CTF【解密】字符串flag被加密成已知新字符串，请解密出flag，可以使用Python解码出WriteUp

原创

皓月盈江 已于 2022-03-07 10:16:07 修改 4241 收藏

分类专栏： CTF 文章标签： python CTF 解密 WriteUp
于 2022-03-06 15:51:48 首次发布

本文链接：https://blog.csdn.net/u013541325/article/details/123311146

CTF-解密： 找出flag

task.py

```
# -*- coding: utf-8 -*-

assert flag[0:5] == 'flag{'

strAlphabet = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'


def encode(strOld, x, y, n):
    strNew = ''

    for i in strOld:
        if i in strAlphabet:
            num = strAlphabet.index(i)  # 返回索引值
            strNew += strAlphabet[(x*num - y) % n]
        else:
            strNew += i

    print(strNew)


if __name__ == '__main__':
    encode(flag, 29, 30, 52)
    # strNew = 'lDwO{kIDCmgI_bm_brI_cBL_crwDDIJOI}'
```

WriteUp：

由上述加密代码可知，我们要找的flag是以`'flag{'`开始的字符串，flag和加密后strNew中的字符都在strAlphabet中，不在的都原字符添加到strNew中，下标存在对应关系。

```
f对应l
l对应D
a对应w
g对应o
```

`(29*num -30) % 52`是对52取余数，取得整数可能是0， 1， 2， …

因此，解码如下：

main.py

```
# -*- coding: utf-8 -*-

strAlphabet = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
strNew = 'lDwO{kIDCmgI_bm_brI_cBL_crwDDIJOI}'


def decode(strEncode, x, y, n):
    strFlag = ''

    for i in strEncode:
        if i in strAlphabet:
            index = strAlphabet.index(i)  # 返回索引值

            for nTemp in range(0, 28):
                if ((index + n*nTemp + y) % x) == 0:
                    strFlag += strAlphabet[(index + n*nTemp + y) // x]
                    break
        else:
            strFlag += i

    print(strFlag)


if __name__ == '__main__':
    decode(strNew, 29, 30, 52)
```

运行结果如下：

```
flg{Welcome_to_the_CTF_Chllenge}
```

由于**a对应w**，所以

```
加密后字符串：'lDwO{kIDCmgI_bm_brI_cBL_crwDDIJOI}'
加密前字符串：'flag{Welcome_to_the_CTF_Challenge}'
```

则，flag就是`'flag{Welcome_to_the_CTF_Challenge}'`

为什么`for nTemp in range(0, 28):`中nTemp最大取值是27，这是因为strAlphabet字符串最大索引值是51，`(index + 52*nTemp + 30) // 29 =51`，那么nTemp=(1449-index)//52,可以看出index最小值是0，nTemp最大可以取值27。