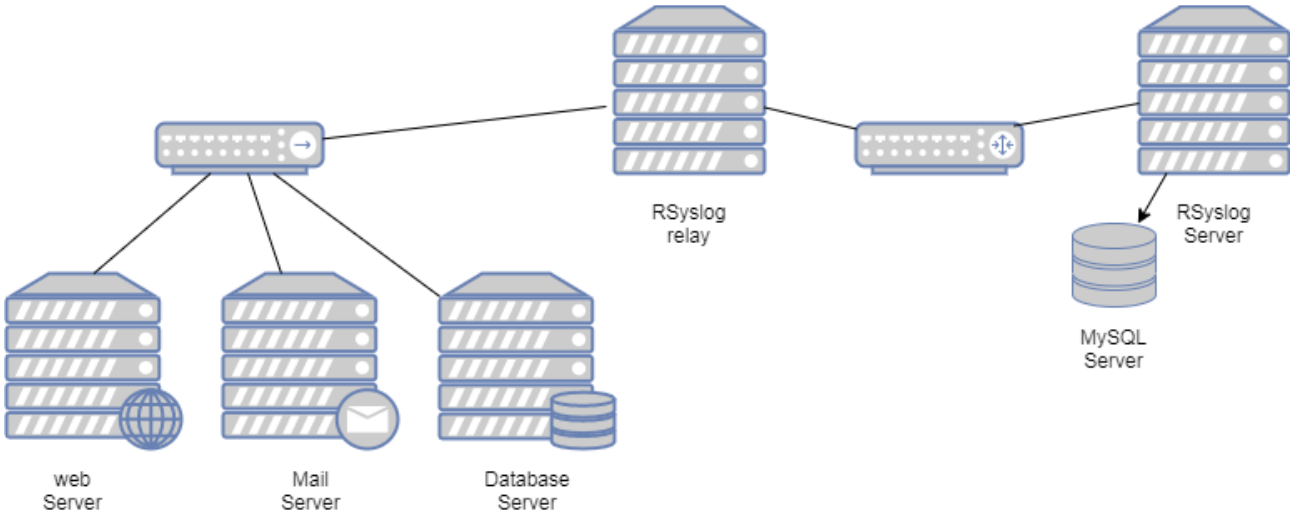
Log management – Rsyslog

Un peu de théorie

Architecture syslog



Syslog permet d'implémenter une architecture distribuée :

- des producteurs de messages : webserver, mailserver, database server, firewall, application server, network devices ...
- des relais permettant de filtrer et transférer les messages
- des collecteurs permettant de stocker les messages sous différentes formes : fichiers texte, MySQL DB, elasticsearch, mongo DB...

Syslog est défini par la RFC 5424 : https://tools.ietf.org/html/rfc5424

La RFC 3164 est obsolète.

syslog – sécurité

Il convient de connaitre les rêgles et bonne pratiques préconisées par l'ANSSI. Mais aussi de savoir les appliquer et les mettre en perspective suivant l'architecture système retenue

Bonnes pratiques – journalisation

- horodatage des événements et synchronisation des horloges
- dimensionnement de l'espace disque et partition séparée, supervision espace disque
- exportation des journaux sur une machine séparée et centralisation des journaux
- sécurisation des transferts: TCP, SSL, réeau privé d'administration
- classification des journaux en arborecence
- rotation, archivage, protection des journaux
- outil de consultation centralisé

- aspects juridiques et réglementaires

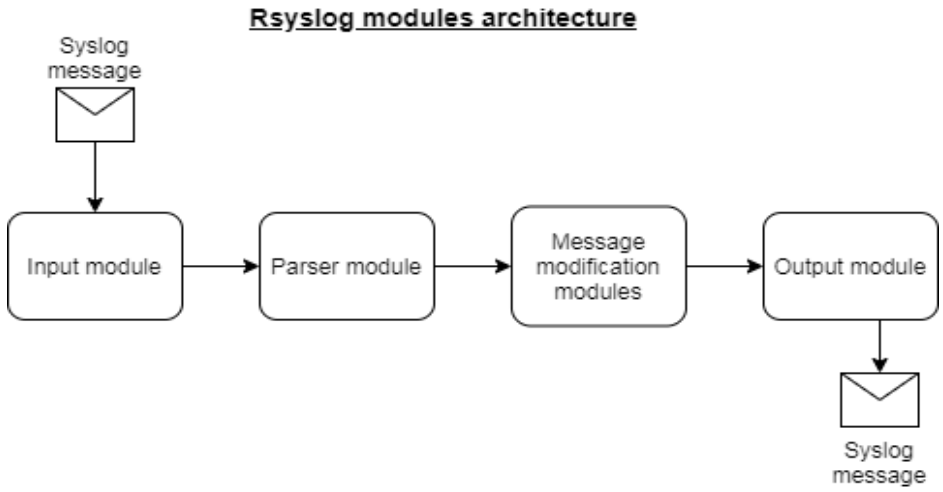
Suivre les recommendations de sécurité de l'ANSSI pour la mise en place d'un système de journalisation : https://www.ssi.gouv.fr/guide/recommandations-de-securite-pour-la-mise-en-oeuvre-dun-systeme-de-journalisation/

Mettre en oeuvre les recommendations des paragraphes 6.7.1 Configuration d'outils et services de monitoring / syslog et 6.7.3 Surveillance du système par auditd : https://www.ssi.gouv.fr/administration/guide/recommandations-de-securite-relatives-a-un-systeme-gnulinux/

syslog – application

Il existe de nombreuses applications permettant de générer, router, stocker, filtrer et interpréter des messages syslog. Nous allons utiliser rsyslog qui est installé sur les distributions ubuntu server et fedora server. D'autres applications avec des couvertures fonctionnelles différentes existent: syslog-ng, ELK, graylog, etc... Elles sont à choisir en fonction de l'architecture système implémentée.

Syslog – rsyslog



Quelques input modules:

- imdocker: Docker Input Module
- imfile: Text File Input Module
- imjournal: Systemd Journal Input Module
- imklog: Kernel Log Input Module
- imtcp: TCP Syslog Input Module
- imudp: UDP Syslog Input Module
- imuxsock: Unix Socket Input Module

quelques output modules :

- omfile: File Output Module
- ommail: Mail Output Module
- omfwd: syslog Forwarding Output Module

- omstdout: stdout output module (testbench tool)
- omusrmsg: notify users
- omjournal: Systemd Journal Output
- ommysql: MySQL Database Output Module
- omelasticsearch: Elasticsearch Output Module
- PostgreSQL Database Output Module (ompgsql)
- ommongodb: MongoDB Output Module
- omrabbitmq: RabbitMQ output module

Quelques parser modules:

- pmciscoios: Cisco IOS
- Log Message Normalization Parser Module (pmnormalize)
- pmrfc3164: Parse RFC3164-formatted messages
- pmrfc5424: Parse RFC5424-formatted messages

Quelques message modification modules:

- IP Address Anonymization Module (mmanon)
- MaxMind/GeoIP DB lookup (mmdblookup)
- JSON/CEE Structured Content Extraction Module (mmjsonparse)
- Log Message Normalization Module (mmnormalize)
- mmtaghostname: message modification module

rsyslog - /etc/rsyslong.conf

Le format sysklogd (format traditionnel)

Les lignes vides ou commençant par un commentaire # sont ignorées

Le format de la ligne est:

facility.priority ACTION

Liste des Facility:

Code	Mot-clé	Description
0	kern	messages du noyau
1	user	messages de l'espace utilisateur
2	mail	messages du système de messagerie
3	daemon	messages des processus d'arrière plan
4	auth	messages d'authentification
5	syslog	messages générés par syslogd lui-même
6	lpr	messages d'impressions
7	news	messages d'actualités
8	uucp	messages UUCP

9	cron	Taches planifiées (at/cron)
10	authpriv	sécurité / élévation de privilèges
11	ftp	logiciel FTP
12	ntp	Synchronisation du temps NTP
13	security	log audit
14	console	log alert
15	solaris-cron	Taches planifiées (at/cron)
16	local0	Utilisation locale libre 0 (local0)
17	local1	Utilisation locale libre 1 (local1)
18	local2	Utilisation locale libre 2 (local2)
19	local3	Utilisation locale libre 3 (local3)
20	local4	Utilisation locale libre 4 (local4)
21	local5	Utilisation locale libre 5 (local5)
22	local6	Utilisation locale libre 6 (local6)
23	local7	Utilisation locale libre 7 (local7)

Liste des Priority:

Code	Gravité	Mot-clé	Description
0	Emergency	emerg (panic)	Système inutilisable.
1	Alert	alert	Une intervention immédiate est nécessaire.
2	Critical	crit	Erreur critique pour le système.
3	Error	err (error)	Erreur de fonctionnement.
4	Warning	warn (warning)	Avertissement
5	Notice	notice	Événement normal méritant d'être signalé.
6	Informational	info	Pour information.
7	Debugging	debug	Message de mise au point.

Types d'Action disponibles:

- Regular File: /var/log/cron, -/var/log/maillog (output bufferisée)
- Named Pipes
- Terminal and Console: /dev/console, /dev/tty12
- Remote Machine UDP: @192.168.0.2
- Remote Machine TCP: @@192.168.0.2
- List of Users: root,admin
- Everyone logged on: *

Le format RainerScript

C'est le nouveau format de fichier de configuration. Nous verrons des exemples pour la configuration d'un collecteur rsyslog.

Configuration d'un client / producteur syslog

/etc/rsyslog.conf

```
#### RULES ####

*.* @@192.168.0.17:514          #envoi TCP tous les logs sur le serveur distant
```

Cette configuration permet d'envoyer tous les messages sur un serveur collecteur syslog. On conserve les autres lignes qui permettent d'avoir aussi les messages en local.

Configuration d'un serveur collecteur syslog

Cette configuration permet de collecter tous les messages syslog du réseau en TCP et UDP. Les messages sont stockés dans une base de données MySQL.

Il est possible de stocker les messages syslog dans une grande variété de formats et de bases de données.

Étape 1: installer MySQL server

```
$ sudo apt install mysql-server
$ sudo systemctl status mysql
? mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: en
   Active: active (running) since Thu 2020-07-09 09:25:52 UTC; 21s ago
 Main PID: 2450 (mysqld)
    Tasks: 27 (limit: 2241)
   CGroup: /system.slice/mysql.service
           +-2450 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid
Jul 09 09:25:52 vm1 systemd[1]: Starting MySQL Community Server...
Jul 09 09:25:52 vm1 systemd[1]: Started MySQL Community Server.

$ sudo mysql
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)
```

étape 2 : configurer le firewall

```
$ sudo ufw enable
$ sudo ufw allow ssh
$ sudo ufw allow 514
$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
```

```
New profiles: skip
To                         Action      From
--                         ------      ----
22/tcp                     ALLOW IN    Anywhere
514                        ALLOW IN    Anywhere
22/tcp (v6)                ALLOW IN    Anywhere (v6)
514 (v6)                   ALLOW IN    Anywhere (v6)
```

étape 3: Configurer la réception UDP et TCP sur le port 514

/etc/rsyslog.conf

```
$ cd /etc
$ sudo cp rsyslog.conf rsyslog.conf.bak
$ sudo vi rsyslog.conf

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

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

étape 4: installer le support MySQL dans rsyslog

La base de données par défaut est Syslog

```
$ sudo apt-get install rsyslog-mysql

$ sudo mysql
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| Syslog             |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

La configuration du module de sortie MySQL se trouve dans le fichier /etc/rsyslog.d/mysql.conf

```
module (load="ommysql")
*.* action(type="ommysql" server="localhost" db="Syslog" uid="rsyslog"
pwd="6OvL5kC0nc10")
```

étape 5: redemarrer le service rsyslog

```
$ sudo systemctl restart rsyslog
$ sudo systemctl status rsyslog
```

```
? rsyslog.service - System Logging Service
   Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; vendor preset:
   Active: active (running) since Thu 2020-07-09 12:52:47 UTC; 7s ago
     Docs: man:rsyslogd(8)
           http://www.rsyslog.com/doc/
 Main PID: 4448 (rsyslogd)
    Tasks: 10 (limit: 2241)
   CGroup: /system.slice/rsyslog.service
           +-4448 /usr/sbin/rsyslogd -n

Jul 09 12:52:47 vm1 systemd[1]: Starting System Logging Service...
Jul 09 12:52:47 vm1 systemd[1]: Started System Logging Service.
Jul 09 12:52:47 vm1 rsyslogd[4448]: imuxsock: Acquired UNIX socket '/run/systemd
Jul 09 12:52:47 vm1 rsyslogd[4448]: rsyslogd's groupid changed to 106
Jul 09 12:52:47 vm1 rsyslogd[4448]: rsyslogd's userid changed to 102
Jul 09 12:52:47 vm1 rsyslogd[4448]:  [origin software="rsyslogd" swVersion="8.32
```

étape 6 : tester en local

```
$ logger "TEST"

$ sudo mysql
mysql> use Syslog
Database changed
mysql> select * from SystemEvents where message like '%TEST%';
+----+------------+--------------------+--------------------+----------+----------+----------+---------
+------------+------------+-------------+-----------+---------------+---------+-----------------
+--------------+-----------+----------+----------+------------+-----------+--------------+-----------------
+----------+
| ID | CustomerID | ReceivedAt          | DeviceReportedTime  | Facility | Priority | FromHost |
Message | NTSeverity | Importance | EventSource | EventUser | EventCategory | EventID |
EventBinaryData | MaxAvailable | CurrUsage | MinUsage | MaxUsage | InfoUnitID |
SysLogTag | EventLogType | GenericFileName | SystemID |
+----+------------+--------------------+--------------------+----------+----------+----------+---------
+------------+------------+-------------+-----------+---------------+---------+-----------------
+--------------+-----------+----------+----------+------------+-----------+--------------+-----------------
+----------+
| 54 |       NULL | 2020-07-09 13:19:13 | 2020-07-09 13:19:13 |        1 |        5 | vm1      |  TEST
|       NULL |       NULL | NULL        | NULL      |          NULL |    NULL | NULL           |
NULL |      NULL |     NULL |     NULL |          1 | candidat: | NULL         | NULL           |
NULL |
+----+------------+--------------------+--------------------+----------+----------+----------+---------
+------------+------------+-------------+-----------+---------------+---------+-----------------
+--------------+-----------+----------+----------+------------+-----------+--------------+-----------------
+----------+
1 row in set (0.00 sec)
```

Facility: user

Priority: Notice

```
$ logger -p local3.info "TEST2"
```

```
$ sudo mysql
mysql> use Syslog
Database changed
mysql> select ID,ReceivedAt,Facility,Priority,FromHost,Message FROM SystemEvents
WHERE Message LIKE '%TEST2%';
+-----+---------------------+----------+----------+----------+---------+
| ID  | ReceivedAt          | Facility | Priority | FromHost | Message |
+-----+---------------------+----------+----------+----------+---------+
| 839 | 2020-07-11 10:50:05 |       19 |        6 | vm1      |  TEST2  |
+-----+---------------------+----------+----------+----------+---------+
1 row in set (0.01 sec)
```

Facility: local3

Priority: info

étape 7: tester à distance sur vm2 (fedora)

```
$ logger - -server 192.168.0.17  - -tcp - -port 514 - -priority local4.info "DISTANT1"
$ logger - -server 192.168.0.17  - -udp - -port 514 - -priority local4.info "DISTANT2"
```

Message reçu dans MySQL:

```
mysql> select ID,ReceivedAt,Facility,Priority,FromHost,Message FROM SystemEvents
WHERE Message LIKE '%DISTANT%';
+-----+---------------------+----------+----------+----------+----------+
| ID  | ReceivedAt          | Facility | Priority | FromHost | Message  |
+-----+---------------------+----------+----------+----------+----------+
| 860 | 2020-07-11 11:05:22 |       20 |        6 | vm2      | DISTANT1 |
| 862 | 2020-07-11 11:06:16 |       20 |        6 | vm2      | DISTANT2 |
+-----+---------------------+----------+----------+----------+----------+
3 rows in set (0.00 sec)
```

Description base de données Syslog

```
$ sudo mysql
mysql> use Syslog
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----------------------+
| Tables_in_Syslog      |
+-----------------------+
| SystemEvents          |
| SystemEventsProperties |
+-----------------------+
2 rows in set (0.00 sec)

mysql> desc SystemEvents;
+-------------------+-----------------+------+-----+---------+----------------+
| Field             | Type            | Null | Key | Default | Extra          |
+-------------------+-----------------+------+-----+---------+----------------+
```

```
| ID                | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| CustomerID        | bigint(20)       | YES  |     | NULL    |                |
| ReceivedAt        | datetime         | YES  |     | NULL    |                |
| DeviceReportedTime | datetime        | YES  |     | NULL    |                |
| Facility          | smallint(6)      | YES  |     | NULL    |                |
| Priority          | smallint(6)      | YES  |     | NULL    |                |
| FromHost          | varchar(60)      | YES  |     | NULL    |                |
| Message           | text             | YES  |     | NULL    |                |
| NTSeverity        | int(11)          | YES  |     | NULL    |                |
| Importance        | int(11)          | YES  |     | NULL    |                |
| EventSource       | varchar(60)      | YES  |     | NULL    |                |
| EventUser         | varchar(60)      | YES  |     | NULL    |                |
| EventCategory     | int(11)          | YES  |     | NULL    |                |
| EventID           | int(11)          | YES  |     | NULL    |                |
| EventBinaryData   | text             | YES  |     | NULL    |                |
| MaxAvailable      | int(11)          | YES  |     | NULL    |                |
| CurrUsage         | int(11)          | YES  |     | NULL    |                |
| MinUsage          | int(11)          | YES  |     | NULL    |                |
| MaxUsage          | int(11)          | YES  |     | NULL    |                |
| InfoUnitID        | int(11)          | YES  |     | NULL    |                |
| SysLogTag         | varchar(60)      | YES  |     | NULL    |                |
| EventLogType      | varchar(60)      | YES  |     | NULL    |                |
| GenericFileName   | varchar(60)      | YES  |     | NULL    |                |
| SystemID          | int(11)          | YES  |     | NULL    |                |
+-------------------+------------------+------+-----+---------+----------------+
24 rows in set (0.00 sec)
```

Quelques champs de la table SystemEvents:

- ID : ID du message
- ReceivedAt: timestamp de réception
- Facility: code facility
- Priority: code priority
- FromHost: Host d'origine du message
- Message: Message syslog

Quelques requêtes utiles:

```
mysql> select ID,ReceivedAt,Facility,Priority,FromHost,Message FROM SystemEvents
WHERE Message LIKE '%DISTANT%';

mysql> select ID,ReceivedAt,Facility,Priority,FromHost,Message FROM SystemEvents
WHERE FromHost = 'vm2' AND Facility = 20;

mysql> select ID,ReceivedAt,Facility,Priority,FromHost,Message FROM SystemEvents
WHERE FromHost = 'vm2' AND Facility = 20 AND Priority < 4;
```

Syslog – fomat de message

Exemples de messages dans /var/log/cron

```
Jul 11 00:01:01 vm2 CROND[1736]: (root) CMD (run-parts /etc/cron.hourly)
Jul 11 00:01:01 vm2 run-parts[1736]: (/etc/cron.hourly) starting 0anacron
Jul 11 00:01:01 vm2 anacron[1745]: Anacron started on 2020-07-11
Jul 11 00:01:01 vm2 anacron[1745]: Will run job `cron.weekly' in 54 min.
Jul 11 00:01:01 vm2 anacron[1745]: Will run job `cron.monthly' in 74 min.
Jul 11 00:01:01 vm2 anacron[1745]: Jobs will be executed sequentially
Jul 11 00:01:01 vm2 run-parts[1736]: (/etc/cron.hourly) finished 0anacron
```

syslog – Envoi de messages à partir des applications

syslog – Envoi de messages syslog en bash

Écriture d'un script permettant de surveiller l'espace disque utilisé sur le système. Si l'espace disque utilisé dépasse le seuil passé en paramètre alors un message de 'warning' est envoyé dans syslog, sinon c'est simplement un message de type 'info'.

```
$ vi disquelibre.sh

#!/bin/bash

pourcent=`df -h | grep '/$' | awk -F ' ' '{print $5 }' | sed 's/%//g'`
message=`echo "Disque utilisé=$pourcent% - Seuil warning=$1%"`
if [ $pourcent -le $1 ]
then
  logger -p local3.info $message
else
  message=$message" - Seuil dépassé"
  logger -p local3.warn $message
fi

$ ./disquelibre.sh 40
$ ./disquelibre.sh 30

$ sudo tail /var/log/messages
Jul 13 09:46:01 vm2 audit[1]: SERVICE_START pid=1 uid=0 auid=4294967295 ses=4294967295
subj=system_u:system_r:init_t:s0 msg='unit=dnf-makecache comm="systemd" exe="/usr/lib/systemd/systemd"
hostname=? addr=? terminal=? res=success'
Jul 13 09:46:01 vm2 audit[1]: SERVICE_STOP pid=1 uid=0 auid=4294967295 ses=4294967295
subj=system_u:system_r:init_t:s0 msg='unit=dnf-makecache comm="systemd" exe="/usr/lib/systemd/systemd"
hostname=? addr=? terminal=? res=success'
Jul 13 10:45:29 vm2 candidat[7464]: Disque utilisé=35% - Seuil warning=40%
Jul 13 10:45:39 vm2 candidat[7473]: Disque utilisé=35% - Seuil warning=30% - Seuil dépassé
```

syslog – Envoi de messages syslog en python

Exemple 1: envoyer un message syslog en python

```
$ python3
>>> print( "Hello World" )
Hello World
>>> import syslog
>>> syslog.openlog( logoption=syslog.LOG_PID, facility=syslog.LOG_LOCAL3 )
>>> syslog.syslog(syslog.LOG_WARNING, 'PYTHON1')
>>> syslog.closelog()
>>> exit()
```

Exemple 2: programme python qui exécute une commande système

```
$ vi df-python.py

import subprocess
import sys
cmd = subprocess.Popen( ["df","-h"], stdin=subprocess.PIPE, stdout=subprocess.PIPE,
stderr=subprocess.PIPE )
for line in cmd.stdout:
   print( line )

$ python3  df-python.py
```

syslog – Envoi de messages dans d'autres langages

Il est possible d'envoyer des messages syslog à partir de tous les langages de programmation.
Pour l'utilisation en langage C, voir :

```
$ man 2 syslog
```

Pour l'utilisation en PHP voir: https://www.php.net/manual/en/function.syslog