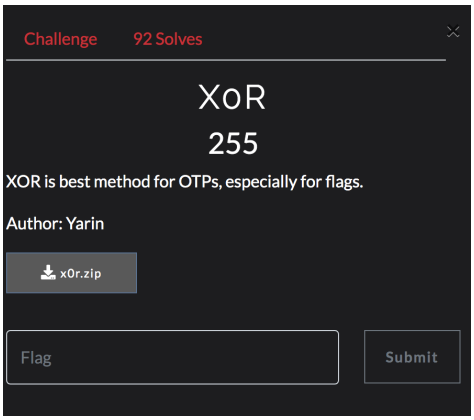
HexionCTF 2020 xor

Saturday, March 7, 2020 12:16 PM



zip expands to:

[flag.enc]
```
".12:9/#
5+60>$/
3/#()/    >
```

[enc.py]
```
from random import choice, randint
from string import ascii_letters
from itertools import cycle

key = ''.join([choice(ascii_letters) for i in range(randint(8, 16))])

with open("flag.txt", "r") as file:
    flag = file.read()

key_gen = cycle(key)
data = []
for i in range(len(flag)):
    data.append(chr(ord(flag[i]) ^ ord(next(key_gen))))

with open("flag.enc", "w+") as file:
    file.write(''.join(data))
```

This generates a random key of upper/lower letters. The length of the key is between 8 and 16 inclusive.

We know the flag is of the form hexCTF{...}

The flag.enc is of length 42 so start with:

```
hexCTF{aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa}  # len 42
```

Then wrote a program which reads flag.enc and XORs with the above string:

```
from itertools import cycle
```

```
flagTemplate = 'hexCTF{aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa}'
with open("flag.enc", "r") as file:
    flagenc = file.read()

print(len(flagenc))

key = []
for i in range(0, len(flagenc)):
    key.append(ord(flagTemplate[i]) ^ ord(flagenc[i]))

print(key)
```

[74, 116, 109, 90, 122, 67, 74, 121, 83, 91, 112, 126, 88, 122, 78, 66, 107, 84, 74, 114, 101, 87, 114, 81, 95, 99, 69, 78, 109, 124, 82, 122, 78, 66, 110, 73, 72, 124, 99, 78, 104, 67]

We know the first 7 chars and the very last char are correct.

So, maybe the key is [74, 116, 109, 90, 122, 67]
The key is repeated over and over so the fact that 74 comes after 67 is a strong clue.

To the above code add:

```
key = [74, 116, 109, 90, 122, 67]
print(''.join([chr(n) for n in key]))

key_gen = cycle(key)
flag = ''
for i in range(len(flagenc)):
    flag = flag + chr(ord(flagenc[i]) ^ k)

print(flag)
```

hexCTF{l_`k\soBypvagilistvluv^yoByukcious}

That flag didn't work. In fact we know this cannot be the key since they key is between 8 and 16 (inclusive chars).

So, let's try to make the key longer. We know the number after 67 MUST be 74 since we know the first 7 chars of the flag.

```
key = [74, 116, 109, 90, 122, 67, 74]
does NOT produce the trailing }

key = [74, 116, 109, 90, 122, 67, 74, 0xff]
does NOT produce the trailing }

key = [74, 116, 109, 90, 122, 67, 74, 0xff, 0xff]
hexCTF{çÍpercaliõÊagilistýÛexpialið×cious}
```

This looks interesting.

Replacing the unknown characters with question marks:

hexCTF{??percali?agilist?expiali?cious}

At this point it looks like a familiar disney move made-up word.

That would make the first ?? be su

Changing the flagTemplate to start with hexCTF{su yields:

[74, 116, 109, 90, 122, 67, 74, 107, 71, 91, 112, 126, 88, 122, 78, 66, 107, 84, 74, 114, 101, 87, 114, 81, 95, 99, 69, 78, 109, 124, 82, 122, 78, 66, 110, 73, 72, 124, 99, 78, 104, 67]

So the real key might be:

[74, 116, 109, 90, 122, 67, 74, 107, 71]

hexCTF{supercaliaragilisticexpialidocious}

Notice this is one letter off from the real made-up work. caliaragil vs. califragil