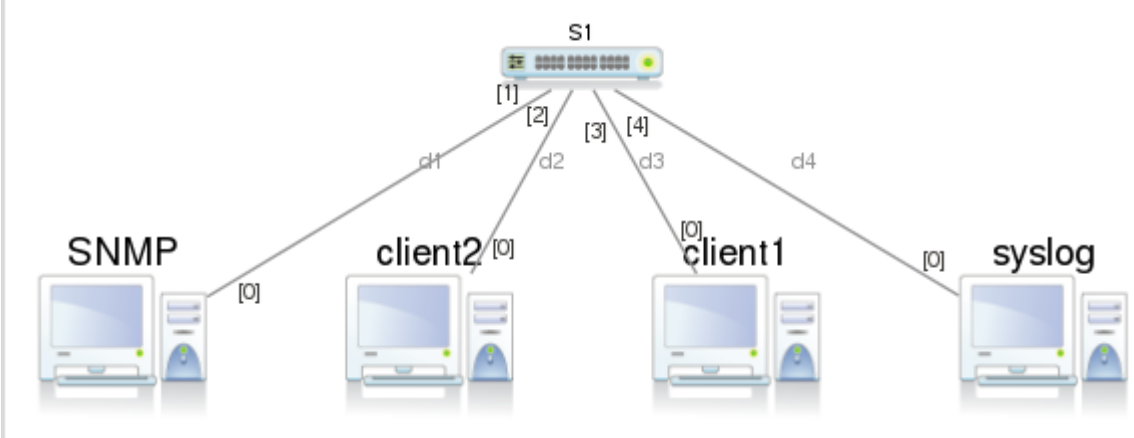
TP 4 — Syslog sous linux avec le service rsyslog

Exercice 1 — Création du réseau



Exercice 2 — Première analyse du fichier de configuration de rsyslog

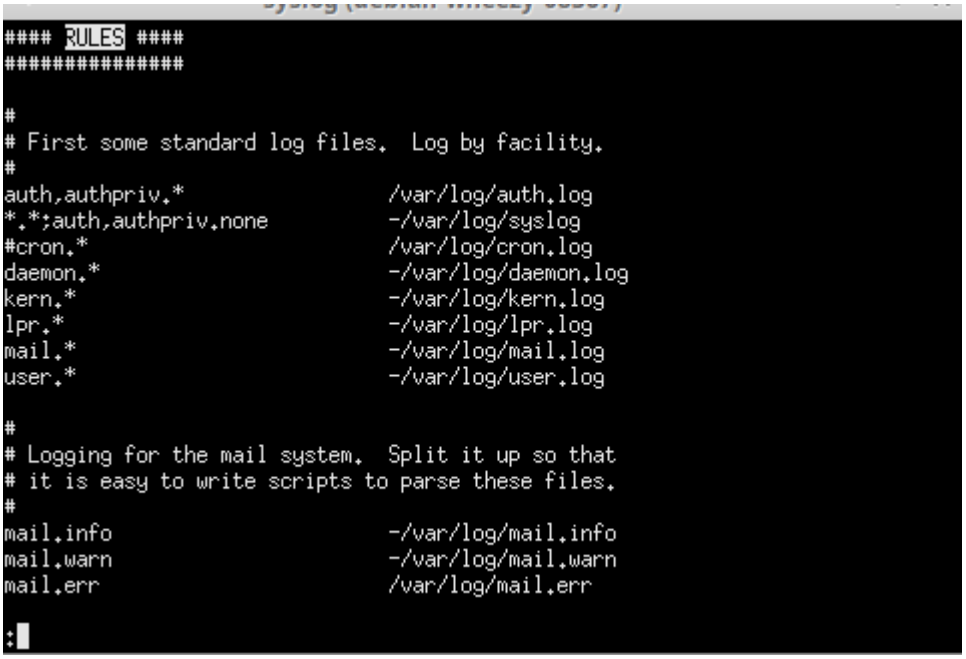
I 2.1 Démarrer le service rsyslog sur la machine syslog

```
[ ok ] Starting enhanced syslogd: rsyslogd.
```

I 2.2 Ouvrir le fichier de configuration de rsyslog (/etc/rsyslog.conf) sur la machine syslog avec la commande less. (Pour quitter less par la suite : touche q.)

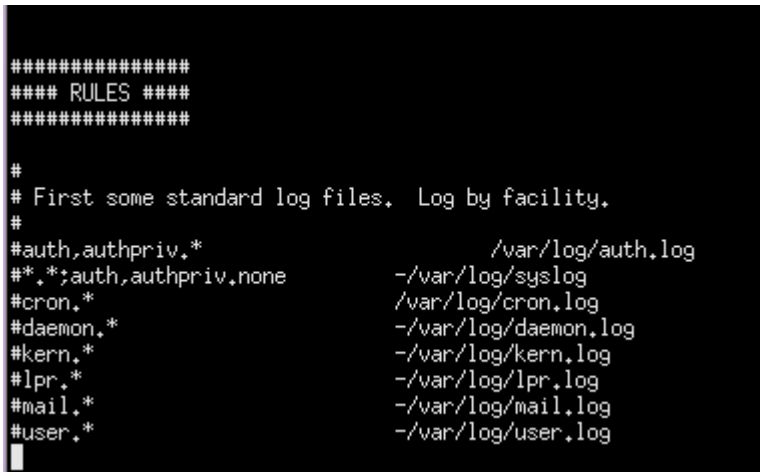
```
syslog (debian-wheezy-08367)
#  /etc/rsyslog.conf    Configuration file for rsyslog.
#
#                       For more information see
#                       /usr/share/doc/rsyslog-doc/html/rsyslog_conf.html


#################
#### MODULES ####
#################

$ModLoad imuxsock # provides support for local system logging
$ModLoad imklog   # provides kernel logging support
#$ModLoad immark  # provides --MARK-- message capability

# provides UDP syslog reception
#$ModLoad imudp
#$UDPServerRun 514

# provides TCP syslog reception
#$ModLoad imtcp
#$InputTCPServerRun 514


/etc/rsyslog.conf
```

I 2.3 Rechercher le mot RULES en tapant /RULES puis entrée

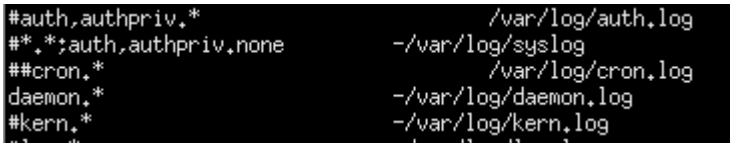


Exercice 3 — Configuration de rsyslog sur les deux clients

Client 1 et 2



1. stocker les messages de la catégorie daemon dans le fichier /var/log/daemon.log



2. stocker tous les messages sauf ceux de la catégorie daemon dans le fichier /var/log/messages.log

***.*;auth,daemon.none /var/log/messages.log**

3. et rediriger les messages de la catégorie daemon ayant un niveau error (ou plus grave) vers la machine syslog en utilisant UDP

```
##cron.*                                /var/log/cron.log
daemon.*                            -/var/log/daemon.log
daemon.err                          @10.23.1.3
*.*;auth,daemon.none                -/var/log/message.log
```

10.23.1.3 étant le serveur syslog

1. rediriger les messages de la catégorie auth ayant un niveau critical (ou plus grave) vers la machine syslog en utilisant UDP ;
2. stocker les messages de la catégorie auth ayant un niveau égal à error dans le fichier /var/log/auth-error.log ;
3. stocker les messages de la catégorie auth ayant un niveau de gravité autres que ceux décrits dans les deux premières clauses dans le fichier /var/log/auth-pas-grave.log.

Pour faire court vous pourrez trouver une capture avec toutes les modifications effectué dans le fichier rsyslog du client 2 :

```
#auth,authpriv.*                        /var/log/auth.log
#*.*;auth,authpriv.none             -/var/log/syslog
##cron.*                                /var/log/cron.log
#daemon.*                           -/var/log/daemon.log
#kern.*                             -/var/log/kern.log
#lpr.*                              -/var/log/lpr.log
#mail.*                             -/var/log/mail.log
#user.*                             -/var/log/user.log
auth.crit                           @10.23.1.3
auth.err                            -/var/log/auth-error.log
auth.*                          ;auth.err,auth.crit  /var/log/auth-pas-grave.log
```

Cette commande utilise **tcpdump**, un outil de capture de paquets réseau, pour écouter les paquets entrant et sortant sur l'interface réseau **eth0 :**

Je lance ça sur le syslog : sudo tcpdump -i eth0 port 514 -A

```
[0 root@syslog ~]$ sudo tcpdump -i eth0 port 514 -A
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
```

Cette commande utilise **logger**, un utilitaire qui envoie des messages au service **Syslog**. L'option **-p daemon.err** indique que le message est de niveau **erreur** (err) :

Je lance ça sur le client 1 : logger -p daemon.err "Test log daemon error"

```
[0 root@client1 ~]$ logger -p daemon.err "Test log daemon error"
[0 root@client1 ~]$
```

On peut voir une réponse sur le syslog :


[0 root@syslog ~]$ sudo tcpdump -i eth0 port 514 -A
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes