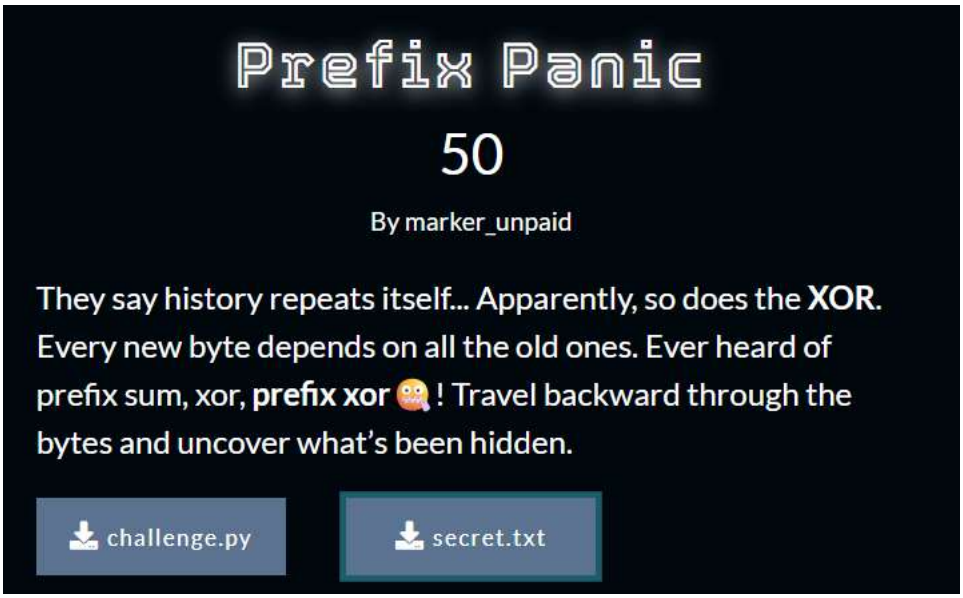
Prefix Panic

22 October 2025 13:40

CATEGORY: Cryptography



STEP BY STEP SOLUTION

It is quite clearly given that prefix-xor was used to get the encrypted text that we see on running the challenge.py file.

To use prefix-xor to encrypt messages, each letter in the message is first converted to its ASCII value. Then the value at the ith position of the encrypted message is obtained by continuously xor-ing all the elements upto the ith element. This value is again converted back to a character using ASCII.

Eg: Let's say [0b111, 0b101, 0b100] are the values
So after prefix-xoring this we'll have
[0b111, (0b111⊕0b101), (0b111⊕0b101⊕0b100)]
ie, [0b111, 0b010, 0b110]
(Notice how the first element does not change after prefix-xoring.)

Now, we need to reverse this. Reversing a xor is easy,
$A \oplus B = C$ implies $B \oplus C = A$.

So, if the ith element is a xor of all the preceding elements including itself, AND we have the first element because it doesn't change, we can work from the first to the last element and reveal the hidden message.

This is the reversing code I used (python):

```
secret = list("I\rN5L|\tV5\x01o0Sc\rcP3\x04[/Gt+O\x7f\x0bx'K{K \x11\x7f\x18G%\x11r
\x19nZ(Ly\x04")

for i in range(len(secret)):
    for j in range(i):
        secret[i] = chr(ord(secret[j])^ord(secret[i]))
for i in secret:
    print(i, end="")
```

And the output I got is:

```
========== RESTART: C:/Users/admin/Deskto
IDC{y0u_c4n_c0nn3c7_th3_d0ts_l00k1ng_b4ckw4rd5}
```

which is our flag.

FLAG: IDC{y0u_c4n_c0nn3c7_th3_d0ts_l00k1ng_b4ckw4rd5}