[악성코드과제2]_[이혜린]_[20190937]

먼저 pngransome.py 에 대한 복호화 파이썬 코드를 만들어 보기로 한다.

pngransome.py 파이썬 코드는 아래와 같다.

```
if not sys.version_info[0] < 3:
    print("Use python2 for this")
    os._exit(0)

if len(sys.argv) != 2:
    print("usage: python pngransome.py [file]")
    os._exit(0)

filename = sys.argv[1]
with open(filename, 'rb') as f:
    buf = f.read()

if buf[1:4] != 'PNG':
    print("this is not PNG file")
    os._exit(0)

enc = ''
for c in buf:
    enc += chr( ord(c) ^ 0xff )

with open(filename+'.enc', 'wb') as f:
    f.write(enc)

print("file is encrypted.")
                                                    28,0-1        Bot
```

pngransome.py에서 파일 내용을 모두 0xff과 XOR 하여 암호화한 것을 알 수 있다.

이때 XOR 은 한번 적용하면 값이 변하지만 두 번 적용하면 값이 원래대로 복원되는 성질이 있다.

따라서 다시 0xff를 다시 전체 파일을 XOR 해주면 원래대로 복호화 될 것이다.

이 때, 암호화된 파일은 buf[1:4] 가 'PNG'가 아닐 것이므로 해당 내용은 주석 처리해준다.

따라서 복호화 파이썬 코드는 다음과 같다.

```python
import os, sys

if not sys.version_info[0] < 3:
    print("Use python2 for this")
    os._exit(0)

if len(sys.argv) != 2:
    print("usage: python pngransome.py [file]")
    os._exit(0)

filename = sys.argv[1]
with open(filename, 'rb') as f:
    buf = f.read()

#if buf[1:4] != 'PNG':
#    print("this is not PNG file")
#    os._exit(0)
dec = ''
for c in buf:
    dec += chr( ord(c) ^ 0xff )

with open(filename+'.dec', 'wb') as f:
    f.write(enc)

print("file is decrypted.")
-- INSERT --                                    27,1          Bot
```

flag.png.enc 를 복호화 해보기로 한다.

```
ellie@ubuntu:~$ python2 restore.py flag.png.enc
file is decrypted.
ellie@ubuntu:~$ eog flag.png.enc.dec
```


다음은 pngransome2.py 에 대한 복호화 파이썬 코드를 만들 차례이다.

아래는 pngransome2.py 파이썬 코드이다.

```
#!/usr/bin/python2
import os, sys

if not sys.version_info[0] < 3:
    print("Use python2 for this")
    os._exit(0)

if len(sys.argv) != 2:
    print("usage: python pngransome2.py [file]")
    os._exit(0)

filename = sys.argv[1]
with open(filename, 'rb') as f:
    buf = f.read()

if buf[:4] != '\x89PNG':
    print("this is not PNG file")
    os._exit(0)

key = '????'
idx = 0
enc = ''
for c in buf:
    k = key[ idx % len(key) ]
    idx += 1
    enc += chr( ord(c) ^ ord(k) )
```

```
with open(filename+'.enc2', 'wb') as f:
    f.write(enc)

print("file is encrypted.")
```

여기서 key가 4 글자이고, 파일 전체가 key 4글자와 순서대로 한글자씩 XOR 되어있다는 것을 알 수 있다.

```
if buf[:4] != '\x89PNG':
    print("this is not PNG file")
    os._exit(0)
```

이 때 이 부분을 통해서 원래 파일의 처음 부분이 항상 0x89 P N G임을 알 수 있고 이를 hexdump 해보면 0x89 0x50 0x4e 0x47 이다.

```
ellie@ubuntu:~$ hexdump -C flag.png.enc2
00000000  e2 63 17 18 66 39 43 55  6b 33 59 52 22 7b 1d 0d  |.c..f9CUk3
YR"{..|
00000010  6b 33 58 ad 6b 33 59 23  63 31 59 5f 6b 2e 67 fa  |k3X.k3Y#c1
Y_k.g.|
00000020  f0 33 59 5f 6a 40 0b 18  29 33 f7 91 77 da 59 5f  |.3Y_j@..)3
..w.Y_|
00000030  6b 37 3e 1e 26 72 59 5f  da bc 52 a3 0a 36 59 5f  |k7>.&rY_..
R..6Y_|
00000040  6b 3a 29 17 32 40 59 5f  65 f0 59 5f 65 f0 58 98  |k:).2@Y_e.
Y e.X.|
```

그리고 암호화된 파일을 hexdump 해보면 처음 4 바이트가 0xe2 0x63 0x17 0x18 임을 알 수 있다.

둘을 XOR 하면 key 값을 알 수 있을 것이다.

XOR 해보면 0x6b 0x33 0x59 0x5f이고 이를 ASCII 코드로 전환하면 key 값이 k3Y_임을 알 수 있다.

이제 키를 전체 파일에 한 글자씩 XOR 해주면 원래 파일로 복호화 될 것이다.

암호화된 파일이므로 /x89PNG를 찾는 내용은 주석 처리한다.

따라서 pngransome2.py에 대한 복호화 파이썬 코드는 다음과 같다.

```
#!/usr/bin/python2
import os, sys

if not sys.version_info[0] < 3:
    print("Use python2 for this")
    os._exit(0)

if len(sys.argv) != 2:
    print("usage: python pngransome2.py [file]")
    os._exit(0)

filename = sys.argv[1]
with open(filename, 'rb') as f:
    buf = f.read()

#if buf[:4] != '\x89PNG':
#    print("this is not PNG file")
#    os._exit(0)

key = 'k3Y_'
idx = 0
dec = ''
for c in buf:
    k = key[ idx % len(key) ]
    idx += 1
    dec += chr( ord(c) ^ ord(k) )
```

```
with open(filename+'.dec2', 'wb') as f:
    f.write(dec)

print("file is decrypted.")
```

flag.png.enc2를 복호화 해보았다.

```
ellie@ubuntu:~$ python2 restore2.py flag.png.enc2
file is decrypted.
```

```
ellie@ubuntu:~$ eog flag.png.enc2.dec2
```