# Administration Système UNIX
# TD7 – Syslog

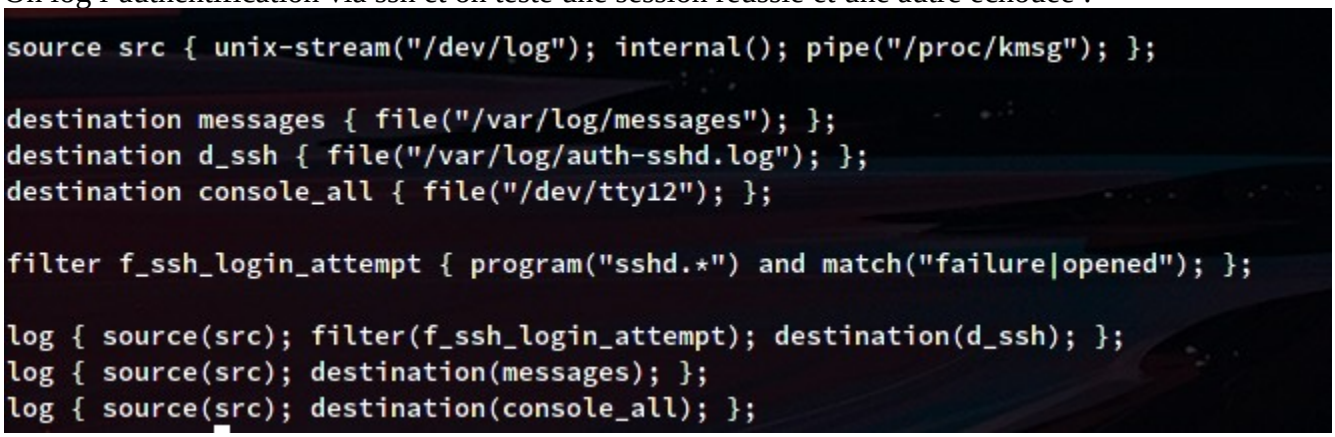
Nawfal Massine MALKI, STI 4A, TD2

## 1. Gestion des logs avec syslog-ng

**1.**
La configuration par défaut de gestions des logs permet de :

- logger : le socket unix /dev/log, le démon syslog-ng et le log buffer du kernel (/proc/kmsg).
- dans : /var/log/messages et /dev/tty12

On log l'authentification via ssh et on teste une session réussie et une autre échouée :

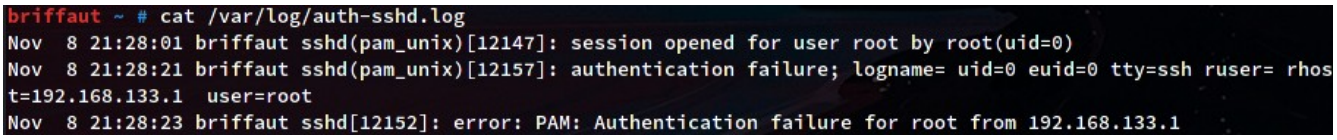


```
source src { unix-stream("/dev/log"); internal(); pipe("/proc/kmsg"); };

destination messages { file("/var/log/messages"); };
destination d_ssh { file("/var/log/auth-sshd.log"); };
destination console_all { file("/dev/tty12"); };

filter f_ssh_login_attempt { program("sshd.*") and match("failure|opened"); };

log { source(src); filter(f_ssh_login_attempt); destination(d_ssh); };
log { source(src); destination(messages); };
log { source(src); destination(console_all); };
```



```
briffaut ~ # cat /var/log/auth-sshd.log
Nov  8 21:28:01 briffaut sshd(pam_unix)[12147]: session opened for user root by root(uid=0)
Nov  8 21:28:21 briffaut sshd(pam_unix)[12157]: authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhos
t=192.168.133.1  user=root
Nov  8 21:28:23 briffaut sshd[12152]: error: PAM: Authentication failure for root from 192.168.133.1
```

**2.**
On distingue deux fichiers de trace différents : un pour les sessions réussies et un autre pour les session échouées :

```
source src { unix-stream("/dev/log"); internal(); pipe("/proc/kmsg"); };

destination messages { file("/var/log/messages"); };
destination d_ssh_success { file("/var/log/auth-sshd-success.log"); };
destination d_ssh_failure { file("/var/log/auth-sshd-failure.log"); };
destination console_all { file("/dev/tty12"); };

filter f_ssh_login_success { program("sshd.*") and match("opened"); };
filter f_ssh_login_failure { program("sshd.*") and match("failure"); };

log { source(src); filter(f_ssh_login_success); destination(d_ssh_success); };
log { source(src); filter(f_ssh_login_failure); destination(d_ssh_failure); };
log { source(src); destination(messages); };
log { source(src); destination(console_all); };
```

```
briffaut ~ # cat /var/log/auth-sshd-success.log
Nov  8 21:42:23 briffaut sshd(pam_unix)[12471]: session opened for user root by root(uid=0)
briffaut ~ # cat /var/log/auth-sshd-failure.log
Nov  8 21:42:17 briffaut sshd(pam_unix)[12469]: authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhos
t=192.168.133.1  user=root
Nov  8 21:42:19 briffaut sshd[12464]: error: PAM: Authentication failure for root from 192.168.133.1
```

**3.**

Dans le fichier **syslog-ng.conf.gentoo.hardened** :

- on distingue comme sources les logs du kernel de /dev/log et de syslog
- on considère plusieurs filtres : des filtres agissant sur les services avec la fonction facility et des filtres de priorité (info, warn, error, critic, notice) avec la fonction level()
- on concatène ces filtres pour séparer les différents niveaux d'erreurs des différents services. Par exemple on distingue, pour le service mail, les infos, les warning et les errors avec les trois combinaisons suivantes :

```
log { source(src); filter(f_mail); filter(f_info); destination(mailinfo); };
log { source(src); filter(f_mail); filter(f_warn); destination(mailwarn); };
log { source(src); filter(f_mail); filter(f_err); destination(mailerr); };
```

Ce fichier permet une gestion plus rigoureuse des flux de redirection de logs. Il est à utiliser pour logger correctement les différents services d'un serveur par exemple.

**4.**

Rotation des logs :

On crée un fichier **/etc/logrotate.d/sshd** avec le contenu suivant :

```
 GNU nano 2.0.2                    File: /etc/logrotate.d/sshd

/var/log/auth-sshd-* {
        daily
        missingok
        rotate 60
        compress
        delaycompress
        notifempty
        create 640 root
        sharedscripts
        postrotate
                /etc/init.d/sshd reload > /dev/null 2>&1 || true
        endscript
}
```

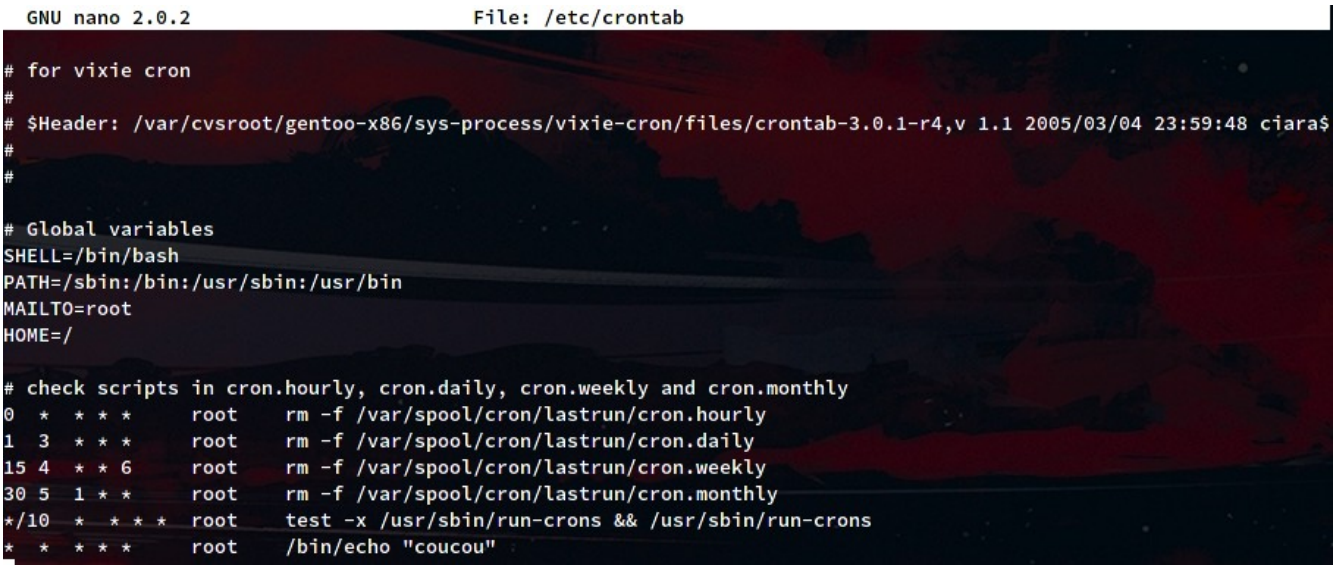
- **daily** permet d'effectuer la rotation tous les jours
- **missingok** signifie que l'absence du log n'est pas anormale. Sans cette option, l'administrateur recevra une notification comme quoi un log est introuvable.
- **Rotate 60** permet de conserver 60 fichiers de log, soit 60 jours.
- **Compress** permet de compresser les fichiers de log des jours précédents suite à une rotation
- **delaycompress** permet de reporter la compression du journal précédent à la prochaine rotation, utile quand un programme ne peut pas fermer son journal.
- **Notifempty** permet de ne pas permuter un journal s'il est vide
- **create 640 root** les fichiers, suite à une rotation, auront pour créateur root et les droits 640
- **postrotate/endscript** permet d'exécuter un script avant la permutation du journal.
- **Sharedscripts** permet d'exécuter ce script qu'une seule fois suite à une rotation

On force une rotation puis on se rend compte que les vieux logs ont été archivés dans une archive .gz. On retente une authentification ssh :

```
briffaut log # logrotate -f /etc/logrotate.conf
briffaut log # cat auth-sshd-success.log
briffaut log # cat auth-sshd-success.log
Nov  8 23:06:31 briffaut sshd(pam_unix)[14086]: session opened for user root by root(uid=0)
```

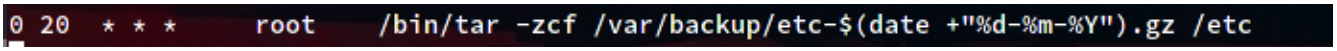
# 3. Tâches planifiées : cron

**5.**
On rajoute l'entrée suivante dans **/etc/crontab :**




Pour vérifier que la commande s'exécute bien, on redémarre le service vixie-cron, on attend deux minutes et on vérifie les logs de cron dans /var/log/messages :


**6.**




On rajoute cette entrée à crontab qui se lancera tous les soirs à 20h.
On utilise une archive gunzip car zip n'est pas installé sur la machine et est impossible à installer via emerge (version dépréciée de gentoo donc repositories pas à jour)

# 3. Gestion des utilisateurs Unix

```
briffaut etc # wc -l /etc/passwd
24 /etc/passwd
briffaut etc # cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
adm:x:3:4:adm:/var/adm:/bin/false
lp:x:4:7:lp:/var/spool/lpd:/bin/false
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/bin/false
news:x:9:13:news:/usr/lib/news:/bin/false
uucp:x:10:14:uucp:/var/spool/uucppublic:/bin/false
operator:x:11:0:operator:/root:/bin/bash
man:x:13:15:man:/usr/share/man:/bin/false
postmaster:x:14:12:postmaster:/var/spool/mail:/bin/false
smmsp:x:209:209:smmsp:/var/spool/mqueue:/bin/false
portage:x:250:250:portage:/var/tmp/portage:/bin/false
nobody:x:65534:65534:nobody:/:/bin/false
ldap:x:439:439::/usr/lib/openldap:/dev/null
sshd:x:22:22:added by portage for openssh:/var/empty:/sbin/nologin
cron:x:16:16:added by portage for cronbase:/var/spool/cron:/sbin/nologin
john:x:1000:100::/home/john:/bin/bash
ftp:x:21:21:added by portage for ftpbase:/home/ftp:/sbin/nologin
proftpd:x:101:441:added by portage for proftpd:/dev/null:/sbin/nologin
apache:x:81:81:added by portage for apache:/var/www:/sbin/nologin
```

### 3.1. Définition des Utilisateurs

**3.1.1.**
Nous avons 24 comptes utilisateurs dont un humain (john, uid 1000) et un compte root (uid 0).
Tous les comptes utilisateurs humain ont un uid supérieur ou égal à 1000.

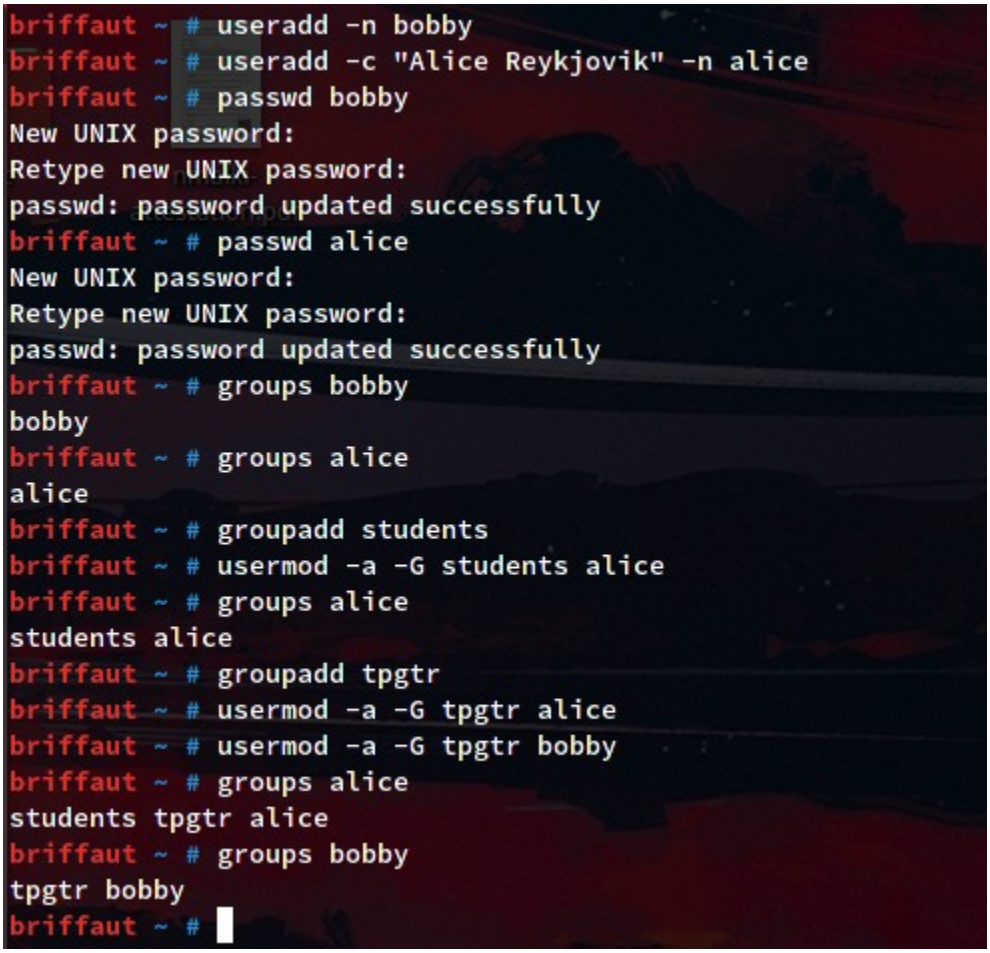
**3.1.2.**
root : /bin/bash
halt : /bin/halt
sert à arrêter le système correctement.

**3.1.3.**
/sbin/nologin est le shell des certains services auxquels les utilisateurs humains n’ont pas le droit de se connecter. /sbin/nologin indique simplement qu’il n’est pas autorisé de se connecter avec ces utilisateurs.

```
briffaut etc # /sbin/nologin
This account is currently not available.
```

### 3.2. Création des Utilisateurs et des Groupes

L'option -n de useradd permet de créer un groupe du même nom que l'utilisateur et d'ajouter l'utilisateur à ce groupe.
L'option -c permet de spécifier un commentaire, souvent utilisé pour définir le nom et prénom.
On modifie le mot de passe avec passwd, autrement il n'est pas possible de se connecter avec cet utilisateur.
On crée un groupe avec groupadd et on ajoute un utilisateur à un groupe avec usermod -a -G groupe utilisateur.
On affiche les groupes d'appartenance d'un utilisateur avec groups.




### 3.3. Droits

#### 3.3.1.

On crée un utilisateur avec l'option -n pour créer un groupe au même nom et l'option -m pour créer un home directory pour cet utilisateur.

```
briffaut ~ # useradd -n -m etudiant
```

On donne tous les droits à l'owner mais aucun droit à autrui. On ajoute -R pour la récursivité.

```
etudiant@briffaut ~ $ chmod -R 700 /home/etudiant
```

On essaye d'accéder au home directory de etudiant via le compte d'alice par exemple.

```
alice@briffaut /root $ cd /home/etudiant
bash: cd: /home/etudiant: Permission denied
```

On remet les droits ce /home/etudiant comme ils étaient.

**3.3.2.**

Au groupe etudiant, on donne les droits rx à /tmp/toto et les droits rwx à /tmp/toto/titi.
Un utilisateur du groupe etudiant peut modifier le fichier toto mais pas le supprimer car n'a pas les droits d'écriture sur /tmp/toto.

```
briffaut ~ # mkdir /tmp/toto
briffaut ~ # chgrp etudiant /tmp/toto
briffaut ~ # chmod 050 /tmp/toto
briffaut ~ # touch /tmp/toto/titi
briffaut ~ # chgrp etudiant /tmp/toto/titi
briffaut ~ # chmod 060 /tmp/toto/titi
briffaut ~ # nano /tmp/toto/titi
briffaut ~ # su alice
alice@briffaut /root $ cat /tmp/toto/titi
hey
alice@briffaut /root $ echo hola >> /tmp/toto/titi
alice@briffaut /root $ cat /tmp/toto/titi
hey
hola
alice@briffaut /root $ echo hello > /tmp/toto/titi
alice@briffaut /root $ cat /tmp/toto/titi
hello
alice@briffaut /root $ rm /tmp/toto/titi
rm: cannot remove `/tmp/toto/titi': Permission denied
alice@briffaut /root $
```

**3.4. Droits d'accès**

**3.4.1.**

```
etudiant@briffaut /root $ rm /var/log/messages
rm: remove write-protected regular file `/var/log/messages'? y
rm: cannot remove `/var/log/messages': Permission denied
etudiant@briffaut /root $ ls -l /var/log/messages
-rw------- 1 root root 285356 Nov  9 01:08 /var/log/messages
```

Il n'y a que le propriétaire (ici root) qui peut lire ou modifier le fichier /var/log/messages.

**3.4.2.**

```
etudiant@briffaut /root $ id
uid=1004(etudiant) gid=1006(etudiant) groups=1006(etudiant)
```

L'utilisateur etudiant n'appartient qu'au groupe etudiant.

**3.4.3.**

```
etudiant@briffaut ~ $ echo coucou > test
etudiant@briffaut ~ $ chmod 444 test
etudiant@briffaut ~ $ echo hey >> test
bash: test: Permission denied
etudiant@briffaut ~ $
```

**3.4.4./3.4.5.**

```
etudiant@briffaut ~ $ mkdir toto
etudiant@briffaut ~ $ echo hello > toto/titi
etudiant@briffaut ~ $ chmod 700 toto
etudiant@briffaut ~ $ ls toto
titi
etudiant@briffaut ~ $ cat toto/titi
hello
etudiant@briffaut ~ $ exit
exit
briffaut ~ # su alice
alice@briffaut /root $ ls /home/etudiant/toto
ls: cannot access /home/etudiant/toto: Permission denied
```

Les fichiers placés dans toto ne sont pas lisibles par les autres utilisateurs mais ceux-là peuvent très bien lire le contenu de titi car on a créé le fichier avant le chmod et on n'a pas spécifié l'option -R (récursif).

```
alice@briffaut /root $ cat /home/etudiant/toto/titi
hello
```

**3.4.6.**

```
briffaut ~ # find /usr/bin -perm /4000
/usr/bin/netselect
/usr/bin/expiry
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/chage
/usr/bin/newgrp
/usr/bin/gpasswd
```

Le SUID (Set Owner User Id up on execution) est un type spécial de fichiers qui permet de donner les droits de l'owner temporairement à un utilisateur pour ouvrir un fichier ou exécuter un programme. Il s'agit typiquement des commandes d'administration de groupe.

**3.5. Droits d'accès**

Dans le fichier /etc/security/limits.conf, on limite l'utilisateur bobby à 3 connexions maximum, 20 processus, 1 minutes de temps CPU.

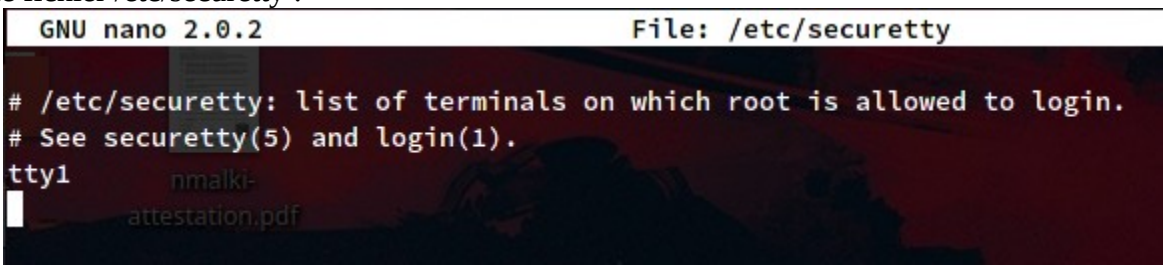
```
bobby           hard    maxlogins       3
bobby           hard    nproc           20
bobby           hard    cpu             1
```

On vérifie que nproc en ouvrant un processus une vingtaine de fois ou en exécutant un programme qui fork plusieurs fois. On peut afficher le nombre de processus d'un utilisateur avec la commande : **ps H -u [utilisateur] | wc -l**

On vérifie le max logins en ouvrant plusieurs terminaux. À partir de la 4ème connexiona avec l'user bobby, la connexion est refusée.

On vérifie le temps CPU en réalisant une boucle infinie.

Dans le fichier /etc/securetty :




Quand on essaye de se connecter sur tty2, la connexion est refusée.

Dans le fichier /etc/shells :


On peut installer un shell comme zsh et changer le shell de bobby avec chsh pour vérifier que cela ne fonctionne pas.