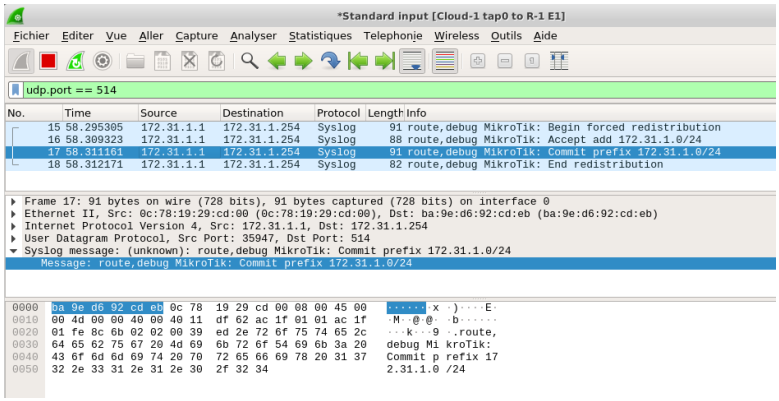
# TP - Les logs 14.03.2023

Auteur : Pascal Fougeray




## 1 Introduction

Dans ce premier TP sur GNS3, je vous propose de créer et/ou de récupérer un design déjà tout fait.
C'est selon si vous travaillez vite et/ou si vous avez travaillé chez vous.

Le design est simple, on met 2 routeurs de type Mikrotik reliés à un switch et reliés au Host via l'interface tap0 d'adresse IP 172.31.1.254/24 .

Pas juste étudier les logs !

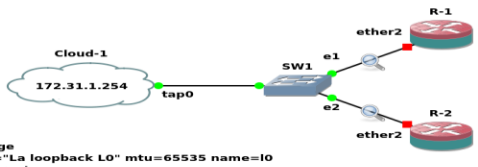
Vous allez devoir faire aussi de l'administration système et comprendre l'utilité des commandes ! ! !

## 2 TD 1ière partie

1.
2. Configuration d'un routeur, allez on lit la doc Mikrotik ☺
   https://wiki.mikrotik.com/index.php?title=Manual:System/Log&printable=yes
3. Revoir comment on installe, gère etc... un serveur, un service, ici rsyslog, les ports etc ...

## 3 TP

La figure suivante illustre ce principe, mais vous pouvez choisir une autre structure, à votre guise.



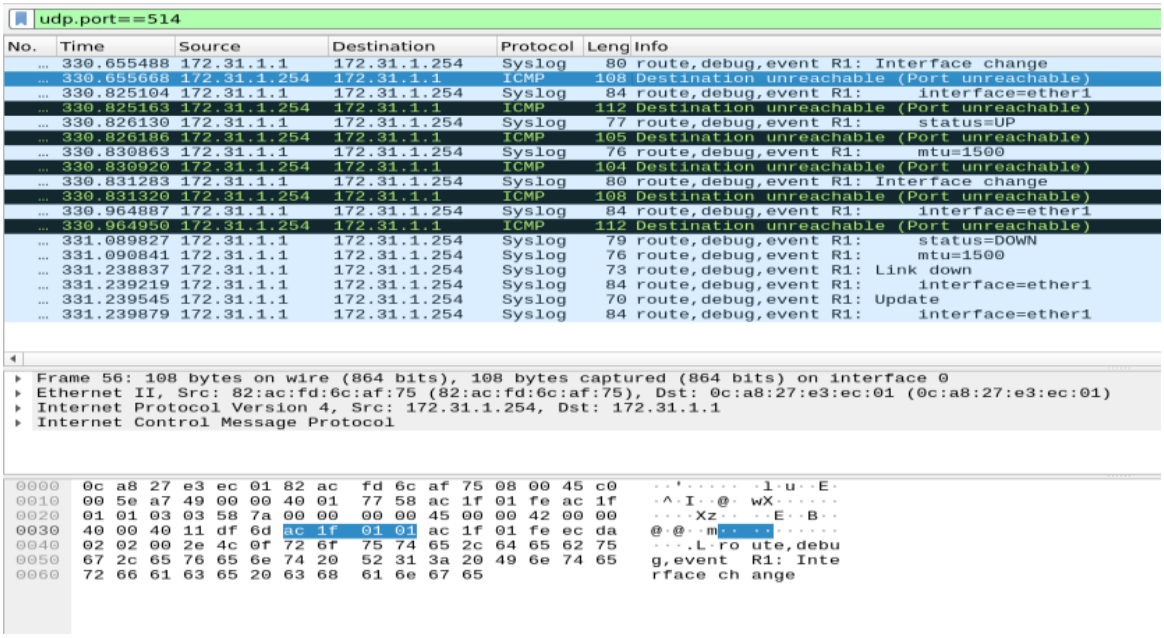

```
/interface bridge
add comment="La loopback L0" mtu=65535 name=l0
/interface ethernet
set [ find default-name=ether1 ] disable-running-check=no disabled=yes
set [ find default-name=ether2 ] disable-running-check=no
/system logging action
add name=ServeurLog remote=172.31.1.254 target=remote
/ip address
add address=1.1.1.1 interface=l0 network=1.1.1.1
add address=172.31.1.1/24 interface=ether2 network=172.31.1.0
/system clock
set time-zone-name=Europe/Paris
/system identity
set name=R1
/system logging
add action=ServeurLog prefix=R1 topics=event
```

## 3.1 Le routeur

1. **Créez** le schéma sous GNS3 ou **récupérez** le sur ecampus.
2. **Placez** tout de suite une sonde wireshark comme sur le schéma et **sélectionnez *udp.port == 514*** (pourquoi cette valeur ?)
3. **Configurez** le routeur de manière qu'il puisse pinguer le serveur de Log comme sur l'image. Ou vérifiez !

```
[admin@R1] > export
# aug/27/2019 11:49:35 by RouterOS 6.43.8
# software id =
#
#
#
/interface bridge
add comment="La loopback L0" mtu=65535 name=l0
/interface ethernet
set [ find default-name=ether1 ] disable-running-check=no
set [ find default-name=ether2 ] disable-running-check=no
set [ find default-name=ether3 ] disable-running-check=no disabled=yes
set [ find default-name=ether4 ] disable-running-check=no
set [ find default-name=ether5 ] disable-running-check=no
set [ find default-name=ether6 ] disable-running-check=no
/interface wireless security-profiles
set [ find default=yes ] supplicant-identity=MikroTik
/system logging action
add name=ServeurLog remote=172.31.1.254 target=remote
/ip address
add address=172.31.1.1/24 interface=ether1 network=172.31.1.0
add address=1.1.1.1 interface=l0 network=1.1.1.1
/snmp
set contact=pascal.fougeray@unicaen.fr enabled=yes engine-id=1664 location=\
    Campus2 src-address=172.31.1.1 trap-generators=interfaces \
    trap-interfaces=all trap-target=172.31.1.254 trap-version=3
/system clock
set time-zone-name=Europe/Paris
/system identity
set name=R1
/system logging
add action=ServeurLog prefix=R1 topics=event
[admin@R1] >
```

4. **Vérifiez** qu'il est à la même heure que le serveur Linux ! : ***system clock print*** et si pas le cas **ajustez** l'heure :
   (a) ***system clock set time-zone-name=manual time=10 :19 :00*** pour 10h19mn00s...
   (b) ou alors ***time-zone-name=Europe/Paris*** ce qui est mieux non ?
   (c) ou être client ***ntp*** : donc mettre un serveur ntp sur linux qui interrogera un autre serveur ntp... pas le temps ! ?
5. **Vérifiez** à nouveau que les 2 sont à la même heure.
6. **Configurez** le manière qu'il envoie tous les évènements (oui oui tous ! ! !) au serveur de Logs donc ***topics=event*** !
   (a) ***system logging add action=ServeurLog prefix=R-1-Mikrotik topics=event***
   (b) après ***topics=*** **appuyez** sur la touche ? ou tab et vous verrez tous les évènements possibles... comme ***error***, ***warning*** etc ...
7. **Vérifiez** que le routeur envoie bien des Logs !
   (a) **Lancez** sur le routeur ! ! ! les commandes ***interface ethernet disable ether1*** puis ***interface ethernet enable ether1***
   (b) **Visualisez** les captures sur wireshark !
   Si c'est OK, vous devez obtenir quelque chose comme cela !




(c) **Expliquez** et **justifiez** les trames de protocole **ICMP** ! ! !

8. Allez ! on passe à la suite, il nous faut un serveur qui écoute le client ☺

## 3.2 Le serveur de Log

Le client est OK, passons au serveur !

— Nous avons 2 possibilités soit utiliser **rsyslog**, soit **syslog-ng**.

— Nous allons utiliser rsyslog si vous voulez utiliser syslog-ng vous pouvez en relisant le cours appliqué à Cisco.

— https://debian-handbook.info/browse/fr-FR/stable/sect.syslog.html

1. **Vérifiez** que le port du serveur est ouvert à l'aide de la commande ***netstat -upan | grep514*** (**expliquez** le rôle de ces 4 paramètres) voir : ***https://fr.wikipedia.org/wiki/Netstat***

   Ou de la NOUVELLE commande ***ss -lu4 | grep syslog***

2. **Installez** le paquet rsyslog : ***apt install rsyslog***

3. **Lancez** dans un terminal dédié à cela la commande ***tail -f /var/log/syslog*** et **vérifiez** que des logs arrivent !

4. Sur le routeur R1 **lancez** la commande : ***interface ethernet disable ether1*** puis ***interface ethernet enable ether1***

5. Est-ce que cela s'affiche dans le terminal dédié ?

   **Nous allons valider les log de type udp !**

6. **Configurez *rsyslog***

   Ce n'est pas en 2h de TP que vous saurez le faire... Pour plus d'informations : le support de cours, les man et les docs etc...

   (a) **Faites** une sauvegarde du fichier **/etc/rsyslog.conf** ! ! ! : ***cp rsyslog.conf rsyslog.conf-SAV***

   (b) **Ouvrez** ce fichier avec nano et **expliquez** dans les grandes lignes ce que vous comprenez !

      i. Les modules

      ii. les directives globales

      iii. les règles et les droits, permissions etc !

   (c) **Dé-commentez les 2 lignes** correspondantes à ***udp*** et le ***port 514*** dans ce fichier **/etc/rsyslog.conf**

```
#################
#### MODULES ####
#################

module(load="imuxsock") # provides support for local system logging
#module(load="immark")  # provides --MARK-- message capability

# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514")

# provides TCP syslog reception
#module(load="imtcp")
#input(type="imtcp" port="514")

# provides kernel logging support and enable non-kernel klog messages
module(load="imklog" permitnonkernelfacility="on")
```

7. **Relancez** le serveur rsyslog : ***/etc/init.d/rsyslog restart*** ou ***systemctl restart rsyslog.service***
8. **Vérifiez** que tout va bien : ***systemctl status rsyslog.service***

```
root@debian-10-etu:/etc# systemctl status rsyslog.service
● rsyslog.service - System Logging Service
   Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2019-09-28 10:41:20 CEST; 7s ago
     Docs: man:rsyslogd(8)
           https://www.rsyslog.com/doc/
 Main PID: 11201 (rsyslogd)
    Tasks: 5 (limit: 4915)
   Memory: 1.0M
   CGroup: /system.slice/rsyslog.service
           └─11201 /usr/sbin/rsyslogd -n -iNONE

sept. 28 10:41:20 debian-10-etu systemd[1]: Starting System Logging Service...
sept. 28 10:41:20 debian-10-etu rsyslogd[11201]: imuxsock: Acquired UNIX socket '/run/systemd/journal/syslog' (fd 3) from systemd.  [v8.1901.0]
sept. 28 10:41:20 debian-10-etu systemd[1]: Started System Logging Service.
sept. 28 10:41:20 debian-10-etu rsyslogd[11201]:  [origin software="rsyslogd" swVersion="8.1901.0" x-pid="11201" x-info="https://www.rsyslog.com"] start
root@debian-10-etu:/etc#
```

9. **Vérifiez** que le port du serveur est ouvert à l'aide de la commande ***netstat -lu4*** !
10. **Lancez** dans un terminal dédié à cela la commande ***tail -f /var/log/syslog***
11. Sur le routeur R1 **lancez** la commande : ***interface ethernet disable ether1*** puis ***interface ethernet enable ether1***
12. Est-ce que cela s'affiche dans le terminal dédié ?

## 3.3 Le protocole syslog

Comme on a validé tous les évènements possibles sur le routeur, la moindre modification (même le fait de se loguer, enverra un message de typeLog)

1. **Lancez** les commandes ***interface ethernet disable ether1*** puis ***interface ethernet enable ether1***
2. **Relevez** une ou plusieurs trames **syslog** et **expliquez** ce que vous comprenez en vous appuyant sur le cours...
3. **Allez** voir ce qui s'est passé dans le terminal où vous avez lancé la commande ***tail -f /var/log/syslog***

```
root@PAF: ~                                                    R-1
Mar 14 09:08:24 172.31.1.1 route,debug,event R1:     interface=ether2
Mar 14 09:08:24 172.31.1.1 route,debug,event R1: Added candidate route
Mar 14 09:08:24 172.31.1.1 route,debug,event R1:     dst-prefix=172.31.1.0/24
Mar 14 09:08:24 172.31.1.1 route,debug,event R1:     attributes
Mar 14 09:08:24 172.31.1.1 route,debug,event R1:         protocol=CONNECT
Mar 14 09:08:24 172.31.1.1 route,debug,event R1:         distance=0
Mar 14 09:08:24 172.31.1.1 route,debug,event R1:         scope=10
Mar 14 09:08:24 172.31.1.1 route,debug,event R1:         target-scope=0
Mar 14 09:08:24 172.31.1.1 route,debug,event R1:         connected-net= address=172.31.1.1/24 interface=ether2
Mar 14 09:08:24 172.31.1.1 route,debug,event R1:         routing-mark=main
Mar 14 09:08:24 172.31.1.1 route,debug,event R1:         table=main
Mar 14 09:08:24 172.31.1.1 route,debug,event R1:         origin-type=CONNECTED
Mar 14 09:07:41 PAF rtkit-daemon[2568]: message repeated 7 times: [ Supervising 6 threads of 4 processes of 1 users.]
Mar 14 09:09:01 PAF CRON[48163]: (root) CMD (  [ -x /usr/lib/php/sessionclean ] && if [ ! -d /run/systemd/system ]; then /usr/lib/php/sessionclean; fi)
Mar 14 09:09:03 PAF systemd[1]: Starting Clean php session files...
Mar 14 09:09:03 PAF systemd[1]: phpsessionclean.service: Deactivated successfully.
Mar 14 09:09:03 PAF systemd[1]: Finished Clean php session files.
Mar 14 09:09:04 172.31.1.1 route,debug,event R1: Interface change
Mar 14 09:09:04 172.31.1.1 route,debug,event R1:     interface=ether1
Mar 14 09:09:04 172.31.1.1 route,debug,event R1:     status=UP
Mar 14 09:09:04 172.31.1.1 route,debug,event R1:     mtu=1500
Mar 14 09:09:04 172.31.1.1 route,debug,event R1: Update
Mar 14 09:09:04 172.31.1.1 route,debug,event R1:     interface=ether1
```

## 3.4 Mieux Gérer les logs

Imaginons que nous avons des dizaines de routeurs à gérer... la gestion des Logs deviendrait fastidieuse.

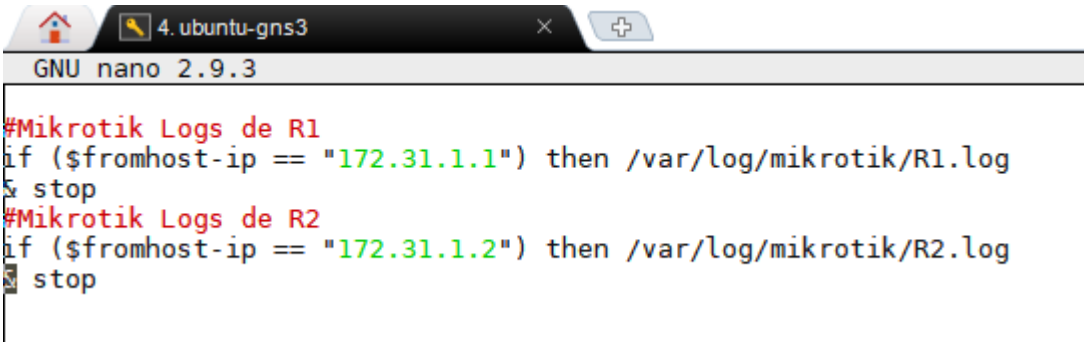
Nous allons donc générer un fichier de Log par routeur
la version anglaise du TP, et oui le prof pique sur Internet ?^ ☺
https://www.karlbooklover.com/collect-mikrotik-routeros-logs-with-rsyslog/

1. **Ajoutez** un second routeur avec la même conf pour la partie Logging mais ayant comme adresse **Ip 172.31.1.2/24 sur ether2**
2. **Créez** un fichier nommé : **/etc/rsyslog.d/10-Mikrotik.conf** et **insérez** les lignes suivantes (**l'erreur pour R1 est volontaire ! ! !**)
   ***#Mikrotik Logs de R1***
   ***if ($fromhost-ip = "172.31.1.1") then /var/log/mikrotik/R1.log***
   ***if ($fromhost-ip == "172.31.1.2") then /var/log/mikrotik/R2.log***
3. **Créez** le répertoire **/var/log/mikrotik/** *ainsi que les 2 fichiers* ***R1.Log*** *et* ***R2.Log***
4. **Changez** les permissions sur ce répertoire et ces 2 fichiers de manière que le daemon **Rsyslog** ait les droits d'accès à ces fichiers créés par le root !
   (a) ***chown root :adm -R /var/log/mikrotik***
   (b) **Expliquez** pourquoi nous faisons cela !
   (c) **Visionnez** les 2 fichiers **/etc/passwd** et **/etc/group** en lançant les commandes ***cat /etc/passwd | grep root*** et ***cat /etc/passwd | grep root***
   (d) **Expliquez** ce que vous comprenez
5. Si ça ne marche pas, vous pouvez faire un ***chmod 664*** sur les 2 fichiers ***R1.Log*** *et* ***R2.Log***
   J'ai testé sur ubuntu mais pas sur une debian... de toute manière on est là pour apprendre ☺
6. **Relancez** le serveur rsyslog : ***/etc/init.d/rsyslog restart*** ou ***systemctl restart rsyslog.service***
7. **Vérifiez** que tout va bien : ***systemctl status rsyslog.service***

```
root@debian-10-etu:/var/log# systemctl restart rsyslog
root@debian-10-etu:/var/log# systemctl status rsyslog
● rsyslog.service - System Logging Service
   Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2019-09-28 11:40:59 CEST; 5s ago
     Docs: man:rsyslogd(8)
           https://www.rsyslog.com/doc/
 Main PID: 11906 (rsyslogd)
    Tasks: 5 (limit: 4915)
   Memory: 1.0M
   CGroup: /system.slice/rsyslog.service
           └─11906 /usr/sbin/rsyslogd -n -iNONE

sept. 28 11:40:59 debian-10-etu systemd[1]: Starting System Logging Service...
sept. 28 11:40:59 debian-10-etu systemd[1]: Started System Logging Service.
sept. 28 11:40:59 debian-10-etu rsyslogd[11906]: error during parsing file /etc/rsyslog.d/10-Mikrotik.conf, on or before line 2: syntax error on token '=' [v8.1901.0 try https://www.rsyslog.
sept. 28 11:40:59 debian-10-etu rsyslogd[11906]: could not interpret master config file '/etc/rsyslog.conf'. [v8.1901.0 try https://www.rsyslog.com/e/2207 ]
sept. 28 11:40:59 debian-10-etu rsyslogd[11906]: imuxsock: Acquired UNIX socket '/run/systemd/journal/syslog' (fd 3) from systemd.  [v8.1901.0]
sept. 28 11:40:59 debian-10-etu rsyslogd[11906]:  [origin software="rsyslogd" swVersion="8.1901.0" x-pid="11906" x-info="https://www.rsyslog.com"] start
lines 1-17/17 (END)
```

8. **Corrigez** et **vérifiez** à nouveau. Pratique le ***status*** ?
9. **Vérifiez** que les logs provenant des 2 routeurs sont bien enregistrés et sauvegardés dans ces 2 fichiers !
   Avec par exemple un ***tail -f /var/log/R1.log***
10. Si vous voulez que les logs provenant des 2 routeurs ne polluent plus le fichier ***/var/log/syslog*** il suffit d'ajouter
    ***& stop***
    à la fin de chaque directive du fichier **10-Mikrotik.conf**




11. **Relancez** le serveur ***rsyslog*** !
12. **Vérifiez** !
13. **Concluez** !

# 4 Concluez

Les logs ça sert à quoi ?

Pour ceux qui voudraient se faire plaisir et aller plus loin : voir **grafana**, **kibana** et les BDD des logs
Et pour Mikrotik un article daté du 18 août 2019...
https://systemzone.net/mikrotik-send-browsing-log-to-remote-syslog-server/