HackOn

2024

Crypto baby

CRYPTO TREE

KDLK99

Índice

1. Descripción

My uncle sent me his Instagram password by email because he suspects that it was leaked in some very popular dictionary with passwords from several years ago, the problem is that we lost the part of the message where the password was, luckily I asked him to send me the root of the Merkle tree to check that the message had not been altered along the way. Do you think that with this you can find out his password?

Message: The password that I use is the same as in the Google account it is ????

Root: 30c085686aa4b1d76ac1c72dfefab6f4a02f5e3865acd76f868b6d5781d2efc8

Once you get the password, the flag is HackOn{hash md5 of the password }

2. Flag

HackOn{827cfbe07f8ffde1c7b457d148075082}

3. Resolución

Un árbol de Merkle es una estructura de datos en forma de árbol en la cual cada hoja es un hash de un bloque de datos, y cada nodo interno es el hash concatenado de sus nodos hijos, de esta forma obtenemos un nodo raíz el cual está formado por todos los nodos hoja. Este tipo de estructura de dato típicamente se utiliza para comprobar la integridad de un mensaje, pero en este caso no tenemos el mensaje completo para comprobar que no ha sido cambiado.

En la descripción del reto nos dicen que la contraseña es posible que esté en algun diccionario popular con contraseñas como por ejemplo el rockyou, podemos utilizar este dato para intentar adivinar la contraseña, para saber si la contraseña que hemos encontrado es la correcta tenemos la raíz del árbol de Merkle con la cual podemos comprobar si el mensaje que hemos reconstruido es el que nos había envíado inicialmente.

Para ello nos tendremos que hacer un script en python el cual lea el diccionario de contraseñas y pruebe cada una de ellas añadiéndola al mensaje recibido y reconstruyendo el árbol para calcular la nueva raíz, si ésta coincide con la que hemos recibido habremos encontrado la contraseña. Para construir un árbol de Merkle tenemos que saber que tipo de hash debemos utilizar, para ello metemos el hash de la raíz en cualquier página *hash identifier* y nos dirá que es un hash SHA256.

4. solver.py

```
import hashlib

def buildTree(word):
    message = 'The password that I use is the same as in the Google account it is ' + word
    message = message.split(' ')
    leaves1 = [hashlib.sha256(word.encode()).hexdigest() for word in message]
    i = 0

    while len(leaves1) > 1:
        leaves2 = []
        i = 0
        while(i < len(leaves1)):
            aux = leaves1[i]
            if i + 1 < len(leaves1):
                aux += leaves1[i+1]
            leaves2.append(hashlib.sha256(aux.encode()).hexdigest())
            i += 2
        leaves1 = leaves2
    return leaves1[0]

originalRoot = '30c085686aa4b1d76ac1c72dfefab6f4a02f5e3865acd76f868b6d5781d2efc8'

with open("rockyou.txt") as file:
    for line in file:
        newRoot = buildTree(line.strip())
        print("Trying password ", line)
        if(originalRoot == newRoot):
            print("The password is:", line)
            break
```