# picoCTF Write-Up: Codebook

John Gahagan

## Challenge: Codebook

**Category:** Python Exploitation
**Author:** LT 'syreal' Jones

### Challenge Description

Run the Python script `code.py` in the same directory as `codebook.txt`.

**Files Provided:**

- `code.py`
- `codebook.txt`

### Step-by-Step Solution

#### Understanding the Code

The code uses an XOR cipher, where each character of an encrypted flag is XOR'd with a password derived from `codebook.txt`.

Listing 1: Password extraction

```
password = codebook[4] + codebook[14] + codebook[13] + codebook[14] +\
           codebook[23]+ codebook[25] + codebook[16] + codebook[0]  +\
           codebook[25]
```

Given the contents of `codebook.txt`:

```
azbycxdwevfugthsirjqkplomn
```

We extract the characters at the specified indices:

- codebook[4] = `c`
- codebook[14] = `h`
- codebook[13] = `t`
- codebook[14] = `h`

- codebook[23] = `m`
- codebook[25] = `n`
- codebook[16] = `s`
- codebook[0] = `a`
- codebook[25] = `n`

Thus, the password becomes:

`"chthmnsan"`

### Decrypting the Flag

Next, the script XORs the encrypted flag (`flag_enc`) with the password using the `str_xor` function.

Listing 2: Decryption Function

```
def str_xor(secret, key):
    new_key = key
    i = 0
    while len(new_key) < len(secret):
        new_key = new_key + key[i]
        i = (i + 1) % len(key)
    return "".join([chr(ord(secret_c) ^ ord(new_key_c))
                    for (secret_c, new_key_c) in zip(secret, new_key)])
```

To decrypt, run the script with both `code.py` and `codebook.txt` in the same folder:

```
$ python3 code.py
```

## Flag

The script outputs the flag:

```
picoCTF{boo_ya_its_a_codebook}
```

# Conclusion

This challenge demonstrates a basic XOR encryption scheme using a custom password derived from positional indexing in a codebook. The key to solving it was recognizing how the password was formed and applying the XOR decryption using the provided logic.