

Academia Hacker INCIBE

reto03 programa_exclusivo

ÍNDICE

ÍNDICE DE FIGURAS	2
ÍNDICE DE TABLAS	2
1. Contexto	3
2. Datos generales	4
3. Firma e integridad del fichero	5
3.1 SHA256sum	5
4. Descripción para participantes	6
5. Pistas	7
5.1 Pista 1: El fichero pyc es compilado en “byte-code”. Usa uncompyle6 para ver el código fuente del programa	7
5.2 Pista 2: Analiza el código fuente. Se realiza un XOR con una clave incluida en el programa. Has de ejecutar el programa con el primer argumento del texto cifrado	7
5.3 Pista 3: Modifica el script para realizar la decodificación (decode=True en línea 24) y ejecuta python xor.py Ax8VCB4HEAAGDDMEBRERPhENAh8DLQQcEQERMAGaBjERAhwEGgADH A== 7	
6. Solución	8

ÍNDICE DE FIGURAS

Ilustración 1: Fichero descompilado	¡Error! Marcador no definido.
Ilustración 2: Modificación del script y ejecución	¡Error! Marcador no definido.

ÍNDICE DE TABLAS

No se encuentran elementos de tabla de ilustraciones.

1. CONTEXTO

Después de muchos años recopilando documentación, fotos y artículos, se ha decidido realizar una limpieza y llevar la documentación antigua a una nueva sala que le ha cedido el instituto. En esta sala se pueden archivar por fecha de una forma sencilla todos los documentos del periódico.

2. DATOS GENERALES

- **Nombre:** Nota cifrada.
- **Conocimientos y/o Habilidades:** Python byte-code, codificación xor.
- **Perfiles:** Estudiantes
- **Tiempo de resolución:** 30m.
- **Prueba de solución:** `flag{tengo_esta_clave_entre_mis_papeles}`

3. FIRMA E INTEGRIDAD DEL FICHERO

3.1 SHA256sum

No aplica.

4. DESCRIPCIÓN PARA PARTICIPANTES

Al darle la vuelta a la hoja, veis que también tiene otro texto escrito en esa parte, aunque tampoco lo entiendes. Se adjunta también un fichero con extensión pyc. ¿Nos servirá de ayuda?

¿Podrás proporcionarnos la otra información que está en la otra cara de la hoja?

Datos proporcionados:

Ax8VCB4HEAAGDDMEBRERPhENAh8DLQQcEQERMAGaBjERAhwEGgADHA==

Datos extra:

Os proporcionamos alguna indicación para empezar a resolver el reto: uncompile6 es una herramienta que descompila el byte-code de Python. Devuelve el fichero original, que necesitarás para resolver la prueba.

5. PISTAS

- 5.1 **Pista 1: El fichero pyc es compilado en “byte-code”. Usa uncompiler6 para ver el código fuente del programa.**
- 5.2 **Pista 2: Analiza el código fuente. Se realiza un XOR con una clave incluida en el programa. Has de ejecutar el programa con el primer argumento del texto cifrado.**
- 5.3 **Pista 3: Modifica el script para realizar la decodificación (decode=True en línea 24) y ejecuta python xor.py
Ax8VCB4HEAAGDDMEBRERPhENAh8DLQQcEQERMAGA
BjERAhwEGgADHA==**

6. SOLUCIÓN

Para poder resolver el reto debemos analizar el texto proporcionado y analizar el fichero .pyc para desensamblar su byte-code y convertirlo a su lenguaje, en este caso Python.

Para poder descompilarlo, existen algunos métodos manuales que requieren conocer las cabeceras y versión de este tipo de fichero. En el reto se añade un método directo para descompilar el fichero pyc. El primer paso es abrir un terminal en nuestra Kali desde el directorio /home/incibe/Escritorio/reto03 con el botón derecho y abrir terminal. Luego tecleamos uncompile6 xor.pyc y nos mostrará la salida con el código fuente.

Otra alternativa para resolver este reto es hacer uso de **pycdc**, herramienta similar que producirá la misma salida.

```
# uncompile6 xor.pyc
# uncompile6 version 3.7.4
# Python bytecode 2.7 (62211)
# Decompiled from: Python 3.8.6 (default, Sep 25 2020, 09:36:53)
# [GCC 10.2.0]
# Embedded file name: xor.py
# Compiled at: 2021-02-01 16:35:44
import binascii, itertools, base64, sys

def xor_crypt_string(data, key='esto es una clave para cifrar', encode=False, decode=False):
    from itertools import izip, cycle
    import base64
    if decode:
        data = base64.decodestring(data)
    xored = ('').join(chr(ord(x) ^ ord(y)) for x, y in izip(data, cycle(key)))
    if encode:
        return base64.encodestring(xored).strip()
    return xored

secret_data = sys.argv[1]
print 'Cifrado'
print xor_crypt_string(secret_data, encode=True)
print 'Descifrado'
print xor_crypt_string(xor_crypt_string(secret_data, encode=True), decode=True)
# okay decompiling xor.pyc
```

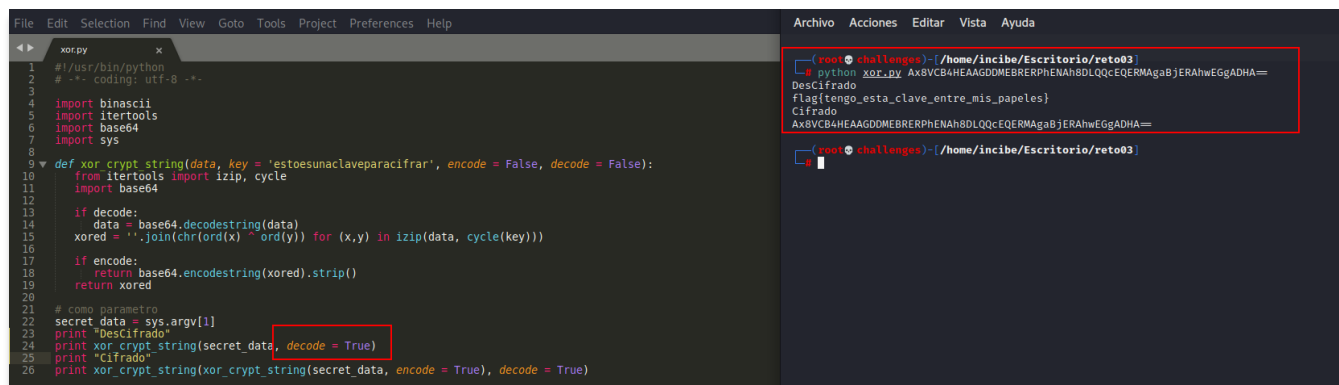
Ilustración 1: Fichero descompilado.

```
# Python bytecode 2.7 (62211)
# Decompiled from: Python 3.8.6 (default, Sep 25 2020, 09:36:53)
# [GCC 10.2.0]
# Embedded file name: xor.py
# Compiled at: 2021-02-01 16:35:44
import binascii, itertools, base64, sys
```



```
def xor_crypt_string(data, key='esto es una clave para cifrar', encode=False, decode=False):  
    from itertools import izip, cycle  
    import base64  
    if decode:  
        data = base64.decodestring(data)  
    xored = ('').join(chr(ord(x) ^ ord(y)) for x, y in izip(data, cycle(key)))  
    if encode:  
        return base64.encodestring(xored).strip()  
    return xored  
  
secret_data = sys.argv[1]  
print 'Cifrado'  
print xor_crypt_string(secret_data, encode=True)  
print 'Descifrado'  
print xor_crypt_string(xor_crypt_string(secret_data, encode=True), decode=True)  
# okay decompiling xor.pyc
```

La función “xor_crypt_string” tiene cuatro argumentos (data, key, encode y decode). Como disponemos explícitamente de la clave y el cifrado xor se emplea en dos direcciones para cifrar y descifrar de igual manera, podemos cambiar el valor decode = True y pasarle el cifrado al programa.



```
File Edit Selection Find View Goto Tools Project Preferences Help  
1 #!/usr/bin/python  
2 # -*- coding: utf-8 -*-  
3  
4 import binascii  
5 import itertools  
6 import base64  
7 import sys  
8  
9  
10 def xor_crypt_string(data, key = 'esto es una clave para cifrar', encode = False, decode = False):  
11     from itertools import izip, cycle  
12     import base64  
13  
14     if decode:  
15         data = base64.decodestring(data)  
16     xored = ''.join(chr(ord(x) ^ ord(y)) for (x,y) in izip(data, cycle(key)))  
17  
18     if encode:  
19         return base64.encodestring(xored).strip()  
20     return xored  
21  
22 # como parametro  
23 secret_data = sys.argv[1]  
24 print "DesCifrado"  
25 print xor_crypt_string(secret_data, decode = True)  
26 print "Cifrado"  
27 print xor_crypt_string(xor_crypt_string(secret_data, encode = True), decode = True)  
  
[root@challenges] ~/home/incibe/Escritorio/reto03  
# python xor.py Ax8VCB4HEAAGDDMEBRERPhENah8DLQcEQERMagaBjERAhwEgGADHA=  
DesCifrado  
flag[tengo_esta_clave_entre_mis_papeles]  
Cifrado  
Ax8VCB4HEAAGDDMEBRERPhENah8DLQcEQERMagaBjERAhwEgGADHA=
```

Ilustración 2: Modificación del script y ejecución.