Promotion 2023

Année A4

Souhail AIT LAHCEN

Report Crypto Project
Année 4

12/2021 - Document confidentiel

Table des matières

Report Crypto Project 2021-2022

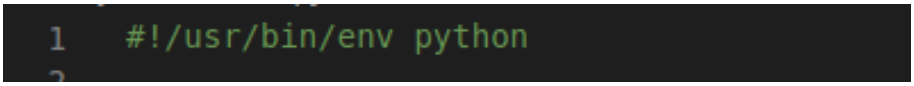
Souhail AIT LAHCEN

31 Décembre 2021

I Introduction

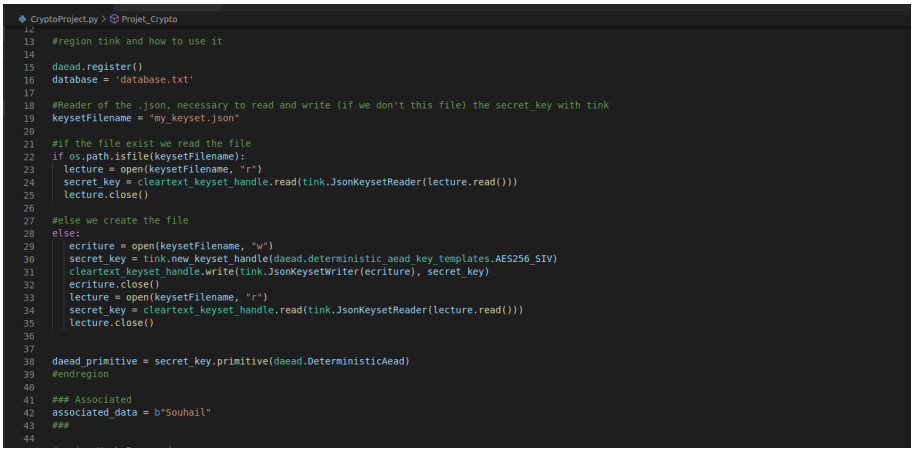
For this project, I need to create a script to allows the user to login or register with a password, username etc.. by using **Tink** and **AES-SIV**.
We can fin my script, report etc.. in my repository GitHub :
`https://github.com/Souhail99/Project-Cryptography`.

Also, **as you can see in my README.md on my repository GitHub**, you don't need (in your terminal) to write **/usr/bin/python** before the name of my code but you need to put the all path , because I use this :



II Tink

First, we need to create or read the file that generate to have the all information about the secret key :



And with this, we can use the encryption correctly.

III Hash Password

So, we need also to hash the password, for this we need to choose a properly and useful mode of hashing, so for this project, I have chosen **bcrypt**.

```
44
45  #region Hash Password
46  def hash_password(pwd:str,salt=None):
47      if salt==None:
48          salt=bcrypt.gensalt()
49      else:
50          salt=salt
51      hash=bcrypt.hashpw(pwd.encode(),salt)
52      return hash,salt
53  #endregion
54
55
```

IV Encryption

For the encryption, we use **AES-SIV**, thanks to **Tink**, we can use this with a single line of code :

```
55
56  #region encryption
57  def encryption_machine(msg:bytes):
58      # encrypt using AES-SIV
59      ciphertext=daead_primitive.encrypt_deterministically(msg, associated_data)
60      return ciphertext
61  #endregion
62
```

V Register and Login

The user must be able to register or login. So, we need to create two functions.
The first is register **(inscription in my script)**, so this function ask the user to put a username and a password and then save them in the database.
The first is login **(connexion in my script)**, so this function will verify if in our database we have this user.

```python
#region register
def inscription():
    user=input("Your username please > ")
    pwd=input("Your password please > ")
    print("\n")
    save_to_database(user,pwd)
    print("\n")
    print("Register finish")
#endregion

#region login
def connexion():
    user=input("Your username please > ")
    pwd=input("Your password please > ")
    test,error=check_password(user, pwd)
    if test==True and error==0:
        print("Good password and good user, welcome :",user)
    if test==False and error==1:
        print("Error, this isn't the good user or pwd")
#endregion
```

VI Save To Database

After a user register its usersame and password, we need to be able to write this information in our database **(here the file is 'database.txt')** (or create the file if we don't have this database the firstime) :

```python
#region save to database
def save_to_database(user, pwd):
  # use a file as a database
  # format: user, hashed_password
  # for example: file.write(user, hash_password(pwd))
  hash,salt=hash_password(pwd)
  pwdisencrypted=encryption_machine(hash)
  pswrd = open(database, "a")
  pswrd.write(f'{user},{pwdisencrypted.hex()},{salt.hex()}\n')
  pswrd.close()

#endregion
```

VII Check Password

I think this is the most important part of my script.
We need to be able to check in our database, after a user want to be login, if this user is in our database.
We check in every line in our database if the username exist and also if for the same line the password is the same.
After this we return the answer :

```
#region check
def check_password(user, pwd):
  # read from database
  with open(database, 'r') as f:
    error=1
    test=False
    print("Welcome user :",user," we will verify if you are in our database...")
    for line in f.readlines():
      userdatabase,encrypteddatabase,salt= line.split(',')
      hash,salt2=hash_password(pwd,bytes.fromhex(salt))
      password_of_the_current_user=encryption_machine(hash)
      # and check for authentication
      encrypteddatabase=bytes.fromhex(encrypteddatabase)
      if (user == userdatabase) and (encrypteddatabase == password_of_the_current_user):
        test=True
        error=0
        return test,error
  return test,error
#endregion
```

VIII Project Crypto

Finally, I create a menu to be able of register, login and exit.
If the file database doesn't exist, I ask the user to register in firstime, to create this database.
Then, I create a region **main**, to launch the function.

```
#endregion

#region Project
def Projet_Crypto():
    c=True
    print("Welcome to my first Crypto-Project ! ")
    while c:
        print("\n")
        print("-------------start-------------")
        print("\n")
        case=input("Do you want to register (Press 1) or login (Press 2) or exit (Press 3) > ")
        print("\n")
        if case == "1":
            inscription()
        elif case =="2":
            if os.path.isfile(database):
              connexion()
            else:
              print("Database isn't yet created, you need to register to initialize the database for the firstime")
        elif case =="3":
            c=False
            print("Au revoir !")
        else:
            print("Something, you don't press an avalaible numbers, please restart")

        print("-------------end--------------")
#endregion

#region main
if __name__ == '__main__':
    Projet_Crypto()
```