

# Unimod

# NahamCon CTF 2022

## The Problem:

I was trying to implement ROT-13, but got carried away.

The task has 2 files:

unimod.py

```
import random
```

```
flag = open('flag.txt', 'r').read()
```

$$ct = 11$$

```
k = random.randrange(0,0xFFFD)
```

```
for c in flag:
```

```
ct += chr((ord(c) + k) % 0xFFFD)
```

```
open( 'out', 'w' ).write(ct)
```

out (I renamed out.txt)

𩚑 𩚒 𩚓 𩚔 𩚕 𩚖 𩚗 𩚘 𩚙 𩚚 𩚛 𩚜 𩚝 𩚞 𩚟 𩚠 𩚡 𩚢 𩚣 𩚤 𩚥 𩚦 𩚧 𩚨 𩚩 𩚪 𩚫 𩚬 𩚭 𩚮 𩚯 𩚰 𩚱 𩚲 𩚳 𩚴 𩚵 𩚶 𩚷 𩚸 𩚹 𩚺 𩚻 𩚼 𩚽 𩚾 𩚿 𩛀 𩛁 𩛂 𩛃 𩛄 𩛅 𩛆 𩛇 𩛈 𩛉 𩛊 𩛋 𩛌 𩛍 𩛎 𩛏 𩛐 𩛑 𩛒 𩛓 𩛔 𩛕 𩛖 𩛗 𩛘 𩛙 𩛚 𩛛 𩛜 𩛝 𩛞 𩛟 𩛠 𩛡 𩛢 𩛣 𩛤 𩛥 𩛦 𩛧 𩛨 𩛩 𩛪 𩛫 𩛬 𩛭 𩛮 𩛯 𩛰 𩛱 𩛲 𩛳 𩛴 𩛵 𩛶 𩛷 𩛸 𩛹 𩛺 𩛻 𩛼 𩛽 𩛾 𩛿 𩜀 𩜁 𩜂 𩜃 𩜄 𩜅 𩜆 𩜇 𩜈 𩜉 𩜊 𩜋 𩜌 𩜍 𩜎 𩜏 𩜐 𩜑 𩜒 𩜓 𩜔 𩜕 𩜖 𩜗 𩜘 𩜙 𩜚 𩜛 𩜜 𩜝 𩜞 𩜟 𩜠 𩜡 𩜢 𩜣 𩜤 𩜥 𩜦 𩜧 𩜨 𩜩 𩜪 𩜫 𩜬 𩜭 𩜮 𩜯 𩜰 𩜱 𩜲 𩜳 𩜴 𩜵 𩜶 𩜷 𩜸 𩜹 𩜺 𩜻 𩜼 𩜽 𩜾 𩜿 𩝀 𩝁 𩝂 𩝃 𩝄 𩝅 𩝆 𩝇 𩝈 𩝉 𩝊 𩝋 𩝌 𩝍 𩝎 𩝏 𩝐 𩝑 𩝒 𩝓 𩝔 𩝕 𩝖 𩝗 𩝘 𩝙 𩝚 𩝛 𩝜 𩝝 𩝞 𩝟 𩝠 𩝡 𩝢 𩝣 𩝤 𩝥 𩝦 𩝧 𩝨 𩝩 𩝪 𩝫 𩝬 𩝭 𩝮 𩝯 𩝰 𩝱 𩝲 𩝳 𩝴 𩝵 𩝶 𩝷 𩝸 𩝹 𩝺 𩝻 𩝼 𩝽 𩝾 𩝿 𩞀 𩞁 𩞂 𩞃 𩞄 𩞅 𩞆 𩞇 𩞈 𩞉 𩞊 𩞋 𩞌 𩞍 𩞎 𩞏 𩞐 𩞑 𩞒 𩞓 𩞔 𩞕 𩞖 𩞗 𩞘 𩞙 𩞚 𩞛 𩞜 𩞝 𩞞 𩞟 𩞠 𩞡 𩞢 𩞣 𩞤 𩞥 𩞦 𩞧 𩞨 𩞩 𩞪 𩞫 𩞬 𩞭 𩞮 𩞯 𩞰 𩞱 𩞲 𩞳 𩞴 𩞵 𩞶 𩞷 𩞸 𩞹 𩞺 𩞻 𩞼 𩞽 𩞾 𩞿 𩟀 𩟁 𩟂 𩟃 𩟄 𩟅 𩟆 𩟇 𩟈 𩟉 𩟊 𩟋 𩟌 𩟍 𩟎 𩟏 𩟐 𩟑 𩟒 𩟓 𩟔 𩟕 𩟖 𩟗 𩟘 𩟙 𩟚 𩟛 𩟜 𩟝 𩟞 𩟟 𩟠 𩟡 𩟢 𩟣 𩟤 𩟥 𩟦 𩟧 𩟨 𩟩 𩟪 𩟫 𩟬 𩟭 𩟮 𩟯 𩟰 𩟱 𩟲 𩟳 𩟴 𩟵 𩟶 𩟷 𩟸 𩟹 𩟺 𩟻 𩟼 𩟽 𩟾 𩟿 𩠀 𩠁 𩠂 𩠃 𩠄 𩠅 𩠆 𩠇 𩠈 𩠉 𩠊 𩠋 𩠌 𩠍 𩠎 𩠏 𩠐 𩠑 𩠒 𩠓 𩠔 𩠕 𩠖 𩠗 𩠘 𩠙 𩠚 𩠛 𩠜 𩠝 𩠞 𩠟 𩠠 𩠡 𩠢 𩠣 𩠤 𩠥 𩠦 𩠧 𩠨 𩠩 𩠪 𩠫 𩠬 𩠭 𩠮 𩠯 𩠰 𩠱 𩠲 𩠳 𩠴 𩠵 𩠶 𩠷 𩠸 𩠹 𩠺 𩠻 𩠼 𩠽 𩠾 𩠿 𩡀 𩡁 𩡂 𩡃 𩡄 𩡅 𩡆 𩡇 𩡈 𩡉 𩡊 𩡋 𩡌 𩡍 𩡎 𩡏 𩡐 𩡑 𩡒 𩡓 𩡔 𩡕 𩡖 𩡗 𩡘 𩡙 𩡚 𩡛 𩡜 𩡝 𩡞 𩡟 𩡠 𩡡 𩡢 𩡣 𩡤 𩡥 𩡦 𩡧 𩡨 𩡩 𩡪 𩡫 𩡬 𩡭 𩡮 𩡯 𩡰 𩡱 𩡲 𩡳 𩡴 𩡵 𩡶 𩡷 𩡸 𩡹 𩡺 𩡻 𩡼 𩡽 𩡾 𩡿 𩢀 𩢁 𩢂 𩢃 𩢄 𩢅 𩢆 𩢇 𩢈 𩢉 𩢊 𩢋 𩢌 𩢍 𩢎 𩢏 𩢐 𩢑 𩢒 𩢓 𩢔 𩢕 𩢖 𩢗 𩢘 𩢙 𩢚 𩢛 𩢜 𩢝 𩢞 𩢟 𩢠 𩢡 𩢢 𩢣 𩢤 𩢥 𩢦 𩢧 𩢨 𩢩 𩢪 𩢫 𩢬 𩢭 𩢮 𩢯 𩢰 𩢱 𩢲 𩢳 𩢴 𩢵 𩢶 𩢷 𩢸 𩢹 𩢺 𩢻 𩢼 𩢽 𩢾 𩢿 𩣀 𩣁 𩣂 𩣃 𩣄 𩣅 𩣆 𩣇 𩣈 𩣉 𩣊 𩣋 𩣌 𩣍 𩣎 𩣏 𩣐 𩣑 𩣒 𩣓 𩣔 𩣕 𩣖 𩣗 𩣘 𩣙 𩣚 𩣛 𩣜 𩣝 𩣞 𩣟 𩣠 𩣡 𩣢 𩣣 𩣤 𩣥 𩣦 𩣧 𩣨 𩣩 𩣪 𩣫 𩣬 𩣭 𩣮 𩣯 𩣰 𩣱 𩣲 𩣳 𩣴 𩣵 𩣶 𩣷 𩣸 𩣹 𩣺 𩣻 𩣼 𩣽 𩣾 𩣿 𩤀 𩤁 𩤂 𩤃 𩤄 𩤅 𩤆 𩤇 𩤈 𩤉 𩤊 𩤋 𩤌 𩤍 𩤎 𩤏 𩤐 𩤑 𩤒 𩤓 𩤔 𩤕 𩤖 𩤗 𩤘 𩤙 𩤚 𩤛 𩤜 𩤝 𩤞 𩤟 𩤠 𩤡 𩤢 𩤣 𩤤 𩤥 𩤦 𩤧 𩤨 𩤩 𩤪 𩤫 𩤬 𩤭 𩤮 𩤯 𩤰 𩤱 𩤲 𩤳 𩤴 𩤵 𩤶 𩤷 𩤸 𩤹 𩤺 𩤻 𩤼 𩤽 𩤾 𩤿 𩥀 𩥁 𩥂 𩥃 𩥄 𩥅 𩥆 𩥇 𩥈 𩥉 𩥊 𩥋 𩥌 𩥍 𩥎 𩥏 𩥐 𩥑 𩥒 𩥓 𩥔 𩥕 𩥖 𩥗 𩥘 𩥙 𩥚 𩥛 𩥜 𩥝 𩥞 𩥟 𩥠 𩥡 𩥢 𩥣 𩥤 𩥥 𩥦 𩥧 𩥨 𩥩 𩥪 𩥫 𩥬 𩥭 𩥮 𩥯 𩥰 𩥱 𩥲 𩥳 𩥴 𩥵 𩥶 𩥷 𩥸 𩥹 𩥺 𩥻 𩥼 𩥽 𩥾 𩥿 𩦀 𩦁 𩦂 𩦃 𩦄 𩦅 𩦆 𩦇 𩦈 𩦉 𩦊 𩦋 𩦌 𩦍 𩦎 𩦏 𩦐 𩦑 𩦒 𩦓 𩦔 𩦕 𩦖 𩦗 𩦘 𩦙 𩦚 𩦛 𩦜 𩦝 𩦞 𩦟 𩦠 𩦡 𩦢 𩦣 𩦤 𩦥 𩦦 𩦧 𩦨 𩦩 𩦪 𩦫 𩦬 𩦭 𩦮 𩦯 𩦰 𩦱 𩦲 𩦳 𩦴 𩦵 𩦶 𩦷 𩦸 𩦹 𩦺 𩦻 𩦼 𩦽 𩦾 𩦿 𩧀 𩧁 𩧂 𩧃

## The Solve:

I examined the coding procedure and the out(put).txt result file.

The task name gave a hint too. Finally the range value 0xFFFFD said we are working with unicode characters.

The coding procedure is weak. It generate only one shift value.

The "`k = random.randrange(0,0xFFFD)`" value adding is outside from the for loop.

So we need to know what is that exactly. We know the flag format, the first letter is 'f'. Then read an 'out.txt' name file and calculate the shift in a string the first index is 0 "*cch = flag[0]*". With this simple value increment method, we have got the value of k shift with condition: "*if cch == chr((ord('f') + i) % 0xFFFD):*".

Finally we use value k for whole 'flag' string.

unirev.py

```
import string
```

```
flag = open('out.txt', 'r').read()
```

pt = 1

```
cch = flag[0]
```

```

print(str(ord(cch)))
for i in range(0,0xFFFD):
    if cch == chr((ord('f') + i) % 0xFFFD):
        k = i

print(str(k),"\n")
for cch in flag:
    print(str(ord(cch)))
    for ch in string.printable:
        if cch == chr((ord(ch) + k) % 0xFFFD):
            pt += ch
print(pt)
open('rev', 'w').write(pt)

```

Run:

*python3 unirev.py*

39239

39137 → (shift) value k

39239

39245

39234

39240

39260

39189

39238

39191

39193

39237

39186

39191

39234

39191

39186

39235

39236

39187

39238

39234

39192

39187

39237

39190

39239

39194

39192

39186

39188

39189  
39189  
39238  
39193  
39189  
39239  
39186  
39186  
39262  
39147

After reverse we have the flag:

**flag{4e68d16a61bc2ea72d5f971344e84f11}**