Solutions Sudoku Crypto Challenge

1. Crack the hashes
   1. 8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92   
      This is a SHA256, the solution is 123456. It can be found be google search because it is a common one, or using a sha256 cracker  
      the grid is then

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | 3 |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | 4 |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | 5 |  |  |  | 6 |
|  |  |  |  |  |  |  |  |  |

* 1. dac93e391714bd522e88ab23e382191f

This is a md5, can be crack using a classic cracker, the solution is 788349

The grid is then

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 7 |  |  |  |  |  |
|  |  |  |  |  | 8 |  |  |  |
|  |  |  |  |  |  |  | 3 |  |
|  |  | 8 |  |  |  |  |  |  |
|  |  |  |  |  |  | 3 |  |  |
|  |  |  |  |  | 4 |  |  |  |
|  | 4 |  |  |  |  |  |  |  |
|  |  |  |  | 5 |  |  |  | 6 |
|  |  |  |  |  |  | 9 |  |  |

* 1. 4fbca79cb2b9a0c138ed6ccb34e203e45fa4b7f6

This hash is a sha1, can be cracked as well with online cracker, the solution is 76218

The grid is then

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 7 |  |  |  |  |  |
|  | 7 |  |  |  | 8 |  |  |  |
|  |  |  |  |  |  |  | 3 |  |
|  |  | 8 |  |  |  |  |  |  |
|  |  |  |  |  |  | 3 |  |  |
|  |  | 6 |  |  | 4 |  |  |  |
| 2 | 4 |  |  |  |  |  |  |  |
|  |  |  |  | 5 |  |  | 1 | 6 |
|  |  |  |  | 8 |  | 9 |  |  |

Final hash is a bcrypt one. It is also base64 encoded so it needs to be decoded first and then it can be cracker.   
It needs to be cracked using a code because I don’t think there is online cracker.  
  
A code like this is working   
  
from passlib.hash import bcrypt

import itertools

digits = "123456789"

length = 5

target\_hash = b"$2a$10$Q4od6yVwTMNHdFaGWs1mIeJExJwJlWNBCitx6KfRORc1cNP5CO.gy"

combinations = itertools.product(digits, repeat=length)

for combination in combinations:

string = "".join(combination)

print(string)

combination\_bytes = string.encode()

is\_match = bcrypt.verify(combination\_bytes, target\_hash)

if is\_match:

print("Password is a match!")

print(combination)

sys.exit(0)

The solution for this one is 68751. It can take up to one hour for the code to run.

The grid is then

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 7 |  |  |  |  |  |
|  | 7 |  | 6 |  | 8 |  |  |  |
|  |  |  |  |  |  | 8 | 3 |  |
|  |  | 8 |  |  | 7 |  |  |  |
|  |  |  |  |  |  | 3 | 5 |  |
|  |  | 6 |  |  | 4 |  |  |  |
| 2 | 4 |  |  |  |  |  |  |  |
|  |  |  |  | 5 |  |  | 1 | 6 |
| 1 |  |  |  | 8 |  | 9 |  |  |

1. The player now needs to solve the grid. It is a normal sudoku.

The solved grid is

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 1 | 2 | 7 | 3 | 5 | 6 | 4 | 9 |
| 9 | 7 | 3 | 6 | 4 | 8 | 1 | 2 | 5 |
| 6 | 5 | 4 | 2 | 9 | 1 | 8 | 3 | 7 |
| 4 | 3 | 8 | 5 | 9 | 7 | 2 | 6 | 1 |
| 7 | 9 | 1 | 8 | 6 | 2 | 3 | 5 | 4 |
| 5 | 2 | 6 | 3 | 1 | 4 | 7 | 9 | 8 |
| 2 | 4 | 9 | 1 | 7 | 6 | 5 | 8 | 3 |
| 3 | 8 | 7 | 2 | 5 | 9 | 4 | 1 | 6 |
| 1 | 6 | 5 | 4 | 8 | 3 | 9 | 7 | 2 |

1. Getting the key.  
   Using all the cases in grey color, the player is now able to read the key which is :  
   8645443213158857
2. Getting the IV  
   Using all the cases in orange, the player is now able to read the IV which is :

5465716519896332

1. Decypher  
   Using the key, iv and the good algortithm (AES 256 CBC) as written in the upper cases in the title, the player can decrypt the cypher :  
   eda46763508ed30dbe658769329f54e2b6dfc14282f1092ab47d1f1af1ffa3d8 is in hex format but if you get it from hex and then apply the algo you get the flag which is   
     
   CTF{Fun\_w1th\_h4sh\_&\_Sudoku}