VishwaCTF

CHALLENGE NAME : UNPREDICTABLE

DEV : ANKUSH KAUDI

CATEGORY : CRYPTOGRAPHY

LEVEL : MEDIUM

2024

DESCRIPTION :

From somewhere I got this this scrambled little string and a python file. Not being a tech savy guy, I can't understand a single thing from this. Maybe you can help me understand what it is?

Scrambled string : TgwhsoMVJd,|mxqzv>/|@x+|l{}gp,lb

ATTACHMENT : unpredictable.py

SOLUTION :

```python
import random

class challenge1:
    key = random.randint(1,26)

    def level1(self, message):
        encrypted = ""

        for char in message:
            if char.isalpha():
                shifted = ord(char) + self.key

                if char.islower():
                    if shifted > ord('z'):
                        shifted -= 26

                    elif shifted < ord('a'):
                        shifted += 26

                elif char.isupper():
                    if shifted > ord('Z'):
                        shifted -= 26

                    elif shifted < ord('A'):
                        shifted += 26

                encrypted += chr(shifted)

            else:
                encrypted += char

        return encrypted

    def level2(self, message):
        message = self.level1(message)
        encrypted = ""
        key = random.randint(1,100)

        for char in message:
            charInt = ord(char)
            res = charInt ^ key
            encrypted += chr(res)

        return encrypted


encrypt = challenge1()
print(encrypt.level2(flag))
```

Given python file seems to be an encryption algorithm which is of two steps presented as two function **level1** and **level2.**

The function level1 seems to be generating a random integer value between 1 and 26 and using it as key for shifting the char by generated key times forward in ASCII (basically this function is returning Ceaser Cipher with key generated randomly between 1 and 26).

The function level2 is also working in the same way. It is generating a random value between 1 and 100 and performs XOR operation on each characters ASCII value and the key.

To decrypt the given string we can follow the reverse procedure. Firstly, using the key and reversing the process in level 2 as follows :

if c = a XOR b then a = c XOR b

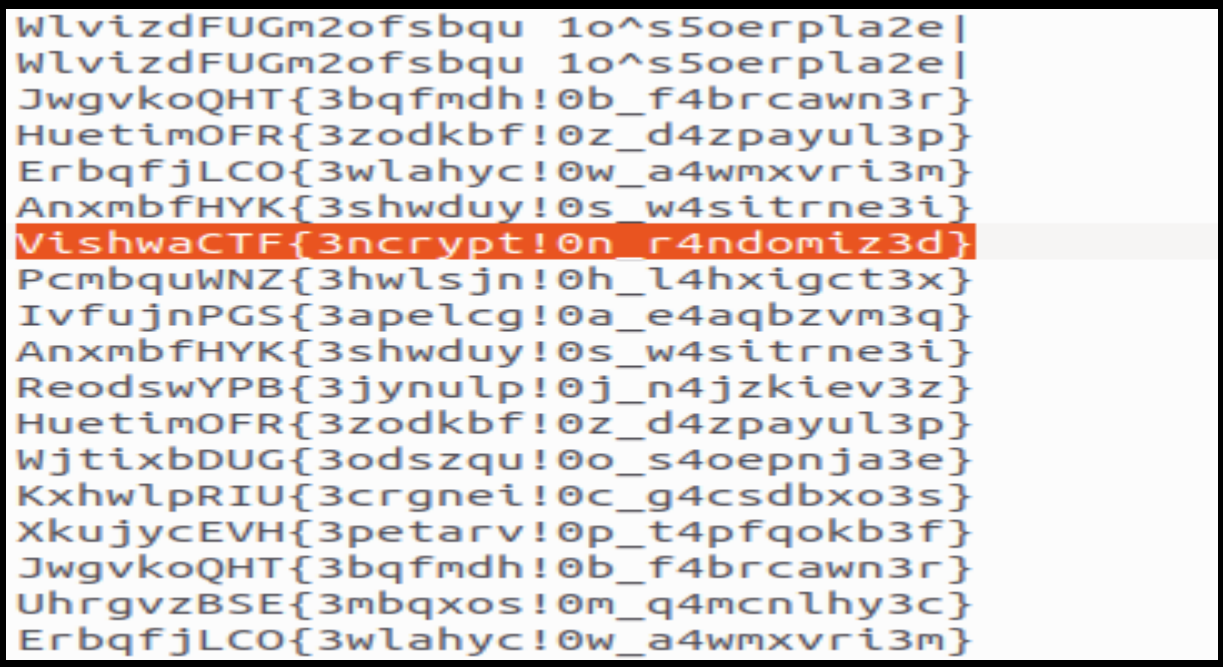
We can use the same procedure on each character's ASCII. Now, since the range is limited we can brute-force the algorithm with all values from the range. This decrypts the level2.

Now, brute-force the level2 output using random keys from the range (1, 26) i.e., perform Ceaser Cipher with keys from range (1, 26). Since, there will be thousands of combination, we can save all the outputs in text and later search for the flag in text file.

The code to decrypt the string is as follows :

```
class challenge1:
    def level1(self, message, key):
        decrypted = ""

        for char in message:
            if char.isalpha():
                shifted = ord(char) - key

                if char.islower():
                    if shifted > ord('z'):
                        shifted -= 26

                    elif shifted < ord('a'):
                        shifted += 26

                elif char.isupper():
                    if shifted > ord('Z'):
                        shifted -= 26

                    elif shifted < ord('A'):
                        shifted += 26

                decrypted += chr(shifted)

            else:
                decrypted += char

        return decrypted

    def level2(self, message):
        decrypted = ""
        for key in range(1, 100):
            for char in message:
                charInt = ord(char)
                res = charInt ^ key
                decrypted += chr(res)

            for ckey in range(1,27):
                decrypted = self.level1(decrypted, ckey)

                f = open("results.txt", "a")
                f.write(decrypted + "\n")

            decrypted = ""

decrypt = challenge1()
encrypted_flag = "TgwhsoMVJd,|mxqzv>/|@x+|l{}gp,lb"
decrypt.level2(encrypted_flag)
```

The above script will brute force the algorithm with the provided ranges and decrypts it, writes it to results.txt. We can find the flag from the text file



Flag : VishwaCTF{3ncrypt!0n_r4ndomiz3d}