

AP - 9

Présenté par :

Yassir Chellik ET Gaetan Bracale

Année 2024 - 2025

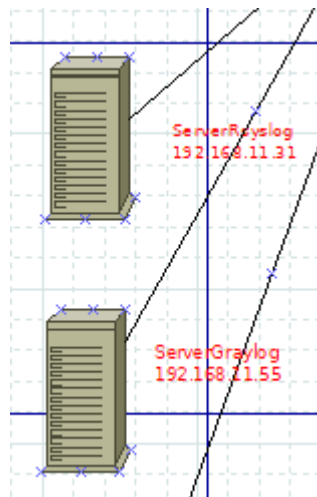
Organisation des tâches avec Gantt :

Ap – 9 Gantt Chellik Yassir Gaetan

Compléter le tableau de l'infra réseau :

AP : Yassir Gaetan Gestion VLAN et VMs du contexte "Menuimétal"

Modification du schéma réseau :



1. Comprendre l'importance des journaux (logs)

Les logs sont essentiels pour la sécurité et la gestion des systèmes, permettant de retracer les événements, d'identifier les accès non autorisés, et d'enquêter sur les incidents. Selon la CNIL, leur conservation aide à démontrer la conformité et fournit des preuves en cas d'incident de sécurité. Tous les systèmes ne génèrent pas automatiquement des logs, certains doivent être configurés. Un volume excessif de logs peut encombrer le stockage et compliquer l'analyse ; il est donc préférable de n'enregistrer que les informations essentielles. Centraliser les logs facilite la gestion et l'analyse pour détecter des anomalies, mais il est judicieux de cibler les activités critiques et les systèmes sensibles pour optimiser les ressources. En cas de piratage, les logs permettent de suivre les actions de l'attaquant et de retracer les étapes de l'attaque. L'horodatage est crucial pour contextualiser les événements, avec une synchronisation temporelle fiable essentielle pour les audits et enquêtes.

1. Est-ce que tous les systèmes produisent des logs systématiquement ? Non, tous les systèmes et logiciels ne produisent pas de logs par défaut. Certains équipements ou applications nécessitent une configuration spécifique pour activer l'enregistrement des logs.
2. Est-il judicieux de produire énormément de logs ? Non, produire trop de logs peut surcharger le stockage et compliquer l'analyse. Il est plus efficace de définir les informations essentielles à enregistrer pour assurer une sécurité sans encombrer les ressources.
3. Pourquoi est-il intéressant de centraliser les logs ? Est-il judicieux de tout centraliser ? La centralisation simplifie la gestion et l'analyse des logs, permettant de détecter des anomalies plus facilement. Cependant, tout centraliser n'est pas toujours optimal : il est préférable de se concentrer sur les systèmes critiques pour optimiser l'analyse.
4. Qu'apporte la gestion des logs en cas de piratage ? En cas de piratage, les logs sont cruciaux pour retracer les actions de l'attaquant, identifier les vulnérabilités exploitées et comprendre la séquence des événements pour améliorer la sécurité.
5. Est-ce que l'horodatage des logs est important ? Oui, l'horodatage est essentiel pour contextualiser les événements, permettant une analyse chronologique fiable et précise en cas d'incident. Une synchronisation correcte des horodatages, via un serveur NTP par exemple, est cruciale pour les enquêtes et audits de sécurité.

2. Étudier les outils de gestion des logs (Systemd vs Syslog)

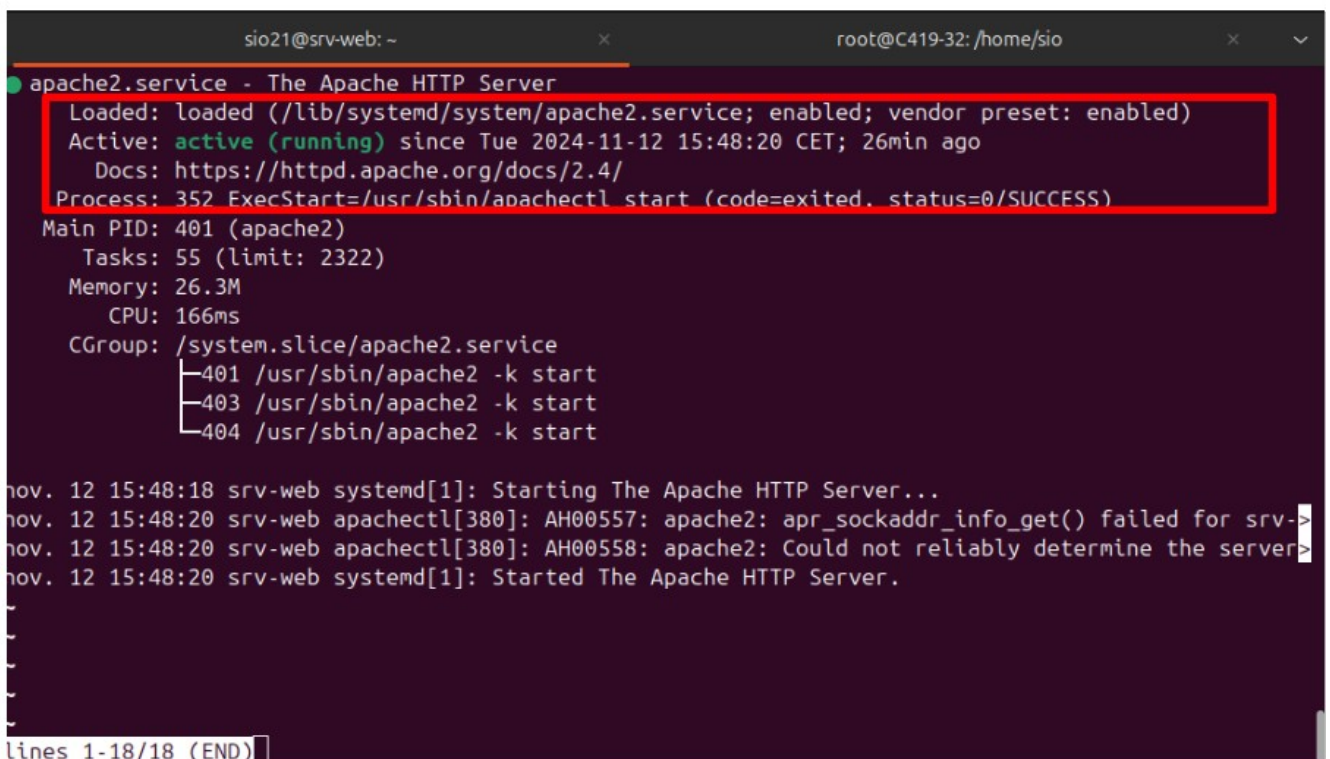
- Analyser les différences entre `systemd` et `syslog`.
- **Objectif principal :**
 - **syslog** : Système standard pour collecter et stocker les logs sous forme de fichiers texte, utilisé principalement pour centraliser les journaux d'un réseau.
 - **systemd** : Gestionnaire de services pour Linux, incluant **journald** pour la gestion des logs au sein du système. Il remplace **syslog** dans les distributions modernes de Linux.
- **Stockage des logs :**
 - **syslog** : Stocke les logs en texte brut, facile à lire mais difficile à analyser avec de gros volumes.
 - **systemd-journald** : Stocke les logs en format binaire, permettant des recherches plus rapides et des options de filtrage avancées via `journalctl`.
- **Sécurité et intégrité :**
 - **syslog** : Moins sécurisé, car les fichiers texte peuvent être modifiés facilement.
 - **systemd-journald** : Offre une meilleure sécurité avec un format binaire et des options pour signer et garantir l'intégrité des logs.
- **Performance :**
 - **syslog** : Moins performant avec de grandes quantités de logs, souvent nécessite des outils externes pour l'analyse.
 - **systemd-journald** : Plus performant, conçu pour fonctionner de manière optimale avec les systèmes modernes.
- **Compatibilité :**
 - **syslog** : Compatible avec de nombreux systèmes et outils tiers, idéal pour centraliser les logs dans un environnement multi-serveurs.
 - **systemd-journald** : Principalement local à `systemd`, mais peut être configuré pour fonctionner avec des systèmes `syslog`.

En résumé, **syslog** reste un choix classique et flexible, mais **systemd-journald** offre une solution plus moderne, sécurisée et performante pour la gestion des logs sur les systèmes Linux.

3. Réaliser les tâches sur les serveurs Linux (Apache2, SSH, MariaDB)

- Afficher les logs pour chaque service (Apache2, SSH, MariaDB).

Le premier serveur où l'on peut trouver Apache2 et SSH est notre server Web qui est dans le VLAN DMZ (192.168.12.2), Nous pouvons confirmer que le service apache 2 est activé, le ssh également car nous nous sommes connecté dessus :



```
sio21@srv-web: ~
root@C419-32: /home/sio

● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-11-12 15:48:20 CET; 26min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 352 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 401 (apache2)
   Tasks: 55 (limit: 2322)
  Memory: 26.3M
     CPU: 166ms
   CGroup: /system.slice/apache2.service
           └─401 /usr/sbin/apache2 -k start
             └─403 /usr/sbin/apache2 -k start
               └─404 /usr/sbin/apache2 -k start

Nov. 12 15:48:18 srv-web systemd[1]: Starting The Apache HTTP Server...
Nov. 12 15:48:20 srv-web apachectl[380]: AH00557: apache2: apr_sockaddr_info_get() failed for srv->
Nov. 12 15:48:20 srv-web apachectl[380]: AH00558: apache2: Could not reliably determine the server>
Nov. 12 15:48:20 srv-web systemd[1]: Started The Apache HTTP Server.

lines 1-18/18 (END)
```

Le Second serveur ou nous allons retrouver maria DB et ssh est notre serveur maria db qui est actuellement sur notre infra menuimetal dans le VLAN LAN (192.168.11.1) voici une preuve que le service maria db est actif et que le ssh également car on y est connecté dessus sur notre terminal :

```
sio21@srv-web: ~
sio21@mariadb: ~
root@mariadb:~# systemctl status mariadb
* mariadb.service - MariaDB 10.3.29 database server
  Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2024-11-05 14:10:50 CET; 1 weeks 0 days ago
  Docs: man:mysql(8)
        https://mariadb.com/kb/en/library/systemd/
  Process: 378 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysql (code=ex>
  Process: 388 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=e>
  Process: 391 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= || VAR=`cd /u>
  Process: 499 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=>
  Process: 501 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
  Main PID: 454 (mysqld)
  Status: "Taking your SQL requests now..."
  Tasks: 30 (limit: 2322)
  Memory: 296.2M
  CPU: 11min 30.267s
  CGroup: /system.slice/mariadb.service
          └─454 /usr/sbin/mysqld

Nov 05 14:10:26 mariadb systemd[1]: Starting MariaDB 10.3.29 database server...
Nov 05 14:10:35 mariadb mysqld[454]: 2024-11-05 14:10:35 0 [Note] /usr/sbin/mysqld (mysqld 10.3.29>
Nov 05 14:10:50 mariadb systemd[1]: Started MariaDB 10.3.29 database server.
Nov 05 14:10:57 mariadb debian-start[525]: WARNING: tempfile is deprecated; consider using mktemp >
lines 1-22/22 (END)
```

- Effectuer les recherches nécessaires (erreurs, connexions, taille des journaux).

Quel système de log (systemd ou syslog) est utilisé sur les équipements réseaux ?

Sur ma machine Server Web ou est installé apache2, le système de log installé est systemd et syslog :

```
root@srv-web:~# ps -p 1
  PID TTY          TIME CMD
   1  ?           00:00:01 systemd
root@srv-web:~#
```

```
root@srv-web:~# systemctl status syslog
● rsyslog.service - System Logging Service
  Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2024-11-12 15:48:19 CET; 1h 6min ago
  TriggeredBy: ● syslog.socket
  Docs: man:rsyslogd(8)
        man:rsyslog.conf(5)
        https://www.rsyslog.com/doc/
  Main PID: 359 (rsyslogd)
  Tasks: 4 (limit: 2322)
  Memory: 2.6M
  CPU: 20ms
  CGroup: /system.slice/rsyslog.service
          └─359 /usr/sbin/rsyslogd -n -iNONE
```

Sur mon serveur mariaDB le system log est systemd et syslog comme sur mon serveur web :

```
root@mariaadb:~# ps -p 1
  PID TTY          TIME CMD
   1 ?            00:00:38 systemd
root@mariaadb:~# systemctl status syslog
* rsyslog.service - System Logging Service
  Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2024-11-05 14:10:27 CET; 1 weeks 0 days ago
```

Afficher toutes les informations de journalisation pour un service

Ici nous pouvons voir grâce à la commande : `journalctl -u apache2` les infos de journalisation du service apache2 (ce qui s'est passé, quand a t'il été start and stop, quand il a échoué...)

```
- Journal begins at Tue 2021-09-28 18:29:15 CEST, ends at Tue 2024-11-12 16:29:55 CET. --
ept. 24 19:49:13 srv-web systemd[1]: Starting The Apache HTTP Server...
ept. 24 19:49:13 srv-web apachectl[16965]: AH00557: apache2: apr_sockaddr_info_get() failed for srv-web
ept. 24 19:49:13 srv-web apachectl[16965]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directive globally to localhost.
ept. 24 19:49:13 srv-web systemd[1]: Started The Apache HTTP Server.
ept. 24 19:50:12 srv-web systemd[1]: Stopping The Apache HTTP Server...
ept. 24 19:50:12 srv-web apachectl[17238]: AH00557: apache2: apr_sockaddr_info_get() failed for srv-web
ept. 24 19:50:12 srv-web apachectl[17238]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directive globally to localhost.
ept. 24 19:50:12 srv-web systemd[1]: apache2.service: Succeeded.
ept. 24 19:50:12 srv-web systemd[1]: Stopped The Apache HTTP Server.
-- Boot b806df8307cd448aaefc1abe29eed901 --
ept. 24 19:53:13 srv-web systemd[1]: Starting The Apache HTTP Server...
ept. 24 19:53:14 srv-web apachectl[392]: AH00557: apache2: apr_sockaddr_info_get() failed for srv-web
ept. 24 19:53:14 srv-web apachectl[392]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directive globally to localhost.
ept. 24 19:53:14 srv-web systemd[1]: Started The Apache HTTP Server.
ept. 24 19:57:03 srv-web systemd[1]: Stopping The Apache HTTP Server...
ept. 24 19:57:03 srv-web apachectl[668]: AH00557: apache2: apr_sockaddr_info_get() failed for srv-web
ept. 24 19:57:03 srv-web apachectl[668]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directive globally to localhost.
ept. 24 19:57:03 srv-web systemd[1]: apache2.service: Succeeded.
ept. 24 19:57:03 srv-web systemd[1]: Stopped The Apache HTTP Server.
-- Boot c2d333c0000000000000000000000000 --
ept. 24 19:57:21 srv-web systemd[1]: Starting The Apache HTTP Server...
ept. 24 19:57:24 srv-web apachectl[387]: AH00557: apache2: apr_sockaddr_info_get() failed for srv-web
ept. 24 19:57:24 srv-web apachectl[387]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directive globally to localhost.
ept. 24 19:57:24 srv-web systemd[1]: Started The Apache HTTP Server.
ept. 24 20:04:30 srv-web systemd[1]: Stopping The Apache HTTP Server...
ept. 24 20:04:30 srv-web apachectl[742]: AH00557: apache2: apr_sockaddr_info_get() failed for srv-web
ept. 24 20:04:30 srv-web apachectl[742]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directive globally to localhost.
ept. 24 20:04:30 srv-web systemd[1]: apache2.service: Succeeded.
ept. 24 20:04:30 srv-web systemd[1]: Stopped The Apache HTTP Server.
-- Boot 9620c126300f41800567626bec7ce818 --
ept. 24 20:04:47 srv-web systemd[1]: Starting The Apache HTTP Server...
ept. 24 20:04:53 srv-web apachectl[386]: AH00557: apache2: apr_sockaddr_info_get() failed for srv-web
ept. 24 20:04:53 srv-web apachectl[386]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directive globally to localhost.
ept. 24 20:04:53 srv-web systemd[1]: Started The Apache HTTP Server.
ept. 24 20:28:39 srv-web systemd[1]: Stopping The Apache HTTP Server...
ept. 24 20:29:07 srv-web apachectl[550]: AH00557: apache2: apr_sockaddr_info_get() failed for srv-web
ept. 24 20:29:07 srv-web apachectl[550]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directive globally to localhost.
ept. 24 20:29:07 srv-web systemd[1]: apache2.service: Succeeded.
ept. 24 20:29:07 srv-web systemd[1]: Stopped The Apache HTTP Server.
-- Boot dd22155c464745a99fffb1844f7e283a3 --
ept. 04 18:02:59 srv-web systemd[1]: Starting The Apache HTTP Server...
[...]
```

Afficher les informations les plus récentes pour un service

Dans ce screen ci dessous nous pouvons remarquer la date qui est très récente, et la journalisation du jour (ce qui c'est passé avec le service apache2)

```
-- Boot 1351a134560d4887940eacd776f1f3e4 --
nov. 12 15:48:18 srv-web systemd[1]: Starting The Apache HTTP Server...
nov. 12 15:48:20 srv-web apachectl[380]: AH00557: apache2: apr_sockaddr_info_get() failed for srv-web
nov. 12 15:48:20 srv-web apachectl[380]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directive globally to localhost.
nov. 12 15:48:20 srv-web systemd[1]: Started The Apache HTTP Server.
```

Afficher les informations de type « erreur » pour un service

Voici les informations de type erreur pour un service

```
root@srv-web:~# journalctl -u apache2 -p err
-- Journal begins at Tue 2021-09-28 18:29:15 CEST, ends at Tue 2024-11-12 17:17:01 CET. --
-- No entries --
```

Afficher les connexions SSH depuis le dernier démarrage :

Dans ce screen nous pouvons confirmer que c'est la date la plus récente car la connexions SSH à été faite le 14 novembre :

commande utilisé : journalctl -u ssh.service -b

```
-- Journal begins at Tue 2021-09-20 18:29:15 CEST, ends at Thu 2024-11-14 08:42:18 CET. --
nov. 12 15:48:18 srv-web systemd[1]: Starting OpenBSD Secure Shell server...
nov. 12 15:48:20 srv-web sshd[400]: Server listening on 0.0.0.0 port 22.
nov. 12 15:48:20 srv-web sshd[400]: Server listening on :: port 22.
nov. 12 15:48:20 srv-web systemd[1]: Started OpenBSD Secure Shell server.
nov. 12 16:14:20 srv-web sshd[489]: Invalid user sio from 192.168.12.251 port 45304
nov. 12 16:14:22 srv-web sshd[489]: pan_unix(sshd:auth): check pass; user unknown
nov. 12 16:14:22 srv-web sshd[489]: pan_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.12.251
nov. 12 16:14:24 srv-web sshd[489]: Failed password for invalid user sio from 192.168.12.251 port 45304 ssh2
nov. 12 16:14:29 srv-web sshd[489]: Connection closed by invalid user sio 192.168.12.251 port 45304 [preauth]
nov. 12 16:14:35 srv-web sshd[492]: Accepted password for sio21 from 192.168.12.251 port 60424 ssh2
nov. 12 16:14:35 srv-web sshd[492]: pan_unix(sshd:session): session opened for user sio21(uid=1000) by (uid=0)
nov. 12 16:29:48 srv-web sshd[639]: Connection closed by 127.0.0.1 port 57084 [preauth]
nov. 12 16:29:48 srv-web sshd[640]: Connection closed by 127.0.0.1 port 57098 [preauth]
nov. 12 16:29:48 srv-web sshd[642]: Unable to negotiate with 127.0.0.1 port 57124: no matching host key type found. Their offer: sk-ecdsa-sha2-nistp256@openssh.com [preauth]
nov. 12 16:29:48 srv-web sshd[643]: Unable to negotiate with 127.0.0.1 port 57136: no matching host key type found. Their offer: sk-ssh-ed25519@openssh.com [preauth]
nov. 13 16:25:42 srv-web sshd[1466]: Connection closed by 127.0.0.1 port 58396 [preauth]
nov. 13 16:25:42 srv-web sshd[1464]: Connection closed by 127.0.0.1 port 58370 [preauth]
nov. 13 16:25:42 srv-web sshd[1465]: Connection closed by 127.0.0.1 port 58380 [preauth]
nov. 14 08:42:11 srv-web sshd[2048]: Accepted password for sio21 from 192.168.12.251 port 40622 ssh2
nov. 14 08:42:11 srv-web sshd[2048]: pan_unix(sshd:session): session opened for user sio21(uid=1000) by (uid=0)
```

Afficher la taille totale des journaux :

Cette commande nous fournit la taille totale des logs : journalctl --disk-usage

```
root@srv-web:/home/sio21# journalctl --disk-usage
Archived and active journals take up 56.0M in the file system.
root@srv-web:/home/sio21#
```

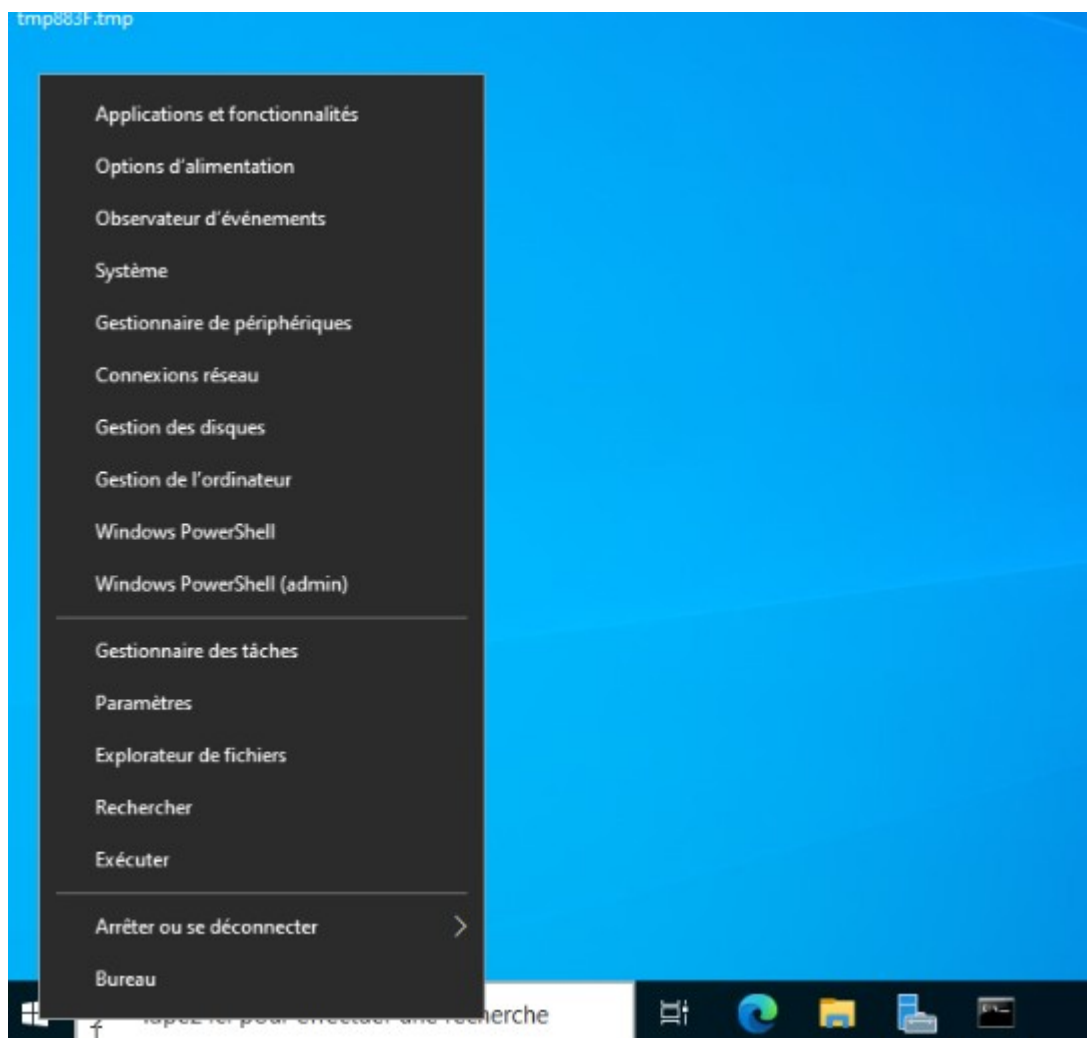
4. Réaliser les tâches sur les serveurs Windows (Active Directory, DHCP)

- Utiliser l'observateur d'événements pour rechercher les logs des services.

Il faut déjà comprendre qu'est ce que l'observateur d'événements : l'observateur d'événements enregistre tous les événements systèmes. Windows y consigne des évènements mais aussi des erreurs graves ou critiques. L'utilisateur peut ensuite consulter les journaux d'évènements.

Il y'a plusieurs résultat comme une erreur, un avertissement, ou que sa fonctionne.

Clic droit sur le menu Démarrer puis observateur d'événements :



Ensuite il faut aller dans la partie Observateur d'élément, affichage personnalisé et les rôles de server :

A gauche **1**, les journaux de Windows répartis en applications, systèmes et installation et plus bas les journaux d'applications.

En **2**, les événements du journal sélectionné dans le volet de gauche

Puis en **3**, les détails de l'événement sélectionné depuis la liste

Tout à droite, en 4 : Les boutons d'action liés au journal, comme la possibilité de créer un filtre, enregistrer tous les événements.

Enfin en 5 : Les boutons d'action comme la possibilité de copier l'événement sélectionné.

The screenshot shows the Windows Event Viewer application. The left pane (1) displays the navigation tree with 'Serveur DHCP' selected. The main pane (2) shows a list of events for the DHCP server. The bottom pane (3) shows the details for event ID 1044. The right pane (4) shows the 'Actions' menu for the selected event, with 'Rechercher...' highlighted. The bottom right pane (5) shows the 'Copier' button in the context menu.

Niveau	Date et heure	Source	ID de l'événement	Catégorie de la tâche
Information	14/11/2024 09:04:46	DHCP-Server	1044	Aucun
Avertissement	14/11/2024 09:04:46	DHCP-Server	10020	Aucun
Avertissement	14/11/2024 09:04:42	DHCP-Server	1056	Aucun
Erreur	14/11/2024 08:47:43	DHCP-Server	1059	Aucun
Erreur	14/11/2024 08:47:43	DHCP-Server	1046	Aucun
Erreur	14/11/2024 08:47:43	DHCP-Server	1059	Aucun
Avertissement	14/11/2024 08:47:43	DHCP-Server	10020	Aucun
Erreur	12/11/2024 14:38:54	DHCP-Server	1041	Aucun
Avertissement	12/11/2024 14:38:54	DHCP-Server	10020	Aucun
Erreur	12/11/2024 14:34:39	DHCP-Server	1041	Aucun
Avertissement	12/11/2024 14:34:39	DHCP-Server	10020	Aucun
Erreur	12/11/2024 13:53:54	DHCP-Server	1041	Aucun
Avertissement	12/11/2024 13:53:54	DHCP-Server	10020	Aucun
Erreur	08/11/2024 15:54:07	DHCP-Server	1041	Aucun
Avertissement	08/11/2024 15:54:07	DHCP-Server	10020	Aucun
Erreur	08/11/2024 15:32:20	DHCP-Server	1041	Aucun
Avertissement	08/11/2024 15:32:20	DHCP-Server	10020	Aucun

Détails de l'événement 1044, DHCP-Server

Général

Le service DHCP/BINL sur l'ordinateur local, appartenant au domaine administratif Windows win.menuimetal.fr, a déterminé qu'il est autorisé à démarrer le traitement des clients maintenant.

Journal : Système
Source : DHCP-Server
Événement : 1044
Niveau : Information
Utilisateur : N/A

Connecté : 14/11/2024 09:04:46
Catégorie : Aucun
Mots-clés : Classique
Ordinateur : winsrv.win.menuimetal.fr

Actions

- Ouvrir le journal enregistré...
- Créer une vue personnalisée...
- Importer une vue personnalisée...
- Filter la vue personnalisée actuelle...
- Propriétés
- Rechercher...
- Enregistrer tous les événements...
- Exporter la vue personnalisée...
- Copier la vue personnalisée...
- Joindre une tâche à cette vue...
- Affichage
- Actualiser
- Aide

Événement 1044, DHCP-Server

- Propriétés de l'événement
- Joindre une tâche à cet événement...
- Copier
- Enregistrer les événements sélectionnés...
- Actualiser
- Aide

PAS FAIT PROBLEME GENERAL !

5. Installer et configurer Graylog (outil de centralisation des logs)

- Créer une VM pour installer Graylog.
- Installer et configurer MongoDB pour Graylog.
- Configurer Graylog pour collecter les logs des serveurs Linux et Windows

6. Tester la configuration de Graylog

- Créer une première entrée de log via l'interface web de Graylog.

7. Configurer les serveurs OMV et GLPI pour transmettre les logs à Graylog

- Configurer rsyslog sur les serveurs Linux (OMV, GLPI) pour transmettre les logs à Graylog.
- Tester la transmission des logs.

8. Configurer les serveurs Windows pour transmettre les logs à Graylog (NXLog)

- Installer et configurer NXLog sur les serveurs Windows.
- Vérifier que les logs sont bien transmis à Graylog.

9. Installer et configurer Rsyslog pour la centralisation des logs

- Créer un serveur Rsyslog central.
- Configurer les serveurs Linux pour envoyer leurs logs vers le serveur Rsyslog.

Nous allons commencer par définir ce qu'est Rsyslog :

- Rsyslog est un programme open source comprenant un serveur Syslog, qui permet la centralisation de l'ensemble des logs de divers serveurs, switches, ou firewalls. Il se base sur le protocole Syslog (System Logging Protocol) qui sert à l'envoi et la réception des fichiers du journal Système ou des messages liés à des événements. Il permet de réaliser des requêtes pour en extraire des informations et générer des tableaux de bord.

Dans ce screen nous pouvons voir l'interface réseau de notre serveur rsyslog, grâce à la commande `ip a`, cette VM se situe dans le VLAN LAN, 431, 11.0, et nous pouvons également constater que nous avons un accès ssh.

```
root@serv-syslog:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:79:c8:e4 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 192.168.11.31/24 brd 192.168.11.255 scope global ens18
        valid_lft forever preferred_lft forever
    inet6 fe80::be24:11ff:fe79:c8e4/64 scope link
        valid_lft forever preferred_lft forever
root@serv-syslog:~#
```

Ensuite procédons à l'installation de rsyslog avec la commande :
`apt install -y rsyslog` et vérification du paquet avec la commande
`dpkg -l | grep rsyslog`

```
root@serv-syslog:~# dpkg -l | grep rsyslog
ii rsyslog                        8.2302.0-1                amd64        reliable system and kernel logging daemon
root@serv-syslog:~#
```

Ensuite la fichier va nous crée plusieurs fichiers de configuration dont le principale rsyslog.conf, il faudra le configuré avec la commande **vim /etc/rsyslog.conf** en activant les modules pour l'écoute des logs en ajoutant ou décommentant les lignes suivantes pour accepter les connexions depuis le réseau :

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

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

Ensuite procéder à une vérification du status de rsyslog :

```
root@serv-syslog:~# systemctl status rsyslog.service
● rsyslog.service - System Logging Service
   Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; preset: enabled)
   Active: active (running) since Thu 2024-11-14 15:15:39 CET; 5s ago
   TriggeredBy: ● syslog.socket
   Docs: man:rsyslogd(8)
         man:rsyslog.conf(5)
         https://www.rsyslog.com/doc/
   Main PID: 7505 (rsyslogd)
   Tasks: 10 (limit: 2306)
   Memory: 1.1M
   CPU: 5ms
   CGroup: /system.slice/rsyslog.service
           └─7505 /usr/sbin/rsyslogd -n -iNONE

nov. 14 15:15:39 serv-syslog systemd[1]: Starting rsyslog.service - System Logging Service...
nov. 14 15:15:39 serv-syslog systemd[1]: Started rsyslog.service - System Logging Service.
nov. 14 15:15:39 serv-syslog rsyslogd[7505]: imuxsock: Acquired UNIX socket '/run/systemd/journal/syslog' (fd 3) from systemd. [v8.2302.0]
nov. 14 15:15:39 serv-syslog rsyslogd[7505]: [origin software="rsyslogd" swVersion="8.2302.0" x-pid="7505" x-info="https://www.rsyslog.com"] start
root@serv-syslog:~#
```

J'ai ensuite supervisionner ma machine dans nagios en installant le nrpe et glpi en installant l'agent glpi (désolé j'ai perdu le screen pour glpi faite semblant que vous le voyez...)

Current Network Status

Last Updated: Thu Nov 14 15:05:23 CET 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as *nagiosadmin*

[View Service Status Detail For All Host Groups](#)
[View Status Overview For All Host Groups](#)
[View Status Summary For All Host Groups](#)
[View Status Grid For All Host Groups](#)







Host Status Totals

Up	Down	Unreachable	Pending
4	2	0	0
<i>All Problems</i>		<i>All Types</i>	
2		6	

Service Status Totals

Ok	Warning
17	0
<i>All Problems</i>	
2	

Limit Results: 100 ▾

Host ↑↓	Status ↑↓	L
debianclt	 DOWN	1
localhost	 UP	1
omvdebian	 UP	1
serverdhcpdebian	 DOWN	1
serverradius	 UP	1
serversyslog	 UP	1

J'ai ensuite référencer ma VM sur le DNS :


```
~
root@dns:~# nslookup
> serv-syslog
Server:          192.168.12.1
Address:         192.168.12.1#53

Name:   serv-syslog.menuimetal.fr
Address: 192.168.11.31
>
```

10. Récupérer les logs depuis les serveurs Linux (MariaDB, Web, SSH)

- Configurer les logs des serveurs Linux (MariaDB, Apache2/WordPress/Dokuwiki, SSH) pour les envoyer vers le serveur Rsyslog.

Pour la première étape il faut vérifier que rsyslog est bien installé sur ma machine qui doit envoyé les logs donc il faut regarder le status du service rsyslog

```
root@mariadb:/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 Thu 2024-11-14 16:09:28 CET; 4min 51s ago
   TriggeredBy: * syslog.socket
     Docs: man:rsyslogd(8)
           man:rsyslog.conf(5)
           https://www.rsyslog.com/doc/
   Main PID: 7820 (rsyslogd)
     Tasks: 4 (limit: 2322)
    Memory: 960.0K
       CPU: 7ms
   CGroup: /system.slice/rsyslog.service
           └─7820 /usr/sbin/rsyslogd -n -iNONE

Nov 14 16:09:28 mariadb systemd[1]: rsyslog.service: Succeeded.
Nov 14 16:09:28 mariadb systemd[1]: Stopped System Logging Service.
Nov 14 16:09:28 mariadb systemd[1]: Starting System Logging Service...
Nov 14 16:09:28 mariadb rsyslogd[7820]: inuxsock: Acquired UNIX socket '/run/systemd/journal/syslog' (fd 3) from systemd. [v8.2102.0]
Nov 14 16:09:28 mariadb rsyslogd[7820]: [origin software="rsyslogd" swVersion="8.2102.0" x-pid="7820" x-info="https://www.rsyslog.com"] start
Nov 14 16:09:28 mariadb systemd[1]: Started System Logging Service.
root@mariadb:/etc#
```

Il faut ensuite configurer le fichier `/etc/rsyslog.conf` de ma machine mariadb en rajoutant la ligne

```
*.*@@192.168.11.31:514
```

qui va permettre de envoyer les logs vers le serveur central (serversyslog)

```
#ligne
*.* @@192.168.11.31:514
"rsyslog.conf" 96L, 2007B
```

Il ne faut pas oublier de restart le service rsyslog !!!!!

Ensuite il faut aller sur notre machine serveur syslog et voir le fichier où l'on reçoit les logs donc le fichier /var/log/syslog et notamment chercher la partit mariadb pour voir ce qui c'est passé :

On peut apercevoir que les logs nous montre le systemctl status que l'on a fait pour le service syslog qui à succéder :

```
2024-11-14T16:09:28+01:00 mariadb systemd[1]: Stopping System Logging Service...
2024-11-14T16:09:28+01:00 mariadb systemd[1]: rsyslog.service: Succeeded.
2024-11-14T16:09:28+01:00 mariadb rsyslogd: [origin software="rsyslogd" swVersion="8.2102.0" x-pid="380" x-info="https://www.rsyslog.com"] exiting on signal 15.
2024-11-14T16:09:28+01:00 mariadb systemd[1]: Stopped System Logging Service.
2024-11-14T16:09:28+01:00 mariadb systemd[1]: Starting System Logging Service...
2024-11-14T16:09:28+01:00 mariadb rsyslogd: imuxsock: Acquired UNIX socket '/run/systemd/journal/syslog' (fd 3) from systemd. [v8.2102.0]
2024-11-14T16:09:28+01:00 mariadb rsyslogd: [origin software="rsyslogd" swVersion="8.2102.0" x-pid="7820" x-info="https://www.rsyslog.com"] start
2024-11-14T16:09:28+01:00 mariadb systemd[1]: Started System Logging Service.
/mariadb
```

Ensuite nous allons faire de même pour wordpress :

```
root@srvwebwordpress:/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 Thu 2024-11-14 16:29:40 CET; 6s ago
 TriggeredBy: ● syslog.socket
   Docs: man:rsyslogd(8)
        man:rsyslog.conf(5)
        https://www.rsyslog.com/doc/
 Main PID: 61570 (rsyslogd)
   Tasks: 4 (limit: 2322)
  Memory: 2.9M
     CPU: 13ms
  CGroup: /system.slice/rsyslog.service
          └─61570 /usr/sbin/rsyslogd -n -iNONE

nov. 14 16:29:39 srvwebwordpress systemd[1]: Starting System Logging Service...
nov. 14 16:29:40 srvwebwordpress rsyslogd[61570]: imuxsock: Acquired UNIX socket '/run/systemd/journal/syslog' (fd 3) from systemd. [v8.2102.0]
nov. 14 16:29:40 srvwebwordpress rsyslogd[61570]: [origin software="rsyslogd" swVersion="8.2102.0" x-pid="61570" x-info="https://www.rsyslog.com"] start
nov. 14 16:29:40 srvwebwordpress systemd[1]: Started System Logging Service.
root@srvwebwordpress:/etc#
```

```
#ligne
*.* @@192.168.11.31:514
"rsyslog.conf" 96L, 2008B
```

```
2024-11-14T16:29:39+01:00 srvwebwordpress systemd[1]: Stopping System Logging Service...
2024-11-14T16:29:39+01:00 srvwebwordpress systemd[1]: rsyslog.service: Succeeded.
2024-11-14T16:29:39+01:00 srvwebwordpress rsyslogd: [origin software="rsyslogd" swVersion="8.2102.0" x-pid="351" x-info="https://www.rsyslog.com"] exiting on signal 15.
2024-11-14T16:29:39+01:00 srvwebwordpress systemd[1]: Stopped System Logging Service.
2024-11-14T16:29:39+01:00 srvwebwordpress systemd[1]: rsyslog.service: Consumed 1.320s CPU time.
2024-11-14T16:29:39+01:00 srvwebwordpress systemd[1]: Starting System Logging Service...
2024-11-14T16:29:40+01:00 srvwebwordpress rsyslogd: imuxsock: Acquired UNIX socket '/run/systemd/journal/syslog' (fd 3) from systemd. [v8.2102.0]
2024-11-14T16:29:40+01:00 srvwebwordpress rsyslogd: [origin software="rsyslogd" swVersion="8.2102.0" x-pid="61570" x-info="https://www.rsyslog.com"] start
2024-11-14T16:29:40+01:00 srvwebwordpress systemd[1]: Started System Logging Service.
2024-11-14T16:30:01+01:00 srvwebwordpress CRON[61579]: (www-data) CMD ([ -x /usr/share/awstats/tools/update.sh ] && /usr/share/awstats/tools/update.sh)
2024-11-14T16:30:02+01:00 srvwebwordpress CRON[61578]: (CRON) info (No MTA installed, discarding output)
```

Nous allons ensuite envoyer les logs d'apache en envoyant les logs d'erreur et d'accès sur le serveur de logs. Nous allons donc dans la configuration de notre vhost et ajouter (ou modifier si les lignes existent) les lignes CustomLog et ErrorLogs comme suivant :

Ici, on redirige les logs d'erreur et les logs d'accès vers l'outil qui permet d'écrire les logs en affectant le mot clé "apache" et la facilitie "local6".

```
ErrorLog "/usr/bin/logger -t apache -p local6.info"
CustomLog "|/usr/bin/logger -t apache -p local6.info" combined
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
"/etc/apache2/sites-available/000-default.conf" 33L, 1449B
```

Dans un deuxième temps, nous allons aller dans le fichier **"/etc/rsyslog.conf"** , on indique à rsyslog d'envoyer tous les logs avec **la facilitie local6** vers le serveur **192.168.11.31**. On peut ensuite générer des logs en allant sur notre serveur web correspondant au vhost configuré.

```
#ligne
*. * @@192.168.11.31:514
local6.* @192.168.11.31
"/etc/rsyslog.conf" 97L, 2032B
```

Sur le serveur, ces logs arriveront dans **/var/log/syslog**, on pourra les identifier via le mot clé "apache" précisé et le nom de l'hôte :

Dans ces logs ont a comme information le service apache2 qui

fonctionne, et qui s'arrête, et qui redémarre encore...

```
2024-11-14T16:42:18+01:00 srvwebwordpress systemd[1]: Started System Logging Service.
2024-11-14T16:42:33+01:00 srvwebwordpress systemd[1]: Stopping The Apache HTTP Server...
2024-11-14T16:42:33+01:00 srvwebwordpress apachectl[61689]: AH00558: apache2: Could not reliably det
directive globally to suppress this message
2024-11-14T16:42:33+01:00 srvwebwordpress systemd[1]: apache2.service: Succeeded.
2024-11-14T16:42:33+01:00 srvwebwordpress systemd[1]: Stopped The Apache HTTP Server.
2024-11-14T16:42:33+01:00 srvwebwordpress systemd[1]: apache2.service: Consumed 48.210s CPU time.
2024-11-14T16:42:33+01:00 srvwebwordpress systemd[1]: Starting The Apache HTTP Server...
2024-11-14T16:42:33+01:00 srvwebwordpress apachectl[61695]: AH00558: apache2: Could not reliably det
directive globally to suppress this message
2024-11-14T16:42:33+01:00 srvwebwordpress systemd[1]: Started The Apache HTTP Server.
```

Pour la récupération des logs ssh d'un serveur (j'ai utilisé wordpress) il faut aller dans le fichier `/etc/rsyslog.conf` et rajouter la règle :
`auth,authpriv.* @@@192.168.11.31:514`

Nous pouvons remarquer les logs ssh, notamment l'autorisation de connexion avec le nom d'utilisateurs et également L'IP source mais il y'a également le SU pour passer en root...

```
2024-11-14T17:00:57+01:00 srvwebwordpress sshd[61756]: Accepted password for sio21 from 192.168.11.251 port 46324 ssh2
2024-11-14T17:00:57+01:00 srvwebwordpress sshd[61756]: Accepted password for sio21 from 192.168.11.251 port 46324 ssh2
2024-11-14T17:00:57+01:00 srvwebwordpress sshd[61756]: pam_unix(sshd:session): session opened for user sio21(uid=1000) by (uid=0)
2024-11-14T17:00:57+01:00 srvwebwordpress sshd[61756]: pam_unix(sshd:session): session opened for user sio21(uid=1000) by (uid=0)
2024-11-14T17:00:57+01:00 srvwebwordpress systemd-logind[353]: New session 2014 of user sio21.
2024-11-14T17:00:57+01:00 srvwebwordpress systemd-logind[353]: New session 2014 of user sio21.
2024-11-14T17:01:28+01:00 srvwebwordpress su: (to root) sio21 on pts/0
2024-11-14T17:01:28+01:00 srvwebwordpress su: pam_unix(su-l:session): session opened for user root(uid=0) by sio21(uid=1000)
2024-11-14T17:01:28+01:00 srvwebwordpress su: (to root) sio21 on pts/0
2024-11-14T17:01:28+01:00 srvwebwordpress su: pam_unix(su-l:session): session opened for user root(uid=0) by sio21(uid=1000)
```

11. Analyser les logs avec un script Bash

- Créer un script Bash pour analyser les logs SSH (connexions erronées).

Voici notre script bash, dans ce script nous pouvons retrouver divers information comme : Tout d'abord, le script vérifie si l'on a bien fourni un argument, à savoir le nom de l'utilisateur pour lequel on veut analyser les logs.

```
if [ "$#" -ne 1 ]; then
    echo "Usage: $0 <nom_utilisateur>"
    exit 1
fi
```

Explication :

"\$#" est une variable qui contient le nombre d'arguments passés au script.

-ne 1 signifie "si ce n'est pas égal à 1".

Si l'utilisateur n'a pas fourni un seul argument (le nom d'utilisateur), le script affiche un message d'erreur et arrête son exécution avec exit 1.

Ensuite, le script définit des variables pour les fichiers de logs et l'utilisateur :

```
UTILISATEUR=$1
LOG_SSH="/var/log/auth.log"
```

Explication :

UTILISATEUR=\$1 : L'argument que vous avez donné en entrée (le nom de l'utilisateur) est stocké dans la variable UTILISATEUR.

LOG_SSH

contiennent les chemins des fichiers de logs où sont enregistrées les connexions SSH

Vérification de l'Existence des Fichiers de Log

Le script vérifie ensuite si les fichiers de logs existent, sinon il arrête le script et affiche une erreur.

```
if [ ! -f "$LOG_SSH" ]; then
    echo "Fichier log SSH non trouvé : $LOG_SSH"
    exit 1
fi
```

Explication :

-f vérifie si un fichier existe.

Si l'un des fichiers n'existe pas, le script affiche un message d'erreur et s'arrête.

Analyse des Connexions Erronées pour SSH

Ensuite, le script cherche et affiche les tentatives de connexion SSH échouées pour l'utilisateur spécifié. Ces échecs peuvent être dus à une mauvaise combinaison login/mot de passe.

```
echo "== Connexions SSH erronées pour l'utilisateur $UTILISATEUR =="
```

```
grep -i "Failed password" "$LOG_SSH" | grep "$UTILISATEUR" |
awk '{print "Adresse IP:", $NF}' | sort | uniq -c
```

Explication :

`grep -i "Failed password" "$LOG_SSH"` recherche dans le fichier des logs SSH toutes les lignes contenant "Failed password", qui signifie qu'une tentative de connexion SSH a échoué.

`grep "$UTILISATEUR"`

filtre les résultats pour ne garder que ceux liés à l'utilisateur spécifié.

`awk '{print "Adresse IP:", $NF}'`

extrait l'adresse IP de chaque ligne de log.

`$NF` est une variable d'awk qui contient la dernière colonne (l'adresse IP dans ce cas).

`sort`

trie les adresses IP par ordre croissant.

`uniq -c`

compte combien de fois chaque adresse IP apparaît, ce qui permet de voir combien de tentatives échouées proviennent de chaque adresse IP.

Fin du Script

Le script affiche simplement un message indiquant qu'il a terminé.

```
echo "== Script terminé =="
```

```
#!/bin/bash

# Vérification du nombre d'arguments
if [ "$#" -ne 1 ]; then
    echo "Usage: $0 <nom_utilisateur>"
    exit 1
fi

# Variables
UTILISATEUR=$1
LOG_SSH="/var/log/auth.log" # Chemin du log SSH

# Vérification de l'existence du fichier de log SSH
if [ ! -f "$LOG_SSH" ]; then
    echo "Fichier log SSH non trouvé : $LOG_SSH"
    exit 1
fi

# 1. Analyser les logs SSH pour les connexions erronées de l'utilisateur
echo "== Connexions SSH erronées pour l'utilisateur $UTILISATEUR =="

# Extraire les tentatives de connexion échouées (mot de passe ou login incorrect)
grep -i "Failed password" "$LOG_SSH" | grep "$UTILISATEUR" | awk '{print "Adresse IP:", $(NF-3)}' | sort | uniq -c

echo "== Script terminé =="
```

- Identifier l'adresse IP source des erreurs de connexion.

Nous avons tester depuis nos deux poste le script en question, avec l'utilisateur sio il nous donne les connections échoués, le nombre de fois et l'adresse qui se connecte :

```
root@serv-syslog:~# bash script.sh sio
== Connexions SSH erronées pour l'utilisateur sio ==
      1 Adresse IP: 192.168.11.250
      2 Adresse IP: 192.168.11.251
== Script terminé ==
root@serv-syslog:~#
```

dans le fichier de log `/var/log/auth.log` nous pouvons voir les tentative de connexion qui ont échoué avec le nom d'utilisateur et l'ip du poste mais également ce qui ont été accepté :

```
2024-11-15T14:33:09.084159+01:0 serv-syslog sshd[8704]: Invalid user toto from 192.168.11.251 port 34658
2024-11-15T14:33:11.954644+01:0 serv-syslog sshd[8704]: pam_unix(sshd:auth): check pass; user unknown
2024-11-15T14:33:11.954976+01:0 serv-syslog sshd[8704]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.11.251
2024-11-15T14:33:14.473165+01:0 serv-syslog sshd[8704]: Failed password for invalid user toto from 192.168.11.251 port 34658 ssh2
2024-11-15T14:33:17.403438+01:0 serv-syslog sshd[8704]: Connection closed by invalid user toto 192.168.11.251 port 34658 [preauth]
2024-11-15T14:33:24.183449+01:0 serv-syslog sshd[8708]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.11.251 user=sio
2024-11-15T14:33:26.156015+01:0 serv-syslog sshd[8708]: Failed password for sio from 192.168.11.251 port 36000 ssh2
2024-11-15T14:33:28.515334+01:0 serv-syslog sshd[8708]: Accepted password for sio from 192.168.11.251 port 36000 ssh2
2024-11-15T14:33:28.516597+01:0 serv-syslog sshd[8708]: pam_unix(sshd:session): session opened for user sio(uid=1000) by (uid=0)
2024-11-15T14:33:28.533472+01:0 serv-syslog systemd-logind[427]: New session 38 of user sio.
2024-11-15T14:33:28.553076+01:0 serv-syslog sshd[8708]: pam_env(sshd:session): deprecated reading of user environment enabled
2024-11-15T14:33:33.309619+01:0 serv-syslog su[8718]: (to root) sio on pts/0
2024-11-15T14:33:33.311289+01:0 serv-syslog su[8718]: pam_unix(su-l:session): session opened for user root(uid=0) by sio(uid=1000)
```

12. Configurer le commutateur HP pour envoyer les logs à Rsyslog

- Configurer le commutateur HP pour envoyer ses logs via Rsyslog au serveur

L'envoi des logs se fera via **UDP** donc il faut activer sur le fichier **/etc/rsyslog.conf** mais il est déjà fait nous avons montré sa dans le screen de se fichier, ensuite il faut configurer le switch avec la commande **logging** qui configure l'envoi des logs générés par l'équipement à par exemple un serveur distant : **logging facility local0** qui d'identifier facilement les logs d'équipements réseau dans un système de journalisation centralisé comme Syslog et **logging 192.168.11.31** pour ciblé le serveur distant

```
Debug Logging
Destination:
Logging --
  192.168.11.31
  Facility = local0

Enabled debug types:
event

commutateur#
```

Nous pouvons ensuite voir les logs du commutateur avec la commande `show logging`, cela nous montre plusieurs information comme le tftp qui est activé et actif, ou les vlans configuré avec les ports et IP affectés...

```
--- Event Log listing: Events Since Boot ---
01/01/90 00:00:04 sys: 'System reboot due to Power Failure'
01/01/90 00:00:04 system: -----
01/01/90 00:00:04 system: System went down without saving crash information
01/01/90 00:00:15 udpf: DHCP relay agent feature enabled
01/01/90 00:00:15 stack: Stack Protocol enabled
01/01/90 00:00:15 tftp: Enable succeeded
01/01/90 00:00:15 system: System Booted.
01/01/90 00:00:15 cdp: CDP enabled
01/01/90 00:00:15 lldp: LLDP - enabled
01/01/90 00:52:07 ports: port 6 is Blocked by LACP
01/01/90 00:52:10 ports: port 6 is now on-line
01/01/90 00:52:10 vlan: gestion virtual LAN enabled
01/01/90 00:52:10 ip: gestion: network enabled on 192.168.13.20
01/01/90 01:31:45 ports: port 6 is now off-line
01/01/90 01:31:45 vlan: gestion virtual LAN disabled
01/01/90 01:31:45 ip: gestion: network disabled on 192.168.13.20
01/01/90 01:32:10 ports: port 23 is Blocked by LACP
01/01/90 01:32:13 ports: port 23 is now on-line
01/01/90 01:32:13 vlan: DEFAULT_VLAN virtual LAN enabled
01/01/90 01:32:13 vlan: gestion virtual LAN enabled
01/01/90 01:32:13 vlan: lan virtual LAN enabled
01/01/90 01:32:13 vlan: invites virtual LAN enabled
01/01/90 01:32:14 ip: DEFAULT_VLAN: network enabled on 192.168.10.110
01/01/90 01:32:14 ip: gestion: network enabled on 192.168.13.20
01/01/90 01:32:40 ports: port 23 is now off-line
01/01/90 01:32:40 vlan: DEFAULT_VLAN virtual LAN disabled
01/01/90 01:32:40 vlan: gestion virtual LAN disabled
01/01/90 01:32:40 vlan: lan virtual LAN disabled
01/01/90 01:32:40 vlan: invites virtual LAN disabled
01/01/90 01:32:41 ip: DEFAULT_VLAN: network disabled on 192.168.10.110
01/01/90 01:32:41 ip: gestion: network disabled on 192.168.13.20
01/01/90 01:32:44 ports: port 5 is Blocked by LACP
01/01/90 01:32:44 ports: port 5 is now off-line
01/01/90 01:32:48 ports: port 6 is Blocked by LACP
01/01/90 01:32:50 ports: port 6 is now on-line
01/01/90 01:32:50 vlan: gestion virtual LAN enabled
01/01/90 01:32:51 ip: gestion: network enabled on 192.168.13.20
01/01/90 01:32:51 ports: port 6 is now off-line
01/01/90 01:32:51 vlan: gestion virtual LAN disabled
01/01/90 01:32:51 ip: gestion: network disabled on 192.168.13.20
ommutateur#
```

- Vérifier la transmission des logs.

Dans le server syslog il faut aller dans le fichier de configuration donc `/etc/rsyslog.conf` et rajouter la règle : `local0.*`
`/var/log/switch-hp.log` pour stocker les logs de la "local0" dans un fichier spécifique :

```
local0.* -/var/log/switch-hp.log
```

Ensuite redémarrer le service rsyslog et se connecter via telnet à notre switch pour crée des logs, ensuite un fichier du nom de `switch-hp.log` va se crée avec toutes les infos logs qu'ont a faite comme par exemple la connexion via telnet:

```
root@serv-syslog:/var/log# ls
alternatives.log  apt          btmp        cron.log    dpkg.log.1  installer   kern.log    private    runit      switch-hp.log  user.log
alternatives.log.1  auth.log    btmp.1     dpkg.log    faillog     journal     lastlog    README    samba     syslog         wtmp
root@serv-syslog:/var/log#

root@serv-syslog:/var/log# cat switch-hp.log
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.251 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.251 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 ports: port 6 is now off-line
2024-01-01T02:31:46+01:00 192.168.13.20 vlan: gestion virtual LAN disabled
2024-01-01T02:31:46+01:00 192.168.13.20 ip: gestion: network disabled on 192.168.13.20
2024-01-01T02:31:46+01:00 192.168.13.20 ports: port 23 is Blocked by LACP
2024-01-01T02:31:46+01:00 192.168.13.20 ports: port 23 is now on-line
2024-01-01T02:31:46+01:00 192.168.13.20 vlan: DEFAULT_VLAN virtual LAN enabled
2024-01-01T02:31:46+01:00 192.168.13.20 vlan: gestion virtual LAN enabled
2024-01-01T02:31:46+01:00 192.168.13.20 vlan: lan virtual LAN enabled
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:51+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.251 - MANAGER Mode
2024-01-01T02:33:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:35:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:37:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:39:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
root@serv-syslog:/var/log#
```

13. Configurer le serveur de temps NTP pour synchroniser les équipements

- Configurer le serveur NTP sur le commutateur HP.

Nous allons référencer le server NTP Afin d'exploiter les logs correctement, il est nécessaire que que chaque système (équipements réseau ou système d'exploitation) soient synchronisés avec un serveur de temps (protocole NTP) en l'occurrence se sera ntp.unice.fr :

Nous allons commencer par ping le nom de domaine du serveur afin d'avoir son adresse IP :

```
sio@C419-32:~$ ping ntp.unice.fr
PING ntp.unice.fr (134.59.1.5) 56(84) bytes of data.
64 bytes from ntp.unice.fr (134.59.1.5): icmp_seq=1 ttl=46 time=18.8 ms
64 bytes from ntp.unice.fr (134.59.1.5): icmp_seq=2 ttl=46 time=18.9 ms
^C
--- ntp.unice.fr ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 18.845/18.855/18.865/0.010 ms
sio@C419-32:~$
```

Ensuite il faut configurer le commutateur et référencer le serveur de temps avec son adresse IP , La commande timezone permet d'ajuster le fuseau horaire, ici "+60" car nous sommes à "GMT +1" en hiver. La commande time permet de vérifier l'heure synchronisée.

```
commutateur(config)# timesync sntp
commutateur(config)# sntp unicast
commutateur(config)# sntp server 134.59.1.5
commutateur(config)# time timezone +60
commutateur(config)# show sntp
```

Sntp Configuration

```
Time Sync Mode: Sntp
SNTP Mode : Unicast
Poll Interval (sec) [720] : 720
```

IP Address	Protocol Version
-----	-----
134.59.1.5	3

```
commutateur(config)# time
Mon Jan  1 04:11:09 1990
commutateur(config)#
```

Pour Linux :

```
root@srvgraylog:~# timedatectl set-timezone Europe/Paris
root@srvgraylog:~#
```

16. Mettre en place la sauvegarde des logs sur le serveur OMV

- Configurer la sauvegarde des logs sur le serveur OMV pour garantir la persistance des données.

Création d'une sauvegarde à été faite pour notre serveur glpi, nous pouvons retrouver cette sauvegarde sur `/srv/dev[...]/SauvegardeWebGLPI/`

SauvegardeRsyslog	/dev/md0	SauvegardeRsyslog/	/srv/dev-disk-by-id-md-name-omvdebian-0/ SauvegardeRsyslog	✓
-------------------	----------	--------------------	---	---

preuve sur le terminal de la machine :

```
root@omvdebian:/srv/dev-disk-by-id-md-name-omvdebian-0/SauvegardeRsyslog# ls
root@omvdebian:/srv/dev-disk-by-id-md-name-omvdebian-0/SauvegardeRsyslog#
```

Création d'une analyse d'accès sur notre anti-virus pour notre partage de fichié :

Services	Antivirus	Analyse à l'accès
+		
Activé		Dossier partagé ^
✓		SauvegardeRsyslog

Il faut installer rsync sur ma machine serveur syslog :

```
root@serv-syslog:/var/log# dpkg -l | grep rsync
ii  rsync                    3.2.7-1          amd64        fast, versatile, remote (and local) file-copying tool
root@serv-syslog:/var/log#
```

pour les sauvegarde distante via rsync il faut utiliser cette commande : `rsync -e ssh -aruvz root@192.168.11.31:/var/log`

ATTENTION : Pour que la commande rsync fonctionne il faut activer le root sur ma machien cible donc la machine serv-syslog, en allant dans le fichier de configuration ssh /etc/ssh/sshd_config et en décommentant la ligne PermitRootLogin et rajouter « yes »

(Je vais envoyé tout le répertoire de ma machine syslog vers mon répertoire de sauvegarde crée... :

```
root@omvdebian:/srv/dev-disk-by-id-md-name-omvdebian-0/SauvegardeRsyslog# ls
README      alternatives.log.1  auth.log  btmp.1   dpkg.log  faillog   journal  lastlog  runit  switch-hp.log  user.log
alternatives.log  apt                btmp      cron.log  dpkg.log.1  installer  kern.log  private  samba  syslog         wtmp
root@omvdebian:/srv/dev-disk-by-id-md-name-omvdebian-0/SauvegardeRsyslog# vim switch-hp.log
root@omvdebian:/srv/dev-disk-by-id-md-name-omvdebian-0/SauvegardeRsyslog#
```

On peut comfirmer que les logs se sont bien partagé sur notre répertoire de sauvegarde :

```
root@omvdebian:/srv/dev-disk-by-id-md-name-omvdebian-0/SauvegardeRsyslog# cat switch-hp.log
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.251 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.251 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.251 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:46+01:00 192.168.13.20 ports: port 6 is now off-line
2024-01-01T02:31:46+01:00 192.168.13.20 vlan: gestion virtual LAN disabled
2024-01-01T02:31:46+01:00 192.168.13.20 ip: gestion: network disabled on 192.168.13.20
2024-01-01T02:31:46+01:00 192.168.13.20 ports: port 23 is Blocked by LACP
2024-01-01T02:31:46+01:00 192.168.13.20 ports: port 23 is now on-line
2024-01-01T02:31:46+01:00 192.168.13.20 vlan: DEFAULT_VLAN virtual LAN enabled
2024-01-01T02:31:46+01:00 192.168.13.20 vlan: gestion virtual LAN enabled
2024-01-01T02:31:46+01:00 192.168.13.20 vlan: lan virtual LAN enabled
2024-01-01T02:31:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:31:51+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.251 - MANAGER Mode
2024-01-01T02:33:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:35:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:37:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:39:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:41:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:43:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:45:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:47:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:49:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
2024-01-01T02:51:46+01:00 192.168.13.20 mgr: SME TELNET from 192.168.13.7 - MANAGER Mode
root@omvdebian:/srv/dev-disk-by-id-md-name-omvdebian-0/SauvegardeRsyslog#
```