CTF pyc之stegosaurus隐写

原创

3tefanie丶zhou 已于 2022-03-09 14:36:32 修改 157 收藏

分类专栏： CTF 文章标签： 安全 python web安全

于 2022-03-09 14:35:33 首次发布

文章目录

前言

一个多月没有碰CTF了，今天群里有群友发了一题Misc，随手看看题。

题目

是一个pyc文件

解题过程

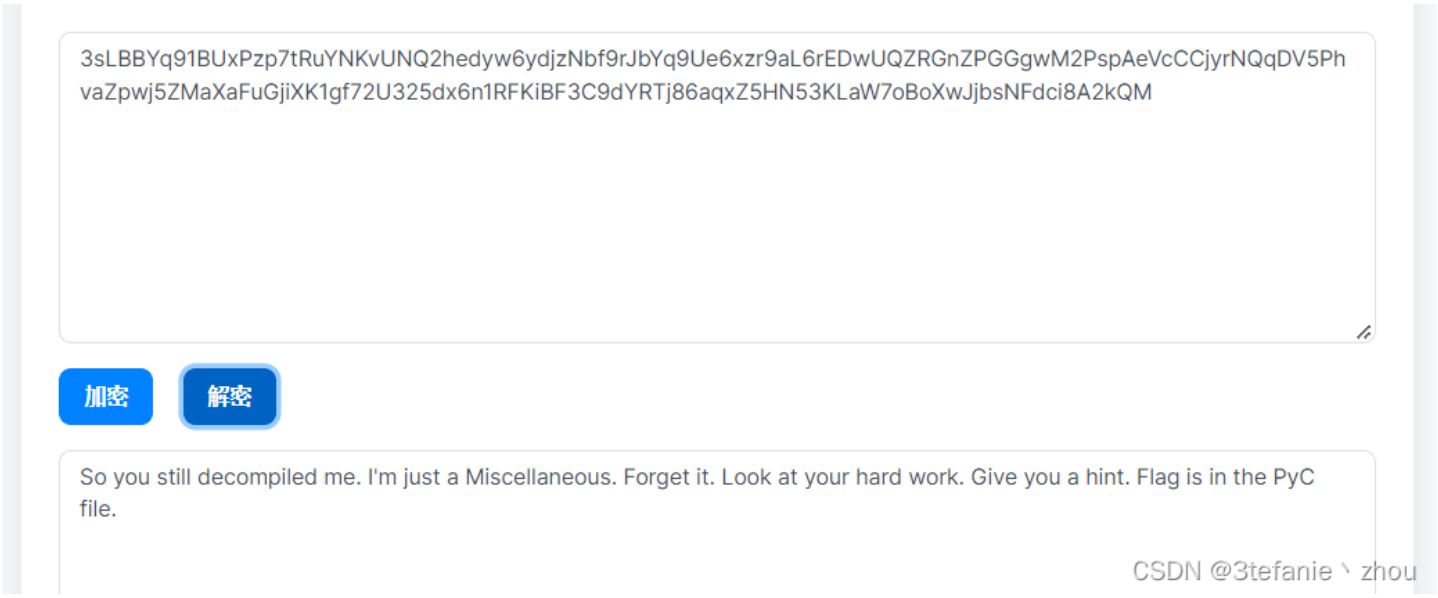
先反编译一下，丢到python反编译在线网站

得到如下代码

```
from hashlib import sha256

__version__ = '1.0.3'
alphabet = b'123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz'
if bytes == str:

    iseq = lambda s: map(ord, s)

    bseq = lambda s: ''.join(map(chr, s))

    buffer = lambda s: s
else:

    iseq = lambda s: s
    bseq = bytes

    buffer = lambda s: s.buffer


def scrub_input(v):
    if isinstance(v, str) and not isinstance(v, bytes):
        v = v.encode('ascii')
    if not isinstance(v, bytes):
        raise TypeError("a bytes-like object is required (also str), not '%s'" % type(v).__name__)


def b58encode_int(i, default_one=(True,)):
    '''Encode an integer using Base58'''
    if not i and default_one:
        return alphabet[0:1]
    string = None
    while i:
        (i, idx) = divmod(i, 58)
        string = alphabet[idx:idx + 1] + string
    return string


def b58encode(v):
    '''Encode a string using Base58'''
    v = scrub_input(v)
    nPad = len(v)
    v = v.lstrip(b'\x00')
    nPad -= len(v)
    (p, acc) = (1, 0)
    for c in iseq(reversed(v)):
        acc += p * c
        p = p << 8

    result = b58encode_int(acc, False, **('default_one',))
    return alphabet[0:1] * nPad + result


def b58decode_int(v):
    '''Decode a Base58 encoded string as an integer'''
    v = v.rstrip()
    v = scrub_input(v)
    decimal = 0
```

```
    decimal = 0
    for char in v:
        decimal = decimal * 58 + alphabet.index(char)

    return decimal


def b58decode(v):
    '''Decode a Base58 encoded string'''
    v = v.rstrip()
    v = scrub_input(v)
    origlen = len(v)
    v = v.lstrip(alphabet[0:1])
    newlen = len(v)
    acc = b58decode_int(v)
    result = []
    while acc > 0:
        (acc, mod) = divmod(acc, 256)
        result.append(mod)
    return b'\x00' * (origlen - newlen) + bseq(reversed(result))


def b58encode_check(v):
    '''Encode a string using Base58 with a 4 character checksum'''
    digest = sha256(sha256(v).digest()).digest()
    return b58encode(v + digest[:4])


def b58decode_check(v):
    '''Decode and verify the checksum of a Base58 encoded string'''
    result = b58decode(v)
    result = result[:-4]
    check = result[-4:]
    digest = sha256(sha256(result).digest()).digest()
    if check != digest[:4]:
        raise ValueError('Invalid checksum')


if __name__ == '__main__':
    if b58encode(input()) == b'3sLBBYq91BUxPzp7tRuYNKvUNQ2hedyw6ydjzNbf9rJbYq9Ue6xzr9aL6rEDwUQZRGnZPGGgwM2PspAeV
cCCjyrNQqDV5PhvaZpwj5ZMaXaFuGjiXK1gf72U325dx6n1RFKiBF3C9dYRTj86aqxZ5HN53KLaW7oBoXwJjbsNFdci8A2kQM':
        print('flag is coming...')
    else:
        print('There is no problem that your input is wrong.')
```

应该是一段base58编码和解码的代码
将main()中的base58编码后的字符串取出进行解码

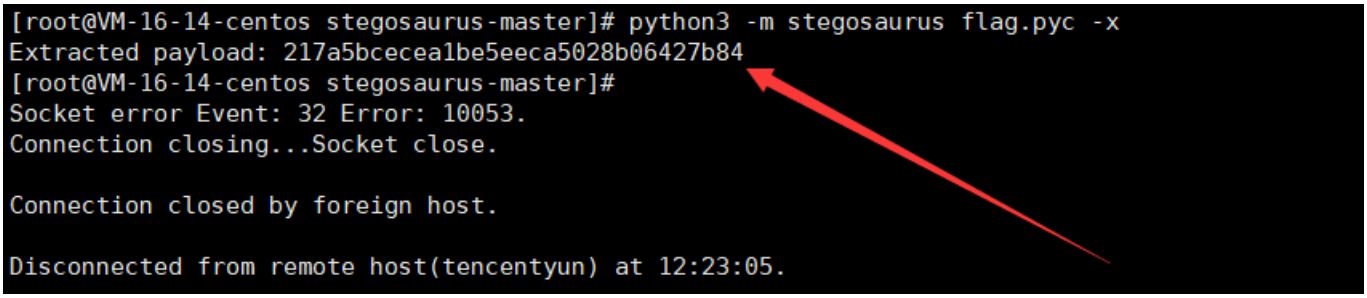


解密后的明文给了一个hint：**flag在pyc文件中**
去年在长安战疫中刚好碰到过一题是pyc文件隐写，可以使用stegosaurus工具获得pyc文件中的隐藏信息

stegosaurus工具项目地址

```
https://github.com/AngelKitty/stegosaurus
```

后面就很简单了，找一个3.6以上的环境跑一下工具即可




```
flag{217a5bcecea1be5eeca5028b06427b84}
```

【施恩宜由淡转浓，由浓转淡反成仇。刑罚宜由严转宽，先宽后严怨其酷。】