

Résolution problème(s)

Introduction

Ce document est un guide pour résoudre certains problèmes qui ont été découverts après publication finale du **Projet Seleenix**, en effet ceux-ci empêchent partiellement ou complètement le déploiement de la solution dans certains environnements. En espérant que ce document aidera à résoudre les problèmes que vous pouvez rencontrer.

Problèmes

Installation impossible

Cette section regroupe les problèmes rencontrés au niveau des installations empêchant le déploiement efficace de la solution sur le réseau du Client.

Script d'installation : clé AES et vecteur initialisation

Les deux scripts d'installations peuvent être lancés avec certains arguments que sont les suivants :

```
usage: install_serveurs.py [-h] [-k KEY] [-i IV] [-db DB]

options:
  -h, --help            show this help message and exit
  -k KEY, --key KEY      AES_KEY
  -i IV, --iv IV         AES_IV
  -db DB, --db DB       Database password
```

menu aide install_serveurs.py

Cela n'a pas été correctement spécifié dans les documentations d'installations mais les arguments KEY et IV respectent une logique propre.

- **KEY doit avoir une longueur de 32 caractères**
- **IV doit avoir une longueur de 16 caractères**
- **Les caractères spéciaux " et ' sont interdits**

Si ces règles ne sont pas respectées on obtient des exceptions lors de l'exécution des scripts tels que:

```

vous pouvez maintenant appeler votre script sans le préfixe python
ex : s_server -h (windows) | server -h (mac)

C:\Users\...\Desktop\SAE Corrections FI\...SAE302>s_server
Traceback (most recent call last):
  File "C:\Users\...\AppData\Roaming\seleenix_venv\Scripts\server.py", line 249, in <module>
    cipher = AES.new(key, AES.MODE_CBC, iv)
              ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\...\AppData\Roaming\seleenix_venv\Lib\site-packages\Crypto\Cipher\AES.py", line 228, in new
    return _create_cipher(sys.modules[__name__], key, mode, *args, **kwargs)
              ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\...\AppData\Roaming\seleenix_venv\Lib\site-packages\Crypto\Cipher\__init__.py", line 79, in _create_c
    ipher
    return modes[mode](factory, **kwargs)
              ^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\...\AppData\Roaming\seleenix_venv\Lib\site-packages\Crypto\Cipher\_mode_cbc.py", line 274, in _create
    _cbc_cipher
    cipher_state = factory._create_base_cipher(kwargs)
                    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\...\AppData\Roaming\seleenix_venv\Lib\site-packages\Crypto\Cipher\AES.py", line 90, in _create_base_c
    ipher
    raise ValueError("Incorrect AES key length (%d bytes)" % len(key))
ValueError: Incorrect AES key length (34 bytes)

C:\Users\...\Desktop\SAE Corrections FI\...SAE302>

```

Concernant les caractères spéciaux c'est uniquement parce **que il faut entourer les valeurs des arguments par des guillemets** , et ces caractères pourraient scinder une des deux variables.

ex d'un bon formatage:

```

py install_serveurs.py -db toto -k "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA" -i "XXXXXXXXXXXXXXXXXXXX"

```

Script d'installation : Variables environnements

A l'origine les variables d'environnements étaient stockées sur les machines où les scripts tournaient, cependant après réflexion il a été jugé non éthique de faire ce genre d'insertion au niveau du système directement, puisque cela peut engendrer des soucis de sécurité ou même la corruption d'applications tierces, dans la même idée que l'environnement virtuel d'exécution pour Python, il a été mis en place un fichier configuration contenant ces variables d'environnements.

Seulement, ce changement a été effectué à la fin donc il y a une erreur qui a été faite au moment où il y a déplacement du fichier d'environnement à l'emplacement des scripts:

```

os.system(f"move .\env {venvPath}\\Scripts\\ /y") if os.name == "nt" \
    else os.system(f"mv .env {venvPath}/bin/")

```

La syntaxe utilisée pour les systèmes Windows est celle de la commande xcopy afin de régler le problème, il suffit de remplacer move par xcopy dans les deux scripts d'installations et d'enlever "." qui précède le nom du fichier.

```

os.system(f"xcopy .env {venvPath}\\Scripts\\ /y") if os.name == "nt" \
    else os.system(f"mv .env {venvPath}/bin/")

```

Installation pré-requis (MySQL Server)

Si vous êtes dans un environnement Linux de la distribution Debian les étapes d'installations de **MySQL Server** des Documents d'installation sont incorrectes.

Voici ce que vous devrez faire pour que cela fonctionne :

```
sudo apt update -y
sudo apt upgrade -y # si besoin uniquement

sudo apt install mariadb-server -y
```

Pensez bien à **mettre un mot de passe pour le compte root**:

```
mysql -u root

ALTER USER 'root'@'localhost' IDENTIFIED BY 'new_password';
FLUSH PRIVILEGES;
```

Importation Base de données (MySQL)

Puisque la base de données a été entièrement établie sur Windows l'encodage utilisée pour les données est incompatible avec celui de MariaDB (linux), donc il faut opérer une commande sur le fichier avant d'essayer l'importation :

```
sed -i "s/0900_ai/unicode/g" [emplacement_fichier_sql]
```

Cette commande permet de remplacer toutes les occurrences mentionnant "0900_ai" par "unicode" permettant de rendre possible l'importation.

Conclusion

Ce document de résolution de problèmes vise à aider à surmonter les difficultés rencontrées lors du déploiement du projet Seleenix. Nous avons abordé plusieurs problèmes, notamment ceux liés à l'installation, aux variables d'environnement et à l'importation de la base de données. En suivant les solutions proposées, vous devriez être en mesure de résoudre ces problèmes et de déployer efficacement la solution dans votre environnement.

N'hésitez pas à consulter les sections appropriées en fonction du problème que vous rencontrez. Si vous avez besoin d'une assistance supplémentaire, n'hésitez pas à me contacter.

Nous espérons que ce guide vous a été utile et vous souhaite beaucoup de succès dans la résolution de vos problèmes.