



Objectifs

- Réaliser l'itération 5 de l'application Gestion Log
 - Ouvrir et filtrer les informations concernant les connexions ssh
 - Écrire les informations dans une structure
 - Sauvegarder les informations dans un nouveau fichier de log

Contenu technique

- Manipulation des structures
- Écriture dans un fichier texte

Durée 6 heures

I. Intégration des logs SSH dans la structure SSHLoggin.

I.1 Git

N'oubliez pas de vous replacer sur la branche **dev** pour cette itération.

I.2 Analyse de la structure SSHLoggin

En suivant la conception du DDC,

- Q1. Lister les champs de la structure SSHLoggin
- Q2. Lister les champs de la structure SshDateTime
- Q3. Que contient le champ `_datetime` de la structure SSHLoggin ?

I.3 Enregistrer les connexions ssh clientes

Tâche1. Consulter la page comprendre et utiliser ssh

(<https://blogs.oracle.com/oracle-france/post/definition-ssh>) pour découvrir ce protocole indispensable .

I.3.1 Préparation

Les logs de connexion cliente ssh ne sont pas enregistrés par défaut sous Linux.

Votre RSSI vous demande de modifier la configuration de votre poste pour enregistrer les demandes de connexion ssh dans un fichier de log.



Voici la procédure qu'il vous a demandé de suivre.

Utiliser un script wrapper pour SSH

- Créer un script pour journaliser les connexions SSH

```
sudo nano /usr/local/bin/ssh-logger
```

- Ajouter le contenu suivant dans /usr/local/bin/ssh-logger

```
#!/bin/bash
echo "$(date) - $USER a exécuté : ssh $" | tee -a /var/log/ssh_client.log
/usr/bin/ssh "$@"
```

- Rendre le script exécutable

```
sudo chmod +x /usr/local/bin/ssh-logger
```

- Remplacer la commande ssh par un alias dans .bashrc

```
echo "alias ssh='/usr/local/bin/ssh-logger'" >> ~/.bashrc
```

- Appliquer immédiatement les modifications faites au fichier .bashrc

```
source ~/.bashrc
```

- Créer le fichier /var/log/ssh_client.log

```
sudo touch /var/log/ssh_client.log
```

- Modifier les droits et le groupe de /var/log/ssh_client.log

```
sudo chmod 644 /var/log/ssh_client.log
sudo chgrp adm /var/log/ssh_client.log
```

Remarque :

la commande `source` ne s'applique qu'au shell en cours d'exécution.

La configuration sera effective pour tous les shell au prochain redémarrage du poste

Tâche2. Vérifier que les demandes de connexions SSH sont bien enregistrées dans le fichier /var/log/ssh_client.log

Résultat attendu:

- ✓ Chaque demande de connexion ssh (ex: `ssh user1@192.168.1.73`) est enregistrée dans le fichier /var/log/ssh_client.log

I.4 Validation

Appeler l'enseignant pour qu'il valide votre travail



I.5 Fonction `sshLog()`

Les informations à extraire du fichier `/var/log/ssh_client.log` sont de la forme

```
mar. 25 mars 2025 13:23:02 CET - sepro a exécuté : ssh sepro@test2.com
```

Q4. En fonction de l'exemple ci-dessus, remplir sur votre compte rendu les champs des structures `SshLoggin` et `SshDateTime` avec les valeurs correspondantes.

Tâche3. Dans votre programme, déclarer la fonction `sshLog()`, ouvrir et afficher les entrées du fichier contenant les logs ssh.

Cette fonction est un peu plus complexe que la précédente, car il va falloir extraire les informations d'un flux pour remplir les champs de la structure.

Pour vous aider à « parser » le flux, vous aller vous aider des fonctions suivantes :

`gethostname()`, `getlogin_r()`, `std::find()`, `std::substr()`

Q5. Chercher et donner la définition des fonctions ci-dessus.

I.5.1 Structure `SshDateTime`

Tâche4. Remplir les champs de la structure `SshDateTime` avec la date (date, hour, minute, second) de la requête de connexion

Remarque :

Vous pouvez utiliser la fonction `stoi()` pour convertir une chaîne de caractère en entier.

I.5.2 Structure `SSHLogin`

La méthode ci-dessous vous aide à extraire les informations du flux en fonction des éléments à votre disposition.

Il vous faudra utiliser les informations données par `gethostname()`, `getlogin_r`, et `@` pour vous repérer dans le flux.

Tâche5. Remplir le champ `hostname`

Tâche6. Remplir le champ `username`



Tâche7. Isoler la partie du flux contenant les informations de connexion
(`test@srv.exemple.com`)

Tâche8. Isoler le nom de login et remplir le champ `sshUser`

Tâche9. Isoler l'adresse du serveur et remplir le champ `sshHost`

Tâche10. Considérer que `sshPortNumber` contient la valeur par défaut pour le port de connexion

I.5.3 Sauvegarde des informations

En suivant le DDC, réaliser la fonction `sauvegarderLogSSH()`

Tâche11. Sauvegarder les informations précédentes dans le fichier `ssh_connexion.txt` dans le répertoire de l'utilisateur.

Q16 – Fournir dans le compte rendu un screenshot du fichier `ssh_connexion.txt`

Résultat attendu:

- ✓ Les informations de connexion clientes SSH sont bien extraites
- ✓ Chacun des champs de la structure contient l'élément adéquat (ex `sshUser` le nom d'utilisateur)
- ✓ Les informations de connexion clientes sont sauvegardées dans le fichier `ssh_connexion.txt`

I.6 Validation

Appeler l'enseignant pour qu'il valide votre travail



II. Dossier de validation

Vérifier que votre application répond aux exigences du cahier des charges.

Q6. Compléter le dossier de vérification Gestion_Log_DDV.odt (ESS002 et ESS003)

III. Outils DevOps

Afin de finaliser votre travail, n'oubliez pas :

- D'indenter correctement votre code
- De documenter le code, les fonctions et les fichiers
- de fusionner votre branche dev avec master

IV. Livrable

Sur le Moodle de la section

- Votre compte rendu de TP avec les réponses aux questions de celui-ci
- Le dossier de validation DDV complété (ESS002 et ESS003)

Sur le serveur GIT de la section

- Le projet Gestion_Log à jour
- Les branches dev et master
- Le tag de la version v5.0