TD7 – Langages de script

Abdallah Ammar

12 janvier 2026

Objectifs du TD

Ce TD a pour objectif d'apprendre à analyser des fichiers de logs similaires à ceux que l'on trouve sur un système Linux réel. À l'issue de la séance, vous devrez être capables de :

- lire et comprendre des logs système ;
- utiliser `grep`, `cut`, `sort`, `uniq` de manière combinée ;
- écrire un script bash générant un rapport lisible ;
- raisonner sur des données textuelles réelles.

> Les fichiers utilisés dans ce TD sont fournis par l'enseignant. N'utilisez pas directement les fichiers de `/var/log` de votre système.

Récupération des fichiers de logs

Créez un répertoire `data` et téléchargez les fichiers de logs :

```
$ mkdir data $ cd data
$ wget https://raw.githubusercontent.com/AbdAmmar/LDS/main/TD/TD7/data/auth.log $
wget https://raw.githubusercontent.com/AbdAmmar/LDS/main/TD/TD7/data/syslog $ wget
https://raw.githubusercontent.com/AbdAmmar/LDS/main/TD/TD7/data/kernel.log
```

1 Lecture et observation des logs

Travail à faire

1. Affichez le contenu de `auth.log`.
2. Combien de lignes contient ce fichier ?
3. Repérez les informations suivantes :
   - date et heure ;
   - service concerné ;
   - message.

2 Recherche d'événements

Travail à faire

À partir de `auth.log` :

1. Affichez les tentatives de connexion échouées.
2. Affichez les connexions réussies.
3. Affichez les lignes contenant `sudo`.

Commandes utiles

```
$ grep Failed auth.log $ grep Accepted auth.log $ grep sudo auth.log
```

3 Analyse par pipelines

Travail à faire

1. Comptez le nombre total de tentatives de connexion échouées.
2. Affichez la liste des utilisateurs concernés.

Exemples

```
$ grep Failed auth.log | wc -l $ grep Failed auth.log | cut -d" " -f9 | sort | uniq
```

4 Analyse temporelle

Travail à faire

1. Recherchez les événements ayant eu lieu à une heure donnée (par exemple 10h).
2. Comptez le nombre d'événements sur cette plage horaire.

```
$ grep "Jan 10 10" auth.log
```

5 Script : génération d'un rapport de sécurité

Travail à faire

Écrivez un script `report.sh` qui :
— prend un répertoire de logs en paramètre ;
— analyse `auth.log` ;
— affiche :
  — le nombre de connexions réussies ;
  — le nombre de connexions échouées ;
  — la liste des utilisateurs impliqués ;
— produit un rapport lisible à l'écran.

Exemple indicatif de sortie

```
=== Rapport de sécurité === Connexions réussies : 3 Connexions échouées : 5
Utilisateurs concernés : alice bob
```

6 Extensions (optionnelles)

— analyser `syslog` ou `kernel.log` ;
— détecter des avertissements ;
— sauvegarder le rapport dans un fichier.

Pour aller plus loin – Exercice ludique (optionnel)

Une fois connecté à la machine distante via SSH, lancez le programme `neofetch`. Comparez les informations affichées avec celles de votre machine locale.