Basic Challenge 1

S Chowdhury

Challenge Description

A 1 byte xor key was used to encrypt the flag (using the xor operation)

The encrypted flag has been provided

encrypted hexademial string: 929895938f80918780ab9298959389

Solution

We have the ciphertext string (in hexadecimal), and we know that a 1 byte xor key was used to do the encryption. A 1 byte xor key could range from 0x00 to 0xff, or 0 to 255. So we can write a python script to perform the brute force.

```
input_string = "929895938f80918780ab9298959389"
input_string_bytearray = bytes.fromhex(input_string)
```

Here we have the input string, then we use the *bytes.fromhex()* method to convert this into a byte array. This means that instead of having the string

9298…

We would have an array of bytes.

```
for ii in range(256):
    result  = encrypt(input_string_bytearray,ii)
    print(f"key: {hex(ii)},  ",end="")
    for jj in range(len(result)):
        print(f"{chr(result[jj])}",end="")
    print("")
```

We have a for loop, and *ii* can have a value from 0 to 255. We call the *encrypt* function on the byte array, and specify *ii* as the key. Please note that for xoring, if we xor a value with a key, we get the xored value. If we xor the xored value with the same key, we get the original value back. This is why we can use the *encrypt()* function for decryption too. The *encrypt* function would return an array of integers. So, we take each item in the array and use that as an input to the *chr()* function. This function converts the integer into the corresponding ASCII character that's represented by that integer.

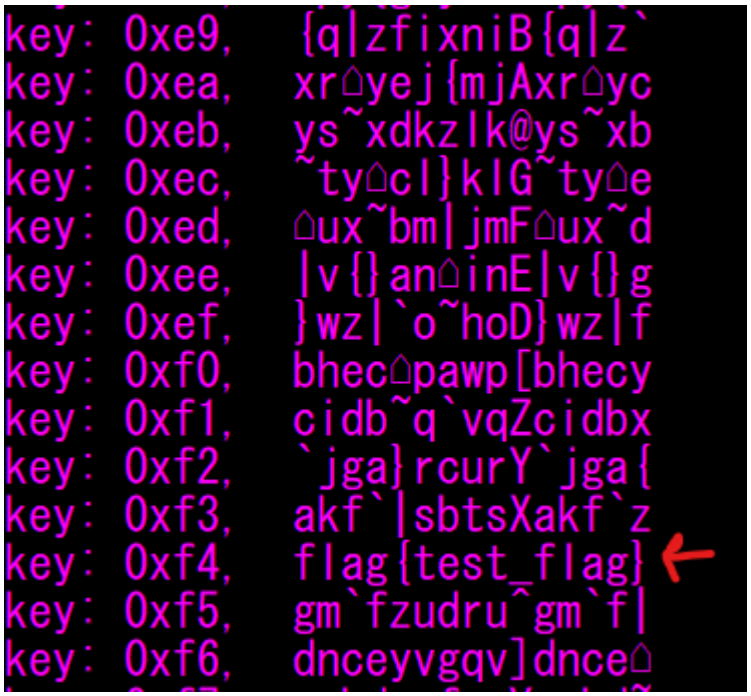
Here's the *encrypt()* function:

```python
def encrypt(data,key):
    final_encrypted = []
    encrypted_byte = 0
    for ii in range(len(data)):
        encrypted_byte = data[ii] ^ key
        final_encrypted.append(encrypted_byte)

    return final_encrypted
```

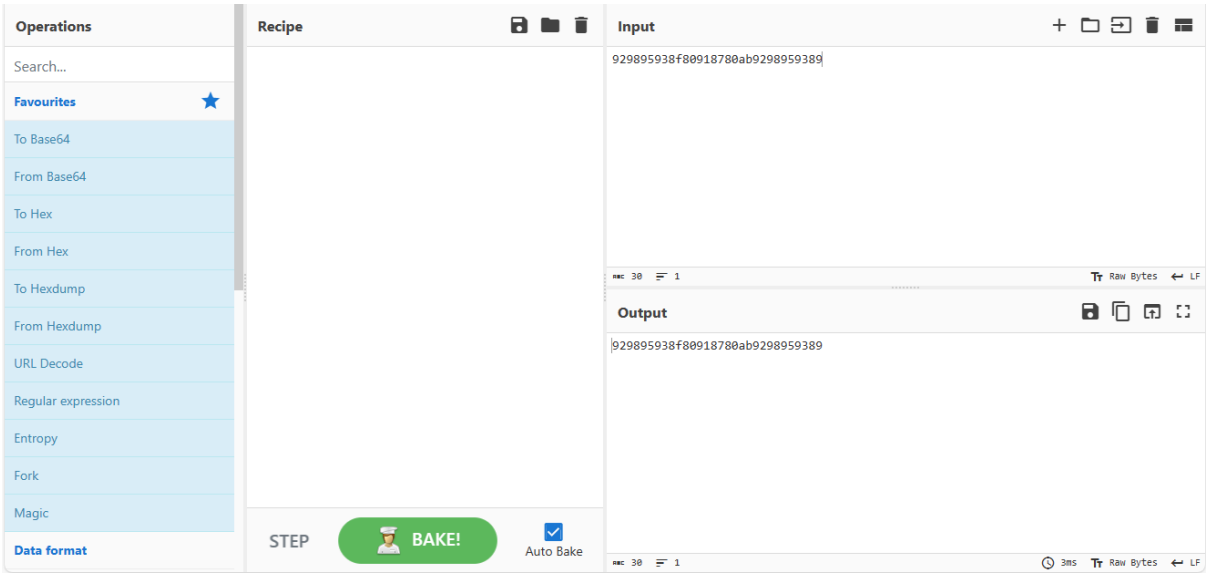
We just take a particular item in the data array, then xor it with the key. The encrypted byte gets stored into the array *final_encrypted*, which gets returned.

Here's the result, we can see the flag and the key:

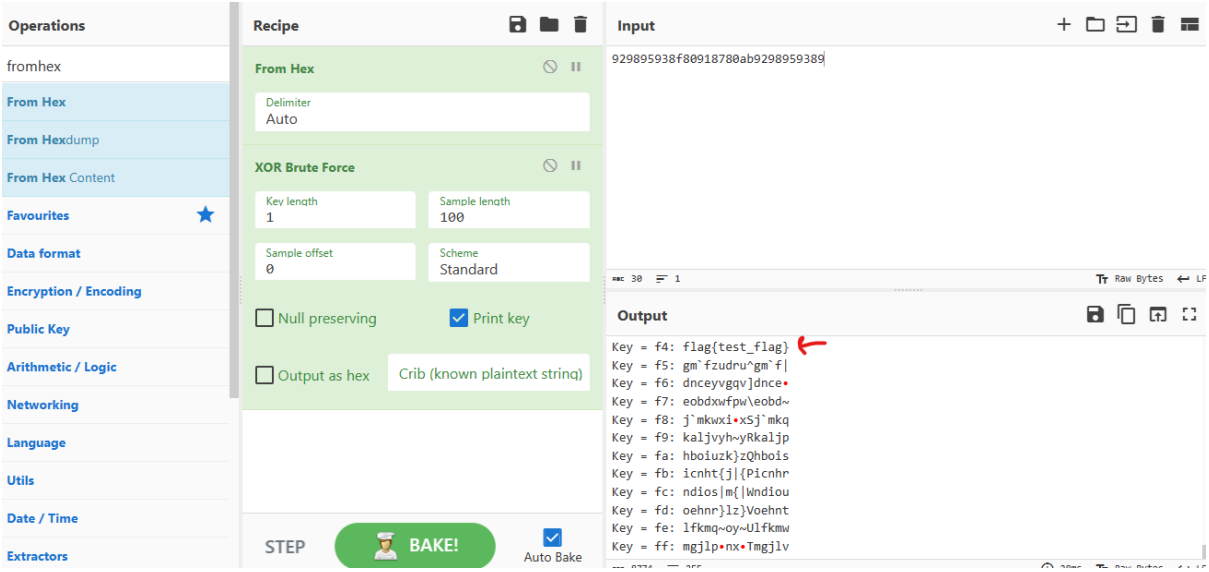


We could have also used the CyberChef website.

First, put the input into CyberChef:



Then, we need the XOR brute force operation as well as the from hex operation:



Python Program

```
def encrypt(data,key):
    final_encrypted = []
    encrypted_byte = 0
    for ii in range(len(data)):
        encrypted_byte = data[ii] ^ key
        final_encrypted.append(encrypted_byte)

    return final_encrypted



input_string = "929895938f80918780ab9298959389"
input_string_bytearray = bytes.fromhex(input_string)




for ii in range(256):
    result  = encrypt(input_string_bytearray,ii)
    print(f"key: {hex(ii)},  ",end="")
    for jj in range(len(result)):
        print(f"{chr(result[jj])}",end="")
    print("")
```