 4 & 6	ESP IP version 4 & 6
HMAC MD5	X	X
HMAC SHA1	X	X
DES CBC 8		X
Triple DES CBC		X
ESP Null		X

Prise en charge des tunnels manuels

Les tunnels manuels fournissent la compatibilité amont, ainsi qu'avec des postes ne prenant pas en charge les protocoles de gestion de clés IKE. L'inconvénient de ces tunnels manuels est que les valeurs de clés sont statiques. Les clés d'authentification et de chiffrement sont identiques pour toute la durée de vie du tunnel et doivent être mises à jour manuellement.

Le tableau suivant montre les algorithmes d'authentification pouvant être utilisés avec les protocoles de sécurité AH et ESP pour les tunnels manuels.

Algorithme	AH IP Version 4	AH IP Version 6	ESP IP Version 4	ESP IP Version 6
HMAC MD5	X	X	X	X
HMAC SHA1	X	X	X	X
Triple DES CBC			X	X
DES CBC 8			X	X
DES CBC 4			X	X

Grâce à l'efficacité de la sécurité de ses tunnels, IKE est la méthode de gestion des clés la plus répandue.

Fonctions de filtrage natif

Le *filtrage* est une fonction de base qui permet d'accepter ou de refuser les paquets entrants ou sortants en fonction de certains critères. L'utilisateur ou l'administrateur peut alors configurer l'hôte pour contrôler le trafic entre cet hôte et les autres. Le filtrage s'effectue à partir des propriétés des paquets, telles que les adresses source et de destination, la version IP (4 ou 6), les masques de sous–réseau, le protocole, le port, les propriétés de routage, la fragmentation, l'interface et la définition des tunnels.

Les règles, ou *règles de filtrage*, permettent d'associer certains trafics à un tunnel particulier. Dans une configuration de base pour tunnels manuels, lorsqu'un utilisateur définit un tunnel hôte à hôte, des règles de filtrage sont générées automatiquement afin de canaliser tout le trafic de cet hôte vers le tunnel sécurisé. Si vous souhaitez avoir d'autres types de trafic plus spécifiques (sous–réseau à sous–réseau par exemple), vous pouvez modifier ou remplacer les règles de filtrage pour autoriser un contrôle précis du trafic utilisant un tunnel particulier.

Pour les tunnels IKE, les règles de filtrage sont également générées automatiquement et insérées dans le tableau de filtre dès que le tunnel est activé.

De même, lorsqu'un tunnel est modifié ou supprimé, les règles de filtrage de ce tunnel sont automatiquement supprimées, ce qui simplifie considérablement la configuration de la sécurité IP et réduit le risque d'erreur humaine. Les définitions de tunnel peuvent être diffusées et partagées avec d'autres machines et pare–feu à l'aide d'utilitaires d'importation et d'exportation, ce qui contribue à simplifier l'administration d'un grand nombre de systèmes.

Les règles de filtrage associent des types particuliers de trafic à un tunnel, mais les données filtrées n'ont pas forcément besoin de passer par un tunnel. Ces règles permettent au système d'exploitation d'assurer des fonctions élémentaires de pare–feu pour les utilisateurs souhaitant limiter le flux de certains types de trafic avec leur machine. Ceci est particulièrement utile pour la gestion de machines au sein d'un réseau interne ou ne bénéficiant pas de la protection d'un pare–feu. Dans ce cas, les règles de filtrage édifient une deuxième protection autour d'un groupe de machines.

Dès que les règles de filtrage sont générées, elles sont enregistrées dans un tableau puis chargées dans le noyau. Lorsqu'un échange de paquets se prépare sur le réseau, les règles de filtrage sont successivement étudiées, de haut en bas, afin de déterminer si le prochain paquet doit être accepté, refusé ou envoyé via un tunnel. Les critères de la règle et les caractéristiques des paquets sont confrontés jusqu'à ce qu'une concordance soit trouvée ou que la règle par défaut soit atteinte.

La fonction de sécurité IP met également en œuvre un système de filtrage des paquets non sécurisés en fonction de critères définis par l'utilisateur avec une granularité élevée. Cela peut être utile pour le contrôle du trafic IP entre réseaux et postes n'exigeant pas le recours à la sécurité IP.

Prise en charge des certificats numériques

IPsec accepte les certificats numériques X.509 version 3. Key Manager gère les demandes de certificat et la base de données de clés, ainsi que d'autres fonctions administratives.

Le fonctionnement des certificats numériques est décrit à la section Configuration des certificats numériques, page 12-25. Key Manager et ses fonctions sont décrits à la section Utilisation de Key Manager, page 12-25

Virtual Private Networks (VPN) et sécurité IP

Un réseau privé virtuel prolonge le réseau interne d'une entreprise sur un réseau public tel qu'Internet. Les VPN permettent d'échanger des informations via Internet, dans un tunnel privé, avec des utilisateurs distants, des succursales et des partenaires ou fournisseurs. L'accès à Internet par des prestataires de services Internet (ISP), via des lignes directes ou des numéros de téléphone locaux, permet de s'affranchir des lignes louées coûteuses, des appels longue distance et des numéros gratuits. IPsec peut être utilisée pour une solution VPN, car c'est la structure de sécurité choisie par l'IETF pour les environnements IPv4 et 6, et aucune modification des applications n'est nécessaire.

Installation de la sécurité IP

La fonction de sécurité IP sous AIX s'installe et se charge séparément. Les fichiers à installer sont les suivants :

- **bos.net.ipsec.rte** (environnement d'exécution pour l'environnement et les commandes du noyau de sécurité IP)
- **bos.msg.** *LANG***.net.ipsec** (où *LANG* est la langue de votre choix, par exemple **en_US**)
- **bos.net.ipsec.keymgt**
- **bos.net.ipsec.websm**
- **bos.crypto–priv** (fichier défini pour le chiffrement DES et triple DES)

Le fichier **bos.crypto–priv** se trouve dans Expansion Pack. Pour le support de signature numérique IKE, installez l'ensemble de fichiers **gskit.rte** (AIX Version 4) ou **gskkm.rte** (AIX 5.1) à partir de Expansion Pack.

Pour la prise en charge de la sécurité IP dans Web–based System Manager, vous devez installer l'ensemble de fichiers **Java131.ext.xml4j** au niveau 1.3.1.1 ou version ultérieure.

Une fois installée, la sécurité IP peut être chargée séparément pour IPv4 et IPv6, soit en suivant la procédure recommandée à la section Chargement de la fonction de sécurité IP, page 12-7, soit en utilisant la commande **mkdev**.

Chargement de la fonction de sécurité IP

Attention : Le chargement de la sécurité IP active la fonction de filtrage. Avant le chargement, il est important de vérifier que les règles de filtrage sont correctement créées. Sinon, toutes les communications extérieures peuvent être bloquées.

Utilisez SMIT ou Web–based System Manager pour charger automatiquement les modules de sécurité IP au moment du démarrage de la sécurité IP. De plus, SMIT et Web–based System Manager garantissent que les extensions du noyau et les démons IKE sont chargés dans le bon ordre.

Si le chargement s'est correctement déroulé, la commande **lsdev** indique que les unités de sécurité IP sont `Available` (disponibles).

```
lsdev -C -c ipsec

    ipsec_v4 Available IP Version 4 Security Extension
    ipsec_v6 Available IP Version 6 Security Extension
```

Une fois l'extension du noyau de sécurité IP chargée, vous pouvez configurer les tunnels et les filtres.

Planification de la configuration de la sécurité IP

Avant de configurer IPsec, vous devez configurer les tunnels et les filtres. Si un seul tunnel est défini pour l'ensemble des échanges de données, les règles de filtrage peuvent être générées automatiquement. Pour définir un système de filtres plus élaboré, les règles peuvent être configurées séparément.

Vous pouvez configurer la sécurité IP à l'aide du plug–in Web–based System Manager Network, du plug–in Virtual Private Network ou de l'outil System Management Interface Tool (SMIT). Pour SMIT, vous bénéficiez des raccourcis suivants :

smit ips4_basic
Configuration de base pour IP version 4

smit ips6_basic
Configuration de base pour IP version 6

Avant de configurer la sécurité IP sur votre site, vous devez définir la méthode à utiliser ; par exemple, l'utilisation de tunnels ou de filtres (ou les deux), le type de tunnel qui correspond le mieux à vos besoins, etc. Vous devez étudier les informations des sections suivantes avant de prendre de telles décisions :

- Accélération matérielle, page 12-8
- Tunnels / Filtres, page 12-10
- Tunnels et liens de sécurité, page 12-11
- Choix d'un type de tunnel, page 12-15
- Utilisation d'IKE avec DHCP ou Dynamically Assigned Addresses (affectation dynamique des adresses), page 12-15

Accélération matérielle

La carte PCI Ethernet 10/100 Mbits/s type II (Code 4962) offre la sécurité IP. Elle est conçue assurer ces fonctions à la place du système d'exploitation AIX. Lorsque cette carte est présente dans le système AIX, la pile de sécurité IP en utilise les fonctions suivantes :

- Chiffrement et déchiffrement à l'aide des algorithmes DES et Triple DES
- Authentification à l'aide des algorithmes MD5 ou SHA–1
- Stockage des informations des liens de sécurité.

Les fonctions de la carte sont utilisées à la place du logiciel. La carte PCI Ethernet 10/100 Mbits/s type II est disponible pour les tunnels manuels et IKE.

La fonction d'accélération matérielle de la sécurité IP est disponible à partir du niveau **5.1.0.25** des ensembles de fichiers **bos.net.ipsec.rte** et **devices.pci.1410ff01.rte**.

Le nombre de liens de sécurité entrants pouvant être traités par la carte réseau est limité. Pour le trafic sortant, tous les paquets qui utilisent une configuration prise en charge sont traités par la carte. Certaines configurations de tunnel ne peuvent pas être traitées.

La carte PCI Ethernet 10/100 Mbits/s de type II prend en charge les éléments suivants :

- Chiffrement DES, 3 DES ou NULL avec ESP
- Authentification HMAC–MD5 ou HMAC–SHA–1 avec ESP ou AH, mais pas les deux. (si ESP et AH sont utilisés tous les deux, vous devez effectuer ESP en premier. Toujours vrai pour les tunnels IKE, mais l'utilisateur peut sélectionner l'ordre pour les tunnels manuels.)
- Mode de transport et de tunnel
- Déchargement de paquets IPv4

Remarque : La carte 10/100 Mbps Ethernet PCI Adapter II ne peut pas traiter les paquets avec des options IP.

Pour activer la sécurité IP avec la carte PCI Ethernet 10/100 Mbits/s type II, il faudra peut–être déconnecter l'interface réseau, puis activer la fonction de déchargement IPsec.

Pour déconnecter l'interface réseau à l'aide de l'interface SMIT, procédez comme suit :

1. Connectez–vous en tant qu'utilisateur **root** (root).
2. Tapez `smitty inet` en ligne de commande puis appuyez sur Entrée.
3. Sélectionnez l'option **Suppression d'une interface réseau** puis appuyez sur Entrée.
4. Sélectionnez l'interface correspondant à la carte PCI Ethernet 10/100 Mbits/s type II puis appuyez sur Entrée.

Pour activer la fonction de déchargement IPsec à l'aide de l'interface SMIT, procédez comme suit :

1. Connectez–vous en tant qu'utilisateur **root** (root).
2. Tapez `smitty eadap` en ligne de commande puis appuyez sur Entrée.
3. Sélectionnez l'option **Modif/affich caractéristiques d'une carte Ethernet** et appuyez sur Entrée.
4. Sélectionnez la carte PCI Ethernet 10/100 Mbits/s type II puis appuyez sur Entrée.
5. Définissez la zone **Déchargement IPsec** sur **oui** puis appuyez sur Entrée.

Pour activer l'attribut de déchargement IPsec à partir de la ligne de commande, tapez :

```
# ifconfig en   X   detach
```

Pour activer l'attribut de déchargement IPsec à partir de la ligne de commande, tapez ce qui suit :

```
# chdev -l ent   X   -a ipsec_offload=yes
```

Pour vérifier que l'attribut de déchargement IPsec a bien été activé, tapez ce qui suit sur la ligne de commande :

```
# lsattr -El ent   X   detach
```

Pour désactiver l'attribut de déchargement IPsec à partir de la ligne de commande, tapez ce qui suit :

```
# chdev -l ent   X   -a ipsec_offload=no
```

Utilisez la commande **enstat** pour vous assurer que la configuration de votre tunnel utilise l'attribut de déchargement IPsec. La commande **enstat** montre toutes les statistiques de réception et de transmission des paquets IPsec lorsque l'attribut de déchargement IPsec est activé. Par exemple, pour une interface Ethernet *ent*, entrez :

```
# entstat -d   ent1
```

Le résultat de cette commande se présente comme suit :

```
.
.
.
 10/100 Mbps Ethernet PCI Adapter II (1410ff01) Specific Statistics:
 ---------------------------------------------
.
.
.
 Transmit IPsec packets: 3
 Transmit IPsec packets dropped: 0
 Receive IPsec packets: 2
 Receive IPsec packets dropped: 0
```

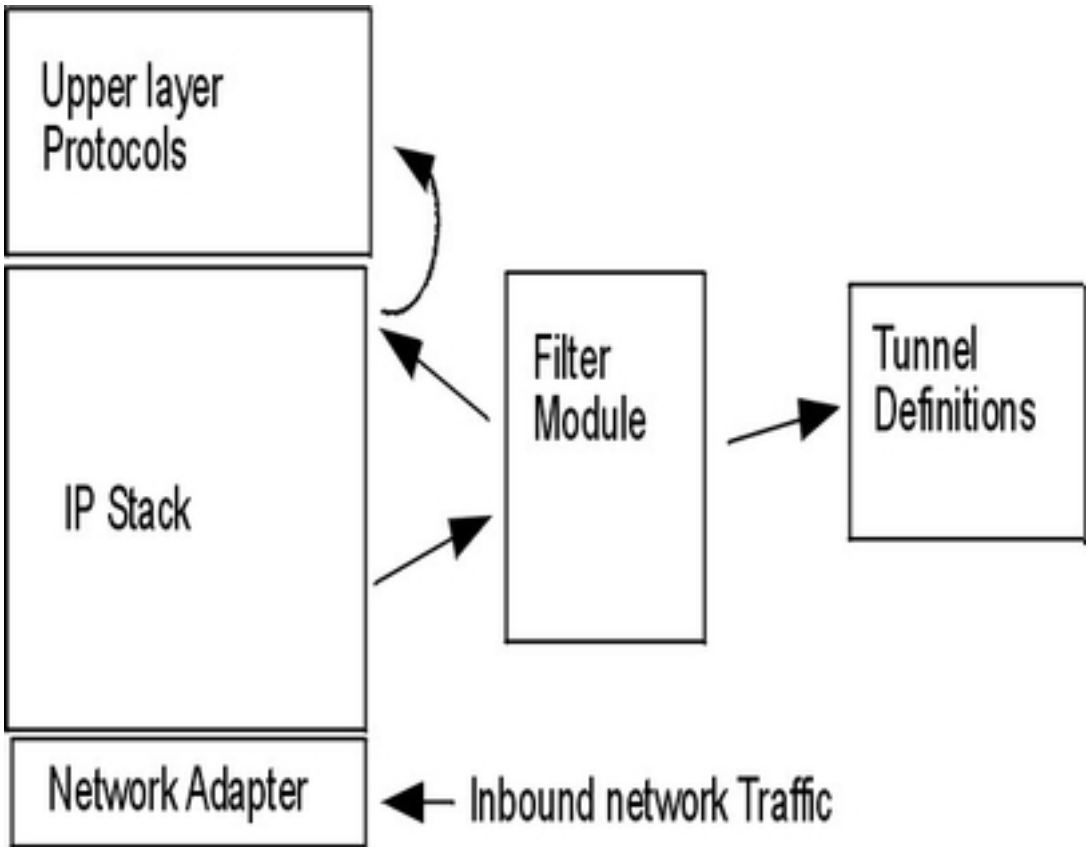
Tunnels / Filtres

La sécurité IP comporte deux parties distinctes, les *tunnels* et les *filtres*.
Les tunnels ont besoin des filtres, mais les filtres peuvent se passer des tunnels.

- Le *filtrage* est une fonction qui permet d'accepter ou de refuser les paquets entrants et sortants en fonction d'un certain nombre de critères, appelés *règles*. Ainsi, un administrateur peut configurer le système hôte afin de gérer l'échange de données entre cet hôte et d'autres systèmes hôtes. Le filtrage s'effectue à partir des propriétés des paquets, telles que les adresses source et de destination, la version IP (4 ou 6), les masques de sous–réseau, le protocole, le port, les propriétés de routage, la fragmentation, l'interface et la définition des tunnels. Ce filtrage s'effectue au niveau de la couche IP ; aucune modification ne s'impose donc au niveau des applications.
- Les *tunnels* définissent un lien de sécurité entre deux systèmes hôtes. Ces liens de sécurité impliquent des paramètres de sécurité spécifiques qui sont partagés par les systèmes aux extrémités du tunnel.

Le schéma suivant illustre la manière dont un paquet arrive de la carte réseau sur la pile IP. A partir de là, le module de filtrage est appelé pour déterminer si le paquet est autorisé ou refusé. Si un ID de tunnel est spécifié, le paquet subit un contrôle par rapport aux définitions de tunnel. Si la décapsulation du tunnel se déroule correctement, la paquet est transmis au protocole de la couche supérieure. Cette fonction s'effectue dans l'ordre inverse pour les paquets sortants. Le tunnel s'appuie sur une règle de filtrage qui associe le paquet à un tunnel donné, mais la fonction de filtrage peut se produire sans transmission du paquet au tunnel.

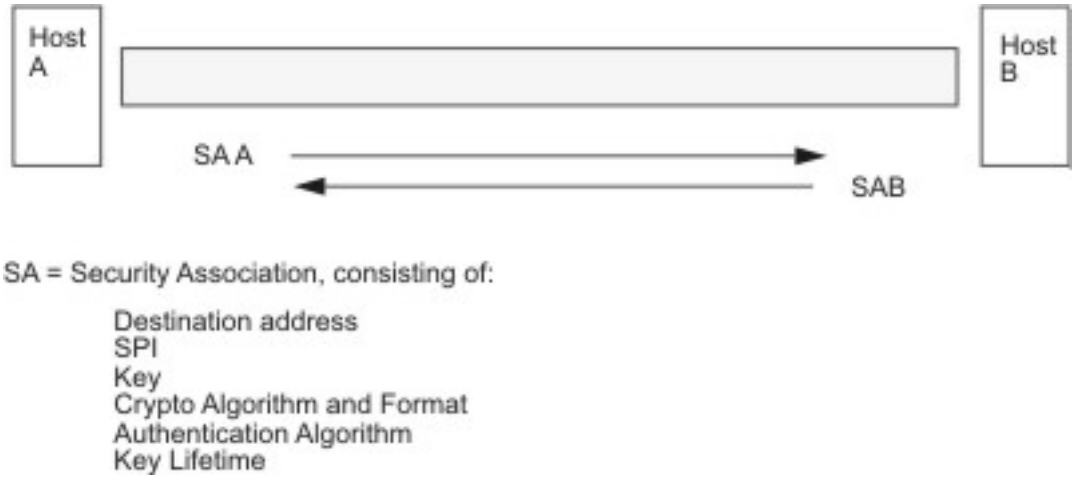
Figure 7. Routage de paquet réseau. Ce schéma présente le chemin emprunté par un paquet réseau. En provenance du réseau, le paquet entre dans la carte réseau. Il est ensuite acheminé vers la pile IP d'où il est envoyé pour aller dans le module filtre. Depuis le module filtre, le paquet est envoyé aux définitions de tunnel ou bien retourné vers la pile IP, qui le transmettra aux protocoles de la couche supérieure.



Tunnels et liens de sécurité

Les tunnels servent à authentifier et/ou à chiffrer les données. Les tunnels sont définis en spécifiant un lien de sécurité entre deux systèmes hôte. Le lien de sécurité définit les paramètres des algorithmes de chiffrement et d'authentification, et les caractéristiques du tunnel. Le schéma suivant présente un tunnel virtuel entre les hôtes A et B.

Figure 8. Etablissement d'un tunnel sécurisé entre les hôtes A et B. Ce schéma représente un tunnel virtuel entre l'hôte A et l'hôte B. Le lien de sécurité A est une flèche dirigée de l'hôte A vers l'hôte B. Le lien de sécurité B est une flèche dirigée de l'hôte B vers l'hôte A. Un lien de sécurité comprend l'adresse de destination, le SPI, la clé, le format et l'algorithme de chiffrement, l'algorithme d'authentification et la durée de vie de la clé.



La valeur SPI (Security Parameter Index) et l'adresse de destination identifient un lien de sécurité unique. Ces paramètres sont nécessaires pour définir un tunnel de manière unique. Vous pouvez spécifier d'autres paramètres, comme l'algorithme de chiffrement, l'algorithme d'authentification, les clés et la durée de vie, ou utiliser les valeurs par défaut.

Remarques sur le tunnel

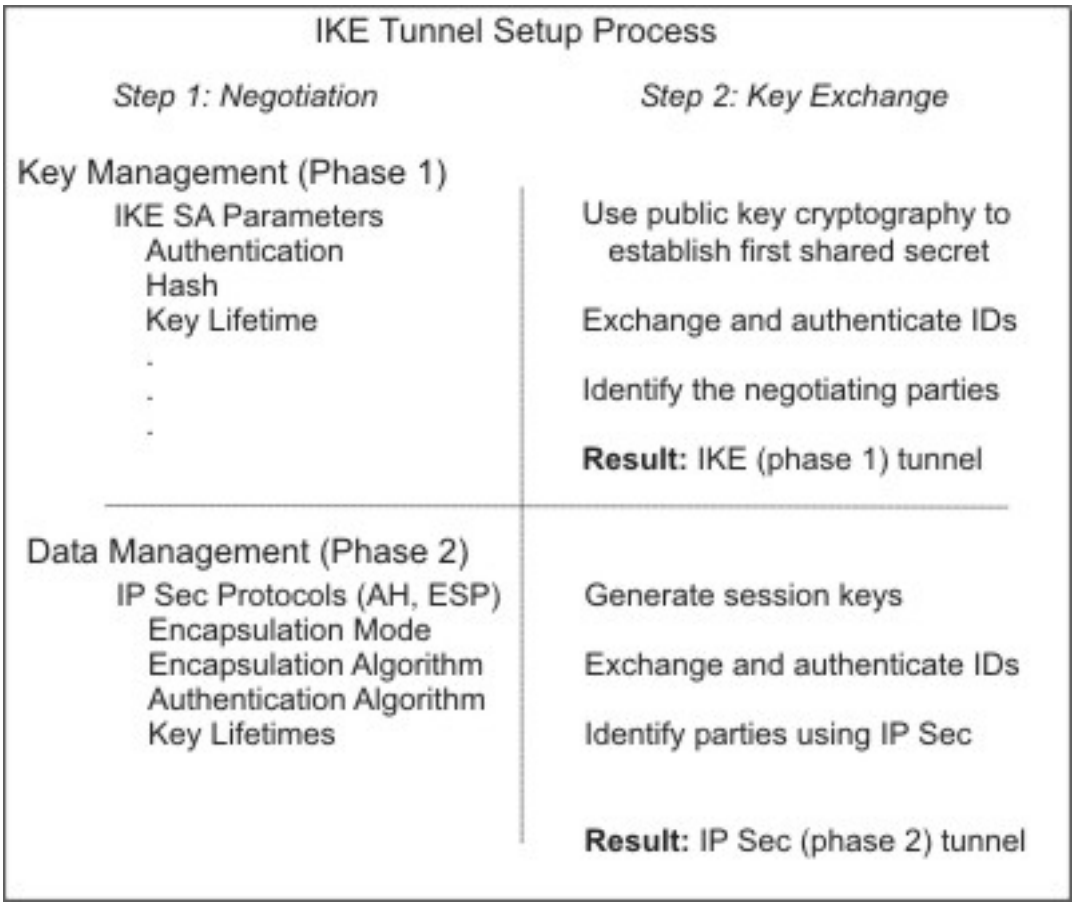
Les tunnels IKE se distinguent des tunnels manuels dans la mesure où la stratégie de sécurité fait l'objet d'un processus distinct par rapport à la définition des points d'extrémité du tunnel. Dans IKE, le processus de négociation se divise en deux étapes. Chaque étape est appelée *phase* et chaque phase peut avoir sa propre stratégie de sécurité.

Lors du démarrage des négociations IKE, un tunnel sécurisé doit être défini. Cette phase est appelée phase de *gestion de clés* ou *phase 1*. Lors de cette phase, chaque correspondant utilise des clés pré–partagées ou des certificats numériques pour authentifier l'autre et transmettre les informations d'ID. Cette phase définit un lien de sécurité pendant lequel les deux correspondants déterminent les protections à appliquer pour communiquer en toute sécurité pendant la seconde phase. Le résultat de cette phase est un tunnel *IKE* ou tunnel *phase 1*.

La seconde étape est appelée phase de *gestion de données* ou *phase 2*. A l'aide du tunnel IKE, elle crée les liens de sécurité pour AH et ESP, lesquels protègent les échanges. La deuxième phase détermine également les données qui circuleront dans le tunnel de la sécurité IP. Par exemple, elle peut définir les éléments suivants :

- Un masque de sous–réseau
- Une plage d'adresse
- Un numéro de port et un protocole

Figure 9. Processus de configuration du tunnel IKE. Ce schéma présente les deux phases du processus de configuration d'un tunnel IKE.



Dans certains cas, les extrémités du tunnel (IKE) de gestion de clés seront identiques aux extrémités du tunnel (sécurité IP) de gestion des données. Les points d'extrémité du tunnel IKE sont les ID des postes exécutant les négociations. Les points d'extrémité du tunnel de la sécurité IP décrivent le type de trafic qui circule dans ce tunnel. Pour les tunnels hôte à hôte simples, dans lesquels tous les échanges entre deux tunnels sont protégés avec le même tunnel, les extrémités du tunnel des phases 1 et 2 sont identiques. Lorsque la négociation se déroule entre deux passerelles, ce sont elles les points d'extrémité du tunnel IKE, et les points d'extrémité du tunnel de la sécurité IP correspondent aux machines ou aux sous–réseaux (derrière les passerelles) ou bien à la plage d'adresses (derrière les passerelles) des utilisateurs du tunnel.

Paramètres et stratégie de la gestion de clés

La phase 1 (gestion de clés) définit les paramètres suivants de la configuration d'un tunnel IKE :

Tunnel (phase 1) de gestion de clés	Nom de ce tunnel IKE. Pour chaque tunnel, vous devez indiquer les extrémités de négociation. Ce sont les deux postes qui comptent envoyer et valider des messages IKE. Le nom du tunnel peut décrire ses extrémités, comme `VPN Boston` ou `VPN Acme`.
Type d'identité hôte	Type d'ID qui sera utilisé lors d'un échange IKE. Le type et la valeur de l'ID doivent correspondre à la valeur de la clé pré–partagée afin de garantir une recherche correcte de clés. Si un ID séparé est utilisé pour rechercher une valeur de clé pré–partagée, l'*ID hôte* est l'ID de clé et son *type* est KEY_ID. Le type KEY_ID est utile si un seul hôte possède plusieurs valeurs de clé pré–partagée.
Identité hôte	Valeur de l'ID hôte représentée sous la forme d'une adresse IP, d'un nom de domaine qualifié complet (FQDN) ou d'un utilisateur sur le nom de domaine qualifié complet (*utilisateur@FQDN*). Par exemple, jdoe@studentmail.ut.edu.
Adresse IP	Adresse IP de l'hôte distant. Cette valeur est nécessaire si le type d'ID hôte est KEY_ID ou s'il ne correspond pas à un type pouvant être résolu en une adresse IP. Par exemple, si le nom d'utilisateur ne peut être résolu par le serveur de noms local, vous devez saisir l'adresse IP de la partie distante.

Vous pouvez également créer une stratégie personnalisée en spécifiant les paramètres à utiliser lors de la négociation IKE. Par exemple, vous pouvez utiliser des stratégies de gestion de clés pour l'authentification via les clés pré–partagées ou le mode signature. Pour la phase 1, l'utilisateur doit indiquer certaines propriétés de sécurité de gestion de clés avec lesquelles l'échange doit se réaliser.

Paramètres et stratégie de la gestion de données

Les paramètres de proposition de la gestion de données sont définis lors de la phase 2 de la configuration d'un tunnel IKE. Il s'agit des mêmes paramètres utilisés pour la sécurité IP dans les tunnels manuels ; ils identifient le type de protection à utiliser pour le trafic des données dans le tunnel. Vous pouvez démarrer plusieurs tunnels de phase 2 sous le même tunnel de phase 1.

Les types d'ID de points d'extrémité suivants décrivent le type de données devant utiliser le tunnel de données de sécurité IP :

Hôte, **Sous–réseau** ou **Plage**	Ces éléments précisent si le trafic de données empruntant le tunnel est destiné à un hôte, à un sous–réseau ou à une plage d'adresses.
ID hôte/sous–réseau	Contient l'identité hôte ou sous–réseau des systèmes locaux ou distants acheminant des données via ce tunnel. Détermine les ID envoyés au cours de la négociation de la phase 2 et les règles de filtrage qui seront établies si la négociation se déroule correctement.
Masque de sous–réseau	Décrit toutes les adresses IP au sein du sous–réseau (par exemple, hôte 9.53.250.96 et masque 255.255.255.0)
Plage d'adresses IP de début	Fournit l'adresse IP de début de la plage d'adresses qui utilisera le tunnel (par exemple, 9.53.250.96 de 9.53.250.96 à 9.53.250.93)
Plage d'adresses IP de fin	Fournit l'adresse IP de fin pour la plage d'adresses qui utilisera le tunnel (par exemple, 9.53.250.93 de 9.53.250.96 à 9.53.250.93)
Port	Décrit les données utilisant un numéro de port spécifique (par exemple, 21 ou 23)
Protocole	Décrit les données acheminées via un protocole spécifique (par exemple, TCP ou UDP). Détermine le protocole envoyé au cours de la négociation de phase 2 et les règles de filtrage qui seront établies si la négociation se déroule correctement. Le protocole de l'extrémité locale doit correspondre à celui de l'extrémité distante.

Choix d'un type de tunnel

Le choix entre des tunnels manuels ou IKE dépend de la prise en charge du tunnel par le système distant situé à l'autre extrémité et du type de gestion de clés choisi. Nous vous recommandons les tunnels IKE (si disponibles) car ils assurent la négociation sécurisée des clés et leur mise à jour. Ils bénéficient également des types d'en–tête de l'IETF, ESP et AH, et acceptent la protection contre les répétitions. Vous pouvez configurer le mode de signature pour permettre les certificats numériques.

Si le système distant à l'autre extrémité utilise l'un des algorithmes nécessitant des tunnels manuels, utilisez les tunnels manuels. Ils garantissent une compatibilité avec un grand nombre d'hôtes. Mais comme les clés sont statiques, difficiles à modifier et à mettre jour, elles ne sont pas aussi sûres. Les tunnels manuels peuvent être utilisés entre un hôte avec ce système d'exploitation et toute autre machine dotée de la sécurité IP, et proposant un ensemble commun d'algorithmes de chiffrement et d'authentification. Dans leur grande majorité, les fournisseurs proposent Keyed MD5 avec DES ou HMAC MD5 avec DES. Cette répartition fonctionne avec pratiquement toutes les implémentations d'IPsec.

Lors de la configuration des tunnels manuels, la procédure varie selon que vous configurez le premier hôte du tunnel ou le deuxième, dont les paramètres doivent correspondre à la configuration du premier hôte. Si vous configurez le premier hôte, les clés peuvent être générées automatiquement, et les algorithmes définis par défaut. Pour configurer le deuxième hôte, vous devez importer, si possible, les informations du tunnel à partir du système distant.

Autre élément important : déterminer si le système distant est situé derrière un pare–feu. Si tel est le cas, la configuration doit comporter les informations sur le pare–feu en question.

Utilisation d'IKE avec DHCP ou Dynamically Assigned Addresses (affectation dynamique des adresses)

IPsec s'utilise couramment pour sécuriser un système d'exploitation lorsque les systèmes distants initialisent des sessions IKE avec un serveur, sans que leur identité puisse être liée à une adresse IP en particulier. Ce cas de figure se rencontre dans un environnement LAN, lorsque vous utilisez IPSec pour vous connecter à un serveur LAN et que vous souhaitez chiffrer les données. Il peut s'agir aussi de clients distants qui composent le numéro d'un serveur et utilisent soit un nom de domaine qualifié complet (FQDN), soit une adresse électronique (utilisateur@FQDN) pour identifier l'ID distant.

Pendant la phase de gestion de clés (phase 1), une signature RSA est le seul mode d'authentification pris en charge si vous utilisez le mode principal avec des ID d'adresse non IP. En d'autres termes, si vous souhaitez utiliser une authentification de clés pré–partagée, vous devez utiliser un mode agressif ou un mode principal avec des adresses IP en tant qu'ID. En fait, lorsque le nombre des clients DHCP avec lesquels vous souhaitez établir des tunnels IPsec est important, il devient difficile de définir des clés uniques et pré–partagées pour chaque client DHCP. Dans ce cas, il est recommandé d'utiliser l'authentification par signature RSA. Vous pouvez aussi utiliser un ID de groupe comme ID distant dans la définition de tunnel de manière à définir le tunnel une seule fois avec tous les clients DHCP (voir l'exemple de fichier de définition de tunnel **/usr/samples/ipsec/group_aix_responder.xml**). L'ID de groupe est une fonction unique de IPsec AIX. Vous pouvez définir un ID de groupe pour inclure tous les ID IKE (comme une adresse IP unique), un nom de domaine qualifié complet (FQDN), un utilisateur@FQDN, une plage ou un ensemble d'adresses IP, etc. Vous pouvez ensuite utiliser cet ID de groupe comme ID distant de phase 1 ou de phase 2 dans vos définitions de tunnel.

Remarque : Si l'ID de groupe est utilisé, le tunnel doit être défini comme rôle d'appelé uniquement. Ceci signifie que vous devez activer ce tunnel depuis le côté client DHCP.

Pendant la phase de gestion de données (phase 2), lorsque les liens de sécurité IP sont créés pour chiffrer les échanges TCP ou UDP, un tunnel générique de gestion de données peut être configuré. Ainsi, toutes les requêtes authentifiées lors de la phase 1 utiliseront le tunnel générique pour la phase de Gestion des données définie, au cas où l'adresse IP ne serait pas configurée de façon explicite dans la base de données. Ceci permet à toute adresse de correspondre au tunnel générique et d'être utilisée aussi longtemps que la validation rigoureuse d'une sécurité reposant sur des clés publiques aboutit pendant la phase 1.

Utilisation de XML pour définir un tunnel générique de gestion de données

Vous pouvez définir un tunnel générique de gestion de données à l'aide du format XML, compris par **ikedb**. Reportez–vous à la section intitulée Interface de la ligne de commande pour la configuration d'un tunnel IKE, page 12-21, pour plus d'informations sur l'interface IKE XML et la commande **ikedb**. Les tunnels génériques de gestion de données sont utilisés avec le DHCP. Le format XML utilise le nom balise IPSecTunnel pour ce que le Web–based System Manager appelle un tunnel de gestion de données. Dans d'autres contextes, on parle aussi de *tunnel de phase 2*. Un *tunnel de gestion des données génériques* n'est pas un véritable tunnel, mais un **IPSecProtection** utilisé si un message entrant de gestion des données (sous un tunnel spécifique de gestion des clés) ne correspond à aucun tunnel défini pour ce tunnel de gestion de clés. Il est utilisé uniquement lorsque le système AIX est l'appelé. L'indication d'un tunnel générique de gestion de données **IPSecProtection** est facultative.

Le tunnel générique de gestion de données est défini dans l'élément **IKEProtection**, à l'aide de deux attributs XML, appelés **IKE_IPSecDefaultProtectionRef** et **IKE_IPSecDefaultAllowedTypes**.

Tout d'abord, vous devez définir un **IPSecProtection** que vous utiliserez par défaut si aucun **IPSecTunnel** (tunnel de gestion des données) ne correspond. Un **IPSecProtection** utilisé par défaut doit avoir un **IPSec_ProtectionName** qui commence par `_defIPSprot_`.

Allez ensuite sur le **IKEProtection** pour lequel vous souhaitez utiliser le **IPSecProtection** par défaut. Indiquez un attribut **IKE_IPSecDefaultProtectionRef** qui contient le nom du **IPSec_Protection** par défaut.

Vous devez également indiquer une valeur pour l'attribut **IKE_IPSecDefaultAllowedTypes** dans ce **IKEProtection**. Une ou plusieurs des valeurs suivantes peuvent être attribuées (séparez les valeurs multiples par un espace) :

```
Local_IPV4_Address
 Local_IPV6_Address
 Local_IPV4_Subnet
 Local_IPV6_Subnet
 Local_IPV4_Address_Range
 Local_IPV6_Address_Range
 Remote_IPV4_Address
 Remote_IPV6_Address
 Remote_IPV4_Subnet
 Remote_IPV6_Subnet
 Remote_IPV4_Address_Range
 Remote_IPV6_Address_Range
```

Ces valeurs correspondent aux types d'ID indiqués par l'appelant. Dans la négociation IKE, les ID actuels sont ignorés. Le **IPSecProtection** indiqué est utilisé si l'attribut **IKE_IPSecDefaultAllowedTypes** contient une chaîne commençant par `Local_`, correspondant au type d'ID local de l'appelant, et une chaîne commençant par `Remote_`, correspondant à son type d'ID distant. En d'autres termes, vous devez avoir au minimum une valeur **Local**_ et une **Remote**_ pour chaque attribut **IKE_IPSecDefaultAllowedTypes**, pour pouvoir utiliser l'**IPSec_Protection** correspondant.

Exemple

Un appelant envoie ce qui suit au système AIX dans un message de phase 2 (gestion des données) :

```
local ID type:    IPV4_Address
 local ID:         192.168.100.104

 remote ID type:   IPV4_Subnet
 remote ID:        10.10.10.2
 remote netmask:   255.255.255.192
```

Le système AIX ne dispose pas de tunnel de gestion des données correspondant à ces ID. Par contre, il dispose d'un **IPSecProtection** avec les attributs suivants :

```
IKE_IPSecDefaultProtectionRef="_defIPSprot_protection4"
 IKE_IPSecDefaultAllowedTypes="Local_IPV4_Address
                               Remote_IPV4_Address
                               Remote_IPV4_Subnet
                               Remote_IPV4_Address_Range"
```

Le type d'ID local du message entrant, **IPV4_Address**, correspond à l'une des valeurs **Local_** des types autorisés, **Local_IPV4_Address**. L'ID distant du message, **IPV4_Subnet**, correspond également à la valeur **Remote_IPV4_Subnet**. Par conséquent, la négociation du tunnel de gestion des données continuera avec `_defIPSprot_protection4` en tant que **IPSecProtection**.

Le fichier **/usr/samples/ipsec/default_p2_policy.xml** est un fichier XML définissant un **IPSecProtection** générique qui peut être utilisé comme exemple.

Utilisation de Web–based System Manager pour définir un tunnel générique de gestion de données

Procédez comme suit pour définir un tunnel générique de gestion de données à l'aide de Web–based System Manager :

1. Sélectionnez un tunnel de gestion de clés dans le conteneur de tunnels IKE, puis sélectionnez l'action permettant de définir un tunnel de gestion de données.
2. Sélectionnez le tunnel générique de gestion de données. Les panneaux de configuration sont identiques à ceux utilisés pour définir un tunnel de gestion de données. Cependant, les choix pour les types d'ID diffèrent. Il n'est pas nécessaire de préciser des ID explicites. Les types d'ID, IP v4 ou v6 Address Only, IP v4 ou v6 Subnet Only et IP v4 ou v6 Address ou Subnet, couvrent tous les cas disponibles d'ID.
3. Configurez l'information restante de la même façon que pour le tunnel de gestion des données, puis cliquez sur OK. Chaque tunnel de gestion de clés ne peut être lié qu'à un seul tunnel générique.

Remarque : Un tunnel générique de gestion de données peut *uniquement* être utilisé dans les cas où le système AIX est l'appelé.

Configuration d'un tunnel d'échange de clés par Internet (IKE)

Cette section fournit des informations sur la configuration des tunnels d'échange de clés par Internet (IKE) à l'aide de l'interface Web–based System Manager, du SMIT (System Management Interface Tool) ou de la ligne de commande.

Utilisation de Web–based System Manager pour configurer les tunnels IKE

La section Utilisation de l'assistant de configuration de base, page 12-18, offre un moyen simple pour définir un tunnel IKE avec des clés pré–partagées. Pour plus d'informations sur les options avancées, reportez–vous à la section Configuration avancée des tunnels IKE, page 12-19.

Utilisation de l'assistant de configuration de base

Vous pouvez définir un tunnel IKE dans Web–based System Manager à l'aide de clés pré–partagées ou de certificats comme méthode d'authentification. Web–based System Manager ajoute au sous–système de sécurité IP de nouveaux tunnels IKE pour la gestion des clés et des données. Il vous permet également d'entrer un minimum de données et de choisir certaines options. Il utilise enfin les valeurs par défaut courantes pour des paramètres tels que la durée de vie du tunnel.

Lors de l'utilisation de l'assistant de configuration de base, tenez compte des conditions suivantes :

- Vous pouvez utiliser l'assistant uniquement pour la configuration du tunnel initial. Pour modifier, supprimer ou activer un tunnel, utilisez la barre des tâches ou le plug–in **Tunnel IKE**.
- Le nom du tunnel doit être unique sur le système, mais peut être utilisé sur un système distant. A titre d'exemple, sur les systèmes local et distant, le nom du tunnel peut être *hôteA_à_hôteB*, mais les zones Adresse IP locale et Adresse IP distante (points d'extrémité) sont interverties.
- Les tunnels des phases 1 et 2 sont définis avec les mêmes algorithmes de chiffrement et d'authentification.
- La clé pré–partagée que vous entrez doit être en hexadécimal (sans `0x` devant) ou en texte ASCII.
- Si vous choisissez les certificats numériques comme méthode d'authentification, vous devez utiliser l'utilitaire Key Manager décrit page 12-25 pour créer un certificat numérique.
- Le seul type d'ID hôte accepté est `Adresse IP`.
- Les noms attribués aux conversions et aux propositions que vous créez se terminent par le nom du tunnel défini par l'utilisateur. Vous pouvez afficher les conversions et propositions dans Web–based System Manager via **VPN** et le plug–in **tunnel IKE**.

Pour configurer un nouveau tunnel via l'assistant, procédez comme suit :

1. Ouvrez Web–based System Manager à l'aide de la commande **wsm** depuis la ligne de commande.
2. Sélectionnez le plug–in réseau.
3. Sélectionnez **Réseaux privés virtuels (sécurité IP)**.
4. A partir de la zone Console, choisissez le dossier **Généralités et tâches**.
5. Sélectionnez **Assistant de configuration de tunnel de base**.
6. Cliquez sur **Suivant** dans l'écran Introduction de l'étape 1, puis suivez les étapes pour configurer un tunnel IKE.

   L'aide en ligne est disponible.

   Une fois le tunnel défini via l'assistant, sa définition apparaît dans la liste des tunnels IKE de Web–based System Manager ; le tunnel peut être activé ou modifié.

Configuration avancée des tunnels IKE

Vous pouvez configurer séparément les tunnels de gestion des clés et des données à l'aide des procédures suivantes.

Configuration de tunnels de gestion des clés

Les tunnels IKE sont configurés à l'aide du Web–based System Manager.
La procédure suivante permet d'ajouter un tunnel de gestion des clés :

1. Ouvrez Web–based System Manager à l'aide de la commande **wsm**.
2. Sélectionnez le plug–in réseau.
3. Sélectionnez **Réseaux privés virtuels (sécurité IP)**.
4. Dans la zone Console, choisissez **Généralités et tâches**.
5. Sélectionnez **Démarrage de la sécurité IP**. Cette opération charge les extensions du noyau de sécurité IP et lance les démons **isakmpd**, **tmd** et **cpsd**.

   Un tunnel est créé grâce à la définition des points d'extrémité de gestion des clés et de gestion des données ainsi qu'à la définition des conversions et propositions de sécurité associées.

   – La gestion des clés est la phase d'authentification. Elle permet de définir un tunnel sécurisé pour la négociation, nécessaire avant que les paramètres de sécurité IP et les clés ne soient calculés.

   – La gestion des données identifie le type de trafic admis sur un tunnel donné. Elle peut être configurée pour un seul hôte ou un groupe d'hôtes (à l'aide de sous–réseaux ou de plages d'adresses IP) en y associant un protocole et des numéros de port.

   Vous pouvez utiliser le même tunnel de gestion des clés pour protéger plusieurs négociations de gestion de données et rafraîchissements de clés, tant que ces opérations ont lieu entre les mêmes points d'extrémité (par exemple, entre deux passerelles).
6. Pour définir les points d'extrémité du tunnel de gestion de clés, cliquez sur **Tunnels IKE** dans l'onglet Identification.
7. Entrez les informations relatives aux identités des systèmes qui font partie des négociations. Dans la plupart des cas, vous devez utiliser les adresses IP et créer une politique compatible avec la partie distante.

   Dans l'onglet Conversions, utilisez des conversions correspondantes des deux côtés, ou contactez l'administrateur de la partie distante pour définir une conversion correspondante. Vous pouvez créer une conversion qui comporte plusieurs choix pour plus de souplesse lors des opérations de proposition ou de correspondance.
8. Si vous utilisez des clés pré–partagées pour l'authentification, entrez la clé pré–partagée dans l'onglet **Clé**. La valeur doit être identique sur le poste distant et sur le poste local.
9. Créez une conversion à associer à ce tunnel à l'aide du bouton **Ajouter** de l'onglet Conversions.

   Pour activer la prise en charge des certificats numériques et du mode signature, choisissez une méthode d'authentification **Signature RSA** ou **Signature RSA avec vérification des listes CRL**.

   Pour plus d'informations sur les certificats numériques, reportez–vous à la section Utilisation des certificats numériques et du Key Manager, page 12-25.

Configuration des tunnels de gestion des données

Pour configurer les extrémités et les conversions du tunnel de gestion des données et effectuer la configuration du tunnel IKE, ouvrez le Web–based System Manager, en suivant la procédure décrite à la section Configuration de tunnels de gestion des clés, page 12-19. Pour créer un tunnel de gestion des données, procédez comme suit :

1. Sélectionnez un tunnel de gestion des clés puis définissez chaque option unique. Vous pouvez conserver la valeur par défaut de la plupart des options de gestion des données.
2. Vous devez spécifier les types de points d'extrémité tels que l'adresse IP, le sous–réseau ou la plage des adresses IP dans l'onglet Points d'extrémité. Vous pouvez sélectionner un numéro de port et un protocole, ou accepter les valeurs par défaut.
3. Dans l'écran Propositions, vous pouvez créer une nouvelle proposition en cliquant sur le bouton **Ajouter** ou sur **OK**. Si vous utilisez plusieurs propositions, vous pouvez utiliser les boutons Monter ou Descendre pour modifier l'ordre des recherches.

Prise en charge des groupes

Depuis AIX 5.1, la sécurité IP prend en charge le regroupement des ID IKE dans une définition de tunnel, pour associer plusieurs ID à une seule politique de sécurité sans avoir à créer des définitions de tunnels séparées. Le regroupement est particulièrement utile lors de la configuration de connexions à plusieurs hôtes distants, car cela permet d'éviter de configurer ou de gérer plusieurs définitions de tunnels. De plus, si une politique de sécurité doit être modifiée, il n'est pas nécessaire de modifier plusieurs définitions de tunnels.

Un groupe doit être défini avant que son nom ne soit utilisé dans une définition de tunnel. La taille du groupe est limitée à 1 Ko. Du côté de l'appelant de la négociation, vous pouvez utiliser des groupes comme ID distant uniquement dans des définitions de tunnel de gestion des données. Du côté de l'appelé de la négociation, vous pouvez utiliser des groupes comme ID distant dans des définitions de tunnel de gestion des données et de gestion de clés.

Un groupe est composé d'un nom de groupe et d'une liste d'ID IKE et de types d'ID. Les ID peuvent tous être du même type ou un mélange des types suivants :

- Adresses IPv4
- Adresses IPv6
- FQDN
- utilisateur@FQDN
- Types de DN X500

Pendant une négociation de lien de sécurité, les ID dans un groupe sont recherchés dans l'ordre, jusqu'à la première correspondance.

Vous pouvez utiliser Web–based System Manager pour définir un groupe à utiliser comme point d'extrémité distant d'un tunnel de gestion des clés. Pour définir un groupe à l'aide du Web–based System Manager, procédez comme suit :

1. Sélectionnez un tunnel de gestion des clés dans le conteneur **Tunnel IKE**.
2. Ouvrez la boîte de dialogue **Propriétés**.
3. Sélectionnez l'onglet **Identification**.
4. Choisissez **Définition de l'ID groupe** pour le type d'identité de l'hôte distant.
5. Activez le bouton **Configuration de la définition de groupe**, puis entrez ses membres dans la fenêtre.

Reportez–vous à la section Interface de ligne de commande pour la configuration d'un tunnel IKE, page 12-21 pour plus d'informations sur la définition de groupes depuis la ligne de commande.

Utilisation de l'interface SMIT pour la configuration d'un tunnel IKE

L'interface SMIT permet de configurer des tunnels IKE et d'effectuer des fonctions de base sur la base de données IKE. SMIT se sert des fonctions XML pour effectuer des ajouts, des suppressions et des modifications aux définitions de tunnel IKE. SMIT IKE permet de configurer rapidement des tunnels IKE et fournit des exemples de syntaxes XML utilisée pour créer des définitions de tunnel IKE tunnel. Les menus SMIT IKE permettent également de sauvegarder, restaurer et initialiser la base de données IKE.

Pour configurer un tunnel IKE IPv4, utilisez le raccourci **smitty ike4**. Pour configurer un tunnel IKE IPv6, utilisez le raccourci **smitty ike6**. Les fonctions de la base de données IKE se trouvent dans le menu Configuration avancée de la sécurité IP.

Toutes les entrées de la base de données IKE ajoutées via SMIT peuvent être affichées ou modifiées via l'outil Web–based System Manager.

Interface de la ligne de commande pour la configuration d'un tunnel IKE

La commande **ikedb**, disponible dans AIX 5.1 et versions ultérieures, permet de récupérer, de mettre à jour, de supprimer, d'importer et d'exporter des informations dans la base de données IKE à l'aide d'une interface XML. La commande **ikedb** permet d'écrire (ajouter) dans la base de données IKE ou de lire (récupérer) à partir de la base de données IKE. Le format d'entrée et de sortie est un fichier XML (Extensible Markup Language). Le format d'un fichier XML est indiqué par sa DTD (Définition de type de document). La commande **ikedb** permet de voir la DTD utilisée pour valider le fichier XML lors d'un ajout. La seule modification possible sur une DTD est l'ajout de déclarations d'entité, à l'aide de l'indicateur **–e**. Toute déclaration de DOCTYPE externe dans le fichier d'entrée XML sera ignorée et toute déclaration DOCTYPE interne peut engendrer une erreur. Les règles suivies pour interpréter le fichier XML à l'aide de la DTD sont conformes au standard XML. Le fichier **/usr/samples/ipsec** est un exemple de fichier XML typique qui définit les scénarios de tunnel les plus courants. Pour obtenir des détails sur la syntaxe, reportez–vous à la description de la commande **ikedb** dans le manuel *AIX 5L Version 5.3 Commands Reference*.

La commande **ike** permet de démarrer, d'arrêter et de gérer les tunnels IKE. Cette commande permet également d'activer, de supprimer ou de répertorier les tunnels de sécurité IP et IKE. Pour obtenir des détails sur la syntaxe, reportez–vous à la description de la commande **ike** dans le manuel *AIX 5L Version 5.3 Commands Reference*.

Les exemples suivants montrent comment utiliser **ike**, **ikedb** et d'autres commandes afin de configurer et de vérifier l'état de votre tunnel IKE :

1. Pour lancer une négociation de tunnel (*activation* d'un tunnel) ou pour permettre au système de réception d'agir en tant qu'appelé (selon le rôle spécifié), utilisez la commande **ike** associée à un numéro de tunnel :

```
# ike cmd=activate numlist=1
```

   Vous pouvez également utiliser des adresses IP ou des ID distants comme dans l'exemple suivant :

```
# ike cmd=activate remid=9.3.97.256
 # ike cmd=activate ipaddr=9.3.97.100, 9.3.97.256
```

   Le traitement des commandes pouvant prendre plusieurs secondes, le résultat est renvoyé après le début de la négociation.

2. Pour afficher l'état du tunnel, utilisez la commande **ike** de la manière suivante :

```
# ike cmd=list
```

   Le résultat de cette commande se présente comme suit :

```
Phase 1 Tunnel ID       [1]
 Phase 2 Tunnel ID       [1]
```

   Le résultat montre les tunnels des phases 1 et 2 actuellement actifs.

3. Pour obtenir une liste plus détaillée du tunnel, utilisez la commande **ike** de la manière suivante :

```
# ike cmd=list verbose
```

Le résultat de cette commande se présente comme suit :

```
Phase 1 Tunnel ID       1
Local ID Type:          Fully_Qualified_Domain_Name
Local ID:               bee.austin.ibm.com
Remote ID Type:         Fully_Qualified_Domain_Name
Remote ID:              ipsec.austin.ibm.com
Mode:                   Aggressive
Security Policy:        BOTH_AGGR_3DES_MD5
Role:                   Initiator
Encryption Alg:         3DES-CBC
Auth Alg:               Preshared Key
Hash Alg:               MD5
Key Lifetime:           28800 Seconds
Key Lifesize:           0 Kbytes
Key Rem Lifetime:       28737 Seconds
Key Rem Lifesize:       0 Kbytes
Key Refresh Overlap:    5%
Tunnel Lifetime:        2592000 Seconds
Tunnel Lifesize:        0 Kbytes
Tun Rem Lifetime:       2591937 Seconds
Status:                 Activité

Phase 2 Tunnel ID       1
Local ID Type:          IPv4_Address
Local ID:               10.10.10.1
Local Subnet Mask:      N/A
Local Port:             any
Local Protocol:         all
Remote ID Type:         IPv4_Address
Remote ID:              10.10.10.4
Remote Subnet Mask:     N/A
Remote Port:            any
Remote Portocol:        all
Mode:                   Oakley_quick
Security Policy:        ESP_3DES_MD5_SHA_TUNNEL_NO_PFS
Role:                   Initiator
Encryption Alg:         ESP_3DES
AH Transform:           N/A
Auth Alg:               HMAC-MD5
PFS:                    No
SA Lifetime:            600 Seconds
SA Lifesize:            0 Kbytes
SA Rem Lifetime:        562 Seconds
SA Rem Lifesize:        0 Kbytes
Key Refresh Overlap:    15%
Tunnel Lifetime:        2592000 Seconds
Tunnel Lifesize:        0 Kbytes
Tun Rem Lifetime:       2591962 Seconds
Assoc P1 Tunnel:        0
Encap Mode:             ESP_tunnel
Status:                 Active
```

4. Pour afficher les règles de filtrage dans la table de filtre dynamique du tunnel IKE qui vient d'être activé, utilisez la commande **lsfilt** de la manière suivante :

```
# lsfilt -d
```

Le résultat de cette commande se présente comme suit :

```
1 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 no udp eq 4001 eq 4001 both both no all
   packets 0 all
 2 *** Dynamic filter placement rule *** no
 0 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 yes all any 0 any 0 both both no all
   packets 0 all

 *** Dynamic table ***

 0 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 no udp eq 500 eq 500 local both no all
   packets 0
 0 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 no ah any 0 any 0 both inbound no all
   packets 0
 0 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 no esp any 0 any 0 both inbound no all
   packets 0
 1 permit 10.10.10.1 255.255.255.255 10.10.10.4 255.255.255.255 no all any 0 any
   0 both outbound yes all packets 1
 1 permit 10.10.10.4 255.255.255.255 10.10.10.1 255.255.255.255 no all any 0 any
   0 both inbound yes all packets 1
```

Cet exemple présente un poste avec uniquement un tunnel IKE. La règle de l'emplacement du filtre dynamique (règle n°2 dans le résultat de cet exemple de la table statique) peut être déplacée pour contrôler l'emplacement en fonction de toutes les autres règles définies par l'utilisateur. Les règles de la table dynamique sont élaborées automatiquement au fur et à mesure que les tunnels sont négociés et les règles correspondantes sont insérées dans la table de filtrage. Ces règles peuvent être affichées mais ne sont pas modifiables.

5. Pour activer la journalisation des règles de filtrage dynamiques, définissez l'option de journalisation de la règle n°2 sur yes (oui) à l'aide de la commande **chfilt** suivante :

```
# chfilt -v 4 -n 2 -l y
```

Pour plus d'informations sur la journalisation des échanges IKE, reportez–vous à la section Fonctions de journalisation, page 12-50.

6. Pour désactiver le tunnel, utilisez la commande **ike** comme suit :

```
# ike cmd=remove numlist=1
```

7. Pour afficher les définitions du tunnel, utilisez la commande **ikedb** comme suit :

```
# ikedb -g
```

8. Pour insérer des définitions dans la base de données IKE à partir d'un fichier XML généré par un poste homologue, et écraser tous les objets portant le même nom, utilisez la commande **ikedb** comme suit :

```
# ikedb -pFs peer_tunnel_conf.xml
```

Le fichier **peer_tunnel_conf.xml** est le fichier XML généré par un poste homologue.

9. Pour obtenir la définition du tunnel de la phase 1 appelé *tunnel_sys1_and_sys2*, et de tous les tunnels dépendants de la phase 2 avec leurs propositions et protections respectives, utilisez la commande **ikedb** comme suit :

```
# ikedb -gr -t IKETunnel -n tunnel_sys1_and_sys2
```

10. Pour supprimer toutes les clés pré–partagées de la base de données, utilisez la commande **ikedb** comme suit :

```
# ikedb -d -t IKEPresharedKey
```

Pour obtenir des informations générales sur la prise en charge du groupe de tunnel IKE, reportez–vous à la section Prise en charge des groupes, page 12-20. La commande **ikedb** permet de définir des groupes à partir de la ligne de commande.

Ressemblances entre IKE sous AIX et Linux

Pour configurer un tunnel IKE sous AIX à l'aide de fichiers de configuration Linux (AIX versions 5.1 et ultérieures), utilisez la commande **ikedb** avec l'indicateur **–c** (option de conversion), ce qui vous permet d'utiliser les fichiers de configuration Linux **/etc/ipsec.conf** et **/etc/ipsec.secrets** comme définitions de tunnels IKE. La commande **ikedb** analyse les fichiers de configuration Linux, crée un fichier XML et ajoute éventuellement les définitions du tunnel XML à la base de données IKE. Les définitions de tunnels peuvent ensuite être affichées à l'aide de la commande **ikedb –g** ou depuis Web–based System Manager.

Scénarios de configuration d'un tunnel IKE

Les scénarios suivants décrivent le type de situations les plus rencontrées lors de la configuration de tunnels. Ces scénarios peuvent être décrits comme des cas de succursales, de partenaires et d'accès à distance.

- Dans le cas de succursales, le client souhaite connecter ensemble deux réseaux sécurisés : les services d'ingénierie de deux sites différents. Dans cet exemple, des passerelles sont connectées entre elles et tous les échanges qui circulent entre elles utilisent le même tunnel. Quelle que soit l'extrémité du tunnel, les échanges sont décapsulés et transférés en clair sur l'Intranet.

  Dans la première phase de la négociation IKE, le lien de sécurité est créé entre les deux passerelles. Les échanges qui circulent dans le tunnel de sécurité IP se font entre deux sous–réseaux. Les ID des sous–réseaux sont utilisés dans la négociation de phase 2. Un numéro de tunnel est créé dès qu'une politique de sécurité et des paramètres sont créés pour ce tunnel. Utilisez la commande **ike** pour démarrer le tunnel.

- Dans le cadre d'un partenariat, les réseaux ne sont pas sécurisés et l'administrateur peut vouloir limiter l'accès à un petit nombre d'hôtes derrière la passerelle de sécurité. Dans ce cas, le tunnel entre les hôtes transporte le trafic protégé par la sécurité IP et circulant entre deux hôtes donnés. Le protocole du tunnel de phase 2 est AH ou ESP. Ce tunnel hôte à hôte est sécurisé à l'intérieur d'un tunnel passerelle à passerelle.

- Dans le cas de l'accès à distance, les tunnels sont configurés à la demande et un niveau de sécurité élevé est appliqué. Les adresses IP n'étant pas forcément significatives, il est préférable d'adopter des noms de domaines complets ou des adresses de type *utilisateur@*nom_de_domaine_complet. Vous avez la possibilité d'utiliser KEYID pour associer une clé à un ID hôte.

Utilisation des certificats numériques et du Key Manager

Les certificats numériques relient une identité et une clé publique, avec laquelle vous pouvez vérifier l'identité de l'expéditeur ou du destinataire d'un transfert chiffré. A partir de AIX 4.3.3, la sécurité IP utilise les certificats numériques pour activer le *chiffrement par clé publique*, connu également sous le nom de *chiffrement asymétrique* ; cette fonction chiffre les données en utilisant une clé privée connue uniquement par l'utilisateur, et les déchiffre en utilisant une clé publique (partagée) associée, issue d'une paire de clés publique–privée. Les *paires de clés* sont de longues chaînes de données qui sont des clés de chiffrement.

Dans le chiffrement par clé publique, cette clé publique est disponible à toute personne avec laquelle l'utilisateur souhaite communiquer. L'expéditeur appose une signature numérique à toutes les communications à l'aide de la clé privée correspondante. Le destinataire utilise la clé publique pour vérifier la signature de l'expéditeur. Si le message est déchiffré avec succès à l'aide de la clé publique, le destinataire peut vérifier l'authenticité de l'identité de l'expéditeur.

Le chiffrement par clé publique fait appel à des entités tierces accréditées, connues sous le nom d'*autorités d'accréditation* (*CA*), pour l'émission de certificats numériques fiables. C'est le destinataire qui indique quels sont les organisations ou organismes émetteurs accrédités. Le certificat est émis pour une période de temps définie ; après sa date d'expiration, il doit être remplacé.

AIX 4.3.3 et les versions ultérieures incluent l'utilitaire Key Manager, conçu pour la gestion des certificats numériques. Vous trouverez dans les sections suivantes des informations sur les certificats. Les tâches de gestion de ces certificats sont décrites à la section Utilisation des certificats numériques et du Key Manager, page 12-25.

Format des certificats numériques

Le certificat numérique contient des informations spécifiques sur l'identité du propriétaire du certificat et sur l'autorité d'accréditation. Reportez–vous à la figure suivante pour l'illustration d'un certificat numérique.

Figure 10. Contenus d'un certificat numérique. Ce schéma présente les quatre parties d'un certificat numérique. En partant du haut, ce sont : nom spécifique (DN) du propriétaire, clé publique du propriétaire, nom spécifique de l'autorité d'accréditation (CA) et signature du CA.

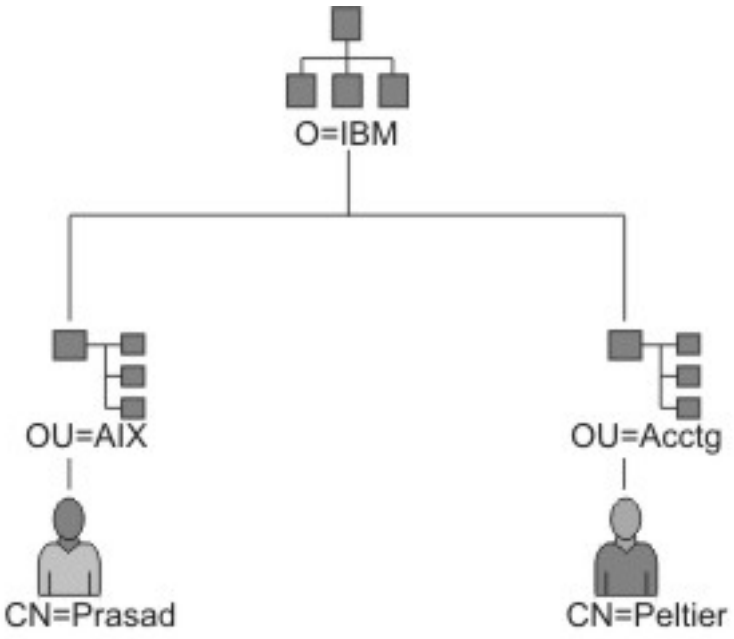
Contents of a Digital Certificate

La liste suivante décrit plus en détail le contenu du certificat numérique :

Nom spécifique (DN) du propriétaire
: Combinaison du nom usuel du propriétaire et du contexte (position) dans l'arborescence de répertoire. Dans l'exemple suivant, qui décrit une simple arborescence de répertoires, Prasad est le nom usuel du propriétaire et le contexte est country (pays) = US, organization (organisation) = ABC, lower organization (unité d'organisation) = SERV ; par conséquent le nom spécifique est :

```
/C=US/O=ABC/OU=SERV/CN=prasad.austin.ibm.com
```

Figure 11. Exemple de Nom spécifique dérivé de l'arborescence de répertoires. Ce schéma est une arborescence de répertoires avec en haut O = ABC et se divisant en deux parties au second niveau. Le second niveau contient OU = AIX et OU = Acctg sur des branches séparées ; chacune d'elles a une branche qui mène à une entité simple du dernier niveau. Le dernier niveau contient respectivement CN=Prasad et CN=Peltier.



Example of Deriving Distinguished Name from Directory Tree

Clé publique du propriétaire
: Utilisée par les destinataires pour déchiffrer les données.

Nom d'utilisateur supplémentaire
: Il s'agit d'un ID tel qu'une adresse IP, une adresse électronique, un nom de domaine complet, etc.

Date d'émission Date à laquelle le certificat numérique a été émis.

Date d'expiration
: Date à laquelle le certificat numérique expire.

Nom spécifique (DN) de l'émetteur
: Nom spécifique de l'autorité d'accréditation (CA).

Signature numérique de l'émetteur
: Signature numérique utilisée pour valider un certificat.

Remarques sur la sécurité des certificats numériques

La présence du certificat numérique n'est pas une preuve d'identité. Le certificat numérique permet uniquement de vérifier l'identité du propriétaire d'un certificat numérique en fournissant une clé publique nécessaire pour contrôler la signature numérique du propriétaire. L'envoi de votre clé publique à un correspondant ne comporte aucun risque car vos données ne peuvent pas être déchiffrées sans l'autre partie de la paire de clés, à savoir la clé privée. Par conséquent, le propriétaire doit protéger la clé privée associée à la clé publique du certificat numérique. Toutes les communications du propriétaire d'un certificat numérique peuvent être déchiffrées si la clé privée est disponible. Sans la clé privée, l'utilisation illicite du certificat numérique est quasiment impossible.

Autorités d'accréditation et hiérarchies sécurisées

Un certificat numérique est aussi fiable que l'autorité d'accréditation (CA) qui l'émet. Il faut donc bien comprendre les politiques d'émission des certificats qui font partie des mécanismes d'accréditation. Chaque organisation ou utilisateur doit déterminer quelles autorités d'accréditation sont fiables.

L'utilitaire Key Manager permet également de créer vos propres certificats auto–signés, qui peuvent être utiles pour les tests ou les environnements composés d'un nombre réduit d'utilisateurs ou de machines.

Si vous utilisez un service de sécurité, vous devez connaître ses clés publiques pour obtenir et valider tout certificat numérique. En outre, le fait de recevoir un certificat numérique ne garantit pas son authenticité. Pour en vérifier l'authenticité, vous devez détenir la clé publique de l'autorité d'accréditation qui a émis ce certificat numérique. Si vous ne possédez pas déjà une copie accréditée de la clé publique, vous devez vous procurer un certificat numérique supplémentaire pour obtenir la clé publique de l'autorité d'accréditation (CA).

Listes de révocation des certificats (CRL)

Un certificat numérique est supposé s'utiliser tout au long de sa période de validité. Cependant, il peut s'avérer nécessaire d'invalider un certificat avant sa date d'expiration. C'est le cas si par exemple, un employé quitte la société ou si la confidentialité d'une clé privée a été compromise. Pour invalider un certificat, vous devez informer l'autorité d'accréditation (CA) appropriée sur les circonstances. Lorsqu'une CA révoque un certificat, elle ajoute son numéro de série à la liste de révocation des certificats (CRL).

Les CRL sont des structures de données signées, délivrées périodiquement et disponibles dans une base de données publique. Les CRL peuvent être récupérées à partir de serveurs HTTP ou LDAP. Chaque CRL contient l'horodate en cours et une horodate **nextUpdate** correspondant à la prochaine mise à niveau. Dans la liste, chaque certificat révoqué est identifié par son numéro de série.

Si vous utilisez les certificats numériques comme méthode d'authentification pour configurer un tunnel IKE, vous pouvez contrôler leur validité en sélectionnant Signature RSA avec vérification des listes CRL. Si la vérification des listes CRL est activée, la liste est localisée et vérifiée lors de la négociation pour établir le tunnel de gestion des clés.

Remarque : Pour pouvoir utiliser cette fonction de sécurité IP, le système doit être configuré pour la prise en charge d'un serveur SOCKS (version 4 pour les serveurs HTTP), d'un serveur LDAP ou des deux. Si vous savez quel type de serveur SOCKS ou LDAP sera utilisé pour obtenir les listes CRL, vous pouvez effectuer la configuration nécessaire avec le Web–based System Manager. Sélectionnez **Configuration CRL** dans le menu Certificats numériques.

Utilisation des certificats numériques dans les applications Internet

Les applications Internet qui utilisent les systèmes de chiffrement à clé publique doivent utiliser les certificats numériques pour obtenir les clés publiques. Diverses applications utilisent le chiffrement par clé publique, y compris :

Virtual Private Networks (VPN)

Les réseaux privés virtuels (VPN) ou *tunnels sécurisés*, peuvent être configurés entre des systèmes tels que les pare–feu pour réaliser des connexions chiffrées entre des réseaux sécurisés, sur des liaisons de communication non sécurisées. Tout le trafic circulant sur ces réseaux et destiné à ces systèmes sera chiffré.

Les protocoles utilisés dans la configuration des tunnels répondent aux normes de la sécurité IP et IKE, ce qui permet d'obtenir une connexion chiffrée entre un client distant (par exemple, une personne travaillant en télétravail) et un hôte ou réseau sécurisé.

Secure Sockets Layer (SSL)

SSL est un protocole de sécurité qui assure la confidentialité et l'intégrité des communications. Il est utilisé par les serveurs Web pour sécuriser leurs connexions avec les navigateurs Web ; par les serveurs LDAP (Lightweight Directory Access Protocol) pour sécuriser leurs connexions avec leurs clients LDAP ; et par les serveurs Host–on–Demand V.2 pour les connexions entre le client et le système hôte. Le protocole SSL utilise les certificats numériques pour l'échange des clés, l'authentification du serveur et éventuellement, pour l'authentification du client.

Secure Electronic Mail

Pour chiffrer et déchiffrer les messages électroniques, les systèmes de messages sécurisés, basés sur des normes telles que PEM ou S/MIME, utilisent les certificats numériques pour les signatures numériques et l'échange de clés.

Certificats numériques et demandes de certificats

Un certificat numérique signé contient les parties suivantes : nom spécifique (DN) du propriétaire, clé publique du propriétaire, nom spécifique de l'autorité d'accréditation (CA) et signature du CA. Un certificat numérique auto–signé contient le nom spécifique, la clé publique et la signature de son propriétaire.

Pour demander un certificat numérique, vous devez créer une *demande de certificat* et l'envoyer à une autorité d'accréditation (CA). La demande de certificat contient les parties suivantes : nom spécifique, clé publique et signature du demandeur. Le CA vérifie la signature du demandeur avec la clé publique du certificat numérique pour s'assurer des conditions suivantes :

- La demande de certificat n'a pas été modifiée lors du transit entre le demandeur et le CA.
- Le demandeur possède la clé privée correspondant à la clé publique incluse dans la demande de certificat.

Le CA est également responsable de la vérification de certains aspects concernant l'identité du demandeur. Ce type de vérification peut varier d'une certitude quasi nulle à l'assurance absolue de l'identité du propriétaire.

Utilitaire Key Manager

L'utilitaire Key Manager gère les certificats numériques, et se trouve dans le fichier **gskkm.rte** sur l'Expansion Pack.

Cette section décrit comment utiliser Key Manager pour effectuer les tâches suivantes :

1. Création d'une base de données de clés, page 12-30
2. Ajout de certificat numérique root d'une autorité d'accréditation, page 12-31
3. Etablissement de paramètres sécurisés, page 12-32
4. Suppression de certificat numérique root d'une autorité d'accréditation, page 12-32
5. Demande de certificat numérique, page 12-33
6. Ajout (Réception) d'un nouveau certificat numérique, page 12-33
7. Suppression d'un certificat numérique, page 12-34
8. Modification de mot de passe de la base de données, page 12-35
9. Création de tunnels IKE avec certificats numériques, page 12-35

Vous devez effectuer les tâches 1, 2, 3, 4, 6 et 7 pour définir la prise en charge des certificats et des signatures numériques. Utilisez ensuite Web–based System Manager pour créer un tunnel IKE et associer une politique au tunnel qui utilise la méthode d'authentification Signature RSA.

Vous pouvez créer et configurer une base de données de clés à partir de la fenêtre Présentation du Web–based System Manager VPN en sélectionnant l'option **Gestion des certificats numériques** ou en utilisant la commande **certmgr** pour ouvrir l'utilitaire Key Manager depuis la ligne de commande.

Création d'une base de données de clés

Une base de données de clés autorise la connexion des points d'extrémité d'un VPN, à l'aide de certificats numériques valides. Les VPN d'IPsec utilisent le format de base de données de clé *.kdb.

Les types de certificats d'autorité d'accréditation suivants sont fournis avec Key Manager :

- Autorité d'accréditation RSA Secure Server
- Autorité d'accréditation Thawte Personal Premium
- Autorité d'accréditation Thawte Personal Freemail
- Autorité d'accréditation Thawte Personal Basic
- Autorité d'accréditation Thawte Personal Server
- Autorité d'accréditation Thawte Server
- Autorité d'accréditation primaire Verisign de classe 1
- Autorité d'accréditation primaire Verisign de classe 2
- Autorité d'accréditation primaire Verisign de classe 3
- Autorité d'accréditation primaire Verisign de classe 4

Les signatures de ces certificats numériques permettent aux clients de se connecter aux serveurs qui possèdent des certificats numériques autorisés. Après avoir créé une base de données de clés, vous pouvez l'utiliser pour vous connecter à un serveur possédant un certificat numérique signé par l'une de ces autorités d'accréditation.

Pour utiliser un certificat numérique de signature qui n'est pas sur cette liste, vous devez en faire la demande auprès de l'autorité d'accréditation et l'ajouter à votre base de données de clés. Reportez–vous à la section Ajout de certificat numérique root d'une autorité d'accréditation, page 12-31.

Pour créer une base de données de clés à l'aide de la commande **certmgr**, procédez comme suit :

1. Démarrez l'utilitaire Key Manager en saisissant :

```
# certmgr
```

2. Sélectionnez **Nouveau** (New) à partir du menu déroulant Fichier de base de données de clés (Key Database File).
3. Dans la zone **Type de base de données de clés**, choisissez la valeur par défaut **Fichier de base de données de clés CMS**.
4. Entrez ce nom dans la zone **Nom de fichier** :

```
ikekey.kdb
```

5. Entrez le chemin de la base de données dans la zone **Emplacement** (Location) :

```
/etc/security
```

   Remarque : La base de données de clés doit être nommée **ikekey.kbd** et doit se trouver dans le répertoire **/etc/security**. Sinon, la sécurité IP ne fonctionne pas correctement.

6. Cliquez sur **OK**. L'invite **Mot de passe** s'affiche.
7. Entrez un **mot de passe** dans la zone correspondante, puis une nouvelle fois dans la zone **Confirmation du mot de passe**.

8. Si vous voulez changer le nombre de jours avant l'expiration du mot de passe, saisissez le nombre de jours souhaités dans la zone **Définir le délai d'expiration ?**. La valeur par défaut est 60 jours. Si souhaitez que le mot de passe n'expire jamais, laissez vide la zone **Définir le délai d'expiration ?**
9. Pour sauvegarder une version chiffrée du mot de passe dans un fichier cachette, sélectionnez **Enregistrer le mot de passe dans un fichier cachette ?** (Stash the password to a file ?), et entrez–y **oui** (yes)

   Remarque : Vous devez mettre en cache le mot de passe pour pouvoir utiliser les certificats numériques avec IPSec.
10. Cliquez sur **OK**. Un écran de confirmation s'affiche pour vous permettre de vérifier que vous avez créé une base de données de clés.
11. Cliquez à nouveau sur **OK** pour voir s'afficher l'écran principal Gestion des clés. Vous pouvez alors exécuter une autre tâche ou quitter l'utilitaire.

Ajout de certificat numérique root d'une autorité d'accréditation

Une fois reçu le certificat numérique root de l'autorité d'accréditation, ajoutez–le à votre base de données. La plupart des certificats numériques root sont de type *.arm, comme suit :

```
cert.arm
```

Pour ajouter un certificat root d'autorité d'accréditation à une base de données, procédez de la manière suivante :

1. Si Key Manager n'est pas démarré, faites–le en saisissant :

```
# certmgr
```

2. Dans l'écran principal, sélectionnez **Ouvrir** dans le menu déroulant Fichier de la base de données de clés.
3. Sélectionnez le fichier auquel vous souhaitez ajouter le certificat root d'autorité d'accréditation, puis cliquez sur **Ouvrir**.
4. Entrez le mot de passe, puis cliquez sur **OK**. Une fois le mot de passe accepté, l'écran principal Gestion des clés s'affiche. La barre de titre affiche le nom du fichier de la base de données de clés sélectionné, et indique qu'il est ouvert et prêt à être utilisé.
5. Sélectionnez **Certificats accrédités** à partir du menu déroulant Certificats accrédités/personnels.
6. Cliquez sur **Ajouter** (Add).
7. Sélectionnez un type de données à partir du menu déroulant correspondant, par exemple :

```
Base64-encoded ASCII data
```

8. Entrez un nom de fichier de certification et un chemin d'accès pour le certificat root d'autorité d'accréditation ou cliquez sur **Parcourir** (Browse) pour le sélectionner.
9. Cliquez sur **OK**.
10. Entrez le libellé du certificat root d'autorité d'accréditation, par exemple `Tester un certificat root d'autorité d'accréditation` (Test CA Root Certificate), puis cliquez sur **OK**. Vous êtes ramené à l'écran principal `Gestion des clés`. Le champ **Certificats accrédités** affiche désormais le libellé du certificat root d'autorité d'accréditation que vous venez d'ajouter. Vous pouvez alors exécuter d'autres tâches ou quitter l'utilitaire.

Etablissement de paramètres sécurisés

Les certificats d'autorité d'accréditation sont définis par défaut sur **sécurisé** (trusted). Pour modifier cette valeur, procédez comme suit :

1. Si Key Manager n'est pas démarré, faites–le en saisissant :

```
# certmgr
```

2. Dans l'écran principal, sélectionnez **Ouvrir** dans le menu déroulant Fichier de la base de données de clés.
3. Sélectionnez le fichier de base de données de clés dont vous souhaitez modifier la valeur de certificat numérique par défaut, puis cliquez sur **Ouvrir**.
4. Entrez le mot de passe, puis cliquez sur **OK**. Une fois le mot de passe accepté, l'écran principal Gestion des clés s'affiche. La barre de titre affiche le nom du fichier de la base de données de clés sélectionné, indiquant que le fichier est ouvert.
5. Sélectionnez **Certificats accrédités** à partir du menu déroulant Certificats accrédités/personnels.
6. Sélectionnez le certificat que vous souhaitez modifier, puis cliquez sur **Affichage/Modification** (View/Edit) ou double–cliquez sur l'entrée. L'écran Information sur les clés s'affiche pour le certificat sélectionné.
7. Pour qu'il devienne un certificat root sécurisé, cochez la case située à côté de **Définir le certificat en tant que root sécurisée** (Set the certificate as a trusted root) puis cliquez sur **OK**. Si le certificat n'est pas sécurisé, décochez la case puis cliquez sur **OK**.
8. Cliquez sur **OK** dans l'écran Certificats accrédités. Vous êtes ramené à l'écran principal Gestion des clés Vous pouvez alors exécuter une autre tâche ou quitter l'utilitaire.

Suppression de certificat numérique root d'une autorité d'accréditation

Si vous ne souhaitez plus prendre en compte l'une des autorités d'accréditation de la liste des certificats numériques de signature, vous devez supprimer son certificat root d'autorité d'accréditation.

> **Remarque :** Avant de supprimer un certificat root d'autorité d'accréditation, créez une copie de sauvegarde au cas où vous souhaiteriez recréer la root de l'autorité d'accréditation.

Pour supprimer de la base de données un certificat root d'autorité d'accréditation, procédez comme suit :

1. Si Key Manager n'est pas démarré, faites–le en saisissant :

```
# certmgr
```

2. Dans l'écran principal, sélectionnez **Ouvrir** du menu déroulant `Fichier de la base de données des clés`.
3. Sélectionnez le fichier de base de données de clés à partir duquel vous souhaitez supprimer le certificat root d'autorité d'accréditation, puis cliquez sur **Ouvrir**.
4. Entrez le mot de passe, puis cliquez sur **OK**. Une fois le mot de passe accepté, l'écran principal **Gestion des clés** s'affiche. La barre de titre affiche le nom du fichier de la base de données de clés sélectionné, et indique qu'il est ouvert et prêt à être modifié.
5. Sélectionnez **Certificats accrédités** à partir du menu déroulant `Certificats accrédités/personnels`.
6. Sélectionnez le certificat que vous souhaitez supprimer, puis cliquez sur **Supprimer** (Delete). L'écran `Confirmer votre choix ?` (Confirm) s'affiche.
7. Cliquez sur **Oui** (Yes). Vous êtes ramené à l'écran principal Gestion des clés. Le libellé du certificat root d'autorité d'accréditation n'apparaît plus dans la zone **Certificats accrédités**. Vous pouvez alors exécuter une autre tâche ou quitter l'utilitaire.

Demande de certificat numérique

Pour obtenir un certificat numérique, générez une demande à l'aide de Key Manager et soumettez–la à une autorité d'accréditation. Le fichier de demande est au format PKCS#10. L'autorité d'accréditation vérifie votre identité, puis vous envoie un certificat numérique.

Pour demander un certificat numérique, procédez comme suit :

1. Si Key Manager n'est pas démarré, faites–le en saisissant :

```
# certmgr
```

2. Dans l'écran principal, sélectionnez **Ouvrir** dans le menu déroulant Fichier de la base de données de clés.
3. Sélectionnez le fichier de la base de données de clés **/etc/security/ikekey.kdb** à partir duquel vous souhaitez générer la demande, puis cliquez sur **Ouvrir**.
4. Entrez le mot de passe, puis cliquez sur **OK**. Une fois le mot de passe accepté, l'écran principal Gestion des clés s'affiche. La barre de titre affiche le nom du fichier de la base de données de clés sélectionné, et indique qu'il est ouvert et prêt à être modifié.
5. Sélectionnez **Demandes de certificat personnel** (Personal Certificate Requests) depuis le menu déroulant Certificats accrédités/personnels (dans la version 4 de AIX) ou sélectionnez **Création —> Nouvelle demande de certificat** (à partir de AIX 5.1).
6. Cliquez sur **Nouveau**.
7. A partir de l'écran qui s'affiche, saisissez un **libellé de clé** (Key Label) pour le certificat numérique auto–signé, par exemple :

```
cletest
```

8. Entrez un **Nom** (Common Name, le nom hôte par défaut) et une **Organisation**, puis sélectionnez un **Pays** (Country). Pour les zones restantes, vous pouvez accepter la valeur par défaut ou choisir d'autres valeurs.
9. Définissez un nom d'**utilisateur supplémentaire** (Subject Alternate). Les zones en option associées au nom **d'utilisateur supplémentaire** sont l'adresse électronique, l'adresse IP et le nom DNS. L'adresse IP d'un type tunnel doit être identique à celle du tunnel IKE. Pour un *utilisateur@FQDN* de type ID de tunnel, complétez la zone de l'adresse électronique. Pour un FQDN de type ID de tunnel, entrez un nom de domaine complet (par exemple, *nomhôte.nomsociété*.com) dans la zone de nom DNS.
10. En bas de l'écran, entrez un nom pour le fichier, par exemple :

```
certreq.arm
```

11. Cliquez sur **OK**. Un écran de confirmation s'affiche pour vous permettre de vérifier que vous avez créé une demande pour un nouveau certificat numérique.
12. Cliquez sur **OK**. Vous êtes ramené à l'écran principal Gestion des clés. La zone **Demandes de certificat personnel** affiche le libellé de clé de la demande de nouveau certificat numérique (PKCS#10).
13. Envoyez le fichier de demande à l'autorité d'accréditation. Vous pouvez alors exécuter une autre tâche ou quitter l'utilitaire.

Ajout (Réception) d'un nouveau certificat numérique

Lorsque vous recevez le nouveau certificat numérique de l'autorité d'accréditation, vous devez l'ajouter à la base de données de clés qui a servi à générer votre demande.

Pour ajouter (recevoir) un nouveau certificat numérique, procédez comme suit :

1. Si Key Manager n'est pas démarré, faites–le en saisissant :

```
# certmgr
```

2. Dans l'écran principal, sélectionnez **Ouvrir** dans le menu déroulant Fichier de la base de données de clés.

3. Sélectionnez le fichier de la base de données de clés à partir duquel vous avez généré la demande de certificat, puis cliquez sur **Ouvrir**.
4. Entrez le mot de passe, puis cliquez sur **OK**. Une fois le mot de passe accepté, l'écran principal Gestion des clés s'affiche. La barre de titre affiche le nom du fichier de la base de données de clés sélectionné, et indique qu'il est ouvert et prêt à être modifié.
5. Sélectionnez **Demandes de certificat personnel** à partir du menu déroulant Certificats accrédités/personnels.
6. Cliquez sur **Réception** (Receive) pour ajouter à la base de données le nouveau certificat numérique reçu.
7. Sélectionnez le type de données du nouveau certificat numérique à partir du menu déroulant **Type de données** (Data type). La valeur par défaut est **Base64–encoded ASCII data**.
8. Entrez le nom du fichier de certification et le chemin d'accès du nouveau certificat numérique ou cliquez sur **Parcourir** pour le sélectionner.
9. Cliquez sur **OK**.
10. Entrez un libellé descriptif pour le nouveau certificat numérique, par exemple :

```
VPN Branch Certificate
```

11. Cliquez sur **OK**. Vous êtes ramené à l'écran principal Gestion des clés. La zone **Certificats personnels** affiche désormais le libellé du nouveau certificat numérique que vous avez ajouté. Vous pouvez alors exécuter une autre tâche ou quitter l'utilitaire.

    Si une erreur se produit lors du chargement du certificat, vérifiez que le fichier de certificat commence par `-----BEGIN CERTIFICATE-----` et se termine par `-----END CERTIFICATE-----`.

    Par exemple :

```
-----BEGIN CERTIFICATE-----
ajdkfjaldfwwwwwwwwwwadafdw
kajf;kdsajkflasasfkjafdaff
akdjf;ldasjkf;safdfdasfdas
kaj;fdljk98dafdas43adfadfa
-----END CERTIFICATE-----
```

    Si ce texte ne figure pas, modifiez le fichier de manière à ce qu'il commence et finisse correctement.

Suppression d'un certificat numérique

Remarque : Avant de supprimer un certificat numérique, créez une copie de sauvegarde au cas où vous souhaiteriez le créer à nouveau.

Pour supprimer de la base de données un certificat numérique, procédez comme suit :

1. Si Key Manager n'est pas démarré, faites–le en saisissant :

```
# certmgr
```

2. Dans l'écran principal, sélectionnez **Ouvrir** dans le menu déroulant Fichier de la base de données de clés.
3. Sélectionnez le fichier de base de données de clés à partir duquel vous souhaitez supprimer le certificat numérique, puis cliquez sur **Ouvrir**.
4. Entrez le mot de passe, puis cliquez sur **OK**. Une fois le mot de passe accepté, l'écran principal Gestion des clés s'affiche. La barre de titre affiche le nom du fichier de la base de données de clés sélectionné, et indique qu'il est ouvert et prêt à être modifié.
5. Sélectionnez **Demandes de certificat personnel** à partir du menu déroulant Certificats accrédités/personnels.

6. Sélectionnez le certificat que vous souhaitez supprimer, puis cliquez sur **Supprimer** (Delete). L'écran Confirmer votre choix ? s'affiche.
7. Cliquez sur **Oui** (yes). Vous êtes ramené à l'écran principal Gestion des clés Le libellé du certificat numérique que vous venez de supprimer n'apparaît plus dans la zone **Certificats personnels**. Vous pouvez alors exécuter une autre tâche ou quitter l'utilitaire.

Modification de mot de passe de la base de données

Pour modifier une base de données de clés, procédez comme suit :

1. Si Key Manager n'est pas démarré, faites–le en saisissant :

```
# certmgr
```

2. Dans l'écran principal, sélectionnez **Modification du mot de passe** (Change Password) dans le menu déroulant Fichier de la base de données de clés
3. Entrez un nouveau **mot de passe** dans la zone correspondante, puis une nouvelle fois dans la zone **Confirmation du mot de passe**.
4. Si vous voulez changer le nombre de jours avant l'expiration du mot de passe, changez–le dans la zone **Définir le délai d'expiration ?**. La valeur par défaut est 60 jours Si souhaitez que le mot de passe n'expire jamais, laissez vide la zone **Définir le délai d'expiration ?**
5. Pour sauvegarder une version chiffrée du mot de passe dans un fichier cachette, sélectionnez **Enregistrer le mot de passe dans un fichier cachette ?** (Stash the password to a file ?), et entrez–y **oui** (yes)

   Remarque : Vous devez mettre en cache le mot de passe pour pouvoir utiliser les certificats numériques avec IPSec.
6. Cliquez sur **OK**. Un message dans la barre d'état indique que la demande a été prise en compte.
7. Cliquez à nouveau sur **OK** pour voir s'afficher l'écran principal Gestion des clés. Vous pouvez alors exécuter une autre tâche ou quitter l'utilitaire.

Création de tunnels IKE avec certificats numériques

Pour créer des tunnels IKE utilisant des certificats numériques, vous devez vous servir du Web–based System Manager et de l'utilitaire Key Manager.

Pour activer l'utilisation de certificats numériques lors de la définition des politiques de gestion des clés du tunnel IKE, vous devez configurer une conversion qui utilise un mode de signature. Un *Mode de signature* utilise un algorithme RSA de signature pour l'authentification. IPsec fournit la boîte de dialogue de Web–based System Manager "Ajout/Modification d'une conversion" permettant de sélectionner une méthode d'authentification de signature RSA, ou de signature RSA avec vérification des listes CRL.

Au moins une extrémité du tunnel doit posséder une politique utilisant une conversion de mode de signature. Vous pouvez également définir d'autres conversions utilisant le mode de signature avec Web–based System Manager.

Les types de tunnels de gestion des clés IKE (zone **Type d'identité de l'hôte** sur l'onglet Identification) pris en charge par IPsec sont les suivants :

- adresse IP
- FQDN (nom de domaine complet)
- *utilisateur@ FQDN*
- Nom distinctif X.500
- Identificateur de la clé

Utilisez **Web–based System Manager** pour sélectionner les types d'identité de l'hôte dans l'onglet Identification – Propriétés du tunnel de gestion des clés. Si vous sélectionnez **Adresse IP, FQDN** ou **user@FQDN**, vous devez entrer les valeurs dans Web–based System Manager et les transmettre à l'autorité d'accréditation. Ces informations sont utilisées pour le Nom d'utilisateur supplémentaire dans le certificat numérique personnel.

Par exemple, si vous choisissez un type d'identité hôte **Nom distinctif X.500** dans la liste déroulante Web–based System Manager de l'onglet **Identification** et que vous entrez la chaîne **/C=US/O=ABC/OU=SERV/CN= *nom*.austin.ibm.com** dans la zone **Identité de l'hôte**, vous devez entrer les valeurs suivantes dans Key Manager lorsque vous créez une demande de certificat numérique :

- Nom : ***nom*.austin.ibm.com**
- Organisation : **ABC**
- Unité d'organisation : **SERV**
- Pays : **Américain**

La valeur de la zone **Nom distinctif X.500** doit correspondre au nom donné lors de la configuration par votre système ou administrateur LDAP. La valeur de l'unité d'organisation est facultative. L'autorité d'accréditation utilise alors cette information lors de la création de certificat numérique.

Autre exemple : si vous choisissez le type d'identité hôte **IP Address** à partir de la liste déroulante et une valeur de **10.10.10.1** pour l'identité hôte, vous devez entrer les valeurs suivantes pour votre demande de certificat numérique :

- Nom : ***name*.austin.ibm.com**
- Organisation : **ABC**
- Unité d'organisation : **SERV**
- Pays : **Américain**
- Champ Subject alternate IP address (adresse IP du nom supplémentaire) : **10.10.10.1**

Une fois ces informations entrées dans le certificat numérique, l'autorité d'accréditation les utilise pour créer un certificat numérique personnel.

Lors d'une demande de certificat numérique, l'autorité d'accréditation a besoin des informations suivantes :

- Vous demandez un certificat X.509.
- Le format de signature MD5 avec chiffrement RSA.
- Si vous indiquez ou non un nom d'utilisateur supplémentaire.
  Les types de noms supplémentaires sont :
  - adresse IP
  - FQDN (Fully qualified domain name)
  - *utilisateur@FQDN*

  Les informations suivantes relatives au nom d'utilisateur supplémentaire sont comprises dans le fichier de demande de certificat.
- L'utilisation prévue pour la clé (le bit de signature numérique doit être défini).
- Le fichier de demande de certificat numérique de Key Manager (au format PKCS#10).

Pour les tâches nécessitant l'utilisation du Key Manager pour créer une demande de certificat, reportez–vous à la section Demande de certificat numérique, page 12-33.

Avant d'activer le tunnel IKE, vous devez ajouter le certificat numérique personnel reçu de l'autorité d'accréditation dans la base de données de Key Manager, **ikekey.kdb**. Pour plus d'informations, reportez–vous à la section Ajout (Réception) d'un nouveau certificat numérique, page 12-33.

La sécurité IP prend en charge les types suivants de certificats numériques personnels :

DN du sujet
: Le nom spécifique de l'utilisateur doit respecter le format et l'ordre suivants :

  `/C=US/O=ABC/OU=SERV/CN=` *`name`* `.austin.ibm.com`

  L'utilitaire Key Manager n'autorise qu'une seule valeur **OU**.

DN du sujet et Nom d'utilisateur supplémentaire comme valeur d'adresse IP
: Le nom spécifique et le nom supplémentaire de l'utilisateur peuvent constituer l'adresse IP, comme dans l'exemple suivant :

  `/C=US/O=ABC/OU=SERV/CN=nom.austin.ibm.com` et `10.10.10.1`

DN du sujet et Nom d'utilisateur supplémentaire comme adresse FQDN
: Le nom spécifique et le nom supplémentaire de l'utilisateur peuvent constituer un nom de domaine complet, comme dans l'exemple suivant :

  `/C=US/O=ABC/OU=SERV/CN=nom.austin.ibm.com` et `bell.austin.ibm.com`.

DN du sujet et Nom d'utilisateur supplémentaire comme *utilisateur@ FQDN*
: Le nom spécifique et le nom d'utilisateur supplémentaire peuvent constituer une adresse utilisateur (*ID_utilisateur@ nom_de_domaine_complet*), comme dans l'exemple suivant :

  `/C=US/O=ABC/OU=SERV/CN=nom.austin.ibm.com` et `nom@austin.ibm.com`.

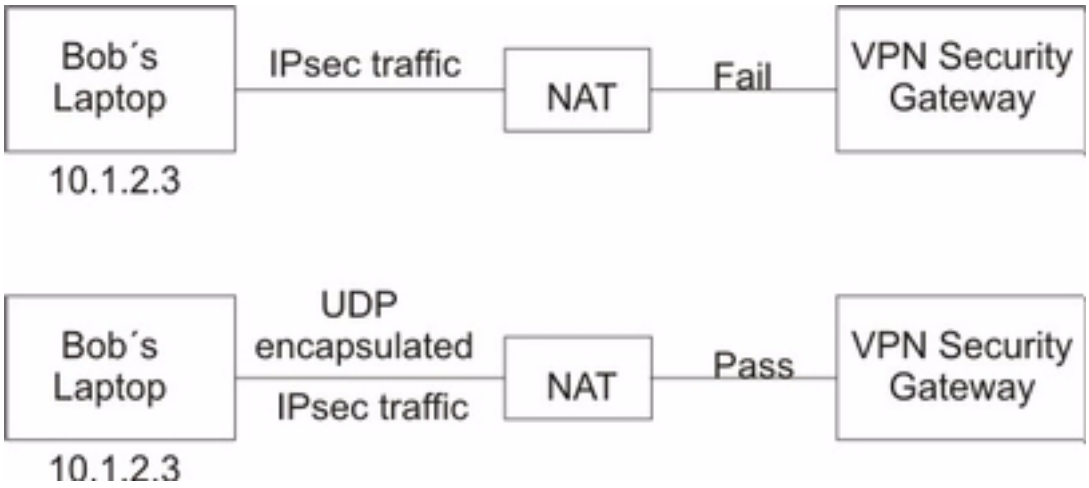
DN du sujet et plusieurs Noms d'utilisateurs supplémentaires
: Le nom spécifique peut être associé à plusieurs noms d'utilisateurs supplémentaires, comme dans l'exemple suivant :

  `/C=US/O=ABC/OU=SERV/CN=nom.austin.ibm.com` et `bell.austin.ibm.com`, `10.10.10.1`, et `utilisateur@nom.austin.ibm.com`.

Utilisation de la traduction d'adresses de réseau

L'implémentation de la sécurité IP dans AIX 5.3 comprend la prise en charge d'unités dont les adresses sont concernées par la traduction d'adresses de réseau (Network Address Translation) (NAT). L'utilisation de la traduction d'adresses de réseau (NAT) est largement répandue dans le cadre de la technologie de pare–feu pour le partage de connexion internet. Il s'agit d'une fonction standard sur les routeurs et les équipements de périphérie (edge devices). Le protocole de sécurité IP repose sur l'identification des points d'extrémité distants et leur stratégie basée sur l'adresse IP distante. Lorsque des unités intermédiaires comme les routeurs et les pare–feux traduisent une adresse privée en adresse publique, le processus d'authentification requis dans le protocole de sécurité IP peut échouer parce que l'adresse dans le paquet IP a été modifiée après le pré–traitement d'authentification. Avec le nouveau support NAT de sécurité IP, les unités configurées derrière un nœud qui effectue la traduction d'adresses de réseau sont en mesure d'établir un tunnel de sécurité IP. Le code de sécurité IP est en mesure de déterminer quand une adresse distante a été traduite. A l'aide de la nouvelle implémentation de sécurité IP, qui intègre la prise en charge de NAT, un client VPN (Virtual Private Network) peut se connecter de son domicile ou sur le trajet du travail via une connexion internet, avec activation de NAT.

Figure 12. Sécurité IP compatible NAT Ce schéma montre la différence entre une installation de sécurité IP compatible NAT avec trafic UDP encapsulé et une installation non compatible NAT.



Configuration de la sécurité IP pour un fonctionnement avec NAT

Afin d'utiliser le support NAT dans la sécurité IP, vous devez définir la variable *ENABLE_IPSEC_NAT_TRAVERSAL* dans le fichier **/etc/isakmpd.conf**. Lorsque cette variable est réglée, les règles de filtrage sont ajoutées pour envoyer et recevoir le trafic sur le port 4500. L'exemple suivant montre les règles de filtrage lorsque la variable *ENABLE_IPSEC_NAT_TRAVERSAL* est définie.

```
Dynamic rule 2:
Rule action          : permit
Source Address       : 0.0.0.0 (any)
Source Mask          : 0.0.0.0 (any)
Destination Address  : 0.0.0.0 (any)
Destination Mask     : 0.0.0.0 (any)
Source Routing       : no
Protocol             : udp
Source Port          : 0 (any)
Destination Port     : 4500
Scope                : local
Direction            : inbound
Fragment control     : all packets
Tunnel ID number     : 0

Dynamic rule 3:
Rule action          : permit
Source Address       : 0.0.0.0 (any)
Source Mask          : 0.0.0.0 (any)
Destination Address  : 0.0.0.0 (any)
Destination Mask     : 0.0.0.0 (any)
Source Routing       : no
Protocol             : udp
Source Port          : 4500
Destination Port     : 0 (any)
Scope                : local
Direction            : outbound
Fragment control     : all packets
Tunnel ID number     : 0
```

Le réglage de la variable *ENABLE_IPSEC_NAT_TRAVERSAL* ajoute également des règles de filtrage dans la table de filtres. Les messages spéciaux IPSEC NAT utilisent l'encapsulation UDP et des règles de filtrage doivent être ajoutées pour assurer la circulation du trafic. De plus, en phase 1, le mode signature est requis. Si l'adresse IP est utilisée comme identifiant dans le certificat, il doit contenir l'adresse IP privée.

La sécurité IP doit aussi envoyer les messages keepalive NAT pour gérer le mappage de l'adresse IP d'origine et l'adresse NAT. L'intervalle est définie par la variable *NAT_KEEPALIVE_INTERVAL* dans le fichier **/etc/isakmpd.conf**. Cette variable indique la fréquence d'envoi (en secondes) des paquets keepalive NAT. Si vous n'indiquez aucune valeur pour *NAT_KEEPALIVE_INTERVAL*, une valeur par défaut égale à 20 secondes est utilisée.

Limitations lors de l'utilisation d'échanges NAT

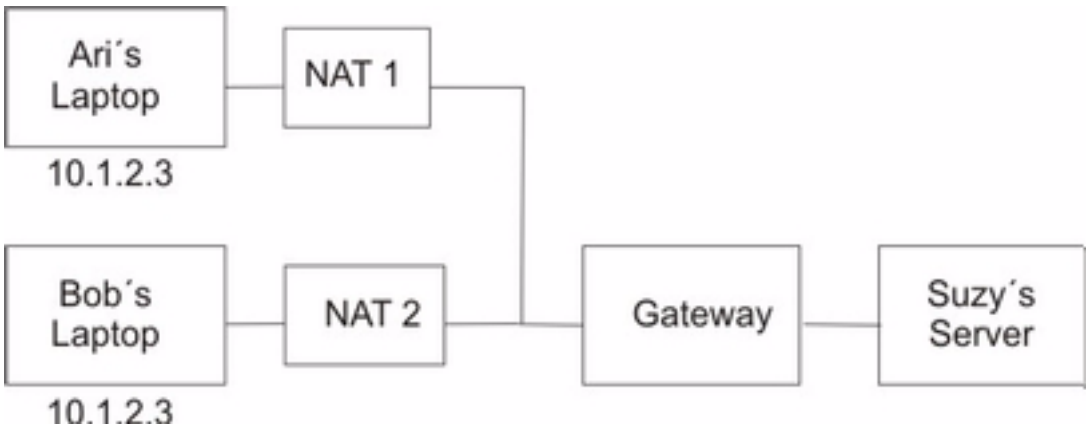
Les points d'extrémité derrière les unités de traduction d'adresses de réseau (NAT) doivent protéger leur trafic à l'aide du protocole ESP. ESP est la principale en–tête sélectionnée pour la sécurité IP et pourra être utilisée pour la majorité des applications client. ESP comprend le hachage des données utilisateur, mais pas de l'en–tête IP. La vérification d'intégrité dans l'en–tête AH inclut les adresses IP source et destination dans le contrôle d'intégrité du message codé. Les unités NAT ou NAT inversé qui opèrent des changements sur les champs d'adresse désactivent le contrôle d'intégrité du message. Par conséquent, si seul le protocole AH est défini dans la stratégie de phase 2 pour un tunnel, et si NAT est détecté au cours d'un échange en phase 1, une charge utile de notification déclarant `NO_PROPOSAL_CHOSEN` est transmise.

De plus, une connexion utilisant NAT doit sélectionner le mode tunnel de sorte que l'adresse IP d'origine soit encapsulée dans le paquet. Le mode Transport et les adresses avec NAT ne sont pas compatibles. Si un NAT est détecté et que seul le mode Transport est proposé en phase 2, une charge utile de notification déclarant `NO_PROPOSAL_CHOSEN` est transmise.

Eviter les conflits de mode tunnel

Les associés distants peuvent négocier des entrées qui se chevauchent dans une passerelle. Ce chevauchement entraîne un conflit de mode Tunnel. La figure suivante montre un conflit de mode Tunnel.

Figure 13. Conflit en mode tunnel Ce schéma montre un exemple de conflit en mode tunnel



La passerelle possède deux associés de sécurité (SA) possibles pour l'adresse IP 10.1.2.3. Ces adresses distantes dupliquées sont à l'origine de confusion quant à la destination des paquets provenant du serveur. Si un tunnel est configuré entre le serveur de Suzy et l'ordinateur portable de Ari, l'adresse IP est utilisée et Suzy ne peut pas configurer un tunnel avec Bob avec la même adresse IP. Pour éviter tout conflit de mode Tunnel, il faut éviter de définir un tunnel avec la même adresse IP. Etant donné que l'adresse distante n'est pas sous le contrôle de l'utilisateur distant, d'autres types d'ID peuvent être utilisés pour identifier l'hôte distant en tant que nom de domaine qualifié complet ou en tant qu'utilisateur@nom_de_domaine_qualifié_complet.

Configuration des tunnels manuels

Les procédures suivantes permettent de configurer la sécurité IP pour l'utilisation des tunnels manuels.

Configuration des tunnels et des filtres

Pour installer un tunnel manuel, il n'est pas nécessaire de configurer séparément les règles de filtrage. Si tout le trafic échangé entre les deux hôtes passe par le tunnel, les règles de filtrage requises sont générées automatiquement. La procédure de configuration d'un tunnel consiste à définir le tunnel à une extrémité, importer la définition à l'autre extrémité, et à activer le tunnel et les règles de filtrage aux deux extrémités. Le tunnel est alors prêt à l'utilisation.

Les informations concernant le tunnel doivent être identiques aux deux extrémités si elles ne sont pas fournies de manière explicite. A titre d'exemple, les algorithmes de chiffrement et d'authentification spécifiés pour l'adresse source seront utilisés pour l'adresse de destination si les valeurs de destination ne sont pas spécifiées.

Création d'un tunnel manuel sur le premier hôte

Vous pouvez configurer un tunnel à l'aide de l'application Web–based System Manager Network, du raccourci SMIT **ips4_basic** (pour IP Version 4) ou du raccourci SMIT **ips6_basic** (pour IP version 6). Vous pouvez également créer le tunnel manuellement en suivant la procédure qui suit.

L'exemple suivant illustre la commande **gentun** utilisée pour créer un tunnel manuel :

```
gentun –v 4 –t manual –s 5.5.5.19 –d 5.5.5.8 \
    –a HMAC_MD5 –e DES_CBC_8 –N 23567
```

Vous pouvez utiliser la commande **lstun –v 4** pour recenser les caractéristiques du tunnel manuel créé dans l'exemple ci–dessus. Le résultat de cette commande se présente comme suit :

```
Tunnel ID            : 1
IP Version           : IP Version 4
Source               : 5.5.5.19
Destination          : 5.5.5.8
Policy               : auth/encr
Tunnel Mode          : Tunnel
Send AH Algo         : HMAC_MD5
Send ESP Algo        : DES_CBC_8
Receive AH Algo      : HMAC_MD5
Receive ESP Algo     : DES_CBC_8
Source AH SPI        : 300
Source ESP SPI       : 300
Dest AH SPI          : 23576
Dest ESP SPI         : 23576
Tunnel Life Time     : 480
Status: Inactive
Target               : –
Target Mask          : –
Replay               : No
New Header           : Yes
Snd ENC–MAC Algo     : –
Rcv ENC–MAC Algo     : –
```

Pour activer le tunnel, entrez ce qui suit :

```
mktun –v 4 –t1
```

Les règles de filtrage associés au tunnel sont automatiquement générées.

Pour afficher les règles de filtrage, utilisez la commande **lsfilt –v 4**. Le résultat de cette commande se présente comme suit :

```
Rule 4:
Rule action           : permit
Source Address        : 5.5.5.19
Source Mask           : 255.255.255.255
Destination Address   : 5.5.5.8
Destination Mask      : 255.255.255.255
Source Routing        : yes
Protocol              : all
Source Port           : any 0
Destination Port      : any 0
Scope                 : both
Direction             : outbound
Logging control       : no
Fragment control      : all packets
Tunnel ID number      : 1
Interface             : all
Auto-Generated        : yes

Rule 5:
Rule action           : permit
Source Address        : 5.5.5.8
Source Mask           : 255.255.255.255
Destination Address   : 5.5.5.19
Destination Mask      : 255.255.255.255
Source Routing        : yes
Protocol              : all
Source Port           : any 0
Destination Port      : any 0
Scope                 : both
Direction             : inbound
Logging control       : no
Fragment control      : all packets
Tunnel ID number      : 1
Interface             : all
Auto-Generated        : yes
```

Pour activer les règles de filtrage, y compris les règles de filtrage par défaut, utilisez la commande **mktun –v 4 –t 1**.

Pour configurer l'autre extrémité, lorsqu'il s'agit d'un autre poste utilisant ce système d'exploitation, la définition du tunnel peut être exportée depuis l'hôte A, puis importée sur l'hôte B.

La commande suivante permet d'exporter la définition du tunnel dans un fichier nommé **ipsec_tun_manu.exp** et ses règles de filtrage associées dans le fichier **ipsec_fltr_rule.exp** du répertoire indiqué par l'indicateur **–f** :

```
exptun –v 4 –t 1 –f /   tmp
```

Création d’un tunnel manuel sur le second hôte

Pour créer l’extrémité du tunnel correspondante, les fichiers d’exportation sont copiés et importés sur le système distant à l’aide de la commande :

```
imptun –v 4 –t 1 –f /   tmp
```

où

`1`	Est le tunnel à importer
/ *tmp*	Est le répertoire contenant les fichiers importés

Le numéro du tunnel est généré par le système. Vous pouvez l’obtenir à partir de la sortie de la commande **gentun** ou via la commande **lstun**, qui répertorient les tunnels et indiquent le numéro du tunnel à importer. L’option **–t** n’est pas nécessaire si le fichier d’importation comporte un seul tunnel, ou si la totalité des tunnels doit être importée.

Si le système distant ne fonctionne pas sous ce système d’exploitation, le fichier d’exportation peut servir de référence pour configurer l’algorithme, les clés et les valeurs SPI de l’autre extrémité du tunnel.

Les fichiers exportés par un pare–feu peuvent être importés pour créer des tunnels. Pour ce faire, utilisez le paramètre **–n** lors de l’importation du fichier :

```
imptun –v 4 –f /   tmp   –n
```

Configuration des filtres

Le filtrage peut être simple, utilisant en grande partie les règles de filtrage générées automatiquement, ou élaboré en définissant des fonctions de filtre spécifiques à partir des propriétés des paquets IP. La mise en correspondance des paquets entrants s'effectue en comparant l'adresse source et de la valeur SPI avec les valeurs répertoriées dans la table de filtres. Cette paire doit donc être unique.

Chaque ligne de la table de filtres est une *règle*. Une série de règles définit les paquets qui seront acceptés au départ et à l'arrivée du système, et leur mode de routage. Les règles de filtrage peuvent contrôler différents aspects de la communication, y compris les adresses et les masques source et de destination, le protocole, le numéro de port, la direction, le contrôle des fragments, le routage source, le tunnel et l'interface.

Les types de règles de filtrage sont les suivants :

- Les Règles de filtrage statiques, page 12-44, sont créées dans la table de filtres et sont destinées au filtrage général du trafic ou à l'association avec des tunnels manuels. Elles peuvent être ajoutées, supprimées, modifiées et déplacées. Une zone de texte de description peut être ajoutée pour identifier une règle spécifique.
- Les Règles de filtrage générées automatiquement et définies par l'utilisateur, page 12-48, (également appelées règles de filtrage *générées automatiquement*), sont un ensemble spécifique de règles créées pour l'utilisation des tunnels IKE. Les règles de filtrage statiques et dynamiques reposent sur les informations et la négociation du tunnel de gestion des données.
- Les Règles de filtrage prédéfinies, page 12-48, sont des règles de filtrage génériques qui ne peuvent pas être modifiées, déplacées ou supprimées, telles que, par exemple, `all traffic` (tout le trafic), `ah` et `esp`. Elles s'appliquent à l'ensemble du trafic.

Les *Masques de sous–réseau*, page 12-49, dont les ID de groupe sont associés à une règle de filtrage, et l'option de configuration hôte–pare–feu–hôte, page 12-49, sont associés à ces règles de filtrage. Les sections suivantes décrivent les différents types de règles de filtrage et les caractéristiques qui leur sont associées.

Règles de filtrage statiques

Chaque règle de filtrage statique contient plusieurs zones séparées par des espaces. La liste qui suit fournit le nom de chaque zone (avec, entre parenthèses, un exemple de la règle 1) :

- Rule_number (`1`)
- Action (`permit`)
- Source_addr (`0.0.0.0`)
- Source_mask (`0.0.0.0`)
- Dest_addr (`0.0.0.0`)
- Dest_mask (`0.0.0.0`)
- Source_routing (`no`)
- Protocol (`udp`)
- Src_prt_operator (`eq`)
- Src_prt_value (`4001`)
- Dst_prt_operator (`eq`)
- Dst_prt_value (`4001`)
- Scope (`both`)

- Direction (`bot h`)
- Logging (`no`)
- Fragment (`all packets`)
- Tunnel (`0`)
- Interface (`all`).

Les règles de filtrage statiques sont expliquées plus en détail à la suite de cet exemple :

```
1 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 no udp eq 4001 eq 4001 both both no all
packets 0 all

2 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 no ah any 0 any 0 both both no all
packets 0 all

3 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 no esp any 0 any 0 both both no all
packets 0 all

4 permit 10.0.0.1 255.255.255.255 10.0.0.2 255.255.255.255 no all any 0 any 0
both outbound no all packets 1 all   outbound traffic

5 permit 10.0.0.2 255.255.255.255 10.0.0.1 255.255.255.255 no all any 0 any 0
both inbound no all packets 1 all

6 permit 10.0.0.1 255.255.255.255 10.0.0.3 255.255.255.255 no tcp lt 1024 eq 514
local outbound yes all packets 2 all

7 permit 10.0.0.3 255.255.255.255 10.0.0.1 255.255.255.255 no tcp/ack eq 514 lt
1024 local inbound yes all packets 2 all

8 permit 10.0.0.1 255.255.255.255 10.0.0.3 255.255.255.255 no tcp/ack lt 1024 lt
1024 local outbound yes all packets 2 all

9 permit 10.0.0.3 255.255.255.255 10.0.0.1 255.255.255.255 no tcp lt 1024 lt
1024 local inbound yes all packets 2 all

10 permit 10.0.0.1 255.255.255.255 10.0.0.4 255.255.255.255 no icmp any 0 any 0
local outbound yes all packets 3 all

11 permit 10.0.0.4 255.255.255.255 10.0.0.1 255.255.255.255 no icmp any 0 any 0
local inbound yes all packets 3 all

12 permit 10.0.0.1 255.255.255.255 10.0.0.5 255.255.255.255 no tcp gt 1023 eq 21
local outbound yes all packets 4 all

13 permit 10.0.0.5 255.255.255.255 10.0.0.1 255.255.255.255 no tcp/ack eq 21 gt
1023 local inbound yes all packets 4 all

14 permit 10.0.0.5 255.255.255.255 10.0.0.1 255.255.255.255 no tcp eq 20 gt 1023
local inbound yes all packets 4 all

15 permit 10.0.0.1 255.255.255.255 10.0.0.5 255.255.255.255 no tcp/ack gt 1023
eq 20 local outbound yes all packets 4 all

16 permit 10.0.0.1 255.255.255.255 10.0.0.5 255.255.255.255 no tcp gt 1023 gt
1023 local outbound yes all packets 4 all

17 permit 10.0.0.5 255.255.255.255 10.0.0.1 255.255.255.255 no tcp/ack gt 1023
gt 1023 local inbound yes all packets 4 all

18 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 no all any 0 any 0 both both yes all
packets
```

Chaque règle de l'exemple précédent est décrite de la manière suivante :

Règle 1 Concerne le démon de clé de session. Cette règle n'apparaît que dans les tables de filtres IPv4. Elle utilise le port 4001 pour contrôler les paquets pour actualiser la clé de session. La règle 1 illustre l'utilisation du numéro de port pour une tâche spécifique.

Remarque : Ne modifiez en aucun cas cette règle de filtrage, sauf dans un but de journalisation.

Règles 2 et 3 Autorisent le traitement des entêtes d'authentification AH et d'encapsulation ESP.

Remarque : Ne modifiez en aucun cas les règles 2 et 3, sauf dans un but de journalisation.

Règles 4 et 5 Des règles générées automatiquement pour filtrer les échanges entre les adresses 10.0.0.1 et 10.0.0.2 via le tunnel 1. La règle 4 concerne le trafic sortant, la règle 5 le trafic entrant.

Remarque : La règle 4 est définie par l'utilisateur en tant que *trafic sortant*.

Règles 6 à 9 Règles définies par l'utilisateur pour filtrer les services sortants **rsh**, **rcp**, **rdump**, **rrestore** et **rdist**, entre les adresses 10.0.0.1 et 10.0.0.3 via le tunnel 2. A noter que la journalisation est définie sur `yes` (oui) et permet à l'administrateur de gérer ce type de trafic.

Règles 10 et 11 Ensemble de règles définies par l'utilisateur qui filtrent à la fois les services entrant et sortant **icmp** de tous types échangés entre les adresses 10.0.0.1 et 10.0.0.4 via le tunnel 3.

Règles 12 à 17 Règles définies par l'utilisateur pour filtrer le service FTP sortant depuis les adresses 10.0.0.1 et 10.0.0.5 via le tunnel 4.

Règle 18 Règle générée automatiquement, toujours placée en fin de table. Dans cet exemple, elle accorde l'autorisation à tous les paquets qui ne correspondent pas aux règles précédentes. Vous pouvez cependant lui dire de refuser tous les paquets qui ne correspondent pas aux autres règles de filtrage.

Chaque règle peut être affichée séparément (avec **lsfilt**) afin d'énumérer chaque champ accompagné de sa valeur. Par exemple :

```
Rule 1:
Rule action          : permit
Source Address       : 0.0.0.0
Source Mask          : 0.0.0.0
Destination Address  : 0.0.0.0
Destination Mask     : 0.0.0.0
Source Routing       : yes
Protocol             : udp
Source Port          : eq  4001
Destination Port     : eq  4001
Scope                : both
Direction            : both
Logging control      : no
Fragment control     : all packets
Tunnel ID number     : 0
Interface            : all
Auto-Generated       : yes
```

Vous trouverez ci–dessous la liste de tous les paramètres pouvant être spécifiés dans une règle de filtrage :

–v Version IP : 4 ou 6.

–a Action :

d
Accès refusé

p
Accès accordé

–s Adresse source. Il peut s'agir d'une adresse IP ou du nom de l'hôte.

–m Masque de sous–réseau source.

–d Adresse de destination. Il peut s'agir d'une adresse IP ou du nom de l'hôte.

–M Masque de sous–réseau de destination.

–g Contrôle du routage source : `y` (oui) ou `n` (non).

–c Protocole. Les valeurs possibles sont `udp`, `icmp`, `tcp`, `tcp/ack`, `ospf`, `pip`, `esp`, `ah` et `all`.

–o Port source ou opération de type ICMP.

–p Port source ou valeur de type ICMP.

–O Port de destination ou opération de code ICMP.

–P Port de destination ou valeur de code ICMP.

–r Routage :

r
Paquets retransmis

l
Paquets d'origine ou de destination locales

b
Les deux

–l Gestion des journaux.

y
Inclure dans le journal

n
Ne pas inclure dans le journal.

–f Fragmentation.

y
S'applique aux en–têtes de fragment, aux fragments et aux non fragmentés

o
S'applique seulement aux fragments et aux en–têtes de fragment

n
Ne s'applique qu'aux non fragmentés

h
S'applique seulement aux non–fragmentés et aux en–têtes de fragment

–t ID du tunnel.

–i Interface, telle que `tr0` ou `en0`.

Pour plus d'informations, reportez–vous aux descriptions des commandes **genfilt** et **chfilt**.

Règles de filtrage générées automatiquement et définies par l'utilisateur

Certaines règles sont générées automatiquement pour l'utilisation du filtre de sécurité IP et du code tunnel. Les règles générées automatiquement comprennent :

- Les règles concernant le démon de clé de session destiné à actualiser les clés IP version 4 dans IKE (AIX 4.3.2 et versions ultérieures)
- Les règles chargées de traiter les paquets AH et ESP.

Les règles de filtrage sont également générées automatiquement lors de la définition des tunnels. Elles spécifient les valeurs de l'adresse source et de destination et du masque, ainsi que l'ID du tunnel. La totalité des données échangées entre ces adresses transite via le tunnel.

Pour les tunnels IKE, les règles de filtrage générées automatiquement déterminent le protocole et les numéros de port pendant la négociation IKE. Les règles de filtrage IKE sont conservées dans une table séparée, dans laquelle les recherches s'effectuent après les règles de filtrage statiques et avant les règles générées automatiquement. Les règles de filtrage IKE sont placées dans une position par défaut dans la table de filtres statiques, mais l'utilisateur peut les déplacer.

Les règles générées automatiquement autorisent tous les échanges via le tunnel. Les règles définies par l'utilisateur peuvent établir des restrictions sur certains types de trafic. Ces règles définies par l'utilisateur doivent être placées avant les règles générées automatiquement car la sécurité IP utilise la première règle qui s'applique au paquet. L'exemple ci–dessous illustre des règles de filtrage définies par l'utilisateur permettant de filtrer les échanges en fonction de l'opération ICMP.

```
1 permit 10.0.0.1 255.255.255.255 10.0.0.4 255.255.255.255 no icmp any 8
any 0 local outbound no all packets 3 all
2 permit 10.0.0.4 255.255.255.255 10.0.0.1 255.255.255.255 no icmp any 0
any 0 local inbound no all packets 3 all
3 permit 10.0.0.4 255.255.255.255 10.0.0.1 255.255.255.255 no icmp any 8
any 0 local inbound no all packets 3 all
4 permit 10.0.0.1 255.255.255.255 10.0.0.4 255.255.255.255 no icmp any 0
any 0 local outbound no all packets 3 all
```

Les règles de filtrage sont générées automatiquement lorsque les tunnels sont définis. Cela simplifie la configuration d'un seul tunnel. Cette fonction peut être supprimée en indiquant l'indicateur **–g** dans **gentun**. Vous pouvez trouver un exemple de filtre avec les commandes **genfilt** permettant de générer des règles de filtrage pour différents services TCP/IP dans **/usr/samples/ipsec/filter.sample**.

Règles de filtrage prédéfinies

Plusieurs règles de filtrage sont générées automatiquement avec certains événements. Lorsque l'unité `ipsec_v4` ou `ipsec_v6` est chargée, une règle prédéfinie est insérée dans la table de filtres puis activée. Par défaut, cette règle autorise tous les trafics, mais l'utilisateur peut la configurer, par exemple pour qu'elle refuse tous les paquets.

> **Remarque :** Lors d'une configuration à distance, prenez garde de ne pas activer la règle Accès refusé avant la fin de la configuration, pour éviter le verrouillage de votre session. Afin d'éviter ce cas de figure, attribuez par défaut la règle Accès autorisé ou configurez un tunnel sur la machine à distance avant d'activer IPsec.

Les tables de filtres IPv4 et IPv6 ont chacune une règle prédéfinie. Chacune peut être définie sur Accès refusé, indépendamment de l'autre. Cette opération empêchera le trafic de circuler, sauf s'il est autorisé par d'autres règles de filtrage. L'option **–l** de la commande **chfilt** est la seule autre option à modifier dans les règles prédéfinies, puisqu'elle permet la journalisation des paquets correspondants.

Pour les tunnels IKE, une règle de filtrage dynamique est placée dans la table de filtres IPv4. C'est l'emplacement réservé aux règles de filtrage dynamique dans la table de filtres. L'utilisateur peut contrôler cet emplacement en le déplaçant dans la table de filtres, vers le haut ou vers le bas. Dès que le démon de gestion des tunnels et le démon **isakmpd** sont initialisés pour permettre la négociation des tunnels IKE, les règles sont automatiquement créées dans la table de filtres dynamique pour traiter aussi bien les messages IKE que les paquets AH et ESP.

Masques de sous–réseau

Les masques de sous–réseau sont utilisés pour regrouper un ensemble d'ID associés à une règle de filtrage. Une opération AND est effectuée entre la valeur du masque et l'ID dans les règles de filtrage, et comparée à l'ID spécifié dans le paquet. Par exemple, une règle de filtrage comportant l'adresse IP source `10.10.10.4` et le masque de sous–réseau `255.255.255.255` impose une correspondance exacte avec l'adresse IP décimale, comme suit :

	Binaire	**Décimal**
Adresse IP source	`1010.1010.1010.0100`	`10.10.10.4`
Masque de sous–réseau	`1111.1111.1111.1111`	`255.255.255.255`

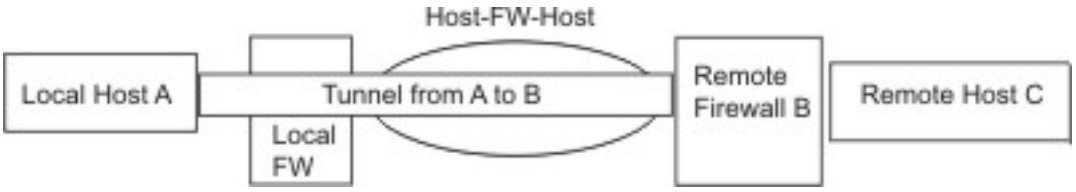
Un masque de sous–réseau de `10.10.10.x` correspond à `1111.1111.1111.0` ou à `255.255.255.0`. Le masque de sous–réseau est appliqué à l'adresse entrante, puis la combinaison se comparée avec l'ID présent dans la règle de filtrage. Par exemple, une adresse de `10.10.10.100` devient `10.10.10.0` après l'application du masque de sous–réseau correspondant à la règle de filtrage.

Un masque de sous–réseau de `255.255.255.240` autorise n'importe quelle valeur pour les 4 derniers bits de l'adresse.

Configuration Hôte–Pare–feu–Hôte

L'option de configuration hôte–pare–feu–hôte permet de créer un tunnel entre votre hôte et un pare–feu, puis de générer automatiquement les règles de filtrage nécessaires à une communication correcte avec un hôte situé derrière le pare–feu. Les règles de filtrage générées automatiquement autorisent toutes les règles entre les deux hôtes via le tunnel spécifié. Les règles par défaut (pour les en–têtes UDP, AH et ESP) devraient déjà gérer la communication hôte–pare–feu. Le pare–feu devra être paramétré de façon appropriée pour compléter la configuration. Nous vous recommandons d'utiliser le fichier d'exportation du tunnel que vous avez créé pour entrer les valeurs SPI et les clés nécessaires au pare–feu.

Figure 14. Hôte–Pare–feu–Hôte. Ce schéma présente la configuration Hôte–Pare–feu–Hôte. L'hôte A dispose d'un tunnel passant à travers un pare–feu local et sortant vers le réseau Internet. Il passe ensuite dans le pare–feu distant B, puis dans l'hôte distant C.



Fonctions de journalisation

Cette section décrit la configuration et le format des journaux système relatifs à la sécurité IP. Etant donné que les hôtes communiquent entre eux, les paquets transférés peuvent être journalisés dans le démon journal système, **syslogd**. D'autres messages importants concernant la sécurité IP s'affichent également. Un administrateur peut choisir de modifier cette information de journalisation pour effectuer des analyses de trafic et des opérations de débogage. Vous trouverez ci–après les étapes de configuration des fonctions de journalisation.

1. Modifiez le fichier **/etc/syslog.conf** pour ajouter l'entrée suivante :

```
local4.debug var/adm/ipsec.log
```

   Utilisez `local4` pour enregistrer les événements liés aux échanges et à la sécurité IP. Les niveaux de priorité standard du système d'exploitation s'appliquent. Vous devez régler le niveau de priorité de `debug` jusqu'à ce que le trafic via les filtres et tunnels de sécurité IP montrent une certaine stabilité et un mouvement correct.

   Remarque : La journalisation des événements de filtre peut entraîner une activité importante au niveau de l'hôte de sécurité IP et nécessiter une capacité de stockage importante.

2. Enregistrez **/etc/syslog.conf**.
3. Allez dans le répertoire indiqué pour le fichier journal, puis créez un fichier vide portant le même nom. Dans le cas cité ci–dessus, vous passeriez au répertoire **/var/adm** puis lanceriez la commande :

```
touch ipsec.log
```

4. Envoyez une commande **refresh** (actualiser) au sous–système **syslogd** :

```
refresh –s syslogd
```

5. Si vous utilisez des tunnels IKE, vérifiez que le fichier **/etc/isakmpd.conf** indique le niveau de journalisation **isakmpd** souhaité. (Pour plus d'informations sur la journalisation IKE, reportez–vous à la section Identification des incidents liés à la sécurité IP, page 12-54.)
6. Lorsque vous créez des règles de filtrage pour votre système hôte, si vous souhaitez que les paquets respectant une règle en particulier soient journalisés, attribuez la valeur **Y** (oui) au paramètre **–l** de la règle, à l'aide des commandes **genfilt** ou **chfilt**.
7. Enfin, activez la journalisation des paquets et lancez le démon **ipsec_logd** à l'aide de la commande suivante :

```
mkfilt –g start
```

   Vous pouvez arrêter la journalisation avec la commande suivante :

```
mkfilt –g stop
```

L'exemple de fichier journal suivant contient des entrées de trafic et de journal de sécurité IP :

L'exemple de fichier journal suivant contient des entrées de trafic et de journal de sécurité IP :

```
1. Aug 27 08:08:40 host1 : Filter logging daemon ipsec_logd (level 2.20)
    initialized at 08:08:40 on 08/27/97A
2. Aug 27 08:08:46 host1 : mkfilt: Status of packet logging set to Start
    at 08:08:46 on 08/27/97
3. Aug 27 08:08:47 host1 : mktun: Manual tunnel 2 for IPv4, 9.3.97.244,
    9.3.97.130
    activated.
4. Aug 27 08:08:47 host1 : mkfilt: #:1 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
    udp eq  4001 eq  4001  both both l=n f=y t=0 e= a=
5. Aug 27 08:08:47 host1 : mkfilt: #:2 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
    ah any 0 any 0  both both l=n f=y t=0 e= a=
6. Aug 27 08:08:47 host1 : mkfilt: #:3 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
    esp any 0 any 0  both both l=n f=y t=0 e= a=
7. Aug 27 08:08:47 host1 : mkfilt: #:4 permit 10.0.0.1 255.255.255.255 10.0.0.2
    255.255.255.255 icmp any 0 any 0  local outbound l=y f=y t=1 e= a=
8. Aug 27 08:08:47 host1 : mkfilt: #:4 permit 10.0.0.2 255.255.255.255 10.0.0.1
    255.255.255.255 icmp any 0 any 0  local inbound l=y f=y t=1 e= a=
9. Aug 27 08:08:47 host1 : mkfilt: #:6 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
     all any 0 any 0  both both l=y f=y t=0 e= a=
10. Aug 27 08:08:47 host1 : mkfilt: Filter support (level 1.00) initialized at
     08:08:47 on 08/27/97
11. Aug 27 08:08:48 host1 : #:6 R:p  o:10.0.0.1 s:10.0.0.1 d:10.0.0.20 p:udp
    sp:3327 dp:53 r:l a:n f:n T:0 e:n l:67
12. Aug 27 08:08:48 host1 : #:6 R:p  i:10.0.0.1 s:10.0.0.20 d:10.0.0.1 p:udp
     sp:53 dp:3327 r:l a:n f:n T:0 e:n l:133
13. Aug 27 08:08:48 host1 : #:6 R:p  i:10.0.0.1 s:10.0.0.15 d:10.0.0.1 p:tcp
    sp:4649 dp:23 r:l a:n f:n T:0 e:n l:43
14. Aug 27 08:08:48 host1 : #:6 R:p  o:10.0.0.1 s:10.0.0.1 d:10.0.0.15 p:tcp
    sp:23 dp:4649 r:l a:n f:n T:0 e:n l:41
15. Aug 27 08:08:48 host1 : #:6 R:p  i:10.0.0.1 s:10.0.0.15 d:10.0.0.1 p:tcp
    sp:4649 dp:23 r:l a:n f:n T:0 e:n l:40
16. Aug 27 08:08:51 host1 : #:4 R:p  o:10.0.0.1 s:10.0.0.1 d:10.0.0.2 p:icmp
     t:8 c:0 r:l a:n f:n T:1 e:n l:84
17. Aug 27 08:08:51 host1 : #:5 R:p  i:10.0.0.1 s:10.0.0.2 d:10.0.0.1 p:icmp
     t:0 c:0 r:l a:n f:n T:1 e:n l:84
18. Aug 27 08:08:52 host1 : #:4 R:p  o:10.0.0.1 s:10.0.0.1 d:10.0.0.2 p:icmp
     t:8 c:0 r:l a:n f:n T:1 e:n l:84
19. Aug 27 08:08:52 host1 : #:5 R:p  i:10.0.0.1 s:10.0.0.2 d:10.0.0.1 p:icmp
     t:0 c:0 r:l a:n f:n T:1 e:n l:84
20. Aug 27 08:32:27 host1 : Filter logging daemon terminating at 08:32:27 on
     08/27/971
```

Les paragraphes suivants expliquent les entrées de journal.

1	Démon de journalisation de filtre activé.
2	Journalisation de paquet de filtre activée avec la commande **mkfilt –g start**.
3	Activation du tunnel, affichage de l'ID du tunnel, adresse source, adresse de destination et horodate.
4–9	Les filtres ont été activés. La journalisation montre toutes les règles de filtrage chargées.
10	Message montrant l'activation des filtres.
11–12	Ces entrées montrent une recherche DNS d'un hôte.
13–15	Ces entrées correspondent partiellement à une connexion Telnet (les autres entrées ont été supprimées de cet exemple pour des raisons de place).
16–19	Ces entrées montrent deux Pings.
20	Démon de journalisation de filtre désactivé.

L'exemple ci–dessous illustre les phases 1 et 2 de négociation d'un tunnel entre deux hôtes, du point de vue de l'hôte initiateur. (Le niveau de journalisation **isakmpd** a été indiqué comme **isakmp_events**.)

```
1. Dec  6 14:34:42 host1 Tunnel Manager: 0: TM is processing a
     Connection_request_msg
2. Dec  6 14:34:42 host1 Tunnel Manager: 1: Creating new P1 tunnel object (tid)
3. Dec  6 14:34:42 host1 isakmpd: 192.168.100.103 >>> 192.168.100.104
    (SA PROPOSAL TRANSFORM)
4. Dec  6 14:34:42 host1 isakmpd: ::ffff:192.168.100.103 <<< 192.168.100.104
    (SA PROPOSAL TRANSFORM)
5. Dec  6 14:34:42 host1 isakmpd: Phase I SA Negotiated
6. Dec  6 14:34:42 host1 isakmpd: 192.168.100.103 >>> 192.168.100.104
    (KE NONCE)
7. Dec  6 14:34:42 host1 isakmpd: ::ffff:192.168.100.103 <<< 192.168.100.104
    (KE NONCE)
8. Dec  6 14:34:42 host1 isakmpd: Encrypting the following msg to send:
    (ID HASH)
9. Dec  6 14:34:42 host1 isakmpd: 192.168.100.103 >>> 192.168.100.104
    (Encrypted Payloads)
10. Dec  6 14:34:42 host1 isakmpd: ::ffff:192.168.100.103 <<< 192.168.100.104
    (Encrypted Payloads)
11. Dec  6 14:34:42 host1 Tunnel Manager: 1: TM is processing a
     P1_sa_created_msg (tid)
12. Dec  6 14:34:42 host1 Tunnel Manager: 1:   Received good P1 SA, updating P1
     tunnel (tid)
13. Dec  6 14:34:42 host1 Tunnel Manager: 0: Checking to see if any P2 tunnels
     need to start
14. Dec  6 14:34:42 host1 isakmpd: Decrypted the following received msg:
    (ID HASH)
15. Dec  6 14:34:42 host1 isakmpd:  Phase I Done !!!
16. Dec  6 14:34:42 host1 isakmpd: Phase I negotiation authenticated
17. Dec  6 14:34:44 host1 Tunnel Manager: 0: TM is processing a
     Connection_request_msg
18. Dec  6 14:34:44 host1 Tunnel Manager: 0: Received a connection object for an
     active P1 tunnel
19. Dec  6 14:34:44 host1 Tunnel Manager: 1: Created blank P2 tunnel (tid)
20. Dec  6 14:34:44 host1 Tunnel Manager: 0: Checking to see if any P2 tunnels
     need to start
21. Dec  6 14:34:44 host1 Tunnel Manager: 1: Starting negotiations for P2
     (P2 tid)
22. Dec  6 14:34:45 host1 isakmpd: Encrypting the following msg to send:
     (HASH SA PROPOSAL TRANSFORM NONCE ID ID)
23. Dec  6 14:34:45 host1 isakmpd: 192.168.100.103 >>> 192.168.100.104
    (Encrypted Payloads)
24. Dec  6 14:34:45 host1 isakmpd: ::ffff:192.168.100.103 <<< 192.168.100.104
    (Encrypted Payloads)
25. Dec  6 14:34:45 host1 isakmpd: Decrypted the following received msg:
     (HASH SA PROPOSAL TRANSFORM NONCE ID ID)
26. Dec  6 14:34:45 host1 isakmpd: Encrypting the following msg to send:
     (HASH)
27. Dec  6 14:34:45 host1 isakmpd: 192.168.100.103 >>> 192.168.100.104
    (Encrypted Payloads)
28. Dec  6 14:34:45 host1 isakmpd: Phase II SA Negotiated
29. Dec  6 14:34:45 host1 isakmpd: PhaseII negotiation complete.
30. Dec  6 14:34:45 host1 Tunnel Manager: 0: TM is processing a
      P2_sa_created_msg
31. Dec  6 14:34:45 host1 Tunnel Manager: 1: received p2_sa_created for an
      existing tunnel as initiator (tid)
32. Dec  6 14:34:45 host1 Tunnel Manager: 1: Filter::AddFilterRules: Created
      filter rules for tunnel
33. Dec  6 14:34:45 host1 Tunnel Manager: 0: TM is processing a
      List_tunnels_msg
```

Les paragraphes suivants expliquent les entrées de journal.

1–2 La commande **ike cmd=activate phase=1** active une connexion.

3–10 Le démon **isakmpd** négocie un tunnel de phase 1.

11–12 Le gestionnaire de tunnel reçoit de l'appelé un lien de sécurité valide de phase 1.

13	Ce gestionnaire vérifie si la commande **ike cmd=activate** dispose ou non d'une valeur de phase 2 pour poursuivre plus avant. Elle n'en a pas.
14–16	Le démon **isakmpd** termine la négociation de phase 1.
17–21	La commande **ike cmd=activate phase=2** active un tunnel de phase 2.
22–29	Le démon **isakmpd** négocie un tunnel de phase 2.
30–31	Le gestionnaire de tunnel reçoit de l'appelé un lien de sécurité valide de phase 2.
32	Le gestionnaire de tunnel écrit les règles de filtrage dynamiques.
33	La commande **ike cmd=list** affiche les tunnels IKE.

Libellés des entrées de zone

Les zones des entrées des journaux sont abrégées pour réduire l'espace DADS :

Numéro de la règle qui régit la journalisation de ce paquet.

R Type de règle

p
Accès accordé

d
Accès refusé

i / **o** Direction du paquet lors de son interception par le code de prise en charge du filtre. Identifie l'adresse IP de la carte associée au paquet :

- Pour les paquets entrants (i), c'est l'adresse de la carte sur laquelle est arrivé le paquet.
- Pour les paquets sortants (o), c'est l'adresse de la carte déterminée par la couche IP comme devant traiter la transmission du paquet.

s Indique l'adresse IP de l'expéditeur du paquet (extrait de l'en–tête IP).

d Indique l'adresse IP du destinataire prévu (extrait de l'en–tête IP).

p Indique le protocole de haut niveau utilisé pour créer le message dans la portion de données du paquet. Il peut s'agir d'un nombre ou d'un nom, par exemple : `udp`, `icmp`, `tcp`, `tcp/ack`, `ospf`, `pip`, `esp`, `ah`, or `all`.

sp / **t** Indique le numéro de port du protocole associé à l'expéditeur du paquet (extrait de l'en–tête TCP/UDP). Pour un protocole ICMP ou OSPF, cette zone est remplacée par un **t**, qui précise le type d'IP.

dp / **c** Indique le numéro de port du protocole associé au destinataire prévu (extrait de l'en–tête TCP/UDP). Pour un protocole ICMP, cette zone est remplacée par un **c**, qui indique le code IP.

– Indique qu'aucune information n'est disponible

r Indiquent une affiliation locale du paquet.

f
Paquets retransmis

l
Paquets locaux

o
Sortant

b
Les deux

l Indique la taille d'un paquet en octets.

f Indique si le paquet est un fragment.

T Indique l'ID du tunnel.

i Précise l'interface sur lequel est arrivé le paquet.

Identification des incidents liés à la sécurité IP

Vous trouverez dans la présente section des conseils et astuces pour vous aider à résoudre les incidents. Il est recommandé d'activer la journalisation lors de la configuration initiale de IPSec. Les journaux sont très utiles pour identifier les incidents liés aux filtres et aux tunnels. Reportez–vous à la section Fonctions de journalisation, page 12-50 pour plus d'informations en la matière.

Débogage des erreurs au niveau du tunnel manuel

Erreur : La commande **mktun** retourne le message d'erreur suivant :

```
insert_tun_man4(): write failed : The requested resource is busy.
```

Problème : Le tunnel que vous souhaitez activer est déjà actif ou une collision des valeurs SPI s'est produite.

Solution : Lancez la commande **rmtun** pour désactiver le tunnel, puis utilisez la commande mktun pour le réactiver. Vérifiez si les valeurs SPI du tunnel défaillant correspondent à un autre tunnel actif. Chaque tunnel doit posséder ses propres valeurs SPI, uniques.

Erreur : La commande **mktun** retourne le message d'erreur suivant :

```
Device ipsec_v4 is in Defined status.


Tunnel activation for IP Version 4 not performed.
```

Problème : Vous n'avez pas rendu disponible l'unité de sécurité IP.

Solution : Lancez la commande suivante :

```
mkdev -l ipsec -t 4
```

Vous devez remplacer l'option **–t** par 6 si la même erreur se produit lors de l'activation d'un tunnel IP Version 6. Les unités doivent être dans l'état disponible. Pour vérifiez l'état de l'unité de sécurité IP, lancez la commande suivante :

```
lsdev -Cc ipsec
```

Erreur : La commande **gentun** retourne le message d'erreur suivant

```
Invalid Source IP address
```

Problème : Vous avez saisi une adresse IP incorrecte comme adresse source.

Solution : Pour les tunnels IP version 4, vérifiez si vous avez indiqué une adresse IP version 4 disponible pour la machine locale. Lorsque vous générez des tunnels, vous ne pouvez pas utiliser de nom d'hôte comme adresse source. Ce n'est possible que pour l'adresse de destination.

Pour les tunnels IP version 6, vérifiez si vous avez indiqué une adresse IP version 6 disponible. Si vous entrez `netstat -in` et qu'aucune adresse IP version 6 n'existe, exécutez **/usr/sbin/autoconf6** (interface) pour générer automatiquement une adresse locale (avec l'adresse MAC) ou utilisez **ifconfig** pour attribuer manuellement une adresse.

Erreur : La commande **gentun** retourne le message d'erreur suivant :

```
Invalid Source IP address
```

Problème : Vous avez saisi une adresse IP incorrecte comme adresse source.

Solution : Pour les tunnels IP version 4, vérifiez si vous avez indiqué une adresse IP version 4 disponible pour la machine locale. Lorsque vous générez des tunnels, vous ne pouvez pas utiliser de nom d'hôte comme adresse source. Ce n'est possible que pour l'adresse de destination

Pour les tunnels IP version 6, vérifiez si vous avez indiqué une adresse IP version 6 disponible. Si vous entrez `netstat -in` et qu'aucune adresse IP version 6 n'existe, exécutez **/usr/sbin/autoconf6** (interface) pour générer automatiquement une adresse locale (avec l'adresse MAC) ou utilisez **ifconfig** pour attribuer manuellement une adresse.

Erreur : La commande **mktun** retourne le message d'erreur suivant :

```
insert_tun_man4(): write failed: A system call received a
parameter that is not valid.
```

Problème : La génération du tunnel s'est produite avec une combinaison ESP et AH incorrecte, ou sans l'utilisation du nouveau format d'en–tête lorsque nécessaire.

Solution : Vérifiez la nature des algorithmes d'authentification en cours d'utilisation par le tunnel en question. N'oubliez pas que les algorithmes HMAC_MD5 et HMAC_SHA requièrent le nouveau format d'en–tête. Le nouveau format d'en–tête peut être modifié à l'aide du raccourci SMIT **ips4_basic** ou de l'indicateur **–z** associé à la commande **chtun**. Rappelez–vous également que DES_CBC_4 ne peut pas être utilisé avec le nouveau format d'en–tête.

Erreur : Le lancement d'IPSec à partir de Web-based System Manager génère un message d'erreur `Failure`.

Problème : Les démons IPSec ne sont pas lancés.

Solution : Vérifier quels sont les démons en cours d'exécution avec la commande `ps-ef`. Les démons associés à IPSec sont les suivants :

- **tmd**
- **isakmpd**
- **cpsd**

Le démon **cpsd** n'est actif que lorsque le code du certificat numérique est installé (fichiers **gskit.rte** ou **gskkm.rte**) et lorsque vous avez configuré l'utilitaire Key Manager pour la prise en charge des certificats numériques.

Si les démons ne sont pas actifs, arrêtez IPSec avec Web-based System Manager, puis relancez–le pour lancer automatiquement les démons appropriés.

Erreur : L'utilisation d'IPSec génère le message d'erreur suivant :

```
The installed bos.crypto is back level and must be updated.
```

Problème : Les fichiers **bos.net.ipsec.*** ont été mis à jour avec une version plus récente, mais les fichiers **bos.crypto.*** correspondants ne l'ont pas été.

Solution : Mettez les fichiers **bos.crypto.*** à jour avec la version correspondante aux fichiers **bos.net.ipsec.*** mis à jour.

Débogage des erreurs au niveau des tunnels IKE

Les sections suivantes décrivent les erreurs générées lors de l'utilisation de tunnels IKE.

Organigramme des tunnels IKE

Les tunnels IKE sont configurés via la commande **ike** ou les écrans VPN du Web-based System Manager, avec les démons suivants :

Tableau 8. Démons utilisés par les tunnels IKE.

tmd	Démon de gestion des tunnels
isakmpd	Démon IKE
cpsd	Démon de proxy de certificat

Pour que les tunnels IKE soient configurés correctement, les démons **tmd** et **isakmpd** doivent être actifs. Si la fonction de sécurité IP est lancée lors du redémarrage, l'exécution de ces démons se fait automatiquement. Dans le cas contraire, ils doivent être lancés avec Web–based System Manager.

Le gestionnaire de tunnels demande à **isakmpd** de lancer un tunnel. Si le tunnel existe déjà ou n'est pas valide (en cas d'adresse distante erronée, par exemple), un message d'erreur apparaît. Une fois la négociation lancée, son exécution peut prendre un certain temps, en fonction de la latence du réseau. La commande **ike cmd=list** indiquera l'état du tunnel pour savoir si la négociation s'est bien déroulée. Le gestionnaire de tunnel consigne également des événements dans le journal système **syslog**, au niveau des sections **debug**, **event** et **information**, utilisées pour surveiller l'état d'avancement de la négociation.

La séquence est la suivante :

1. Utilisez Web-based System Manager ou la commande **ike** pour lancer un tunnel.
2. Le démon **tmd** envoie au démon **isakmpd** une demande de connexion pour la gestion de clés (phase 1).
3. Le démon **isakmpd** répond par le message `SA created` (CA créée) ou par l'affichage d'un message d'erreur.
4. Le démon **tmd** envoie au démon **isakmpd** une demande de connexion pour un tunnel de gestion des données (phase 2).
5. Le démon **isakmpd** répond par le message `SA created` ou par l'affichage d'un message d'erreur.
6. Les paramètres de tunnel sont insérés dans le cache de tunnel du noyau.
7. Les règles de filtres sont ajoutées à la table de filtres dynamique du noyau.

Lorsque la machine agit comme répondeur, le démon **isakmpd** informe le démon **tmd** du gestionnaire de tunnels du bon déroulement de la négociation. Un nouveau tunnel est inséré dans le noyau. Le processus commence alors à l'étape 3 et se poursuit jusqu'à l'étape 7, sans que le démon **tmd** ne demande de connexion.

Journalisation IKE

Les démons **isakmpd**, **tmd** et **cpsd** consignent des événements dans **syslog**. Pour le démon **isakmpd**, la journalisation est activée à l'aide de la commande **ike cmd=log**. Le fichier de configuration **/etc/isakmpd.conf** peut être défini pour spécifier le niveau de journalisation. Ce niveau peut prendre les valeurs **none**, **errors**, **isakmp_events** ou **information**.

Remarque : Dans les versions antérieures à AIX 5.1, le démon **isakmpd** consignait les éléments dans un fichier séparé, spécifié dans **/etc/isakmpd.conf**.

Le paramètre du fichier de configuration qui peut être défini pour la journalisation est log_level. Les démons IKE utilisent les niveaux de journalisation suivants :

none	Aucune journalisation (valeur par défaut)
error	Consignation uniquement des erreurs de protocole et d'API
isakmp_events	Consignation uniquement des événements et des erreurs de protocole IKE
Information	Consignation des informations de mise en œuvre et de protocole (conseillé lors d'un débogage)

La syntaxe de cette option est :

```
log_level
```

Le démon **isakmpd** démarre par l'envoi d'une proposition ou répond en évaluant une proposition. Si cette proposition est acceptée, un lien de sécurité est généré et le tunnel est configuré. Si cette proposition est refusée ou si le délai de connexion expire avant la fin de la négociation, **isakmpd** renvoie un message d'erreur. Les entrées du journal système **syslog** dans **tmd** indiquent la réussite ou non de la négociation. Un échec pour cause de certificat non valide est consigné dans **syslog**. Pour déterminer la cause exacte de l'échec d'une négociation, contrôlez le fichier journal spécifié dans **/etc/syslog.conf**.

Syslog ajoute à chaque ligne du journal un préfixe comprenant la date et l'heure, la machine et le programme. Dans l'exemple suivant, le nom de machine est `googly` et le nom de programme est `isakmpd` :

```
Nov 20 09:53:50 googly isakmpd: ISAKMP_MSG_HEADER
Nov 20 09:53:50 googly isakmpd:  Icookie : 0xef06a77488f25315, Rcookie
:0x0000000000000000
Nov 20 09:53:51 googly isakmpd:  Next Payload : 1(SA), Maj Ver : 1, Min Ver
: 0
Nov 20 09:53:51 googly isakmpd:  Xchg Type : 2 (ID protected), Flag= 0,
Encr : No,COMMIT : non
Nov 20 09:53:51 googly isakmpd:  Msg ID  : 0x00000000
```

Pour plus de clarté, vous pouvez utiliser la commande **grep** pour extraire des lignes intéressantes du journal (toute la journalisation **isakmpd** par exemple) et la commande **cut** pour retirer le préfixe de chaque ligne. Les exemples de journalisation **isakmpd** dans le reste de cette section ont été transformé d'une manière similaire.

Fonction de journalisation Parse Payload (analyse de blocs)

Le lien de sécurité (SA) entre deux points terminaux est établi grâce à l'échange de messages IKE. La fonction d'analyse de blocs interprète les messages dans un format lisible par l'homme. Vous pouvez activer la journalisation en modifiant le fichier **/etc/isakmpd.conf**. Dans le fichier **/etc/isakmpd.conf**, l'entrée relative à la journalisation est semblable à la ligne suivante :

```
information
```

Le type de blocs IKE consignés par l'analyse de blocs varie selon le contenu des messages IKE. Les exemples incluent les blocs SA, Key Exchange, Certificate Request, Certificate et Signature. Le journal de l'analyse de blocs de l'exemple suivant possède un en–tête ISAKMP_MSG_HEADER suivi par cinq blocs :

```
ISAKMP_MSG_HEADER
         Icookie : 0x9e539a6fd4540990, Rcookie : 0x0000000000000000
         Next Payload : 1(SA), Maj Ver : 1, Min Ver : 0
         Xchg Type : 4 (Aggressive), Flag= 0, Encr : No,COMMIT : No
         Msg ID  : 0x00000000
         len     : 0x10e(270)
 SA Payload:
         Next Payload : 4(Key Exchange), Payload len : 0x34(52)
         DOI          : 0x1(INTERNET)
         bitmask      : 1(SIT_IDENTITY_ONLY
    Proposal Payload:
         Next Payload : 0(NONE), Payload len : 0x28(40)
         Proposal # : 0x1(1), Protocol-ID : 1(ISAKMP)
         SPI size : 0x0(0), # of Trans : 0x1(1)
    Transform Payload:
         Next Payload : 0(NONE), Payload len : 0x20(32)
         Trans # : 0x1(1), Trans.ID : 1(KEY_IKE)
         Attr : 1(Encr.Alg     ), len=0x2(2)
         Value=0x1(1),(DES-cbc)
         Attr : 2(Hash Alg     ), len=0x2(2)
         Value=0x1(1),(MD5)
         Attr : 3(Auth Method  ), len=0x2(2)
         Value=0x3(3),(RSA Signature)
         Attr : 4(Group Desc   ), len=0x2(2)
         Value=0x1(1),(default 768-bit MODP group)
         Attr : 11(Life Type    ), len=0x2(2)
         Value=0x1(1),(seconds)
         Attr : 12(Life Duration), len=0x2(2)
         Value=0x7080(28800)
 Key Payload:
         Next Payload : 10(Nonce), Payload len : 0x64(100)

         Key Data :
         33 17 68 10 91 1f ea da 38 a0 22 2d 84 a3 5d 5d
         a0 e1 1f 42 c2 10 aa 8d 9d 14 0f 58 3e c4 ec a3
         9f 13 62 aa 27 d8 e5 52 8d 5c c3 cf d5 45 1a 79
         8a 59 97 1f 3b 1c 08 3e 2a 55 9b 3c 50 cc 82 2c
         d9 8b 39 d1 cb 39 c2 a4 05 8d 2d a1 98 74 7d 95
         ab d3 5a 39 7d 67 5b a6 2e 37 d3 07 e6 98 1a 6b

 Nonce Payload:
         Next Payload : 5(ID), Payload len : 0xc(12)

         Nonce Data:
         6d 21 73 1d dc 60 49 93
 ID Payload:
         Next Payload : 7(Cert.Req), Payload len : 0x49(73)
         ID type      : 9(DER_DN), Protocol : 0, Port = 0x0(0)
 Certificate Request Payload:
         Next Payload : 0(NONE), Payload len : 0x5(5)
         Certificate Encoding Type: 4(X.509 Certificate - Signature)
```

Dans chaque bloc se trouve le champ Next Payload qui renvoie au bloc suivant. Si le bloc actif est le dernier du message IKE, le champ `Next Payload` possède la valeur zéro (None).

Chaque bloc de cet exemple contient des informations concernant les négociations en cours. A titre d'exemple, le bloc SA comporte les blocs Proposal (proposition) et Transform (conversion), lesquels à leur tour contiennent l'algorithme de chiffrement, le mode d'authentification, l'algorithme de hachage, le type de cycle SA et la durée SA que l'initiateur propose au répondant.

De plus, le Bloc SA est composé d'un ou de plusieurs bloc Proposal, et d'un ou de plusieurs blocs Transform. La valeur du champ `Next Payload` du bloc Proposal est 0 si ce bloc est unique, ou 2 s'il est suivi par plusieurs blocs Proposal. De même, la valeur du champ `Next Payload` d'un bloc Transform est 0 s'il est unique, ou 3 s'il est suivi par plusieurs blocs Transform, comme dans l'exemple suivant :

```
ISAKMP_MSG_HEADER
         Icookie : 0xa764fab442b463c6, Rcookie : 0x0000000000000000
         Next Payload : 1(SA), Maj Ver : 1, Min Ver : 0
         Xchg Type : 2 (ID protected), Flag= 0, Encr : No,COMMIT : No
         Msg ID  : 0x00000000
         len     : 0x70(112)
 SA Payload:
         Next Payload : 0(NONE), Payload len : 0x54(84)
         DOI          : 0x1(INTERNET)
         bitmask      : 1(SIT_IDENTITY_ONLY
    Proposal Payload:
         Next Payload : 0(NONE), Payload len : 0x48(72)
         Proposal # : 0x1(1), Protocol-ID : 1(ISAKMP)
         SPI size : 0x0(0), # of Trans : 0x2(2)
    Transform Payload:
         Next Payload : 3(Transform), Payload len : 0x20(32)
         Trans # : 0x1(1), Trans.ID : 1(KEY_IKE)
         Attr : 1(Encr.Alg     ), len=0x2(2)
         Value=0x5(5),(3DES-cbc)
         Attr : 2(Hash Alg     ), len=0x2(2)
         Value=0x1(1),(MD5)
         Attr : 3(Auth Method  ), len=0x2(2)
         Value=0x1(1),(Pre-shared Key)
         Attr : 4(Group Desc   ), len=0x2(2)
         Value=0x1(1),(default 768-bit MODP group)
         Attr : 11(Life Type    ), len=0x2(2)
         Value=0x1(1),(seconds)
         Attr : 12(Life Duration), len=0x2(2)
         Value=0x7080(28800)
    Transform Payload:
         Next Payload : 0(NONE), Payload len : 0x20(32)
         Trans # : 0x2(2), Trans.ID : 1(KEY_IKE)
         Attr : 1(Encr.Alg     ), len=0x2(2)
         Value=0x1(1),(DES-cbc)
         Attr : 2(Hash Alg     ), len=0x2(2)
         Value=0x1(1),(MD5)
         Attr : 3(Auth Method  ), len=0x2(2)
         Value=0x1(1),(Pre-shared Key)
         Attr : 4(Group Desc   ), len=0x2(2)
         Value=0x1(1),(default 768-bit MODP group)
         Attr : 11(Life Type    ), len=0x2(2)
         Value=0x1(1),(seconds)
         Attr : 12(Life Duration), len=0x2(2)
         Value=0x7080(28800)
```

L'en–tête de message IKE d'un journal d'analyse de blocs montre le type d'échange – Main (principal) ou Aggressive (agressif) –, la longueur de l'ensemble du message, l'ID du message, etc.

Le Bloc Certificate Request demande un certificat au répondant. Le répondant envoie le certificat dans un message séparé. L'exemple suivant montre les Blocs Certificate et Signature envoyés à un homologue lors d'une négociation SA. Les données relatives au certificat et à la signature sont au format hexadécimal.

```
ISAKMP_MSG_HEADER
        Icookie : 0x9e539a6fd4540990, Rcookie : 0xc7e0a8d937a8f13e
        Next Payload : 6(Certificate), Maj Ver : 1, Min Ver : 0
        Xchg Type : 4 (Aggressive), Flag= 0, Encr : No,COMMIT : No
        Msg ID  : 0x00000000
        len     : 0x2cd(717)
 Certificate Payload:

        Next Payload : 9(Signature), Payload len : 0x22d(557)
        Certificate Encoding Type: 4(X.509 Certificate - Signature)
        Certificate: (len 0x227(551) in bytes
        82 02 24 30 82 01 8d a0 03 02 01 02 02 05 05 8e
        fb 3e ce 30 0d 06 09 2a 86 48 86 f7 0d 01 01 04
        05 00 30 5c 31 0b 30 09 06 03 55 04 06 13 02 46
        49 31 24 30 22 06 03 55 04 0a 13 1b 53 53 48 20
        43 6f 6d 6d 75 6e 69 63 61 74 69 6f 6e 73 20 53
        65 63 75 72 69 74 79 31 11 30 0f 06 03 55 04 0b
        13 08 57 65 62 20 74 65 73 74 31 14 30 12 06 03
        55 04 03 13 0b 54 65 73 74 20 52 53 41 20 43 41
        30 1e 17 0d 39 39 30 39 32 31 30 30 30 30 30 30
        5a 17 0d 39 39 31 30 32 31 32 33 35 39 35 39 5a
        30 3f 31 0b 30 09 06 03 55 04 06 13 02 55 53 31
        10 30 0e 06 03 55 04 0a 13 07 49 42 4d 2f 41 49
        58 31 1e 30 1c 06 03 55 04 03 13 15 62 61 72 6e
        65 79 2e 61 75 73 74 69 6e 2e 69 62 6d 2e 63 6f
        6d 30 81 9f 30 0d 06 09 2a 86 48 86 f7 0d 01 01
        01 05 00 03 81 8d 00 30 81 89 02 81 81 00 b2 ef
        48 16 86 04 7e ed ba 4c 14 d7 83 cb 18 40 0a 3f
        55 e9 ad 8f 0f be c5 b6 6d 19 ec de 9b f5 01 a6
        b9 dd 64 52 34 ad 3d cd 0d 8e 82 6a 85 a3 a8 1c
        37 e4 00 59 ce aa 62 24 b5 a2 ea 8d 82 a3 0c 6f
        b4 07 ad 8a 02 3b 19 92 51 88 fb 2c 44 29 da 72
        41 ef 35 72 79 d3 e9 67 02 b2 71 fa 1b 78 13 be
        f3 05 6d 10 4a c7 d5 fc fe f4 c0 b8 b8 fb 23 70
        a6 4e 16 5f d4 b1 9e 21 18 82 64 6d 17 3b 02 03
        01 00 01 a3 0f 30 0d 30 0b 06 03 55 1d 0f 04 04
        03 02 07 80 30 0d 06 09 2a 86 48 86 f7 0d 01 01
        04 05 00 03 81 81 00 75 a4 ee 9c 3a 18 f2 de 5d
        67 d4 1c e4 04 b4 e5 b8 5e 9f 56 e4 ea f0 76 4a
        d0 e4 ee 20 42 3f 20 19 d4 25 57 25 70 0a ea 41
        81 3b 0b 50 79 b5 fd 1e b6 0f bc 2f 3f 73 7d dd
        90 d4 08 17 85 d6 da e7 c5 a4 d6 9a 2e 8a e8 51
        7e 59 68 21 55 4c 96 4d 5a 70 7a 50 c1 68 b0 cf
        5f 1f 85 d0 12 a4 c2 d3 97 bf a5 42 59 37 be fe
        9e 75 23 84 19 14 28 ae c4 c0 63 22 89 47 b1 b6
        f4 c7 5d 79 9d ca d0
 Signature Payload:
        Next Payload : 0(NONE), Payload len : 0x84(132)

        Signature: len 0x80(128) in bytes
        9d 1b 0d 90 be aa dc 43 95 ba 65 09 b9 00 6d 67
        b4 ca a2 85 0f 15 9e 3e 8d 5f e1 f0 43 98 69 d8
        5c b6 9c e2 a5 64 f4 ef 0b 31 c3 cb 48 7c d8 30
        e3 a2 87 f4 7c 9d 20 49 b2 39 00 fa 8e bf d9 b0
        7d b4 8c 4e 19 3a b8 70 90 88 2c cf 89 69 5d 07
        f0 5a 81 58 2e 15 40 37 b7 c8 d6 8c 5c e2 50 c3
        4d 19 7e e0 e7 c7 c2 93 42 89 46 6b 5f f8 8b 7d
        5b cb 07 ea 36 e5 82 9d 70 79 9a fe bd 6c 86 36
```

Incidents liés au certificat numérique et au mode de signature

Erreur : Le démon **cpsd** (Certificate Proxy Server daemon) ne démarre pas. Une entrée similaire à la suivante apparaît dans le fichier journal :

```
Sep 21 16:02:00 ripple CPS[19950]: Init():LoadCaCerts() failed,
rc=-12
```

Problème : Le système n'a pas pu ouvrir ou trouver la base de données des certificats.

Solution : Assurez–vous que les bases de données de certificats de Key Manager se trouvent dans **/etc/security**. La base de données est constituée des fichiers suivants : **ikekey.crl**, **ikekey.kdb**, **ikekey.rdb** et **ikekey.sth**.

Si seul **ikekey.sth** est manquant, c'est que l'option `stash password` n'a pas été sélectionnée lors de la création de la base de données d'e Key Manager. Le mot de passe doit être sécurisé dans le fichier stash pour activer l'utilisation des certificats numériques avec IPSec. Pour en savoir plus, reportez–vous à Création d'une base de données de clés, page 12-30.)

Erreur : Key Manager génère le message d'erreur suivant lors de la réception d'un certificat :

```
Invalid Base64-encoded data was found
```

Problème : Des données surnuméraires ont été trouvées dans le fichier certificat ou bien des données ont été perdues ou endommagées.

Solution : Le certificat chiffré 'DER' doit se trouver entre les chaînes de l'exemple ci–dessous. Aucun caractère ne doit figurer avant et après les chaînes BEGIN CERTIFICATE et END CERTIFICATE.

```
-----BEGIN CERTIFICATE-----
 MIICMTCCAZqgAwIBAgIFFKZtANowDQYJKoZIhvcNAQEFBQAwXDELMAkGA1UEBhMC
 RkkxJDAiBgNVBAoTG1NTSCBDb21tdW5pY2F0aW9ucyBTZWN1cml0eTERMA8GA1UE
 CxMIV2ViIHRlc3QxFDASBgNVBAMTC1Rlc3QgUlNBIENBMB4XDTk5MDkyMTAwMDAw
 MFoXDTk5MTAyMTIzNTk1OVowOzELMAkGA1UEBhMCVVMxDDAKBgNVBAoTA01CTTEe
 MBwGA1UEAxMVcmlwcGxlLmF1c3Rpbi5pYm0uY29tMIGfMA0GCSqGSIb3DQEBAQUA
 A4GNADCBiQKBgQC5EZqo6n7tZrpAL6X4L7mf4yXQSm+m/NsJLhp6afbFpPvXgYWC
 wq4pvOtvxgum+FHrE0gysNjbKkE4Y6ixC9PGGAKHnhM3vrmvFjnl1G6KtyEz58Lz
 BWW39QS6NJ1LqqP1nT+y3+Xzvfv8Eonqzno8mglCWMX09SguLmWoU1PcZQIDAQAB
 oyAwHjALBgNVHQ8EBAMCBaAwDwYDVR0RBAgwBocECQNhhzANBgkqhkiG9w0BAQUF
 AOBgQA6bgp4Zay34/fyAlyCkNNAYJRrN3Vc4NHN7IGjUziN6jK5UyB5zL37FERW
 hT9ArPLzK7yEZs+MDNvB0bosyGWEDYPZr7EZHhYcoBP4/cd0V5rBFmA8Y2gUthPi
 Ioxpi4+KZGHYyLqTrm+8Is/DVJaQmCGRPynHK35xjT6WuQtiYg==
 -----END CERTIFICATE-----
```

Les options suivantes peuvent aider au diagnostic et à la résolution de cet incident.

- Si les données ont été perdues ou endommagées, recréez le certificat.
- Analysez le certificat pour en vérifier la validité à l'aide d'un analyseur de type ASN.1 (disponible sur le Web).

Erreur : Key Manager génère le message d'erreur suivant lors de la réception d'un certificat personnel :

```
No request key was found for the certificate
```

Problème : La demande de certificat personnel du certificat en cours de réception n'existe pas.

Solution : Créez à nouveau la demande de certificat personnel et demandez un nouveau certificat.

Erreur : Lors de la configuration d'un tunnel IKE, Web-based System Manager génère le message d'erreur suivant :

```
Error 171 in the Key Management (Phase 1) Tunnel operation:
 PUT_IRL_FAILED
```

Problème : Il est possible que le type d'identité de l'hôte ne soit pas valide ; ce type est configuré par la boîte de dialogue IKE (onglet Identification). Ceci a lieu lorsque le type d'identité de l'hôte sélectionné dans la liste déroulante ne correspond pas au type attendu de la zone `Host Identity`. A titre d'exemple, si le type d'identité de l'hôte sélectionné est **X500 Distinguished Name**, vous devez entrer un nom spécifique correctement formaté dans la zone `Host Identity`.

Solution : Assurez–vous que le nom spécifique entré est correct pour le type d'identité de l'hôte sélectionné dans la liste déroulante.

Erreur : Une négociation IKE échoue et une entrée similaire au message suivant apparaît dans le fichier journal :

```
inet_cert_service::channelOpen():clientInitIPC():error,rc =2
 (No such file or directory)
```

Problème : Le démon **cpsd** n'est pas actif ou s'est arrêté.

Solution : Lancez IPSec avec Web–based System Manager. Cette action lance également les démons appropriés.

Erreur : Une négociation IKE échoue et une entrée similaire au message suivant apparaît dans le fichier journal :

```
CertRepo::GetCertObj:  DN Does Not Match:
("/C=US/O=IBM/CN=ripple.austin.ibm.com")
```

Problème : Le type X500 Distinguished Name (DN, nom spécifique) sélectionné lors de la définition du tunnel IKE ne correspond pas au type X500 DN du certificat personnel.

Solution : Modifiez la définition du tunnel IKE dans Web-based System Manager pour la faire correspondre au nom spécifique du certificat.

Erreur : Lors de la définition des tunnels IKE dans Web–based System Manager, la case Certificat numérique est désactivée dans l'onglet Méthode d'authentification.

Problème : La politique associée à ce tunnel n'utilise pas l'authentification en mode signature RSA.

Solution : Pour utiliser la méthode d'authentification des signatures RSA, modifiez la conversion de la politique associée. A titre d'exemple, lors de la définition d'un tunnel IKE, choisissez la politique de gestion des clés *IBM_low_CertSig*.

Fonctions de suivi

Il s'agit d'outils de débogage utilisés pour le suivi des événements du noyau. La fonction de suivi peut être utilisée pour obtenir de plus amples informations sur les erreurs ou les événements qui se sont produits dans le filtre du noyau et le code du tunnel.

SMIT possède une fonction de suivi pour la sécurité IP, disponible via le menu Configuration avancée de la sécurité IP. Parmi les informations qui entrent dans le champ d'application de cette fonction de suivi figurent les informations sur les erreurs, les filtres, les tunnels, l'encapsulation/décapsulation et le chiffrement. Par conception, le suivi d'erreurs fournit les informations les plus importantes. L'utilitaire de suivi d'informations peut générer un volume d'informations important et nuire aux performances du système. Cette opération de suivi vous fournit des indices permettant d'identifier l'incident. Les informations de suivi seront nécessaires lorsque vous serez en contact avec un mainteneur. Pour accéder à la fonction de suivi, utilisez le raccourci SMIT **smit ips4_tracing** (pour IPv4) ou **smit ips6_tracing** (pour IPv6).

ipsecstat

Vous pouvez lancer la commande **ipsecstat** pour générer l'exemple de rapport suivant. Ce rapport indique que les unités de sécurité IP sont disponibles, que trois algorithmes d'authentification et trois algorithmes de chiffrement sont installés, et qu'il existe un rapport sur l'activité des paquets. Ces informations peuvent servir à identifier l'origine d'un incident si vous cherchez à résoudre les incidents liés au trafic de sécurité IP.

```
IP Security Devices:
 ipsec_v4 Available
 ipsec_v6 Available

 Authentication Algorithm:
 HMAC_MD5 -- Hashed MAC MD5 Authentication Module
 HMAC_SHA -- Hashed MAC SHA Hash Authentication Module
 KEYED_MD5 -- Keyed MD5 Hash Authentication Module

 Encryption Algorithm:
 CDMF -- CDMF Encryption Module
 DES_CBC_4 -- DES CBC 4 Encryption Module
 DES_CBC_8 -- DES CBC 8 Encryption Module
 3DES_CBC -- Triple DES CBC Encryption Module

 IP Security Statistics -
 Total incoming packets:  1106
 Incoming AH packets:326
 Incoming ESP packets:  326
 Srcrte packets allowed:  0
 Total outgoing packets:844
 Outgoing AH packets:527
 Outgoing ESP packets:  527
 Total incoming packets dropped:  12
   Filter denies on input:  12
   AH did not compute: 0
   ESP did not compute:0
   AH replay violation:0
   ESP replay violation:  0
 Total outgoing packets dropped:0
   Filter denies on input:0
 Tunnel cache entries added: 7
 Tunnel cache entries expired:  0
 Tunnel cache entries deleted:  6
```

Remarque : Depuis AIX 4.3.3, la prise en charge CDMF a été supprimée car DES est désormais disponible dans le monde entier. Reconfigurez tous les tunnels qui utilisent CDMF en DES ou Triple DES.

Informations de référence sur la fonction de sécurité IP

Liste des commandes

ikecmd=activate	Démarre une négociation IKE (Internet Key Exchange) (AIX versions 4.3.2 et versions supérieures).
ikecmd=remove	Désactive les tunnels IKE (AIX versions 4.3.2 et versions supérieures)
ike cmd=list	Répertorie les tunnels IKE (AIX versions 4.3.2 et versions supérieures)
ikedb	Fournit l'interface vers la base de données des tunnels IKE (AIX versions 5.1 et versions supérieures)
gentun	Crée une définition de tunnel
mktun	Active une ou plusieurs définitions de tunnel
chtun	Change la définition d'un tunnel
rmtun	Supprime la définition d'un tunnel
lstun	Répertorie une ou plusieurs définitions de tunnel
exptun	Exporte une ou plusieurs définitions de tunnel
imptun	Importe une ou plusieurs définitions de tunnel
genfilt	Crée une définition de filtre
mkfilt	Active une ou plusieurs définitions de filtre
mvfilt	Déplace une règle de filtrage
chfilt	Change une définition de filtre
rmfilt	Supprime une définition de filtre
lsfilt	Répertorie une ou plusieurs définitions de filtre
expfilt	Exporte une ou plusieurs définitions de filtre
impfilt	Importe une ou plusieurs définitions de filtre
ipsec_convert	Indique l'état de la sécurité IP
ipsecstat	Indique l'état de la sécurité IP
ipsectrcbuf	Indique le contenu du tampon de suivi de la sécurité IP
unloadipsec	Décharge un module de chiffrement

Liste des méthodes

defipsec	Définit une instance de sécurité IP pour IP version 4 ou IP version 6
cfgipsec	Configure et charge **ipsec_v4** ou **ipsec_v6**
ucfgipsec	Supprime la configuration de **ipsec_v4** ou **ipsec_v6**

Cette section explique comment migrer vos tunnels IKE, filtres et clés pré–partagées de AIX 4.3 vers AIX 5.2.

Migration de la sécurité IP

Pour la migration de vos tunnels, effectuez les étapes suivantes sur un système fonctionnant avec AIX 4.3 :

1. Exécutez le script **bos.net.ipsec.keymgt.pre_rm.sh**. Lorsque vous exécutez ce script, les fichiers suivants sont créés dans le répertoire **/tmp** :
   a. **p2proposal.bos.net.ipsec.keymgt**
   b. **p1proposal.bos.net.ipsec.keymgt**
   c. **p1policy.bos.net.ipsec.keymgt**
   d. **p2policy.bos.net.ipsec.keymgt**
   e. **p1tunnel.bos.net.ipsec.keymgt**
   f. **p2tunnel.bos.net.ipsec.keymgt**

   Remarque : Exécutez ce script une seule fois. Si vous mettez à jour la base de données et exécutez de nouveau le script, vous perdrez tous vos fichiers et vous ne pourrez pas les récupérer. Lisez le script qui se trouve dans Le script bos.net.ipsec.keymgt.pre_rm.sh, page 12-66, avant de migrer vos tunnels.
2. Enregistrez les fichiers créés par le script et le fichier **/tmp/lpplevel** sur un support externe, comme un CD ou une disquette.

Migration de clés pré–partagées

La base de données de clés pré–partagées de tunnel IKE est aussi altérée pendant la migration. Pour mettre à jour le format de clés pré–partagées, effectuez les étapes suivantes sur le système dont la migration vers AIX 5.2 a été effectuée :

1. Enregistrez la sortie de la commande **ikedb –g** en exécutant la commande suivante :

```
ikedb –g > out.keys
```

2. Modifiez le fichier **out.keys** pour remplacer **FORMAT=ASCII** par **FORMAT=HEX** pour le format de clés pré–partagé.
3. Entrez le fichier XML en exécutant la commande suivante :

```
ikedb –pF out.keys
```

Migration de filtres

1. Exportez les fichiers de règles de filtrage vers le répertoire **/tmp** à l'aide de SMIT en effectuant les étapes suivantes :
   a. Exécutez la commande **smitty ipsec4**.
   b. Sélectionnez **Configuration avancée de la sécurité IP—>Configuration des règles de filtrage de sécurité IP—>Exportation de règles de filtrage de sécurité IP**.
   c. Entrez **/tmp** comme nom de répertoire.
   d. Sous l'option **Règles de filtrage**, appuyez sur F4 et sélectionnez **tous** dans la liste.
   e. Appuyez sur Entrée pour enregistrer les règles de filtrage dans le fichier **/tmp/ipsec_fltr_rule.exp** qui se trouve sur le support externe.

   Effectuez cette opération pour tous les systèmes que vous migrez de AIX 4.3 vers AIX 5.2.
2. Copiez les six fichiers tunnel créés par le script, le fichier **/tmp/lpplevel** et le fichier **/tmp/ipsec_fltr_rule.exp** vers le répertoire **/tmp** sur le système migré.
3. Exécutez le script **bos.net.ipsec.keymgt.post_i.sh** pour entrer de nouveau les configurations de tunnel dans la base de données.

4. Exécutez la commande **ikedb –g** pour vérifier que les tunnels sont dans la base de données.

   Remarque : Si vous ne voyez pas les informations de tunnel dans la base de données, exécutez de nouveau le script, mais renommez tous les fichiers ***.loaded** dans le répertoire **/tmp** par leurs noms d'origine.

Sur un système dont la migration a été effectuée vers AIX 5.2, la base de données de filtres est altérée après la migration. Si vous exécutez la commande **lsfilt** sur le système migré, vous obtiendrez l'erreur suivante :

```
Cannot get ipv4 default filter rule
```

Pour mettre à jour la base de données de filtres, effectuez les opérations suivantes :

1. Remplacez le filtre **ipsec_filter** et le filtre **ipsec_filter.vc** dans le répertoire **/etc/security** par les fichiers non altérés, issus d'un système nouvellement migré utilisant AIX 5.2. Si vous ne disposez pas de ces fichiers, vous pouvez en faire la demande auprès de votre technicien de maintenance.
2. Importez les fichiers de règles de filtrage vers le répertoire **/tmp** à l'aide de SMIT en effectuant les étapes suivantes :
   a. Exécutez la commande **smitty ipsec4**.
   b. Sélectionnez **Configuration avancée de la sécurité IP—>Configuration des règles de filtrage de sécurité IP—>Importation de règles de filtrage de sécurité IP**.
   c. Entrez **/tmp** comme nom de répertoire.
   d. Sous l'option **Règles de filtrage**, appuyez sur F4 et sélectionnez **tous** dans la liste.
   e. Appuyez sur Entrée pour créer de nouveau les règles de filtrage. Vous pouvez afficher la liste des règles de filtrage via SMIT ou via la commande **lsfilt**.

Scripts de migration

Le script bos.net.ipsec.keymgt.pre_rm.sh

Le script **bos.net.ipsec.keymgt.pre_rm.sh** enregistre le contenu de la base de données de tunnel sur un système utilisant AIX 4.3.

```
#!/usr/bin/ksh
 keymgt_installed=‘lslpp -Lqc bos.net.ipsec.keymgt 2>/dev/null | awk -F:
’{print $6}’ | head -1‘

 if [ ! ”$keymgt_installed” ]
 then
   exit 0
 fi


 #  Copy the database to a save directory in case changes fail
 if [ -d /etc/ipsec/inet/DB ]
 then
   cp -R /etc/ipsec/inet/DB /etc/ipsec/inet/DB.sav || exit $?
 fi

 #  Remember the level you are migrating from
 VRM=$(LANG=C lslpp -Lqc bos.net.ipsec.keymgt 2>/dev/null | awk -F: ’{print
$3}’ | \
 awk -F. ’{print $1”.”$2”.”$3}’)
 VR=${VRM%.*}
 echo $VRM > /tmp/lpplevel

 IKEDB=$(which ikedb) || IKEDB=/usr/sbin/ikedb

 XMLFILE=/tmp/full_ike_database.bos.net.ipsec.keymgt
 PSKXMLFILE=/tmp/psk_ike_database.bos.net.ipsec.keymgt
```

```
 #  See if ikedb exists.
 if [ -f $IKEDB ]
 then

   # If either of the ikedb calls below fails, that's OK.  Just remove the
   # resulting file (which may contain garbage) and continue.  The post_i
   # script will simply not import the file if it doesn't exist, which will
   # mean part or all of the IKE database is lost, but this is preferable
   # to exiting the script with an error code, which causes the entire
   # migration to fail.

   $IKEDB -g > $XMLFILE
   if [ $? -ne 0 ]
   then
     rm -f $XMLFILE  || exit $?
   fi

   if [[ $VR = "5.1" ]]; then
     # This is a special case.  The 5.1 version of ikedb is the only
     # one that does not include preshared keys in the full database
     # output.  So we have to retrieve those separately.
     $IKEDB -g -t IKEPresharedKey > $PSKXMLFILE
     if [ $? -ne 0 ]
     then
      rm -f $PSKXMLFILE  || exit $?
     fi
   fi

 #  Make sure ikegui command is installed
 elif [ -f /usr/sbin/ikegui ]
 then

   #  Get database information and save to /tmp
   /usr/sbin/ikegui 0 1 0 0 > /tmp/p1proposal.bos.net.ipsec.keymgt
2>/dev/null
   RC=$?
   if [[ $RC -ne 0 ]]
   then
     rm -f /tmp/p1proposal.bos.net.ipsec.keymgt || exit $?
   fi

   /usr/sbin/ikegui 0 1 1 0 > /tmp/p1policy.bos.net.ipsec.keymgt
2>/dev/null
   RC=$?
   if [[ $RC -ne 0 ]]
   then
     rm -f /tmp/p1policy.bos.net.ipsec.keymgt || exit $?
   fi

   /usr/sbin/ikegui 0 2 0 0 > /tmp/p2proposal.bos.net.ipsec.keymgt
2>/dev/null
   RC=$?
   if [[ $RC -ne 0 ]]
   then
     rm -f /tmp/p2proposal.bos.net.ipsec.keymgt || exit $?
   fi

   /usr/sbin/ikegui 0 2 1 0 > /tmp/p2policy.bos.net.ipsec.keymgt
2>/dev/null
   RC=$?
   if [[ $RC -ne 0 ]]
   then
     rm -f /tmp/p2policy.bos.net.ipsec.keymgt || exit $?
   fi

   /usr/sbin/ikegui 0 1 2 0 > /tmp/p1tunnel.bos.net.ipsec.keymgt
2>/dev/null
   RC=$?
   if [[ $RC -ne 0 ]]
   then
     rm -f /tmp/p1tunnel.bos.net.ipsec.keymgt || exit $?
```

```
   fi

   /usr/sbin/ikegui 0 2 2 0 > /tmp/p2tunnel.bos.net.ipsec.keymgt
2>/dev/null
   RC=$?
   if [[ $RC -ne 0 ]]
   then
     rm -f /tmp/p2tunnel.bos.net.ipsec.keymgt || exit $?
   fi

 fi
```

Le script bos.net.ipsec.keymgt.pre_rm.sh

Le script **bos.net.ipsec.keymgt.post_i.sh** sauvegarde le contenu de la base de données de tunnel sur un système migré fonctionnant sur AIX 5.2.

```
#!/usr/bin/ksh

 function PrintDot {
     echo "echo \c"
     echo "\".\c"
     echo "\\\c\c"
     echo "\"\c"
     echo
 }

 function P1PropRestore {
     while :
     do
         read NAME
         read MODE
         if [[ $? = 0 ]]; then
             echo "ikegui 1 1 0 $NAME $MODE \c"
             MORE=1
             while [[ $MORE = 1 ]];
             do
                 read AUTH
                 read HASH
                 read ENCRYPT
                 read GROUP
                 read TIME
                 read SIZE
                 read MORE
                 echo "$AUTH $HASH $ENCRYPT $GROUP $TIME $SIZE $MORE \c"
             done
             echo " > /dev/null 2>&1"
             PrintDot
         else
             return 0
         fi
     done
 }

 function P2PropRestore {
     while :
     do
         read NAME
         FIRST=yes
         MORE=1
         while [[ $MORE = 1 ]];
         do
             read PROT
             if [[ $? = 0 ]]; then
                 read AH_AUTH
                 read ESP_ENCR
                 read ESP_AUTH
                 read ENCAP
                 read TIME
                 read SIZE
```

```
                read MORE
                if [[ $FIRST = "yes" ]]; then
                    echo "ikegui 1 2 0 $NAME $MODE \c"
                fi
                echo "$PROT $AH_AUTH $ESP_ENCR $ESP_AUTH $ENCAP $TIME
$SIZE $MORE \c"
                FIRST=no
            else
                return 0
            fi
        done
        echo " > /dev/null 2>&1"
        PrintDot
    done
 }

 function P1PolRestore {
    while :
    do
        read NAME
        read ROLE
        if [[ $? = 0 ]]; then
            read TIME
            read SIZE
            read OVERLAP
            read TTIME
            read TSIZE
            read MIN
            read MAX
            read PROPOSAL
            echo "ikegui 1 1 1 $NAME $ROLE $OVERLAP $TTIME $TSIZE $MIN
$MAX 1 0 0 $PROPOSAL > \
 /dev/null 2>&1"
            PrintDot
        else
            return 0
        fi
    done
 }

 function P2PolRestore {
    while :
    do
        read NAME
        read ROLE
        if [[ $? = 0 ]]; then
            read IPFS
            read RPFS
            read TIME
            read SIZE
            read OVERLAP
            read TTIME
            read TSIZE
            read MIN
            read MAX
            echo "ikegui 1 2 1 $NAME $ROLE $IPFS $RPFS $OVERLAP $TTIME
$TSIZE $MIN $MAX 1 0 0 \c"
            MORE=1
            while [[ $MORE = 1 ]];
            do
                read PROPOSAL
                read MORE
                echo "$PROPOSAL $MORE \c"
                FIRST=no
            done
        else
            return 0
        fi
        echo " > /dev/null 2>&1"
        PrintDot
    done
```

```
}

function P1TunRestore {
    while :
    do
        read TUNID
        read NAME
        if [[ $? = 0 ]]; then
            read LID_TYPE
            read LID
            if [[ $LPPLEVEL = ”4.3.3” ]]; then
                read LIP
            fi
            read RID_TYPE
            read RID
            read RIP
            read POLICY
            read KEY
            read AUTOSTART
            echo ”ikegui 1 1 2 0 $NAME $LID_TYPE \”$LID\” $LIP $RID_TYPE
\”$RID\” \
 $RIP $POLICY $KEY $AUTOSTART > /dev/null 2>&1”
            PrintDot
        else
            return 0
        fi
    done
}

function P2TunRestore {
    while :
    do
        read TUNID
        read NAME
        if [[ $? = 0 ]]; then
            read P1TUN
            read LTYPE
            read LID
            read LMASK
            read LPROT
            read LPORT
            read RTYPE
            read RID
            read RMASK
            read RPROT
            read RPORT
            read POLICY
            read AUTOSTART
            echo ”ikegui 1 2 2 0 $NAME $P1TUN $LTYPE $LID $LMASK $LPROT
$LPORT $RTYPE \
 $RID $RMASK $RPROT $RPORT $POLICY $AUTOSTART > /dev/null 2>&1”
            PrintDot
        else
            return 0
        fi
    done
}

function allRestoreWithIkedb {

    ERRORS=/tmp/ikedb_msgs.bos.net.ipsec.keymgt
    echo > $ERRORS
    $IKEDB -p $XMLFILE 2>> $ERRORS
    if [ -f $PSKXMLFILE ]
    then
        $IKEDB -p $PSKXMLFILE 2>> $ERRORS
    fi

}

P1PROPFILE=/tmp/p1proposal.bos.net.ipsec.keymgt
```

```
P2PROPFILE=/tmp/p2proposal.bos.net.ipsec.keymgt
P1POLFILE=/tmp/p1policy.bos.net.ipsec.keymgt
P2POLFILE=/tmp/p2policy.bos.net.ipsec.keymgt
P1TUNFILE=/tmp/p1tunnel.bos.net.ipsec.keymgt
P2TUNFILE=/tmp/p2tunnel.bos.net.ipsec.keymgt
XMLFILE=/tmp/full_ike_database.bos.net.ipsec.keymgt
PSKXMLFILE=/tmp/psk_ike_database.bos.net.ipsec.keymgt
CMD_FILE=/tmp/commands
IKEDB=$(which ikedb) || IKEDB=/usr/sbin/ikedb

echo "building ISAKMP database \n"
$IKEDB -x || exit $?

if [ -f $XMLFILE ]; then
    echo "\nRestoring database entries\c"
    allRestoreWithIkedb
    echo "\ndone\n"

elif [ -f /tmp/*.bos.net.ipsec.keymgt ]; then
    echo "\nRestoring database entries\c"

    LPPLEVEL=`cat /tmp/lpplevel`

    echo > $CMD_FILE
    touch $P1PROPFILE; P1PropRestore < $P1PROPFILE >> $CMD_FILE
    touch $P2PROPFILE; P2PropRestore < $P2PROPFILE >> $CMD_FILE
    touch $P1POLFILE; P1PolRestore < $P1POLFILE >> $CMD_FILE
    touch $P2POLFILE; P2PolRestore < $P2POLFILE >> $CMD_FILE
    touch $P1TUNFILE; P1TunRestore < $P1TUNFILE >> $CMD_FILE
    touch $P2TUNFILE; P2TunRestore < $P2TUNFILE >> $CMD_FILE

    mv $P1PROPFILE ${P1PROPFILE}.loaded
    mv $P2PROPFILE ${P2PROPFILE}.loaded
    mv $P1POLFILE ${P1POLFILE}.loaded
    mv $P2POLFILE ${P2POLFILE}.loaded
    mv $P1TUNFILE ${P1TUNFILE}.loaded
    mv $P2TUNFILE ${P2TUNFILE}.loaded

    ksh $CMD_FILE

    echo "done\n"
fi
```

Chapitre 13. Sécurité NIS (Network Information Services) et NIS+

Ce chapitre fournit une vue d'ensemble de la manière dont NIS+ protège son espace de nom et comprend les sections suivantes :

Méthodes de protection du système d'exploitation

La protection du système d'exploitation est assurée par des portes que les utilisateurs doivent franchir avant d'entrer dans l'environnement du système d'exploitation et par des tableaux de droits d'accès qui déterminent ce qu'ils sont autorisés à faire dans cet environnement. Dans certains contextes, les mots de passe *RPC sécurisés* sont appelés *mots de passe réseau*.

Le système complet comprend quatre portes et deux tableaux de droits d'accès :

Porte de numérotation
Pour accéder à un environnement de système d'exploitation donné depuis l'extérieur à l'aide d'un modem et d'une ligne téléphonique, vous devez fournir un mot de passe de numérotation et un ID de connexion valides.

Porte de connexion
Pour accéder à un environnement de système d'exploitation donné, vous devez fournir un mot de passe utilisateur et un ID de connexion valides.

Porte root Pour bénéficier des droits d'accès root, vous devez fournir un mot de passe utilisateur root valide.

Porte RPC sécurisée
Dans un environnement NIS+ fonctionnant au niveau de sécurité 2 (valeur par défaut), lorsque vous essayez d'utiliser des services NIS+ et d'accéder aux objets NIS+ (serveurs, répertoires, tables, entrées de tables, etc), NIS+ confirme votre identité à l'aide du processus RPC sécurisé.

Le franchissement d'une porte RPC sécurisée nécessite un mot de passe RPC sécurisé. Votre mot de passe RPC sécurisé et votre mot de passe de connexion sont généralement identiques. Si tel est le cas, vous franchissez automatiquement la porte sans avoir à entrer de nouveau votre mot de passe. Dans certains contextes, les mots de passe *RPC sécurisés* sont appelés *mots de passe réseau* Reportez–vous à la section Secure RPC Password versus Login Password dans le manuel *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide* pour plus d'informations sur la manière de gérer deux mots de passe qui ne sont pas identiques.

Un ensemble de *données d'identification* est utilisé pour transmettre vos requêtes automatiquement via la porte RPC sécurisée. Le processus qui génère, présente et valide vos données d'identification est nommé *authentification* car il confirme votre identité et le fait que vous disposiez d'un mot de passe RPC sécurisé valide. Cette authentification est automatiquement effectuée à chaque requête au service NIS+.

Dans un environnement NIS+ fonctionnant en mode de compatibilité NIS, la protection fournie par la porte RPC sécurisée est limitée. Chacun dispose des droits d'accès en lecture sur tous les objets NIS+ et des droits de modifier les entrées qui les concernent, qu'ils disposent ou non de données d'identification valides (c'est–à–dire, peu importe que le processus d'authentification ait ou non confirmé leur identité et validé leur mot de passe RPC sécurisé). Comme cette situation permet à *tous* d'accéder en lecture à tous les objets NIS+ et de modifier les entrées qui les concernent, un réseau NIS+ fonctionnant en mode de compatibilité est moins sécurisé qu'un même réseau en mode normal. En terminologie RPC sécurisé, tout utilisateur sans référence valide est considéré comme membre de la classe **nobody**. Reportez–vous à la section Classes d'autorisation, page 13-9 pour une description détaillée de chacune des classes.)

Pour plus d'informations sur la gestion des authentifications et données d'identification NIS+, reportez–vous à la section Administering NIS+ Credentials dans le manuel *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*.

Matrice de fichiers et répertoires

Une fois que vous avez accédé à un environnement de système d'exploitation, les droits d'accès applicables définissent votre capacité à lire, exécuter, modifier, créer et détruire des fichiers et répertoires.

Matrice d'objets NIS+

Une fois que vous avez été authentifié correctement auprès de NIS+, les droits d'accès applicables régissent votre capacité à lire, exécuter, modifier, créer et détruire des objets NIS+. Ce processus est appelé *autorisation NIS+*.

Pour plus d'informations sur les permissions et autorisations NIS+, reportez–vous à la section Administering NIS+ Access Rights dans le manuel *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*.

Systèmes de sécurité NIS+

La sécurité NIS+ fait partie intégrante de l'espace de nom NIS+. Vous ne pouvez pas configurer la sécurité indépendamment de l'espace de nom. Les instructions de configuration de la sécurité sont donc réparties dans les étapes de configuration des autres éléments de l'espace de nom. Une fois qu'un environnement de sécurité NIS+ a été configuré, vous pouvez ajouter et supprimer des utilisateurs, modifier les droits d'accès, modifier la répartition des membres de groupes et exécuter toutes les autres tâches de gestion du réseau.

Les fonctions de sécurité de NIS+ protègent la structure l'espace de nom et les informations qu'il contient contre les accès non autorisés. Sans ces fonctions, tout client NIS+ pourrait obtenir, modifier, voire endommager les informations stockées dans l'espace de nom.

La sécurité NIS+ remplit deux objectifs :

Authentification
L'authentification permet d'identifier les principaux NIS+. Chaque fois qu'un principal (un utilisateur ou un poste) tente d'accéder à un objet NIS+, l'identité et le mot de passe RPC sécurisé de l'utilisateur sont confirmés et validés. Vous n'avez normalement pas besoin d'entrer un mot de passe dans le cadre de l'authentification. Cependant, si votre mot de passe RPC sécurisé est différent de votre mot de passe de connexion, vous devez exécuter un **keylogin** lors de votre première tentative d'accès aux objets ou services NIS+. Pour exécuter un **keylogin**, vous devez fournir un mot de passe RPC sécurisé valide. Reportez–vous à la section Secure RPC Password versus Login Password dans le guide *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*.)

Autorisation L'autorisation permet de définir des droits d'accès. Chaque fois qu'un principal NIS+ tente d'accéder à des objets NIS+, il est placé dans l'une des quatre classes d'autorisation (propriétaire, groupe, monde, personne). Le système de sécurité NIS+ permet aux administrateurs NIS+ de spécifier différents droits de lecture, modification, création, ou destruction sur les objets NIS+ pour chaque classe. Par exemple, une classe donnée pourrait être autorisée à modifier une colonne spécifique de la table passwd mais pas à lire cette colonne, ou bien, une autre classe pourrait être autorisée à lire des entrées d'une table spécifique mais pas des autres.

Par exemple, les informations d'une table NIS+ donnée peuvent être lues et modifiées par une classe, seulement lues par une autre classe, et ni lues ni modifiées par une troisième classe. Le concept est identique à celui des droits d'accès aux fichiers et répertoires du système d'exploitation. Pour plus d'informations sur les classes, reportez–vous à la section Classes d'autorisation, page 13-9.

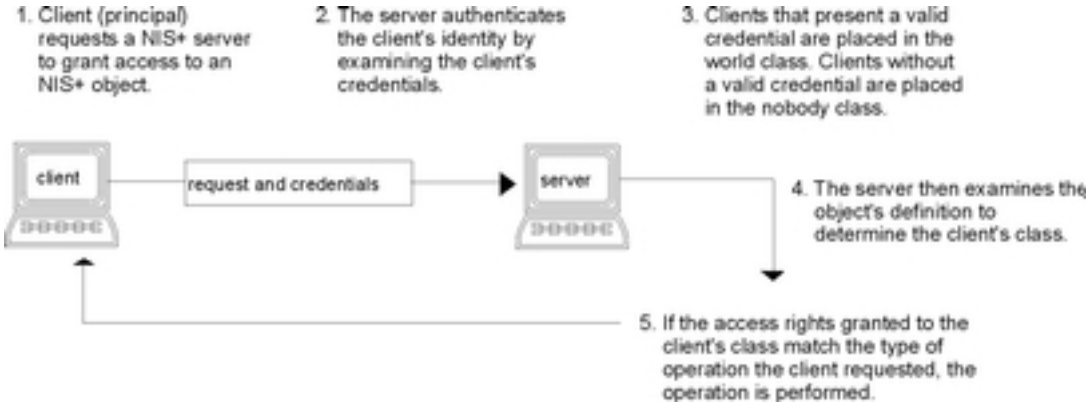
L'authentification et l'autorisation empêchent un utilisateur bénéficiant de droits d'accès root sur un poste A d'utiliser la commande **su** pour prendre l'identité d'un autre utilisateur non connecté, ou connecté sur un poste B, et d'accéder aux objets NIS+ avec les droits d'accès NIS+ de l'autre utilisateur.

Cependant, NIS+ ne peut empêcher un utilisateur connaissant le mot de passe de connexion d'un autre utilisateur de prendre son identité et de bénéficier de ses droits d'accès NIS+. NIS+ ne peut pas non plus empêcher un utilisateur bénéficiant de droits d'accès root de prendre l'identité d'un autre utilisateur connecté depuis le *même* poste.

La figure suivante illustre ce processus.

Figure 15. Résumé du processus de sécurité NIS+ Ce schéma illustre le processus de sécurité NIS+.

1. Le client/principal envoie une requête au serveur NIS+ pour accéder à un objet NIS+.
2. Le serveur authentifie l'identité du client en examinant ses données d'identification.
3. Les clients dont les données d'identification sont valides sont placés dans la classe monde (world).
4. Les clients dont les données d'identification ne sont pas valides sont placés dans la classe personne (nobody).
5. Le serveur examine la définition de l'objet pour déterminer la classe du client.
6. Si les droits d'accès de la classe du client correspondent au type d'opération demandée, l'opération est exécutée.



Principaux NIS+

Les principaux NIS+ sont les entités (clients) qui envoient des requêtes de services NIS+. Un principal NIS+ peut être une personne connectée à un poste client en tant qu'utilisateur courant ou utilisateur root, ou tout processus fonctionnant avec des droits d'accès root sur un poste client NIS+. Ainsi, un principal NIS+ peut être un utilisateur client ou un poste de travail client.

Un principal NIS+ peut aussi être l'entité qui fournit un service NIS+ depuis un serveur NIS+. Tous les serveurs NIS+ étant aussi des clients NIS+, la plupart des sujets couverts s'appliquent aussi aux serveurs.

Niveaux de sécurité NIS+

Les serveurs NIS+ fonctionnent dans l'un des deux niveaux de sécurité. Ces niveaux déterminent les types de références que les principaux doivent fournir pour authentifier leurs requêtes. NIS+ est conçu pour fonctionner au niveau de sécurité le plus élevé, qui est le niveau 2. Le niveau 0 n'est utilisé qu'à des fins de test, d'installation et de débogage. Ces niveaux de sécurité sont résumés dans le tableau suivant.

Remarque : Utilisez Web–based System Manager, SMIT ou la commande **passwd** pour modifier votre mot de passe indépendamment du niveau de sécurité ou de l'état des données d'identification.

Niveaux de sécurité NIS+

Niveau de sécurité	Description
0	Le niveau de sécurité 0 est prévu pour les tests et la configuration de l'espace de nom NIS+ initial. Un serveur NIS+ fonctionnant au niveau 0 accorde à tout principal NIS+ des droits d'accès complets à tous les objets NIS+ du domaine. Le niveau 0 est uniquement destiné à la configuration et ne doit être utilisé que par les administrateurs et dans ce but. Le niveau 0 ne doit *pas* être utilisé sur des réseaux en fonctionnement normal par des utilisateurs courants.
1	Le niveau de sécurité 1 utilise la sécurité AUTH_SYS. Ce niveau n'est pas pris en charge par NIS+ et ne doit *pas* être utilisé.
2	Le niveau de sécurité 2 est le niveau par défaut. C'est le plus haut niveau de sécurité de NIS+. Il n'authentifie que les requêtes qui utilisent les données d'identification DES (data encryption standard). Les requêtes sans donnée d'identification sont affectées à la classe personne et disposent des droits d'accès correspondants. Les requêtes utilisant des données d'identification DES non valides sont réessayées. Après un nouvel échec d'obtention de données d'identification DES valides, les requêtes échouent, accompagnées d'une erreur d'authentification. Une donnée d'identification peut ne pas être valide pour diverses raisons : le principal peut ne pas être connecté via **keylogin** sur le poste, les horloges peuvent être désynchronisées, il peut y avoir une non–correspondance de clé, etc.

Authentification et données d'identification NIS+

Les données d'identification NIS+ authentifient l'identité de chaque principal qui envoie une requête de service NIS+ ou accède à un objet NIS+. Le processus de d'identification/autorisation NIS+ est une implémentation du système RPC sécurisé.

Le système de données d'identification/autorisation empêche un utilisateur de prendre l'identité d'un autre. Il empêche un utilisateur avec des droits d'accès root sur un poste d'utiliser la commande **su** pour prendre l'identité d'un autre utilisateur non connecté ou connecté sur un autre poste, et d'accéder aux objets NIS+ avec les droits d'accès NIS+ de l'autre utilisateur.

Remarque : NIS+ ne peut pas empêcher un utilisateur qui connaît le mot de passe de connexion d'un autre utilisateur de prendre l'identité et les droits d'accès NIS+ de cet utilisateur. NIS+ ne peut pas non plus empêcher un utilisateur bénéficiant de droits d'accès root de prendre l'identité d'un autre utilisateur connecté sur le *même* poste.

Un fois un principal authentifié par un serveur, ce serveur contrôle l'objet NIS+ auquel le principal souhaite accéder pour savoir quelles opérations lui sont accessibles. (Reportez–vous à la section Autorisation et accès NIS+, page 13-9, pour plus d'informations sur les autorisations.)

Données d'identification des utilisateurs et des postes

Il existe deux types de principal, les *utilisateurs* et les *postes*, et donc deux types de données d'identification :

Données d'identification des utilisateurs
: Lorsqu'une personne est connectée à un client NIS+ en tant qu'utilisateur courant, les requêtes de services NIS+ comprennent ses données d'identification.

Données d'identification des postes
: Lorsqu'un utilisateur est connecté à un client NIS+ en tant qu'utilisateur root, les requêtes de services utilisent les données d'identification du poste client.

Données d'identification locales et DES

Les principaux NIS+ peuvent disposer de données d'identification DES ou locales.

Données d'identification DES

Les données d'identification DES (Data Encryption Standard) assurent une authentification sécurisée. Lorsque ce manuel indique que NIS+ contrôle des données d'identification pour authentifier un principal NIS+, cela veut dire que NIS+ valide des données d'identification DES.

Remarque : L'utilisation de données d'identification DES n'est qu'une méthode d'authentification parmi d'autres. Ne confondez pas les données d'identification DES avec les données d'identification NIS+.

Chaque fois qu'un principal formule une demande de service NIS+ ou accède à un objet NIS+, le logiciel utilise les informations d'identification stockées pour ce principal pour générer ses données d'identification. Les données d'identification sont générées à partir des informations créées pour chaque principal par un administrateur NIS+, comme expliqué dans la section Administering NIS+ Credentials dans le manuel *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*.

- Une fois la validité des données d'identification DES d'un principal confirmée par NIS+, ce principal est *authentifié*.

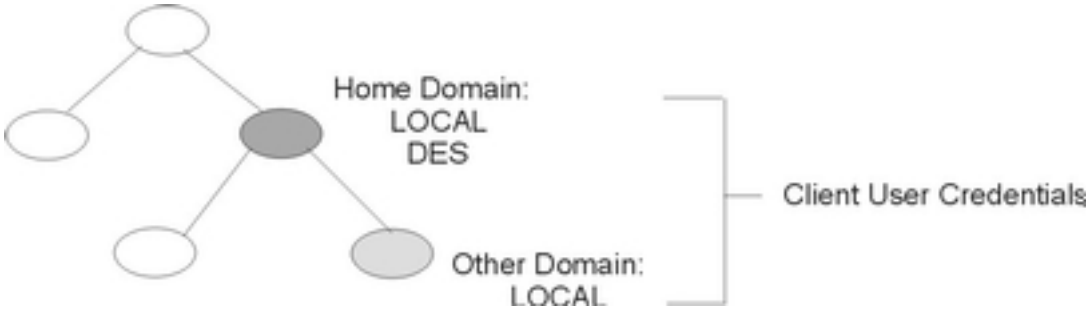
- Un principal doit être authentifié avant d'être placé dans la classe propriétaire, groupe, ou monde. C'est–à–dire que vous devez disposer de données d'identification DES valides pour être placé dans l'une de ces classes. Les principaux sans données d'identification DES valides sont automatiquement placés dans la classe personne.
- Les informations sur les données d'identification DES sont toujours stockées dans la table cred du domaine d'accueil du principal, que ce dernier soit un utilisateur client ou un poste de travail client.

Données d'identification locales

Les données d'identification locales sont une mappe entre un numéro d'ID utilisateur et son nom de principal NIS+, qui comprend son nom de domaine d'accueil. Lorsque des utilisateurs se connectent, le système contrôle leurs données d'identification, et identifie leur domaine d'accueil où sont stockées leurs données d'identification DES. Le système utilise ces informations pour obtenir les données d'identification DES des utilisateurs.

Lorsque des utilisateurs se connectent à un domaine distant, ces requêtes utilisent leurs données d'identification locales qui font référence à leur domaine d'accueil. NIS+ interroge alors le domaine d'accueil de l'utilisateur pour obtenir ses données d'identification DES. L'utilisateur peut ainsi être authentifié sur un domaine distant bien que ses données d'identification DES n'y soient pas stockées. La figure suivante illustre ce concept.

Figure 16. Données d'identification et domaines. Ce schéma représente une hiérarchie de domaine. Le domaine d'accueil de l'utilisateur contient les données d'identification locales et DES. Le sous–domaine ne contient que les données d'identification locales. Le domaine d'accueil et le sous–domaine sont indiqués comme Données d'identification de l'utilisateur client.



Les données d'identification locales peuvent être stockées dans n'importe quel domaine. Pour se connecter à un domaine distant et être authentifié, un utilisateur client *doit* disposer des données d'identification locales dans la table cred du domaine distant. Si ce n'est pas le cas, NIS+ ne peut pas localiser son domaine d'accueil pour obtenir ses données d'identification DES. L'utilisateur ne pourrait alors pas être authentifié et serait placé dans la classe personne.

Types d'utilisateurs et types de données d'identification

Un utilisateur peut disposer des deux types de données d'identification, mais un poste peut *seulement* avoir des données d'identification DES.

Les utilisateurs root ne peuvent accéder par NIS+ aux autres postes, en tant que root, car l'UID root de chaque poste est toujours zéro. Si un utilisateur root (UID=0) du poste A tente d'accéder au poste B en tant qu'utilisateur root, il entre en conflit avec les utilisateurs root existants (UID=0) du poste B. Des données d'identification locales ne sont donc pas appropriées pour un *poste de travail* client. Elles ne conviennent que pour les *utilisateurs* clients.

Autorisation et accès NIS+

La principale fonction de l'autorisation NIS+ est de préciser, pour chaque principal NIS+, les droits d'accès à chaque objet et service NIS+.

Une fois authentifié, un principal qui émet une requête NIS+ est placé dans une classe d'autorisation. Les droits d'accès (autorisations) qui précisent les opérations qui peuvent être effectuées par un principal sur un objet NIS+ donné sont définis en fonction des classes. En d'autres termes, les droits d'accès varient d'une classe d'autorisation à l'autre.

Classes d'autorisation
: Il en existe quatre : Propriétaire (owner), groupe (group), monde (world) et personne (nobody). (Reportez–vous à la section Classes d'autorisation, page 13-9 pour plus d'informations sur les classes.)

Droits d'accès Il sont de quatre types : création, destruction, modification et lecture. (Reportez–vous à la section Droits d'accès NIS+, page 13-12 pour plus d'informations.)

Classes d'autorisation

Les objets NIS+ ne confèrent pas directement de droit d'accès aux principaux NIS+. Ils accordent des droits d'accès aux quatre *classes* de principaux suivants :

Propriétaire Le principal qui est propriétaire (owner) de l'objet dispose des droits accordés à la classe des propriétaires.

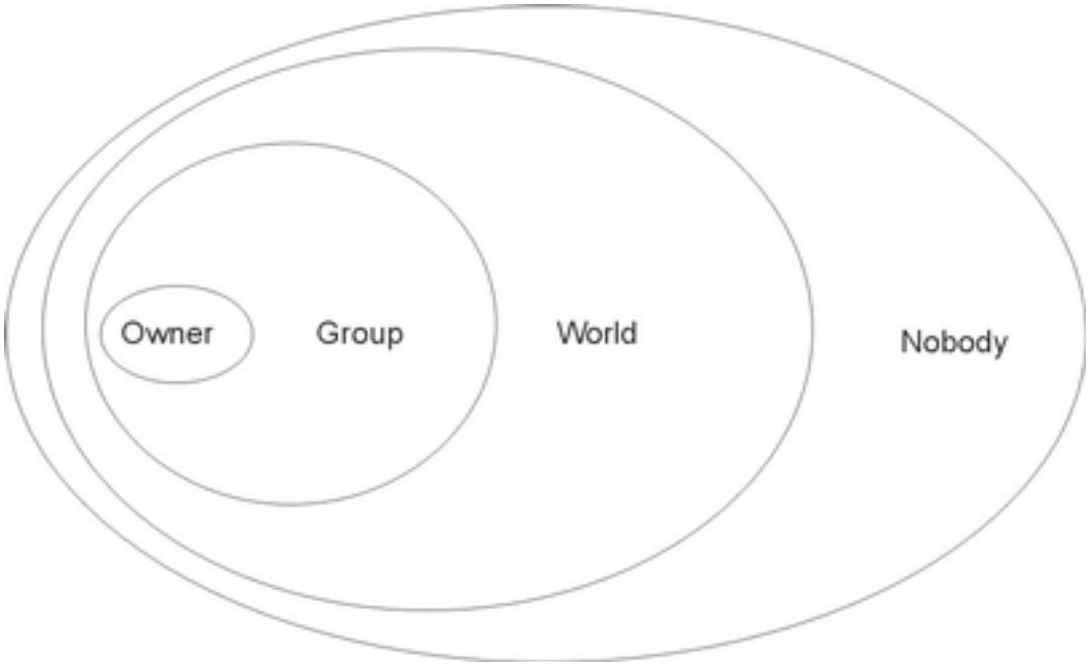
Groupe A chaque objet NIS+ est associé un groupe (group). Les membres du groupe d'un objet sont désignés par l'administrateur NIS+. Les principaux qui appartiennent à la classe Group d'un objet disposent des droits accordés à cette classe (ici, le terme *groupes* désigne les groupes NIS+, et non des groupes de système d'exploitation ou réseau. Pour obtenir une description des groupes NIS+, reportez–vous à la section Classe de groupe, page 13-11.

Monde Cette classe (world) regroupe tous les principaux NIS+ authentifiés par un serveur, c'est–à–dire tous les principaux authentifiés mais n'appartenant ni à la classe Propriétaire ni à la classe Groupe.

Personne Tous les principaux font partie de cette classe (nobody), y compris ceux qui n'ont pas été authentifiés.

La figure suivante illustre la relation entre les classes :

Figure 17. Classes d'autorisation Ce schéma montre les relations entre les classes d'autorisation. Le plus petit ensemble représente le groupe Propriétaire ; il est inclus dans l'ensemble Groupe. Celui–ci est inclus dans le groupe Monde, lui–même inclus dans le groupe Personne.



A chaque requête NIS+, le système détermine à quelle classe appartient le principal à l'origine de la requête. Celui–ci peut ensuite utiliser tous les droits d'accès inhérents à cette classe.

Un objet peut accorder n'importe quelle combinaison de droits d'accès à chaque classe. En général, une classe supérieure dispose généralement des mêmes droits qu'une classe inférieure, et éventuellement de droits supplémentaires.

Par exemple, un objet peut attribuer un accès en lecture aux classes Personne et Monde, un accès en lecture et modification à la classe Groupe, et un accès en lecture, modification, création et destruction à la classe Propriétaire.

L'exemple suivant décrit en détail les classes d'autorisation :

Classe Propriétaire

Le propriétaire est un principal NIS+ *unique*.

Les principaux qui présentent une requête d'accès à un objet NIS+ doivent être authentifiés (présenter des données d'identification DES valides) avant de se voir accorder des droits d'accès de propriétaire.

Par défaut, le propriétaire d'un objet est le principal qui l'a créé. Cependant, il peut céder cette propriété à un autre principal des deux manières suivantes :

- Le principal définit un propriétaire différent lors de la création de l'objet (reportez–vous à la section Specifying Access Rights in Commands du guide *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*).
- Le principal modifie la propriété de l'objet après sa création (reportez–vous à la section Changing Ownership of Objects and Entries dans le guide *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*).

En abandonnant la propriété de l'objet, le principal perd tous les droits afférents et ne conserve que les droits attribués par l'objet à la classe Groupe, Monde ou Personne.

Classe Groupe

Le groupe NIS+ d'un objet est *unique* (ici, le terme *groupe* désigne les groupes NIS+, et non des groupes de système d'exploitation ou réseau).

Les principaux qui présentent une requête d'accès à un objet NIS+ doivent être authentifiés (présenter des données d'identification DES valides) et appartenir au groupe avant de se voir accorder les droits d'accès du groupe.

Un groupe NIS+ est constitué de principaux NIS+ réunis de manière à faciliter l'accès à l'espace de nom. Les droits d'accès accordés à un groupe NIS+ s'appliquent à tous les principaux membres de ce groupe. Le propriétaire d'un objet, cependant, n'a pas besoin d'appartenir au groupe de l'objet.

Le créateur d'un objet peut opter pour un groupe par défaut au moment de la création. Un groupe spécifique peut être défini au moment de la création, ou créé ultérieurement.

Les informations relatives aux groupes NIS+ sont stockées dans les **objets** du groupe NIS+, dans le sous–répertoire **groups_dir** de chaque domaine NIS+. Notez que les informations relatives aux groupes NIS+ sont stockées dans la table du groupe NIS+. Elle contient les informations relatives aux groupes système d'exploitation. Des instructions sur l'administration des groupes NIS+ figurent dans la section Administering NIS+ Groups du guide *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*.

Classe Monde

La classe Monde regroupe tous les principaux NIS+ authentifiés par NIS+, c'est–à–dire tous les membres des classes Propriétaire et Groupe, ainsi que tous les principaux présentant des données d'identification DES valides.

Les droits d'accès accordés à la classe Monde s'appliquent donc à tous les principaux authentifiés.

Classe Personne

Cette classe contient tous les principaux, y compris ceux qui ne présentent pas de données d'identification DES valides.

Classes d'autorisation et hiérarchie des objets NIS+

La sécurité NIS+ utilise des classes d'autorisation indépendamment de la hiérarchie des objets. Les objets répertoires représentent le niveau le plus élevé de la hiérarchie par défaut. Viennent ensuite les objets groupe ou table, puis les colonnes et enfin les entrées. Les définitions suivantes apportent des précisions sur chaque niveau :

Niveau répertoire
: Chaque domaine NIS+ contient deux objets répertoires NIS+ : **groups_dir** et **org_dir**. Chaque objet répertoire **groups_dir** contient plusieurs groupes. Chaque objet répertoire **org_dir** contient plusieurs tables.

Niveau groupe ou table
: Les groupes contiennent des entrées, éventuellement d'autres groupes. Les tables contiennent des colonnes et des entrées.

Niveau colonne
: Un tables comporte une ou plusieurs colonnes.

Niveau entrée (ligne)
: Chaque groupe ou tables comporte une ou plusieurs entrées.

Les quatre classes d'autorisation s'appliquent à tous les niveaux. Par conséquent, un objet répertoire dispose d'un propriétaire et d'un groupe. Chaque table d'un objet répertoire dispose de son propre propriétaire et de son propre groupe, qui peuvent être différents du propriétaire et du groupe de l'objet répertoire. A l'intérieur d'une table, une colonne ou une entrée peut avoir son propre propriétaire ou son groupe, qui peuvent aussi être différents du propriétaire et du groupe de la table ou du répertoire.

Droits d’accès NIS+

Les objets NIS+ définissent des droits d’accès pour les principaux NIS+, de la même façon que les fichiers définissent les permissions des utilisateurs dans un système d’exploitation. Les droits d’accès déterminent les opérations que les principaux NIS+ sont autorisés à effectuer sur les objets NIS+ (vous pouvez les consulter à l’aide de la commande **niscat –o**).

Les opérations NIS+ varient selon les différents types d’objets, mais toutes peuvent être rangées dans l’une des quatre catégories de droits d’accès : création, destruction, modification et lecture.

Lecture Un principal qui dispose de ces droits sur un objet peut lire son contenu.

Modification Un principal qui dispose de ces droits sur un objet peut en modifier le contenu.

Destruction Un principal qui dispose de ces droits sur un objet peut le détruire ou le supprimer.

Création Un principal qui dispose de ces droits à un certain niveau d’objet peut créer des objets à l’intérieur de ce niveau. Ainsi, si vous disposez de droits de création dans un objet répertoire NIS+, vous avez la possibilité de créer de nouvelles tables dans ce répertoire. De même, si vous disposez de droits de création dans une table NIS+, vous pouvez créer de nouvelles colonnes ou entrées dans cette table.

Toutes les communications entre les clients et les serveurs NIS+ sont des requêtes pour l’exécution de l’une de ces opérations sur un objet NIS+ spécifique. Par exemple, lorsqu’un principal NIS+ demande l’adresse IP d’un autre poste de travail, il demande en fait un accès en lecture sur l’objet table **hosts**, qui répertorie ce type d’informations. Lorsqu’un principal demande au serveur d’ajouter un répertoire à l’espace de nom NIS+, il demande en fait un accès en **modification** à l’objet parent du répertoire.

Ces droits se répercutent de manière logique vers les niveaux inférieurs, du répertoire à la table, et de la table à la colonne et à l’entrée. Par exemple, pour créer une nouvelle table, vous devez disposer de droits de création dans l’objet répertoire NIS+ dans lequel elle sera stockée. Lorsque vous créez cette table, vous en devenez le propriétaire par défaut. En tant que tel, vous pouvez vous attribuer des droits de création sur cette table, ce qui vous permettra de créer de nouvelles entrées dans celle–ci. Lorsque vous créez de nouvelles entrées dans une table, vous devenez le propriétaire par défaut de ces entrées. En tant que propriétaire de la table, vous pouvez attribuer à d’autres des droits de création au niveau de la table. Par exemple, vous pouvez attribuer à la classe Groupe de votre table des droits de création au niveau table. Dans ce cas, tout membre du groupe de la table peut créer de nouvelles entrées dans la table. Un membre du groupe qui crée une nouvelle entrée dans la table devient par défaut le propriétaire de cette entrée.

Droits d’administrateur et sécurité NIS+

NIS+ n’impose pas que l’administrateur soit unique. Celui qui dispose de droits d’administration sur un objet (c’est–à–dire des droits de création et de destruction et, pour certains objets, des droits de modification) est considéré comme l’administrateur NIS+ de cet objet.

Les droits d’accès initiaux d’un objet NIS+ sont définis par son créateur. Si le créateur limite les droits d’administrateur au propriétaire de l’objet (le créateur, au départ), alors seul le propriétaire dispose de droits d’administrateur sur cet objet. Si le créateur accorde des droits d’administrateur au groupe de l’objet, alors tous les membres de ce groupe disposent de droits d’administrateur sur l’objet.

En théorie, il est possible d’accorder des droits d’administrateur à la classe Monde (world) et même à la classe Personne (nobody). Le logiciel autorise cette procédure. Cependant, le fait d’attribuer des droits d’administration au–delà de la classe Groupe invalide la sécurité NIS+. Par conséquent, si vous attribuez des droits d’administrateur à la classe Monde ou Personne, vous allez à l’encontre du principe même de la sécurité NIS+.

Informations de référence sur la sécurité NIS+

Les commandes suivantes permettent de gérer les mots de passe, les données d'identification et les clés (pour plus d'informations, reportez–vous aux descriptions des commandes concernées) :

chkey	Change une paire de clés RPC sécurisée d'un principal. Si vous ne souhaitez pas refaire le chiffrement de votre clé privée, utilisez plutôt **passwd**. La commande **chkey** n'affecte pas l'entrée du principal, que ce soit dans la table de mots de passe ou dans le fichier **/etc/passwd**.
keylogin	Déchiffre et stocke la clé privée d'un principal dans le démon **keyserv**.
keylogout	Supprime la clé privée stockée sur le **keyserv**.
keyserv	Autorise le serveur à stocker des clés de chiffrement privées.
newkey	Crée une nouvelle paire de clés dans la base de données de clés publiques.
nisaddcred	Crée des données d'identification pour les principaux NIS+.
nisupdkeys	Met à jour les clés publiques dans les objets de répertoire.
passwd	Change et gère le mot de passe de principal.

Chapitre 14. Sécurité NFS (Network File System)

Le Network File System (NFS) est une technologie très répandue qui permet le partage de données entre différents hôtes sur un réseau. Ces informations ont pour objectif de donner les lignes générales pour la sécurisation de NFS. Utilisez ces suggestions en plus des mesures de sécurité s'appliquant à toute l'infrastructure.

A partir de la version AIX 5.3.0, NFS prend également en charge l'utilisation de l'authentification de Kerberos 5 en plus de DES. La sécurité de Kerberos 5 est fournie via un mécanisme de protocole appelé RPCSEC_GSS. Pour plus d'informations sur l'authentification de Kerberos avec NFS, reportez–vous à la section Configuration d'un réseau pour RPCSEC–GSS dans le manuel *AIX 5L Version 5.3 System Management Guide : Communications and Networks*.

Pour des informations générales sur la prise en charge de NFS sur AIX 5L Version 5.1, reportez–vous à la section Chapitre 10. Système de fichiers réseau dans le manuel System Management Guide : Communications and Networks.

Pour des informations générales sur la prise en charge de NFS sur AIX 5L Version 5.2, reportez–vous à la section Système de fichiers réseau et SMBFS dans le manuel System Management Guide : Communications and Networks.

Le service NFS (Network File Service) s'ajoute au système standard d'authentification apporté par UNIX, et offre un moyen d'authentifier les utilisateurs et machines d'un réseau message par message. Ce système d'authentification complémentaire utilise le chiffrement DES (Data Encryption Standard) et le chiffrement par clé publique.

A partir de AIX 5L Version 5.2, NFS prend également en charge l'utilisation de l'authentification de Kerberos 5 en plus de DES. La sécurité de Kerberos 5 est fournie via un mécanisme de protocole appelé RPCSEC_GSS. Pour plus d'informations sur la manière de gérer et utiliser l'authentification Kerberos avec NFS, reportez–vous au manuel *NFS Administration Guide*.

Utilisez les informations suivantes pour vous familiariser avec les divers points concernant NFS :

Consignes générales pour la sécurisation de NFS

Les informations suivantes représentent des consignes d'aide à la sécurisation de NFS :

- Assurez–vous que les dernières mises à jour de logiciels sont installées. Les mises à jour ayant trait aux problèmes de sécurité revêtent une importance toute particulière. Tout logiciel installé dans une infrastructure donnée doit faire l'objet d'une maintenance. Par exemple, installer des mises à jour sur un système d'exploitation sans installer de mises à jour sur un serveur Web peut favoriser les attaques de pirates informatiques à l'encontre de votre environnement. Ce genre d'intrusion peut être évité si le serveur Web fait également l'objet d'une mise à jour.
- Configurez le serveur NFS de sorte qu'une quantité de privilèges minimum soit requise pour l'exportation de systèmes de fichiers. Si les utilisateurs ont seulement besoin d'accéder à un système de fichiers en lecture seule, ils ne sont alors pas en mesure d'écrire sur ce système de fichiers. Cette restriction permet de réduire les risques d'écrasement de données importantes, de modification de fichiers de configuration ou de création de programmes malveillants exécutables sur un système de fichiers exporté. Définissez les privilèges à l'aide de SMIT ou en modifiant directement le fichier **/etc/exports**.
- Configurez le serveur NFS de manière à exporter explicitement les systèmes de fichiers pour les utilisateurs autorisés à y accéder. La plupart des implémentations de NFS vous permettent de définir les clients NFS autorisés à accéder à un système de fichiers donné. Ceci permet de réduire les risques de tentatives d'accès aux systèmes de fichiers par des utilisateurs non autorisés. Veillez tout particulièrement à éviter que le serveur NFS soit configuré pour exporter des systèmes de fichiers vers lui–même.
- Les systèmes de fichiers exportés doivent se trouver dans leurs propres partitions. Un pirate informatique peut endommager le système en écrivant dans un système de fichiers exporté jusqu'à sa saturation. Dans un tel cas, le système de fichiers risque de devenir inaccessible aux autres applications ou aux utilisateurs qui en ont besoin.
- N'autorisez pas les clients NFS à accéder au système de fichiers avec des droits d'utilisateur root ou des droits d'utilisateur inconnus. La plupart des implémentations de NFS peuvent être configurées pour mapper les requêtes d'un utilisateur privilégié ou inconnu vers un utilisateur sans privilèges. Cette configuration permet d'empêcher les pirates informatiques d'accéder aux fichiers et d'exécuter les opérations sur les fichiers en tant qu'utilisateur privilégié.
- N'autorisez pas les clients NFS à exécuter les programmes suid et sgid sur les systèmes de fichiers exportés. Vous empêcherez ainsi toute exécution de programmes malveillants par des utilisateurs privilégiés sur des clients NFS. Si le pirate informatique est capable de créer le fichier exécutable appartenant à un propriétaire ou groupe privilégié, le serveur NFS peut subir des dommages conséquents. Pour ce faire, définissez l'option de commande **mknfsmnt –y**.
- Utilisez NFS sécurisé. Le NFS sécurisé utilise le chiffrement DES pour authentifier les hôtes impliqués dans les transactions RPC. RPC est un protocole utilisé par NFS pour communiquer les requêtes entre les hôtes. Un NFS sécurisé permet de réduire les risques de tentatives des pirates informatiques visant à perturber les requêtes RPC en codant la même horodate dans les requêtes RPC. Lorsqu'un destinataire parvient à déchiffrer l'horodate et confirme que cette dernière est correcte, ceci prouve que la requête RPC provient d'un hôte sécurisé.
- Désactivez NFS s'il n'est pas utilisé. Vous réduirez ainsi le nombre de vecteurs d'attaques possibles dont dispose un pirate.

Pour plus d'informations sur la mise en oeuvre des points traités, reportez–vous aux publications suivantes :

- Installation et configuration de NFS :
  System Management Guide: Communications and Networks
- Fichiers d'exportation pour NFS :
  Files Reference
- NFS sécurisé :
  System Management Guide: Communications and Networks
- Commande mknfsmnt :
  Référence de commandes

Authentification NFS

NFS fait usage de l'algorithme DES à différentes fins. NFS l'utilise pour chiffrer une horodate dans les messages RPC transitant entre les clients et les serveurs NFS. Cette horodate chiffrée permet d'authentifier les machines de la même façon qu'un jeton pour un expéditeur.

Le fait que NFS puisse authentifier tout message RPC échangé entre des clients et des serveurs NFS offre un niveau de sécurité supplémentaire facultatif pour chaque système de fichiers. Par défaut, les systèmes de fichiers sont exportés avec l'authentification UNIX standard. Pour bénéficier de l'option de sécurité renforcée, spécifiez **secure** lorsque vous exportez un système de fichiers.

Chiffrement par clé publique pour NFS sécurisé

Les clés publique et privée sont toutes deux stockées et indexées par leur nom réseau (net name) dans la mappe **publickey.byname.** La clé privée est chiffrée via DES avec le mot de passe de connexion de l'utilisateur. La commande **keylogin** utilise la clé privée chiffrée, la déchiffre avec le mot de passe de connexion et la transmet à un serveur local sécurisé de clés, pour un usage ultérieur dans les transactions RPC. Les utilisateurs ne connaissent ni leur clé publique, ni leur clé privée, car la commande **yppasswd**, outre le fait de modifier le mot de passe de connexion, génère les clés (publique et privée) automatiquement.

Le démon **keyserv** est un service RPC, actif sur chaque machine NIS et NIS+. Pour plus d'informations sur la manière dont NIS+ utilise **keyserv**, reportez–vous au manuel *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*. Dans NIS, **keyserv** exécute les trois sous–routines à clé publique suivantes :

- **key_setsecret**
- **key_encryptsession**
- **key_decryptsession**

key_setsecret indique au serveur de clés de stocker la clé privée de l'utilisateur (SK_A) pour un usage ultérieur. Elle est normalement appelée par la commande **keylogin**. Le programme client appelle **key_encryptsession** pour générer la clé de conversation chiffrée, qui est passée à la première transaction RPC vers un serveur. Le serveur de clés recherche la clé publique du serveur et la combine à la clé privée du client (définie par une sous–routine **key_setsecret** précédente) pour générer la clé commune. Le serveur demande au serveur de clés de déchiffrer la clé de conversation en appelant la sous–routine **key_decryptsession**.

Ces appels de sous–routines supposent un appelant, qui doit lui aussi être authentifié. Pour ce faire, le serveur de clés ne peut pas utiliser l'authentification DES, qui provoquerait un blocage total. Il résout le problème en stockant les clés privées par leur ID utilisateur (UID) et en ne répondant qu'aux demandes des processus root locaux. Le processus client exécute ensuite une sous–routine **setuid** appartenant à l'utilisateur root, qui effectue la demande de la part du client, indiquant au serveur de clés l'UID réel du client.

Règles d’authentification NFS

L’authentification sur NFS sécurisé est basée sur la capacité d’un expéditeur à chiffrer l’heure courante, que le destinataire peut déchiffrer et comparer avec sa propre horloge. Cette méthode comporte les règles suivantes :

- Que les deux agents soient d’accord sur l’heure,
- Que l’émetteur et le destinataire utilisent la même clé de chiffrement DES.

Accord sur l’heure

Si le réseau utilise la synchronisation d’horloge, le démon **timed** assure la synchronisation des horloges client et serveur. Sinon, le démon détermine l’horodate correcte sur la base de l’horloge du serveur. Pour ce faire, il détermine l’heure du serveur avant d’ouvrir la session RPC et calcule le décalage entre son horloge et celle du serveur. Le client règle ensuite son horodate en conséquence. Si, au cours d’une session RPC, les horloges viennent à être désynchronisées au point que le serveur commence à rejeter les demandes client, il appartient au client de réitérer le réglage.

Accord sur la clé DES

Client et serveur déterminent la clé de chiffrement DES à l’aide du chiffrement par clé publique. Pour tout client A et serveur B, une clé appelée *clé commune* peut uniquement être déduite par A et B. Le client déduit la clé commune par la formule :

$$K_{AB} = PK_B \bmod SK_A$$

où *K* est la clé commune, *PK* est la clé publique et *SK* la clé privée ; chacune de ces clés étant un nombre à 128 bits. Le serveur déduit la clé commune à l’aide de la formule :

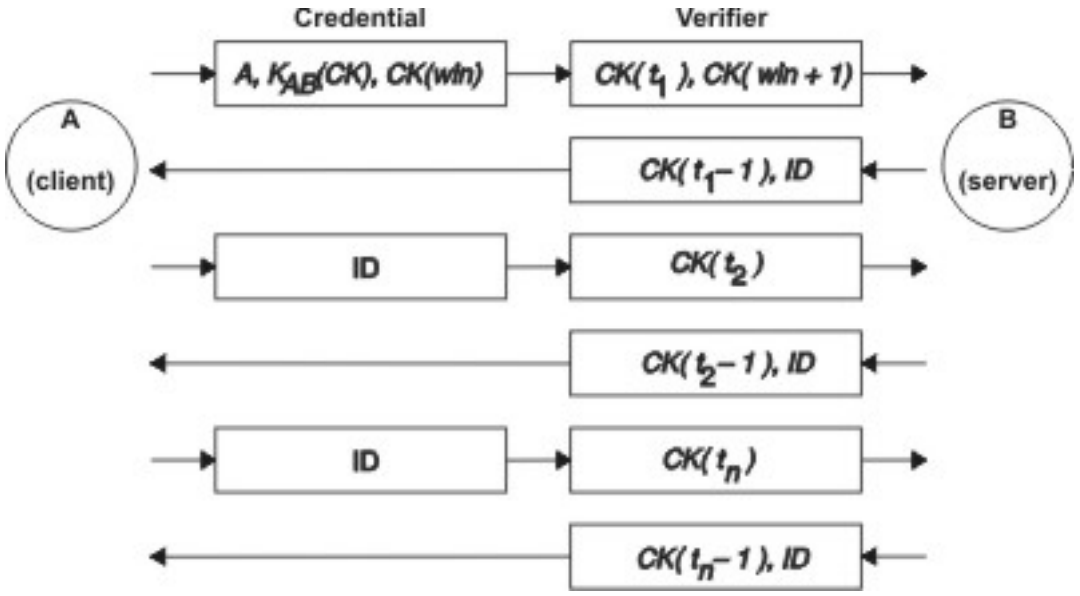
$$K_{AB} = PK_A \bmod SK_B$$

Le calcul de cette clé commune, dans lequel intervient la clé privée de chacun, ne peut être effectué que par le client et le serveur concernés. Cette clé ayant 128 bits et DES utilisant une clé de 56 bits, le client et le serveur extraient 56 bits de la clé commune pour constituer la clé DES.

Process d’authentification NFS

Lorsqu’un client souhaite ”parler” à un serveur, il génère de façon aléatoire une clé, utilisée pour chiffrer les horodates. Cette clé est appelée *clé de conversation* (*CK*). Le client chiffre cette clé via la clé DES commune (reportez–vous à la section Règles d’authentification, page 14-4) et l’envoie au serveur dans la première transaction RPC. La figure suivante illustre ce processus.

Figure 18. Process d’authentification. Cette figure illustre le process d’authentification.



La figure illustre la connexion entre le client A et le serveur B. Le terme *K*(*CK*) signifie que *CK* est chiffrée par la clé DES commune *K*. Dans la première demande, l’identification RPC du client contient son nom (*A*), la clé de conversation (*CK*) et la variable *win* (window) chiffrée via *CK* (dont la valeur par défaut est de 30 minutes). Le vérificateur client dans la première demande contient l’horodate chiffrée et un vérificateur chiffré de la fenêtre spécifiée, *win* + 1. Ce dernier vérificateur complique encore la possibilité de deviner l’identification correcte, la sécurité en est accrue d’autant.

Après authentification du client, le serveur enregistre dans une table les éléments suivants :

- Le nom du client, *A*
- La clé de conversation, *CK*
- La fenêtre,
- L’horodate.

Le serveur n’accepte que les horodates postérieures à la dernière reçue ; ainsi, les transactions répétées sont rejetées. Le serveur renvoie au client dans le vérificateur un ID index dans la table d’authentification, ainsi que l’horodate du client moins 1, chiffrée avec *CK*. Le client sait alors que seul le serveur peut avoir envoyé ce vérificateur, car il est le seul à connaître l’horodate envoyée par le client. On soustrait la valeur 1 à l’horodate pour s’assurer que l’horodate n’est pas valable et ne peut être réutilisée comme vérificateur client. Après la première transaction RPC, le client envoie seulement son ID et une horodate chiffrée au serveur, lequel lui renvoie l’horodate moins 1, chiffrée par *CK*.

Nom des entités réseau pour l'authentification DES

L'authentification DES se base sur les noms réseau (net names). Pour plus d'informations sur la manière dont NIS+ gère l'authentification DES, reportez–vous au manuel *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*.

Un *nom réseau* est une chaîne de caractères imprimables destinés à l'identification. Les clés privées et publiques sont stockées par nom réseau plutôt que par nom d'utilisateur. La mappe **netid.byname** place le nom réseau dans un UID local et une liste d'accès de groupe.

Les noms d'utilisateur sont uniques dans un domaine. Les noms réseau sont formés par concaténation des ID système et utilisateur avec les noms de domaine NIS et Internet. Pour nommer les domaines, optez pour la convention qui consiste à ajouter le nom Internet du domaine (com, edu, gov, mil) à son nom de domaine local.

Les noms réseau sont attribués aux machines et aux utilisateurs. Le nom réseau d'une machine est formé à peu près comme celui d'un utilisateur. Par exemple, un machine nommée `hal` dans le domaine `eng.xyz.com` possède le nom réseau `unix.hal@eng.xyz.com`. Une authentification correcte des machines est essentielle, surtout lorsqu'il s'agit de machines sans disque qui nécessitent un accès total à leur répertoire home dans le réseau.

Pour authentifier les utilisateurs d'un domaine distant, insérez les entrées correspondantes dans deux bases de données NIS : une entrée pour leurs clés publiques et privées, l'autre pour le mappage UID local et liste d'accès groupe. Les utilisateurs du domaine distant peuvent alors accéder à tous les services du réseau local (NFS, connexion à distance, etc.).

Fichier /etc/publickey

Le fichier **/etc/publickey** contient les noms et les clés publiques utilisées par NIS et NIS+ pour créer la mappe **publickey**. La mappe **publickey** assure la sécurité du réseau. Chaque entrée du fichier est constituée du nom d'un utilisateur du réseau (référençant un utilisateur ou un hôte), suivi de la clé publique de l'utilisateur (en hexadécimal), d'un signe deux points et de la clé privée chiffrée de l'utilisateur (également en hexadécimal). Par défaut, l'unique utilisateur inscrit dans le fichier **/etc/publickey** est nobody.

N'utilisez pas un éditeur de texte pour modifier le fichier **/etc/publickey,** car il contient des clés de chiffrement. Pour modifier le fichier **/etc/publickey**, utilisez la commande **chkey** ou la commande **newkey**.

Remarques sur l'amorçage des systèmes à clé publique

Lorsque vous réamorcez une machine après une coupure de courant, toutes les clés privées stockées sont perdues et aucun processus ne peut accéder aux services sécurisés du réseau (tel le montage d'un NFS). Les processus root peuvent se poursuivre, sous réserve que quelqu'un puisse indiquer le mot de passe qui déchiffre la clé privée de l'utilisateur root. La solution est de stocker la clé privée de l'utilisateur root déchiffrée dans un fichier accessible par le serveur de clés.

Tous les appels **setuid** n'aboutissent pas. Par exemple, si la sous–routine **setuid** est appelée par le propriétaire `A`, qui ne s'est pas reconnecté depuis le réamorçage de la machine, elle ne peut accéder aux services réseau sécurisés en tant que `A`. Toutefois, la plupart des appels **setuid** sont la propriété de l'utilisateur root, dont la clé privée est toujours enregistrée au moment de l'amorçage.

Remarques sur les performances de NFS sécurisé

Travailler sous NFS sécurisé n'est pas sans incidence sur les performances du système.

- Le client et le serveur doivent tous les deux calculer la clé commune. Ce calcul demande environ 1 seconde. Autrement dit, il faut environ 2 secondes pour établir la connexion RPC initiale, le client et le serveur devant tous deux effectuer cette opération. Une fois cette connexion établie, le serveur conserve le résultat de l'opération en mémoire cache, ce qui évite de recalculer la clé à chaque fois.
- Chaque transaction RPC nécessite les opérations de chiffrement DES suivantes :
  1. Le client chiffre l'horodate de la demande.
  2. Le serveur la déchiffre.
  3. Le serveur chiffre l'horodate de la réponse.
  4. Le client la déchiffre.

Les performances du système peuvent être diminuées par NFS sécurisé, c'est pourquoi vous devez évaluer les avantages d'une sécurité accrue par rapport aux exigences de performances du système.

Administration de NFS sécurisé

Vérifiez les points suivants pour vous assurer que NFS sécurisé fonctionne correctement :

- Lorsque vous montez un système de fichiers sur un client, en spécifiant **–secure**, le nom du serveur doit correspondre au nom d'hôte du serveur tel qu'il apparaît dans le fichier **/etc/hosts**. Si un serveur de noms sert à la résolution des noms d'hôte, vérifiez que les informations hôte renvoyées par le serveur de noms correspondent à l'entrée du fichier **/etc/hosts**. Faute de quoi, des erreurs d'authentification risquent de se produire, car les noms réseau des machines sont basés sur les entrées principales du fichier **/etc/hosts,** et que c'est le nom réseau qui sert à l'accès aux clés de la mappe **publickey**.
- Ne panachez pas montages et exportations sécurisés et non sécurisés : l'accès aux fichiers risque d'être mal déterminé. Ainsi, si une machine client monte un système de fichiers sécurisé sans option **secure** ou un système non sécurisé avec option **secure,** les utilisateurs y accéderont en tant que `nobody`, et non en tant qu'eux–mêmes. Cette situation se produit également si un utilisateur inconnu de NIS ou NIS+ tente de créer ou de modifier des fichiers d'un système de fichiers sécurisé.
- NIS doit diffuser une nouvelle mappe après chaque utilisation des commandes **chkey** et **newkey**, c'est pourquoi vous ne devez utiliser ces commandes uniquement lorsque le réseau est peu sollicité.
- Ne supprimez ni le fichier **/etc/keystore** ni le fichier **/etc/.rootkey**. Si vous réinstallez, déplacez ou mettez à jour une machine, sauvegardez les fichiers **/etc/keystore** et **/etc/.rootkey**.
- Dites aux utilisateurs d'employer la commande **yppasswd** plutôt que la commande **passwd** pour changer les mots de passe : mots de passe et clés privées resteront ainsi synchronisés.
- La commande **login** ne recherchant pas de clés dans la mappe **publickey** pour le démon **keyserv**, l'utilisateur doit exécuter la commande **keylogin.** Vous pouvez placer la commande **keylogin** dans le fichier **profile** de chaque utilisateur pour qu'elle soit exécutée automatiquement. Notez que la commande **keylogin** exige que l'utilisateur saisisse son mot de passe une deuxième fois.

- Lorsque vous générez les clés de l'utilisateur root au niveau de chaque hôte, via la commande **newkey–h** ou **chkey**, vous devez exécuter la commande **keylogin** pour transmettre les nouvelles clés au démon **keyserv**. Les clés sont stockées dans le fichier **/etc/.rootkey**, lu par le démon **keyserv** chaque fois qu'il est lancé.
- Vérifiez régulièrement que les démons **yppasswdd** et **ypupdated** sont actifs sur le serveur maître NIS. Ces démons sont requis pour maintenir la mappe **publickey**.
- Vérifiez régulièrement que le démon **keyserv** est actif sur toutes les machines utilisant NFS sécurisé.

Configuration de NFS sécurisé

Pour configurer NFS sécurisé sur les serveurs NIS maître et esclaves, passez par l'application Web–based System Manager Network ou procédez comme suit. Pour plus d'informations sur l'utilisation de NFS avec NIS+, reportez–vous au manuel *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*.

1. Sur le serveur NIS maître, créez une entrée pour chaque utilisateur dans le fichier NIS **/etc/publickey** à l'aide de la commande **newkey**.
   - Pour un utilisateur standard, entrez :

     ```
     smit newkey
     ```

     OU

     ```
     newkey –u username
     ```

     Pour un utilisateur root sur une machine hôte, entrez :

     ```
     newkey –h hostname
     ```
   - Les utilisateurs peuvent également définir leurs propres clés publiques via la commande **chkey** ou **newkey**.
2. Créez la mappe NIS **publickey** en suivant les instructions du manuel *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*. La mappe correspondante **publickey.byname** ne doit résider que sur les serveurs.
3. Annulez la mise en commentaire des strophes suivantes dans le fichier **/etc/rc.nfs**:

```
#if [ –x /usr/sbin/keyserv ]; then
 #  startsrc –s keyserv
 #fi
 #if [ –x /usr/lib/netsvc/yp/rpc.ypupdated –a –d /etc/yp/‘domainname‘ ];
then
 #  startsrc –s ypupdated
 #fi
 #DIR=/etc/passwd
 #if [ –x /usr/lib/netsvc/yp/rpc.yppasswdd –a –f $DIR/passwd ]; then
 #  startsrc –s yppasswdd
 #fi
```

4. Lancez les démons **keyserv**, **ypupdated** et **yppasswdd** à l'aide de la commande **startsrc**.

Pour configurer NFS sécurisé sur des clients NIS, lancez le démon **keyserv** à l'aide de la commande **startsrc**.

Exportation d'un système de fichiers via NFS sécurisé

Vous pouvez exporter un NFS sécurisé via l'application réseau Web–based System Manager ou utiliser l'une des procédures suivantes.

- Pour exporter un fichier NFS sécurisé via SMIT :

  1. Vérifiez que NFS est déjà actif en exécutant la commande **lssrc –g nfs**. La sortie indique que les démons **nfsd** et **rpc.mountd** sont actifs.
  2. Vérifiez que la mappe **publickey** existe et que le démon **keyserv** est actif. Pour plus d'informations, reportez–vous à la section Configuration de NFS sécurisé, page 14-8.
  3. Lancez le raccourci **smit mknfsexp.**
  4. Renseignez les zones CHEMIN D'ACCES du répertoire à exporter, MODE d'accès au répertoire exporté et EXPORTER répertoire maintenant, initsyst. ou les deux. Indiquez oui dans le champ Utiliser l'option SECURE.
  5. Spécifiez toute autre caractéristique optionnelle ou acceptez les valeurs par défaut.
  6. Quittez SMIT. Si le fichier **/etc/exports** n'existe pas, il est créé.
  7. Répétez les étapes 3 à 6 pour chaque répertoire à exporter.

- Pour exporter un système de fichiers NFS sécurisé à l'aide d'un éditeur de texte :

  1. Ouvrez le fichier **/etc/exports** avec votre éditeur favori.
  2. Créez une entrée pour chaque répertoire à exporter, en indiquant son chemin d'accès complet. Répertoriez tous les répertoires à exporter en commençant à la marge gauche. Ne spécifiez pas de répertoire qui en contient un autre déjà exporté. Reportez–vous à la documentation de fichiers **/etc/exports** pour obtenir une description de la syntaxe complète des entrées dans le fichier **/etc/exports**, y compris comment spécifier l'option **secure** (sécurisé).
  3. Sauvegardez et fermez le fichier **/etc/exports**.
  4. Si NFS est en cours d'exécution, entrez :

     ```
     /usr/sbin/exportfs –a
     ```

     L'utilisation de l'option **–a** avec la commande **exportfs** envoie toutes les informations contenues dans le fichier **/etc/exports** vers le noyau.

- Pour exporter temporairement un système de fichiers NFS (c'est–à–dire sans modifier le fichier **/etc/exports**), entrez :

```
exportfs –i –o secure /   dirname
```

où `dirname` est le nom du système de fichiers que vous souhaitez exporter. La commande **exportfs –i** spécifie de ne pas rechercher le répertoire dans le fichier **/etc/exports**, et que toutes les options sont directement issues de la ligne de commande.

Montage d’un système de fichiers à l’aide de NFS sécurisé

Procédez comme suit pour monter explicitement un répertoire NFS sécurisé :

1. Vérifiez que le serveur NFS a exporté le répertoire à l’aide de la commande :

```
showmount –e   ServerName
```

   où `ServerName` est le nom du serveur NFS. Cette commande affiche le nom des répertoires exportés du serveur NFS. Si le répertoire à monter ne s’y trouve pas, exportez–le.

2. Définissez le point de montage local à l’aide de la commande **mkdir**. La réussite d’un montage NFS suppose la présence d’un répertoire servant de point de montage. Ce répertoire doit être vide. La création de ce point de montage ne diffère en rien de celle de n’importe quel répertoire, et aucun attribut particulier ne doit être spécifié.

3. Vérifiez que la mappe **publickey** existe et que le démon **keyserv** est actif. Pour plus d’informations, reportez–vous à la section Configuration de NFS sécurisé, page 14-8.

4. Entrez :

```
mount –o secure   ServerName :   /remote/directory    /local/directory
```

   où `ServerName` est le nom du serveur NFS, `/remote/directory` , le répertoire du serveur NFS que vous souhaitez monter et `/local/directory` , le point de montage sur le client NFS.

   Remarque : Seul un utilisateur root peut monter un système de fichiers NFS sécurisé.

Chapitre 15. Enterprise Identity Mapping (EIM)

Les environnements réseau actuels se composent d'un groupe complexe de systèmes et d'applications. Il est donc nécessaire de gérer plusieurs registres d'utilisateurs. La présence de plusieurs registres d'utilisateurs entraîne rapidement un important problème administratif qui affecte les utilisateurs, les administrateurs et les développeurs. EIM permet aux administrateurs et aux développeurs de traiter ce problème simplement.

Ce chapitre décrit les problèmes et les méthodes actuelles, y compris la méthode EIM.

Gestion de plusieurs registres d'utilisateurs

De nombreux administrateurs gèrent des réseaux qui incluent différents systèmes et serveurs, ayant chacun sa propre méthode de gestion des utilisateurs à l'aide de différents registres. Sur ces réseaux complexes, les administrateurs doivent gérer les identités et les mots de passe de chaque utilisateur sur plusieurs systèmes. Ils doivent souvent également synchroniser ces identités et mots de passe. Les utilisateurs doivent se souvenir de leurs nombreux mots de passe et identités et assurer leur synchronisation. Les administrateurs perdent souvent un temps précieux à résoudre les échecs de connexions et à redéfinir les mots de passe oubliés.

Le problème que constitue la gestion de plusieurs registres d'utilisateurs affecte aussi les développeurs d'applications hétérogènes ou en plusieurs couches. D'importantes données d'entreprise sont réparties sur différents types de systèmes, qui ont chacun leurs propres registres d'utilisateurs. Les développeurs doivent donc créer des registres d'utilisateurs propriétaires et associer des codes de sécurité à leurs applications. Ils résolvent ainsi leur problème, mais augmentent la charge de travail des utilisateurs et administrateurs.

Méthodes actuelles

Il existe plusieurs méthodes pour résoudre les problèmes inhérents à la gestion de plusieurs registres d'utilisateurs, mais aucune ne fournit une solution complète. Par exemple, le protocole LDAP fournit une solution de registres d'utilisateurs répartis. Cependant, pour utiliser de telles solutions, les administrateurs doivent gérer un registre d'utilisateurs et un code de sécurité supplémentaires, ou remplacer des applications conçues pour les autres registres.

Ils doivent alors gérer plusieurs systèmes de sécurité pour des ressources individuelles, augmentant ainsi leur charge de travail, ainsi que les failles potentielles au niveau de la sécurité. Lorsque plusieurs systèmes supportent une même ressource, il est fréquent de modifier une autorité pour un système et d'oublier de la modifier pour les autres systèmes. Par exemple, la sécurité peut être contournée lorsqu'un utilisateur se voit refuser l'accès de manière appropriée par une interface, mais qu'il bénéficie de l'accès via d'autres interfaces.

Une fois leur travail terminé, les administrateurs comprennent qu'ils n'ont pas complètement réglé le problème. Généralement, les entreprises ont trop investi dans leurs registres d'utilisateurs actuels et dans les codes de sécurité associés pour que l'utilisation de ce type de solution soit pratique. La création d'un autre registre d'utilisateurs et du code de sécurité associé règle le problème du développeur, mais pas celui des utilisateurs ni des administrateurs.

Une autre solution consiste à utiliser une méthode unique de connexion. Plusieurs produits permettent aux administrateurs de gérer des fichiers contenant tous les mots de passe et identités des utilisateurs. Cependant, cette méthode comporte plusieurs inconvénients :

- Elle ne règle que l'un des problèmes des utilisateurs. Elle leur permet de se connecter à plusieurs systèmes en fournissant une identité et un mot de passe, mais ils doivent toujours gérer leurs mots de passe et les utiliser sur d'autres systèmes.
- Elle créé un nouveau problème car des mots de passe déchiffrables ou non chiffrés sont stockés dans ces fichiers. Les mots de passe ne doivent jamais être stockés dans des fichiers non chiffrés ou accessibles facilement par d'autres personnes, y compris les administrateurs.
- Elle ne résout pas les problèmes des développeurs d'applications tierces hétérogènes ou à plusieurs couches. Les développeurs doivent toujours donc créer des registres d'utilisateurs propriétaires pour leurs applications.

Malgré ces inconvénients, des entreprises utilisent ces méthodes car elle simplifient en partie la gestion de plusieurs registres d'utilisateurs.

Utilisation de l'EIM

L'architecture EIM décrit les relations entre individus ou entités (tels que des serveurs de fichiers et d'impressions) d'une entreprise, et leurs nombreuses identités au sein de l'entreprise. EIM fournit un ensemble d'API permettant aux applications de poser des questions sur ces relations.

Par exemple, connaissant l'identité utilisateur d'une personne dans un registre d'utilisateurs, vous pouvez connaître son identité dans un autre registre. Si l'utilisateur s'est authentifié avec une identité et que vous pouvez mapper cette identité avec l'identité correspondante d'un autre registre, l'utilisateur n'a pas besoin de fournir à nouveau de données d'identification. Il suffit de connaître l'identité correspondant à cet utilisateur dans un autre registre. EIM fournit un fonction généralisée de mappage d'identité dans l'entreprise.

La possibilité de trouver la correspondance entre les identités d'un utilisateur dans différents registres comporte plusieurs avantages. Les applications peuvent utiliser un registre pour l'authentification et un autre registre pour les autorisations. Par exemple, un administrateur peut mapper une identité SAP pour accéder aux ressources SAP.

Le mappage d'identités nécessite que les administrateurs exécutent la procédure suivante :

1. Création d'identifiants EIM représentant les personnes ou entités dans l'entreprise.
2. Création de définitions de registres EIM décrivant les registres d'utilisateurs de l'entreprise.
3. Définition des relations entre les identités des utilisateurs dans ces registres et les identifiants EIM créés.

Il est inutile de modifier les codes des registres existants. Il n'est pas nécessaire de procéder au mappage de toutes les identités d'un registre d'utilisateurs. EIM permet des mappages one–to–many (c'est à dire un utilisateur qui a plusieurs identités dans un même registre). EIM permet aussi des mappages many–to–one (c'est à dire plusieurs utilisateurs partageant la même identité dans un même registre), bien qu'ils ne soient pas recommandés pour des raisons de sécurité. Un administrateur peut représenter tout type de registre d'utilisateurs dans EIM.

EIM ne nécessite pas de copier des données dans un nouveau répertoire et d'essayer d'assurer la synchronisation des deux exemplaires. Les seules nouvelles données inhérentes à EIM sont les informations sur les relations. Les administrateurs les placent dans un répertoire LDAP, pouvant ainsi les gérer à un emplacement et disposer de copies là où ces informations sont utiles.

Chapitre 16. Kerberos

Kerberos est un service d'authentification réseau qui permet de vérifier les identités des principaux sur des réseaux physiquement non sécurisés. Kerberos permet l'authentification mutuelle, l'intégrité et la confidentialité des données, en tenant compte des risques non négligeables d'interception, de contrôle et de substitution dont peut faire l'objet le trafic sur le réseau.

Les tickets Kerberos sont des données d'identification permettant de contrôler votre identité. On distingue deux types de tickets : le *ticket d'émission de tickets* et le *ticket de service*. Le ticket d'émission de tickets est utilisé pour votre requête d'identité initiale. Lorsque vous vous connectez à un système hôte, votre identité doit être vérifiée, par exemple par un mot de passe ou un jeton. Une fois que vous possédez votre ticket d'émission de tickets, vous pouvez l'utiliser pour créer des requêtes de tickets pour des services spécifiques. La méthode à deux tickets est appelée *entité tierce sécurisée* de Kerberos. Votre ticket d'émissions de ticket permet de vous authentifier auprès du serveur Kerberos et votre ticket de service permet un accès sécurisé au service.

L'entité tierce sécurisée ou intermédiaire dans Kerberos s'appelle le *Centre de distribution de clés* (KDC). Le serveur KDC émet tous les tickets Kerberos aux clients.

La base de données Kerberos conserve un enregistrement pour chaque principal ; cet enregistrement contient le nom, la clé privée, la date d'expiration du principal et d'autres informations administratives sur chaque principal. Le KDC maître contient la copie maître de la base de données et la transmet aux KDC esclaves.

Ce chapitre comporte les informations suivantes concernant Kerberos :

Présentation des commandes à distance sécurisées

Remarques :

1. A partir de la version 2.2 DCE (Distributed Computing Environment), le serveur de sécurité DCE peut renvoyer des tickets Kerberos Version 5.
2. A partir de AIX 5.2, toutes les commandes distantes sécurisées (rcmds) utilisent la bibliothèque Kerberos Version 5 fournie par le service d'authentification réseau (Network Authentication Service, NAS) version 1.3. Dans un domaine DCE, la commande **ftp** utilise la bibliothèque GSSAPI à partir de la bibliothèque DCE **libdce.a** et, dans un domaine natif, la commande **ftp** utilise la bibliothèque GSSAPI à partir de NAS version 1.3. NAS version 1.3 se trouve sur le CD Expansion Pack. Le seul programme sous licence (LPP) requis est l'ensemble de fichiers **krb5.client.rte**.
3. Si vous effectuez une migration vers AIX 5.2 et que Kerberos (version 4 ou 5) est installé, les scripts d'installation invitent l'utilisateur à installer **krb5.client.rte**.

Les commandes rcmds sécurisées sont **rlogin**, **rcp**, **rsh**, **telnet** et **ftp**. Ces commandes sont communément appelées méthode *AIX Standard*. (Cette méthode fait référence à la méthode d'authentification utilisée par AIX 4.3 et les versions précédentes.) Les méthodes supplémentaires fournies sont Kerberos, versions 4 et 5.

Lors de l'utilisation de la méthode d'authentification Kerberos Version 5, le client obtient un ticket Kerberos Version 5 provenant du serveur de sécurité DCE ou du serveur Kerberos. Le ticket est une portion du DCE actuel de l'utilisateur ou de ses données d'identification locales, chiffrées pour le serveur TCP/IP avec lequel il souhaite établir une connexion. Le démon du serveur TCP/ IP déchiffre le ticket. Ceci permet au serveur TCP/IP d'identifier l'utilisateur. Si le principal DCE ou local décrit dans le ticket est autorisé à accéder au compte de l'utilisateur du système d'exploitation, la connexion se poursuit. Les commandes rcmds sécurisées prennent en charge les clients et serveurs Kerberos depuis Kerberos Version 5 et DCE.

Outre l'authentification du client, Kerberos Version 5 transfère les données d'identification de l'utilisateur actuel au serveur TCP/IP. Si les données d'identification sont marquées comme transférables, le client les envoie au serveur comme un TGT Kerberos (ticket d'octroi d'autorisations). Côté serveur TCP/IP, si un utilisateur communique avec le serveur de sécurité DCE, le démon met à niveau le TGT dans les données d'identification DCE complètes à l'aide de la commande **k5dcecreds**.

La commande **ftp** utilise une méthode d'authentification différente des autres commandes rcmds sécurisées. Elle utilise le mécanisme de sécurité GSSAPI pour transmettre l'authentification entre la commande **ftp** et le démon **ftpd**. A l'aide des sous–commandes **clear**, **safe** et **private**, le client **ftp** prend en charge le chiffrement des données.

La commande **ftp** permet des transferts de plusieurs octets pour les connexions de données chiffrées entre les clients et les serveurs du système d'exploitation. Les normes définissent uniquement les transferts mono–octets pour les connexions de données chiffrées. Pour les connexions à des machines tierces utilisant le chiffrement de données, la commande **ftp** suit la limite de transfert mono–octet.

Configuration de système

Pour l'ensemble des commandes rcmds sécurisées, un mécanisme de configuration au niveau du système détermine quelles méthodes d'authentification sont autorisées pour ce système. La configuration commande à la fois les connexions entrantes et sortantes.

La configuration de l'authentification comprend la bibliothèque **libauthm.a**, de même que les commandes **lsauthent** et **chauthent**, qui fournissent un accès en ligne de commande aux routines de bibliothèque **get_auth_methods** et **set_auth_methods**.

La méthode d'authentification définit quelle méthode est utilisée pour authentifier un utilisateur sur un réseau. Le système prend en charge les méthodes d'authentification suivantes :

- Kerberos Version 5 est la méthode la plus répandue et constitue la base pour DCE.
- Kerberos Version 4 est utilisé exclusivement par les commandes rcmds sécurisées **rlogin**, **rsh** et **rcp**. Kerberos Version 4 prend en charge la rétrocompatibilité sur les systèmes SP uniquement. Les tickets Kerberos Version 4 ne sont pas mis à niveau vers les données d'identification DCE.
- AIX Standard est la méthode d'authentification utilisée par AIX 4.3 et versions précédentes.

Si plusieurs méthodes d'authentification ont été configurées et que la première échoue lors de la connexion, le client demande l'authentification à l'aide de la méthode d'authentification configurée suivante.

Les méthodes d'authentification peuvent être configurées dans n'importe quel ordre. La seule exception concerne AIX Standard, qui doit être la méthode d'authentification configurée en dernier, puisqu'il n'existe aucune option de procédure de secours. Si AIX Standard n'est pas une méthode d'authentification configurée, aucune authentification par mot de passe n'est tentée et toute tentative de connexion utilisant cette méthode est rejetée.

Vous pouvez configurer le système sans recourir à la moindre méthode d'authentification. Dans ce cas, la machine refuse toute connexion de et vers toute machine exploitant des commandes rcmds sécurisées. De plus, Kerberos Version 4 n'étant pris en charge que par les commandes **rlogin**, **rsh** et **rcp**, un système configuré pour utiliser uniquement Kerberos Version 4 n'autorise pas les connexions via **telnet** ou **ftp**.

Validation utilisateur Kerberos Version 5

Lors de l'utilisation de la méthode d'authentification Kerberos Version 5, le client TCP/IP obtient un ticket de service chiffré pour le serveur TCP/IP. Lorsque le serveur déchiffre le ticket, il dispose d'une méthode sécurisée d'identification de l'utilisateur (par principal local ou DCE). Cependant, le serveur doit toujours déterminer si ce principal local ou DCE est autorisé à accéder au compte local. La mise en correspondance entre le principal local ou DCE et le compte du système d'exploitation local est traitée par une bibliothèque partagée, **libvaliduser.a**, qui possède une sous–routine unique nommée kvalid_user. Si vous préférez une méthode de mise en correspondance différente, l'administrateur système doit fournir une autre bibliothèque que **libvaliduser.a**.

Configuration DCE

Pour utiliser les commandes rcmds sécurisées, deux principaux DCE doivent exister pour chaque interface réseau à laquelle ils peuvent se connecter. Ce sont :

```
host/   FullInterfaceName
 ftp/   FullInterfaceName
```

où :

FullInterfaceName
: Nom d'interface et nom de domaine

Configuration locale

Pour utiliser les commandes rcmds sécurisées, deux principaux locaux doivent exister pour chaque interface réseau à laquelle ils peuvent se connecter. Ce sont :

```
host/   FullInterfaceName@Realmname
 ftp/   FullInterfaceName@Realmname
```

où :

FullInterfaceName
: Nom d'interface et nom de domaine

RealmName
: Nom du domaine local Kerberos Version 5

Informations connexes

- Les sous–routines get_auth_method et set_auth_method dans le manuel *AIX 5L Version 5.3 Technical Reference: Communications Volume 2*
- La commande chauthent dans le manuel *AIX 5L Version 5.3 Commands Reference, Volume 1*
- La commande lsauthent dans le manuel *AIX 5L Version 5.3 Commands Reference, Volume 3*

Authentification sous AIX à l'aide de Kerberos

AIX fournit les modules d'authentification Kerberos **KRB5** et **KRB5A**. Même si les deux modules prennent en charge l'authentification Kerberos, le module de chargement **KRB5** effectue la gestion Kerberos des principaux, contrairement au module de chargement **KRB5A**. Le module de chargement **KRB5** utilise l'interface de base de données Service d'authentification réseau NAS (Network Authentication Services) Kerberos pour gérer les identités et principaux Kerberos. A l'aide d'un module de chargement **KRB5**, un administrateur système SWsym.AIX peut gérer les utilisateurs authentifiés Kerberos et leurs principaux Kerberos associés en utilisant les commandes d'administration utilisateur existantes SWsym.AIX, sans procéder au moindre changement. Par exemple, pour créer un utilisateur SWsym.AIX, de même qu'un principal Kerberos associé à cet utilisateur, exécutez la commande **mkuser**.

Le module de chargement **KRB5A** effectue uniquement l'authentification. La gestion Kerberos des principaux est effectuée séparément via les outils Kerberos de gestion des principaux. Le module de chargement **KRB5A** est utilisé dans un environnement où les principaux Kerberos, étant stockés sur un système autre que SWsym.AIX, ne peuvent être gérés depuis AIX via l'interface de base de données Kerberos. **KRB5A** est destiné à être utilisé avec un serveur Active Directory Windows 2000 où la gestion des principaux Kerberos est effectuée à l'aide des outils et des interfaces API de gestion de compte Active Directory.

Installation et configuration du système pour la connexion intégrée Kerberos à l'aide de KRB5

Network Authentication Services (implémentation Kerberos) est disponible sur l'Expansion Pack. Pour installer le module client Kerberos Version 5, installez l'ensemble de fichiers **krb5.client.rte**. Pour installer le module serveur Kerberos Version 5, installez l'ensemble de fichiers **krb5.server.rte**. Pour installer l'ensemble du module Kerberos Version 5, installez le module **krb5**.

Pour éviter les collisions d'espace de nom entre les commandes DCE et Kerberos (c'est à dire, entre les commandes **klist**, **kinit**et **kdestroy**), les commandes Kerberos sont installées dans les répertoires **/usr/krb5/bin** et **/usr/krb5/sbin**. Vous pouvez ajouter ces répertoires à votre définition **PATH**. Sinon, pour exécuter les commandes Kerberos, vous devez préciser les noms complets des chemins des commandes.

Une documentation Network Authentication Services (NAS) est fournie dans le module **krb5.doc.** *lang***.pdf|html**, *lang* désignant la langue prise en charge.

Configuration des serveurs Kerberos Version 5 KDC et kadmin

Remarques :

1. L'installation des logiciels serveur Kerberos et DCE sur le même système physique n'est pas recommandée. Si vous devez procéder ainsi, les numéros de ports Internet opérationnels par défaut doivent être changés soit en clients/serveur DCE, soit en clients/serveur Kerberos. Dans les deux cas, ce changement peut affecter l'interopérabilité avec les déploiements de DCE et Kerberos dans votre environnement. Pour plus d'informations sur la coexistence entre DCE et Kerberos, reportez–vous à la documentation Network Authentication Services (NAS).

2. Kerberos Version 5 est configuré pour refuser les demandes de tickets provenant de tout hôte dont l'horloge ne se situe pas dans le décalage d'horloge maximum indiqué du KDC. La valeur par défaut pour le décalage d'horloge est de 300 secondes (cinq minutes). Kerberos requiert une certaine synchronisation temporelle entre les serveurs et les clients. Il est recommandé d'utiliser **xntpd** ou **timed** pour la synchronisation temporelle. Procédez comme suit pour utiliser le démon **timed** :

   a. Configurez le serveur KDC comme un serveur temporel en démarrant le démon **timed**, comme suit :

      ```
      timed -M
      ```

   b. Démarrez le démon **timed** sur chaque client Kerberos.

      ```
      timed -t
      ```

      Pour configurer les serveurs Kerberos KDC et **kadmin**, exécutez la commande **mkkrb5srv**. Par exemple, pour configurer Kerberos pour le domaine `MYREALM`, le serveur `sundial` et le domaine `xyz.com`, entrez ce qui suit :

      ```
      mkkrb5srv -r MYREALM -s sundial.xyz.com -d xyz.com -a admin/admin
      ```

      Patientez quelques minutes pour le démarrage des commandes **kadmind** et **krb5kdc** depuis **/etc/inittab**.

L'exécution de la commande **mkkrb5srv** entraîne les actions suivantes :

1. Génère le **/etc/krb5/krb5.conf**. Les valeurs pour l'ID de domaine, le serveur admin Kerberos et le nom de domaine sont définies comme spécifiées sur la ligne de commande. Le fichier **/etc/krb5/krb5.conf** définit aussi les chemins pour les fichiers journaux **default_keytab_name**, **kdc** et **admin_server**.
2. Crée le fichier **/var/krb5/krb5kdc/kdc.conf**. Le fichier **/var/krb5/krb5kdc/kdc.conf** définit les valeurs pour les variables **kdc_ports**, **kadmin_port**, **max_life**, **max_renewable_life**, **master_key_type** et **supported_enctypes**. Ce fichier définit également les chemins pour les variables **database_name**, **admin_keytab**, **acl_file**, **dict_file** et **key_stash_file**.
3. Crée le fichier **/var/krb5/krb5kdc/kadm5.acl**. Configure le contrôle d'accès pour les principaux **admin**, **root** et **host**.
4. Crée la base de données et un principal **admin**. On vous demande de définir une clé principale Kerberos et de nommer et de définir le mot de passe pour une identité de principal administratif Kerberos. Afin de pouvoir récupérer les données en cas de problème grave, la clé principale ainsi que l'identité et le mot de passe du principal administratif doivent absolument être stockés de manière sécurisée.

Pour plus d'informations, reportez–vous aux sections Exemples d'exécution, page 16-8 et Messages d'erreurs et actions pour la reprise, page 16-7.

Configuration des clients Kerberos Version 5

Une fois l'installation Kerberos terminée, le fait que la technologie Kerberos soit en cours d'utilisation n'apparaît pas de manière évidente aux utilisateurs normaux. Le processus de connexion au système d'exploitation demeure inchangé. Cependant, les utilisateurs disposent désormais des tickets d'octroi d'autorisations (TGT) associés avec leur processus d'exécution. Pour configurer des systèmes pour utiliser Kerberos comme moyen principal d'authentification utilisateur, exécutez la commande **mkkrb5clnt** à l'aide des paramètres suivants:

```
mkkrb5clnt -c KDC -r realm -a admin -s server -d domain
-A -i database -K -T
```

Par exemple, pour configurer le KDC `sundial.xyz.com` avec le domaine `MYREALM`, le serveur admin `sundial.xyz.com`, le domaine `xyz.com` et la base de données `files`, entrez ce qui suit :

```
mkkrb5clnt -c sundial.xyz.com -r MYREALM -s sundial.xyz.com -d xyz.com -A
-i files -K -T
```

L'exemple précédent entraîne les actions suivantes :

1. Génère le **/etc/krb5/krb5.conf**. Les valeurs pour l'ID de domaine, le serveur admin Kerberos et le nom de domaine sont définies comme spécifiées sur la ligne de commande. Met également à jour les chemins pour les fichiers journaux **default_keytab_name**, **kdc** et **kadmin**.
2. L'indicateur **–i** configure une connexion complètement intégrée. La base de données indiquée constitue l'emplacement où sont stockées les informations d'identification des utilisateurs SWsym.AIX. Cet emplacement est différent de l'emplacement de stockage des principaux Kerberos. L'emplacement de stockage des principaux Kerberos est défini lors de la configuration de Kerberos.
3. L'indicateur **–K** permet de configurer Kerberos comme processus d'authentification par défaut. Ceci permet aux utilisateurs d'être identifiés avec Kerberos au moment de la connexion.
4. L'indicateur **–A** ajoute une entrée dans la base de données Kerberos pour faire d'un utilisateur admin pour Kerberos un utilisateur root.
5. L'indicateur **–T** permet d'acquérir le ticket admin basé sur TGT de l'administrateur serveur.

Si un système installé se trouve dans un domaine DNS différent du KDC, vous devez effectuer les actions supplémentaires suivantes :

1. Modifiez le fichier **/etc/krb5/krb5.conf** et ajoutez une nouvelle entrée après **[domain realm]**.
2. Etablissez une correspondance entre les différents domaines.

Par exemple, si vous souhaitez inclure un client qui se trouve dans le domaine `abc.xyz.com` dans votre domaine `MYREALM`, le fichier **/etc/krb5/krb5.conf** contient l'entrée supplémentaire suivante :

```
[domain realm]
      .abc.xyz.com = MYREALM
```

Messages d'erreurs et actions pour la reprise

Les erreurs qui peuvent survenir en utilisant la commande **mkkrb5srv** sont les suivantes :

- Si les fichiers **krb5.conf**, **kdc.conf** ou **kadm5.acl** existent déjà, la commande **mkkrb5srv** ne modifie pas les valeurs. Vous recevez un message indiquant que le fichier existe déjà. Toutes les valeurs de configuration peuvent être modifiées en éditant les fichiers **krb5.conf**, **kdc.conf** ou **kadm5.acl**.
- Si votre saisie est incorrecte et qu'aucune base de données n'est créée, supprimez les fichiers de configuration créés et ré–exécutez la commande.
- En cas d'incohérence entre la base de données et les valeurs de configuration, supprimez la base de données contenue dans le répertoire **/var/krb5/krb5kdc/***et ré–exécutez la commande.
- Vérifiez que les démons **kadmind** et **krb5kdc** sont démarrés sur votre ordinateur. Utilisez la commande **ps** pour vérifier que démons sont exécutés. Si ces démons n'ont pas été lancés, vérifiez le fichier journal.

Les erreurs qui peuvent survenir en utilisant la commande **mkkrb5clnt** sont les suivantes :

- Les valeurs incorrectes pour **krb5.conf** peuvent être corrigées en modifiant le fichier **/etc/krb5/krb5.conf**.
- Les valeurs incorrectes pour l'indicateur **krb5.conf** peuvent être corrigées en modifiant le fichier **/usr/lib/security/methods.cfg**.

Fichiers créés

La commande **mkkrb5srv** crée les fichiers suivants :

- **/etc/krb5/krb5.conf**
- **/var/krb5/krb5kdc/kadm5.acl**
- **/var/krb5/krb5kdc/kdc.conf**

La commande **mkkrb5clnt** crée le fichier suivant :

- **/etc/krb5/krb5.conf**

L'option **mkkrb5clnt –i** *fichiers* ajoute la strophe suivante au fichier **/usr/lib/security/methods.cfg** :

```
KRB5:
                program =
                options =
 KRB5files:
                options =
```

Exemple d'exécutions

Ce qui suit constitue un exemple de la commande **mkkrb5srv** :

```
# mkkrb5srv –r MYREALM –s sundial.xyz.com –d xyz.com –a admin/admin
```

Le résultat affiché sera semblable au suivant :

```
  Fileset                      Level  State      Description
----------------------------------------------------------------------------
Path: /usr/lib/objrepos
   krb5.server.rte           1.3.0.0  COMMITTED  Network Authentication Service
                                                     Server
Path: /etc/objrepos
   krb5.server.rte           1.3.0.0  COMMITTED  Network Authentication Service
                                                     Server

    The –s option is not supported.
    The administration server will be the local host.
    Initializing configuration...
    Creating /etc/krb5/krb5.conf...
    Creating /var/krb5/krb5kdc/kdc.conf...
    Creating database files...
    Initializing database '/var/krb5/krb5kdc/principal' for realm 'MYREALM'
    master key name 'K/M@MYREALM'
    You will be prompted for the database Master Password.
    It is important that you NOT FORGET this password.
    Enter database Master Password:
    Re-enter database Master Password to verify:
    WARNING: no policy specified for admin/admin@MYREALM;
       defaulting to no policy. Note that policy may be overridden by
       ACL restrictions.
    Enter password for principal "admin/admin@MYREALM":
    Re-enter password for principal "admin/admin@MYREALM":
    Principal "admin/admin@MYREALM" created.
    Creating keytable...
    Creating /var/krb5/krb5kdc/kadm5.acl...
    Starting krb5kdc...
    krb5kdc was started successfully.
    Starting kadmind...
    kadmind was started successfully.
    The command completed successfully.
    Restarting kadmind and krb5kdc
```

Ce qui suit est un exemple de la commande **mkkrb5clnt** :

```
mkkrb5clnt –r MYREALM –c sundial.xyz.com –s sundial.xyz.com \
            –a admin/admin –d xyz.com –i files –K –T –A
```

Le résultat affiché sera semblable au suivant :

```
Initializing configuration...
Creating /etc/krb5/krb5.conf...
The command completed successfully.
Password for admin/admin@MYREALM:
Configuring fully integrated login
Authenticating as principal admin/admin with existing credentials.
WARNING: no policy specified for host/diana.xyz.com@MYREALM;
   defaulting to no policy. Note that policy may be overridden by
   ACL restrictions.
Principal "host/diana.xyz.com@MYREALM" created.

Administration credentials NOT DESTROYED.
Authenticating as principal admin/admin with existing credentials.

Administration credentials NOT DESTROYED.
Authenticating as principal admin/admin with existing credentials.
Principal "kadmin/admin@MYREALM" modified.

Administration credentials NOT DESTROYED.
Configuring Kerberos as the default authentication scheme
Making root a Kerberos administrator
Authenticating as principal admin/admin with existing credentials.
WARNING: no policy specified for root/diana.xyz.com@MYREALM;
   defaulting to no policy. Note that policy may be overridden by
   ACL restrictions.
Enter password for principal "root/diana.xyz.com@MYREALM":
Re-enter password for principal "root/diana.xyz.com@MYREALM":
Principal "root/diana.xyz.com@MYREALM" created.

Administration credentials NOT DESTROYED.
Cleaning administrator credentials and exiting.
```

Elimination de la dépendance envers le démon kadmind lors de l'authentification

L'authentification à l'aide du module de chargement KRB5 échoue si le démon **kadmind** n'est pas disponible. Cette dépendance envers le démon **kadmind** durant l'authentification peut être éliminée en configurant le paramètre d'options kadmind dans le fichier **methods.cfg**. Les valeurs possibles sont `no` ou `false` pour la désactivation des recherches **kadmind** et `yes` ou `true` pour l'activation des recherches **kadmind** (la valeur par défaut est `yes`). Si la valeur de cette option est `no`, le démon **kadmind** n'est pas contacté lors de l'authentification. Ainsi, les utilisateurs peuvent se connecter au système quel que soit l'état du démon **kadmind** (à condition que l'utilisateur entre le mot de passe approprié à l'invite du système). Cependant, les commandes d'administration des utilisateurs AIX telles que **mkuser**, **chuser** ou **rmuser** ne fonctionneront pas pour administrer les utilisateurs intégrés Kerberos si le démon n'est pas disponible (par exemple, lorsque le démon ne fonctionne pas ou lorsque la machine n'est pas accessible).

La valeur par défaut du paramètre d'option kadmind est `yes`. Cela signifie que les recherches **kadmind** seront effectuées durant l'authentification. Avec le paramètre par défaut, lorsque le démon n'est pas disponible, le délai d'authentification peut être plus long.

Pour désactiver le contrôle du démon **kadmind** durant l'authentification, modifiez les strophes du fichier **methods.cfg** de la manière suivante :

```
KRB5:
         program = /usr/lib/security/KRB5
         options = kadmind=no


KRB5files:
         options = db=BUILTIN,auth=KRB5
```

Lorsque le démon **kadmind** n'est pas disponible, l'utilisateur root ne peut pas modifier les mots de passe des utilisateurs. Si un utilisateur oublie son mot de passe, cette situation pourra être résolue si le démon **kadmind** est disponible. De même, si un utilisateur saisit le nom d'un principal Kerberos à l'invite de connexion, ce nom principal sera tronqué en fonction de la limite des noms d'utilisateur définie sous AIX. Ce nom tronqué sera utilisé pour la récupération des données d'identification de l'utilisateur AIX (par exemple, votre répertoire personnel).

Si le démon **kadmind** n'est pas disponible (ne fonctionne pas ou est inaccessible), la commande **mkuser** renvoie l'erreur suivante :

```
3004-694 Error adding "krb5user": You do not have permission.
```

De plus, les commandes **chuser** et **lsuser** gèrent uniquement les attributs liés à AIX et non ceux liés à Kerberos. La commande **rmuser** ne supprime pas le principal Kerberos et la commande **passwd** échoue pour les utilisateurs authentifiés Kerberos.

Si le réseau sur lequel se trouve le démon **kadmind** n'est pas accessible, le temps de réponse peut être plus long. En attribuant la valeur `no` à l'option kadmind dans le fichier **methods.cfg**, vous éliminez le délai d'authentification lorsque la machine n'est pas accessible.

Lorsque le démon **kadmind** ne fonctionne pas, les utilisateurs dont le mot de passe a expiré ne peuvent pas se connecter. Les mots de passe arrivés à expiration doivent être changés. Le changement de mot de passe requiert la disponibilité du démon **kadmind**. Par conséquent, les utilisateurs qui nécessitent un changement de mot de passe ou dont le mot de passe a expiré ne pourront pas se connecter lorsque le démon **kadmind** ne sera pas disponible.

Si la valeur de l'option kadmind est `no`, lorsque le démon **kadmind** fonctionne, **lsuser** ne pourra pas récupérer les attributs Kerberos. Cependant, les commandes **login**, **su**, **passwd**, **mkuser**, **chuser** et **rmuser** pourront être exécutées avec succès.

Installation et configuration du système pour la connexion intégrée Kerberos à l'aide de KRB5A

Lorsque le module de chargement **KRB5A** est utilisé pour l'authentification, vous devez effectuer une série d'opérations, comme la création de principaux Kerberos.

La section suivante explique comment authentifier un client Network Authentication Service SWsym.AIX à l'aide d'un KDC Active Directory.

Installez l'ensemble de fichiers **krb5.client.rte** depuis l'Extension Pack.

Remarque : Le module de chargement d'authentification KRB5A est pris en charge uniquement sous AIX 5.2 et versions ultérieures.

Configuration de clients AIX Kerberos Version 5 avec un serveur Active Directory Windows 2000

Utilisez la commande **config.krb5** pour configurer un client Kerberos AIX. Des informations sur le serveur Kerberos sont nécessaires pour la configuration du client. Si un serveur Active Directory Windows 2000 est sélectionné comme serveur Kerberos, les options suivantes peuvent être utilisées avec la commande **config.krb5** :

```
-r realm = Windows 2000 Active Directory server domain name
-d domain = Domain name of the machine hosting the Windows 2000 Active
Directory server
-c KDC = Host name of the Windows 2000 Server
-s server = Host name of the Windows 2000 Server
```

1. Utilisez la commande **config.krb5** comme indiqué dans l'exemple suivant :

```
config.krb5 –C –r MYREALM –d xyz.com –c w2k.xyz.com –s w2k.xyz.com
```

2. Windows 2000 prend en charge les types de chiffrement DES–CBC–MD5 et DES–CBC–CRC. Modifiez le fichier **krb5.conf** de sorte qu'il contienne des informations similaires à ce qui suit :

```
[libdefaults]
       default_realm = MYREALM
       default_keytab_name=FILE:/etc/krb5/krb5.keytab
       default_tkt_enctypes = des-cbc-crc des-cbc-md5
       default_tgs_enctypes = des-cbc-crc des-cbc-md5
```

3. Ajoutez les strophes suivantes dans le fichier **methods.cfg** :

```
KRB5A:
                 program = /usr/lib/security/KRB5A
                 options = authonly
KRB5Afiles:
                 options = db=BUILTIN,auth=KRB5A
```

4. Effectuez les opérations suivantes sur un serveur Active Directory Windows 2000 :
   a. Utilisez l'outil Active Directory Management pour créer un nouveau compte utilisateur pour l'hôte SWsym.AIX *krbtest*, comme suit :
      i. Sélectionnez le dossier Users.
      ii. Avec le bouton droit de la souris, cliquez sur **New**.
      iii. Sélectionnez **user**.
      iv. Entrez le nom `krbtest`.
   b. Utilisez la commande **Ktpass** depuis la ligne de commande pour créer un fichier **keytab** et configurez le compte pour l'hôte SWsym.AIX comme suit : Par exemple, pour créer un fichier **keytab** nommé **krbtest.keytab**, entrez :

      ```
      Ktpass –princ host/krbtest.xyz.com@MYREALM –mapuser krbtest –pass
      password –out krbtest.keytab
      ```

   c. Copiez le fichier **keytab** sur le système hôte SWsym.AIX.
   d. Fusionnez le fichier **keytab** dans le fichier **/etc/krb5/krb5.keytab** en procédant comme suit :

      ```
      $ ktutil
      ktutil: rkt krbtest.keytab
      ktutil: wkt /etc/krb5/krb5.keytab
      ktutil: q
      ```

   e. Créez des comptes de domaine Windows 2000 à l'aide des outils de gestion des utilisateurs d'Active Directory.
   f. Créez des comptes SWsym.AIX correspondant aux comptes de domaine Windows 2000, de sorte que le processus de connexion utilise l'authentification Kerberos, comme suit :

      ```
      mkuser registry=KRB5Afiles SYSTEM=KRB5Afiles user0
      ```

Questions sur le module de chargement d'authentification KRB5A et informations sur le dépannage

La section suivante fournit des réponses aux questions relatives au module de chargement d'authentification KRB5A, de même que des informations sur le dépannage.

Remarque : Le module de chargement d'authentification KRB5A est pris en charge uniquement sous AIX 5.2 et versions ultérieures.

Comment configurer un client Kerberos AIX qui sera authentifié par un serveur Active Directory KDC ?

Utilisez la commande **config.krb5** pour configurer un client Kerberos AIX. Des informations sur le serveur Kerberos sont nécessaires pour la configuration du client. Si un serveur Windows 2000 Active Directory est choisi comme serveur Kerberos, utilisez la commande **config.krb5** avec les options suivantes :

–r realm	Nom de domaine de Active Directory
–d domain	Nom de domaine de la machine hébergeant le service Active Directory
–c KDC	Nom d'hôte du serveur Windows 2000
–s server	Nom d'hôte du serveur Windows 2000

Utilisez la commande **config.krb5** comme indiqué dans l'exemple suivant :

```
config.krb5 –C –r MYREALM –d xyz.com –c w2k.xyz.com –s w2k.xyz.com
```

Windows 2000 prend en charge les types de chiffrement DES–CBC–MD5 et DES–CBC–CRC. Modifiez le fichier **krb5.conf** de sorte qu'il contienne des informations similaires à ce qui suit :

```
[libdefaults]
       default_realm = MYREALM
       default_keytab_name=FILE:/etc/krb5/krb5.keytab
       default_tkt_enctypes = des-cbc-crc des-cbc-md5
       default_tgs_enctypes = des-cbc-crc des-cbc-md5
```

Ajoutez les strophes suivantes dans le fichier **methods.cfg** :

```
KRB5A:
                 program = /usr/lib/security/KRB5A
                 options = authonly
KRB5Afiles:
                 options = db=BUILTIN,auth=KRB5A
```

Effectuez les opérations suivantes sur le serveur Active Directory :

1. Utilisez l'outil Active Directory Management pour créer un nouveau compte utilisateur pour l'hôte AIX *krbtest*.
   - Sélectionnez le dossier Users (Utilisateurs).
   - Cliquez sur le bouton droit de la souris et sélectionnez New (Nouveau).
   - Sélectionnez user (utilisateur).
   - Entrez le nom *krbtest*.
2. Utilisez la commande **Ktpass** depuis la ligne de commande pour créer un fichier **krbtest.keytab** et configurez le compte pour l'hôte AIX comme suit :

```
Ktpass –princ host/krbtest.xyz.com@MYREALM –mapuser krbtest –pass password
\
        –out krbtest.keytab
```

3. Copiez le fichier **krbtest.keytab** sur le système hôte AIX.

4. Fusionnez le fichier **krbtest.keytab** dans le fichier **/etc/krb5/krb5.keytab** en procédant comme suit :

```
$ ktutil
ktutil: rkt krbtest.keytab
ktutil: wkt /etc/krb5/krb5.keytab
ktutil: q
```

5. Créez des compte de domaine Windows 2000 à l'aide des outils de gestion des utilisateurs Active Directory.
6. Créez des comptes AIX correspondant aux comptes de domaine Windows 2000, de sorte que le processus de connexion puisse utiliser l'authentification Kerberos, comme suit :

```
mkuser registry=KRB5Afiles SYSTEM=KRB5Afiles user0
```

Comment modifier la configuration AIX pour la connexion intégrée Kerberos ?

Pour activer une connexion intégrée Kerberos, modifiez le fichier **methods.cfg**. L'entrée de module de chargement composé doit être ajoutée au fichier **methods.cfg**. Le côté authentification est `KRB5A`. Le côté base de données peut être choisi en tant que `BUILTIN` ou `LDAP`. `BUILTIN` est le référentiel standard de compte utilisateur AIX qui utilise les fichiers ASCII. Par exemple, si vous choisissez `BUILTIN` comme référentiel de compte utilisateur AIX, modifiez le fichier **methods.cfg** comme suit :

```
Example: Local file system is chosen as the AIX user account repository.

KRB5A:
         program = /usr/lib/security/KRB5A
         options=authonly

KRB5Afiles:
         options = db=BUILTIN,auth=KRB5A


Example: LDAP is chosen as the AIX user account repository.

KRB5A:
         program = /usr/lib/security/KRB5A
         options=authonly

LDAP:
         program = /usr/lib/security/LDAP

KRB5ALDAP:
         options = auth=KRB5A,db=LDAP
```

Comment créer un utilisateur AIX pour la connexion intégrée Kerberos avec le module de chargement KRB5A ?

Pour créer un utilisateur AIX pour la connexion intégrée Kerberos à l'aide du module de chargement KRB5A, utilisez la commande **mkuser** comme suit :

```
mkuser registry=KRB5Afiles SYSTEM=KRB5Afiles auth_domain=MYREALM foo
```

Pour plus d'informations sur les attributs **auth_name** et **auth_domain**, reportez–vous à la section Rôle des attributs auth_name et auth_domain, page 16-14.

Comment créer des principaux Kerberos sur Active Directory ?

La création de comptes utilisateur Windows 2000 implique la création des principaux. Par exemple, si vous créez un compte utilisateur nommé `foo` sur Active Directory, le principal `foo@MYREALM` associé à l'utilisateur `foo` est également créé. Pour plus d'informations concernant la création d'utilisateurs sur Active Directory, reportez–vous à la documentation Active Directory relative à la gestion des utilisateurs.

Comment modifier le mot de passe d'un utilisateur authentifié Kerberos ?

Pour modifier le mot de passe d'un utilisateur authentifié Kerberos, utilisez la commande **passwd** comme suit :

```
passwd –R KRB5Afiles foo
```

Comment supprimer un utilisateur authentifié Kerberos ?

Pour supprimer un utilisateur authentifié Kerberos, utilisez la commande **rmuser**. Cependant, cette procédure supprime uniquement l'utilisateur de AIX. L'utilisateur doit également être supprimé de Active Directory à l'aide des outils de gestion des utilisateurs Active Directory.

```
passwd –R KRB5Afiles foo
```

Comment migrer un utilisateur AIX vers un utilisateur authentifié Kerberos ?

Si l'utilisateur possède déjà un compte sur Active Directory, la commande **chuser** convertit l'utilisateur en un utilisateur authentifié Kerberos, comme indiqué dans l'exemple suivant :

```
chuser registry=KRB5Afiles SYSTEM=KRB5Afiles auth_domain=MYREALM foo
```

Si l'utilisateur ne possède pas de compte sur Active Directory, vous devez en créer un. Utilisez ensuite la commande **chuser**. Le compte Active Directory et le nom d'utilisateur AIX ne doivent pas nécessairement être identiques. Si les deux noms sont différents, utilisez l'attribut **auth_name** pour établir une correspondance avec le nom Active Directory. Par exemple, pour établir une correspondance entre le nom d'utilisateur AIX `chris` et le nom Active Directory `christopher`, entrez :

```
chuser registry=KRB5Afiles SYSTEM=KRB5Afiles auth_name=christopher
auth_domain=MYREALM chris
```

Que faire en cas d'oubli de mot de passe ?

Sur Active Directory, le mot de passe doit être modifié par l'administrateur. Sur AIX, l'utilisateur root ne peut pas définir le mot de passe d'un principal Kerberos.

Quel est le rôle des attributs auth_name et auth_domain ?

Les attributs **auth_name** et **auth_domain** servent à établir une correspondance entre les noms d'utilisateur AIX et les noms des principaux Kerberos sur Active Directory. Par exemple, si pour l'utilisateur AIX `chris`, `auth_name=christopher` et `auth_domain=SOMEREALM`, le nom de principal Kerberos est `christopher@SOMEREALM`. Le nom de domaine `SOMEREALM` diffère du nom de domaine par défaut `MYREALM`. Ceci permet à l'utilisateur `chris` d'être authentifié auprès du domaine `SOMEREALM` au lieu du domaine `MYREALM`.

Ces attributs sont facultatifs.

Un utilisateur authentifié Kerberos peut–il être authentifié via l'authentification AIX standard ?

La réponse est oui. Effectuez les opérations suivantes pour authentifier l'utilisateur authentifié Kerberos à l'aide de l'authentification AIX :

1. L'utilisateur définit le mot de passe AIX (**/etc/security/passwd**) à l'aide de la commande **passwd**, comme suit :

```
passwd –R files foo
```

2. Modifiez l'attribut **SYSTEM** de l'utilisateur en procédant comme suit :

```
chuser –R KRB5Afiles SYSTEM=compat foo.
```

   L'authentification Kerberos est modifiée en authentification cryptée.

Si vous souhaitez utiliser l'authentification cryptée comme mécanisme de sauvegarde, l'attribut **SYSTEM** est modifié comme suit :

```
chuser –R KRB5Afiles SYSTEM="KRB5Afiles or compat" foo.
```

Est–il nécessaire de configurer le serveur Kerberos (KDC) sur AIX en cas d'utilisation d'un serveur Windows 2000 Active Directory ?

Non, car les utilisateurs sont authentifiés par un KDC Active Directory, il n'est donc pas nécessaire de configurer le KDC sur AIX. Si vous souhaitez utiliser le KDC des services d'authentification réseau AIX en tant que serveur Kerberos, ce dernier doit être configuré.

AIX refuse mon mot de passe

Vérifiez que le mot de passe répond aux exigences de AIX et de Kerberos.
Vérifiez aussi que KDC est configuré et qu'il fonctionne correctement.

Connexion au système impossible

Si vous ne pouvez pas vous connecter au système, procédez de la manière suivante :

- Vérifiez que le KDC est lancé et activé.
  - Sur les systèmes AIX, entrez :

    ```
    ps –ef | grep krb5kdc
    ```

  - Sur les systèmes Windows 2000, procédez comme suit :
    1. Dans le Panneau de configuration, cliquez deux fois sur l'icone Outils d'administration
    2. Cliquez deux fois sur l'icône Services.
    3. Vérifiez que l'état du centre de distribution de clés (KDC) Kerberos est `Démarré`.
- Sur les systèmes AIX, vérifiez que le fichier **/etc/krb5/krb5.conf** désigne le KDC approprié et qu'il contient des paramètres valides.
- Sur les systèmes AIX, vérifiez que le fichier **keytab** client contient le ticket hôte. Par exemple, supposez que le fichier **keytab** par défaut est **/etc/krb5/krb5.keytab**. Entrez la commande suivante :

```
$ ktutil
ktutil: rkt /etc/krb5/krb5.keytab
ktutil: l

slot   KVNO   Principal
------ ------ ---------------------------------------------------
     1      4 host/krbtest.xyz.com@MYREALM

ktutil: q
```

- Si les attributs **auth_name** et **auth_domain** sont définis, vérifiez qu'ils se rapportent à un nom principal valable sur le serveur ADS KDC.
- Vérifiez que l'attribut **SYSTEM** est défini pour la connexion Kerberos (**KRB5Afiles** ou **KRB5ALDAP**).
- Vérifiez que le mot de passe est toujours valide.

Comment désactiver la vérification de TGT ?

Le principal **host/ *Nom_Hôte*** permet de vérifier un TGT. Les clés de ce principal hôte sont stockées dans le fichier **keytab** par défaut de Kerberos et ce fichier **keytab** doit être transmis de façon sécurisée depuis le serveur Active Directory Windows 2000 vers la machine cliente comme décrit dans le Guide de Sécurité. La vérification de TGT peut être désactivée en définissant une option dans la strophe KRB5A du fichier **/usr/lib/security/methods.cfg**, comme suit :

```
KRB5A:
         program = /usr/lib/security/KRB5A
         options = tgt_verify=no
KRB5Afiles:
         options = db=BUILTIN,auth=KRB5A
```

Les valeurs possibles de **tgt_verify** sont `no` ou `false` pour la désactivation, et `yes` ou `true` pour l'activation. Par défaut, la vérification de TGT est désactivée. Si la valeur de **tgt_verify** est `no`, la vérification de TGT n'est pas effectuée et il n'est pas nécessaire de transférer les clés du principal hôte. Cela élimine la nécessité d'utiliser le fichier **keytab** pour l'authentification lorsque le module KRB5A est utilisé.

Module Kerberos

Rôle

Intégrité des services Kerberos et appels confidentiels

Description

Le module Kerbos est une extension du noyau utilisée par le client NFS et le code serveur. Ce module permet au client NFS et au code serveur de traiter les fonctions de confidentialité et d'intégrité de messages Kerberos sans avoir à effectuer des appels vers le démon **gss**. Le module est chargé par le démon **gss**. Les méthodes utilisées sont basées sur le service d'authentification réseau version 1.2, qui, à son tour, est basé sur MIT Kerberos.

Emplacement

usr/lib/drivers/krb5.ext

Informations connexes

Démon **gss**.

Chapitre 17. Serveur RADIUS (Remote Authentication Dial–In User Service)

Le protocole RADIUS (Remote Authentication Dial–In User Service) est un protocole d'accès au réseau conçu pour effectuer l'authentification (page 17-2),

l'autorisation (page 17-5) et la comptabilité. Il s'agit d'un protocole fonctionnant au niveau des ports, qui définit les communications entre les serveurs d'accès au réseau (NAS) et les serveurs d'authentification et de comptabilité.

Un serveur NAS fonctionne en tant que client du serveur RADIUS. Les transactions entre le client et le serveur RADIUS sont authentifiées via l'utilisation d'un *secret partagé* qui n'est pas envoyé sur le réseau. Tous les mots de passe utilisateur transférés entre le client et le serveur RADIUS sont chiffrés.

Le client est chargé de transmettre les informations utilisateur vers les serveurs RADIUS désignés, puis d'agir après réception de la réponse. Les serveurs RADIUS sont chargés de recevoir les requêtes de connexion des utilisateurs, d'authentifier les utilisateurs, puis de renvoyer toutes les informations de configuration nécessaires pour permettre au client de fournir le service aux utilisateurs. Dans le cas d'une configuration avancée de proxy, un serveur RADIUS peut agir en tant que client proxy (page 17-8) pour d'autres serveurs RADIUS. RADIUS utilise le protocole de transfert **UDP** (User Datagram Protocol).

Le protocole d'authentification et d'autorisation RADIUS repose sur la norme http://www.ietf.org. Le serveur fournit également le protocole de comptabilité défini dans http://www.ietf.org. Les autres normes acceptées sont celles indiquées sur http://www.ietf.org, une partie de celles indiquées sur http://www.ietf.org ainsi que les messages d'expiration de mot de passe indiqués sur http://www.ietf.org. Toutes les normes RFC indiquées ci–dessus sont répertoriées à l'adresse http://www.ietf.org.

Principales caractéristiques de RADIUS :

- Authentification, autorisation et comptabilité (méthode AAA)
- Les méthodes d'authentification via le protocole **PAP** (Password Authentication Protocol), le protocole **CHAP** (Challenge Handshake Authentication Protocol) et le protocole **EAP** (Extensible Authentication Protocol).
- Transmission de paquets par proxy pour l'authentification et l'autorisation
- Prise en charge de plusieurs bases de données d'authentification (stockage des informations utilisateur)
  - UNIX (AIX **/etc/password**)
  - Base de données locale
  - Répertoire LDAP
- Attributs de retour et stratégie utilisateur d'autorisation par défaut
  - Définition de stratégie utilisateur individuelle
- Interface SMIT d'administration
- Prise en charge de l'expiration de mot de passe
- Prise en charge de plusieurs serveurs
- Prise en charge des attributs spécifiques au fournisseur
- Compatibilité NLS

Installation du serveur RADIUS

Vous pouvez installer le serveur RADIUS à l'aide de la commande **installp** ou bien de l'outil SMIT. Le logiciel RADIUS se trouve sur le support de base AIX, dans les images nommées **radius.base** et **bos.msg.<lang>.rte**.

Si vous prévoyez d'utiliser le répertoire LDAP (page 17-3) comme base de données pour stocker les noms et les mots de passe des utilisateurs, vous devez installer **ldap.server**. Le logiciel **ldap.client** doit être installé pour chaque serveur RADIUS installé.

Les démons RADIUS peuvent être lancés à l'aide des commandes SRC. Lors de leur démarrage, plusieurs processus **radiusd** sont exécutés :

- Un processus pour l'autorisation
- Un processus pour la comptabilité
- Un processus destiné à surveiller les autres démons

Au redémarrage, les démons sont automatiquement lancés au niveau d'exécution 2. Pour modifier cette routine, modifiez le fichier **/etc/rc.d/rc2.d/Sradiusd**.

Authentification RADIUS

La méthode d'authentification classique consiste en général à vérifier le nom et le mot de passe fixe lorsque l'utilisateur se connecte à une machine ou demande un service.

La méthode RADIUS repose sur une base de données d'authentification où sont stockées les informations relatives aux utilisateurs, tels que leur ID et leur mot de passe. Pour authentifier les utilisateurs, le serveur peut utiliser une base de données locale, des mots de passe UNIX ou LDAP. L'emplacement de la base de données est configuré dans le fichier **/etc/radius/radiusd.conf** du serveur au cours de l'installation, puis vous pouvez modifier ce fichier à l'aide de l'outil SMIT. Pour plus d'informations sur les fichiers de configuration RADIUS, voir Fichiers de configuration RADIUS, page 17-11.

Bases de données utilisateurs

Le logiciel RADIUS peut stocker les informations relatives aux utilisateurs dans différentes bases de données. Un seul type de base de données utilisateurs peut être utilisé parmi les suivants :

- Local, page 17-2
- UNIX, page 17-3
- LDAP, page 17-3

Local

Si la zone **database_location** du fichier **radiusd.conf** ou l'entrée **Database Location** de l'outil SMIT contient le mot `Local`, alors le serveur RADIUS utilise l'emplacement **/etc/radius/dbdata.bin** pour tous les ID et les mots de passe des utilisateurs.

La base de données utilisateurs locale est un fichier ordinaire qui contient l'ID et le mot de passe de chaque utilisateur. Les mots de passe sont enregistrés au format haché. Le hachage est une méthode d'adressage rapide permettant d'accéder directement aux données de l'espace mémoire. Pour ajouter, supprimer ou modifier les mots de passe utilisateur, lancez la commande **raddbm** ou utilisez l'outil SMIT. Lorsque le démon **radiusd** démarre, il lit le fichier **radiusd.conf** et charge les ID et les mots de passe utilisateur dans la mémoire.

Remarque : Un ID utilisateur peut contenir jusqu'à 253 caractères et un mot de passe jusqu'à 128 caractères.

Pour utiliser la base de données utilisateurs locale, sélectionnez `Local` dans la zone **Database Location** comme indiqué ci–dessous :

```
          Configure Server

 RADIUS Directory             /etc/radius
 *Database Location           [Local]
 Local AVL Database File Name [dbdata.bin]
 Local Accounting             [ON]

 Debug Level                  [3]
.
.
.
```

UNIX

L'option d'authentification UNIX permet au serveur RADIUS d'authentifier l'utilisateur au moyen de la méthode d'authentification locale du système.

Pour utiliser l'authentification UNIX locale, modifiez la zone **database_location** du fichier **radiusd.conf** ou sélectionnez `UNIX` dans la zone **Database Location**. Cette méthode d'authentification appelle l'interface API **authenticate()** UNIX pour authentifier l'ID et le mot de passe de l'utilisateur. Les mots de passe sont enregistrés dans le fichier de données utilisé par UNIX, par exemple **/etc/passwords**. La création des ID et des mots de passe est effectuée à l'aide de la commande **mkuser** ou via l'outil SMIT.

Pour utiliser la base de données UNIX, sélectionnez `UNIX` dans la zone **Database Location** comme indiqué ci–dessous :

```
          Configure Server

 RADIUS Directory             /etc/radius
 *Database Location           [UNIX]
 Local AVL Database File Name [dbdata.bin]
 Local Accounting             [ON]

 Debug Level                  [3]
.
.
.
```

LDAP

Le serveur RADIUS peut utiliser LDAP Version 3 pour stocker les données des utilisateurs. Pour accéder à distance aux données des utilisateurs, le serveur RADIUS utilise les appels API de LDAP Version 3. L'accès LDAP Version 3 est établi si la valeur de la zone **database_location** du fichier **/etc/radiusd.conf** est `LDAP` et si le nom du serveur, l'ID utilisateur et le mot de passe de l'administrateur LDAP sont configurés.

AIX utilise les bibliothèques LDAP Version 3 prises en charge et regroupées en modules sur Tivoli Directory Server. LDAP est un protocole évolutif qui permet de centraliser les données utilisateur et les données en cours de traitement, ce qui facilite l'administration du serveur RADIUS. L'utilitaire en ligne de commande **ldapsearch** permet d'afficher les données RADIUS.

Pour permettre à RADIUS d'utiliser le protocole LDAP, vous devez configurer et gérer celui–ci.

Le serveur RADIUS fournit des fichiers **ldif** LDAP qui permettent d'ajouter le schéma RADIUS (y compris les classes d'objets et les attributs) dans un répertoire, mais vous devez installer et configurer LDAP.

Un suffixe distinct spécifique est créé pour RADIUS afin de permettre l'utilisation des objets LDAP RADIUS. Ce suffixe est un conteneur appelé **cn=aixradius** contenant deux classes

d'objets, comme décrit dans Configuration du serveur LDAP RADIUS, page 17-18. Appliquez le fichier **ldif** fournit par RADIUS, qui permet de créer le suffixe et le schéma RADIUS.

Si vous utilisez LDAP comme base de données d'authentification, vous disposerez des fonctionnalités suivantes :

1. Base de données utilisateurs accessible et visualisable depuis tous les serveurs RADIUS
2. Liste des utilisateurs actifs
3. Possibilité de définir un nombre maximum de connexions par ID utilisateur
4. Type **EAP** qui peut être configuré pour chaque utilisateur
5. Date d'expiration du mot de passe

Pour utiliser la base de données LDAP, sélectionnez `LDAP` dans la zone **Database Location** comme indiqué ci–dessous :

```
          Configure Server

 RADIUS Directory             /etc/radius
 *Database Location           [LDAP]
 Local AVL Database File Name [dbdata.bin]
 Local Accounting             [ON]

 Debug Level                  [3]
.
.
.
```

Méthodes d'authentification

RADIUS propose deux méthodes standard de chiffrement du mot de passe fourni par l'utilisateur :

- Protocole **PAP** (Password Authentication Protocol)
- Protocole **CHAP** (Challenge Handshake Authentication Protocol)

Protocole PAP (Password Authentication Protocol)

Le protocole **PAP** (Password Authentication Protocol) assure la sécurité via le codage du mot de passe utilisateur avec un algorithme de hachage MD5 dont la valeur peut être créée par le client et le serveur. Il fonctionne de la manière suivante :

1. Dans des paquets contenant le mot de passe utilisateur, la zone d'authentification contient un nombre aléatoire de 16 octets appelé authentificateur de requête.
2. L'authentificateur de requête et le secret partagé du client sont hachés par l'algorithme MD5. Il en résulte une valeur hachée de 16 octets.
3. Le mot de passe fourni par l'utilisateur est étendu à une longueur de 16 octets avec des valeurs de remplissage nulles.
4. L'opération XOR (ou exclusif) est appliquée entre la valeur hachée de l'étape 2 et le mot de passe étendu. Ce sont ces données qui sont envoyées dans le paquet en tant qu'attribut **user_password**.
5. Le serveur RADIUS calcule la même valeur hachée que celle de l'étape 2.
6. L'opération XOR est appliquée entre cette valeur hachée et les données du paquet de l'étape 4, ce qui permet de récupérer le mot de passe.

Protocole CHAP (Challenge Handshake Authentication Protocol)

RADIUS prend également en charge la méthode de protection du mot de passe CHAP du protocole PPP. Avec le protocole CHAP, le mot de passe de l'utilisateur n'est pas envoyé sur le réseau. A la place du mot de passe, c'est une valeur hachée par l'algorithme MD5 qui est envoyée, puis le serveur RADIUS recrée les données à partir des informations de l'utilisateur, notamment le mot de passe stocké, puis compare ces données à la valeur envoyée par le client.

Protocole EAP (Extensible Authentication Protocol)

Le protocole **EAP** (Extensible Authentication Protocol) est conçu pour prendre en charge plusieurs méthodes d'authentification. **EAP** définit la structure des communications soumises à l'authentification entre un client et un serveur d'authentification, sans définir le contenu des données d'authentification. Ce contenu est défini par la méthode **EAP** spécifique utilisée pour l'authentification. Les principales méthodes **EAP** utilisées sont :

- MD5–Challenge
- One–Time Password (mot de passe à usage unique)
- Generic Token Card (carte à jeton)
- Transport Layer Security (TLS) (sécurité au niveau de la couche transport)

Le protocole **EAP** permet de définir des attributs RADIUS qui permettent le transfert de données **EAP** entre le serveur RADIUS et ses clients. Le serveur RADIUS peut envoyer ces données **EAP** directement aux serveurs d'arrière–plan qui mettent en oeuvre les différentes méthodes d'authentification **EAP**.

Le serveur RADIUS AIX prend en charge uniquement la méthode **EAP** MD5–Challenge. L'utilisation de la méthode **EAP** d'authentification des utilisateurs est définie au niveau utilisateur, dans une entrée de la base de données locale ou LDAP. Par défaut **EAP** est désactivé pour chaque utilisateur.

Autorisation RADIUS

RADIUS accorde des attributs d'autorisation à chaque utilisateur en fonction du contenu des fichiers de stratégie d'autorisation **default.auth** et **default.policy**. Les attributs d'autorisation sont des attributs conformes à la norme RFC du protocole RADIUS et définis dans le fichier **/etc/radius/dictionary**. L'autorisation est facultative et dépend de la configuration du serveur NAS ou du point d'accès. Pour utiliser les attributs d'autorisation, il est nécessaire de les configurer. L'autorisation est possible uniquement après une authentification réussie.

Une stratégie est constituée de paires attribut–valeur configurables qui permettent de définir la façon dont les utilisateurs accèdent au réseau. Une stratégie peut être appliquée au serveur RADIUS de façon globale ou bien à un utilisateur particulier.

Le support de base contient deux fichiers de configuration d'autorisation : **/etc/radius/authorization/default.auth** et **default.policy**. Le fichier **default.policy** permet de faire correspondre les paquets entrants de demande d'accès `access request`. Ce fichier contient des paires attribut–valeur vides à configurer selon les besoins. Après l'authentification, un paquet `access accept` ou `access reject` est renvoyé au client en fonction de la stratégie définie.

Chaque utilisateur peut également être associé à un fichier ***id_utilisateur*.policy**. Lorsqu'un ID utilisateur est associé à un fichier de stratégie spécifique, les attributs de ce fichier sont vérifiés en priorité. Si les paires attribut–valeur du fichier ***id_utilisateur*.policy** ne correspondent pas exactement, le fichier **default.policy** est vérifié. Si les paires d'attributs du paquet de `access request` ne correspondent pas à ce fichier non plus, un paquet `access reject` est envoyé. Si une correspondance est trouvée dans l'un des fichiers, un paquet `access accept` est envoyé au client. Cette stratégie comprend deux niveaux.

Le fichier **default.auth** constitue la liste des paires attribut–valeur à renvoyer au client après la vérification de la stratégie. Le fichier **default.auth** contient également des paires attribut–valeur vides à configurer selon les besoins. Pour configurer les attributs d'autorisation, modifiez directement le fichier **default.auth** ou utilisez l'outil SMIT. Chaque attribut contenant une valeur est automatiquement renvoyé au serveur NAS dans un paquet `access accept`.

Vous pouvez également définir des attributs de retour spécifiques à un utilisateur en créant un fichier d'extension **.auth** en le nommant avec l'ID de l'utilisateur, c'est–à–dire ***id_utilisateur*.auth**. Ce fichier personnalisé doit être placé dans le répertoire **/etc/radius/authorization**. L'outil SMIT permet de créer et de modifier les fichiers utilisateur.

Les attributs d'autorisation de chaque utilisateur sont renvoyés dans un paquet `access accept` avec les attributs d'autorisation par défaut du fichier **default.auth**. Si des valeurs sont communes entre les fichiers **default.auth** et ***id_utilisateur*.auth**, les valeurs de l'utilisateur sont prioritaires sur les valeurs par défaut. Cela permet de définir globalement des attributs d'autorisation (services ou ressources) pour tous les utilisateurs puis de définir des attributs spécifiques au niveau utilisateur.

Le processus d'autorisation est le suivant :

1. Au démarrage du démon, la stratégie par défaut et les listes d'autorisation des fichiers **/etc/radius/authorization/default.policy** et **default.auth** sont lues en mémoire.
2. Authentification de l'ID et du mot de passe utilisateur.
3. Les paires attribut–valeur du paquet entrant sont vérifiées.
   a. Vérification du fichier ***id_utilisateur*.auth**.
   b. Si aucune correspondance n'est trouvée, le fichier **default.policy** est vérifié.
   c. Si aucune correspondance n'est trouvée, un paquet `access reject` est envoyé.
4. Application des attributs d'autorisation de l'utilisateur, le cas échéant.
   a. Lecture des fichiers **/etc/radius/authorization/ *id_utilisateur*.auth** et **default.auth** et comparaison des entrées.
   b. Application de l'entrée du fichier de l'utilisateur avant l'entrée globale.
5. Renvoi des attributs d'autorisation dans un paquet `access accept`.

Comptabilité RADIUS

Le serveur de comptabilité RADIUS reçoit les demandes de comptabilité des clients et leur renvoie une réponse indiquant qu'il a bien reçu la demande et enregistré les données de comptabilité.

Fonctionnement du serveur de comptabilité RADIUS

La comptabilité locale est activée dans le fichier **radiusd.conf**.

Lorsqu'un client est configuré de façon à utiliser la comptabilité RADIUS, il génère un paquet ACCOUNTING_START lors du lancement d'un service, indiquant le type de service fourni et l'utilisateur auquel est destiné ce service. Le client envoie le paquet au serveur de comptabilité RADIUS, lequel renvoie un accusé de bonne réception du paquet. A la fin du service, le client génère un paquet ACCOUNTING_STOP indiquant le type de service fourni et éventuellement des statistiques telles que la durée écoulée, les octets ou le nombre de paquets entrés et sortis. Lorsque le serveur de comptabilité RADIUS reçoit le paquet ACCOUNTING_STOP, il renvoie au client un accusé de bonne réception du paquet.

Dans les deux cas (START ou STOP), le paquet ACCOUNTING_REQUEST est envoyé au serveur de comptabilité RADIUS via le réseau. Il est préférable que le client réitère les tentatives d'envoi du paquet ACCOUNTING_REQUEST jusqu'à l'obtention de l'accusé de

réception. Si aucune réponse n'est obtenue, la demande est de nouveau envoyée. Si le serveur principal ne fonctionne pas ou est inaccessible, le client peut également transférer les demandes à un ou plusieurs serveurs secondaires, via un proxy. Pour plus d'informations, voir Services proxy, page 17-8.

Voici un exemple de données de comptabilité enregistrées dans le fichier de comptabilité. Les données réelles dépendent du contenu du paquet envoyé par le client au serveur. Les données que vous obtenez peuvent varier en fonction de la configuration du client :

```
Thu May 27 14:43:19 2004
         NAS-IP-Address = 10.10.10.1
         NAS-Port = 1
         NAS-Port-Type = Async
         User-Name =   user
         Acct-Status-Type = Start
         Acct-Authentic = RADIUS
         Service-Type = Framed-User
         Acct-Session-Id = "0000000C"
         Framed-Protocol = PPP
         Acct-Delay-Time = 0
         Timestamp = 1085686999

Thu May 27 14:45:19 2004
         NAS-IP-Address = 10.10.10.1
         NAS-Port = 1   <--   l'utilisateur  était connecté au port
physique n 1
         NAS-Port-Type = Async
         User-Name = "rod"
         Acct-Status-Type = Stop
         Acct-Authentic = RADIUS
         Service-Type = Framed-User
         Acct-Session-Id = "0000000C"   <-- notez que les ID de session
sont identiques
ce qui permet de faire correspondre les démarrages et les arrêts
         Framed-Protocol = PPP
         Framed-IP-Address = 10.10.10.2   <-- adresse IP de l'utilisateur
         Acct-Terminate-Cause = User-Request  <-- l'utilisateur a annulé la
session
         Acct-Input-Octets = 4016
         Acct-Output-Octets = 142
         Acct-Input-Packets = 35
         Acct-Output-Packets = 7
         Acct-Session-Time = 120  <-- secondes
         Acct-Delay-Time = 0
         Timestamp = 1085687119  <-- notez que l'utilisateur est resté
connecté durant 120 secondes (2 minutes)
```

Les données de comptabilité sont écrites au format RADIUS standard *attribut=valeur* dans le fichier local **/etc/var/radius/data/accounting**. Les données écrites sont les données de comptabilité récupérées du paquet et horodatées. Si le serveur de comptabilité RADIUS ne parvient pas à enregistrer le paquet de comptabilité, il n'envoie pas d'accusé de réception **Accounting_Response** au client et des informations sur l'erreur sont consignées dans le fichier **syslog**.

Remarques :

1. Vérifiez que la taille du système de fichiers **/var** est suffisante pour contenir toutes les données de comptabilité.
2. Vous pouvez utiliser des scripts Perl tiers pour analyser les données de ce fichier. Vous trouverez des exemples de scripts qui génèrent des rapports à partir de données de comptabilité à l'adresse http://www.pgregg.com/projects/radiusreport.
3. Les paquets de comptabilité peuvent également être transférés via un proxy.

Services proxy

Les services Proxy permettent au serveur RADIUS de transférer les demandes d'un serveur NAS vers un autre serveur RADIUS puis de renvoyer un message de réponse au serveur NAS. Le fonctionnement des services Proxy repose sur des ID de domaine.

Le serveur RADIUS peut fonctionner simultanément en tant que serveur proxy et en tant que serveur d'arrière–plan. Ce mécanisme est applicable à la fois pour les paquets de comptabilité et d'authentification. Pour utiliser un proxy il est important de savoir ce que représente un domaine dans ce contexte. L'identifiant du domaine est indiqué avant ou après les valeurs de l'attribut User–Name. Il permet à un serveur RADIUS d'identifier le serveur à contacter pour lancer le processus d'authentification et de comptabilité.

Par défaut, l'utilisation d'un proxy est désactivée dans le fichier **radiusd.conf**.

Exemple de domaine

L'utilisateur **Joe** est employé par la société **XYZ** à Sacramento. Le domaine correspondant à cette zone est **SAC**. Cependant, **Joe** se trouve actuellement en mission à New York City. Le domaine correspondant à New York City est **NYC**. Lorsque **Joe** se connecte au domaine **NYC**, l'attribut User–Name transmis est **SAC/Joe**. Cela indique au serveur RADIUS du domaine **NYC** que ce paquet doit être transmis au serveur qui effectue l'authentification et la comptabilité des utilisateurs du domaine **SAC**.

Utilisation de préfixes et de suffixes dans l'attribut User–Name

Le routage d'un paquet d'authentification ou de comptabilité dans un domaine repose sur l'attribut User–Name. L'attribut User–Name indique l'ordre des domaines par lesquels transite un paquet avant d'arriver au serveur final qui effectue l'authentification ou la comptabilité. Pour cela, les domaines sont regroupés en chaîne dans l'attribut User–Name. Les domaines insérés dans l'attribut User–Name, qui déterminent le cheminement du paquet, sont choisis par l'administrateur qui déploie la configuration RADIUS. Les différents noms de domaine à traverser peuvent être placés avant ou après l'attribut User–Name. Les caractères délimiteurs les plus souvent utilisés pour séparer les domaines sont la barre oblique (/) en position de préfixe avant l'attribut User–Name, et la perluète (&) en position de suffixe après l'attribut User–Name. Les délimiteurs sont configurés dans le fichier **radiusd.conf**. L'attribut User–Name est analysé de gauche à droite.

Voici un exemple d'attribut User–Name utilisant uniquement la méthode du préfixe :

```
USA/TEXAS/AUSTIN/joe
```

Voici un exemple d'attribut User–Name utilisant uniquement la méthode du suffixe :

```
joe@USA@TEXAS@AUSTIN
```

Il est possible d'utiliser les deux méthodes (préfixe et suffixe). Il est important de noter que les différents domaines indiqués sont analysés de gauche à droite : les préfixes sont donc analysés avant les suffixes. Il est nécessaire que l'utilisateur soit authentifié (ou les données de comptabilité écrites) au niveau d'un noeud.

Les exemples suivants, utilisant les deux méthodes, conduisent au même résultat que les exemples précédents :

```
USA/joe@TEXAS@AUSTIN
```

Configuration de services proxy

Les informations de configuration d'un proxy RADIUS se trouvent dans le fichier **proxy** du répertoire **/etc/radius**. Le fichier **proxy** initial contient des entrées par défaut factices. Le fichier proxy contient trois zones : **Realm Name**, **Next Hop IP address** et **Shared Secret**.

```
----------------------------------------
          Configure Proxy Rules

 List all Proxy
 Add a Proxy
 Change / Show Characteristics of a Proxy
 Remove a Proxy
 ----------------------------------------
```

L'option **List all Proxy** lit le fichier **/etc/radius/proxy** et affiche les trois zones sous forme de colonnes. Les en–têtes de colonne sont les suivants :

```
realm_name   next_hop_address  shared_secret
```

Si vous sélectionnez **Add a Proxy**, l'écran ci–dessous apparaît. Les données de l'écran sont récupérées et ajoutées à la fin du fichier **/etc/radius/proxy**.

```
------------------------------------------------------------
              Add a Proxy
 *Realm Name                                 []  (max 64 chars)
 *Next Hop IP address (dotted decimal)      [xx.xx.xx.xx]
 *Shared Secret                             []  (minimum 6, maximum 64 chars)

 ------------------------------------------------------------
```

Si vous sélectionnez l'option **Change/Show**, la liste des noms de domaine s'affiche. La liste apparaît dans une fenêtre contextuelle et vous devez sélectionner un nom de domaine.

Si vous sélectionnez l'option **Remove a Proxy**, la liste des noms de domaine s'affiche. La liste apparaît dans une fenêtre contextuelle et vous devez sélectionner un nom de domaine. Une fois le nom sélectionné, une fenêtre contextuelle de vérification apparaît avant la suppression du domaine.

Voici un exemple de la section d'un fichier **radiusd.conf** relative aux données de configuration d'un proxy :

```
#-------------------------------------------------------------------#
#       PROXY RADIUS Information                                    #
#                                                                   #
#                                                                   #
#   Proxy_Allow               :  ON or OFF. If ON, then the server  #
#                                  can proxy packets to realms it   #
#                                  knows of and the following       #
#                                  fields must also be configured.  #
#   Proxy_Use_Table           :  ON or OFF. If ON, then the server  #
#                                  can use table for faster         #
#                                  processing of duplicate requests #
#                                  Can be used without proxy ON, but#
#                                  it is required to be ON if       #
#                                  Proxy_Use_Table is set to ON.    #
#   Proxy_Realm_name          :  This field specifies the realm     #
#                                  this server services.            #
#   Proxy_Prefix_delim        :  A list of separators for parsing   #
#                                  realm names added as a prefix to #
#                                  the username.  This list must be #
#                                  mutually exclusive to the Suffix #
#                                  delimiters.                      #
#   Proxy_Suffix_delim        :  A list of separators for parsing   #
#                                  realm names added as a suffix to #
#                                  the username.  This list must be #
#                                  mutually exclusive to the Prefix #
#                                  delimiters.                      #
#   Proxy_Remove_Hops         :  YES or NO.  If YES then the        #
#                                  will remove its realm name, the  #
#                                  realm names of any previous hops #
#                                  and the realm name of the next   #
#                                  server the packet will proxy to. #
#                                                                   #
#   Proxy_Retry_count         :  The number of times to attempt     #
#                                  to send the request packet.      #
#                                                                   #
#   Proxy_Time_Out            :  The number of seconds to wait      #
#                                  in between send attempts.        #
#                                                                   #
#-------------------------------------------------------------------#
Proxy_Allow               :   OFF
Proxy_Use_Table           :   OFF
Proxy_Realm_name          :
Proxy_Prefix_delim        :   $/
Proxy_Suffix_delim        :   @.
Proxy_Remove_Hops         :   NO
Proxy_Retry_count         :   2
Proxy_Time_Out            :   3
```

Configuration du serveur RADIUS

Le démon du serveur RADIUS utilise plusieurs fichiers de configuration. Les données de configuration du serveur sont enregistrées dans le fichier **/etc/radius/radiusd.conf**. Le fichier de configuration fourni avec le serveur contient des valeurs par défaut.

Pour plus d'informations, voir la section Fichiers de configuration RADIUS, page 17-11.

Remarque : Tous les fichiers de configuration appartiennent à l'utilisateur **root** et au groupe **sécurité**. Après toute modification des fichiers de configuration, vous devez relancer le serveur pour appliquer les modifications.

Voici un exemple d'écran SMIT de configuration du serveur RADIUS :

```
          Configure Server

 RADIUS Directory            /etc/radius
 *Database Location           [UNIX]
 Local AVL Database File Name [dbdata.bin]
 Local Accounting             [ON]

 Debug Level                  [3]
 Accept Reply-Message         []
 Reject Reply-Message         []
 Challenge Reply-Message      []
 Password Expired Reply Message []
 Support Renewal of Expired Passwords [NO]
 Require Message Authenticator  [NO]

 *Authentication Port Number  [1812]
 *Accounting Port Number      [1813]

 LDAP Server Name             []
 LDAP Server Port Number      [389]
 LDAP Server Admin Distinguished Name     []
 LDAP Server Admin Password         []
 LDAP Base Distinguished Name       [cn=aixradius]
 LDAP Size Limit                    [0]
 LDAP Hop Limit                     [0]
 LDAP wait time limit               [10]
 LDAP debug level                   [ 0]

 Proxy Allowed                      [OFF]
 Proxy Use table                    [OFF]
 Proxy Realm Name                   []
 Proxy Prefix Delimiters            [$/]
 Proxy Suffix Delimiters            [@.]
                                         NOTE: prefix & suffix are mutually
exclusive
 Proxy Remove Hops                  [NO]
 Proxy Retry Count                  [2]
 Proxy Timeout                      [30]
```

Fichiers de configuration RADIUS

Le démon RADIUS utilise plusieurs fichiers de configuration. Ces fichiers concernent l'installation du serveur, la description des clients et la configuration des proxys. Des exemples de ces fichiers sont livrés avec le serveur RADIUS. Chaque fichier est décrit dans les sections suivantes.

Fichier radiusd.conf

Par défaut, RADIUS recherche le fichier **radiusd.conf** dans le répertoire **/etc/radius**. Les entrées du fichier de configuration doivent être au format indiqué. Si une valeur ou un mot clé n'est pas valide, RADIUS ne l'accepte pas et applique la valeur ou le mot clé par défaut. Au lancement des démons RADIUS, consultez **syslog** pour voir si des erreurs se sont produites sur les paramètres de configuration. Toutes les erreurs de configuration n'entraînent pas l'arrêt du serveur

Ce fichier doit être protégé en lecture et en écriture car il est lié au fonctionnement des serveurs d'authentification et de comptabilité. De plus, ce fichier peut contenir des données confidentielles.

Remarque : Si vous modifiez le fichier **radiusd.conf** ne modifiez pas l'ordre des entrées. Le fonctionnement de l'outil SMIT repose sur l'ordre défini.

```
#------------------------------------------------------------------#
#                Global Configuration                              #
#                                                                  #
#   RADIUSdirectory  :  This is the base directory for the RADIUS  #
#                       daemon. The daemon will search this        #
#                       directory for further configuration files. #
#                                                                  #
#   Database_location : This is the value of where the             #
#                       authentication (user ids & passwords)      #
#                       will be stored and retrieved.              #
#                       Valid values: Local, LDAP, UNIX            #
#                       UNIX  - User defined in AIX system         #
#                       Local - Local AVL Database using raddbm    #
#                       LDAP  - Central Database                   #
#                                                                  #
#   Local_Database   :  This indicates the name of the local       #
#                       database file to be used.                  #
#                       This field must be completed if the        #
#                       Database location is Local.                #
#                                                                  #
#   Local_Accounting :  When this flag is set to ON or TRUE a file #
#                       is created which contains a record of      #
#                       START and STOP packets received from the   #
#                       Network Access Server(NAS).  The log can   #
#                       be found at                                #
#                          /var/radius/data/accounting             #
#                                                                  #
#   Debug_Level      :  This pair sets the debug level at which    #
#                       the RADIUS server will run.  Appropriate   #
#                       values are 0,3 or 9. The default is 3.     #
#                       Output is directed to location specified   #
#                       by *.debug stanza in /etc/syslog.conf      #
#                                                                  #
#                       Each level increases the amount of messages#
#                       sent to syslog. For example "9" includes   #
#                       the new messages provided by "9" as well   #
#                       as all messages generated by level 0 and 3.#
#                                                                  #
#                       0 : provides the minimal output to the     #
#                           syslogd log.  It sends start up        #
#                           and end messages for each RADIUS       #
#                           process.  It also logs error           #
#                           conditions.                            #
#                                                                  #
#                       3 : includes general ACCESS ACCEPT, REJECT #
#                           and DISCARD messages for each packet.  #
#                           This level provides a general audit    #
#                           trail for authentication.              #
#                                                                  #
#                       9 : Maximum amount of log data. Specific   #
#                           values of attributes while a           #
#                           transaction is passing thru            #
#                           processing and more.                   #
#                           [NOT advised under normal operations]  #
```

```
#                                                                      #
#----------------------------------------------------------------------#
RADIUSdirectory   : /etc/radius
Database_location : UNIX
Local_Database    : dbdata.bin
Local_Accounting  : ON
Debug_Level       : 3
#----------------------------------------------------------------------#
#     Reply Message Attributes                                         #
#                                                                      #
#     Accept_Reply-Message : Sent when the RADIUS server               #
#                            replies with an Access-Accept packet      #
#                                                                      #
#     Reject_Reply-Message : Sent when the RADIUS server               #
#                            replies with an Access-Reject packet      #
#                                                                      #
#     Challenge_Reply-Message   : Sent when the RADIUS server              #
#                                 replies with an Access-Challenge     #
#                                 packet                               #
#----------------------------------------------------------------------#
Accept_Reply-Message :
Reject_Reply-Message :
Challenge_Reply-Message :
Password_Expired_Reply-Message :
#----------------------------------------------------------------------#
#    Support Renewal of Expired Password                               #
#                                                                      #
#    Allow_Password_Renewal: YES or NO                                 #
#                            Setting this attribute to YES allows      #
#                            users to update their expired password    #
#                            via the RADIUS protocol. This requires    #
#                            the hardware support of                   #
#                            Access-Password-Request packets.          #
#----------------------------------------------------------------------#
Allow_Password_Renewal  : NO
#----------------------------------------------------------------------#
#    Require Message Authenticator in Access-Request                   #
#                                #                                     #
#    Require_Message_Authenticator: YES or NO                                 #
#                                   Setting this attribute to YES      #
#         checks message authenticator   #
#                                   in Access-Request packet.If not    #
#        present, it will discard the   #
#        packet.                        #
#----------------------------------------------------------------------#
Require_Message_Authenticator : NO
#----------------------------------------------------------------------#
#       Servers ( Authentication and Accounting )                      #
#                                                                      #
#   Authentication_Ports : This field indicates on which port(s)       #
#                          the authentication server(s) will listen    #
#                          on.  If the field is blank an               #
#                          authentication daemon will not be           #
#                          started.                                    #
#                          The value field may contain more than       #
#                          one value.  Each value is REQUIRED to       #
#                          be separated by a comma ',' .               #
#                                                                      #
#                          The value field must contain a numeric      #
#                          value, like "6666".  In this case a         #
#                          server daemon will listen on "6666".        #
#                                                                      #
#   Accounting_Ports     : The same as authentication_Ports.  See      #
#                          above definitions.                          #
#                                                                      #
# [NOTE] There is no check for port conflicts.  If a server is         #
#        currently running on the specified port the deamon will       #
#        error and not run.  Be sure to check the syslog output        #
#        insure that all servers have started without incident.        #
#                                                                      #
#                                                                      #
```

```
# [Example]                                                              #
#   Authentication_Ports : 1812,6666 (No Space between commas)    #
#                                                                        #
#   In the above example a sever will be start for each port             #
#   specified.  In the case                                              #
#                                                                        #
#           6666 : port 6666                                             #
#                                                                        #
#------------------------------------------------------------------------#
Authentication_Ports  : 1812
Accounting_Ports      : 1813
#------------------------------------------------------------------------#
#       LDAP Directory User Information                                  #
#                                                                        #
#   Required if RADIUS is to connect to a LDAP Version 3 Directory       #
#   and the Database_location field=LDAP                                 #
#                                                                        #
#   LDAP_User     : User ID which has admin permission to connect        #
#                   to the remote (LDAP) database. This is the           #
#                   the LDAP administrator's DN.                         #
#                                                                        #
#   LDAP_User_Pwd : Password associated with the above User Id           #
#                   which is required to authenticate to the LDAP        #
#                   directory.                                           #
#                                                                        #
#------------------------------------------------------------------------#
LDAP_User        : cn=root
LDAP_User_Pwd    :
#------------------------------------------------------------------------#
#       LDAP Directory Information                                       #
#                                                                        #
#   If the Database_location field is set to "LDAP" then the             #
#   following fields need to be completed.                               #
#                                                                        #
#   LDAP_Server_name     : This field specifies the fully qualified#
#                          host name where the LDAP Version 3            #
#                          Server is located.                            #
#   LDAP_Server_Port     : The TCP port number for the LDAP server       #
#             The standard LDAP port is 389.          #
#   LDP_Base_DN          : The distinguished name for search start       #
#   LDAP_Timeout         : # seconds to wait for a response from         #
#                          the LDAP server                               #
#   LDAP_Hoplimit        : maximum number of referrals to follow         #
#                          in a sequence                                 #
#   LDAP_Sizelimit       : size limit (in entries) for search            #
#   LDAP_Debug_level     : 0=OFF 1=Trace ON                              #
#                                                                        #
#------------------------------------------------------------------------#
LDAP_Server_name          :
LDAP_Server_port          :  389
LDAP_Base_DN              :  cn=aixradius
LDAP_Timeout              :  10
LDAP_Hoplimit             :  0
LDAP_Sizelimit            :  0
LDAP_Debug_level          :  0
#------------------------------------------------------------------------#
#       PROXY RADIUS Information                                         #
#                                                                        #
#                                                                        #
#   Proxy_Allow               :  ON or OFF. If ON, then the server       #
#                                   can proxy packets to realms it       #
#                                   knows of and the following           #
#                                   fields must also be configured.      #
#   Proxy_Use_Table           :  ON or OFF. If ON, then the server       #
#                                   can use table for faster             #
#                                   processing of duplicate requests     #
#                                   Can be used without proxy ON, but    #
#                                   it is required to be ON if           #
#                                   Proxy_Use_Table is set to ON.        #
#   Proxy_Realm_name          :  This field specifies the realm          #
#                                   this server services.                #
```

```
#   Proxy_Prefix_delim        :  A list of separators for parsing  #
#                                realm names added as a prefix to  #
#                                the username.  This list must be  #
#                                mutually exclusive to the Suffix  #
#                                delimiters.                       #
#   Proxy_Suffix_delim        :  A list of separators for parsing  #
#                                realm names added as a suffix to  #
#                                the username.  This list must be  #
#                                mutually exclusive to the Prefix  #
#                                delimiters.                       #
#   Proxy_Remove_Hops         :  YES or NO.  If YES then the       #
#                                will remove its realm name, the   #
#                                realm names of any previous hops  #
#                                and the realm name of the next    #
#                                server the packet will proxy to.  #
#                                                                  #
#   Proxy_Retry_count         :  The number of times to attempt    #
#                                to send the request packet.       #
#                                                                  #
#   Proxy_Time_Out            :  The number of seconds to wait     #
#                                in between send attempts.         #
#                                                                  #
#------------------------------------------------------------------#
Proxy_Allow               :   OFF
Proxy_Use_Table           :   OFF
Proxy_Realm_name          :
Proxy_Prefix_delim        :   $/
Proxy_Suffix_delim        :   @.
Proxy_Remove_Hops         :   NO
Proxy_Retry_count         :   2
Proxy_Time_Out            :   3
```

Fichier /etc/radius/clients

Le fichier **clients** contient la liste des clients autorisés à envoyer des requêtes au serveur RADIUS. En général, pour chaque client, NAS ou AP, il faut entrer l'adresse IP du client ainsi que le secret partagé entre le serveur RADIUS et le client.

Le fichier est composé d'entrées au format suivant :

```
<Adresse IP du client>    <Secret partagé>
```

Voici un exemple de liste d'entrées :

```
10.10.10.1          secret1
10.10.10.2          secret2
```

Il est recommandé de définir un secret d'au moins 16 caractères. Le même secret partagé doit être configuré sur le poste client. Vous pouvez modifier le fichier **clients** à l'aide de l'outil SMIT.

Fichier /etc/radius/dictionary

Le fichier **dictionary** contient la description des attributs définis par le protocole RADIUS et pris en charge par le serveur RADIUS AIX. Le démon RADIUS l'utilise pour créer et valider des données des paquets. Les attributs spécifiques au fournisseur doivent également être ajoutés à ce fichier. Le fichier dictionary peut être modifié à l'aide de tout éditeur. Aucune interface SMIT n'est disponible.

Voici un exemple de section de fichier dictionary :

```
#########################################################################
#                                                                       #
#   This file contains dictionary translations for parsing              #
#   requests and generating responses.  All transactions are            #
#   composed of Attribute/Value Pairs.  The value of each attribute     #
#   is specified as one of 4 data types.  Valid data types are:         #
#                                                                       #
#   string - 0-253 octets                                               #
#   ipaddr - 4 octets in network byte order                             #
#   integer - 32 bit value in big endian order (high byte first)        #
#   date - 32 bit value in big endian order - seconds since             #
#                                   00:00:00 GMT,  Jan.  1,  1970       #
#                                                                       #
#   Enumerated values are stored in the user file with dictionary       #
#   VALUE translations for easy administration.                         #
#                                                                       #
#   Example:                                                            #
#                                                                       #
#   ATTRIBUTE         VALUE                                             #
#   ---------------   -----                                             #
#   Framed-Protocol = PPP                                               #
#   7               = 1     (integer encoding)                          #
#                                                                       #
#########################################################################
ATTRIBUTE       User-Name                       1       string
ATTRIBUTE       User-Password                   2       string
ATTRIBUTE       CHAP-Password                   3       string
ATTRIBUTE       NAS-IP-Address                  4       ipaddr
ATTRIBUTE       NAS-Port                        5       integer
ATTRIBUTE       Service-Type                    6       integer
ATTRIBUTE       Framed-Protocol                 7       integer
ATTRIBUTE       Framed-IP-Address               8       ipaddr
ATTRIBUTE       Framed-IP-Netmask               9       ipaddr
ATTRIBUTE       Framed-Routing                  10      integer
ATTRIBUTE       Filter-Id                       11      string
.
.
.
```

Remarques :

1. Tout attribut défini dans le fichier **default.policy** ou **default.auth** (ou dans un fichier spécifique **id_utilisateur.policy** ou **id_utilisateur.auth**), doit correspondre à un attribut RADIUS valide défini dans le fichier de configuration "dictionary" AIX local. Si un attribut est introuvable dans le fichier dictionary, le démon **radiusd** n'est pas chargé et un message d'erreur est consigné.
2. Si vous modifiez le fichier dictionary, le fichier **default.policy** ou **default.auth** du système, relancez les démons RADIUS en lançant les commandes **stopsrc** et **startsrc** ou bien en utilisant l'outil SMIT.

Fichier /etc/radius/proxy

Le fichier **/etc/radius/proxy** est un fichier de configuration qui prend en charge l'utilisation d'un proxy. Ce fichier permet de mapper les domaines connus vers lesquels le serveur proxy peut transférer les paquets. Le fichier **/etc/radius/proxy** utilise l'adresse IP du serveur qui gère les paquets pour ce domaine et le secret partagé entre les deux serveurs.

Le fichier **proxy** contient les zones suivantes : **Realm Name**, **Next Hop IP address**, and **Shared Secret**. Vous pouvez modifier le fichier **proxy** à l'aide de l'outil SMIT.

Le contenu de ce fichier est du type suivant :

```
# @(#)91 1.3  src/rad/usr/sbin/config_files/proxy, radconfig, radius530
1/23/04 13:11:14
#############################################################################
#                                                                           #
#       This file contains a list of proxy realms which are                 #
#       authorized to send/receive proxy requests/responses to/from         #
#       this RADIUS server and their Shared secret used in encryption.#
#                                                                           #
#       The first field is the name of the realm of the remote RADIUS       #
#       Server.                                                             #
#                                                                           #
#       The second field is a valid IP address for the remote RADIUS        #
#       Server.                                                             #
#                                                                           #
#       The third column is the shared secret associated with this          #
#       realm.                                                              #
#                                                                           #
#       NOTE: This file contains sensative security information and         #
#             precautions should be taken to secure access to this          #
#             file.                                                         #
#                                                                           #
#############################################################################
# REALM NAME              REALM IP              SHARED SECRET
#-----------------       -------------------     ----------------------
#  monEnvironmt         10.10.10.10              secretpartag
```

Il est recommandé de définir un secret d'au moins 16 caractères. Le même secret partagé doit être configuré sur le tronçon suivant du serveur RADIUS.

Fichier /var/radius/data/accounting

Lors de la première installation, le fichier **/var/radius/data/accounting** est vide. Les données sont écrites dans ce fichier en fonction des données envoyées par le client dans les paquets ACCOUNTING START et ACCOUNTING STOP. Voici un exemple de type de données écrites par le serveur RADIUS AIX dans le fichier **/var/radius/data/accounting**. Les données de votre fichier peuvent être différentes, selon votre configuration système.

```
Thu May 27 14:43:19 2004
         NAS-IP-Address = 10.10.10.1
         NAS-Port = 1
         NAS-Port-Type = Async
         User-Name = "rod"
         Acct-Status-Type = Start
         Acct-Authentic = RADIUS
         Service-Type = Framed-User
         Acct-Session-Id = "0000000C"
         Framed-Protocol = PPP
         Acct-Delay-Time = 0
         Timestamp = 1085686999

Thu May 27 14:45:19 2004
         NAS-IP-Address = 10.10.10.1
         NAS-Port = 1    <-- l'utilisateur "rod" était connecté au port
physique n 1
         NAS-Port-Type = Async
         User-Name = "rod"
         Acct-Status-Type = Stop
         Acct-Authentic = RADIUS
         Service-Type = Framed-User
         Acct-Session-Id = "0000000C"   <-- notez que les ID de session
sont identiques ce qui permet de faire correspondre les démarrages et les
arrêts
         Framed-Protocol = PPP
         Framed-IP-Address = 10.10.10.2   <-- adresse IP de l'utilisateur
rod
         Acct-Terminate-Cause = User-Request  <-- l'utilisateur a annulé la
session
         Acct-Input-Octets = 4016
         Acct-Output-Octets = 142
         Acct-Input-Packets = 35
         Acct-Output-Packets = 7
         Acct-Session-Time = 120  <--- secondes
         Acct-Delay-Time = 0
         Timestamp = 1085687119  <--- notez que l'utilisateur "rod" a été
connecté durant 120 secondes (2 minutes)
```

Configuration du serveur LDAP RADIUS

Lorsque l'authentification de l'utilisateur LDAP est configurée, le schéma du serveur LDAP doit être mis à jour. Avant de définir les utilisateurs LDAP RADIUS, l'administrateur système LDAP doit ajouter les attributs et les classes d'objets définies pour le serveur RADIUS AIX.

Vous devez ajouter un suffixe au serveur LDAP. Le suffixe correspondant au serveur RADIUS est nommé `cn=aixradius`. Un suffixe est un nom spécifique identifiant la première entrée d'une hiérarchie de répertoires.

Lorsqu'un suffixe est ajouté, un conteneur vide est ajouté au répertoire LDAP. Un *conteneur* est une entrée vide qui permet de partitionner l'espace de nom. Un conteneur est similaire à un répertoire de système de fichiers, qui peut contenir des entrées de répertoire. Vous pouvez ajouter des données de profil utilisateur au répertoire LDAP à l'aide de l'outil SMIT. L'ID et le mot de passe de l'administrateur LDAP sont stockés dans le fichier **/etc/radius/radiusd.conf** et peuvent être configurés à l'aide de l'outil SMIT sur un serveur RADIUS.

Les données stockées dans les entrées du répertoires LDAP sont organisées via la définition de classes d'objets dans le schéma. Une classe d'objets est un ensemble d'attributs obligatoires et facultatifs. Les attributs sont définis sous la forme `type=valeur`, où le type est défini par un identifiant d'objet unique (OID) et la valeur doit être indiquée selon une syntaxe prédéfinie. Chaque entrée du répertoire LDAP est l'instance d'un objet.

Remarque : La classe d'objets en elle–même ne définit aucune arborescence de répertoires ni aucun espace de nom. Cela se produit uniquement lorsque des entrées sont créés et qu'un nom spécifique unique est attribué à une instance particulière de classe d'objets. Par exemple, si un nom spécifique (DN) unique est attribué à une classe d'objets de conteneur, celle–ci peut être associée à deux autres entrées qui sont des instances de l'unité d'organisation de la classe d'objets. Il en résulte une structure en arborescence ou un espace de nom.

Les classes d'objets sont spécifiques au serveur RADIUS et sont appliquées à partir d'un fichier **ldif**. Certains attributs proviennent du schéma LDAP et d'autres sont spécifiques à RADIUS. Les classes d'objets RADIUS sont structurelles, par opposition aux classes d'objets de type modèle ou résumé.

Pour des raisons de sécurité, les liens vers le serveur LDAP utilisent l'appel API de lien simple ou SASL, **ldap_bind_s** qui inclut le nom DN, la méthode d'authentification CRAM–MD5 et le mot de passe administrateur LDAP. Ce sont des prétraitements de message qui sont transmis sur le réseau et non les mots de passe eux–mêmes. CRAM–MD5 est un mécanisme de sécurité qui ne nécessite aucune configuration spéciale de chaque côté (client ou serveur).

Remarque : Tous les attributs des classes d'objets comportent une seule valeur.

Les fichiers de schéma LDAP suivants se trouvent dans le répertoire **/etc/radius/ldap**.
V3.radiusbase.schema.ldif
V3.radius.schema.ldif

Le fichier **V3.radiusbase.schema.ldif** définit la classe d'objets prédéfinie du plus haut niveau pour le serveur RADIUS, qui est `cn=aixradius`. Il crée également les branches suivantes dans les classes d'objets `cn=aixradius` :

```
1) ou=ibm-radiususer
 2) ou=ibm-radiusactiveusers
```

Vous pouvez ajouter les données nécessaires à l'aide de la commande suivante :

```
ldapadd –D ldap_admin_id –w password –i
/etc/radius/ldap/IBM.V3.radiusbase.schema.ldif
```

Lancez cette commande sur le système du serveur LDAP, ou lancez–la à distance en ajoutant l'option **–h** (nom du système hôte).

Le fichier **V3.radius.schema.ldif** définit les attributs et les classes d'objets spécifiques à RADIUS. Pour ajouter les nouveaux attributs et les nouvelles classes d'objets RADIUS, entrez la commande suivante :

```
ldapmodify –D ldap_admin_id –w password –i
/etc/radius/ldap/IBM.V3.radius.schema.ldif
```

Indiquez également que `LDAP` est l'emplacement de la base de données sur l'interface SMIT et entrez le nom du serveur LDAP et le mot de passe administrateur. En procédant ainsi, vous pouvez ensuite ajouter des utilisateurs LDAP RADIUS au répertoire à l'aide de l'outil SMIT.

Présentation de l’espace de nom LDAP RADIUS

En haut de la hiérarchie se trouve le conteneur `cn=aixradius`. Sous `cn=aixradius` se trouvent deux unités d’organisation. Une unité d’organisation est un conteneur qui permet d’avoir des entrées uniques.

Schéma LDAP RADIUS

Le graphique suivant représente le schéma LDAP RADIUS.

Figure 19. Espace de nom LDAP RADIUS

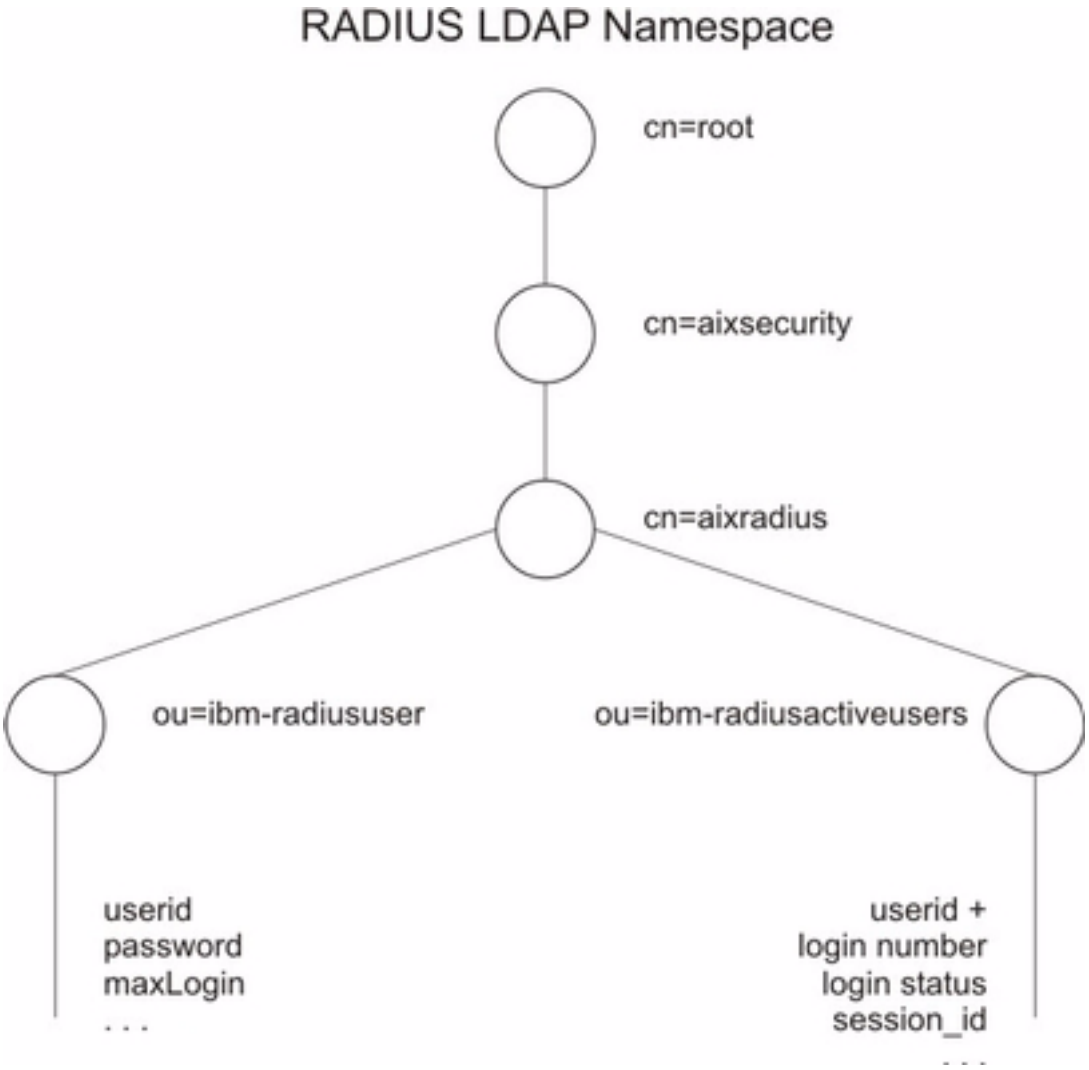


Figure 1. Sur cette figure, les conteneurs et les unités d’organisation sont représentés par des cercles et sont reliés par des lignes ou des branches. Le conteneur aixradius, au centre, est relié à deux unités d’organisation : ibm–radiususer et ibm–radiusactiveusers. Sous le conteneur ibm–radiususer se trouvent les conteneurs userid, password et maxLogin. Sous le conteneur ibmradiusactiveusers se trouvent les conteneurs userid +, login number, login status et session_id. Le conteneur aixsecurity se trouve au–dessus du conteneur aixradius, et tout en haut se trouve le conteneur root.

Classe d'objets de profil utilisateur

Le serveur RADIUS peut authentifier un utilisateur si au préalable un profil utilisateur a été défini sur le système. Un profil contient l'ID et le mot de passe de l'utilisateur. Un objet profil utilisateur fournit les données relatives à un utilisateur particulier ayant accès au réseau et contient les données d'authentification. L'accès à la classe d'objets **ibm–radiususer** est effectué de façon synchrone via des appels API LDAP du démon. La zone unique, qui est le début du nom DN est l'ID utilisateur. La zone **MaxLoginCount** limite le nombre de connexions de l'utilisateur LDAP.

Classe d'objets de liste des connexions actives

La liste des connexions actives contient des informations sur les utilisateurs connectés. Un utilisateur est associé à plusieurs enregistrements, le premier étant **login_number = 1** et le dernier étant le nombre de maximal de connexions **MaxLoginCount** qui est 5. L'ID de session provient du message RADIUS **start_accounting**. Des enregistrements partiellement remplis sont créés lorsqu'un objet **ibm–radiususer** est créé. Cela signifie que la plupart des zones sont vides avant la réception de paquets de comptabilité RADIUS. Après la réception du message **start_accounting** de RADIUS, l'objet **ibm–radiusactiveusers** est mis à jour de façon à indiquer que l'utilisateur est connecté et que les données de la session unique sont écrites avec le numéro de connexion correspondant. Après la réception du message **stop_accounting**, les données de l'enregistrement de la liste de connexions actives sont effacées. L'enregistrement de la connexion active est mis à jour de façon à indiquer que l'utilisateur s'est déconnecté. Les messages de début et de fin de comptabilité contiennent le même numéro de session, qui est un numéro unique. L'accès à la classe d'objets est effectué de façon synchrone avec les appels API LDAP.

Expiration du mot de passe

Si la fonction d'expiration du mot de passe est activée, le serveur RADIUS peut :

1. recevoir une notification à l'expiration d'un mot de passe,
2. mettre à jour le mot de passe de l'utilisateur via le protocole RADIUS.

L'expiration des mots de passe implique la prise en charge de quatre types de paquet supplémentaires et d'un nouvel attribut. Les nouveaux types de paquet sont fournis dans le dictionnaire AIX et la fonction d'expiration des mots de passe doit être activée.

Il peut être préférable d'interdire à RADIUS de mettre à jour les mots de passe arrivés à expiration sur certaines installations RADIUS. Une entrée du fichier **radiusd.conf** permet d'autoriser ou d'interdire au serveur RADIUS la modification des mots de passe arrivés à expiration. Par défaut l'interdiction est sélectionnée. Vous pouvez ajouter un message de réponse **Password_Expired_Reply_Message** destiné à l'utilisateur, qui sera renvoyé dans le paquet **password–expired**. Les attributs de l'ancien et du nouveau mot de passe doivent être chiffrés et déchiffrés par la méthode PAP.

Attributs spécifiques au fournisseur (VSA)

Les attributs spécifiques au fournisseur (VSA) sont définis par chaque fournisseur de serveur à accès distant, en général le fournisseur du matériel, afin de personnaliser le fonctionnement de RADIUS sur leur serveur. Les attributs spécifiques au fournisseur sont nécessaires si vous souhaitez accorder aux utilisateurs plusieurs types d'accès. Il est possible de combiner les attributs spécifiques au fournisseur avec les attributs définis pour RADIUS.

Ces attributs spécifiques au fournisseur sont facultatifs, mais si le fonctionnement du matériel NAS nécessite la configuration d'attributs supplémentaires, vous devez ajouter des attributs spécifiques au fournisseur dans le fichier dictionary.

Les attributs spécifiques au fournisseur peuvent être utilisés pour accorder des autorisations supplémentaires. Vous pouvez accorder des autorisations en combinant les attributs VSA à **User–Name** et **Password**. Côté serveur, le fichier de stratégie d'autorisations contient la liste des attributs à vérifier dans le paquet Access–Request pour un utilisateur particulier. Si le paquet ne contient pas les attributs répertoriés dans le fichier des utilisateurs, un paquet `access_reject` est renvoyé au NAS. Les attributs spécifiques au fournisseur peuvent également servir en tant que liste d'entrées attribut=valeur dans le fichier ***id_utilisateur*.policy**.

Voici un exemple de section relative aux attributs spécifiques au fournisseur, dans le fichier dictionary :

```
#########################################################################
#                                                                       #
#   This section contains examples of dictionary translations for       #
#   parsing vendor specific attributes (vsa). The example below is for  #
#   the vendor Cisco. Before defining an Attribute/Value pair for a     #
#   vendor a "VENDOR" definition is needed.                             #
#                                                                       #
#   Example:                                                            #
#                                                                       #
#   VENDOR          Cisco           9                                   #
#                                                                       #
#   VENDOR:  This specifies that the Attributes after this entry are    #
#            specific to Cisco.                                         #
#   Cisco :  Denotes the Vendor name                                    #
#   9     :  Vendor Id defined in the "Assigned Numbers" RFC            #
#                                                                       #
#########################################################################

#VENDOR          Cisco           9

#ATTRIBUTE       Cisco-AVPair              1       string
#ATTRIBUTE       Cisco-NAS-Port            2       string
#ATTRIBUTE       Cisco-Disconnect-Cause    195     integer
#
#----------------Cisco-Disconnect-Cause---------------------------------#
#
#VALUE           Cisco-Disconnect-Cause    Unknown                 2
#VALUE           Cisco-Disconnect-Cause    CLID-Authentication-Failure 4
#VALUE           Cisco-Disconnect-Cause    No-Carrier              10
#VALUE           Cisco-Disconnect-Cause    Lost-Carrier            11
#VALUE           Cisco-Disconnect-Cause    No-Detected-Result-Codes    12
#VALUE           Cisco-Disconnect-Cause    User-Ends-Session       20
#VALUE           Cisco-Disconnect-Cause    Idle-Timeout            21
#VALUE           Cisco-Disconnect-Cause    Exit-Telnet-Session     22
#VALUE           Cisco-Disconnect-Cause    No-Remote-IP-Addr       23
```

RADIUS Reply–Message Support

Vous pouvez définir et configurer des messages de réponse à l'aide des attributs Reply–Message du fichier **radiusd.conf**. Il sont destinés à être envoyés par NAS ou AP à l'utilisateur. Ces messages peuvent indiquer un succès, un échec ou un challenge. Ils sont constitués de texte lisible et leur contenu dépend de l'installation et ils sont configurés lors de la configuration du serveur. Par défaut, ces attributs ne sont associés à aucun texte. Vous pouvez configurer tous les attributs ou bien aucun, un, deux ou trois attributs.

RADIUS prend en charge les attributs de réponse Reply–Message suivants :

- Accept Reply–Message
- Reject Reply–Message
- CHAP Reply–Message
- Password Expired Reply–Message

Ces attributs sont ajoutés au fichier de configuration **radiusd.conf** et lus dans une structure de configuration globale au démarrage du démon. Pour définir ces valeurs, utilisez l'option **Configure Server** des écrans SMIT RADIUS. Une chaîne peut contenir au maximum 256 octets de caractères.

La fonction est mise en oeuvre de la manière suivante :

1. Lorsqu'il démarre, le démon **radiusd** lit le fichier **radiusd.conf** et définit les attributs Reply–Message.
2. A réception d'un paquet `access request`, l'utilisateur est authentifié.
3. Si la réponse de l'authentification est `access accept`, le texte de l'attribut Accept Reply–Message est vérifié. Si le texte est présent, la chaîne est renvoyée dans le paquet `access accept`.
4. Si l'authentification est refusée, le texte de l'attribut Reject Reply–Message est vérifié et renvoyé dans le paquet `access reject`.
5. Si l'authentification est demandée, alors l'attribut CHAP Reply–Message est vérifié et envoyé dans le paquet `Access-Challenge`.

Ecrans SMIT RADIUS

Lorsque vous utilisez l'outil SMIT pour configurer le serveur RADIUS, certaines zones sont obligatoires, elles sont signalées par un astérisque (*). Le raccourci permettant d'accéder à SMIT est :

```
smitty radius
```

Le menu principal de RADIUS est le suivant :

```
      RADIUS Server

Configure Server
Configure Clients
Configure Users
Configure Proxy Rules
Advanced Server Configuration
Start RADIUS Server daemons
Stop RADIUS Server daemons
```

Voici un exemple d'écran SMIT de configuration du serveur RADIUS :

```
          Configure Server

 RADIUS Directory            /etc/radius
 *Database Location           [UNIX]
 Local AVL Database File Name [dbdata.bin]
 Local Accounting             [ON]

 Debug Level                  [3]
 Accept Reply-Message         []
 Reject Reply-Message         []
 Challenge Reply-Message      []
 Password Expired Reply Message []
 Support Renewal of Expired Passwords [NO]
 Require Message Authenticator  [NO]

 *Authentication Port Number  [1812]
 *Accounting Port Number      [1813]

 LDAP Server Name             []
 LDAP Server Port Number      [389]
 LDAP Server Admin Distinguished Name     []
 LDAP Server Admin Password         []
 LDAP Base Distinguished Name       [cn=aixradius]
 LDAP Size Limit                    [0]
 LDAP Hop Limit                     [0]
 LDAP wait time limit               [10]
 LDAP debug level                   [ 0]

 Proxy Allowed                      [OFF]
 Proxy Use table                    [OFF]
 Proxy Realm Name                   []
 Proxy Prefix Delimiters            [$/]
 Proxy Suffix Delimiters            [@.]
                                         NOTE: prefix & suffix are mutually
exclusive
 Proxy Remove Hops                  [NO]
 Proxy Retry Count                  [2]
 Proxy Timeout                      [30]
```

La touche F1 permet d'obtenir des informations d'aide détaillées sur les menus et les options de l'outil SMIT.

Générateur de numéros aléatoires

Pour générer la zone Authenticator d'un paquet RADIUS, il est nécessaire d'obtenir des numéros aléatoires. Il est important que le générateur soit le plus efficace possible, car un intrus pourrait essayer d'obtenir du serveur RADIUS une réponse à une demande et ensuite utiliser cette réponse pour se faire passer pour le serveur RADIUS. Le serveur RADIUS AIX génère des numéros pseudo–aléatoires à l'aide de l'extension de noyau **/dev/urandom**. Cette extension de noyau collecte des exemples entropiques de sources matérielles via le pseudo pilote d'unité. Le caractère aléatoire de cette unité a été testé par NIST.

Utilitaires de journalisation

Le serveur RADIUS consigne les données d'activité et d'erreur dans le journal SYSLOG.

Il existe trois niveaux de journalisation :

0 Consignation des incidents, des erreurs et du démarrage des démons.

3 Consignation d'un suivi d'audit de messages `access_accept`, `access_reject`, * `discard` et `error`.

9 Journalisation de données de niveau 0 et 3, ainsi que d'autres informations. Le niveau 9 est recommandé uniquement dans un but de débogage.

* Les messages de type `discard` sont consignés lorsqu'un paquet entrant est invalide et qu'aucun message de réponse n'est généré.

Sortie SYSLOG pour journalisation d'audit

Le niveau de journalisation par défaut est 3. Ce niveau permet d'améliorer le niveau d'audit du serveur RADIUS. En fonction du niveau de journalisation du serveur, vous pouvez utiliser les activités stockées dans le journal afin de vérifier des formes d'activités suspectes. En cas de violation de la sécurité, la sortie SYSLOG permet de déterminer comment et quand s'est produite la violation et éventuellement l'ampleur de l'accès. Ces données peuvent permettre de renforcer les mesures de sécurité et de prévenir les incidents potentiels.

Pour plus d'informations sur la configuration de RADIUS de façon à utiliser le démon **syslogd**, voir Fonctions de journalisation, page 12-50.

Description d'une sortie SYSLOG

Le niveau de débogage (0, 3 ou 9) est défini dans la zone Debug_Level du fichier **radiusd.conf**. La valeur par défaut est 3. Voici un exemple de section de débogage du fichier **radiusd.conf** :

```
#.
#.
#.
#   Debug_Level      :  This pair sets the debug level at which    #
#                       the RADIUS server will run.  Appropriate   #
#                       values are 0,3 or 9. The default is 3.     #
#                       Output is directed to location specified   #
#                       by *.debug stanza in /etc/syslog.conf      #
#                                                                  #
#                       Each level increases the amount of messages#
#                       sent to syslog. For example "9" includes   #
#                       the new messages provided by "9" as well   #
#                       as all messages generated by level 0 and 3.#
#                                                                  #
#                       0 : provides the minimal output to the     #
#                           syslogd log.  It sends start up        #
#                           and end messages for each RADIUS       #
#                           process.  It also logs error           #
#                           conditions.                            #
#                                                                  #
#                       3 : includes general ACCESS ACCEPT, REJECT #
#                           and DISCARD messages for each packet.  #
#                           This level provides a general audit    #
#                           trail for authentication.              #
#                                                                  #
#                       9 : Maximum amount of log data. Specific   #
#                           values of attributes while a           #
#                           transaction is passing thru            #
#                           processing and more.                   #
#                           [NOT advised under normal operations]  #
#                                                                  #
#------------------------------------------------------------------#
```

Voici des exemples de sortie SYSLOG (debug) pour chacun des 3 niveaux de débogage.

Paquet de comptabilité avec niveau de débogage 3

```
Aug 18 10:23:57 server1 syslog: [0]:Monitor process [389288] has started
Aug 18 10:23:57 server1 radiusd[389288]: [0]:Local database (AVL) built.
Aug 18 10:23:57 server1 radiusd[389288]: [0]:Authentication process started
: Pid= 549082 Port = 1812
Aug 18 10:23:57 server1 radiusd[389288]: [0]:Accounting process started :
Pid= 643188 Port = 1813
Aug 18 10:23:57 server1 radiusd[643188]: [0]:Socket created [15]
Aug 18 10:23:57 server1 radiusd[643188]: [0]:Bound Accounting socket [15]
Aug 18 10:23:57 server1 radiusd[549082]: [0]:Socket created [15]
Aug 18 10:23:57 server1 radiusd[549082]: [0]:Bound Authentication socket
[15]
Aug 18 10:24:07 server1 radiusd[643188]: [1]:*** Start Process_Packet() ***
Aug 18 10:24:07 server1 radiusd[643188]: [1]:Code 4, ID = 96, Port = 41639
Host = 10.10.10.10
Aug 18 10:24:07 server1 radiusd[643188]: [1]:ACCOUNTING-START - sending
Accounting Ack to User [ user_id1 ]
Aug 18 10:24:07 server1 radiusd[643188]: [1]:Sending Accounting Ack of id
96 to 10.10.10.10 (client1.ibm.com)
Aug 18 10:24:07 server1 radiusd[643188]: [1]:send_acct_reply() Outgoing
Packet:
Aug 18 10:24:07 server1 radiusd[643188]: [1]: Code = 5, Id = 96, Length =
20
Aug 18 10:24:07 server1 radiusd[643188]: [1]:*** Leave Process_Packet() ***
Aug 18 10:24:13 server1 radiusd[643188]: [2]:*** Start Process_Packet() ***
Aug 18 10:24:13 server1 radiusd[643188]: [2]:Code 4, ID = 97, Port = 41639
Host = 10.10.10.10
Aug 18 10:24:13 server1 radiusd[643188]: [2]:ACCOUNTING-STOP - sending
Accounting Ack to User [ user_id1 ]
Aug 18 10:24:14 server1 radiusd[643188]: [2]:Sending Accounting Ack of id
97 to 10.10.10.10 (client1.ibm.com)
Aug 18 10:24:14 server1 radiusd[643188]: [2]:send_acct_reply() Outgoing
Packet:
Aug 18 10:24:14 server1 radiusd[643188]: [2]: Code = 5, Id = 97, Length =
20
Aug 18 10:24:14 server1 radiusd[643188]: [2]:*** Leave Process_Packet() **
```

Paquets de comptabilité au niveau 9

```
Aug 18 10:21:18 server1 syslog: [0]:Monitor process [643170] has started
Aug 18 10:21:18 server1 radiusd[643170]: [0]:Local database (AVL) built.
Aug 18 10:21:18 server1 radiusd[643170]: [0]:Authentication process started
: Pid= 389284 Port = 1812
Aug 18 10:21:18 server1 radiusd[643170]: [0]:Accounting process started :
Pid= 549078 Port = 1813
Aug 18 10:22:03 server1 radiusd[643170]: [0]:PID = [389284] dead
Aug 18 10:22:03 server1 radiusd[643170]: [0]:PID = [549078] dead
Aug 18 10:22:03 server1 radiusd[643170]: [0]:All child processes stopped.
radiusd parent stopping
Aug 18 10:22:09 server1 syslog: [0]:Monitor process [1081472] has started
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:Local database (AVL) built.
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:Inside client_init()
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:Number of client entries
read: 1
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:Inside read_authorize_policy
routine for file: /etc/radius/authorization/default.policy.
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:Inside read_authorize_file
routine for file: /etc/radius/authorization/default.policy.
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:read_authorize_file() routine
complete.
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:Inside read_authorize_file
routine for file: /etc/radius/authorization/default.auth.
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:read_authorize_file() routine
complete.
Aug 18 10:22:09 server1 radiusd[549080]:
[0]:connect_to_LDAP_server:Database Location (where the data resides)=LDAP.
Aug 18 10:22:09 server1 radiusd[549080]: [0]:connect_to_LDAP_server:LDAP
Server name=server1.austin.ibm.com.
Aug 18 10:22:09 server1 radiusd[549080]: [0]:connect_to_LDAP_server:LDAP
Server port= 389.
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:Authentication process
started : Pid= 549080 Port = 1812
Aug 18 10:22:09 server1 radiusd[389286]:
```

```
[0]:connect_to_LDAP_server:Database Location (where the data resides)=LDAP.
Aug 18 10:22:09 server1 radiusd[389286]: [0]:connect_to_LDAP_server:LDAP
Server name=server1.austin.ibm.com.
Aug 18 10:22:09 server1 radiusd[389286]: [0]:connect_to_LDAP_server:LDAP
Server port= 389.
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:Accounting process started :
Pid= 389286 Port = 1813
Aug 18 10:22:10 server1 radiusd[549080]: [0]:Socket created [15]
Aug 18 10:22:10 server1 radiusd[549080]: [0]:Bound Authentication socket
[15]
Aug 18 10:22:10 server1 radiusd[389286]: [0]:Socket created [15]
Aug 18 10:22:10 server1 radiusd[389286]: [0]:Bound Accounting socket [15]
Aug 18 10:22:15 server1 radiusd[389286]: [1]:*** Start Process_Packet() ***
Aug 18 10:22:15 server1 radiusd[389286]: [1]:Incoming Packet:
Aug 18 10:22:15 server1 radiusd[389286]: [1]:  Code = 4, Id = 94, Length =
80
Aug 18 10:22:15 server1 radiusd[389286]: [1]:  Authenticator =
0xC5DBDDFE6EFFFDBD6AE64CA35947DD0F
Aug 18 10:22:15 server1 radiusd[389286]: [1]:    Type =  40, Length =   6,
Value = 0x00000001
Aug 18 10:22:15 server1 radiusd[389286]: [1]:    Type =   1, Length =   8,
Value = 0x67656E747931
Aug 18 10:22:15 server1 radiusd[389286]: [1]:    Type =   4, Length =   6,
Value = 0x00000000
Aug 18 10:22:15 server1 radiusd[389286]: [1]:    Type =   8, Length =   6,
Value = 0x0A0A0A01
Aug 18 10:22:15 server1 radiusd[389286]: [1]:    Type =  44, Length =   8,
Value = 0x303030303062
Aug 18 10:22:15 server1 radiusd[389286]: [1]:    Type =  30, Length =  10,
Value = 0x3132332D34353638
Aug 18 10:22:15 server1 radiusd[389286]: [1]:    Type =  31, Length =  10,
Value = 0x3435362D31323335
Aug 18 10:22:15 server1 radiusd[389286]: [1]:    Type =  85, Length =   6,
Value = 0x00000259
Aug 18 10:22:15 server1 radiusd[389286]: [1]:Starting parse_packet()
Aug 18 10:22:15 server1 radiusd[389286]: [1]:Code 4, ID = 94, Port = 41639
Host = 10.10.10.10
Aug 18 10:22:15 server1 radiusd[389286]: [1]:Acct-Status-Type = Sta
```

Paquet d'authentification au niveau 0 :

```
Aug 18 10:06:11 server1 syslog: [0]:Monitor process [1081460] has started
Aug 18 10:06:11 server1 radiusd[1081460]: [0]:Local database (AVL) built.
Aug 18 10:06:11 server1 radiusd[1081460]: [0]:Authentication process
started : Pid= 549076 Port = 1812
Aug 18 10:06:11 server1 radiusd[1081460]: [0]:Accounting process started :
Pid= 389282 Port = 18
```

Paquet d'authentification au niveau 3 :

```
Aug 18 10:01:32 server2 radiusd[389276]: [3]:*** Start Process_Packet() ***
Aug 18 10:01:32 server2 radiusd[389276]: [3]:Code 1, ID = 72, Port = 41638
Host = 10.10.10.10
Aug 18 10:01:32 server2 radiusd[389276]: [3]:authenticate_password_PAP:
Passwords do not match, user is rejected
Aug 18 10:01:32 server2 radiusd[389276]: [3]:Authentication failed for user
[user_id1] using IP [10.10.10.10]
Aug 18 10:01:32 server2 radiusd[389276]: [3]:ACCESS-REJECT - sending reject
for id 72 to 10.10.10.10 (client1.ibm.com)
Aug 18 10:01:32 server2 radiusd[389276]: [3]:send_reject() Outgoing Packet:
Aug 18 10:01:32 server2 radiusd[389276]: [3]: Code = 3, Id = 72, Length =
30
Aug 18 10:01:32 server2 radiusd[389276]: [3]:*** Leave Process_Packet() ***
Aug 18 10:01:53 server2 radiusd[389276]: [4]:*** Start Process_Packet() ***
Aug 18 10:01:53 server2 radiusd[389276]: [4]:Code 1, ID = 74, Port = 41638
Host = 10.10.10.10
Aug 18 10:01:53 server2 radiusd[389276]: [4]:authenticate_password_PAP:
Passwords Match, user is authenticated
Aug 18 10:01:53 server2 radiusd[389276]: [4]:Authentication successful for
user [user_id1] using IP [10.10.10.10]
Aug 18 10:01:53 server2 radiusd[389276]: [4]:Authorization successful for
user [user_id1] using IP [10.10.10.10]
Aug 18 10:01:53 server2 radiusd[389276]: [4]:ACCESS-ACCEPT - sending accept
for id 74 to 10.10.10.10 (client1.ibm.com)
Aug 18 10:01:53 server2 radiusd[389276]: [4]:send_accept() Outgoing Packet:
Aug 18 10:01:53 server2 radiusd[389276]: [4]: Code = 2, Id = 74, Length =
31
Aug 18 10:01:53 server2 radiusd[389276]: [4]:*** Leave Process_Packet() **
```

Paquet d'authentification au niveau 9 :

```
Aug 18 10:03:56 server1 radiusd[389278]: [1]:*** Start Process_Packet() ***
Aug 18 10:03:56 server1 radiusd[389278]: [1]:*** Start Process_Packet() ***
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Incoming Packet:
Aug 18 10:03:56 server1 radiusd[389278]: [1]:  Code = 1, Id = 77, Length =
58
Aug 18 10:03:56 server1 radiusd[389278]: [1]:  Authenticator =
0xE6CB0F9C22BB4E799854E734104FB2D5
Aug 18 10:03:56 server1 radiusd[389278]: [1]:    Type =   1, Length =   8,
Value = 0x67656E747931
Aug 18 10:03:56 server1 radiusd[389278]: [1]:    Type =   4, Length =   6,
Value = 0x00000000
Aug 18 10:03:56 server1 radiusd[389278]: [1]:    Type =   2, Length =  18,
Value = 0x**********
**********************
Aug 18 10:03:56 server1 radiusd[389278]: [1]:    Type =  7, Length =   6,
Value = 0x00000001
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Starting parse_packet()
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Code 1, ID = 77, Port = 41638
Host = 10.10.10.10
Aug 18 10:03:56 server1 radiusd[389278]: [1]:User-Name = "user_id1"
Aug 18 10:03:56 server1 radiusd[389278]: [1]:NAS-IP-Address = 10.10.10.10
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Framed-Protocol = PPP
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Leaving parse_packet()
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Verifying
Message-Authenticator
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Message-Authenticator
successfully verified
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside proxy_request_needed()
function
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Proxy is not turned on
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Username = [user_id1]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Client IP = [10.10.10.10]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside parse_for_login(
user_id1 )
Aug 18 10:03:56 server1 radiusd[389278]: [1]:User_id remaining after prefix
removal = [user_id1]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:User_id remaining after suffix
removal = [user_id1]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside rad_authenticate()
```

```
function
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Authentication request
received for [client1.austin.ibm.com]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Calling get_ldap_user() to get
LDAP user data
Aug 18 10:03:56 server1 radiusd[389278]: [1]:get_ldap_user:LDAP user id:
user_id1.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:get_ldap_user:LDAP
max_login_cnt:2.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:get_ldap_user:LDAP EAP_type:
4.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:get_ldap_user:LDAP
passwordexpiredweeks: 9.
Aug 18 10:03:56 server1 radiusd[389278]:
[1]:get_ldap_active_sessions:number of free entries= 2.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:get_ldap_active_session:dn
retrieved=
radiusuniqueidentifier=user_id11,ou=radiusActiveUsers,cn=aixradius.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside get_client_secret
routine for ip:10.10.10.10
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Found NAS-IP = [10.10.10.10]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Found shared secret.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:authenticate_password_PAP:
Passwords Match, user is authenticated
Aug 18 10:03:56 server1 radiusd[389278]: [1]:is_ldap_pw:password for user
has NOT expired
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Authentication successful for
user [user_id1] using IP [10.10.10.10]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside rad_authorize()
routine.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside read_authorize_policy
routine for file: /etc/radius/authorization/user_id1.policy.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside read_authorize_file
routine for file: /etc/radius/authorization/user_id1.policy.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Did not open
/etc/radius/authorization/user_id1.policy file. File may not be found.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Error reading policy file:
/etc/radius/authorization/user_id1.policy.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:rad_authorize:default policy
list and userpolicy list were empty.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:In create_def_copy() routine.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Successfully made a copy of
the master authorization list.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside read_authorize_file
routine for file: /etc/radius/authorization/user_id1.auth.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Did not open
/etc/radius/authorization/user_id1.auth file. File may not be found.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:rad_authorize:copy
authorization list and user list were empty.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Authorization successful for
user [user_id1] using IP [10.10.10.10]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:ACCESS-ACCEPT - sending accept
for id 77 to 10.10.10.10 (client1.austin.ibm.com)
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside proxy_response_needed()
function
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Proxy is not turned on
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside get_client_secret
routine for ip:10.10.10.10
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Found NAS-IP = [10.10.10.10]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Found shared secret.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:send_accept() Outgoing Packet:
Aug 18 10:03:56 server1 radiusd[389278]: [1]: Code = 2, Id = 77, Length =
31
Aug 18 10:03:56 server1 radiusd[389278]: [1]:send_accept() Outgoing Packet:
Aug 18 10:03:56 server1 radiusd[389278]: [1]:  Code = 2, Id = 77, Length =
31
Aug 18 10:03:56 server1 radiusd[389278]: [1]:  Authenticator =
0xCCB2B645BBEE86F5E4FC5BE24E904B2A
Aug 18 10:03:56 server1 radiusd[389278]: [1]:    Type =  18, Length =  11,
Value = 0x476F6F646E65737321
Aug 18 10:03:56 server1 radiusd[389278]: [1]:*** Leave Process_Packet() ***
```

```
Aug 18 10:04:18 server1 radiusd[389278]: [2]:*** Start Process_Packet() ***
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Incoming Packet:
Aug 18 10:04:18 server1 radiusd[389278]: [2]:  Code = 1, Id = 79, Length =
58
Aug 18 10:04:18 server1 radiusd[389278]: [2]:  Authenticator =
0x774298A2B6DD90D7C33B3C10C4787D41
Aug 18 10:04:18 server1 radiusd[389278]: [2]:    Type =   1, Length =   8,
Value = 0x67656E747931
Aug 18 10:04:18 server1 radiusd[389278]: [2]:    Type =   4, Length =   6,
Value = 0x00000000
Aug 18 10:04:18 server1 radiusd[389278]: [2]:    Type =   2, Length =  18,
Value = 0x**********************************
Aug 18 10:04:18 server1 radiusd[389278]: [2]:    Type =  7, Length =   6,
Value = 0x00000001
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Starting parse_packet()
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Code 1, ID = 79, Port = 41638
Host = 10.10.10.10
Aug 18 10:04:18 server1 radiusd[389278]: [2]:User-Name = "user_id1"
Aug 18 10:04:18 server1 radiusd[389278]: [2]:NAS-IP-Address = 10.10.10.10
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Framed-Protocol = PPP
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Leaving parse_packet()
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Verifying
Message-Authenticator
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Message-Authenticator
successfully verified
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Inside proxy_request_needed()
function
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Proxy is not turned on
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Username = [user_id1]
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Client IP = [10.10.10.10]
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Inside parse_for_login(
user_id1 )
Aug 18 10:04:18 server1 radiusd[389278]: [2]:User_id remaining after prefix
removal = [user_id1]
Aug 18 10:04:18 server1 radiusd[389278]: [2]:User_id remaining after suffix
removal = [user_id1]
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Inside rad_authenticate()
function
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Authentication request
received for [client1.austin.ibm.com]
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Calling get_ldap_user() to get
LDAP user data
Aug 18 10:04:18 server1 radiusd[389278]: [2]:get_ldap_user:LDAP user id:
user_id1.
Aug 18 10:04:18 server1 radiusd[389278]: [2]:get_ldap_user:LDAP
max_login_cnt:2.
Aug 18 10:04:18 server1 radiusd[389278]: [2]:get_ldap_user:LDAP EAP_type:
4.
Aug 18 10:04:18 server1 radiusd[389278]: [2]:get_ldap_user:LDAP
passwordexpiredweeks: 9.
Aug 18 10:04:18 server1 radiusd[389278]:
[2]:get_ldap_active_sessions:number of free entries= 2.
Aug 18 10:04:18 server1 radiusd[389278]: [2]:get_ldap_active_session:dn
retrieved= radiusuniqueidentifier=user_id11,ou=radiusActiveUsers,
cn=aixradius.
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Inside get_client_secret
routine for ip:10.10.10.10
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Found NAS-IP = [10.10.10.10]
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Found shared secret.
Aug 18 10:04:18 server1 radiusd[389278]: [2]:authenticate_password_PAP:
Passwords do not match, user is rejected
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Authentication failed for user
[user_id1] using IP [10.10.10.10]
Aug 18 10:04:18 server1 radiusd[389278]: [2]:ACCESS-REJECT - sending reject
for id 79 to 10.10.10.10 (client1.austin.ibm.com)
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Inside proxy_response_needed()
function
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Proxy is not turned on
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Inside get_client_secret
routine for ip:10.10.10.10
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Found NAS-IP = [10.10.10.10]
```

```
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Found shared secret.
Aug 18 10:04:18 server1 radiusd[389278]: [2]:send_reject() Outgoing Packet:
Aug 18 10:04:18 server1 radiusd[389278]: [2]: Code = 3, Id = 79, Length =
30
Aug 18 10:04:18 server1 radiusd[389278]: [2]:send_reject() Outgoing Packet:
Aug 18 10:04:18 server1 radiusd[389278]: [2]:  Code = 3, Id = 79, Length =
30
Aug 18 10:04:18 server1 radiusd[389278]: [2]:  Authenticator =
0x05D4865C6EBEFC1A9300D2DC66F3DBE9
Aug 18 10:04:18 server1 radiusd[389278]: [2]:    Type =  18, Length =  10,
Value = 0x4261646E65737321
Aug 18 10:04:18 server1 radiusd[389278]: [2]:*** Leave Process_Packet() **
```

Fonctionnalité ”à la demande”

Vous pouvez lancer plusieurs démons de serveur RADIUS d’authentification et de comptabilité selon les besoins. Chaque serveur écoute un port distinct. Par défaut, dans le fichier **radiusd.conf** le port 1812 est associé à l’authentification et le port 1813 à la comptabilité. Ces numéros de port sont attribués par le service IANA (Internet Assigned Numbers Authority). En mettant à jour **radiusd.conf**, vous pouvez utiliser ces numéros de port ainsi que d’autres, selon les besoins. Veillez à ne pas utiliser des numéros de port déjà attribués à des services existants. Si vous entrez plusieurs numéros de port dans la zone **Authentication_Ports** et **Accounting_Ports** du fichier **radiusd.conf**, un démon **radiusd** est lancé pour chaque port. Ces démons vont alors écouter le port qui leur est attribué.

Démarrage et arrêt du serveur RADIUS

Il est nécessaire d’arrêter et de redémarrer les démons **radiusd** après toute modification du fichier **/etc/radius/radiusd.conf** du serveur RADIUS ou des fichiers d’autorisation par défaut **/etc/radius/authorization/default.policy** ou **/etc/radius/authorization/default.auth**. Vous pouvez effectuer ces actions à l’aide de l’outil SMIT ou en ligne de commande.

Pour arrêter et lancer le serveur RADIUS, utilisez les commandes suivantes :

```
>stopsrc –s radiusd
>startsrc –s radiusd
```

L’arrêt et le redémarrage de RADIUS est nécessaire afin de permettre au démon de construire une table de mémoire avec tous les attributs par défaut des fichiers de configuration ci–dessus. La mémoire partagée est utilisée pour chaque utilisateur local et la table utilisateur locale n’est construite qu’au moment de l’initialisation du démon pour des raisons de performances.

Activation du support NLS

La commande **raddbm** de RADIUS et les écrans de l’outil SMIT sont compatibles avec le support de la langue nationale (NLS) et utilisent les appels API NLS AIX standard pour assurer cette fonction.

Rubrique connexe

Commandes : **installp** , **mkuser** et **raddbm**

Chapitre 18. Prévention des intrusions sous AIX

La fonction de prévention des intrusions d'AIX détecte les données non appropriées, non autorisées ou considérées comme néfastes pour un système. La section suivante décrit les différents types de détection d'intrusion sous AIX.

Détection des intrusions

La détection des intrusions consiste à surveiller et analyser les événements sur le système afin d'intercepter et rejeter toute tentative d'accès illicite au système. Sous AIX, la détection des accès ou tentatives d'accès non autorisés est effectuée via la surveillance de certaines opérations et l'application de règles de filtrage à ces opérations.

Remarque : Pour activer la détection des intrusions, vous devez installer les ensembles de fichiers **bos.net.ipsec** sur le système hôte. Les technologies de détection reposent sur les fonctions AIX de sécurité IP appelées IPsec (Internet Protocol Security).

Règles de filtrage par correspondance de trame

Le filtrage par correspondance de trame repose sur l'utilisation d'une règle de filtrage IPsec permettant de filtrer les paquets sur le réseau. Le masque du filtre peut être une chaîne de texte, une chaîne hexadécimale ou un fichier contenant plusieurs masques. Si ce type de filtrage est appliqué, en cas de correspondance entre le masque du filtre et le contenu d'un paquet du réseau, l'action prédéfinie par la règle de filtrage est effectuée.

Ces règles de filtrage sont appliquées uniquement aux paquets entrants sur le réseau. Utilisez la commande **genfilt** pour ajouter une règle de filtrage à la table de règles de filtrage. Les règles de filtrage générées par cette commande sont appelées règles de filtrage manuelles. Utilisez la commande **mkfilt** pour activer ou désactiver les règles de filtrage. La commande **mkfilt** peut également permettre de gérer la fonction de journalisation du filtrage.

Un fichier de filtrage peut contenir une liste de masques textuels ou héxadécimaux (un par ligne). Les règles de filtrage par correspondance de trame peuvent permettre la protection contre les virus, les surcharges de la mémoire tampon et les attaques contre la sécurité du système.

L'application de règles de filtrage par correpondance de trame peut affecter les performances du système si elles sont utilisées de façon étendue ou si le nombre de masques de filtrage est élevé. Il est préférable de limiter au maximum leur champ d'application. Par exemple, si un masque de filtrage correspondant à un virus s'applique à **sendmail**, indiquez dans la règle de filtrage le port de destination SMTP de **sendmail** (port 25) . Cela permet de laisser passer les autres données et d'éviter ainsi une baisse des performances due à la recherche de la correspondance.

La commande **genfilt** reconnaît et comprend le format des correspondances de trame utilisées dans certaines versions indiquées à l'adresse http://www.clamav.net.

Types de masques de filtrage

Il existe trois types de masque de filtrage : texte, hexadécimal et fichier.
Les règles de filtrage sont appliquées uniquement aux paquets entrants.

Masque de filtrage textuel

Un masque de filtrage textuel est une chaîne ASCII du type suivant :

```
GET /../../../../../../../../../
```

Masque de filtrage hexadécimal

Un masque de filtrage hexadécimal est du type suivant :

```
0x33c0b805e0cd16b807e0cd1650558becc7460200f05d0733ffb8c800b9fffff3abb00150
e670e47132c0e67158fec03c8075f033c033c9b002fa99cd26fb4183f90575f5c3
```

Remarque : Les caractères 0x en début du masque de filtrage hexadécimal permet de différencier celui–ci d'un masque textuel.

Fichier

Un fichier peut contenir une liste de masques de filtrage textuels ou hexadécimaux (un par ligne). Vous trouverez des exemples de fichiers de masques de filtrage à l'adresse http://www.clamav.net.

Règles de filtrage de blocage de port et d'hôte

Une règle de filtrage de blocage est une règle dynamique qui interdit à un hôte éloigné ou à un hôte et un port éloignés d'accéder à la machine locale en cas de concordance avec les critères définis dans la règle.

Comme une attaque est en général précédée d'une analyse de port, les règles de filtrage de blocage de port sont particulièrement utiles dans la prévention des intrusions.

Par exemple, si l'hôte local n'utilise pas le port 37 du serveur, qui correspond au serveur d'horloge, l'hôte éloigné n'accède pas au port 37, sauf s'il effectue une analyse de port. En plaçant une règle de filtrage de blocage sur le port 37, en cas de tentative d'accès de l'hôte éloigné sur ce port, la règle de filtrage de blocage génère une règle effective qui empêche l'hôte éloigné d'accéder au port durant la période définie dans la zone **expiration time** de la règle.

Règles de filtrage de blocage d'hôte

Lorsque le critère d'une règle de filtrage de blocage d'hôte est rencontré, la règle effective dynamiquement créée bloque ou rejette l'ensemble du trafic réseau issu de l'hôte éloigné durant la période de temps définie.

Règles de filtrage de blocage de port

Lorsque le critère d'une règle de filtrage de blocage de port est rencontré, la règle effective dynamiquement créée bloque ou rejette le trafic réseau issu du port durant la période de temps définie.

Si la valeur de la zone **expiration time** est 0, la règle effective dynamiquement créée n'expire pas.

Remarques :

1. Le délai d'expiration défini pour une règle de filtrage de blocage de port s'applique uniquement à la règle effective dynamiquement créée.
2. Les règles effectives dynamiquement créées peuvent être visualisées uniquement à l'aide de la commande **lsfilt –a** .

Règles de filtrage avec état

Les filtres avec état analysent les données telles que les adresses source et destination, les numéros de port et l'état. En appliquant les règles de filtrage IF, ELSE ou ENDIF à ces indicateurs d'en–tête, les systèmes avec état peuvent faire des choix de filtrage sur l'ensemble d'une session au lieu de le faire sur un paquet particuler et ses données d'en–tête.

Le contrôle avec état analyse les paquets entrants et sortants. Si les règles de filtrage avec état sont activées à l'aide de la commande **mkfilt –u**, les règles du bloc ELSE sont toujours analysées jusqu'à ce que la règle IF soit vraie. Lorsque la règle ou la condition IF est vraie, les règles du bloc IF sont appliquées jusqu'à ce que les règles de filtrage soient réactivées avec la commande **mkfilt –u**.

La commande **ckfilt** vérifie la syntaxe des règles de filtrage avec état et les affiche de la manière suivante :

```
%ckfilt –v4
Beginning of IPv4 filter rules.
Rule 2
IF Rule 3
    IF Rule 4
        Rule 5
    ELSE Rule 6
        Rule 7
    ENDIF Rule 8
ELSE Rule 9
    Rule 10
ENDIF Rule 11
Rule 0
```

Règles temporisées

Dans une règle temporisée, une durée en secondes définit le délai durant lequel la règle de filtrage est appliquée une fois que la commande **mkfilt –v [4|6] –u** l'a rendue effective. Le délai d'expiration est défini à l'aide de la commande **genfilt –e**. Pour plus d'informations, consultez la description des commandes **mkfilt** et **genfilt**.

Remarque : La temporisation n'a aucun effet sur les règles IF, ELSE et ENDIF. Si un délai d'expiration est défini dans une règle de blocage de port ou d'hôte, la durée s'applique uniquement à la règle effective lancée par la règle de blocage. Les règles de blocage ne sont associées à aucun délai d'expiration.

Accès aux règles de filtrage à l'aide de l'outil SMIT

Pour configurer des règles de filtrage à l'aide de l'outil SMIT, procédez comme suit :

1. Sur la ligne de commande, entrez :

```
smitty ipsec4
```

2. Sélectionnez **Advanced IP Security Configuration**.
3. Sélectionnez **Configure IP Security Filter Rules**.
4. Sélectionnez **Add an IP Security Filter Rule**.

```
                        Add an IP Security Filter Rule

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                  [Entry Fields]
* Rule Action                                        [permit]                +
* IP Source Address                                  []
* IP Source Mask                                     []
  IP Destination Address                             []
  IP Destination Mask                                []
* Apply to Source Routing? (PERMIT/inbound only)     [yes]                   +
* Protocol                                           [all]                   +
* Source Port / ICMP Type Operation                  [any]                   +
* Source Port Number / ICMP Type                     [0]                      #
* Destination Port / ICMP Code Operation             [any]                   +
* Destination Port Number / ICMP Type                [0]                      #
* Routing                                            [both]                  +
* Direction                                          [both]                  +
* Log Control                                        [no]                    +
* Fragmentation Control                              [0]                     +
* Interface                                          []                      +
  Expiration Time (sec)                              []                       #
  Pattern Type                                       [none]                  +
  Pattern / Pattern File                             []
  Description                                        []


Where "Pattern Type" may be one of the following
  x   none                                                                  x#
  x   pattern                                                               x
  x   file                                                                  x
  x   Anti-Virus patterns
```

Les valeurs possibles de la zone **action** sont les suivantes : `permit, deny, shun_host, shun_port, if, else, endif`.

Si un fichier de masque de filtrage est défini, il doit être lisible lorsque les règles de filtrage sont activées avec la commande **mkfilt –a**. Les règles de filtrage sont stockés dans la base de données **/etc/security/ipsec_filter**.

Rubrique connexe

Commandes : **chfilt**, **ckfilt**, **expfilt**, **genfilt**, **impfilt**, **lsfilt**, **mkfilt**, **mvfilt** et **rmfilt**.

Partie 3. Annexes

Annexe A. Vérification de la sécurité

Cette annexe indique les vérifications de sécurité à effectuer sur un système nouveau ou existant. Bien que cette liste ne soit pas exhaustive, elle peut servir de base à la création d'une liste complète pour votre environnement.

- Lors de l'installation d'un nouveau système, installez AIX depuis un support de base sécurisé. Exécutez les procédures suivantes au moment de l'installation :
  - N'installez pas d'interface graphique telles que CDE, GNOME ou KDE sur des serveurs.
  - Installez les correctifs de sécurité requis et les correctifs de niveau de maintenance recommandés. Contactez le responsable de la maintenance.
  - Sauvegardez le système au terme de l'installation initiale et stockez la sauvegarde en lieu sûr.
- Dressez des listes de contrôle des accès pour les fichiers et répertoires réservés.
- Désactivez les comptes utilisateur et les comptes système qui ne sont pas nécessaires, tels que démon, bin, sys, adm, lp et uucp. Il est déconseillé de supprimer des comptes car vous perdriez des informations sur eux, telles que les ID utilisateur et noms d'utilisateurs, qui peuvent toujours être associés à des données dans les sauvegardes système. Si un utilisateur est créé avec un ID utilisateur qui a été supprimé et si la sauvegarde système est restaurée sur le système, le nouvel utilisateur risque d'avoir un accès non souhaité au système restauré.
- Consultez régulièrement les fichiers **/etc/inetd.conf**, **/etc/inittab**, **/etc/rc.nfs** et **/etc/rc.tcpip** et retirez tous les démons et services qui ne sont pas nécessaires.
- Vérifiez que les droits d'accès aux fichiers suivants sont définis correctement :

```
-rw-rw-r-- root     system  /etc/filesystems
 -rw-rw-r-- root     system  /etc/hosts
 -rw------- root     system  /etc/inittab
 -rw-r--r-- root     system  /etc/vfs
 -rw-r--r-- root     system  /etc/security/failedlogin
 -rw-rw---- root     audit   /etc/security/audit/hosts
```

- Désactivez la connexion à distance au compte root. La connexion au compte root ne doit être possible que depuis la console du système.
- Activez l'audit du système. Pour plus d'informations, reportez–vous à la section Audit, page 4-1.
- Activez une politique de contrôle des connexions. Pour plus d'informations, reportez–vous à la section Fenêtre de connexion, page 1-21.
- Désactivez les droits des utilisateurs pour lancer la commande **xhost**. Pour plus d'informations, reportez–vous à la section Gestion des problèmes sous X11 et CDE, page 1-24.
- Empêchez les modifications non autorisées de la variable d'environnement **PATH**. Pour plus d'informations, reportez–vous à la section Variable d'environnement PATH, page 2-13.
- Désactivez telnet, rlogin, et rsh. Pour plus d'informations, reportez–vous à la section Sécurité TCP/IP, page 10-1.
- Créez des contrôles de comptes utilisateur. Pour plus d'informations, reportez–vous à la section Contrôle des comptes utilisateur, page 2-11.

- Mettez en place une politique stricte de mots de passe.
  Pour plus d'informations, reportez–vous à la section Mots de passe, page 2-20.
- Etablissez des quotas de disque pour les comptes utilisateur. Pour plus d'informations, reportez–vous à la section Reprise après un dépassement de quota, page 2-29.
- N'autorisez que les comptes d'administration à utiliser la commande **su**.
  Contrôlez les journaux de la commande **su** dans le fichier **/var/adm/sulog**.
- Activez le verrouillage d'écran sous X–Windows.
- Limitez l'accès aux commandes **cron** et **at** aux seuls comptes ayant besoin d'y accéder.
- Créez un alias de la commande **ls** pour afficher les fichiers cachés et les caractères cachés dans un nom de fichier.
- Utilisez un alias de la commande **rm** pour éviter toute suppression involontaire de fichiers du système.
- Désactivez les services réseau qui ne sont pas nécessaires.
  Pour plus d'informations, reportez–vous à la section Services réseau, page 11-1.
- Effectuez des sauvegardes système régulières et vérifiez leur intégrité.
- Inscrivez–vous aux listes de distribution des e–mails ayant trait à la sécurité.

Annexe B. Sources d'informations sur la sécurité

Cette annexe fournit des informations sur les ressources concernant la sécurité.

Sites Web concernant la sécurité

CERIAS (Center for Education and Research in Information Assurance and Security): http://www.cerias.purdue.edu/

CERT (Computer Emergency Response Team, at Carnegie Mellon University): http://www.cert.org

CIAC (Computer Incident Advisory Capability): http://ciac.llnl.gov

Computer Security Resource Clearinghouse: http://csrc.ncsl.nist.gov/

FIRST (Forum of Incident Response and Security Teams): http://www.first.org/

OpenSSH: http://www.openssh.org/

Listes de diffusion de sécurité

CERT : http://www.cert.org/contact_cert/certmaillist.html

Références de sécurité en ligne

FAQ sur les concepts de critères communs : http://www.radium.ncsc.mil/tpep/process/faq–sect3.html

Bibliothèque Rainbow : http://www.radium.ncsc.mil/tpep/library/rainbow/

faqs.org : http://www.faqs.org/faqs/computer–security/

Annexe C. Résumé des principaux services système AIX

Le tableau suivant répertorie les services système les plus courants sous AIX. Ce tableau servira de point de départ pour la sécurisation de votre système.

Avant de commencer la sécurisation de votre système, sauvegardez tous vos fichiers de configuration, notamment :

- **/etc/inetd.conf**
- **/etc/inittab**
- **/etc/rc.nfs**
- **/etc/rc.tcpip**

Service	Démon	Lancé par	Fonction	Commentaires
inetd/bootps	inetd	/etc/inetd.conf	services bootp pour clients sans disque	• Nécessaire pour l'environnement NIM (Network Installation Management) et le démarrage à distance des systèmes • Fonctionne avec tftp • Désactivez dans la plupart des cas
inetd/chargen	inetd	/etc/inetd.conf	générateur de caractères (tests seulement)	• Disponible en tant que service TCP et UDP • Attaques possibles par refus de service • Désactivez à moins que vous ne testiez votre réseau
inetd/cmsd	inetd	/etc/inetd.conf	service de calendrier (comme utilisé par CDE)	• Exécuté en tant que root, donc problèmes de sécurité • Désactivez à moins que vous n'ayez besoin de ce service avec CDE • Désactivez sur les serveurs de bases de données back–office
inetd/comsat	inetd	/etc/inetd.conf	Signale les messages électroniques entrants	• Exécuté en tant que root, donc problèmes de sécurité • Rarement nécessaire • Désactivez

Service	Démon	Lancé par	Fonction	Commentaires
inetd/daytime	inetd	/etc/inetd.conf	service de date obsolète (tests seulement)	• Exécuté en tant que root • Disponible en tant que service TCP et UDP • Possibilités d'attaques PING par refus de service • Service obsolète utilisé seulement pour les tests • Désactivez
inetd/discard	inetd	/etc/inetd.conf	service /dev/null (tests seulement)	• Disponible en tant que service TCP et UDP • Utilisé lors d'attaques par refus de service • Service obsolète utilisé seulement pour les tests • Désactivez
inetd/dtspc	inetd	/etc/inetd.conf	Commande de sous–processus CDE	• Ce service est lancé automatiquement par le démon **inetd** en réponse à une demande d'un client CDE de lancer un processus sur l'hôte du démon. Vulnérable aux attaques • Désactivez sur les serveurs de back–office sans CDE • CDE doit pouvoir fonctionner sans ce service • Désactivez à moins que vous n'en ayez absolument besoin
inetd/echo	inetd	/etc/inetd.conf	service echo (tests seulement)	• Disponible en tant que service UDP et TCP • Utilisé lors d'attaques Smurf ou par refus de service • Utilisé comme echo sur un autre utilisateur pour passer un pare–feu ou lancer une attaque de données • Désactivez

Service	Démon	Lancé par	Fonction	Commentaires
inetd/exec	inetd	/etc/inetd.conf	service d'exécution à distance	• Exécuté en tant que root • Nécessite un ID utilisateur et un mot de passe, qui sont transmis sans protection • Un service à haut risque d'espionnage • Désactivez
inetd/finger	inetd	/etc/inetd.conf	informations sur les utilisateurs	• Exécuté en tant que root • Révèle des informations sur vos systèmes et utilisateurs • désactivez
inetd/ftp	inetd	/etc/inetd.conf	protocole de transfert de fichiers	• Exécuté en tant qu'utilisateur root • ID utilisateur et mot de passe transmis sans protection. Risque d'espionnage • Désactivez ce service et utilisez un shell sécurisé du domaine public
inetd/imap2	inetd	/etc/inetd.conf	Protocole d'accès au courrier électronique	• Vérifiez que vous utilisez la dernière version de ce serveur • Nécessaire uniquement sur un serveur de messagerie. Sinon, désactivez • ID utilisateur et mot de passe transmis sans protection
inetd/klogin	inetd	/etc/inetd.conf	Connexion Kerberos	• Activé si votre site utilise l'authentification Kerberos
inetd/kshell	inetd	/etc/inetd.conf	Shell Kerberos	• Activé si votre site utilise l'authentification Kerberos
inetd/login	inetd	/etc/inetd.conf	service rlogin	• Susceptible d'être victime d'intrusion IP ou DNS • Les données, y compris les ID utilisateur et mots de passe, sont transmises sans protection. • Exécuté en tant que root • Utilisez un shell sécurisé plutôt que ce service

Service	Démon	Lancé par	Fonction	Commentaires
inetd/netstat	inetd	/etc/inetd.conf	reporting de l'état du réseau	• Susceptible de révéler des informations réseau aux hackers en cas d'exécution sur votre système • Désactivez
inetd/ntalk	inetd	/etc/inetd.conf	Permet aux utilisateurs de dialoguer entre eux	• Exécuté en tant que root • Pas nécessaire sur les serveurs de production ou de back–office • Désactivez à moins que vous en ayez absolument besoin
inetd/pcnfsd	inetd	/etc/inetd.conf	services de fichiers NFS PC	• Désactivez ce service s'il n'est pas en cours d'utilisation • Si vous avez besoin d'un service similaire, envisagez Samba, car le démon pcnfsd englobe les définitions SMB de Microsoft
inetd/pop3	inetd	/etc/linetd.conf	Protocole POP (Post Office Protocol)	• Les ID utilisateur et les mots de passe sont transmis sans protection • Nécessaire si votre système est un serveur de messagerie et si certains de vos clients utilisent des applications uniquement compatibles POP3 • Préférez IMAP si vos clients l'utilisent, ou bien POP3s. Ce service dispose d'un tunnel SSL (Secure Socket Layer) • Désactivez si vous n'être pas un serveur de messagerie ou n'avez pas de client nécessitant des services POP
inetd/rexd	inetd	/etc/inetd.conf	exécution à distance	• Exécuté en tant que root • Associé à la commande **on** • Désactivez ce service • Utilisez plutôt **rsh** et **rshd**

Service	Démon	Lancé par	Fonction	Commentaires
inetd/quotad	inetd	/etc/inetd.conf	rapports sur les quotas de fichiers (pour clients NFS)	• Nécessaire uniquement en cas d'exécution de services de fichiers NFS. • Désactivez ce service à moins que vous ne deviez répondre à la commande **quota** • Si vous devez utiliser ce service, effectuez les mises à jour et correctifs
inetd/rstatd	inetd	/etc/inetd.conf	Serveur Kernel Statistics	• Si vous devez contrôler des systèmes, utilisez SNMP et désactivez ce service • Nécessaire si vous devez utiliser la commande **rup**
inetd/rusersd	inetd	/etc/inetd.conf	informations sur l'utilisateur connecté	• Un service non indispensable. Désactivez • Exécuté en tant que root • Révèle la liste des utilisateurs en cours sur votre système ; associé à **rusers**
inetd/rwalld	inetd	/etc/inetd.conf	écriture à tous les utilisateurs	• Exécuté en tant que root • Vous devrez peut–être conserver ce service si vos systèmes comptent des utilisateurs interactifs • Ce n'est pas le cas si vos systèmes sont des serveurs de production ou de bases de données • Désactivez
inetd/shell	inetd	/etc/inetd.conf	service rsh	• Désactivez si possible. Remplacez par un shell sécurisé • Si vous devez utiliser ce service, utilisez TCP Wrapper pour empêcher l'espionnage et limiter les risques • Requis pour le programme de distribution de logiciel **Xhier**

Service	Démon	Lancé par	Fonction	Commentaires
inetd/sprayd	inetd	/etc/inetd.conf	tests spray RPC	• Exécuté en tant que root • Peut être nécessaire pour diagnostiquer les problèmes de réseau NFS • Désactivez si vous n'utilisez pas NFS
inetd/systat	inetd	/etc/inted.conf	rapport d'état "ps –ef"	• Permet à des sites distants d'afficher l'état des processus sur votre système • Service désactivé par défaut. Vérifiez régulièrement qu'il n'a pas été activé.
inetd/talk	inetd	/etc/inetd.conf	établissement d'un partage d'écran entre 2 utilisateurs du réseau	• Service non nécessaire • Utilisé avec la commande **talk** • Fournit le service UDP sur le port 517 • Désactivez à moins que vous ayez besoin de plusieurs sessions de chat interactives pour utilisateur UNIX
inetd/ntalk	inetd	/etc/inetd.conf	partage d'écran "new talk" entre 2 utilisateurs du réseau	• Service non nécessaire • Utilisé avec la commande **talk** • Fournit un service UDP sur le port 517 • Désactivez à moins que vous ayez besoin de plusieurs sessions de chat interactives pour utilisateur UNIX
inetd/telnet	inetd	/etc/inetd.conf	service telnet	• Sessions de connexion à distance, ID utilisateur et mots de passe transmis sans protection • Si possible, désactivez ce service et utilisez un shell sécurisé pour l'accès à distance

Service	Démon	Lancé par	Fonction	Commentaires
inetd/tftp	inetd	/etc/inetd.conf	transfert de fichiers	• Fournit un service UDP sur le port 69 • Exécuté en tant que root, risque d'intrusion • Utilisé par NIM • Désactivez à moins que vous utilisiez NIM ou que vous deviez démarrer un poste sans disque
inetd/time	inetd	/etc/inetd.conf	service de date obsolète (tests seulement)	• Fonction interne à **inetd**, utilisée par la commande **rdate**. • Disponible en tant que service TCP et UDP • Parfois utilisée pour synchroniser les horloges lors de l'amorçage • Service obsolète. Remplacez par **ntpdate** • Désactivez après avoir testé vos systèmes (amorçage/redémarrage) sans ce service et constaté leur bon fonctionnement
inetd/ttdbserver	inetd	/etc/inetd.conf	serveur de bases de données tool–talk (pour CDE)	• **rpc.ttdbserverd** est exécuté en tant que root, donc problème de sécurité • Décrit comme requis pour CDE, mais CDE peut fonctionner sans • A ne pas utiliser sur des serveurs back–office ou sur des systèmes dont il faut assurer la sécurité
inetd/uucp	inetd	/etc/inetd.conf	réseau UUCP	• Désactivez à moins que vous ayez une application NIM qui utilise UUCP

Service	Démon	Lancé par	Fonction	Commentaires
inittab/dt	init	script /etc/rc.dt dans /etc/inittab	connexion d'un poste de bureau à l'environnement CDE	• Lance le serveur X11 sur la console • Prend en charge le protocole xdcmp (X11 Display Manager Control Protocol) pour que les autres postes X11 puissent se connecter à la même machine • Service à n'utiliser que sur les stations de travail personnelles. A ne pas utiliser pour des systèmes de back–office
inittab/dt_nogb	init	/etc/inittab	connexion bureau à l'environnement CDE (pas d'amorçage graphique)	• Pas d'affichage graphique avant que le système ne soit entièrement démarré • Mêmes problèmes qu'avec **inittab/dt**
inittab/httpdlite	init	/etc/inittab	serveur Web pour la commande **docsearch**	• Serveur Web par défaut pour le moteur docsearch • Désactivez à moins que votre machine soit un serveur de documentation
inittab/i4ls	init	/etc/inittab	serveurs de gestion des licences	• Activez pour les serveurs de développement • Désactivez pour les serveurs de production • Activez pour les serveurs de bases de données back–office avec des conditions de licence • Gère les compilateurs, logiciels de bases de données, ou autres produits sous licence
inittab/imqss	init	/etc/inittab	moteur de recherche pour "docsearch"	• Elément du serveur Web par défaut pour le moteur docsearch • Désactivez à moins que votre machine soit un serveur de documentation

Service	Démon	Lancé par	Fonction	Commentaires
inittab/lpd	init	/etc/inittab	interface d'imprimante parallèle BSD	• Accepte les travaux d'impression d'autres systèmes • Vous pouvez désactiver ce service et continuer à envoyer des travaux au serveur d'impression • Désactivez une fois que vous êtes sûr que les impressions ne sont pas affectées
inittab/nfs	init	/etc/inittab	Système de fichiers NFS/Services d'informations réseau	• services NFS et NIS basés sur UDP/RPC • Authentification minimale • Devraient être inutiles sur les serveurs de bases de données back–office • Désactivez pour les serveurs de back–office
inittab/piobe	init	/etc/inittab	traitement E/S impressions	• Gère la planification, la mise en cache sur disque et l'impression des travaux soumis par **qdaemon** • Désactivez si vous n'imprimez pas depuis votre système (car vous envoyez des travaux à un serveur)
inittab/qdaemon	init	/etc/inittab	démon de file d'attente (pour l'impression)	• Soumet des travaux au démon **piobe** • Désactivez si vous n'imprimez pas depuis votre système
inittab/uprintfd	init	/etc/inittab	messages du noyau	• Généralement pas nécessaire • Désactivez
inittab/writesrv	init	/etc/inittab	écriture de notes vers ttys	• Uniquement pour les utilisateurs interactifs de stations de travail UNIX • Service à désactiver pour les serveurs, bases de données back–office et serveurs de développement • Activez pour les stations de travail

Service	Démon	Lancé par	Fonction	Commentaires
inittab/xdm	init	/etc/inittab	gestion d'affichage X11 traditionnelle	• Ne pas utiliser sur des serveurs de bases de données ou de production back–office • Ne pas utiliser sur des systèmes de développement, à moins que la gestion d'affichage X11 soit requise • Acceptable sur les stations de travail si l'affichage graphique est nécessaire
rc.nfs/automountd		/etc/rc.nfs	systèmes de fichiers à montage automatique	• Activez pour les stations de travail si vous utilisez NFS • A ne pas utiliser pour le développement ou des serveurs de back–office
rc.nfs/biod		/etc/rc.nfs	Démon d'E/S par blocs (nécessaire pour les serveurs NFS)	• N'activez que pour les serveurs NFS • Sinon, désactivez–le, ainsi que **nfsd** et **rpc.mountd**
rc.nfs/keyserv		/etc/rc.nfs	serveur de clés RPC sécurisé	• Gère les clés requises pour RPC sécurisé • Important pour NIS+ • Désactivez si vous n'utilisez *pas* NFS et NIS et NIS+
rc.nfs/nfsd		/etc/rc.nfs	services NFS (nécessaire pour les serveurs NFS)	• Authentification faible • Peut conduire à détruire le contexte de pile • Activez pour les serveurs de fichiers NFS • Sinon, désactivez aussi **biod**, **nfsd** et **rpc.mountd**
rc.nfs/rpc.lockd		/etc/rc.nfs	verrouillages de fichiers NFS	• Désactivez si vous n'utilisez pas NFS • Désactivez si vous n'utilisez pas de verrouillages de fichiers sur le réseau • Le démon **lockd** est parmi les 10 principales menaces au classement SANS

Service	Démon	Lancé par	Fonction	Commentaires
rc.nfs/rpc.mountd		/etc/rc.nfs	montages de fichiers NFS (requis pour les serveurs NFS)	• Authentification faible • Peut conduire à détruire le contexte de pile • N'activez que pour les serveurs de fichiers NFS • Sinon, désactivez aussi **biod**, et **nfsd**
rc.nfs/rpc.statd		/etc/rc.nfs	verrouillages de fichiers NFS (pour les récupérer)	• Met en œuvre les verrouillages de fichiers sur NFS • Désactivez si vous n'utilisez pas NFS
rc.nfs/rpc.yppasswdd		/etc/rc.nfs	démon de mots de passe NIS (pour le maître NIS)	• Utilisé pour manipuler le fichier local des mots de passe • N'activez que sur le maître NIS
rc.nfs/ypupdated		/etc/rc.nfs	démon de mise à jour NIS (pour esclave NIS)	• Reçoit du maître NIS des mappages de bases de données NIS • Nécessaire uniquement sur les esclaves NIS
rc.tcpip/autoconf6		/etc/rc.tcpip	interfaces IPv6	• Désactivez si vous n'utilisez pas IPV6
rc.tcpip/dhcpcd		/etc/rc.tcpip	protocole DHCP (client)	• Les serveurs back–office ne devraient pas utiliser DHCP. Désactivez ce service • Désactivez si votre hôte n'utilise pas DHCP
rc.tcpip/dhcprd		/etc/rc.tcpip	protocole DHCP (relais)	• Recueille des diffusions DHCP et les envoie à un serveur sur un autre réseau • Réplique d'un service trouvé sur les routeurs • Désactivez si vous n'utilisez pas DHCP ou si vous vous chargez de transmettre les informations entre réseaux

Service	Démon	Lancé par	Fonction	Commentaires
rc.tcpip/dhcpsd		/etc/rc.tcpip	protocole DHCP (serveur)	• Répond aux requêtes DHCP des clients au moment de l'amorçage. Donne des informations, telles que l'adresse de diffusion, le routeur, le masque réseau, le numéro et le nom IP • Désactivez si vous n'utilisez pas DHCP • Désactivé sur les serveurs de production et de back–office avec les hôtes qui n'utilisent pas DHCP
rc.tcpip/dpid2		/etc/rc.tcpip	service SNMP obsolète	• A désactiver si vous n'utilisez pas SNMP
rc.tcpip/gated		/etc/rc.tcpip	routage par porte entre interfaces	• Emule les fonctions d'un router • Désactiver ce service et le remplacer par RIP ou par un routeur
rc.tcpip/inetd		/etc/rc.tcpip	services inetd	• Désactivez pour obtenir une meilleure sécurité, mais pas toujours pratique • Sa désactivation entraîne celle des services de shell distants, nécessaires pour certains serveurs Web et de messagerie
rc.tcpip/mrouted		/etc/rc.tcpip	routage vers plusieurs destinataires	• Emule les fonctions d'un routeur pour l'envoi de paquets à plusieurs destinataires entre segments de réseau • Désactivez ce service. Remplacez par un routeur
rc.tcpip/names		/etc/rc.tcpip	serveur de noms DNS	• N'utilisez que si votre machine est un serveur de noms DNS • Désactivez pour les stations de travail, les serveurs de développement et de production
rc.tcpip/ndp–host		/etc/rc.tcpip	hôte IPv6	• Désactivez si vous n'utilisez pas IPV6

Service	Démon	Lancé par	Fonction	Commentaires
rc.tcpip/ndp–router		/etc/rc.tcpip	routage IPv6	• Désactivez si vous n'utilisez pas IPV6. Envisagez de remplacer IPV6 par un routeur
rc.tcpip/portmap		/etc/rc.tcpip	services RPC	• Service requis • Les serveurs RPC s'enregistrent à l'aide du démon **portmap**. Les clients à la recherche d'un service RPC envoient une requête au démon **portmap** • Ne désactivez que si vous êtes parvenu à supprimer des services RPC de sorte que le seul restant soit **portmap**
rc.tcpip/routed		/etc/rc.tcpip	routage RIP entre interfaces	• Emule les fonctions d'un router • Désactivez si vous avez un routeur de paquets entre réseaux
rc.tcpip/rwhod		/etc/rc.tcpip	Démon "who" distant	• Recueille et diffuse des données aux autres serveurs du même réseau • Désactivez ce service
rc.tcpip/sendmail		/etc/rc.tcpip	services de messagerie	• Exécuté en tant que root • A l'origine de nombreux problèmes de sécurité • Désactivez si la machine n'est pas utilisée comme serveur de messagerie • Si vous le désactivez, effectuez l'une des actions suivantes : – Placez une entrée dans crontab pour vider la file d'attente. Utilisez la commande **/usr/lib/sendmail –q** – Configurez les services DNS afin que les messages pour votre serveur arrivent sur un autre système

Service	Démon	Lancé par	Fonction	Commentaires
rc.tcpip/snmpd		/etc/rc.tcpip	SNMP (Simple Network Management Protocol)	• Désactivez si vous ne contrôlez pas le système à l'aide d'outils SNMP • SNMP peut être requis sur certains serveurs critiques
rc.tcpip/syslogd		/etc/rc.tcpip	journal système des événements	• Il n'est *pas* recommandé de désactiver ce service • Sujet aux attaques par refus de service • Requis dans tout système
rc.tcpip/timed		/etc/rc.tcpip	démon Old Time	• Désactivez ce service et remplacez–le par xntp
rc.tcpip/xntpd		/etc/rc.tcpip	démon New Time	• Assure la synchronisation des horloges des systèmes • Désactivez ce service. • Configurez d'autres systèmes comme serveurs de temps et laissez les autres systèmes se synchroniser à eux à l'aide d'une tâche cron qui appelle ntpdate
dt login		/usr/dt/config/Xaccess	CDE sans restriction	• Si vous ne fournissez pas la connexion CDE à un groupe de stations X11, vous pouvez limiter dtlogin à la console.
service FTP anonyme		user rmuser –p <nomutilisateur>	ftp anonyme	• Le FTP anonyme vous empêche savoir qui utilise le FTP • Retirez l'utilisateur si ce compte existe : **rmuser –p ftp** • Vous pouvez augmenter la sécurité en plaçant dans le fichier **/etc/ftpusers** une liste des utilisateurs qui ne doivent pas accéder par ftp à votre système

Service	Démon	Lancé par	Fonction	Commentaires
écritures par FTP anonyme			envois par FTP anonyme	• Aucun fichier ne doit appartenir à ftp. • Les envois par FTP anonyme permettent d'envoyer un code dangereux à votre système. • Placez les noms des utilisateurs à interdire dans le fichier **/etc/ftpusers** • Voici quelques utilisateurs créés par le système et que vous pouvez empêcher d'envoyer des données via FTP anonyme : root, daemon, bin.sys, admin.uucp, guest, nobody, lpd, nuucp, ladp • Modifiez les droits de groupes et de propriétaires aux fichiers **ftpusers** : **chown root:system /etc/ftpusers** • Limitez les droits d'accès aux fichiers **ftpusers** : **chmod 644 /etc/ftpusers**
ftp.restrict			ftp vers comptes système	• Le fichier **ftpusers** ne doit autoriser aucun utilisateur extérieur à remplacer des fichiers root
root.access		/etc/security/user	rlogin/telnet vers compte root	• Attribuez la valeur false à l'option rlogin dans le **etc/security/user** • Tout utilisateur se connectant comme root doit d'abord se connecter sous son propre nom puis lancer la commande **su** vers root. Vous obtenez ainsi un suivi d'audit

Service	Démon	Lancé par	Fonction	Commentaires
snmpd.readWrite		/etc/snmpd.conf	communautés SNMP readWrite	• Désactiver le démon SNMP si vous n'utilisez *pas* SNMP. • Désactivez community private et community system dans le fichier **/etc/snmpd.conf** • Limitez communauté 'public' aux adresses IP qui contrôlent votre système
syslog.conf			configure syslogd	• Désactivez ce démon si vous n'avez pas configuré **/etc/syslog.conf** • Ne le désactivez pas si vous utilisez **syslog.conf** pour consigner des messages système

Annexe D. Résumé des options de service réseau

Pour obtenir une meilleure sécurité système, il existe plusieurs options de réseau que vous pouvez désactiver avec `0` ou activer avec `1`. La liste suivante indique les paramètres que vous pouvez utiliser avec la commande **no**.

Paramètre	Commande	Fonction
bcastping	/usr/sbin/no –o bcastping=0	Autorise la réponse aux paquets d'écho ICMP à l'adresse de diffusion. Désactiver cette option évite les attaques Smurf.
clean_partial_conns	/usr/sbin/no –o clean_partial_conns=1	Indique si oui ou non les attaques SYN (synchronisation du numéro de séquence) sont évitées.
directed_broadcast	/usr/sbin/no –o directed_broadcast=0	Indique si la diffusion dirigée vers une passerelle est autorisée ou non. La valeur 0 évite aux paquets dirigés d'atteindre un réseau distant.
icmpaddressmask	/usr/sbin/no –o icmpaddressmask=0	Indique si le système répond à une demande de masque d'adresse ICMP. Désactiver cette option empêche l'accès via des attaques par routage source.
ipforwarding	/usr/sbin/no –o ipforwarding=0	Indique si le noyau doit réexpédier des paquets. Désactiver cette option évite que des paquets redirigés n'atteignent un réseau distant.
ipignoreredirects	/usr/sbin/no –o ipignoreredirects=1	Indique s'il faut traiter ou non les redirections reçues.
ipsendredirects	/usr/sbin/no –o ipsendredirects=0	Indique si le noyau doit envoyer des signaux de redirection. Désactiver cette option évite que des paquets redirigés n'atteignent un réseau distant.
ip6srcrouteforward	/usr/sbin/no –o ip6srcrouteforward=0	Indique si le système retransmet des paquets IPv6 routés par la source. Désactiver cette option empêche l'accès via des attaques par routage source.
ipsrcrouteforward	/usr/sbin/no –o ipsrcrouteforward=0	Indique si le système retransmet des paquets routés par la source. Désactiver cette option empêche l'accès via des attaques par routage source.
ipsrcrouterecv	/usr/sbin/no –o ipsrcrouterecv=0	Indique si le système accepte des paquets routés par la source. Désactiver cette option empêche l'accès via des attaques par routage source

Paramètre	Commande	Fonction
ipsrcroutesend	/usr/sbin/no –o ipsrcroutesend=0	Indique si les applications peuvent envoyer des paquets routés par la source. Désactiver cette option empêche l'accès via des attaques par routage source.
nonlocsroute	/usr/sbin/no –o nonlocsrcroute=0	Informe le protocole Internet que seuls des paquets routés par la source peuvent être adressés aux hôtes extérieurs au réseau local. Désactiver cette option empêche l'accès via des attaques par routage source.
tcp_pmtu_discover	/usr/sbin/no –o tcp_pmtu_discover=0	Désactiver cette option empêche l'accès via des attaques par routage source.
udp_pmtu_discover	/usr/sbin/no –o udp_pmtu_discover=0	Active ou désactive la recherche de chemin MTU pour les applications TCP. Désactiver cette option empêche l'accès via des attaques par routage source.

Pour de plus amples informations sur les options réseau, consultez le manuel *AIX 5L Version 5.3 Performance Management Guide*.

Index

Symbols

A

B

C

D

E

F

G

I

J

K

L

M

N

O

P

R

S

T

U

V

X

Vos remarques sur ce document / Technical publication remark form

Titre / Title : Bull AIX 5L Guide de sécurité

Nº Référence / Reference Nº : 86 F2 57EM 01

Daté / Dated : Février 2005

ERREURS DETECTEES / ERRORS IN PUBLICATION

AMELIORATIONS SUGGEREES / SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Vos remarques et suggestions seront examinées attentivement.
Si vous désirez une réponse écrite, veuillez indiquer ci-après votre adresse postale complète.

Your comments will be promptly investigated by qualified technical personnel and action will be taken as required.
If you require a written reply, please furnish your complete mailing address below.

NOM / NAME : _____ Date : _____

SOCIETE / COMPANY : _____

ADRESSE / ADDRESS : _____

_____

Remettez cet imprimé à un responsable BULL ou envoyez-le directement à :

Please give this technical publication remark form to your BULL representative or mail to:

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

Technical Publications Ordering Form

Bon de Commande de Documents Techniques

To order additional publications, please fill up a copy of this form and send it via mail to:
Pour commander des documents techniques, remplissez une copie de ce formulaire et envoyez-la à :

BULL CEDOC
ATTN / Mr. L. CHERUBIN
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

Phone / Téléphone : +33 (0) 2 41 73 63 96
FAX / Télécopie +33 (0) 2 41 73 60 19
E–Mail / Courrier électronique : srv.Cedoc@franp.bull.fr

Or visit our web sites at: / Ou visitez nos sites web à :

http://www.logistics.bull.net/cedoc

http://www-frec.bull.com **http://www.bull.com**

CEDOC Reference # Nº Référence CEDOC	**Qty** Qté	**CEDOC Reference #** Nº Référence CEDOC	**Qty** Qté	**CEDOC Reference #** Nº Référence CEDOC	**Qty** Qté
_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]	
_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]	
_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]	
_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]	
_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]	
_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]	
_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ _ _ _ _ [_ _]	
[_ _] : **no revision number means latest revision** / pas de numéro de révision signifie révision la plus récente					

NOM / NAME : _____ Date : _____

SOCIETE / COMPANY : _____

ADRESSE / ADDRESS : _____

_____

TELEPHONE / PHONE : _____ FAX : _____

E–MAIL : _____

For Bull Subsidiaries / Pour les Filiales Bull :

Identification: _____

For Bull Affiliated Customers / Pour les Clients Affiliés Bull :

Customer Code / Code Client : _____

For Bull Internal Customers / Pour les Clients Internes Bull :

Budgetary Section / Section Budgétaire : _____

For Others / Pour les Autres :

Please ask your Bull representative. / Merci de demander à votre contact Bull.

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

REFERENCE
86 F2 57EM 01

Utiliser les marques de découpe pour obtenir les étiquettes.
Use the cut marks to get the labels.

AIX
AIX 5L Guide
de sécurité

86 F2 57EM 01

AIX
AIX 5L Guide
de sécurité

86 F2 57EM 01

AIX
AIX 5L Guide
de sécurité

86 F2 57EM 01