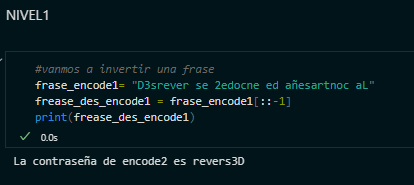
NIVEL1

La frase presenta caracteres alfanuméricos por lo que se descarta el cifrado cesar. Parece estar al reves la frase:

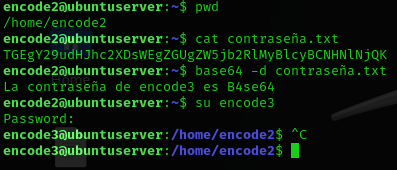


Nota: imagen de todos los niveles a conseguir



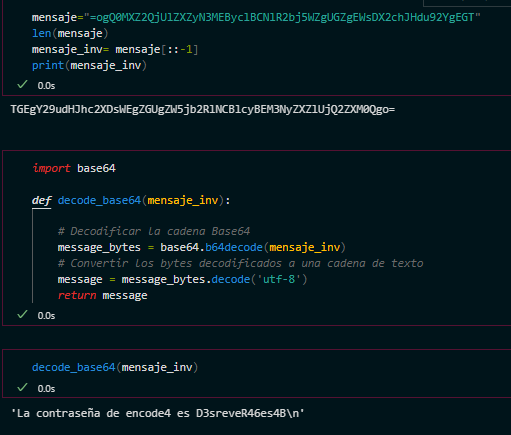
NIVEL2

La contraseña presenta signos de ser base64, ya que presenta caracteres alfanuméricos y su longitud total es múltiplo de 4:



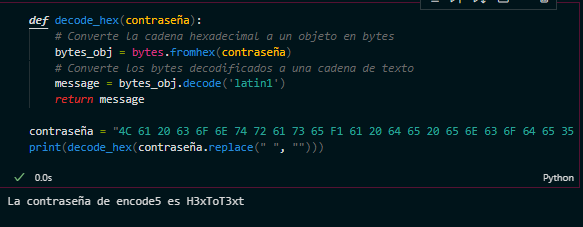
NIVEL3

Se comprueba que tiene la contraseña 60 dígitos, pero al ejecutarse dice que no es base 64. Realizo probaturas, cambiando el símbolo = al final, y hasta intentando darle algún sentido a las letras. Finalmente revierto el orden de la contraseña y así si es válido en base64:



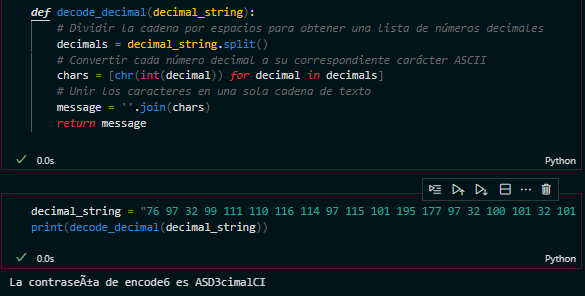
NIVEL4

La contraseña está en formato hexadecimal, por lo que vamos a probar pasarla a una cadena simple de texto :



NIVEL5

Aquí tenemos una cadena de números decimales y necesitamos pasarlo a una cadena de texto, por lo que vamos a pasar cara número decimal a su correspondiente codificación en ASCII y después los unimos en una cadena.

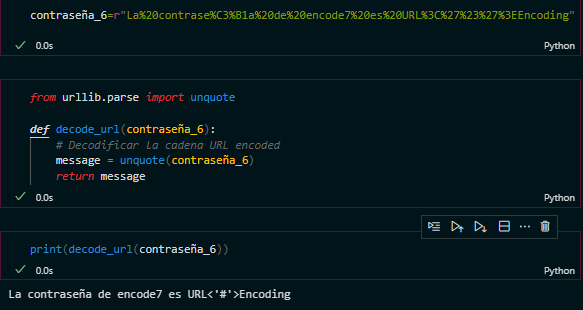


La función “chr” convierte en una lista de comprensión los números decimales a ASCII ( estas listas se lee de atrás hacia delante ) y con “.join” unimos todos los caracteres en una cadena de texto que era la que había que decodificar.

NIVEL6

La contraseña parece estar codificada en “URL encoding”, con el que se codifican las URL, reemplazando los caracteres alfanuméricos en porcentajes seguidos de dos dígitos hexadecimales.

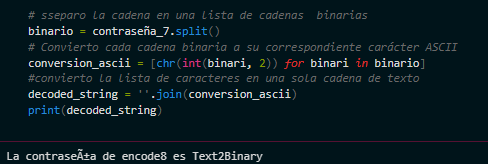
Vamos a intentar usar la función “unquote “de la librería de Python urllib:



NIVEL7

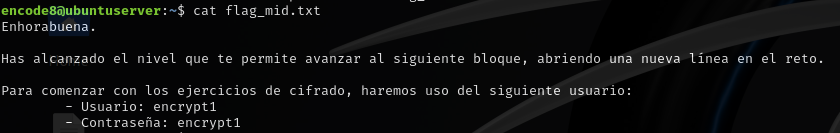
EL código que nos encontramos es binario, el cual es el conocido como lenguaje maquina compuesto por 0 s y 1s.

Voy a volver a usar el comando “chr” para convertir de binario a los 2 tipos de valores que lo componen, para finalmente unirlo toda la cadena de texto con un “join” , decodificando la cadena:



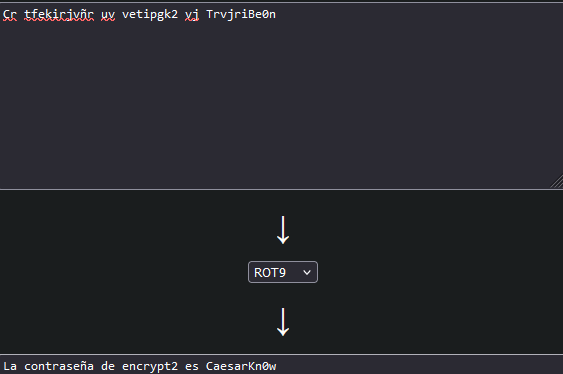
NIVEL8

Aquí encontramos un archivo llamado flag\_mid.txt, que nos abre camino a otro nivel:



ENCRYPT1

Parece que no encontramos ante un cifrado de sustitución (cesar), para ello vamos a usar un recuero disponible en <https://rot13.com/>, indicando que es rot9, lo que significa que cada digito se desplaza en un ciclo de 0 a 9, es decir cada digito + 9 en un ciclo de 10(0 -9) y el desencriptado sería el proceso inverso: digito -9 en un ciclo de 10 (0 a -9),



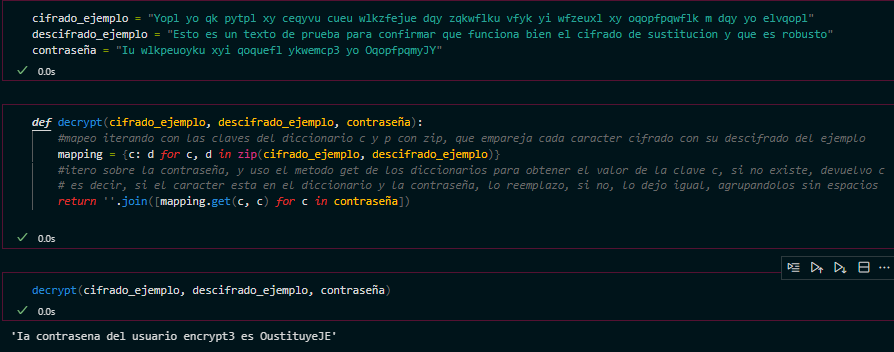
La contraseña de encrypt2 es CaesarKn0w

ENCRYPT2

Mapeo de los ejemplos

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Y | o | p | l | q | k | t | x | c | e | y | v | u | w | z | f | j | d | m | CIFRADO |
| E | s | t | o | u | n | x | d | p | r | a | b | i | f | c | m | q | g | y | DESCIFRADO |

Finalmente, hacemos una función que coge los valores de los ejemplos y el mensaje a descifrar, mapeándolo y devolviendo un mensaje nuevo basado en un diccionario conseguido tras el mapeo. Si la letra de la contraseña esta en el diccionario la cambia y si no, la deja igual.



Parece que le mapeo simple no ha decodificado ciertos caracteres que son la I, O y la J en mayúsculas siendo sus valores L, S, M, por lo que la contraseña queda: SustituyeME

ENCRYPT3

Aquí parece que tenemos un cifrado de transposición por el cual la posición de los caracteres en el texto se cambia siguiendo un sistema, que aquí parece que la clave es TRANSPOSE. Si es así, esta clave determinará un numero de columnas en una tabla donde se escribirá el texto en filas, obteniendo el texto cifrado leyendo las columnas en orden. Para descifrar es revertir el proceso:

