MATHÉMATIQUES ET INFORMATIQUE
Sciences
Université Paris Cité

Chapitre 7: Les traces et l'archivage

Chérifa Boucetta

A.U: 2022-2023

Plan

- Introduction
- Les traces (logs) du système
- Archivage et backup

Introduction

- Qu’est ce qu’un log?
- Quand est ce qu’on a besoin de consulter les messages log?

Principe

- Lorsque le système démarre, fonctionne et effectue tout type d'opérations, ses actions et celles de la plupart de ses services sont tracées dans divers fichiers.
- Trois services sont spécialisés dans la réception des messages à écrire dans ces fichiers :
  - **syslogd** : system log daemon, chargé de la gestion des informations émises par tout type de service et éventuellement le noyau. Il est souvent remplacé par **syslog-ng** ou **rsyslog**.
  - **journald** : composant de systemd, chargé de collecter et d'indexer les traces provenant de tout service, via leurs fichiers ou une API, notamment systemd, mais s'interfaçant aussi avec syslog ou kmsg.
  - **klogd** : kernel log daemon, chargé de la gestion des informations émises par le noyau. Il est moins présent depuis quelques années, car rsyslog ou journald peuvent également écouter les évènements du noyau.

Les traces logs du système

- Les messages importants émis par un composant du système devraient passer par les services syslogd ou journald: les **logs.**
  - Ceci n'empêche pas, au contraire, qu'un service puisse gérer ses propres traces dans ses propres fichiers.
  - Les traces applicatives ne devraient pas être placées dans les traces de système.
  - Exemple: Les traces de connexion au système (via la console, ssh, telnet, etc.) ont un intérêt important et doivent être présentes dans les fichiers logs du système.
- Les messages reçus par les services **syslogd** ou **journald** sont redirigés selon l'émetteur, la sévérité, dans des fichiers, des consoles, sous forme de mails aux utilisateurs du système (root par exemple), etc.

Configuration du syslog

- Le fichier de configuration **/etc/syslog.conf** permet de définir **l'origine, l'importance** et **la destination de chaque message**, sous forme de deux champs.

- Les messages syslog sont inscrits dans les fichiers **/var/log/messages** et **/var/log/syslog** ou dans tout autre fichier paramétré dans **/etc/syslog.conf**.

Configuration du syslog

- **L'origine** définit un ensemble de **systèmes** et de **sous-systèmes** (noyau, services).
- La liste, extensible, est composée à l'origine des éléments suivants.

Sous-système	Signification
auth/authpriv	Service de sécurité et d'authentification.
cron	Service cron.
daemon	Les démons du système.
kern	Le noyau.
lpr	Le service d'impression.
mail	La messagerie.
news	Le réseau.
syslog	Syslog lui-même.
user	Messages des processus utilisateurs.
uucp	Unix to Unix CoPy.

Configuration du syslog

- L'importance ou **niveau** définit le niveau de sévérité du message. L'étoile définit l'ensemble de tous les niveaux. Il y a équivalence entre les niveaux émis par klogd et syslogd.

Niveau	Signification
emerg	Le système est inutilisable.
alert	Une intervention immédiate est indispensable.
crit	Erreur critique pour le sous-système.
err	Erreur de fonctionnement.
warning	Avertissement.
notice	Évènement normal méritant d'être signalé.
info	Pour information seulement.
debug	Message envoyé pour la mise au point.
none	Ignorer les messages.

Configuration du syslog

- **La destination** ou **action** peut être un fichier, un message à un utilisateur, la console, une liste d'utilisateurs… L'étoile indique tout le monde.
- **Exemples:**
  - L'exemple suivant provient d'une installation RHEL utilisant syslogd.

```
# Mails
mail.*                                          -/var/log/maillog

# Crontab
cron.*                                           /var/log/cron

# Messages d'alerte
*.emerg                                          *
```

rsyslog

- Le gestionnaire de traces **rsyslog** est utilisé en remplacement de syslog.
- Sa configuration principale est placée dans **/etc/rsyslog.conf**.
- Les configurations additionnelles sont dans **/etc/rsyslog.early.conf**, utilisé avant l'activation des couches réseau, et le contenu de **/etc/rsyslog.d/*** est chargé par le fichier de configuration principal.

```
user@Ubuntu:~$ ls /etc/rsyslog.
rsyslog.conf  rsyslog.d/
```

```
user@Ubuntu:~$ ls /etc/rsyslog.d
20-ufw.conf  21-cloudinit.conf  50-default.conf
user@Ubuntu:~$
```

Systemd et journald

- Journald est un service démarré par systemd.
- Pouvant être utilisé en complément de syslog, ou pouvant le remplacer, il est devenu un standard sur la majeure partie des distributions Linux.
- Il offre plusieurs avantages :
  - Gestion des traces du noyau via /dev/kmsg.
  - Gestion des traces des services et applications via l'interception des appels à syslog.
  - API native pour la génération de traces structurées.
  - Récupération et indexation des entrées et sorties standard des services.
  - Gestion des entrées d'audit système.

Configuration du syslog

- Les données collectées sont gérées de manière sécurisée. Par défaut, les traces sont stockées au format binaire dans le répertoire **/run/log/journal** ou **/var/log/journal**.

```
$ ls -l /run/log/journal/b8869de6c79d4b25be724208cc49a45c/system.journal
-rw-r-----+ 1 root systemd-journal 5181440 janv. 29 09:51
/run/log/journal/b8869de6c79d4b25be724208cc49a45c/system.journal
```

- Le système de fichiers /run est volatile, il est donc vidé à chaque redémarrage. Pour rendre les logs persistantes au reboot il faut les déplacer vers /var.

Les fichiers de trace

- Les logs systèmes sont situés par convention dans **/var/log**.
- Tous les fichiers de logs de ce répertoire ne proviennent pas de syslogd ou journald. C'est le cas par exemple des informations de connexion.
- Exemple du contenu de ce répertoire:
  - Il contient plusieurs fichiers textes et des répertoires. Des services peuvent décider, sans passer par **syslogd**, de concentrer et d'écrire leurs messages dans cette arborescence.

```
# cd /var/log ; ls -l
-rw-r-----  1 root  root      2460 fev  7 05:34 acpid
drwxr-x---  2 root  root      4096 mar  5  2007 audit
-rw-------  1 root  root       116 mar 27 04:02 boot.log
-rw-------  1 root  root     75487 mar 28 11:10 cron
drwxr-xr-x  2 lp    sys       4096 mar 27 04:02 cups
-rw-r--r--  1 root  root     28359 fev  7 05:34 dmesg
drwx------  2 root  root      4096 aou  7  2007 httpd
-r--------  1 root  root  18747276 mar 28 11:08 lastlog
drwxr-xr-x  2 root  root      4096 jui  1  2007 mail
-rw-------  1 root  root      4537 mar 28 04:02 maillog
-rw-------  1 root  root    178348 mar 28 11:10 messages
drwx------  2 root  root      4096 oct 16 23:21 samba
-rw-------  1 root  root    214999 mar 28 11:08 secure
-rw-r--r--  1 root  root      2734 mar 28 11:01 snmpd.log
-rw-------  1 root  root         0 mar 23 04:02 spooler
drwxr-x---  2 squid squid     4096 jan 22  2007 squid
-rw-------  1 root  root     62165 mar 28 09:13 sudo.log
drwxr-xr-x  2 root  root      4096 oct  5  2004 vbox
-rw-rw-r--  1 root  utmp    127872 mar 28 11:10 wtmp
-rw-------  1 root  root     40557 mar 28 11:03 xferlog
```

journalctl

- La commande journalctl permet d'accéder aux traces stockées par journald.
- Voici quelques exemples d'utilisation de la commande journalctl.
- Afficher toutes les logs :

```
# journalctl
```

- Attendre les nouvelles logs avec un équivalent de l'option -f de tail :

```
# journalctl -f
```

journalctl

- Afficher les traces selon le nom de l'identifiant syslog (l'application ou le service) avec le paramètre -t, c'est généralement le nom du programme ou de l'exécutable :

```
# journalctl -t sshd
janv. 16 22:29:55 ubuntu sshd[986]: Server listening on 0.0.0.0 port 22.
janv. 16 22:29:55 ubuntu sshd[986]: Server listening on :: port 22.
...
```

journalctl

- Affichage en précisant un intervalle de temps avec -S (since) et -U (until) :

```
# journalctl -u ssh -S "2020-02-01" -U "2020-02-02"
-- Logs begin at Thu 2020-01-16 22:29:38 CET, end at Sat 2020-02-08 20:24:08 CET. --
févr. 01 18:22:24 ubuntu systemd[1]: Starting OpenBSD Secure Shell server...
févr. 01 18:22:25 ubuntu sshd[1209]: Server listening on 0.0.0.0 port 22.
févr. 01 18:22:25 ubuntu systemd[1]: Started OpenBSD Secure Shell server.
févr. 01 18:22:25 ubuntu sshd[1209]: Server listening on :: port 22.
févr. 01 18:27:04 ubuntu sshd[1209]: Received signal 15; terminating.
févr. 01 18:27:04 ubuntu systemd[1]: Stopping OpenBSD Secure Shell server...
févr. 01 18:27:04 ubuntu systemd[1]: ssh.service: Succeeded.
févr. 01 18:27:04 ubuntu systemd[1]: Stopped OpenBSD Secure Shell server.
```

La commande logger

- La commande **logger** permet d'émettre des messages à traiter comme logs.
- Le résultat est visible, selon la configuration de syslog, dans /var/log/messages, ou /var/log/syslog.

```
$ logger "Message de test"
$ tail -1 /var/log/syslog
Feb  8 20:28:15 ubuntu seb: Message de test
```

Rotation des logs

- Avec le temps, les fichiers de logs finissent par prendre beaucoup de place pouvant même finir par remplir le système de fichiers.
- Il y a plusieurs raisons à cela :
  - Un programme ou service peut être naturellement très bavard, avec ses paramètres par défaut
  - Certains services sont parfois lancés en mode « debug », y compris en production.
  - Personne ne pense à purger les anciens logs, qui s'accumulent.
  - Après parfois quelques mois, années, ou tout simplement quelques heures, les logs peuvent occuper plusieurs gigaoctets. Leur nombre peut aller jusqu'à saturer un disque, ce qui peut alors avoir des conséquences désastreuses, jusqu'à rendre le système inutilisable. Il est donc nécessaire de les nettoyer régulièrement.

La commande logrotate

- La commande **logrotate** a été créée dans cet objectif.
- Elle effectue une rotation des fichiers de logs, en archivant les anciens messages dans des fichiers de sauvegarde, éventuellement elle les compresse, puis, après un certain nombre de rotations, supprime les fichiers les plus anciens.
- Les fichiers de configuration sont /etc/logrotate.conf accompagné de ceux présents dans le dossier /etc/logrotate.d/*.

Configuration du syslog

- Logrotate prend de nombreux paramètres, dont voici les plus courants :

paramètre	Signification
rotate	nombre de rotations à conserver. Une valeur de 7 garde les sept précédents fichiers. Les fichiers résultants sont numérotés en conséquence.
daily	la commande chaque jour s'exécute.
missingok	l'absence du fichier n'est pas une erreur.
notifempty	ne pas effectuer de rotation si le fichier est vide.
compress	les fichiers traités seront compressés.
delaycompress	la compression du fichier sera effectuée à la prochaine passe. C'est pour cela que syslog.1 n'est pas compressé.
postrotate	le fichier indiqué est exécuté par le shell, cela peut être une ligne de commandes ou un script. Une fois la rotation effectuée, un nouveau fichier vide est créé. Certains services doivent être rechargés pour prendre en compte ce changement.
minsize	la taille minimale du fichier nécessitant une rotation. En-dessous de cette taille, il n'y aura pas de rotation avant l'intervalle donné.
maxsize	si la taille maximale est atteinte avant l'intervalle donné, une rotation a tout de même lieu.

Configuration du syslog

- La configuration de logrotate pour le fichier /var/log/syslog :

```
/var/log/syslog
{
    rotate 7
    daily
    missingok
    notifempty
    delaycompress
    compress
    postrotate
            /usr/lib/rsyslog/rsyslog-rotate
    endscript
}
```

```
# ls -1 /var/log/syslog*
/var/log/syslog
/var/log/syslog.1
/var/log/syslog.2.gz
/var/log/syslog.3.gz
/var/log/syslog.4.gz
/var/log/syslog.5.gz
/var/log/syslog.6.gz
/var/log/syslog.7.gz
```

Plan

- Introduction
- Les traces (logs) du système
- Archivage et backup

Pourquoi sauvegarder?

- Il est très important de définir un plan de sauvegarde, en se posant les bonnes questions :
  - Que faut-il sauvegarder ?
  - Avec quelle fréquence ?
  - Combien de temps conservera-t-on les sauvegardes, à quel endroit, en combien d'exemplaires ?
  - À quel endroit sera stocké l'historique des sauvegardes ?
  - Quel est le support le plus approprié ?
  - Quels sont les besoins, en capacité, du support de sauvegarde ?
  - Combien de temps prévoit-on pour sauvegarder un fichier, un système de fichiers et est-ce raisonnable ?
  - La sauvegarde doit-elle être automatique ou manuelle ?
  - Quelle est la méthode de sauvegarde la plus appropriée ?

Les outils de sauvegarde

- La sauvegarde est un travail important de l'administrateur puisqu'en cas de problème, on passe généralement par une restauration du système depuis une sauvegarde, ou une image du système lorsque celui-ci était encore intègre (bon fonctionnement, pas de corruption).
- Chaque Unix est fourni avec des commandes et des procédures de sauvegarde qui lui sont propres.
- Pour la sauvegarde de fichiers et d'arborescences, on utilise les commandes **tar et cpio**. Ces commandes sauvegardent une arborescence, et pas un système de fichiers.
- Pour la sauvegarde physique de disques et de systèmes de fichiers (des dumps), on utilise la commande **dd**.
- Une sauvegarde incrémentale consiste à sauvegarder une première fois la totalité des données, puis ensuite uniquement les fichiers modifiés.

La commande tar

- La commande tar crée des archives des fichiers, y compris l'arborescence de fichiers, sur tout type de support y compris dans un autre fichier (archive à l'extension .tar).
- Syntaxe:

```
tar cvf nom_archive Fichier(s)
```

- Les paramètres sont les suivants :
  - c : création d'archive.
  - v : mode bavard, tar indique ce qu'il fait.
  - f : le paramètre suivant est le nom de l'archive.

La commande tar

- Par exemple pour placer dans une archive tar le répertoire Desktop :

```
$ tar cvf desktop.tar Desktop/
Desktop/
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
```

La commande tar

- Lister le contenu d'une archive:

```
tar tvf nom_archive
```

- Le paramètre t liste le contenu de l'archive.

```
$ tar tvf desktop.tar
drwx------ seb/users       0 2008-04-17 09:44 Desktop/
-rw-r--r-- seb/users     191 2007-10-20 20:10 Desktop/fusion-icon.desktop
-rw-r--r-- seb/users    4786 2007-09-26 00:43 Desktop/konsole.desktop
-rw-r--r-- seb/users     665 2008-04-08 15:14 Desktop/Support.desktop
-rw-r--r-- seb/users    1051 2007-10-05 10:16 Desktop/Office.desktop
-rw-r--r-- seb/users    4586 2007-12-05 11:37 Desktop/Terminal.desktop
-rw-r--r-- seb/users     829 2007-10-17 12:12 Desktop/MozillaFirefox.desktop
-rw-r--r-- seb/users    3952 2007-10-05 10:16 Desktop/Printer.desktop
-rw-r--r-- seb/users    2053 2007-10-05 10:16 Desktop/.directory
-rw-r--r-- seb/users     450 2007-10-23 11:58 Desktop/myComputer.desktop
-rw-r--r-- seb/users     218 2008-02-22 08:43 Desktop/trash.desktop
-rw-r--r-- seb/users     328 2008-04-08 15:14 Desktop/SuSE.desktop
-rw-r--r-- seb/users     472 2008-04-17 09:44 Desktop/Windows.desktop
```

La commande tar

- Pour restaurer le contenu d'une archive la syntaxe est :

```
tar xvf nom_archive fichiers
```

- Le paramètre **x** permet l'extraction de l'ensemble des fichiers de l'archive, ou du ou des fichiers spécifiés à la suite du nom de l'archive.

```
tar xvf desktop.tar
Desktop/
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
```

La commande tar

- La commande **tar** permet de gérer les formats de compression directement :
  - z : l'archive est compressée au format gzip.
  - Z : l'archive est compressée au format compress.
  - j : l'archive est compressée au format bzip2.
  - J : l'archive est compressée au format xz.

```
$ tar cvzf desktop.tar.gz Desktop/
Desktop/
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
$ ls -l desktop.tar*
-rw-r--r-- 1 seb users 30720 mai 9 11:16 desktop.tar
-rw-r--r-- 1 seb users 7556 mai 9 11:22 desktop.tar.gz
```

Autres commandes

- **compress et uncompress** : compression et décompression des fichiers.
- **gzip, gunzip, zcat** : compression et décompression au format GnuZip.
- **xz, unxz, xzcat** : compression et décompression au format .xz (dérivé de lzma).
- **bzip2, bunzip2, bzcat** : compression et décompression au format .bz2, plus performant que gzip.

Fin du Chapitre 6