

PYTHON_RANSOMWARE POC :

WHY PYTHON ?

Python programming language is a high level language that is preferred for most developer in 2022, research has been proven that this language can be leveraged for malicious intent such as making a malware to disrupt the network of organisations , such kind of malware are spyware like PWOBOT and ransomware such as the PYLOCKY case , even as RAT (such as POETRAT) .

PYTHON RANSOMWARE :

One of the reasons that leads to this study is that regardless of the danger it can represent , this type of attack is considered unlikely to happen to an organisation which pushes the attacker to use this as an attack vector to deploy their malicious code . As a result , money is extorted effortlessly just due to lack of awareness and false belief that if the antivirus does not flag it as malicious ,then it is not malicious.

Nevertheless , It is never too late to learn about it , and here is a demonstration on how this kind of malware can effectively access and encrypt files without alarming the antiviruses. So Please take a note .

HOW IT WORKS :

We assume that the Python_Ransomware is already downloaded by the user , either by phishing attack or by going to an infected website , or via USB .

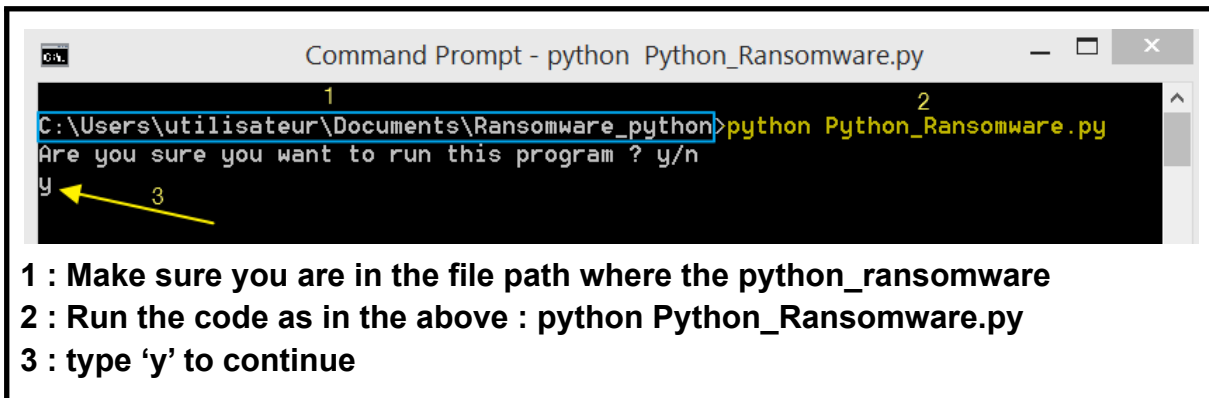
ATTENTION !!! BEFORE RUNNING THE PROGRAM !!! FOLLOW THE FOLLOWING

- Perform the test on windows 8.1 and higher . The experiment has not been performed on any Unix flavoured OS , nor macbook .
- Make sure python preferably python 3.7.x and above is installed on the victim machine or the machine where to perform the task.
- On the windows desktop , create a folder called **test** , for testing purpose add **.txt** , **.docx**, **.jpg** files within the test folder .
- Open each file and read their content , I believe everything will work fine and normal, for now.

Step 1 :

As this is for study purposes , we do not want to create havoc in our own system , a safeguard has been implemented while running the program if yes or no you want to run it .

After all the settings are complete , you can safely type “y” and enter.

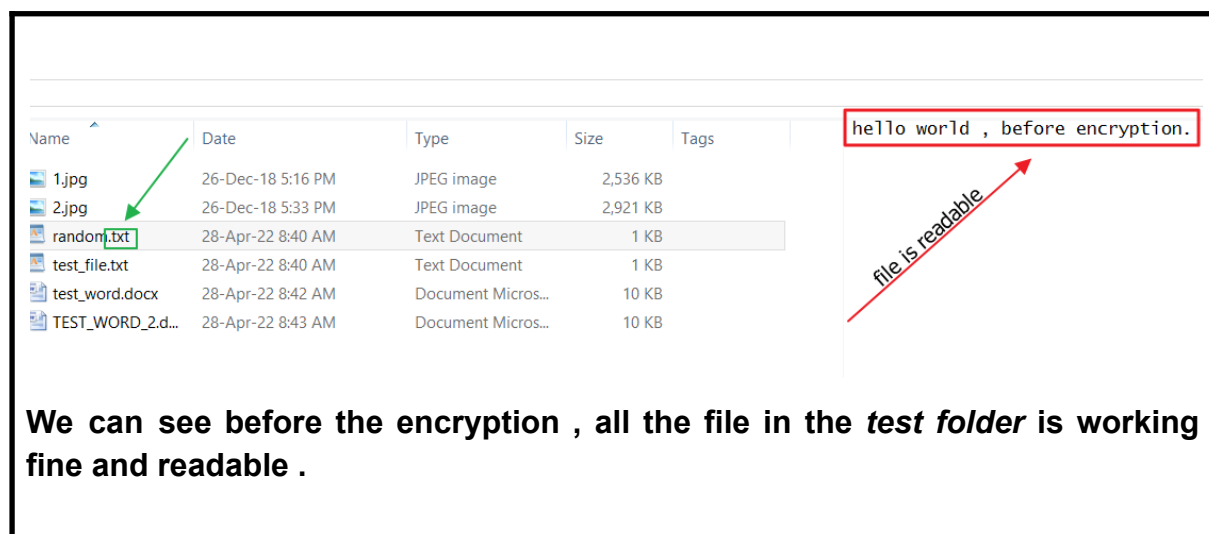


Step 2:

The program will select all the files present in the path mentioned using the following line of code

```
for root, dirs, files in  
os.walk("C:\\Users\\"+pc_username+"\\Desktop\\test")
```

Warning : The path should be the test folder path that you would want to run the code .This is crucial as you may encrypt all the files from the desktop having the particular extension.So be mindful of the file path



Step 3 :

In the next lines of code , the program will extracted the extension of all selected files and consider only the files having extension mentioned in

```
Extension to be encrypted = ( '.txt', '.docx' , '.jpg' )
```

In our case , it will select text file , word documents , and jpg files .

The program will select those particular files and put group them in the list `file_lists_paths = []` waiting for the encryption process .

Step 4 :

This is where the program will generate a random key based on all existing characters , ie, Upper case , Lower case alphabet , digits and special character .

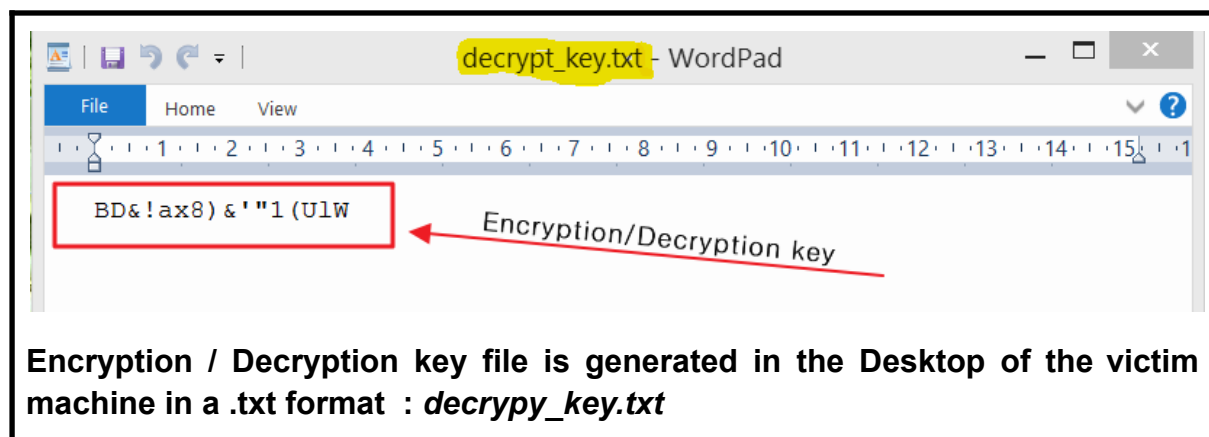
The key will be stored in `key = ""` , this key will be used to encrypt the files later .

Each time the program is run , this key will change .

This key will be stored in a text file on the Desktop for later decryption as this is for testing purposes only .

```
with open(desktop_path + extend + decrypt_key , "w", encoding="utf-8") as f:  
    f.write(key)
```

Where `decrypt_key = "decrypt_key.txt"` #created on desktop after encryption is successful

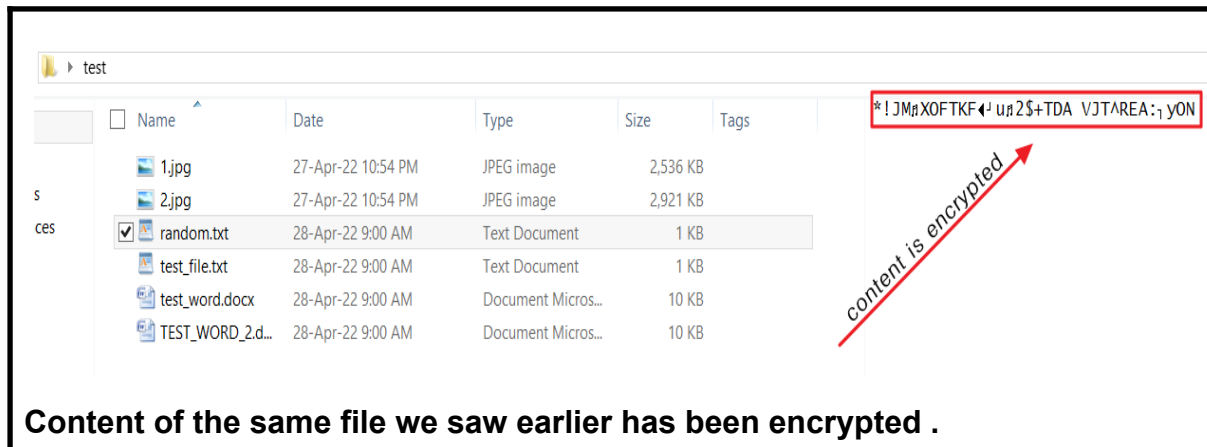


Step 5 : The process of encryption

For the encryption , XOR cipher is being used by our Pyhon_Ransomware using the randomly generated key and the files binary values as shown in following line of code.

```
with open(file,"rb") as f : # read each file in binary mode
    data = f.read()
with open(file,"wb") as f :
    for byte in data : # rewriting the bytes using XOR
        operation
            xor_byte = byte ^ ord(key[index]) #ord grabs the ASCII
            value of the character
            f.write(xor_byte.to_bytes(1,"little"))
```

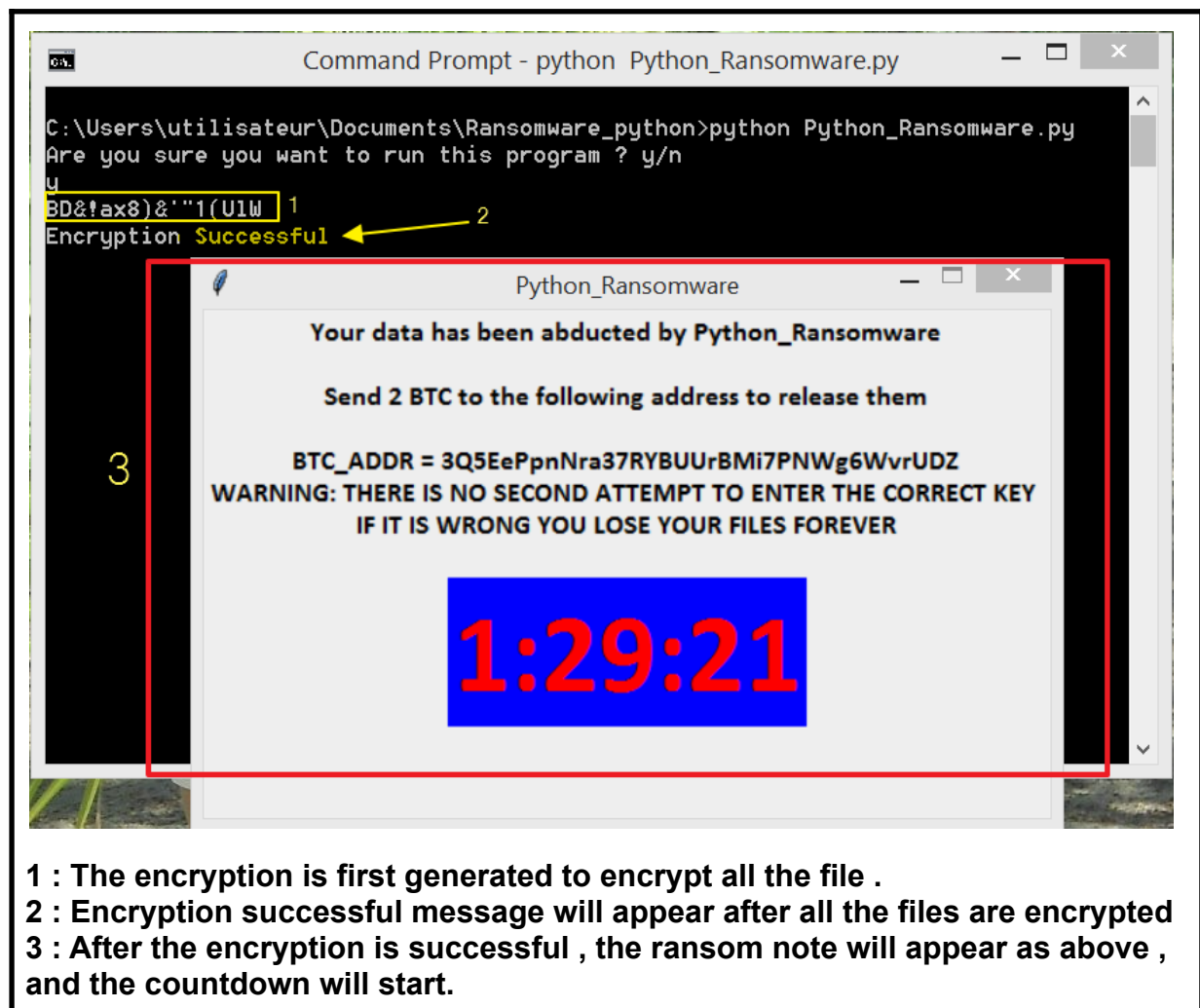
XOR cipher is a very simple cryptography technique as the same key for encryption is needed to reverse the process . (more in the decryptor.py documentation)



Content of the same file we saw earlier has been encrypted .

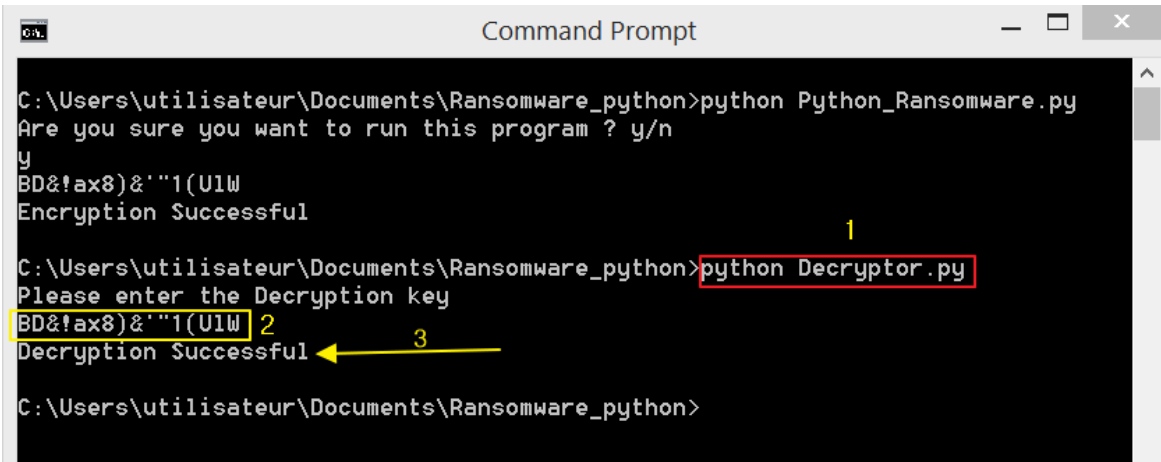
Step 6 : Ransom Note

The ransom note is created using the **tkinter library** of python to generate a user interface warning the victim that the computer has been infected by the ransomware and that the files risks to be lost forever if the given amount of cryptocurrency is paid.

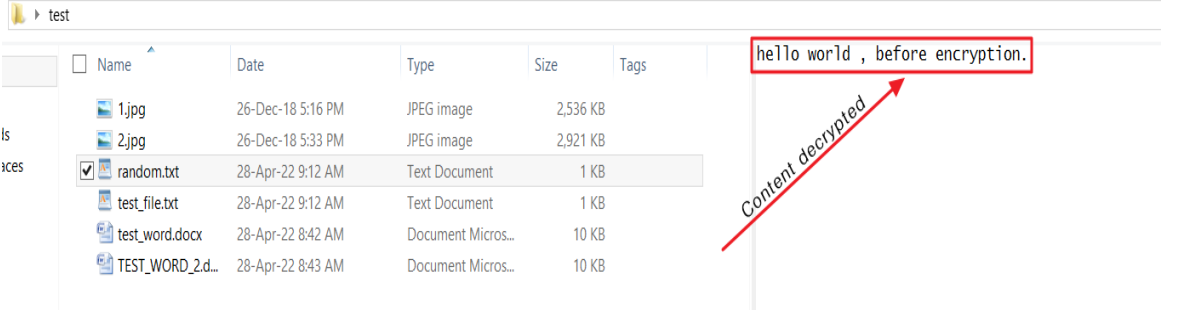


Step 7 : Decryption Process

We just need to run the **Decryptor.py** and paste the decryption key there.



1 : Run the decryptor.py from the same folder as Python_Ransomware.py
2 : Type or paste the same key used for encryption
3 : Decryption Successful message will appear after the decryption process is complete .



We can see now the files has been reverted back to normal.
The content is same as before the encryption .

WHAT TO IMPROVE ?

- Using python language to code malware itself is not always favourable until and unless python is installed on the victim machine which is not the case for newly bought computers . So to avoid this , an executable is required to be generated out of our compiled Python_Ransomware to make it run on all the windows machines .
- This is a very basic ransomware function , as it only encrypts and displays the ransom note . This can be greatly improved if more functionality had to be added such as locking the computer , deleting files , crashing the windows OS ,having the victim machine connected to C2C server , and many more .
- On top of this , stronger cryptography techniques can be used to encrypt the files such as the use of AES or RSA which uses symmetric and asymmetric keys .
- In addition to that, the file extension could have been tampered with as well , to make it more like other ransomware .
- The use of a high level language is not practical as a simple code editor can read the python file ,therefore a need to obfuscate the code is required, creating more confusion among the victims .

CONCLUSION :

Without any doubt , the advancement of malware developers using python programming language reach their end is not anymore out of scope , not for the antivirus company alone , but as well as for the IT industry in general . Therefore a modern mechanism or detection and prevention techniques are required to face this kind of adversary as relying alone on antiviruses and general knowledge is no longer enough . This includes ransomwares strongly as loss is far more than just financial , it involves the loss of all potential data critical to the business .