Debian GNU/Linux

Vers une administration de haute sécurité

informatique technique

eni Editions

Collection EPSILON

Philippe PIERRE

Les éléments à télécharger sont disponibles à l'adresse suivante :
http://www.editions-eni.fr
Saisissez la référence de l'ouvrage **EPDEBS** dans la zone de recherche
et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Chapitre 2
Installation

Partie 2 : Le système d'exploitation

Chapitre 3

Sécurisation du système d'exploitation

Chapitre 4

Sécurisation de l'environnement

Chapitre 5
Sécurisation du système

Chapitre 7

Sécurité et services d'infrastructure

Partie 4 : Les applications

Chapitre 8

Sécurisation des bases de données

Chapitre 9
Sauvegardes

Chapitre 10
Sécurité des applications

Partie 5 : La surveillance
Chapitre 11
Supervision

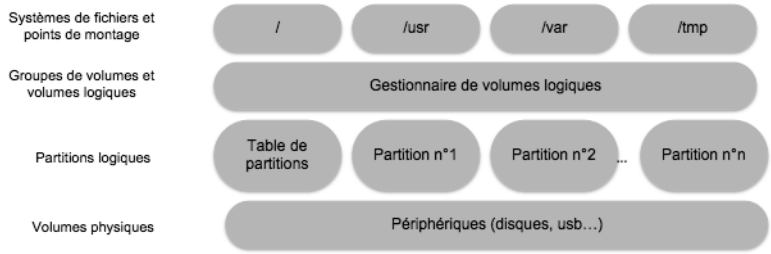
Conclusion

Chapitre 4
Sécurisation de l'environnement

1. Systèmes de fichiers

1.1 Présentation et types

Il a été vu dans les chapitres précédents comment les périphériques étaient partitionnés afin de présenter les données stockées sur les volumes logiques. Ces derniers doivent être présentés à leur point de montage pour pouvoir accéder aux informations s'y trouvant.



Donc, les données sont accessibles aux utilisateurs ainsi qu'aux programmes via une architecture structurée, selon un modèle arborescent de répertoires et de fichiers. On appelle système de fichiers le format de présentation de ces données sous forme de succession de blocs organisés.

Les formats les plus courants sur des systèmes GNU/Linux sont ext2, ext3 et ext4. Mais, il faut aussi souligner que les lecteurs CD/DVD ont leur propre système de fichiers :

- Système de fichiers ISO9660 (pour les lecteurs CD et les formats Joliet et Rock Ridge).
- Système de fichiers *Universal Disk Format* ou UDF (pour les lecteurs DVD).

Voici quelques systèmes de fichiers bien connus pour être utilisés sous Linux Debian :

Nom FS	Commentaire
ext2	Système de fichiers initial (pas de journalisation).
ext3	Système de fichiers évolués (avec journalisation).
ext4	Système de fichiers journalisé étendu à 1024 Po.
btrfs	Amélioration d'ext4 16 Eo basé sur les références arrière (back reference) pour la protection des données et la compression.

Lorsque l'on a créé les volumes logiques à l'aide de la commande `lvcreate`, il devient alors possible de créer un ou plusieurs systèmes de fichiers. Pour cela, il suffit d'utiliser la commande `mkfs`. En réalité, cette commande est une interface permettant de formater le système de fichiers selon le type souhaité : ext2, ext3, btrfs ou autres.

Option	Format long	Commentaire
`-V`	`--version`	Permet de lister toutes les commandes exécutées.
`-t <Type>`	`--type`	Indique le type de système de fichiers à créer.
`-v`	`--verbose`	Affiche plus d'informations (mode verbeux).
`-b`		Donne la taille en blocs multiples de 512 octets.
`-c`		Vérifie les blocs défectueux avant la création du système de fichiers.
`-i`		Fournit le ratio octets/inodes.
`-m`		Vérifie les blocs défectueux avant la création du système de fichiers.
`-L`		Décrit le label du système de fichiers.
`-J`		Permet de créer une journalisation au système de fichiers (pour ext2).

Lorsque l'on précise un type de système de fichiers, le programme fait alors appel à une commande dédiée au formatage :

- mkfs.ext2
- mkfs.ext3
- mkfs.ext4
- ...

Ensuite, lorsque le système de fichiers est créé, on peut générer un point de montage : généralement un répertoire suffit, pour monter le périphérique de type bloc, ou la partition, sur laquelle le système de fichiers existe. La commande `mount` permet d'attacher le répertoire racine du système de fichiers, au répertoire appelé « point de montage ».

Chaque montage réussi provoque l'écriture d'une ligne correspondante dans le fichier `/etc/mtab`.

```
root@Philo:/home/uadmin# more /etc/mtab
rootfs / rootfs rw 0 0
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
udev /dev devtmpfs rw,relatime,size=10240k,nr_inodes=30340,mode=755 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0
tmpfs /run tmpfs rw,nosuid,noexec,relatime,size=25444k,mode=755 0 0
/dev/mapper/Philo-root / ext4 rw,relatime,errors=remount-ro,user_xattr,barrier=1,data=ordered 0 0
tmpfs /run/lock tmpfs rw,nosuid,nodev,noexec,relatime,size=5120k 0 0
tmpfs /run/shm tmpfs rw,nosuid,nodev,noexec,relatime,size=50880k 0 0
/dev/sda1 /boot ext2 rw,relatime,errors=continue 0 0
/dev/mapper/Philo-home /home ext4 rw,relatime,user_xattr,barrier=1,data=ordered 0 0
/dev/mapper/Philo-tmp /tmp ext4 rw,relatime,user_xattr,barrier=1,data=ordered 0 0
/dev/mapper/Philo-usr /usr ext4 rw,relatime,user_xattr,barrier=1,data=ordered 0 0
/dev/mapper/Philo-var /var ext4 rw,relatime,user_xattr,barrier=1,data=ordered 0 0
rpc_pipefs /var/lib/nfs/rpc_pipefs rpc_pipefs rw,relatime 0 0
binfmt_misc /proc/sys/fs/binfmt_misc binfmt_misc rw,nosuid,nodev,noexec,relatime 0 0
```

Les informations concernant les différents montages peuvent être consultées dans le fichier `/proc/mounts`. Il s'agit d'une copie virtuelle du fichier `mtab`.

L'ordre de montage répond à une syntaxe bien particulière permettant de monter aussi bien des systèmes de fichiers à partir de périphérique, de label ou encore d'identifiant :

```
# mount -t ext4 /dev/sdb1 /mnt/data
# mount -t ext4 -L DATA /data2
# mount -t ext3 -U 66f6ce4b8-635c-4103-9a81-977fb7bb29fe /home
```

L'ensemble des ordres de montages peuvent être effectués automatiquement dès lors que leurs paramètres sont indiqués dans le fichier `/etc/fstab`.

Ce dernier contient les noms des périphériques et leur point de montage et les options du montage à réaliser. Le format de ce fichier est le suivant :

Champ	Commentaire
`Périphérique`	Périphérique à monter. Cela peut être sous forme de nom, de label ou d'UUID.
`Point de montage`	Répertoire d'accès au système de fichiers monté
`Type de système`	Type du système de fichiers : ext2, ext3, ext4...
`options`	Options de montage, séparées par des virgules.
`dump`	Fréquence de dump pour les outils de sauvegarde système.
`fsck`	Fréquence de vérification fsck du système : 0 = à ignorer, 1 = en premier, 2 = en second... Les systèmes de même valeur sont parallélisés.

Pour monter l'ensemble des partitions proposées dans le fichier `/etc/fstab`, on peut exécuter la commande suivante :

```
# mount -a
```

A contrario, il est possible de monter un volume unitaire en le précisant derrière la commande `mount` :

```
# mount /data
Pour réaliser un démontage, on peut alors invoquer la commande
umount de la façon suivante :
# umount /mnt/perso
```

Remarque

Toutefois, il n'existe pas de commande permettant le démontage de l'intégralité des partitions.

Il existe de nombreuses options concernant les montages et leurs propriétés et permettant de sécuriser les accès :

Option	Commentaire
`defauts`	Reprend l'ensemble des options : `rw`, `suid`, `dev`, `exec`, `auto`, `noauto` et `async`.
`sync/async`	Active ou désactive les écritures synchrones. Ceci est préférable avec des supports externes.
`exec/noexec`	Exécute (ou non) les fichiers binaires sur le support.
`noatime`	Empêche la mise à jour de l'horodatage lors de chaque accès aux fichiers. Cette option est utile pour des supports externes, des pages web ou des forums de news.
`auto/noauto`	Monte automatiquement (explicitement depuis fstab) le système de fichiers.
`user/nouser`	Autorise (ou non) les utilisateurs standards à effectuer les montages.
`remount`	Autorise la remontée du système afin de prendre en compte de nouvelles options.
`ro/rw`	Réalise le montage en lecture seule ou en lecture-écriture.
`dev/nodev`	Permet d'interpréter (ou non) les fichiers spéciaux.
`noload`	Évite de charger le journal (valable pour les systèmes de fichiers journalisés).
`usrquota/ grpquota`	Permet d'être ignorée par l'option du système de fichiers et n'est prise en compte que par le sous-système de gestion de quotas.
`acl`	Autorise l'utilisation de contrôle d'accès (ou *Access Control Lists*).
`user_xattr`	Accepte les attributs étendus sur les fichiers : encodage de texte, champs d'indexation... (valable pour les systèmes de fichiers ext3, ext4...).
`umask`	Permet pour les systèmes de fichiers NAT/NTFS d'appliquer un masque différent de la valeur par défaut.

Lorsque l'on effectue le montage d'un système de fichiers utilisant l'option `sync`, cela permet de s'affranchir de toute forme de cache en écriture attachée au disque. Cela permet alors de fiabiliser les opérations d'écritures. En fait, cette option permet de vider ponctuellement le cache sur un système de fichiers ne bénéficiant pas de cette option de montage.

Lorsqu'un système de fichiers est en cours, il est possible d'avoir des informations concernant le processus actif sur tel ou tel point de montage. Il suffit juste d'utiliser la commande `lsof`, permettant de lister les sockets ou les processus attachés à un fichier ou un point montage, en l'occurrence :

```
root@Philo:/home/uadmin# lsof /tmp
COMMAND    PID   USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
gnome-ter 3259 uadmin   17w   REG  254,5     9437   14 /tmp/vteI1ARJX (deleted)
gnome-ter 3259 uadmin   18u   REG  254,5     1504   18 /tmp/vte8LARJX (deleted)
```

Lorsqu'un point de montage semble occupé et qu'il n'est alors pas possible de libérer la ressource, on peut interroger le point de montage à l'aide de la commande `fuser` pour forcer la libération du point de montage et démonter celui-ci :

```
# fuser -km /mnt/data
```

On peut également modifier certains paramètres du système de fichiers en utilisant la commande `tune2fs` :

```
# tune2fs -j /dev/sdb1  (pour journaliser le système de fichiers)
# tune2fs -O^has_journal /dev/sdb1 (pour supprimer la
journalisation)
```

ou encore `e2label`, permettant de générer un label à un système de fichiers existant :

```
# e2label /dev/sdb1 LABEL
```

Remarque

La commande seule avec le nom du périphérique permet de lister le label du système pour ce même périphérique :

```
# e2label /dev/sdb1
```

ATTENTION : ces commandes sont propres au type du système de fichiers manipulé. En effet, pour un système reiserfs, la commande sera plutôt `reiserfstune`.

La commande `tune2fs` est très complète et permet de réaliser de nombreuses tâches. Voici les options principales :

Option	Commentaire
`-c <N>`	Détermine le nombre de montages avant lequel la vérification `fsck` sera effectuée. La valeur 0 permet de désactiver cette vérification automatique.
`-i <N>`	Représente l'intervalle entre deux vérifications fsck : d(ay), w(eek), m(onth). Par défaut l'unité est le jour.
`-j`	Active la journalisation (valable uniquement sur des systèmes non journalisés).

Option	Commentaire
`-L`	Modifie le label.
`-e <Err>`	Permet de décrire le comportement du noyau en cas d'erreur. Les valeurs sont `continue`, `panic` et `remount-ro`. Par défaut, la valeur est `continue`.
`-m <N>`	Permet d'empêcher le remplissage du système de fichiers en laissant (par défaut) 5 % de réservation au système afin de laisser travailler les daemons klogd et syslogd.
`-o ^`	Ajoute ou supprime l'option de montage indiquée (voir ci-dessus pour la journalisation).
`-u <UUID>`	Modifie la valeur de l'UUID selon un format hexadécimal. On peut le supprimer avec l'option `clear` ou en générer un nouveau avec l'option `random`.
`-s [0| 1]`	Active ou désactive le bit *sparse super feature* sur les disques de grande taille. En activant cette option, on réduit le nombre de blocs de secours et on optimise ainsi l'espace occupé sur les disques.
`-l`	Liste la configuration et le paramétrage d'un périphérique.

Exemple : configuration du système de fichiers `/dev/sda1`

```
root@Philo:/home/uadmin# tune2fs -l /dev/sda1
tune2fs 1.42.5 (29-Jul-2012)
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          33936108-2eb9-44dd-a5a0-ae6447566f4a
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      filetype sparse_super
Default mount options:    (none)
Filesystem state:         not clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              124496
Block count:              248832
Reserved block count:     12441
Free blocks:              215303
Free inodes:              124256
First block:              1
Block size:               1024
Fragment size:            1024
Blocks per group:         8192
Fragments per group:      8192
Inodes per group:         4016
Inode blocks per group:   502
Last mount time:          n/a
Last write time:          Mon Aug  4 18:02:19 2014
Mount count:              9
Maximum mount count:      30
Last checked:             Thu Jan  1 01:00:00 1970
Check interval:           0 (<none>)
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               128
```