Faculté des Sciences
Université d'Etat D'Haïti

Université d'État d'Haïti
ueh

FDS-UEH

D3v0!r d3 S3cur!t3

Préparé par:
ALTIDOR Jean-Bernard T.
DUBUCHE Kevin J.
THEODORE Barbara G.
[
Section: Génie Electronique
Année : 3ième
Professeur: M. Vladimyr MATHIEU
]

Décembre 2020

Table des matières

Devoir de Sécurité
Vigenère Cypher

Enoncé :
I-Implémenter en python une fonction permettant d'encrypter un texte avec le chiffrement de Vigenère.

II- Implémenter en python une fonction permettant de décrypter un texte codé avec le chiffrement de Vigenère en connaissant la clé.

III- Implémenter en python une fonction permettant de décrypter un texte codé avec le chiffrement de Vigenère sans avoir la clé au préalable.

IV- Implémenter un système peep to peer utilisant le chiffrement de Vigenère pour communiquer.

Introduction

Le chiffre de Vigenère est un système de chiffrement par substitution polyalphabétique mais une même lettre du message clair peut, suivant sa position dans celui-ci, être remplacée par des lettres différentes, contrairement à un système de chiffrement mono alphabétique comme le chiffre de César.

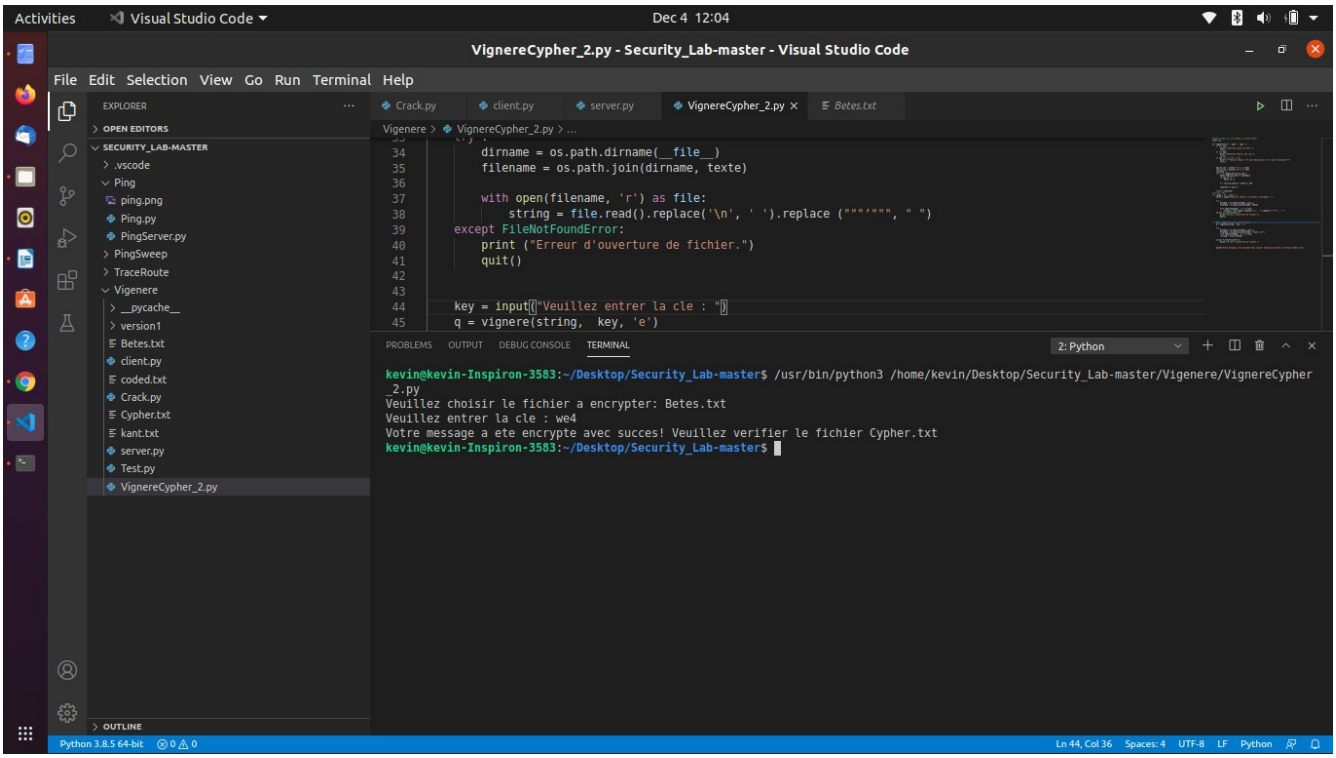
I-Implémenter en python une fonction permettant d'encrypter un texte avec le chiffrement de Vigenère.

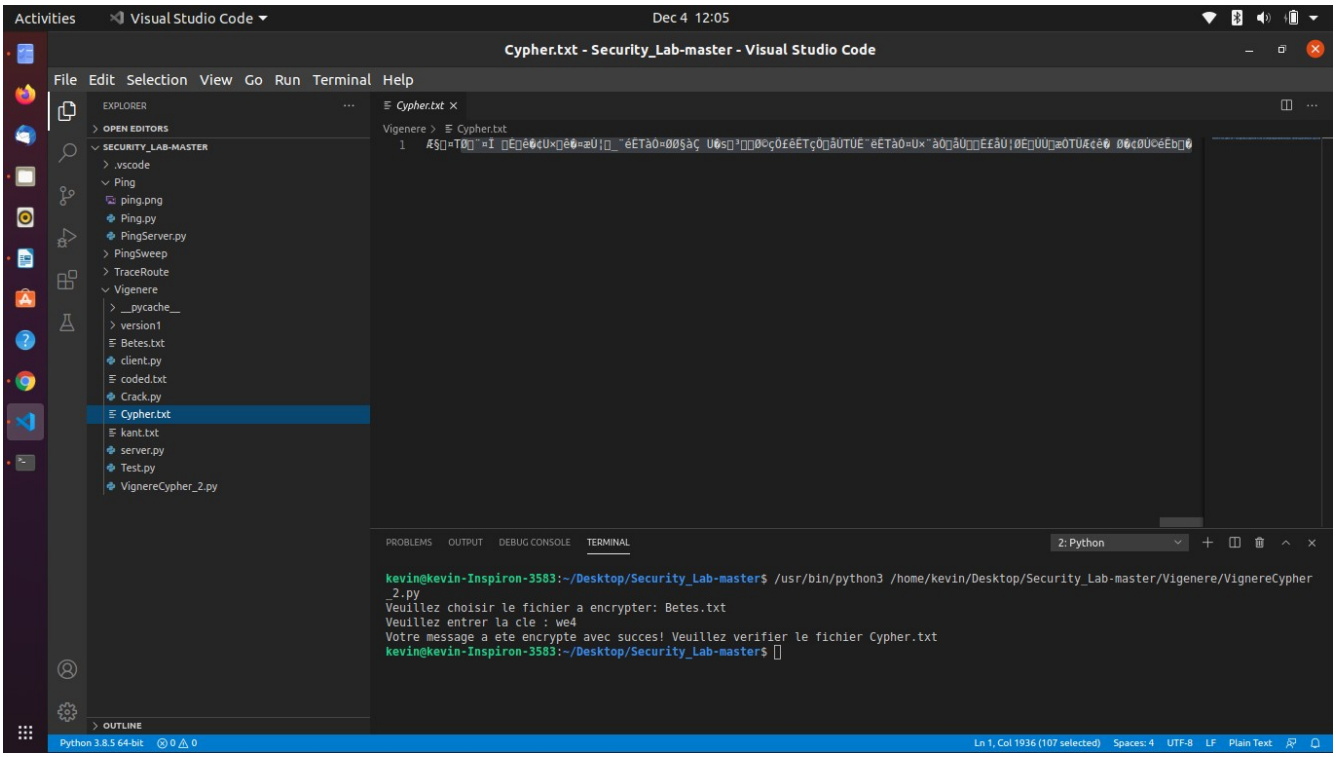
C'est une fontion qui prend trois paramètres :
1- le texte à coder
2- la clé de chiffrement
3- un caractère : 'e' pour encryption

Il s'agit d'ajouter le décalage correspondant au caractère se trouvant à la position i de la clé (se répétant de manière cyclique).

```
#Vignere code 2.0 ,can handle all ASCII symbol
import os

def vignere(txt='', key='', typ=''):
if not txt:
print ("Veuillez entrer un text.")
return
if not key:
print ("Veuillez fournir une cle.")
return
if typ not in ('d', 'e'):
print ("""Veullez choisir "d" pour decryption ou "e" pour encryption""")
return

key_to_int = [ord(i) for i in key]
txt_to_int = [ord(i) for i in txt]
resultat = ''
for i in range(len(txt_to_int)):
shift =key_to_int[i % len(key)]
if typ == 'd':
shift *= -1

v = (txt_to_int[i] + shift) % 240

resultat += chr(v )

return (resultat)
# Driver code
```

```
if __name__ == "__main__":
texte = input("Veuillez choisir le fichier a encrypter: ")
try :
dirname = os.path.dirname(__file__)
filename = os.path.join(dirname, texte)
with open(filename, 'r') as file:
string = file.read().replace('\n', ' ').replace ("""", " ")
except FileNotFoundError:
print ("Erreur d'ouverture de fichier.")
quit()

key = input("Veuillez entrer la cle : ")
q = vignere(string, key, 'e')
try :
dirname = os.path.dirname(__file__)
filename = os.path.join(dirname, "Cypher.txt")
with open(filename, 'w') as file:
string = file.write(q)
except FileNotFoundError:
print ("Erreur d'ouverture de fichier.")
print("Votre message a ete encrypte avec succes! Veuillez verifier le fichier Cypher.txt")
```





II- Implémenter en python une fonction permettant de décrypter un texte codé avec le chiffrement de Vigenère en connaissant la clé.

C'est une fonction qui prend trois paramètres :
1- le texte encodé
2- la clé de chiffrement
3- un caractère : 'd' pour décrypter

```
#Vignere code 2.0 ,can handle all ASCII symbol
import os

def vignere(txt='', key='', typ=''):
if not txt:
print ("Veuillez entrer un text.")
return
if not key:
print ("Veuillez fournir une cle.")
return
if typ not in ('d', 'e'):
print ("""Veullez choisir "d" pour decryption ou "e" pour encryption""")
```

```
return

key_to_int = [ord(i) for i in key]
txt_to_int = [ord(i) for i in txt]
resultat = ''
for i in range(len(txt_to_int)):
shift =key_to_int[i % len(key)]
if typ == 'd':
shift *= -1

v = (txt_to_int[i] + shift) % 240

resultat += chr(v )

return (resultat)
# Driver code
if __name__ == "__main__":
texte = input("Veuillez choisir le fichier a encrypter: ")
try :
dirname = os.path.dirname(__file__)
filename = os.path.join(dirname, texte)
with open(filename, 'r') as file:
string = file.read().replace('\n', ' ').replace ("""’""", " ")
except FileNotFoundError:
print ("Erreur d'ouverture de fichier.")
quit()

key = input("Veuillez entrer la cle : ")
q = vignere(string, key, 'e')
try :
dirname = os.path.dirname(__file__)
filename = os.path.join(dirname, "Cypher.txt")
with open(filename, 'w') as file:
string = file.write(q)
except FileNotFoundError:
print ("Erreur d'ouverture de fichier.")
print("Votre message a ete encrypte avec succes! Veuillez verifier le fichier Cypher.txt")
```

III- Implémenter en python une fonction permettant de décrypter un texte codé avec le chiffrement de Vigenère sans avoir la clé au préalable.

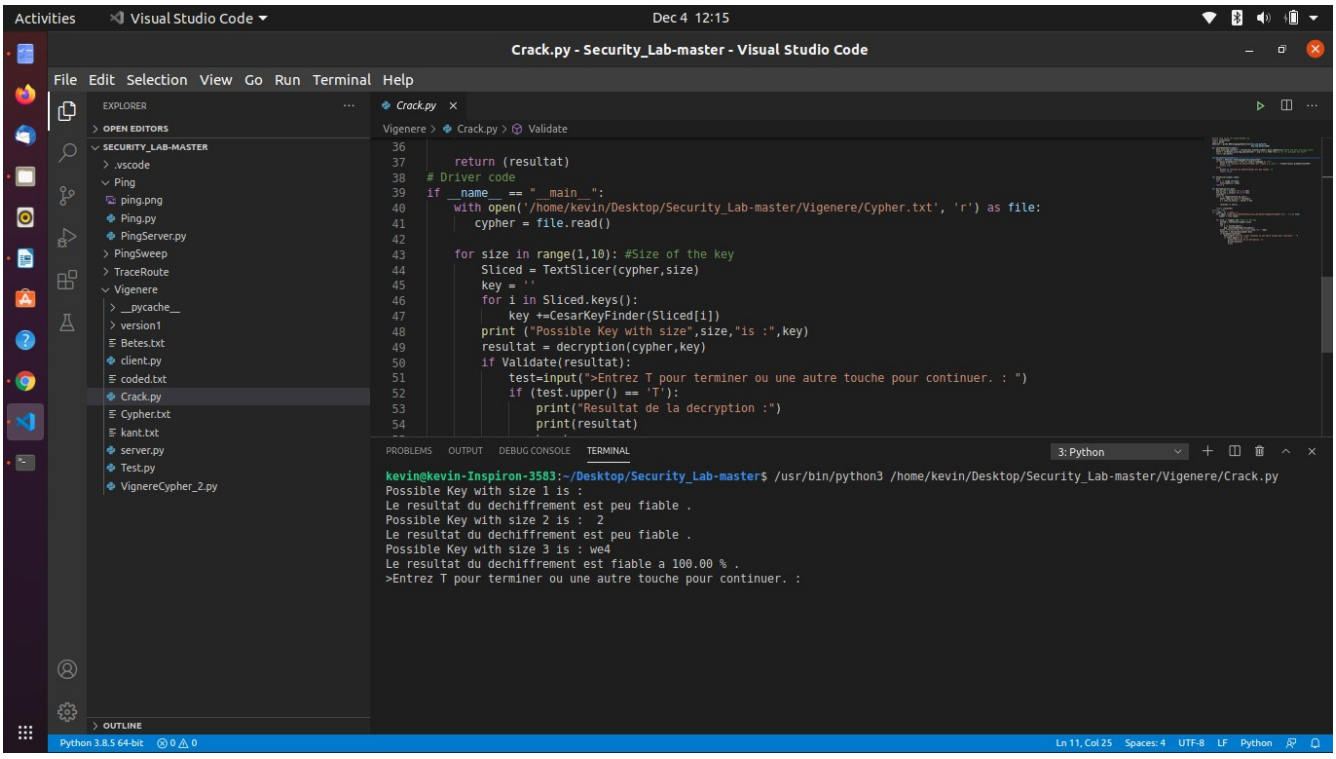
Pour décrypter le message sans la clé, on se base sur la probabilité d'occurrence de la lettre "e" en français. On commence par découper le texte encrypté en blocs, avec la fonction *TextSlicer*, et analyser

quel élément à la i ème position de chacun des blocs apparait le plus. "e" étant le caractère le plus fréquent, après "espace", on considère alors que le deuxième plus fréquent est probablement la lettre "e" et on calcule son écart par rapport à cette dernière pour déterminer la clé. Ceci se fait dans la fonction "*CesarKeyFinder*", car on peut considérer les éléments à la même position des différents blocs comme encryptés par la méthode de César puisqu'ils ont subi un déplacement par la même clé. Il ne nous reste donc qu'à itérer sur la longueur de clé choisie et vérifier à chaque fois si le texte décrypté, avec la fonction "*decryption*" , est effectivement du français, avec la fonction *Validate*.

Il a lieu de mentionner que cette méthode ne fonctionne que s'il y a assez de "e" dans le texte considéré.

```
#Crack that works for VignereCypher.py
import collections
import gcld3
detector = gcld3.NNetLanguageIdentifier(min_num_bytes=0,
max_num_bytes=1000)
def CesarKeyFinder(cypher):
most_occuring_charater = collections.Counter(cypher).most_common()[1] #Find the most
occuring letter
shift = (ord(most_occuring_charater[0]) - ord ('e')) %240 #Maps it to "e" and gets the shift
return chr(shift)

def Validate(plaintext):
result = detector.FindLanguage(text=plaintext)
if result.probability > 0.50 and result.language == 'fr':
print ("Le resultat du dechiffrement est fiable a {:.2f} % .".format(result.probability*100))
return True
else:
print("Le resultat du dechiffrement est peu fiable .")
return False

def TextSlicer(cypher,step):
C={}
for i in range (0,step):
C[i]=cypher[i::step]
return C

def decryption(txt,key):
key_to_int = [ord(i) for i in key]
txt_to_int = [ord(i) for i in txt]
resultat = ''
for i in range(len(txt_to_int)):
shift =key_to_int[i % len(key)]
v = (txt_to_int[i] - shift) % 255
```

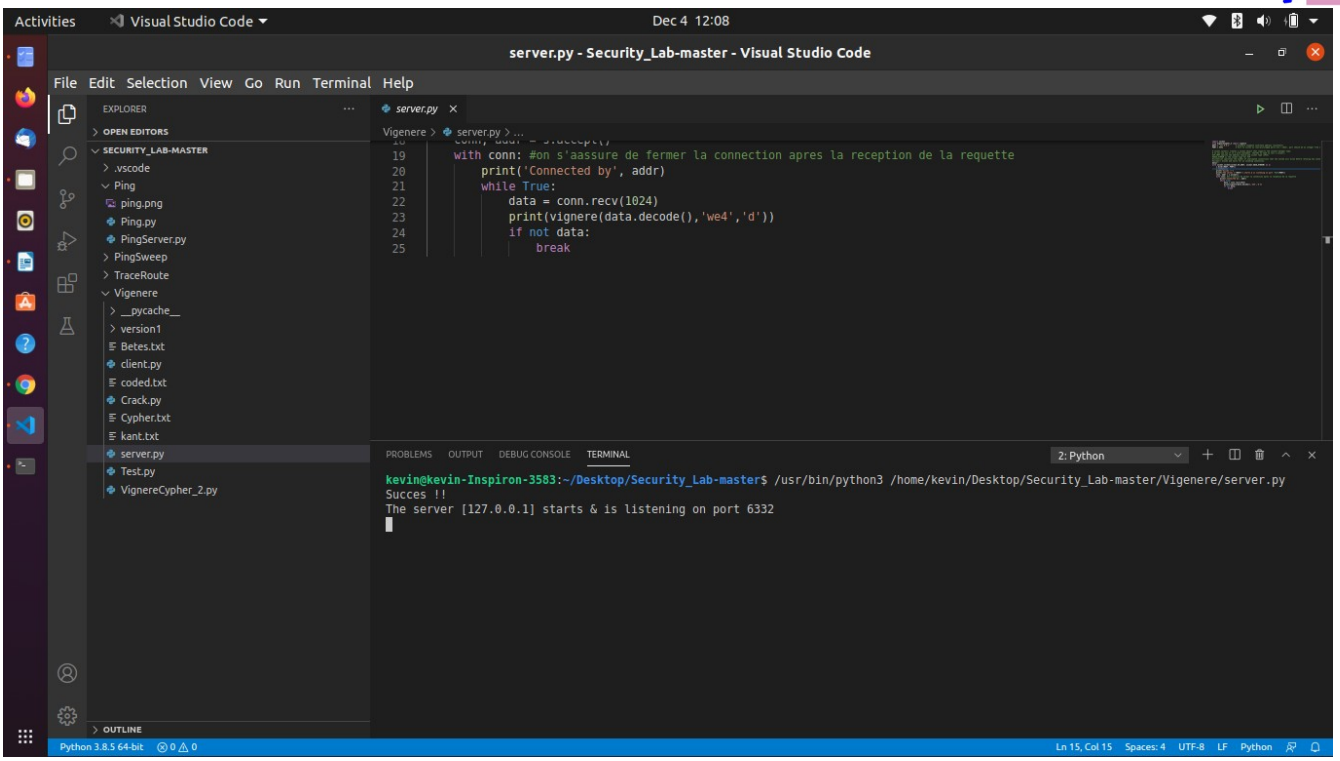
```
resultat += chr(v)

return (resultat)
# Driver code
if __name__ == "__main__":
with open('/home/kevin/Desktop/Security_Lab-master/Vigenere/Cypher.txt', 'r') as file:
cypher = file.read()
for size in range(1,10): #Size of the key
Sliced = TextSlicer(cypher,size)
key = ''
for i in Sliced.keys():
key +=CesarKeyFinder(Sliced[i])
print ("Possible Key with size",size,"is :",key)
resultat = decryption(cypher,key)
if Validate(resultat):
test=input(">Entrez T pour terminer ou une autre touche pour continuer. : ")
if (test.upper() == 'T'):
print("Resultat de la decryption :")
print(resultat)
break
```

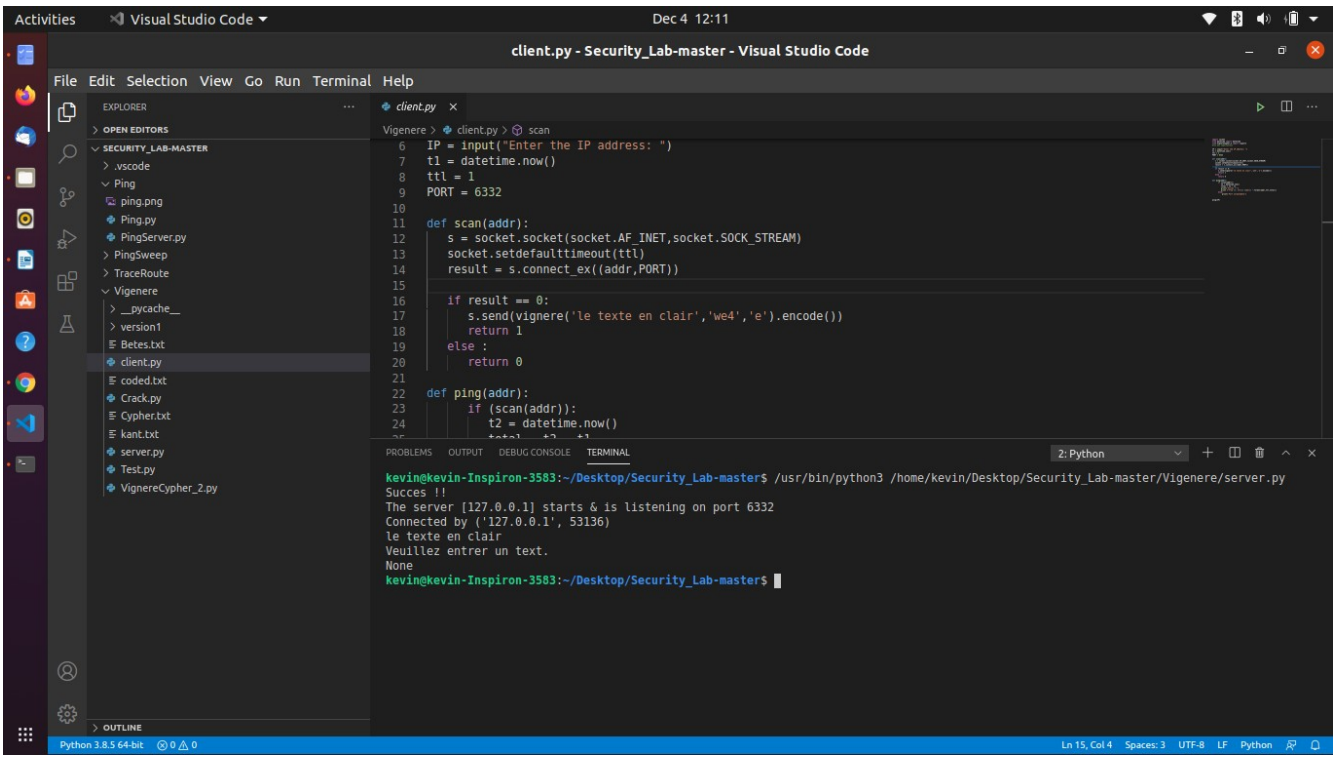


IV- Implémenter un système peer to peer utilisant le chiffremment de Vigenère pour communiquer.

Server qui attend la connection du client pour déchiffrer le message reçu.

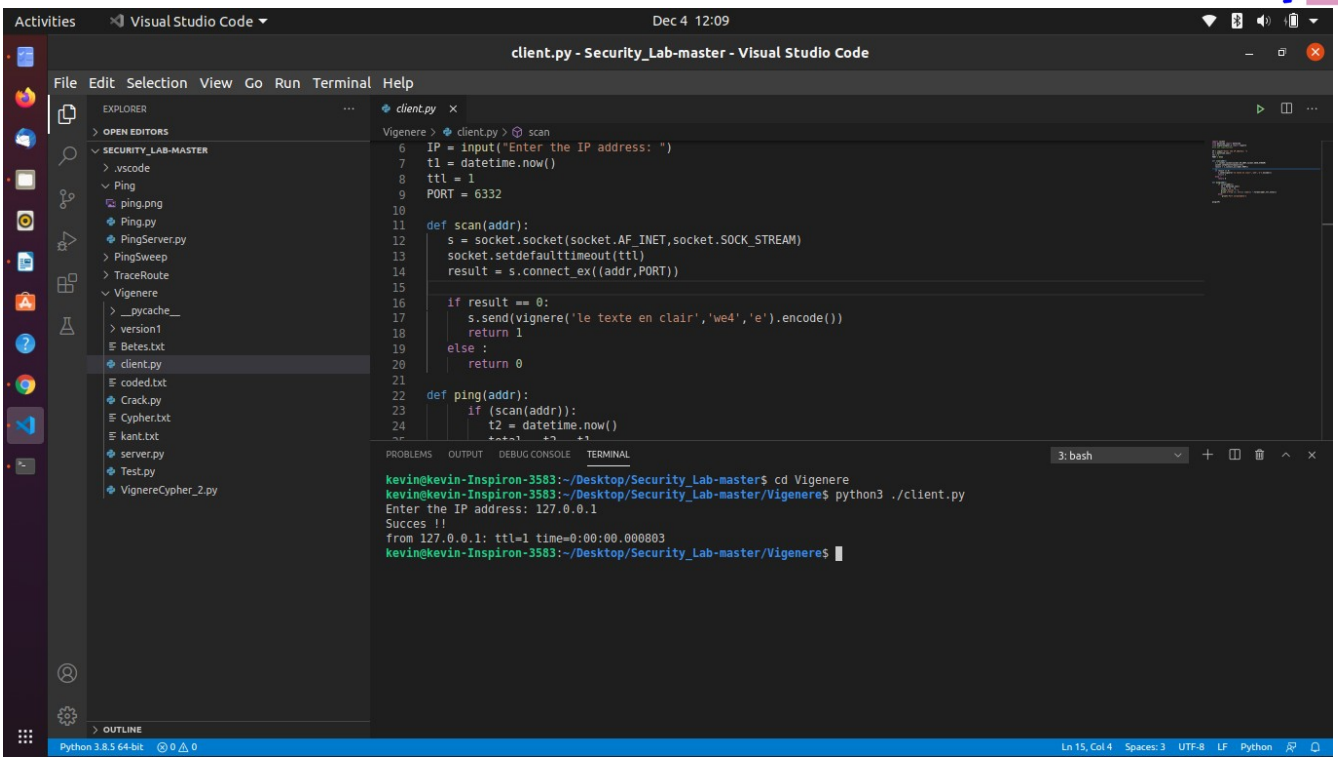
```
import socket
from VignereCypher_2 import vignere
HOST = '127.0.0.1' # Standard loopback interface address (localhost)
PORT = 6332 # Port to listen on (non-privileged ports are > 1023). port should be an integer
from 1-65535

# socket.socket() creates a socket object that supports the context manager type,
# so you can use it in a with statement. There's no need to call s.close():
#AF_INET,specify the address family and socket type (IPv4)
#SOCK_STREAM is the socket type for TCP
#listen()It specifies the number of unaccepted connections that the system will allow before
refusing new connections.
#accept() blocks and waits for an incoming connection.
data=''
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
s.bind((HOST, PORT))
s.listen()
print('Succes !!')
print('The server ['+HOST+'] starts & is listening on port '+str(PORT))
conn, addr = s.accept()
with conn: #on s'aassure de fermer la connection apres la reception de la requette
print('Connected by', addr)
while True:
data = conn.recv(1024)
print(vignere(data.decode(),'we4','d'))
if not data:
break
```

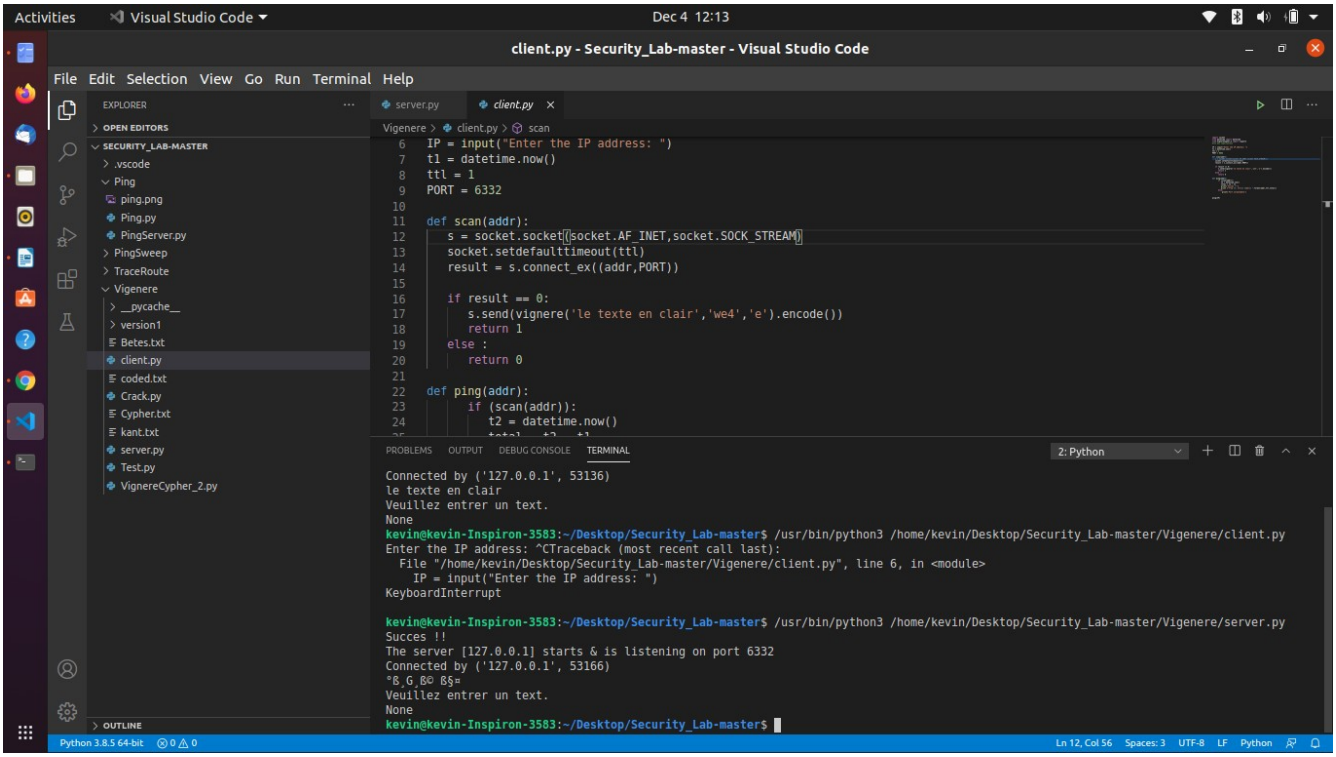

Activities
Visual Studio Code
Dec 4 12:08
server.py - Security_Lab-master - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
SECURITY_LAB-MASTER
.vscode
Ping
ping.png
Ping.py
PingServer.py
PingSweep
TraceRoute
Vigenere
__pycache__
version1
Betes.txt
client.py
coded.txt
Crack.py
Cypher.txt
kant.txt
server.py
Test.py
VignereCypher_2.py
OUTLINE
server.py
Vigenere > server.py > ...
with conn: #on s'aassure de fermer la connection apres la reception de la requette
print('Connected by', addr)
while True:
data = conn.recv(1024)
print(vignere(data.decode(),'we4','d'))
if not data:
break
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
2: Python
kevin@kevin-Inspiron-3583:~/Desktop/Security_Lab-master$ /usr/bin/python3 /home/kevin/Desktop/Security_Lab-master/Vigenere/server.py
Succes !!
The server [127.0.0.1] starts & is listening on port 6332
Python 3.8.5 64-bit
Ln 15, Col 15 Spaces: 4 UTF-8 LF Python


Activities
Visual Studio Code
Dec 4 12:11
client.py - Security_Lab-master - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
SECURITY_LAB-MASTER
.vscode
Ping
ping.png
Ping.py
PingServer.py
PingSweep
TraceRoute
Vigenere
__pycache__
version1
Betes.txt
client.py
coded.txt
Crack.py
Cypher.txt
kant.txt
server.py
Test.py
VignereCypher_2.py
OUTLINE
client.py
Vigenere > client.py > scan
IP = input("Enter the IP address: ")
t1 = datetime.now()
ttl = 1
PORT = 6332
def scan(addr):
s = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
socket.setdefaulttimeout(ttl)
result = s.connect_ex((addr,PORT))
if result == 0:
s.send(vignere('le texte en clair','we4','e').encode())
return 1
else :
return 0
def ping(addr):
if (scan(addr)):
t2 = datetime.now()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
2: Python
kevin@kevin-Inspiron-3583:~/Desktop/Security_Lab-master$ /usr/bin/python3 /home/kevin/Desktop/Security_Lab-master/Vigenere/server.py
Succes !!
The server [127.0.0.1] starts & is listening on port 6332
Connected by ('127.0.0.1', 53136)
le texte en clair
Veuillez entrer un text.
None
kevin@kevin-Inspiron-3583:~/Desktop/Security_Lab-master$
Python 3.8.5 64-bit
Ln 15, Col 4 Spaces: 3 UTF-8 LF Python

Client qui se connecte au server pour envoyer un message codé.

```
import socket
from datetime import datetime
from VignereCypher_2 import vignere
#fin des importations

IP = input("Enter the IP address: ")
t1 = datetime.now()
ttl = 1
PORT = 6332

def scan(addr):
s = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
socket.setdefaulttimeout(ttl)
result = s.connect_ex((addr,PORT))
if result == 0:
s.send(vignere('le texte en clair','we4','e').encode())
return 1
else :
return 0

def ping(addr):
if (scan(addr)):
t2 = datetime.now()
total = t2 - t1
print('Succes !!')
print ("from {}: ttl={} time={} ".format(addr,ttl,total))
else:
print('Port unreachable')

ping(IP)
```


Activities
Visual Studio Code
Dec 4 12:09
client.py - Security_Lab-master - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
SECURITY_LAB-MASTER
.vscode
Ping
ping.png
Ping.py
PingServer.py
PingSweep
TraceRoute
Vigenere
__pycache__
version1
Betes.txt
client.py
coded.txt
Crack.py
Cypher.txt
kant.txt
server.py
Test.py
VignereCypher_2.py
OUTLINE
client.py
Vigenere > client.py > scan
IP = input("Enter the IP address: ")
tl = datetime.now()
ttl = 1
PORT = 6332
def scan(addr):
s = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
socket.setdefaulttimeout(ttl)
result = s.connect_ex((addr,PORT))
if result == 0:
s.send(vignere('le texte en clair','we4','e').encode())
return 1
else :
return 0
def ping(addr):
if (scan(addr)):
t2 = datetime.now()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
3: bash
kevin@kevin-Inspiron-3583:~/Desktop/Security_Lab-master$ cd Vigenere
kevin@kevin-Inspiron-3583:~/Desktop/Security_Lab-master/Vigenere$ python3 ./client.py
Enter the IP address: 127.0.0.1
Succes !!
from 127.0.0.1: ttl=1 time=0:00:00.000803
kevin@kevin-Inspiron-3583:~/Desktop/Security_Lab-master/Vigenere$
Python 3.8.5 64-bit
Ln 15, Col 4 Spaces: 3 UTF-8 LF Python


Activities
Visual Studio Code
Dec 4 12:13
client.py - Security_Lab-master - Visual Studio Code
File Edit Selection View Go Run Terminal Help
server.py
client.py
Vigenere > client.py > scan
IP = input("Enter the IP address: ")
tl = datetime.now()
ttl = 1
PORT = 6332
def scan(addr):
s = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
socket.setdefaulttimeout(ttl)
result = s.connect_ex((addr,PORT))
if result == 0:
s.send(vignere('le texte en clair','we4','e').encode())
return 1
else :
return 0
def ping(addr):
if (scan(addr)):
t2 = datetime.now()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
2: Python
Connected by ('127.0.0.1', 53136)
le texte en clair
Veuillez entrer un text.
None
kevin@kevin-Inspiron-3583:~/Desktop/Security_Lab-master$ /usr/bin/python3 /home/kevin/Desktop/Security_Lab-master/Vigenere/client.py
Enter the IP address: ^CTraceback (most recent call last):
File "/home/kevin/Desktop/Security_Lab-master/Vigenere/client.py", line 6, in <module>
IP = input("Enter the IP address: ")
KeyboardInterrupt
kevin@kevin-Inspiron-3583:~/Desktop/Security_Lab-master$ /usr/bin/python3 /home/kevin/Desktop/Security_Lab-master/Vigenere/server.py
Succes !!
The server [127.0.0.1] starts & is listening on port 6332
Connected by ('127.0.0.1', 53166)
°ß¸G¸ß© ß§=
Veuillez entrer un text.
None
kevin@kevin-Inspiron-3583:~/Desktop/Security_Lab-master$
Python 3.8.5 64-bit
Ln 12, Col 56 Spaces: 3 UTF-8 LF Python