Sécurisation des fichiers et vérification de l'intégrité des fichiers dans Oracle® Solaris 11.2

Référence: E53947
Juillet 2014

Table des matières

Utilisation de cette documentation

- **Présentation** : description de la protection des fichiers légitimes, de l'affichage des permissions sur les fichiers cachés, de la localisation et de l'interdiction d'exécution des fichiers non fiables. Explique également comment vérifier l'intégrité des fichiers au fil du temps sur les systèmes Oracle Solaris.
- **Public visé** : administrateurs système.
- **Connaissances requises** : critères du site en matière de sécurité.

Bibliothèque de documentation produit

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

Accès aux services de support Oracle

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

Commentaires

Faites part de vos commentaires sur cette documentation à l'adresse http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CHAPITRE 1

Contrôle de l'accès aux fichiers

Ce chapitre explique comment protéger les fichiers dans Oracle Solaris. En outre, ce chapitre explique comment protéger le système de fichiers dont les autorisations pourraient compromettre le système.

Remarque - ZFS est le système de fichiers du SE par défaut. Pour protéger les fichiers ZFS à l'aide de listes de contrôle d'accès (ACL), reportez-vous au Chapitre 7, “ Utilisation des ACL et des attributs pour protéger les fichiers Oracle Solaris ZFS ” du manuel “ Gestion des systèmes de fichiers ZFS dans OracleSolaris 11.2 ”.

Ce chapitre comprend les sections suivantes :

Utilisation des autorisations UNIX pour protéger les fichiers

Vous pouvez sécuriser les fichiers par le biais d'ACL et d'autorisations de fichiers UNIX. Les fichiers avec sticky bit et les fichiers exécutables nécessitent des mesures de sécurité spéciales.

Commandes d'affichage et de sécurisation des fichiers

Ce tableau décrit les commandes pour la surveillance et la sécurisation des fichiers et répertoires.

TABLEAU 1-1 Commandes de sécurisation des fichiers et répertoires

Commande	Description	Page de manuel
`ls`	Affiche la liste des fichiers dans un répertoire et des informations sur les fichiers.	`ls`(1)
`chown`	Modifie la propriété d'un fichier.	`chown`(1)
`chgrp`	Modifie le groupe propriétaire d'un fichier.	`chgrp`(1)
`chmod`	Modifie les autorisations de fichier. Vous pouvez utiliser le mode symbolique, qui utilise des lettres et des symboles, ou le mode absolu, qui utilise les octaux, pour modifier les autorisations d'un fichier.	`chmod`(1)

Propriété des fichiers et des répertoires

Les autorisations de fichiers UNIX classiques peuvent attribuer la propriété à trois catégories d'utilisateurs :

- **user** : le propriétaire du fichier ou du répertoire, qui est généralement l'utilisateur qui a créé le fichier. Le propriétaire d'un fichier peut décider qui a le droit de lire le fichier, d'écrire dans le fichier (pour y effectuer des modifications) ou, si le fichier est une commande, d'exécuter le fichier.
- **group** : les membres d'un groupe d'utilisateurs.
- **others** : tous les autres utilisateurs qui ne sont ni le propriétaire du fichier, ni membres du groupe.

Le propriétaire du fichier peut généralement affecter ou modifier les autorisations de fichier. En outre, le compte `root` peut modifier la propriété d'un fichier. Pour remplacer la stratégie du système, reportez-vous à l'Exemple 1-2, “Modification par les utilisateurs de la propriété de leurs propres fichiers”.

Il existe sept types de fichier. Chaque type est indiqué par un symbole :

- (signe moins)	Texte ou programme
b	Fichier spécial en mode bloc
c	Fichier spécial en mode caractère
d	Répertoire
l	Lien symbolique
s	Socket
D	Porte
P	Tube nommé (FIFO)

Autorisations des fichiers UNIX

Le tableau ci-dessous répertorie et décrit les autorisations que vous pouvez attribuer à chaque classe d'utilisateur pour un fichier ou un répertoire.

TABLEAU 1-2 Autorisations des fichiers et répertoires

Symbole	Autorisation	Objet	Description
`r`	Lecture	Fichier	Les utilisateurs autorisés peuvent ouvrir et lire le contenu d'un fichier.
		Répertoire	Les utilisateurs autorisés peuvent afficher la liste des fichiers dans le répertoire.
`w`	Ecriture	Fichier	Les utilisateurs autorisés peuvent modifier le contenu du fichier ou le supprimer.
		Répertoire	Les utilisateurs autorisés peuvent ajouter des fichiers ou des liens dans le répertoire. Ils peuvent également supprimer des fichiers ou des liens dans le répertoire.
`x`	Exécution	Fichier	Les utilisateurs autorisés peuvent exécuter le fichier, s'il s'agit d'un programme ou d'un script shell. Ils peuvent également exécuter le programme avec l'un des appels système `exec(2)`.
		Répertoire	Les utilisateurs autorisés peuvent ouvrir les fichiers ou exécuter des fichiers dans le répertoire. Ils peuvent également définir le répertoire et les répertoires inférieurs comme étant actuels.
`-`	Refusé	Fichier et répertoire	Les utilisateurs désignés ne peuvent pas lire, écrire ni exécuter le fichier.

Les autorisations des fichiers s'appliquent aux fichiers classiques et aux fichiers spéciaux tels que les périphériques, les sockets et les tubes nommés (FIFO).

Pour un lien symbolique, les autorisations qui s'appliquent sont celles du fichier vers lesquels pointe le lien.

Vous pouvez protéger les fichiers dans un répertoire et ses sous-répertoires en définissant des autorisations de fichiers restrictives sur ce répertoire. Notez, cependant, que le rôle `root` a accès à tous les fichiers et répertoires sur le système.

Autorisations de fichiers spéciales à l'aide des commandes `setuid`, `setgid` et sticky bit

Il existe trois types d'autorisations pour les fichiers exécutables et les répertoires publics : `setuid`, `setgid` et sticky bit. Lorsqu'elles sont définies, n'importe quel utilisateur qui exécute ce fichier exécutable prend l'ID du propriétaire (ou du groupe) du fichier exécutable.

Vous devez être très prudent lorsque vous définissez des autorisations spéciales, car elles constituent un risque de sécurité. Par exemple, un utilisateur peut obtenir des capacités `root` en

exécutant un programme qui définit l'ID utilisateur (UID) sur `0`, qui est l'UID de `root`. En outre, tous les utilisateurs peuvent définir des autorisations spéciales pour les fichiers qu'ils détiennent, ce qui constitue un autre problème de sécurité.

Il est recommandé de surveiller votre système pour toute utilisation non autorisée des autorisations `setuid` et `setgid` pour obtenir les capacités `root`. Une autorisation suspecte accorde la propriété d'un programme d'administration à un utilisateur plutôt qu'à `root` ou `bin`. Pour rechercher et afficher la liste de tous les fichiers qui utilisent ces autorisations spéciales, reportez-vous à la section “Recherche de fichiers avec des autorisations de fichier spéciales” à la page 23.

Autorisation `setuid`

Quand l'autorisation `setuid` est définie sur un fichier exécutable, un processus qui exécute ce fichier se voit accorder l'accès sur la base du propriétaire du fichier. L'accès n'est *pas* basé sur l'utilisateur qui exécute le fichier exécutable. Ces autorisations spéciales permettent à un utilisateur d'accéder aux fichiers et répertoires qui sont normalement disponibles uniquement pour le propriétaire.

Par exemple, les autorisations `setuid` sur la commande `passwd` permettent aux utilisateurs de modifier les mots de passe. Une commande `passwd` avec une autorisation `setuid` doit ressembler à ceci :

```
-r-sr-sr-x   1 root     sys       56808 Jun 17 12:02 /usr/bin/passwd
```

Cette autorisation spéciales présente un risque de sécurité. Certains utilisateurs déterminés peuvent trouver un moyen de conserver les autorisations qui sont leur sont accordées par le processus `setuid` même lorsque le processus a terminé de s'exécuter.

Remarque - L'utilisation des autorisations `setuid` avec des UID réservés (de 0 à 100) à partir d'un programme risque d'entraîner une définition incorrecte de l'UID réel. Utilisez d'un script shell ou évitez d'utiliser les UID réservés avec les autorisations `setuid`.

Autorisation `setgid`

Les autorisations `setgid` sont similaires aux autorisations `setuid`. L'ID de groupe (GID) effectif du processus est remplacé par le groupe qui est propriétaire du fichier, et un utilisateur se voit accorder les autorisations qui sont accordées au groupe. La commande `/usr/bin/mail` dispose des autorisations `setgid` :

```
-r-x--s--x   1 root   mail     71212 Jun 17 12:01 /usr/bin/mail
```

Lorsque les autorisations `setgid` sont appliquées à un répertoire, les fichiers qui ont été créés dans ce répertoire appartiennent au groupe auquel appartient le répertoire. Les fichiers n'appartiennent pas au groupe auquel le processus de création appartient. Tout utilisateur qui dispose d'autorisations d'écriture et d'exécution dans le répertoire peut y créer un fichier. Toutefois, le fichier appartient au groupe qui est propriétaire du répertoire, et non au groupe auquel appartient l'utilisateur.

Vous devez surveiller votre système pour toute utilisation non autorisée de l'autorisation `setgid` pour obtenir des capacités `root`. Des autorisations suspectes accordent l'accès de groupe à un tel programme à un groupe inhabituel plutôt qu'à `root` ou `bin`. Pour rechercher et afficher la liste de tous les fichiers qui utilisent ces autorisations, reportez-vous à la section “Recherche de fichiers avec des autorisations de fichier spéciales” à la page 23.

Sticky Bit

Le *sticky bit* est un bit d'autorisation qui protège les fichiers d'un répertoire. Si le sticky bit est défini pour le répertoire, un fichier peut être supprimé uniquement par le propriétaire du fichier, le propriétaire du répertoire ou par un utilisateur privilégié. L'utilisateur `root` est un exemple d'utilisateur privilégié. Le sticky bit empêche un utilisateur de supprimer les fichiers d'autres utilisateurs dans des répertoires publics tels que `/tmp` :

```
drwxrwxrwt 7  root  sys   400 Sep  3 13:37 tmp
```

Veillez à définir le sticky bit manuellement lorsque vous définissez un répertoire public directory dans un système de fichiers TMPFS. Pour plus d'instructions, reportez-vous à l'Exemple 1-5, “Définition des autorisations de fichiers spéciales en mode absolu”.

Valeur `umask` par défaut

Lorsque vous créez un fichier ou un répertoire, vous devez le créer avec un jeu d'autorisations par défaut. Les valeurs par défaut du système sont ouvertes. Un fichier texte dispose de `666` autorisations, et accorde des autorisations de lecture et d'écriture à tout le monde. Un répertoire et un fichier exécutable disposent de `777` autorisations, et accordent des autorisations de lecture, d'écriture et d'exécution à tout le monde. En règle générale, les utilisateurs remplacent les valeurs par défaut du système dans leurs fichiers d'initialisation du shell, tels que `.bashrc` et `.kshrc.user`. L'administrateur peut également définir des valeurs par défaut dans le fichier `/etc/profile`.

La valeur affectée par la commande `umask` est soustraite de la valeur par défaut. Ce processus a pour effet de refuser les autorisations de la même manière que la commande `chmod` les accorde. Par exemple, la commande `chmod 022` permet d'accorder l'autorisation d'écriture au groupe et aux autres. La commande `umask 022` refuse l'accès en écriture au groupe et aux autres.

Le tableau suivant présente quelques exemples typiques de valeurs `umask` et leur effet sur un fichier exécutable.

TABLEAU 1-3 Paramètres `umask` pour différents niveaux de sécurité

Niveau de sécurité	Paramètre `umask`	Autorisations refusés
Permissif (`744`)	`022`	`w` pour le groupe et les autres
Modéré (`751`)	`026`	`w` pour le groupe, `rw` pour les autres utilisateurs
Strict (`740`)	`027`	`w` pour le groupe, `rwx` pour les autres utilisateurs
Grave (`700`)	`077`	`rwx` pour le groupe et les autres

Pour plus d'informations sur la définition de la valeur `umask`, reportez-vous à la page de manuel `umask(1)`.

Modes d'autorisation de fichier

La commande `chmod` vous permet de modifier les autorisations d'un fichier. Vous devez être connecté en tant qu'utilisateur `root` ou le propriétaire d'un fichier ou d'un répertoire pour modifier ses autorisations.

Vous pouvez utiliser la commande `chmod` pour définir les autorisations dans l'un des deux modes suivants :

- **Mode absolu –** utilisez des numéros pour représenter les autorisations de fichier. Lorsque vous modifiez les autorisations l'aide du mode absolu, vous représentez les autorisations pour chaque triplet par un numéro de mode octal. Le mode absolu est la méthode la plus couramment utilisée pour définir les autorisations.
- **Mode symbolique –** utilisez des combinaisons de lettres et de symboles pour ajouter ou supprimer des autorisations.

Le tableau suivant répertorie les valeurs octales pour la définition des autorisations en mode absolu. Ces numéros s'utilisent en ensembles de trois pour définir les autorisations pour le propriétaire, le groupe et les autres, dans cet ordre. Par exemple, la valeur `644` définit les autorisations de lecture et d'écriture pour le propriétaire, et les autorisations de lecture seule pour le groupe et les autres.

TABLEAU 1-4 Définition des autorisations de fichiers en mode absolu

Valeur octale	Ensemble d'autorisations de fichier	Description des autorisations
`0`	`---`	Aucune autorisation
`1`	`--x`	Autorisation d'exécution uniquement
`2`	`-w-`	Autorisation d'écriture uniquement

Valeur octale	Ensemble d'autorisations de fichier	Description des autorisations
3	-wx	Autorisations d'exécution et d'écriture
4	r--	Autorisation de lecture seule
5	r-x	Autorisations de lecture et d'exécution
6	rw-	Autorisations de lecture et d'écriture
7	rwx	Autorisations de lecture, d'écriture et d'exécution

Le tableau suivant répertorie les symboles pour la définition des autorisations de fichier en mode symbolique. Les symboles peuvent spécifier pour qui les autorisations doivent être définies ou modifiées, l'opération à effectuer et les autorisations à affecter ou modifier.

TABLEAU 1-5 Définition des autorisations de fichiers en mode symbolique

Symbole	Fonction	Description
u	*who*	Utilisateur (propriétaire)
g	*who*	Groupe
o	*who*	Autres
a	*who*	Toutes
=	*operator*	Assigner
+	*operator*	Ajouter
-	*operator*	Supprimer
r	*permissions*	Lecture
w	*permissions*	Ecriture
x	*permissions*	Exécution
l	*permissions*	Verrouillage obligatoire, bit `setgid` activé, bit d'exécution du groupe désactivé
s	*permissions*	Bit `setuid` ou `setgid` activé
t	*permissions*	Sticky bit activé, bit d'exécution pour les autres activé

Les désignations des *who operator permissions* dans la colonne des fonctions spécifient les symboles qui modifient les autorisations du fichier ou du répertoire.

who Spécifie pour qui les autorisations doivent être modifiées.

operator Indique l'opération à effectuer.

permissions Spécifie les autorisations à modifier.

Vous pouvez définir des autorisations spéciales pour un fichier en mode absolu ou en mode symbolique. Cependant, vous devez utiliser le mode symbolique pour définir ou supprimer les autorisations `setuid` sur un répertoire. En mode absolu, vous définissez les autorisations

spéciales en ajoutant une nouvelle valeur octale à la gauche du triplé d'autorisation. Voir Exemple 1-5, “Définition des autorisations de fichiers spéciales en mode absolu”. Le tableau suivant répertorie les valeurs octales pour la définition des autorisations spéciales d'un fichier.

TABLEAU 1-6 Définition des autorisations de fichiers spéciales en mode absolu

Valeur octale	Autorisations de fichiers spéciales
1	Sticky bit
2	`setgid`
4	`setuid`

Utilisation des ACL pour protéger les fichiers UFS

Les protections de fichier UNIX conventionnelles fournissent des autorisations de lecture, d'écriture et d'exécution pour les trois classes d'utilisateur : propriétaire de fichier, groupe de fichier et autre. Dans un système de fichiers UFS, une liste de contrôle d'accès (ACL) offre une meilleure sécurité des fichiers en permettant d'effectuer les opérations suivantes :

- Définir les autorisations de fichier pour le propriétaire du fichier, le groupe, les autres, ainsi que des utilisateurs et des groupes spécifiques
- Définir les autorisations par défaut pour chacune des catégories précédentes.

Remarque - Pour les ACL dans le système de fichiers ZFS et les ACL sur les fichiers NFSv4, reportez-vous au Chapitre 7, “ Utilisation des ACL et des attributs pour protéger les fichiers Oracle Solaris ZFS ” du manuel “ Gestion des systèmes de fichiers ZFS dans OracleSolaris 11.2 ”.

Par exemple, si vous souhaitez que tous les membres d'un groupe soient en mesure de lire un fichier, vous pouvez simplement accorder des autorisations de lecture de groupe sur ce fichier. Toutefois, si vous souhaitez qu'une seule personne du groupe soit en mesure d'écrire dans ce fichier, vous pouvez utiliser un ACL.

Pour plus d'informations relatives aux ACL sur les systèmes de fichiers UFS, reportez-vous au *System Administration Guide: Security Services* for the Version Oracle Solaris 10.

Protection contre les problèmes de sécurité causés par les fichiers exécutables

Les programmes lisent et écrivent des données sur la pile. D'une manière générale, ils s'exécutent à partir de portions de mémoire en lecture seule qui sont spécifiquement désignées

pour ce code. Certaines attaques provoquant des tampons sur la pile jusqu'au débordement tentent d'insérer un nouveau code sur la pile et forcent le programme à l'exécuter. Ces attaques peuvent être mises en échec en supprimant les autorisations d'exécution de la Suppression de droits d'exécution la mémoire de la pile. C'est-à-dire que la plupart de ces programmes peuvent fonctionner correctement sans utiliser les piles exécutables.

Les processus 64 bits disposent toujours de piles non exécutables. Par défaut, les processus SPARC 32 bits possèdent des piles exécutables. La variable `noexec_user_stack` vous permet d'indiquer si les piles des processus 32 bits sont exécutables.

Une fois que cette variable est définie, un signal `SIGSEGV` est envoyé aux programmes qui tentent d'exécuter du code sur leur pile. Ce signal résulte généralement en un arrêt du programme à l'aide d'un dump noyau. Ces programmes génèrent également un message d'avertissement qui inclut le nom du programme concerné, l'ID de processus, et l'UID réel de l'utilisateur à l'origine de l'exécution du programme. Par exemple :

```
a.out[347] attempt to execute code on stack by uid 555
```

Le message est consigné par le démon `syslog` lorsque la fonctionnalité `syslog kern` est définie sur le niveau `notice`. Cet enregistrement est défini par défaut dans le fichier `syslog.conf`, ce qui signifie que le message est envoyé à la console et au fichier `/var/adm/messages`. Pour plus d'informations, reportez-vous aux pages de manuel `syslogd`(1M) et `syslog.conf`(4).

Le message `syslog` est utile pour observer les potentiels problèmes de sécurité. Le message identifie également les programmes valides qui dépendent des piles exécutables dont le bon fonctionnement est empêché par la définition de la variable`noexec_user_stack`. Si vous ne voulez pas que les messages soient consignés, définissez la variable de journalisation `noexec_user_stack_log` sur zéro dans le fichier `/etc/system`. Bien que les messages ne soient pas consignés, le signal `SIGSEGV` peut continuer d'entraîner l'arrêt du programme d'exécution à l'aide d'un dump noyau.

Les programmes peuvent marquer explicitement ou empêcher l'exécution des piles. Dans les programmes, la fonction `mprotect()` marque explicitement les piles comme étant exécutables. Pour plus d'informations, reportez-vous à la page de manuel `mprotect`(2). Un programme compilé avec `-M /usr/lib/ld/map.noexstk` rend la pile non exécutable indépendamment du paramètre à l'échelle du système.

Protection des fichiers

Les procédures d'installation suivantes permettent de protéger les fichiers avec des autorisations UNIX, localiser des fichiers présentant des risques liés à la sécurité et de protéger le système contre toute compromission par ces fichiers.

Protection des fichiers à l'aide d'autorisations UNIX

La liste des tâches suivante présente les procédures permettant de répertorier les autorisations de fichiers, des les modifier et de protéger les fichiers avec les autorisations de fichiers spéciales.

Tâche	Voir
Affichage des informations de fichier.	“Affichage des informations de fichier” à la page 16
Modification des propriétaires de fichier locaux.	“Modification du propriétaire d'un fichier” à la page 17 “Modification de la propriété de groupe d'un fichier” à la page 18
Modification des autorisations de fichiers locaux.	“Modification des autorisations de fichier en mode symbolique” à la page 19 “Modification des autorisations de fichier en mode absolu” à la page 20 “Modification des autorisations de fichier spéciales en mode absolu” à la page 21

▼ Affichage des informations de fichier

Affichez les informations sur tous les fichiers d'un répertoire en utilisant la commande `ls`.

- **Tapez la commande suivante pour afficher une longue liste de tous les fichiers dans le répertoire en cours.**

```
% ls -la
```

`-l` Affiche le format long qui inclut la propriété d'utilisateur, la propriété de groupe et les autorisations du fichier.

`-a` Affiche tous les fichiers, y compris les fichiers cachés qui commencent par un point (`.`).

Pour connaître toutes les options de la commande `ls`, reportez-vous à la page de manuel `ls`(1).

Exemple 1-1 Affichage des informations de fichier

Dans l'exemple suivant, une liste partielle de fichiers dans le répertoire `/sbin` s'affiche.

```
% cd /sbin
```

```
% ls -l
total 4960
-r-xr-xr-x   1 root   bin      12756 Dec 19  2013 6to4relay
lrwxrwxrwx   1 root   root        10 Dec 19  2013 accept -> cupsaccept
-r-xr-xr-x   1 root   bin      38420 Dec 19  2013 acctadm
-r-xr-xr-x   2 root   sys      70512 Dec 19  2013 add_drv
-r-xr-xr-x   1 root   bin       3126 Dec 19  2013 addgnupghome
drwxr-xr-x   2 root   bin         37 Dec 19  2013 amd64
-r-xr-xr-x   1 root   bin       2264 Dec 19  2013 applygnupgdefaults
-r-xr-xr-x   1 root   bin        153 Dec 19  2013 archiveadm
-r-xr-xr-x   1 root   bin      12644 Dec 19  2013 arp
.
.
.
```

Chaque ligne affiche des informations sur un fichier dans l'ordre suivant :

- Type de fichier : par exemple, `d`. Pour obtenir la liste des types de fichiers, reportez-vous à la section “Propriété des fichiers et des répertoires” à la page 8.
- Autorisations : par exemple, `r-xr-xr-x`. Pour obtenir une description, reportez-vous à la section “Propriété des fichiers et des répertoires” à la page 8.
- Nombre de liens fixes : par exemple, `2`.
- Propriétaire du fichier : par exemple, `root`.
- Groupe du fichier : par exemple, `bin`.
- Taille du fichier, en octets : par exemple, `12644`.
- Date à laquelle le fichier a été créé ou modifié pour la dernière fois : par exemple, `Dec 19 2013`.
- Nom du fichier : par exemple, arp.

▼ Modification du propriétaire d'un fichier

Avant de commencer

Si vous n'êtes pas le propriétaire du fichier ou du répertoire, le profil de droits Object Access Management (gestion de l'accès aux objets) doit vous être affecté. Pour modifier un fichier qui est un objet public, vous devez posséder le rôle `root`.

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

1. **Affichez les autorisations sur un fichier local.**

```
% ls -l example-file
-rw-r--r--   1 janedoe   staff   112640 May 24 10:49 example-file
```

2. **Modifiez le propriétaire du fichier.**

```
# chown stacey example-file
```

3. **Vérifiez que le propriétaire du fichier a bien été modifié.**

```
# ls -l example-file
-rw-r--r--   1 stacey   staff   112640 May 26 08:50 example-file
```

Pour modifier les autorisations sur les fichiers montés via NFS, reportez-vous au Chapitre 5, " Commandes de gestion des systèmes de fichiers NFS " du manuel " Gestion des systèmes de fichiers réseau dans Oracle Solaris 11.2 ".

Exemple 1-2 Modification par les utilisateurs de la propriété de leurs propres fichiers

Considération de sécurité : vous devez avoir une bonne raison de modifier la valeur de la variable `rstchown` à zéro. Le paramètre par défaut empêche les utilisateurs de répertorier leurs fichiers comme appartenant à d'autres utilisateurs afin de contourner les quotas d'espace.

Dans cet exemple, la valeur de la variable `rstchown` est définie sur zéro dans le fichier `/etc/system`. Ce paramètre permet au propriétaire d'un fichier d'utiliser la commande `chown` pour modifier la propriété du fichier à un autre utilisateur. Ce paramètre permet également au propriétaire d'utiliser la commande `chgrp` pour définir le groupe propriétaire d'un fichier sur un groupe auquel dont le propriétaire n'appartient pas. Le changement entre en vigueur lors du redémarrage du système.

```
set rstchown = 0
```

Pour plus d'informations, reportez-vous aux pages de manuel `chown`(1) et `chgrp`(1).

▼ Modification de la propriété de groupe d'un fichier

Avant de commencer Si vous n'êtes pas le propriétaire du fichier ou du répertoire, le profil de droits Object Access Management (gestion de l'accès aux objets) doit vous être affecté. Pour modifier un fichier qui est un objet public, vous devez posséder le rôle `root`.

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

1. **Modifiez la propriété de groupe d'un fichier.**

```
% chgrp scifi example-file
```

Pour obtenir des informations sur la configuration des groupes, reportez-vous au Chapitre 1, " Gestion des comptes et des environnements utilisateur " du manuel " Gestion des comptes utilisateur et des environnements utilisateur dans Oracle Solaris 11.2 ".

2. **Vérifiez que la propriété de groupe du fichier a changé.**

```
% ls -l example-file
-rw-r--r--   1 stacey   scifi   112640 June 20 08:55  example-file
```

Reportez-vous également à l'Exemple 1-2, “Modification par les utilisateurs de la propriété de leurs propres fichiers”.

▼ Modification des autorisations de fichier en mode symbolique

Dans la procédure ci-dessous, un utilisateur modifie les autorisations d'un fichier qui lui appartient.

1. **Modifiez les autorisations en mode symbolique.**

```
% chmod who operator permissions filename
```

who	Spécifie pour qui les autorisations doivent être modifiées.
operator	Indique l'opération à effectuer.
permissions	Spécifie les autorisations à modifier. Pour obtenir la liste des symboles valides, reportez-vous au Tableau 1-5, “Définition des autorisations de fichiers en mode symbolique”.
filename	Spécifie le fichier ou répertoire.

2. **Vérifiez que les autorisations du fichier ont changé.**

```
% ls -l filename
```

Remarque - Si vous n'êtes pas le propriétaire du fichier ou du répertoire, le profil de droits Object Access Management (gestion de l'accès aux objets) doit vous être affecté. Pour modifier un fichier qui est un objet public, vous devez posséder le rôle `root`.

Exemple 1-3 Modification des autorisations en mode symbolique

Dans l'exemple suivant, le propriétaire enlève de droits d'accès en lecture d'autres utilisateurs.

```
% chmod o-r example-file1
```

L'exemple suivant, le propriétaire ajoute les autorisations de l'utilisateur en lecture et en exécution, le groupe et les autres.

```
% chmod a+rx example-file2
```

Dans l'exemple suivant, le propriétaire ajoute droits en lecture, écriture et exécution pour les membres du groupe.

```
% chmod g=rwx example-file3
```

▼ Modification des autorisations de fichier en mode absolu

Dans la procédure ci-dessous, un utilisateur modifie les autorisations d'un fichier qui lui appartient.

1. **Modifiez les autorisations en mode absolu.**

```
% chmod nnn filename
```

nnn	Spécifie les valeurs octales qui représentent les autorisations du propriétaire du fichier, du groupe de fichiers et autres, dans cet ordre. Pour obtenir la liste des valeurs octales, reportez-vous au Tableau 1-4, “Définition des autorisations de fichiers en mode absolu”.
filename	Spécifie le fichier ou répertoire.

Remarque - Si vous utilisez la commande `chmod` pour modifier les autorisations de fichier ou répertoire sur des objets qui possèdent des entrées ACL existantes, celles-ci sont susceptibles d'être modifiées également. Les modifications exactes dépendent des modifications d'autorisation `chmod` et des valeurs des propriétés `aclmode` et `aclinherit` du système de fichiers.

Pour plus d'informations, reportez-vous au Chapitre 7, “ Utilisation des ACL et des attributs pour protéger les fichiers Oracle Solaris ZFS ” du manuel “ Gestion des systèmes de fichiers ZFS dans OracleSolaris 11.2 ”.

2. **Vérifiez que les autorisations du fichier ont changé.**

```
% ls -l filename
```

Remarque - Si vous n'êtes pas le propriétaire du fichier ou du répertoire, le profil de droits Object Access Management (gestion de l'accès aux objets) doit vous être affecté. Pour modifier un fichier qui est un objet public, vous devez posséder le rôle `root`.

Exemple 1-4 Modification des autorisations en mode absolu

Dans l'exemple ci-dessous, l'administrateur modifie les autorisations d'un répertoire ouvert au public, de `744` (lecture, écriture, exécution ; lecture seule ; et lecture seule) en `755` (lecture, écriture, exécution ; lecture et exécution ; et lecture et exécution).

```
# ls -ld public_dir
drwxr--r--  1 jdoe   staff    6023 Aug  5 12:06 public_dir
# chmod 755 public_dir
# ls -ld public_dir
drwxr-xr-x  1 jdoe   staff    6023 Aug  5 12:06 public_dir
```

Dans l'exemple suivant, le propriétaire d'un fichier modifie les autorisations d'un script de shell exécutable, de lecture et écriture en lecture, écriture et exécution.

```
% ls -l my_script
-rw------- 1 jdoe   staff    6023 Aug  5 12:06 my_script
% chmod 700 my_script
% ls -l my_script
-rwx------ 1 jdoe   staff    6023 Aug  5 12:06 my_script
```

▼ Modification des autorisations de fichier spéciales en mode absolu

Avant de commencer

Si vous n'êtes pas le propriétaire du fichier ou du répertoire, le profil de droits Object Access Management (gestion de l'accès aux objets) doit vous être affecté. Pour modifier un fichier qui est un objet public, vous devez posséder le rôle `root`.

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

1. Modifiez les autorisations spéciales en mode absolu.

```
% chmod nnnn filename
```

nnnn — Spécifie les valeurs octales qui modifient les autorisations du fichier ou du répertoire. La valeur octale le plus à gauche définit les autorisations spéciales du fichier. Pour obtenir la liste de valeurs octales valides pour les autorisations spéciales, reportez-vous au Tableau 1-6, "Définition des autorisations de fichiers spéciales en mode absolu".

filename — Spécifie le fichier ou répertoire.

Remarque - Lorsque vous utilisez la commande `chmod` pour modifier les autorisations de groupe sur un fichier avec des entrées d'ACL, les autorisations de groupe de fichiers et le masque d'ACL sont modifiés et reflètent les nouvelles autorisations. N'oubliez pas que les nouvelles autorisations du masque d'ACL peuvent modifier les autorisations d'autres utilisateurs et de groupes qui disposent d'entrées d'ACL sur le fichier. Utilisez la commande `getfacl` pour vous assurer que les autorisations appropriées sont définies pour toutes les entrées d'ACL. Pour plus d'informations, reportez-vous à la page de manuel `getfacl`(1).

2. **Vérifiez que les autorisations du fichier ont changé.**

```
% ls -l filename
```

Exemple 1-5 Définition des autorisations de fichiers spéciales en mode absolu

Dans l'exemple suivant, l'administrateur définit les droits d'accès sur le fichier `setuiddbprog`

```
# chmod 4555 dbprog
# ls -l dbprog
-r-sr-xr-x   1 db     staff        12095 May  6 09:29 dbprog
```

Dans l'exemple suivant, l'administrateur définit les droits d'accès sur le fichier `setgiddbprog2`

```
# chmod 2551 dbprog2
# ls -l dbprog2
-r-xr-s--x   1 db     staff       24576 May  6 09:30 dbprog2
```

Dans l'exemple suivant, l'administrateur définit le sticky bit sur le répertoire `public_dir`.

```
# chmod 1777 public_dir
# ls -ld public_dir
drwxrwxrwt   2 jdoe   staff          512 May 15 15:27 public_dir
```

Protection contre les programmes présentant des risques de sécurité

La liste des tâches suivante présente les procédures permettant trouver les exécutables à risque dans le système, et qui empêchent les programmes d'exploiter une pile exécutable.

Tâche	Description	Voir
Recherche de fichiers avec des autorisations spéciales.	Localise les fichiers avec l'ensemble de bits `setuid`, mais non détenus par l'utilisateur `root`.	“Recherche de fichiers avec des autorisations de fichier spéciales” à la page 23

Tâche	Description	Voir
Empêchement du débordement de pile exécutable.	Empêche les programmes d'exploiter une pile exécutable.	“Désactivation de l'utilisation de piles exécutables par les programmes” à la page 24
Empêchement de la journalisation des messages de pile exécutable.	Désactive la journalisation des messages de pile exécutable.	Exemple 1-7, “Désactivation de la journalisation des messages de pile exécutable”

▼ Recherche de fichiers avec des autorisations de fichier spéciales

Cette procédure permet de localiser les utilisations non autorisées potentielles des autorisations `setuid` et `setgid` sur les programmes. Un fichier exécutable suspect accorde la propriété à un utilisateur plutôt qu'à `root` ou `bin`.

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

1. **Recherchez les fichiers avec des autorisations `setuid` en utilisant la commande `find`.**

   ```
   # find directory -user root -perm -4000 -exec ls -ldb {} \; >/tmp/filename
   ```

`find` *directory*	Vérifie tous les chemins montés en commençant par le *répertoire* spécifié, qui peut être root (`/`), `/usr`, `/opt`, etc.
`-user root`	Affiche les fichiers appartenant uniquement à `root`.
`-perm -4000`	Affiche les fichiers uniquement avec les autorisations définies sur `4000`.
`-exec ls -ldb`	Affiche le résultat de la commande `find` au format `ls -ldb`. Reportez-vous à la page de manuel `ls`(1).
`/tmp/`*filename*	Il s'agit du fichier qui contient les résultats de la commande `find`.

   Pour plus d'informations, reportez-vous à `find`(1).

2. **Affichez les résultats dans `/tmp/`*filename*.**

   ```
   # more /tmp/filename
   ```

   Pour obtenir des informations générales, reportez-vous à la section “Autorisation `setuid`” à la page 10.

Exemple 1-6 Recherche de fichiers avec des autorisations `setuid`

La sortie de l'exemple suivant montre qu'un utilisateur d'un groupe appelé `rar` a effectué une copie personnelle de `/usr/bin/rlogin` et a défini les autorisations `setuid` sur `root`. Par conséquent, le programme `/usr/rar/bin/rlogin` s'exécute avec les autorisations `root`.

Après avoir examiné le répertoire `/usr/rar` et supprimé la commande `/usr/rar/bin/rlogin`, l'administrateur archive la sortie de la commande `find`.

```
# find /usr -user root -perm -4000 -exec ls -ldb {} \; > /var/tmp/ckprm
# cat /var/tmp/ckprm
-rwsr-xr-x  1 root   sys   28000  Jul 14 14:14  /usr/bin/atq
-rwsr-xr-x  1 root   sys   32364  Jul 14 14:14  /usr/bin/atrm
-r-sr-xr-x  1 root   sys   41432  Jul 14 14:14  /usr/bin/chkey
-rwsr-xr-x  1 root   bin   82804  Jul 14 14:14  /usr/bin/cdrw
-r-sr-xr-x  1 root   bin    8008  Jul 14 14:14  /usr/bin/mailq
-r-sr-sr-x  1 root   sys   45348  Jul 14 14:14  /usr/bin/passwd
-rwsr-xr-x  1 root   bin   37724  Jul 14 14:14  /usr/bin/pfedit
-r-sr-xr-x  1 root   bin   51440  Jul 14 14:14  /usr/bin/rcp
---s--x---  1 root   rar   41592  Jul 24 16:14  /usr/rar/bin/rlogin
-r-s--x--x  1 root   bin  166908  Jul 14 14:14  /usr/bin/sudo
-r-sr-xr-x  4 root   bin   24024  Jul 14 14:14  /usr/bin/uptime
-r-sr-xr-x  1 root   bin   79488  Jul 14 14:14  /usr/bin/xlock
# mv /var/tmp/ckprm /var/share/sysreports/ckprm
```

▼ Désactivation de l'utilisation de piles exécutables par les programmes

Pour obtenir une description des risques de sécurité liés aux piles exécutables 32 bits, reportez-vous à la section “Protection contre les problèmes de sécurité causés par les fichiers exécutables” à la page 14.

Avant de commencer

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

1. **Modifiez le fichier `/etc/system`, puis ajoutez les lignes suivantes :**

```
# pfedit /etc/system
...
set noexec_user_stack=1
set noexec_user_stack_log=1
```

2. **Réinitialisez le système.**

```
# reboot
```

Exemple 1-7 Désactivation de la journalisation des messages de pile exécutable

Dans cet exemple, l'administrateur désactive la journalisation des messages de pile exécutable, au redémarrage du système.

```
# cat /etc/system
set noexec_user_stack=1
set noexec_user_stack_log=0
# reboot
```

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

- https://blogs.oracle.com/gbrunett/entry/solaris_non_executable_stack_overview
- https://blogs.oracle.com/gbrunett/entry/solaris_non_executable_stack_continued
- https://blogs.oracle.com/gbrunett/entry/solaris_non_executable_stack_concluded

♦ ♦ ♦ CHAPITRE 2

Vérification de l'intégrité des fichiers à l'aide de l'utilitaire BART

Ce chapitre décrit l'outil d'intégrité des fichiers, BART. BART est un outil de ligne de commande qui permet de vérifier l'intégrité des fichiers sur un système dans le temps. Ce chapitre comprend les sections suivantes :

A propos de BART

BART est un outil de génération de rapports et d'analyse de l'intégrité des fichiers qui utilise des sommes de contrôle cryptographiques et les métadonnées du système de fichiers pour déterminer les modifications. BART peut vous aider à détecter des violations de sécurité et résoudre les problèmes de performances sur un système en identifiant les fichiers endommagés ou inhabituels. L'utilisation de BART peut réduire les coûts d'administration d'un réseau de systèmes en créant facilement et de façon fiable des rapports sur les écarts dans les fichiers installés sur les systèmes déployés.

BART vous permet de déterminer les modifications survenues au niveau des fichiers sur un système, par rapport à une ligne de base connue. Utilisez BART pour créer une ligne de base ou un *manifeste de contrôle* à partir d'un système entièrement installé et configuré. Vous pouvez comparer cette ligne de base à un instantané du système à un moment ultérieur, générant ainsi un rapport qui répertorie les modifications au niveau des fichiers survenues sur le système depuis son installation.

Fonctionnalités BART

BART utilise une syntaxe simple, à la fois puissante et flexible. L'outil vous permet d'effectuer le suivi des modifications apportées aux fichiers sur un système donné au fil du temps. Vous pouvez également suivre les différences entre les fichiers de systèmes similaires. Ces

comparaisons peuvent vous aider à localiser les fichiers endommagés ou inhabituels, ou les systèmes dont les logiciels sont obsolètes.

Les utilisations et avantages supplémentaires de BART sont les suivants :

- Vous pouvez spécifier les fichiers à surveiller. Par exemple, vous pouvez surveiller les personnalisations locales, qui peuvent vous aider à reconfigurer les logiciels de façon simple et efficace.
- Vous pouvez résoudre les problèmes de performances du système.

Composants BART

BART crée deux fichiers principaux, un *manifeste* et un fichier de comparaison, ou *rapport*. Un *fichier de règles* vous permet de personnaliser le manifeste et le rapport.

Manifeste BART

Un *manifeste* est un instantané au niveau des fichiers d'un système à un moment donné. Il contient des informations sur les attributs des fichiers, pouvant inclure des informations d'identification uniques, comme par exemple une somme de contrôle. Les options de la commande `bart create` peuvent cibler des fichiers et répertoires spécifiques. Un fichier de règles permet d'utiliser des filtres plus précis, comme décrit à la section “Fichier de règles BART” à la page 29.

Remarque - Par défaut, BART répertorie tous les systèmes de fichiers ZFS sous le répertoire root (`/`). D'autres types de système de fichiers (NFS ou TMPFS, par exemple) et les CD-ROM montés sont classifiés.

Vous pouvez créer un manifeste d'un système immédiatement après une première installation d'Oracle Solaris. Vous pouvez également créer un manifeste après avoir configuré le système afin qu'il réponde à la stratégie de sécurité de votre site. Ce type de manifeste de contrôle vous fournit une base en vue de comparaisons ultérieures.

Un manifeste de base peut être utilisé pour effectuer le suivi de l'intégrité des fichiers sur le même système dans le temps. Il peut également servir de base pour la comparaison avec d'autres systèmes. Par exemple, vous pouvez créer un instantané d'autres systèmes de votre réseau, puis comparer ces manifestes avec le manifeste de base. Les écarts signalés entre fichiers indiquent ce que vous devez faire pour synchroniser les autres systèmes avec le système de base.

Pour connaître le format d'un manifeste, reportez-vous à la section “Format de fichier manifeste BART” à la page 41. Pour créer un manifeste, utilisez la commande `bart create`, comme décrit dans la section “Création d'un manifeste de contrôle” à la page 31.

Rapport BART

Un rapport BART répertorie les écarts par fichier entre deux manifestes. Un *écart* est une modification apportée à un attribut d'un fichier donné, répertorié pour les deux manifestes. Les ajouts ou suppressions d'entrées de fichiers sont également considérés comme des écarts.

Pour une comparaison utile, les deux manifestes doivent cibler les mêmes systèmes de fichiers. Vous devez également créer et comparer les manifestes avec les mêmes options et le même fichier de règles.

Pour connaître le format d'un rapport, reportez-vous à la section “Génération de rapports BART” à la page 43. Pour créer un rapport, utilisez la commande `bart compare`, comme décrit dans la section “Comparaison des manifestes pour le même système dans le temps” à la page 34.

Fichier de règles BART

Un fichier de règles BART est un fichier que vous créez pour filtrer ou cibler certains fichiers et attributs de fichiers à des fins d'inclusion ou d'exclusion. Vous utilisez ensuite ce fichier lors de la création des manifestes et des rapports BART. Lorsque vous comparez des manifestes, le fichier de règles facilite le marquage des écarts entre les manifestes.

Remarque - Lorsque vous créez un manifeste à l'aide d'un fichier de règles, vous devez utiliser le même fichier de règles pour créer le manifeste de comparaison. Vous devez également utiliser le fichier de règles pour comparer les manifestes. Sinon, le rapport répertorie de nombreux écarts incorrects.

L'utilisation d'un fichier de règles pour surveiller des fichiers spécifiques et des attributs de fichiers sur un système requiert une planification. Avant de créer un fichier de règles, déterminez les fichiers et attributs de fichiers du système que vous souhaitez surveiller.

Suite à une erreur de l'utilisateur, un fichier de règles peut également contenir des erreurs de syntaxe et autres informations ambiguës. Si un fichier de règles contient des erreurs, ces erreurs sont également signalées.

Pour connaître le format d'un fichier de règles, reportez-vous à la section “Format de fichier de règles BART” à la page 42 et à la page de manuel `bart_rules`(4). Pour créer un fichier de règles, reportez-vous à la section “Personnalisation d'un rapport BART en utilisant un fichier de règles” à la page 39.

A propos de l'utilisation de BART

La commande `bart` permet de créer et comparer des manifestes. Tous les utilisateurs peuvent exécuter cette commande. Toutefois, les utilisateurs peuvent uniquement répertorier et contrôler les fichiers pour lesquels ils disposent de droits d'accès. Par conséquent, les utilisateurs et la plupart des rôles peuvent répertorier utilement les fichiers dans leur répertoire personnel, mais le compte `root` permet de répertorier tous les fichiers, y compris les fichiers système.

Considérations de sécurité BART

Les manifestes et rapports BART sont lisibles par n'importe quel utilisateur. Si la sortie BART contient des informations confidentielles, protégez-la en effectuant les actions nécessaires. Par exemple, utilisez les options générant des fichiers de sortie avec des autorisations restreintes ou placez les fichiers de sortie dans un répertoire protégé.

Utilisation de BART

Tâche	Description	Voir
Création d'un manifeste BART.	Génère une liste d'informations sur chaque fichier installé sur un système.	“Création d'un manifeste de contrôle” à la page 31
Création d'un manifeste BART personnalisé.	Génère une liste d'informations sur des fichiers spécifiques installés sur un système.	“Personnalisation d'un manifeste” à la page 33
Comparaison de manifestes BART.	Génère un rapport qui compare les modifications apportées à un système dans le temps. Ou génère un rapport qui compare un ou plusieurs systèmes à un système de contrôle.	“Comparaison des manifestes pour le même système dans le temps” à la page 34 “Comparaison de manifestes de différents systèmes” à la page 36
(Facultatif) Personnalisation d'un rapport BART.	Génère un rapport BART personnalisé de l'une des manières suivantes : ■ En spécifiant les attributs ■ En utilisant un fichier de règles	“Personnalisation d'un rapport BART en spécifiant des attributs de fichiers” à la page 39 “Personnalisation d'un rapport BART en utilisant un fichier de règles” à la page 39

▼ Création d'un manifeste de contrôle

La procédure suivante décrit la création d'une ligne de base, ou, le manifeste de contrôle pour la comparaison. Utilisez ce type de manifeste lorsque vous installez de nombreux systèmes à partir d'une image centrale. Vous pouvez également utiliser ce type de manifeste pour effectuer des comparaisons lorsque vous souhaitez vérifier que les installations sont identiques. Pour plus d'informations sur les manifestes de contrôle, reportez-vous à la section “Manifeste BART” à la page 28. Pour connaître les conventions de format, reportez-vous à la section Exemple 2-1, “Explication du format de manifeste BART”.

Remarque - Ne tentez pas de répertorier les systèmes de fichiers en réseau. L'utilisation de BART dans le cadre de la surveillance des systèmes de fichiers en réseau consomme d'importantes ressources pour générer des manifestes de faible valeur.

Avant de commencer

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

1. **Après avoir personnalisé votre système Oracle Solaris en fonction des exigences de sécurité de votre site, créez un manifeste de contrôle et redirigez la sortie vers un fichier.**

   ```
   # bart create options > control-manifest
   ```

`-R`	Spécifie le répertoire root pour le manifeste. Tous les chemins d'accès spécifiés par les règles sont interprétés par rapport à ce répertoire. Tous les chemins d'accès signalés dans le manifeste sont relatifs à ce répertoire.
`-I`	Accepte une liste de fichiers individuels à classifier, soit sur la ligne de commande soit à lire dans l'entrée standard.
`-r`	Nom du fichier de règles pour ce manifeste. Un argument `-` lit le fichier de règles de l'entrée standard.
`-n`	Désactive les signatures de contenu de tous les fichiers standard dans la liste de fichiers. Cette option peut être utilisée pour améliorer les performances. Ou bien vous pouvez l'utiliser s'il est prévu que le contenu de la liste de fichiers change, comme dans le cas des fichiers journaux du système.

2. **Examinez le contenu du manifeste.**

   Pour obtenir une explication sur le format, reportez-vous à l'Exemple 2-1, “Explication du format de manifeste BART”.

3. **(Facultatif) Protégez le manifeste.**

   Pour protéger les manifestes système, une méthode consiste à les placer dans un répertoire auquel seul le compte `root` peut accéder.

   ```
   # mkdir /var/adm/log/bartlogs
   # chmod 700 /var/adm/log/bartlogs
   # mv control-manifest /var/adm/log/bartlogs
   ```

   Choisissez un nom explicite pour le manifeste. Par exemple, utilisez le nom du système et la date de création du manifeste comme dans `mach1-120313`.

Exemple 2-1 Explication du format de manifeste BART

Dans cet exemple, l'échantillon de sortie est suivi d'une explication du format de manifeste.

```
# bart create
! Version 1.1
! HASH SHA256
! Saturday, September 07, 2013 (22:22:27)
# Format:
#fname D size mode acl dirmtime uid gid
#fname P size mode acl mtime uid gid
#fname S size mode acl mtime uid gid
#fname F size mode acl mtime uid gid contents
#fname L size mode acl lnmtime uid gid dest
#fname B size mode acl mtime uid gid devnode
#fname C size mode acl mtime uid gid devnode
/ D 1024 40755 user::rwx,group::r-x,mask:r-x,other:r-x
3ebc418eb5be3729ffe7e54053be2d33ee884205502c81ae9689cd8cca5b0090 0 0
.
.
.
/zone D 512 40755 user::rwx group::r-x,mask:r-x,other:r-x 3f81e892
154de3e7bdfd0d57a074c9fae0896a9e2e04bebfe5e872d273b063319e57f334 0 0
.
.
.
```

Chaque manifeste se compose d'un en-tête et d'entrées de fichier. Chaque entrée de fichier constitue une seule ligne, selon le type de fichier. Par exemple, pour chaque entrée de fichier dans la sortie ci-dessus, le type `F` indique un fichier et le type `D` un répertoire. Sont également répertoriées des informations sur la taille, le contenu, l'ID utilisateur, l'ID de groupe et les autorisations. Les entrées de fichier dans la sortie sont triées par versions codées des noms de fichier de sorte à traiter correctement les caractères spéciaux. Toutes les entrées sont stockées dans l'ordre croissant par nom de fichier. Pour tous les noms de fichiers non standard, tels que ceux contenant des caractères de retour à la ligne ou de tabulation, les caractères non standard sont mis entre guillemets avant que le tri ne soit effectué.

Les lignes commençant par `!` fournissent des métadonnées sur le manifeste. La ligne de version du manifeste indique la version de spécification du manifeste. La ligne de hachage indique le

mécanisme de hachage qui a été utilisé. Pour plus d'informations sur le hachage `SHA256` utilisé en tant que somme de contrôle, reportez-vous à la page de manuel `sha2`(3EXT).

La ligne de date indique la date de création du manifeste. Reportez-vous à la page de manuel `date`(1). Certaines lignes sont ignorées par l'outil de comparaison de manifestes. Les lignes ignorées incluent des métadonnées, des lignes vides, des lignes composées uniquement d'espaces blancs et des commentaires commençant par `#`.

▼ Personnalisation d'un manifeste

Vous pouvez personnaliser un manifeste de l'une des façons suivantes :

- En spécifiant une sous-arborescence

  Spécifier une sous-arborescence est un moyen efficace de surveiller les modifications apportées aux fichiers importants sélectionnés, tels que tous les fichiers du répertoire `/etc`.
- En spécifiant un nom de fichier

  Spécifier un nom de fichier est un moyen efficace pour surveiller les fichiers particulièrement sensibles, tels que les fichiers de configuration et d'exécution d'une application de base de données.
- En utilisant un fichier de règles

  L'utilisation d'un fichier de règles pour créer et comparer des manifestes vous donne la possibilité de spécifier des attributs multiples pour plusieurs fichiers ou sous-arborescences. Sur la ligne de commande, vous pouvez spécifier une définition d'attribut générale s'appliquant à tous les fichiers inclus dans un manifeste ou un rapport. A partir d'un fichier de règles, vous pouvez indiquer les attributs qui ne s'appliquent pas au niveau global.

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

1. **Déterminez les fichiers que vous souhaitez classifier et surveiller.**
2. **Créez un manifeste personnalisé à l'aide de l'une des options suivantes :**

   - En spécifiant une sous-arborescence :

     ```
     # bart create -R subtree
     ```
   - En spécifiant un ou des noms de fichiers :

     ```
     # bart create -I filename...
     ```

     Par exemple :

     ```
     # bart create -I /etc/system /etc/passwd /etc/shadow
     ```

- En utilisant un fichier de règles :

  ```
  # bart create -r rules-file
  ```

3. **Examinez le contenu du manifeste.**

4. **(Facultatif) Enregistrez le manifeste dans un répertoire protégé en vue d'une utilisation ultérieure.**

   Pour obtenir un exemple, reportez-vous à l'Étape 3 in “Création d'un manifeste de contrôle” à la page 31.

Astuce - Si vous avez utilisé un fichier de règles, enregistrez-le avec le manifeste. Pour que la comparaison soit utile, vous devez l'exécuter avec le fichier de règles.

▼ Comparaison des manifestes pour le même système dans le temps

En comparant les manifestes au fil du temps, vous pouvez localiser les fichiers endommagés ou inhabituels, détecter les violations de sécurité et résoudre les problèmes de performances sur un système.

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

1. **Créez un manifeste de contrôle des fichiers pour contrôler le système.**

   ```
   # bart create -R /etc > control-manifest
   ```

2. **(Facultatif) Enregistrez le manifeste dans un répertoire protégé en vue d'une utilisation ultérieure.**

   Pour obtenir un exemple, reportez-vous à l'Étape 3 in “Création d'un manifeste de contrôle” à la page 31.

3. **Plus tard, préparez un manifeste identique au manifeste de contrôle.**

   ```
   # bart create -R /etc > test-manifest
   ```

4. **Protégez le second manifeste.**

   ```
   # mv test-manifest /var/adm/log/bartlogs
   ```

5. **Comparez les deux manifestes.**

Utilisez les mêmes options de ligne de commande et le même fichier de règles pour comparer les manifestes qui vous avez utilisés pour les créer.

```
# bart compare options control-manifest test-manifest > bart-report
```

6. Recherchez les singularités dans le rapport BART

Exemple 2-2 Suivi des modifications de fichier pour le même système au fil du temps

Cet exemple illustre le suivi des modifications dans le répertoire `/etc` au fil du temps. Ce type de comparaison vous permet de localiser sur le système les fichiers importants qui ont été compromis.

- Créez un manifeste de contrôle.

```
# cd /var/adm/logs/manifests
# bart create -R /etc > system1.control.090713
! Version 1.1
! HASH SHA256
! Saturday, September 07, 2013 (11:11:17)
# Format:
#fname D size mode acl dirmtime uid gid
#fname P size mode acl mtime uid gid
#fname S size mode acl mtime uid gid
#fname F size mode acl mtime uid gid contents
#fname L size mode acl lnmtime uid gid dest
#fname B size mode acl mtime uid gid devnode
#fname C size mode acl mtime uid gid devnode
/.cpr_config F 2236 100644 owner@:read_data/write_data/append_data/read_xattr/wr
ite_xattr/read_attributes/write_attributes/read_acl/write_acl/write_owner/synchr
onize:allow,group@:read_data/read_xattr/read_attributes/read_acl/synchronize:all
ow,everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
4e271c59 0 0 3ebc418eb5be3729ffe7e54053be2d33ee884205502c81ae9689cd8cca5b0090
/.login F 1429 100644 owner@:read_data/write_data/append_data/read_xattr/write_x
attr/read_attributes/write_attributes/read_acl/write_acl/write_owner/synchronize
:allow,group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow,ev
eryone@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
4bf9d6d7 0 3 ff6251a473a53de68ce8b4036d0f569838cff107caf1dd9fd04701c48f09242e
.
.
.
```

- Plus tard, créez un manifeste test en utilisant les mêmes options de ligne de commande.

```
# bart create -R /etc > system1.test.101013
Version 1.1
! HASH SHA256
! Monday, October 10, 2013 (10:10:17)
```

```
# Format:
#fname D size mode acl dirmtime uid gid
#fname P size mode acl mtime uid gid
#fname S size mode acl mtime uid gid
#fname F size mode acl mtime uid gid contents
#fname L size mode acl lnmtime uid gid dest
#fname B size mode acl mtime uid gid devnode
#fname C size mode acl mtime uid gid devnode
/.cpr_config F 2236 100644 owner@:read_data/write_data/append_data/read_xattr/wr
ite_xattr/read_attributes/write_attributes/read_acl/write_acl/write_owner/synchr
onize:allow,group@:read_data/read_xattr/read_attributes/read_acl/synchronize:all
ow,everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
4e271c59 0 0 3ebc418eb5be3729ffe7e54053be2d33ee884205502c81ae9689cd8cca5b0090
.
.
.
```

- Comparez les manifestes.

```
# bart compare system1.control.090713 system1.test.101013
/security/audit_class
mtime  4f272f59
```

La sortie indique que le temps de modification sur le fichier `audit_class` a changé depuis que le manifeste de contrôle a été créé. Si cette modification est inattendue, vous pouvez effectuer une analyse plus approfondie.

▼ Comparaison de manifestes de différents systèmes

En comparant des manifestes de différents systèmes, vous pouvez déterminer si les systèmes sont installés de façon identique ou s'ils ont été mis à niveau de façon synchronisée. Par exemple, si vous avez personnalisé vos systèmes en fonction d'un objectif de sécurité spécifique, cette comparaison recherche tous les écarts entre le manifeste qui représente cet objectif et les manifestes des autres systèmes.

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

1. Créez un manifeste de contrôle.

```
# bart create options > control-manifest
```

Pour connaître les options, reportez-vous à la page de manuel `bart`(1M).

2. **(Facultatif) Enregistrez le manifeste dans un répertoire protégé en vue d'une utilisation ultérieure.**

   Pour obtenir un exemple, reportez-vous à l'Étape 3 in “Création d'un manifeste de contrôle” à la page 31.

3. **Sur le système test, utilisez les mêmes options `bart` pour créer un manifeste.**

   ```
   # bart create options > test1-manifest
   ```

4. **(Facultatif) Enregistrez le manifeste dans un répertoire protégé en vue d'une utilisation ultérieure.**

5. **Pour effectuer la comparaison, copiez les manifestes dans un emplacement central.**

   Par exemple :

   ```
   # cp control-manifest /net/test-server/var/adm/logs/bartlogs
   ```

   Si le système test n'est pas un système monté via NFS, utilisez `sftp` ou un autre moyen fiable pour copier les manifestes à un emplacement central.

6. **Comparez les manifestes et redirigez la sortie vers un fichier.**

   ```
   # bart compare control-manifest test1-manifest > test1.report
   ```

7. **Recherchez les singularités dans le rapport BART**

Exemple 2-3 Identification d'un fichier suspect dans le répertoire `/usr/bin`

Cet exemple compare le contenu du répertoire `/usr/bin` sur les deux systèmes.

- Créez un manifeste de contrôle.

```
# bart create -R /usr/bin > control-manifest.090713
! Version 1.1
! HASH SHA256
! Saturday, September 07, 2013 (11:11:17)
# Format:
#fname D size mode acl dirmtime uid gid
#fname P size mode acl mtime uid gid
#fname S size mode acl mtime uid gid
#fname F size mode acl mtime uid gid contents
#fname L size mode acl lnmtime uid gid dest
#fname B size mode acl mtime uid gid devnode
#fname C size mode acl mtime uid gid devnode
/2to3 F 105 100555 owner@:read_data/read_xattr/write_xattr/execute/read_attribut
es/write_attributes/read_acl/write_acl/write_owner/synchronize:allow,group@:read
```

```
_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow,everyone@:re
ad_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow 4bf9d261 0
2 154de3e7bdfd0d57a074c9fae0896a9e2e04bebfe5e872d273b063319e57f334
/7z F 509220 100555 owner@:read_data/read_xattr/write_xattr/execute/read_attribu
tes/write_attributes/read_acl/write_acl/write_owner/synchronize:allow,group@:rea
d_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow,everyone@:r
ead_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow 4dadc48a 0
2 3ecd418eb5be3729ffe7e54053be2d33ee884205502c81ae9689cd8cca5b0090
...
```

- Créez un manifeste identique pour chaque système que vous souhaitez comparer avec le système de contrôle.

```
# bart create -R /usr/bin > system2-manifest.101013
! Version 1.1
! HASH SHA256
! Monday, October 10, 2013 (10:10:22)
# Format:
#fname D size mode acl dirmtime uid gid
#fname P size mode acl mtime uid gid
#fname S size mode acl mtime uid gid
#fname F size mode acl mtime uid gid contents
#fname L size mode acl lnmtime uid gid dest
#fname B size mode acl mtime uid gid devnode
#fname C size mode acl mtime uid gid devnode
/2to3 F 105 100555 owner@:read_data/read_xattr/write_xattr/execute/read_attribut
es/write_attributes/read_acl/write_acl/write_owner/synchronize:allow,group@:read
_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow,everyone@:re
ad_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow 4bf9d261 0
2 154de3e7bdfd0d57a074c9fae0896a9e2e04bebfe5e872d273b063319e57f334
...
```

- Copiez les manifestes dans le même emplacement.

```
# cp control-manifest.090713 /net/system2.central/bart/manifests
```

- Comparez les manifestes.

```
# bart compare control-manifest.090713 system2.test.101013 > system2.report
/su:
gid  control:3  test:1
/ypcat:
mtime  control:3fd72511  test:3fd9eb23
```

La sortie indique que l'ID de groupe du fichier `su` dans le répertoire `/usr/bin` est différent de celui du système de contrôle. Ces informations peuvent indiquer qu'une version différente du logiciel a été installée sur le système test. L'ID de groupe est différent car quelqu'un a probablement modifié le fichier.

▼ Personnalisation d'un rapport BART en spécifiant des attributs de fichiers

Cette procédure est utile pour filtrer la sortie des manifestes existants afin de vérifier des attributs de fichier spécifiques.

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

1. **Déterminez les attributs de fichier à vérifier.**
2. **Comparez deux manifestes contenant les attributs de fichier à vérifier.**

   Par exemple :

   ```
   # bart compare -i lnmtime,mtime control-manifest.121513 \
     test-manifest.010514 > bart.report.010514
   ```

   Utilisez une virgule dans la syntaxe de ligne de commande pour séparer chaque attribut de fichier.
3. **Recherchez les singularités dans le rapport BART**

▼ Personnalisation d'un rapport BART en utilisant un fichier de règles

A l'aide d'un fichier de règles, vous pouvez personnaliser un manifeste BART pour des fichiers spécifiques et des attributs de fichier importants. En utilisant différents fichiers de règles sur des manifestes BART par défaut, vous pouvez exécuter différentes comparaisons pour les mêmes manifestes.

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

1. **Déterminez les fichiers et attributs de fichier à surveiller.**
2. **Créez un fichier de règles avec les directives appropriées.**
3. **Créez un manifeste de contrôle avec le fichier de règles que vous avez créé.**

   `# bart create -r` *myrules1-file* `>` *control-manifest*

4. **(Facultatif) Enregistrez le manifeste dans un répertoire protégé en vue d'une utilisation ultérieure.**

   Pour obtenir un exemple, reportez-vous à l'Étape 3 in “Création d'un manifeste de contrôle” à la page 31.

5. **Créez un manifeste identique sur un autre système, ultérieurement, ou les deux.**

   ```
   # bart create -r myrules1-file > test-manifest
   ```

6. **Comparez les manifestes en utilisant le même fichier de règles.**

   ```
   # bart compare -r myrules1-file control-manifest test-manifest > bart.report
   ```

7. **Recherchez les singularités dans le rapport BART**

Exemple 2-4 Utilisation d'un fichier de règles pour personnaliser les manifestes BART et le rapport de comparaison

Le fichier de règles suivant redirige la commande `bart create` pour répertorier tous les attributs des fichiers du répertoire `/usr/bin`. En outre, le fichier de règles indique à la commande `bart compare` de signaler uniquement les modifications de taille et de contenu dans le même répertoire.

```
# Check size and content changes in the /usr/bin directory.
# This rules file only checks size and content changes.
# See rules file example.

IGNORE all
CHECK size contents
/usr/bin
```

- Créez un manifeste de contrôle avec le fichier de règles que vous avez créé.

  ```
  # bart create -r usrbinrules.txt > usr_bin.control-manifest.121013
  ```

- Préparez un manifeste identique chaque fois que vous voulez surveiller les modifications apportées au répertoire `/usr/bin`.

  ```
  # bart create -r usrbinrules.txt > usr_bin.test-manifest.121113
  ```

- Comparez les manifestes en utilisant le même fichier de règles.

  ```
  # bart compare -r usrbinrules.txt usr_bin.control-manifest.121013 \
  usr_bin.test-manifest.121113
  ```

- Examinez la sortie de la commande `bart compare`.

  ```
   /usr/bin/gunzip:  add
  /usr/bin/ypcat:
  delete
  ```

La sortie ci-dessus indique que le fichier `/usr/bin/ypcat` a été supprimé et le fichier `/usr/bin/gunzip` ajouté.

Manifestes BART, fichiers de règles et rapports

Cette section décrit le format des fichiers utilisés et créés par BART.

Format de fichier manifeste BART

Chaque entrée de fichier manifeste constitue une seule ligne, selon le type de fichier. Chaque entrée commence par *fname*, qui est le nom du fichier. Pour éviter les problèmes d'analyse provoqués par des caractères spéciaux dans les noms de fichiers, ces noms de fichier sont codés. Pour plus d'informations, reportez-vous à la section “Format de fichier de règles BART” à la page 42.

Les champs ci-dessous représentent les attributs de fichier suivants :

type
: Type de fichier avec les valeurs possibles suivantes :
- B pour un noeud de périphérique en mode bloc
- C pour un noeud de périphérique en mode caractère
- D pour un répertoire
- F pour un fichier
- L pour un lien symbolique
- P pour un tuyau
- S pour un socket

size
: Taille de fichier en octets.

mode
: Nombre octal représentant les autorisations du fichier.

acl
: Attributs ACL du fichier. Pour un fichier avec des attributs ACL, contient la sortie de `acltotext()`.

uid
: ID utilisateur numérique du propriétaire de cette entrée.

gid
: L'ID de groupe numérique du propriétaire de cette entrée.

dirmtime
: Heure de la dernière modification, en secondes, depuis 00:00:00 UTC, le 1er janvier 1970, pour les répertoires.

lnmtime
: Heure de la dernière modification, en secondes, depuis 00:00:00 UTC, le 1er janvier 1970, pour les liens.

mtime	Heure de la dernière modification, en secondes, depuis 00:00:00 UTC, le 1er janvier 1970, pour les fichiers.
contents	Valeur de somme de contrôle du fichier. Cet attribut n'est défini que pour des fichiers standard. Si vous désactivez la vérification du contexte ou si les sommes de contrôle ne peuvent pas être calculées, la valeur de ce champ est `-`.
dest	Destination d'un lien symbolique.
devnode	Valeur du noeud de périphérique. Cet attribut est réservé aux fichiers de périphérique en mode caractère et aux fichiers de périphériques en mode bloc.

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

Format de fichier de règles BART

Les fichiers de règles sont des fichiers texte constitués de lignes qui spécifient les fichiers à inclure dans le manifeste et les attributs de fichier à inclure dans le manifeste ou le rapport. Les lignes commençant par `#`, les lignes vides et les lignes qui contiennent des espaces ne sont pas prises en compte par l'outil.

Les fichiers d'entrée ont trois types de directives :

- Directive de sous-arborescence, avec modificateurs de concordance avec un modèle en option
- Directive `CHECK`
- Directive `IGNORE`

EXEMPLE 2-5 Format de fichier de règles

```
<Global CHECK/IGNORE Directives>
<subtree1> [pattern1..]
<IGNORE/CHECK Directives for subtree1>

<subtree2> [pattern2..]
<subtree3> [pattern3..]
<subtree4> [pattern4..]
<IGNORE/CHECK Directives for subtree2, subtree3, subtree4>
```

Remarque - Toutes les directives sont lues dans l'ordre. Des directives ultérieures peuvent remplacer des directives antérieures.

La directive de sous-arborescence *doit* commencer par un chemin d'accès absolu, suivi d'un zéro ou de plusieurs instructions de concordance avec un modèle.

Attributs du fichier de règles BART

Les instructions `CHECK` et `IGNORE` définissent les attributs de fichier à suivre ou ignorer, respectivement. Les métadonnées qui commencent chaque manifeste répertorient les *mots-clés* d'attribut par type de fichier. Reportez-vous à l'Exemple 2-1, “Explication du format de manifeste BART”.

Le mot-clé `all` indique tous les attributs de fichier.

Syntaxe de la BART Quoting

Le langage de spécification du fichier de règles utilisé par BART est la syntaxe de citation UNIX standard pour la représentation des noms de fichier non standard. Les caractères spéciaux, tabulations, nouvelle ligne, espace intégrés sont codés dans leurs formes octales pour activer l'outil permettant de lire des noms de fichier. Cette syntaxe de citation non uniforme empêche certains noms de fichiers, tels que ceux contenant un retour chariot intégré, d'être traités correctement dans un pipeline de commandes. Le langage de spécification des règles autorise l'expression de critères complexes de filtrage de noms de fichier, qui seraient difficiles à décrire à l'aide uniquement de la syntaxe shell.

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

Génération de rapports BART

En mode par défaut, un rapport BART vérifie tous les fichiers installés sur le système, à l'exception des horodatages de répertoire modifiés (`dirmtime`) :

```
CHECK all
IGNORE dirmtime
```

Si vous fournissez un fichier de règles, les directives globales `CHECK all` et `IGNORE dirmtime`, dans cet ordre, sont automatiquement ajoutées au fichier de règles.

Sortie BART

Les valeurs de sortie renvoyées sont les suivantes :

0	Succès
1	Erreur non fatale lors du traitement de fichiers, telle que des problèmes d'autorisation
>1	Erreur fatale, telle qu'une option de ligne de commande non valide

Le mécanisme de génération de rapports fournit deux types de sortie : détaillée et programmatique :

- La sortie détaillée est la sortie par défaut et est localisée et présentée sur plusieurs lignes. La sortie détaillée est internationalisée et lisible par l'homme. Lorsque la commande `bart compare` compare deux manifestes de système, une liste des différences de fichiers est générée.

  La structure de la sortie se présente comme suit :

  filename attribute control:control-val test:test-val

filename	Nom du fichier qui diffère entre le manifeste de contrôle et celui de test.
attribut	Nom de l'attribut de fichier qui diffère entre les manifestes comparés. *control-val* précède *test-val*. Lorsque des écarts de plusieurs attributs se produisent dans le même fichier, chaque différence est indiquée sur une ligne distincte.

  L'exemple suivant illustre des différences d'attribut pour le fichier `/etc/passwd`. La sortie indique que les attributs `size`, `mtime` et `contents` ont été modifiés.

  ```
  /etc/passwd:
  size control:74 test:81
  mtime control:3c165879 test:3c165979
  contents control:daca28ae0de97afd7a6b91fde8d57afa
  test:84b2b32c4165887355317207b48a6ec7
  ```

- La sortie programmatique est générée en ajoutant l'option `-p` à la commande `bart compare`. Cette sortie convient à la manipulation programmatique.

  La structure de la sortie se présente comme suit :

  filename attribute control-val test-val [*attribute control-val test-val*]*

filename	Identique à l'attribut *filename* dans le format par défaut
attribute control-val test-val	Description des attributs de fichier qui diffèrent entre les manifestes de contrôle et de test pour chaque fichier.

Pour obtenir une liste des attributs pris en charge par la commande `bart`, reportez-vous à la section “Attributs du fichier de règles BART” à la page 43.

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

Glossaire de la sécurité

AES Standard de chiffrement avancé (Advanced Encryption Standard). Technique de chiffrement de données symétrique par blocs de 128 bits. Le gouvernement des Etats-Unis a adopté la variante Rijndael de l'algorithme comme norme de chiffrement en octobre 2000. AES remplace le chiffrement principal d'utilisateur comme norme administrative.

algorithme Algorithme cryptographique. Il s'agit d'une procédure de calcul récursive établie qui chiffre ou hache une entrée.

algorithme cryptographique Voir algorithme.

allocation de périphériques Protection des périphériques au niveau de l'utilisateur. L'allocation de périphériques met en oeuvre l'utilisation exclusive d'un périphérique par un utilisateur à la fois. Les données des périphériques sont purgées avant toute réutilisation d'un périphérique. Des autorisations peuvent être utilisées pour limiter les utilisateurs autorisés à allouer un périphérique.

amorce Valeur numérique de départ pour la génération de nombres aléatoires. Lorsque cette valeur provient d'une source aléatoire, l'amorce est appelée *amorce aléatoire*.

application privilégiée Application pouvant remplacer les contrôles système. L'application vérifie les attributs de sécurité, tels que des UID spécifiques, des ID de groupe, des autorisations ou des privilèges.

attributs de sécurité Remplacements de la stratégie de sécurité qui permettent à une commande d'administration de s'exécuter correctement lorsqu'elle est exécutée par un utilisateur autre que le superutilisateur. Dans le modèle de superutilisateur, les programmes `setuid root` et `setgid` sont des attributs de sécurité. Lorsque ces attributs sont appliqués à une commande, la commande s'exécute correctement, quel que soit l'utilisateur qui l'exécute. Dans le modèle de privilège, les privilèges du noyau et les autres Droits remplacent les programmes `setuid root` en tant qu'attributs de sécurité. Le modèle de privilège est compatible avec le modèle de superutilisateur dans la mesure où le modèle de privilège reconnaît également les programmes `setuid` et `setgid` comme des attributs de sécurité.

authentificateur Des authentificateurs sont transmis par des clients lors de la demande de tickets (à un KDC) et de services (à un serveur). Ils contiennent des informations générées par l'utilisation d'une clé de session connue uniquement du client et du serveur, dont l'origine récente peut être prouvée, indiquant ainsi que la transaction est sécurisée. Lorsqu'un authentificateur est utilisé avec un ticket, il peut permettre d'authentifier un principal d'utilisateur. Un authentificateur inclut le

nom du principal de l'utilisateur, l'adresse IP de l'hôte de l'utilisateur, ainsi qu'un horodatage. A la différence d'un ticket, un authentificateur ne peut servir qu'une seule fois, généralement lorsque l'accès à un service est demandé. Un authentificateur est chiffré à l'aide de la clé de session pour ce client et ce serveur.

authentification Processus de vérification de l'identité déclarée d'un principal.

autorisation 1. Dans Kerberos, processus consistant à déterminer si un principal peut utiliser un service et à définir les objets auxquels il peut accéder, ainsi que le type d'accès autorisé pour chaque objet.

2. Dans la gestion des droits des utilisateurs, autorisation pouvant être attribuée à un rôle ou un utilisateur (ou intégrée dans un profil de droits) en vue de l'exécution d'une classe d'opérations autrement interdites par la stratégie de sécurité. Les "feux verts" sont mises en oeuvre au niveau de l'application utilisateur, et non dans le noyau.

Blowfish Algorithme de chiffrement par bloc symétrique de longueur de clé variable (entre 32 et 448 bits). Son créateur, Bruce Schneier, affirme que Blowfish est optimisé pour les applications pour lesquelles la clé n'a pas besoin d'être régulièrement modifiée.

cache d'informations d'identification Espace de stockage (généralement un fichier) contenant des informations d'identification reçues de KDC.

champ d'application du service de noms Champ d'application dans lequel un rôle est autorisé à fonctionner, c'est-à-dire un hôte particulier ou tous les hôtes desservis par un service de noms spécifié, tel que NIS LDAP.

chiffrement par clé privée Dans le cas du chiffrement par clé privée, l'expéditeur et le destinataire utilisent la même clé de chiffrement. Voir également chiffrement par clé publique.

chiffrement par clé publique Modèle de chiffrement où chaque utilisateur dispose de deux clés, l'une publique et l'autre privée. Dans le cas du chiffrement par clé publique, l'expéditeur chiffre le message à l'aide de la clé publique du destinataire, et ce dernier se sert d'une clé privée pour le déchiffrer. Le service Kerberos est un système à clé privée. Voir également chiffrement par clé privée.

clé 1. En règle générale, l'un des deux principaux types de clés :

- *clé symétrique* : clé de chiffrement identique à la clé de déchiffrement. Les clés symétriques sont utilisées pour chiffrer des fichiers.
- *clé asymétrique* ou *clé publique* : clé utilisée dans les algorithmes à clé publique, tels que Diffie-Hellman ou RSA. Les clés publiques contiennent une clé privée, connue uniquement d'un utilisateur, une clé publique utilisée par le serveur ou des ressources générales, et une paire de clés publique-privée combinant les deux. Une clé privée est également qualifiée de clé *secrète*. La clé publique est également qualifiée de clé *partagée* ou *commune*.

2. Entrée (nom de principal) dans un fichier keytab. Voir également fichier keytab.

3. Dans Kerberos, clé de chiffrement dont il existe trois types :

- *Clé privée* : clé de chiffrement partagée par un principal et le KDC, et distribuée en dehors des limites du système. Voir également clé privée.
- *Clé de service* : clé remplissant la même fonction que la clé privée, mais utilisée par des serveurs et des services. Voir également clé de service.
- *Clé de session* : clé de chiffrement temporaire utilisée entre deux principaux, avec une durée de vie limitée à la durée d'une seule session de connexion. Voir également clé de session.

clé de service Clé de chiffrement partagée par un principal de service et le KDC, et distribuée en dehors des limites du système. Voir également clé.

clé de session Clé générée par le service d'authentification ou le service d'octroi de ticket. Une clé de session est générée dans le but de sécuriser les transactions entre un client et un service. La durée de vie d'une clé de session est limitée à une seule session de connexion. Voir également clé.

clé privée Chaque principal d'utilisateur reçoit une clé qui n'est connue que du KDC et de l'utilisateur du principal. Pour les principaux d'utilisateur, la clé est basée sur le mot de passe de l'utilisateur. Voir également clé.

clé secrète Voir clé privée.

client Au sens strict, il s'agit d'un processus utilisant un service réseau pour le compte d'un utilisateur, par exemple, une application utilisant `rlogin`. Dans certains cas, un serveur peut être lui-même le client d'un autre serveur ou service.

Au sens large, il s'agit d'un hôte qui a) reçoit des informations d'identification Kerberos et b) utilise un service fourni par un serveur.

Dans la pratique, ce terme désigne un principal utilisant un service.

confidentialité Voir confidentialité.

confidentialité Un service de sécurité, dans lequel les données transmises sont chiffrées avant leur envoi. La confidentialité inclut également l'intégrité des données et l'authentification de l'utilisateur. Voir également authentification, intégrité, service.

conscient des privilèges Programmes, scripts et commandes qui activent et désactivent l'utilisation des privilèges dans leur code. Dans un environnement de production, les privilèges qui sont activés doivent être fournis au processus, par exemple, en demandant aux utilisateurs du programme d'utiliser un profil de droits qui ajoute les privilèges au programme. Pour une description complète des privilèges, reportez-vous à la page de manuel `privileges`(5).

consommateur Dans la fonctionnalité Structure cryptographique d'Oracle Solaris, un consommateur est un utilisateur des services cryptographiques provenant de fournisseurs. Les consommateurs peuvent être des applications, des utilisateurs finaux ou des opérations de noyau. Kerberos, IKE et IPsec sont des exemples de consommateurs. Pour consulter des exemples de fournisseurs, reportez-vous à la section fournisseur.

DES Standard de chiffrement de données (Data Encryption Standard). Méthode de chiffrement à clé symétrique développée en 1975 et standardisée par l'ANSI en 1981 comme ANSI X.3.92. Le DES utilise une clé de 56 bits.

domaine 1. Réseau logique desservi par une seule base de données Kerberos et un ensemble de centre de distribution des clés (KDC).

2. Troisième partie d'un nom de principal. Pour le nom de principal `jdoe/admin@CORP.EXAMPLE.COM`, le domaine est `CORP.EXAMPLE.COM`. Voir également nom de principal.

Droits Alternative au modèle tout ou rien des superutilisateurs. La gestion des droits des utilisateurs et la gestion des droits de processus permettent à une organisation de répartir les privilèges du superutilisateur et de les affecter à des utilisateurs ou des rôles. Les droits dans Oracle Solaris sont implémentés en tant que privilèges de noyau, d'autorisation et de capacité à exécuter un processus sous la forme d'un UID ou GID spécifique. Les droits peuvent être regroupés dans un profil de droits et un rôle.

DSA Algorithme de signature numérique (Digital Signature Algorithm). Algorithme de clé publique dont la longueur de clé varie de 512 à 4 096 bits. La norme du gouvernement américain, DSS, atteint 1 024 bits. L'algorithme DSA repose sur l'algorithme SHA1 en entrée.

écart d'horloge Ecart maximal toléré entre les horloges système internes de tous les hôtes participant au système d'authentification Kerberos. Si l'écart d'horloge est dépassé entre des hôtes participants, les demandes sont rejetées. L'écart d'horloge est spécifié dans le fichier `krb5.conf`.

ECDSA Algorithme de signature numérique de courbe elliptique. Algorithme de clé publique basé sur une courbe elliptique mathématique. La taille d'une clé ECDSA est de beaucoup plus petite que celle d'une clé publique DSA nécessaire pour générer une signature de la même longueur.

escalade des privilèges Obtention de l'accès à des ressources situées en dehors du domaine d'application des droits qui vous sont attribués, y compris des droits permettant de remplacer les valeurs par défaut. Le résultat est qu'un processus peut effectuer des opérations non autorisées.

événement d'audit asynchrone Les événements asynchrones constituent la minorité des événements système. Ces événements ne sont associés à aucun processus, de sorte qu'aucun processus ne peut être bloqué puis réactivé ultérieurement. L'initialisation système initiale et les événements d'entrée et de sortie PROM sont des exemples d'événements asynchrones.

événement d'audit non allouable Evénement d'audit dont l'initiateur ne peut pas être déterminé, tel que l'événement AUE_BOOT.

événement d'audit synchrone Majorité des événements d'audit. Ces événements sont associés à un processus dans le système. Un événement non allouable associé à un processus est un événement synchrone, tel que l'échec d'une connexion.

fichier de ticket Voir cache d'informations d'identification.

fichier keytab Fichier de table de clés contenant une ou plusieurs clés (principaux). Un hôte ou un service utilise un fichier keytab à peu près de la même façon qu'un utilisateur se sert d'un mot de passe.

fichier stash Un fichier stash contient une copie chiffrée de la clé principale pour le KDC. Cette clé principale est utilisée lorsqu'un serveur est réinitialisé pour authentifier automatiquement le KDC avant qu'il ne démarre les processus `kadmind` et `krb5kdc`. Etant donné que le fichier stash inclut la clé principale, ce fichier et toutes ses sauvegardes doivent être sécurisés. Si le chiffrement est compromis, la clé peut être utilisée pour accéder à la base de données KDC ou la modifier.

fichiers d'audit Journaux d'audit binaires. Les fichiers d'audit sont stockés séparément dans un système de fichiers d'audit.

fournisseur Dans la fonctionnalité Structure cryptographique d'Oracle Solaris, service de cryptographie fourni aux consommateurs. Les bibliothèques PKCS #11, les modules cryptographiques du noyau et les accélérateurs de matériel sont des exemples de fournisseurs. Les fournisseurs se connectent à la structure cryptographique et sont donc également appelés *plug-ins*. Pour consulter des exemples de consommateurs, voir consommateur.

fournisseur de logiciel Dans la fonctionnalité Structure cryptographique d'Oracle Solaris, module logiciel de noyau ou bibliothèque PKCS#11 fournissant des services cryptographiques. Voir également fournisseur.

fournisseur de matériel Dans la fonctionnalité Structure cryptographique d'Oracle Solaris, un pilote de périphérique et son accélérateur matériel. Les fournisseurs de matériel déchargent le système informatique d'opérations cryptographiques coûteuses, libérant ainsi les ressources de l'unité centrale pour d'autres utilisations. Voir également fournisseur.

FQDN Nom de domaine complet. Par exemple, `central.example.com` (et pas simplement `denver`).

GSS-API Interface de programmation d'application générique de service de sécurité. Couche réseau assurant la prise en charge de différents services de sécurité modulaires, y compris le service Kerberos. GSS-API fournit des services d'authentification, d'intégrité et de confidentialité. Voir également authentification, intégrité, confidentialité.

hôte Système accessible par l'intermédiaire d'un réseau.

hôte principal Instance spécifique d'un principal de service dans lequel le principal (indiqué par le nom primaire `host`) est configuré de manière à fournir une gamme de services réseau, comme `ftp`, `rcp` ou `rlogin`. `host/central.example.com@EXAMPLE.COM` est un exemple d'hôte principal. Voir également principal de serveur.

image système unique Une image système unique est utilisée dans l'audit d'Oracle Solaris afin de décrire un groupe de systèmes audités utilisant le même service de noms. Ces systèmes envoient leurs enregistrements d'audit à un serveur d'audit central où les enregistrements peuvent être comparés comme s'ils provenaient d'un même système.

informations d'identification Package d'informations comprenant un ticket et une clé de session correspondante. Informations utilisées pour authentifier l'identité d'un principal. Voir également ticket, clé de session.

instance Deuxième partie d'un nom de principal, une instance qualifie le primaire du principal. Dans le cas d'un principal de service, l'instance est requise. L'instance est le nom de domaine complet de l'hôte, comme dans `host/central.example.com`. Pour les principaux d'utilisateur, une instance est facultative. Notez, cependant, que `jdoe` et `jdoe/admin` sont des principaux uniques. Voir également primaire, nom de principal, principal de service, principal d'utilisateur.

intégrité Service de sécurité qui assure, outre l'authentification de l'utilisateur, la validité des données transmises par le biais de sommes de contrôle cryptographiques. Voir également authentification, confidentialité.

jeu autorisé Jeu des privilèges disponibles pour l'utilisation par un processus.

jeu de base Jeu de privilèges affecté à un processus d'utilisateur lors de la connexion. Sur un système non modifié, le jeu héritable initial de chaque utilisateur correspond au jeu de base défini au moment de la connexion.

jeu de privilèges Ensemble de privilèges. Chaque processus comporte quatre jeux de privilèges qui déterminent s'il peut utiliser un privilège particulier. Voir jeu limite, jeu effectif, jeu autorisé et jeu hérité.

De même, le jeu de base de privilèges est l'ensemble des privilèges affectés aux processus d'un utilisateur au moment de la connexion.

jeu effectif Jeu de privilèges actuellement en vigueur sur un processus.

jeu hérité Jeu de privilèges dont un processus peut hériter en appelant `exec`.

jeu limite Limite extérieure des privilèges disponibles pour un processus et ses enfants.

KDC Centre de distribution de clés (Key Distribution Center). Machine disposant de trois composants Kerberos V5.

- Principal et base de données de clés
- Service d'authentification
- Service d'octroi de tickets

Chaque domaine dispose d'un KDC maître et doit avoir un ou plusieurs KDC esclaves.

KDC esclave Copie d'un KDC maître capable de réaliser la plupart des fonctions du maître. Chaque domaine dispose généralement de plusieurs KDC esclaves et d'un seul KDC maître. Voir également KDC, KDC maître.

KDC maître KDC maître dans chaque domaine, comprenant un serveur d'administration Kerberos, `kadmind` et un démon d'authentification et d'octroi de tickets, `krb5kdc`. Chaque domaine doit disposer d'au moins un KDC maître, et peut avoir plusieurs KDC de duplication ou esclaves fournissant des services d'authentification aux clients.

Kerberos Service d'authentification, protocole utilisé par ce service ou code servant à mettre en oeuvre ce service.

Implémentation Kerberos dans Oracle Solaris étroitement basée sur l'implémentation Kerberos V5.

Bien que techniquement différents, "Kerberos" et "Kerberos V5" sont souvent utilisés de façon interchangeable dans la documentation Kerberos.

Dans la mythologie grecque, Kerberos (en français Cerbère) était un chien féroce tricéphale qui gardait la porte des Enfers.

keystore Un keystore contient des mots de passe, des phrases de passe, des certificats et d'autres objets d'authentification pour l'extraction par des applications. Il peut être spécifique à une technologie, ou à un emplacement utilisé par plusieurs applications.

kvno Numéro de version de la clé. Numéro de série faisant le suivi d'une clé spécifique selon l'ordre dans lequel elle a été générée. Plus le numéro kvno est élevé, plus la clé est récente.

liste de contrôle d'accès (ACL) Une liste de contrôle d'accès (ACL) offre une sécurité des fichiers plus précise que la protection de fichier UNIX conventionnelle. Par exemple, une ACL vous permet d'autoriser l'accès en lecture de groupe à un fichier, tout en autorisant un seul membre de ce groupe à écrire dans le fichier.

MAC 1. Voir MAC.

2. Egalement appelé étiquetage. Dans la terminologie de sécurité gouvernementale, MAC signifie Mandatory Access Control (contrôle d'accès obligatoire). Les étiquettes telles que Top Secret et Confidential sont des exemples de MAC. MAC diffère de DAC (Discretionary Access Control, contrôle d'accès discrétionnaire). Les autorisations UNIX constituent un exemple de DAC.

3. Dans le matériel, il s'agit de l'adresse système unique sur un réseau local (LAN). Si le système est sur un réseau Ethernet, le MAC est l'adresse Ethernet.

MAC MAC garantit l'intégrité des données et authentifie leur origine. MAC ne protège aucunement contre l'écoute frauduleuse des informations échangées.

MD5 Fonction de hachage cryptographique répétitive utilisée pour authentifier les messages, y compris les signatures numériques. Elle a été développée en 1991 par Rivest. Son utilisation a été désapprouvée.

mécanisme 1. Package logiciel spécifiant des techniques cryptographiques pour assurer l'authentification ou la confidentialité des données. Exemples : Kerberos V5, clé publique Diffie-Hellman

2. Dans la fonctionnalité Structure cryptographique d'Oracle Solaris, mise en oeuvre d'un algorithme destiné à un usage particulier. Par exemple, un mécanisme DES appliqué à l'authentification, tel que CKM_DES_MAC, est un mécanisme distinct d'un mécanisme DES appliqué au chiffrement, CKM_DES_CBC_PAD.

mécanisme de sécurité Voir mécanisme.

minimisation Installation du système d'exploitation minimal nécessaire pour l'exécution du serveur. Tout logiciel n'étant pas directement lié au fonctionnement du serveur n'est pas installé, ou est supprimé après l'installation.

modèle de privilège Modèle de sécurité plus strict que le modèle superutilisateur sur un système informatique. Dans le modèle de privilège, les processus nécessitent des privilèges pour s'exécuter. L'administration du système peut être divisée en quatre parties discrètes basées sur les privilèges dont les administrateurs disposent dans leurs processus. Des privilèges peuvent être affectés au processus de connexion d'un administrateur, ou des privilèges peuvent être affectés de manière à ne s'appliquer qu'à certaines commandes.

modèle de superutilisateur Modèle de sécurité UNIX standard sur un système informatique. Dans le modèle de superutilisateur, un administrateur dispose d'un contrôle de type tout ou rien sur le système. En règle générale, pour l'administration de la machine, un utilisateur se connecte en tant que superutilisateur (`root`) et peut effectuer toutes les activités d'administration.

moindre privilège Modèle de sécurité qui octroie à un processus spécifié uniquement un sous-ensemble de pouvoirs de superutilisateur. Le modèle de moindre privilège octroie aux utilisateurs standard les privilèges suffisants pour qu'ils puissent effectuer des tâches d'administration personnelles, telles que le montage de systèmes de fichiers et la modification de l'appartenance des fichiers. Par ailleurs, les processus s'exécutent uniquement avec les privilèges dont ils ont besoin pour terminer la tâche, et non avec tous les pouvoirs du superutilisateur, c'est-à-dire, tous les privilèges. Les dommages causés par les erreurs de programmation, comme les débordements de la mémoire tampon, peuvent être limités à un utilisateur non root, qui n'a pas accès à des fonctions essentielles comme la lecture ou l'écriture de fichiers système protégés ou l'arrêt de la machine.

moteur d'analyse Application tierce, résidant sur un hôte externe, qui examine un fichier à la recherche de virus connus.

nom de principal 1. Le nom d'un principal, au format *primary/instance@REALM*. Voir également instance, primaire, domaine.

2. (RPCSEC_GSS API) Voir principal de client, principal de serveur.

NTP Network Time Protocol. Logiciel de l'Université de l'Etat du Delaware qui vous permet de gérer avec précision la synchronisation de l'heure ou de l'horloge réseau, ou les deux, dans un environnement réseau. Vous pouvez utiliser le protocole NTP pour préserver l'écart d'horloge dans un environnement Kerberos. Voir également écart d'horloge.

objet public Fichier appartenant à l'utilisateur `root` et lisible par tout le monde, tel que n'importe quel fichier du répertoire `/etc`.

PAM Module d'authentification enfichable (Pluggable Authentication Module). Structure permettant d'utiliser plusieurs mécanismes d'authentification sans recompilation des services recourant à ces mécanismes. PAM permet l'initialisation de la session SEAM au moment de son ouverture

phrase de passe Phrase utilisée pour vérifier qu'une clé privée a été créée par l'utilisateur de la phrase de passe. Une bonne phrase de passe contient 10 à 30 caractères alphanumériques et évite les noms et proses simples. Vous êtes invité à saisir la phrase de passe pour authentifier l'utilisation de la clé privée pour chiffrer et déchiffrer les communications.

piste d'audit Collection de tous les fichiers d'audit provenant de tous les hôtes.

primaire Première partie du nom d'un nom de principal. Voir également instance, nom de principal, domaine.

principal 1. Client/utilisateur dont le nom est unique ou instance de serveur/service participant à une communication en réseau. Les transactions Kerberos impliquent des interactions entre des principaux (principaux de service et d'utilisateur) ou entre des principaux et des KDC. En d'autres termes, un principal est une entité unique à laquelle Kerberos peut attribuer des tickets. Voir également nom de principal, principal de service, principal d'utilisateur.

2. (RPCSEC_GSS API) Voir principal de client, principal de serveur.

principal admin Principal d'utilisateur dont le nom est au format *nom_utilisateur*`/admin` (comme dans `jdoe/admin`). Un principal admin peut disposer de davantage de privilèges (de modification de stratégies par exemple) qu'un principal d'utilisateur standard. Voir également nom de principal et principal d'utilisateur.

principal d'utilisateur Principal attribué à un utilisateur particulier. Le nom primaire d'un principal d'utilisateur est un nom d'utilisateur et son instance facultative est un nom utilisé pour décrire l'utilisation prévue des informations d'identification correspondantes (`jdoe` ou `jdoe/admin` par exemple). Egalement appelé instance d'utilisateur. Voir également principal de service.

principal de client (RPCSEC_GSS API) Client (utilisateur ou application) utilisant des services réseau sécurisés RPCSEC_GSS. Les noms de principaux client sont stockés sous forme de structures `rpc_gss_principal_t`.

principal de serveur (RPCSEC_GSS API) Principal fournissant un service. Le principal de serveur est stocké sous forme d'une chaîne de caractères ASCII dont le format est *service@hôte*. Voir également principal de client.

principal de service Principal assurant l'authentification Kerberos pour un ou plusieurs services. Pour les principaux de service, le nom primaire est un nom de service, tel que `ftp`, et son instance est le nom d'hôte complet du système fournissant le service. Voir également hôte principal, principal d'utilisateur.

principe du moindre privilège Voir moindre privilège.

privilège 1. En règle générale, le pouvoir ou la possibilité d'effectuer une opération sur un système informatique dépassant les pouvoirs d'un utilisateur standard. Les privilèges de superutilisateur sont l'ensemble des Droits qui sont octroyés au superutilisateur. Un utilisateur privilégié ou

une application privilégiée est un utilisateur ou une application auquel/à laquelle des droits supplémentaires ont été accordés.

2. Droit discret dans le cadre d'un processus dans un système Oracle Solaris. Les privilèges offrent un contrôle des processus plus détaillé que `root`. Les privilèges sont définis et appliqués dans le noyau. Les privilèges sont également appelés *privilèges de processus* ou *privilèges de noyau*. Pour une description complète des privilèges, reportez-vous à la page de manuel `privileges`(5).

profil de droits On parle aussi de profil. Ensemble de remplacements de sécurité pouvant être affecté à un rôle ou à un utilisateur. Un profil de droits peut se composer d'autorisations, de privilèges, de commandes avec des attributs de sécurité et d'autres profils de droits appelés des profils supplémentaires.

Profil de droits authentifiés profil de droits qui nécessite que l'utilisateur ou le rôle qui lui est affecté saisisse un mot de passe avant d'exécuter une opération à partir du profil. Ce comportement est similaire au comportement de `sudo`. La durée de temps pendant laquelle le mot de passe est valide est configurable.

protocole Diffie-Hellman Egalement appelé cryptographie par clé publique. Protocole d'accord de clés cryptographiques asymétriques mis au point par Diffie et Hellman en 1976. Ce protocole permet à deux utilisateurs d'échanger une clé secrète via un moyen non sécurisé sans secrets préalables. Diffie-Hellman est utilisé par Kerberos.

QOP Qualité de la protection. Paramètre servant à sélectionner les algorithmes cryptographiques utilisés en association avec le service d'intégrité ou de confidentialité.

RBAC Contrôle d'accès basé sur les rôles, fonction de gestion des droits des utilisateurs d'Oracle Solaris. Voir Droits

Réauthentification Obligation de fournir un mot de passe pour effectuer une opération sur un ordinateur. En règle générale, les opérations `sudo` nécessitent une réauthentification. Les profils de droits peuvent contenir des commandes qui nécessitent la réauthentification. Voir Profil de droits authentifiés.

relation Variable ou relation de configuration définie dans les fichiers `kdc.conf` ou `krb5.conf`.

rôle Identité spéciale destinée à l'exécution d'applications privilégiées et ne pouvant être prise que par des utilisateurs assignés.

RSA Méthode permettant d'obtenir des signatures numériques et des systèmes de cryptographie par clé publique. Cette méthode datant de 1978 a été décrite par trois développeurs (Rivest, Shamir et Adleman).

SEAM Nom de produit pour la version initiale de Kerberos sur les systèmes Solaris. Ce produit est basé sur la technologie Kerberos V5 développée au Massachusetts Institute of Technology. SEAM est maintenant appelé le service Kerberos. Il continue d'être légèrement différent de la version MIT.

Secure Shell Un protocole particulier pour une connexion à distance sécurisée et d'autres services de réseau sécurisé via un réseau non sécurisé.

Sécurisation Modification de la configuration par défaut du système d'exploitation pour supprimer les failles de sécurité inhérentes à l'hôte.

séparation des tâches Composante de la notion de moindre privilège. La séparation des tâches permet d'empêcher un utilisateur d'exécuter ou d'approuver les opérations terminant une transaction. Par exemple, dans RBAC, vous pouvez séparer la création d'un utilisateur de connexion de l'affectation des remplacements de sécurité. Un rôle crée l'utilisateur. Un autre rôle peut affecter les attributs de sécurité, tels que des profils de droits, des rôles et des privilèges aux utilisateurs existants.

serveur Principal fournissant une ressource aux clients réseau. Si vous vous connectez par `ssh` au système `central.example.com` par exemple, ce système est le serveur fournissant le service `ssh`. Voir également principal de service.

serveur d'application Voir serveur d'application réseau.

serveur d'application réseau Serveur fournissant une application réseau, telle que `ftp`. Un domaine peut contenir plusieurs serveurs d'application réseau.

service 1. Ressource fournie aux clients du réseau, souvent par plusieurs serveurs. Si vous vous connectez par `rlogin` à la machine `central.example.com` par exemple, cette machine est le serveur fournissant le service `rlogin`.

2. Service de sécurité (d'intégrité ou de confidentialité) fournissant un niveau de protection supérieur à l'authentification. Voir également intégrité et confidentialité.

service de sécurité Voir service.

SHA1 Algorithme de hachage sécurisé (Secure Hashing Algorithm). L'algorithme s'applique à toute longueur d'entrée inférieure à 2^{64} afin d'obtenir une synthèse des messages. L'algorithme SHA1 sert d'entrée à DSA.

shell de profil Dans la gestion des droits, shell permettant à un rôle (ou un utilisateur) d'exécuter, à partir de la ligne de commande, toutes les applications privilégiées affectées au profil de droits du rôle. Les versions du shell de profil correspondent aux shells disponibles sur le système, comme la version `pfbash` de `bash`

stratégie En règle générale, plan ou ensemble d'actions qui influence ou détermine les décisions et actions. Pour les systèmes informatiques, la stratégie fait généralement référence à la stratégie de sécurité. La stratégie de sécurité de votre site constitue un ensemble de règles qui définissent la sensibilité des informations traitées et les mesures prises pour protéger les informations contre tout accès non autorisé. Par exemple, la stratégie de sécurité peut exiger que les systèmes soient audités, que les périphériques soient protégés par des privilèges et que les mots de passe soient modifiés toutes les six semaines.

Pour l'implémentation des stratégies dans des zones spécifiques du SE Oracle Solaris, voir stratégie d'audit, stratégie dans la structure cryptographique, stratégie de périphériques, stratégie Kerberos, stratégie de mot de passe et stratégie de droits.

stratégie d'audit Paramètres globaux et par utilisateur qui déterminent les événements d'audit enregistrés. Les paramètres globaux s'appliquant au service d'audit déterminent généralement les informations facultatives à inclure dans la piste d'audit. Deux paramètres, `cnt` et `ahlt`, affectent le fonctionnement du système lorsque la file d'attente de l'audit est pleine. Par exemple, la stratégie d'audit peut exiger qu'un numéro de séquence fasse partie de chaque enregistrement d'audit.

stratégie dans la structure cryptographique Dans la fonctionnalité Structure cryptographique d'Oracle Solaris, la stratégie correspond à la désactivation de mécanismes de chiffrement existants. Les mécanismes ne peuvent ensuite plus être utilisés. La stratégie de la structure cryptographique peut empêcher l'utilisation d'un mécanisme particulier tel que `CKM_DES_CBC` provenant d'un fournisseur tel que DES.

stratégie de droits Stratégie de sécurité associée à une commande. Actuellement, `solaris` est la stratégie valide pour Oracle Solaris. La stratégie `solaris` reconnaît les privilèges et la stratégie de privilège étendue, les autorisations et les attributs de sécurité de `setuid`.

stratégie de mot de passe Algorithmes de chiffrement pouvant être utilisés pour générer des mots de passe. Peut également faire référence à des questions plus générales concernant les mots de passe, telles que la fréquence à laquelle le mot de passe doit être modifié, le nombre de tentatives autorisées de saisie d'un mot de passe et d'autres considérations relatives à la sécurité. La stratégie de sécurité requiert des mots de passe. La stratégie de mot de passe peut requérir des mots de passe chiffrés avec l'algorithme AES, et imposer d'autres exigences relatives à la force des mots de passe.

stratégie de périphériques Protection d'un périphérique au niveau du noyau. La stratégie de périphériques repose sur deux jeux de privilèges pour un périphérique. Un jeu de privilèges contrôle l'accès en lecture au périphérique. Le deuxième jeu de privilèges contrôle l'accès en écriture sur le périphérique. Voir également stratégie.

stratégie de sécurité Voir stratégie.

stratégie Kerberos Ensemble de règles régissant l'utilisation des mots de passe dans le service Kerberos. Les stratégies peuvent réguler les accès des principaux ou des paramètres de ticket, tels que la durée de vie.

stratégie pour technologies à clé publique Dans la structure de gestion des clés (KMF), la stratégie correspond à la gestion de l'utilisation des certificats. La base de données de stratégies KMF peut définir des contraintes s'appliquant à l'utilisation des clés et des certificats gérés par la bibliothèque KMF.

stratégie RBAC Voir stratégie de droits.

stratégies de réseau Paramètres configurés par les utilitaires réseau pour protéger le trafic réseau. Pour plus d'informations sur la sécurité réseau, reportez-vous à la section “ Sécurisation du réseau dans Oracle Solaris 11.2 ”.

synthèse Voir synthèse de message.

synthèse de message Valeur de hachage calculée à partir d'un message. La valeur de hachage identifie le message de manière presque unique. Une synthèse permet de vérifier l'intégrité d'un fichier.

TGS Service d'octroi de tickets (Ticket-Granting Service) Partie du KDC responsable de l'émission des tickets.

TGT Ticket d'octroi de tickets (Ticket-Granting Ticket) Ticket émis par le KDC, permettant au client de demander des tickets pour d'autres services.

ticket Paquet d'informations servant à transmettre en toute sécurité l'identité d'un utilisateur à un serveur ou un service. Un ticket n'est valable que pour un client et un service particulier sur un serveur spécifique. Il contient le nom de principal du service, le nom de principal de l'utilisateur, l'adresse IP de l'hôte de l'utilisateur, un horodatage et une valeur définissant la durée de vie du ticket. La création d'un ticket s'effectue à l'aide d'une clé de session aléatoire utilisée par le client et le service. Une fois le ticket créé, il peut être réutilisé jusqu'à son expiration. Un ticket sert uniquement à authentifier un client lorsqu'il est présenté avec un nouvel authentificateur. Voir également authentificateur, informations d'identification, service, clé de session.

ticket initial Ticket émis de manière directe, et pas à partir d'un ticket d'octroi de tickets. Certains services, tels que les applications modifiant les mots de passe, peuvent nécessiter des tickets marqués comme étant `initiaux` afin d'assurer que le client peut démontrer qu'il connait sa clé secrète. Cette garantie est importante car un ticket initial indique que le client s'est récemment authentifié et ne dépend pas d'un ticket d'octroi de tickets qui peut exister depuis un certain temps.

ticket non valide Ticket postdaté n'étant pas encore utilisable. Un ticket non valide est rejeté par un serveur d'application jusqu'à ce qu'il soit validé. Pour être validé, un ticket non valide doit être présenté au KDC par le client dans une demande de TGS, avec l'indicateur `VALIDATE`, après l'heure de début. Voir également ticket postdaté.

ticket postdaté Un ticket postdaté ne devient valide qu'un certain temps après sa création. Par exemple, un tel ticket peut être utile avec les tâches exécutées par lots la nuit car le ticket, s'il est volé, ne peut pas être utilisé tant que l'exécution de ces tâches n'a pas eu lieu. Lorsqu'un ticket postdaté est émis, il est émis en tant que `non valide` et le reste jusqu'à ce que a) son heure de début soit dépassée et b) le client demande la validation par le KDC. Un ticket postdaté est normalement valide jusqu'à l'heure d'expiration du ticket d'octroi de tickets. Toutefois, si le ticket postdaté est marqué comme `renouvelable`, sa durée de vie est normalement égale à la durée de vie entière du ticket d'octroi de tickets. Voir également ticket non valide, ticket renouvelable.

ticket renouvelable Etant donné que tout ticket dont la durée de vie est très longue peut présenter un risque pour la sécurité, les tickets peuvent être conçus pour être `renouvelables`. Un ticket renouvelable

possède deux moments d'expiration : a) l'heure à laquelle l'instance courante du ticket expire et b) la durée de vie maximale de tout ticket. Si un client souhaite continuer à utiliser un ticket, il peut le renouveler avant sa première expiration. Par exemple, un ticket peut être valide pendant une heure, et tous les tickets ont une durée de vie maximale de dix heures. Si le client détenant le ticket souhaite le conserver plus d'une heure, il doit le renouveler. Lorsqu'un ticket atteint sa durée de vie maximale, celui-ci expire automatiquement et ne peut pas être renouvelé.

ticket transmissible Ticket pouvant être utilisé par un client pour demander un ticket sur un hôte distant sans devoir se soumettre au processus complet d'authentification sur l'hôte concerné. Par exemple, si l'utilisateur `david` obtient un ticket transmissible sur la machine de l'utilisatrice `jennifer`, il peut se connecter à sa propre machine sans devoir demander un nouveau ticket (et donc s'authentifier à nouveau). Voir également ticket utilisable avec proxy.

ticket utilisable avec proxy Ticket pouvant être utilisé par un service pour le compte d'un client afin d'effectuer une opération pour ce dernier. On dit alors que le service agit en tant que proxy du client. Grâce à ce ticket, le service peut prendre l'identité du client. Le service peut utiliser un tel ticket afin d'obtenir un ticket de service d'un autre service, mais non un ticket d'octroi de tickets. La différence entre un ticket utilisable avec proxy et un ticket transmissible réside dans le fait que le premier n'est valide que pour une seule opération. Voir également ticket transmissible.

Utilisateur privilégié Utilisateur auquel des droits dépassant ceux d'un utilisateur standard sont attribués sur un système informatique. Voir également Utilisateurs de confiance.

Utilisateurs de confiance Utilisateurs que vous avez décidé d'autoriser à effectuer des tâches d'administration avec un certain degré de confiance. En règle générale, les administrateurs créent tout d'abord des connexions pour les utilisateurs de confiance et affectent des droits d'administration correspondant au niveau de confiance et aux compétences des utilisateurs. Ces utilisateurs aider à configurer et à tenir à jour le système. Egalement appelés *utilisateurs privilégiés*.

variante Historiquement, la *variante de sécurité* et la *variante d'authentification* désignaient la même chose, lorsqu'une variante indiquait un type d'authentification (AUTH_UNIX, AUTH_DES, AUTH_KERB). RPCSEC_GSS est également une variante de sécurité, même s'il permet d'assurer des services d'intégrité et de confidentialité, en plus de l'authentification.

variante de sécurité Voir variante.

VPN Réseau privé virtuel assurant une communication sécurisée en utilisant les mécanismes de chiffrement et de mise en tunnel pour connecter les utilisateurs via un réseau public.

Index

B

C

D

F

N

O

P

R

S

U

V