安恒月赛2020元旦场Writeup

原创

LetheSec 于 2020-01-01 19:10:37 发布 2017 收藏 3

分类专栏： CTF wp

本文链接：https://blog.csdn.net/qq_42181428/article/details/103794808


CTF 同时被 2 个专栏收录

24 篇文章 8 订阅

订阅专栏


wp

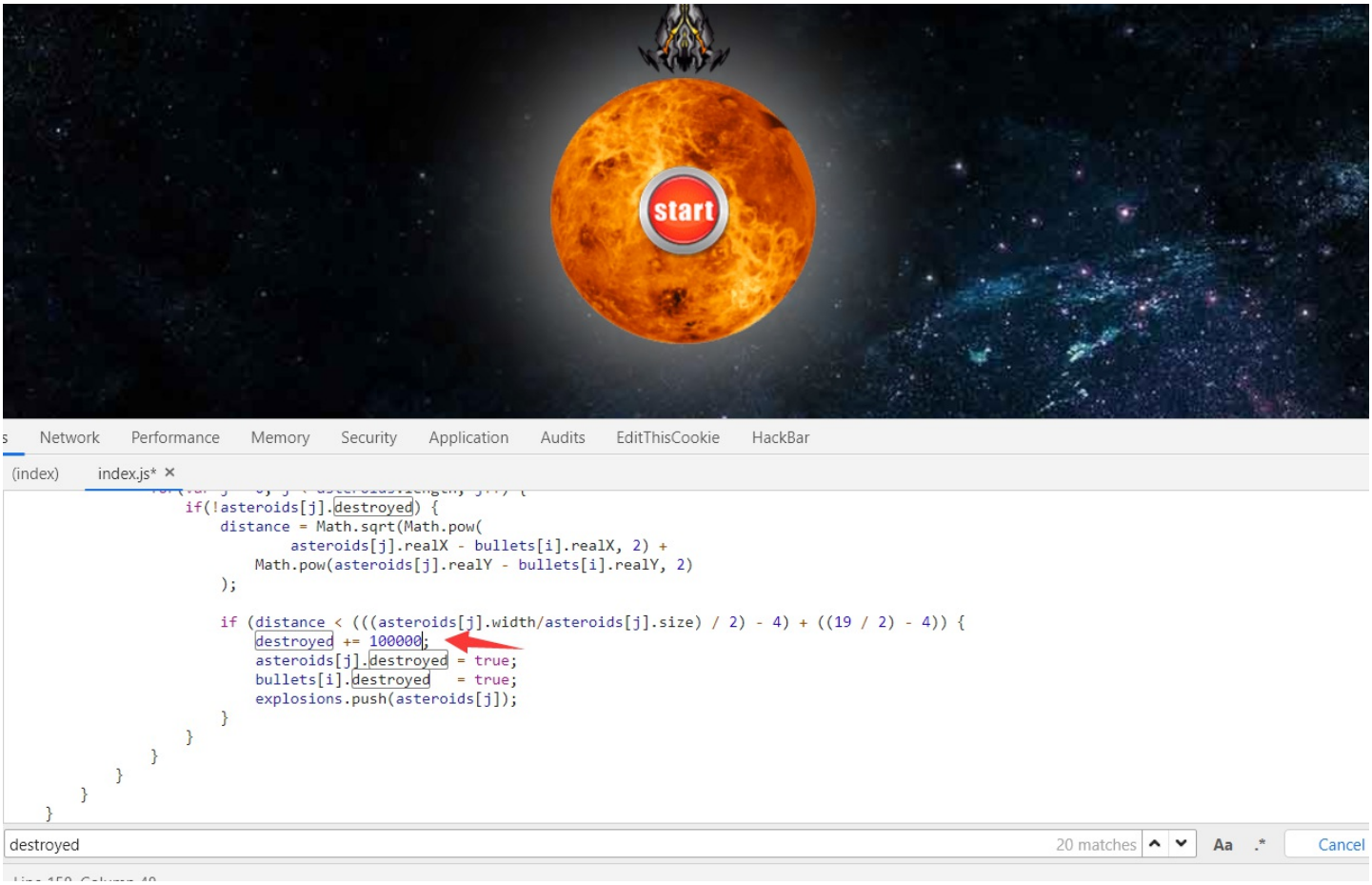
11 篇文章 0 订阅

订阅专栏

正在复习密码学，突然发现今天有月赛了...于是做了一下，很简单的一次题目。。

Web1

进入后是一个JS的游戏，在`index.js`里修改一下`destroyed`的增加，设置大一些，如下：



然后随便玩一下就能得到flag了：

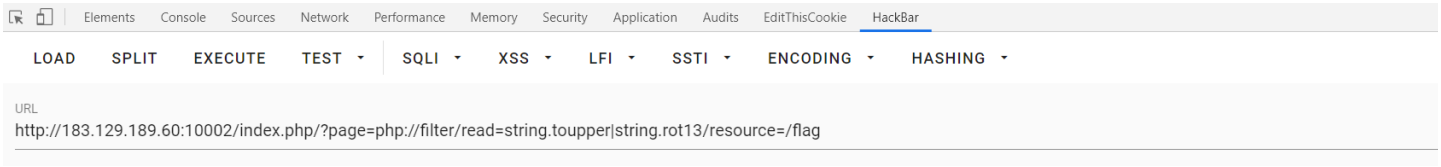
Web2

题目名字叫地图，进入后发现实际上没什么功能，只有`index.php`一个页面，

点击发现url中存在`page=index`，于是尝试文件包含，常用的base64编码的payload会返回`not base`，应该是过滤了base，使用rot13编码读取，先读`index.php`可以发现flag在根目录，payload如下：

```
/index.php/?page=php://filter/read=string.toupper|string.rot13/resource=/flag
```

SYNT{9N48QQNQ2656385SPR58NS47N0RS56PS}



rot13解码得到flag。

Crypto1

题目如下给了3个png和一个加密脚本如下：

```
from itertools import *
from key import key

ki= cycle(key)

fr1 = open("flag.png","rb")
fr2 = open("fflag.png","rb")
fw1 = open("flag_e.png","wb")
fw2 = open("fflag_e.png","wb")

for now in fr1:
    for nowByte in now:
        newByte = nowByte ^ ord(next(ki))
        fw1.write(bytes([newByte]))
fr1.close()
fw1.close()

for now in fr2:
    for nowByte in now:
        newByte = nowByte ^ ord(next(ki))
        fw2.write(bytes([newByte]))
fr2.close()
fw2.close()
```

其中fflag.png是打了马赛克的flag，看一下代码，flag.png加密得到flag_e.png，fflag.png加密得到fflag_e.png，但是加密用的ki不知道。简单分析一下，加密过程就是循环用ki每一位的ascii的与图片进行异或，且两次的加密过程一样，那么思路就清楚了。

我们有fflag.png和加密后的fflag_e.png，前者相当于明文m，后者相当于密文c，加密过程为c=m⊕k，那么现在已知m和c，容易得到k=m⊕c
有如下脚本计算ki的ascii：

```
fr_m = open("fflag.png", "rb")
fr_c = open("fflag_e.png", "rb")
m = []
c = []
ki = []
for now in fr_m:
    for nowByte in now:
        m.append(nowByte)

for now in fr_c:
    for nowByte in now:
        c.append(nowByte)

for i in range(len(m)):
    ki.append(m[i] ^ c[i])

fr_m.close()
fr_c.close()
print(ki)
```

很容易得到ki如下：

```
[65, 108, 105, 116, 97, 95, 105, 115, 95, 115, 111, 95, 99, 117, 116, 101]
```

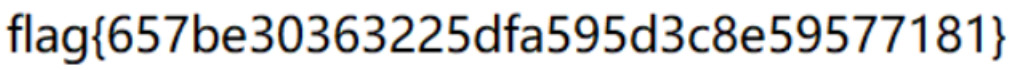
于是对flag_e.png进行解密如下：

```
flag_r = open("flag_e.png","rb")
flag_w = open("flag.png","wb")
ki = [65, 108, 105, 116, 97, 95, 105, 115, 95, 115, 111, 95, 99, 117, 116, 101]*500
c = []
for now in flag_r:
    for nowByte in now:
        c.append(nowByte)

for i in range(len(c)):
    newByte = c[i] ^ ki[i]
    flag_w.write(bytes([newByte]))

flag_r.close()
flag_w.close()
```

运行得到flag.png:



Crypto2

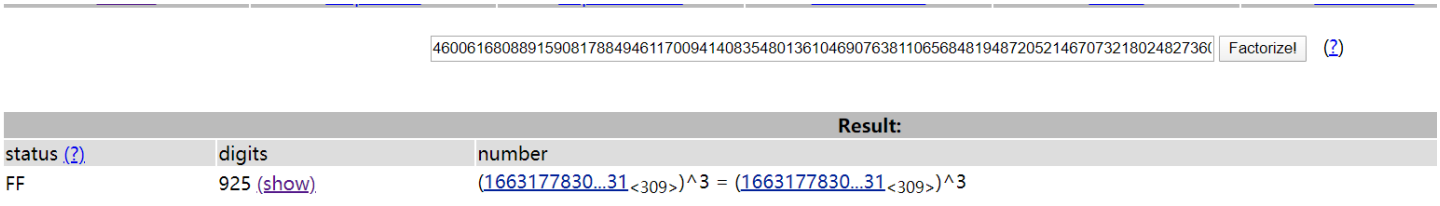
一道RSA的题目，给的脚本如下：

```
from Crypto.Util.number import *
import gmpy2
import random
from flag import flag


p = getPrime(1024)
r = random.randint(2, 10)
e =65537
n = p ** r
m=flag
assert(int(m.encode('hex'), 16) < n)
c = pow(int(m.encode('hex'), 16),e,n)
c=long_to_bytes(c)
print 'c =\n', c.encode('base64'),n


'''
c =
apxy3z3DgGnzaEedcUy3A49wAsqyyn9sqx6eYZL5iDrCq0Wjs8BOY2Ofza5wuaFigm32PVpO5jpu
Dgw9b6oX8KM2ZB9/dDmwQc7JKnAKhCQrIc1v9qt7iQbnTK0DTQj/xvQkz/IBeSjoWBmHOx4s0tDx
ZRAjOPui5wwAywNM3ynULEPczv+xN2v+6HBeoS2YuyfF5mq/pIAMPwZs+QpkuwxSbNQ6xPNP9Ox1
IeKz/41F7/D2fDsGB5CcFdAiQq+r95BhVeGzeaiQBpzwAXAPKIyO+fP6/M9XmpSJwjaMSiAUnksp
9KfVOXgEG9Z0FmxP6rgqPl0vU+rVeJ2RsTUYCSP8Vy+PD3PGwDDdUtNzvcEXKr2BKiNoOUxprBAt
yvcsmGqRLgDl1ZVgzSZ1U4MAmJ9x42mIU0XvolqaOCJZzaym1kJoBlw7/7+Nej4owEtan/c3TIkD
kr/gCenUD/8MSlvnfTUMGdQLkSht2BZiuiHxVVRVzY5ETG6v+w9AtDMC
460061680889159081788494611700941408354801361046907638110656848194872052146707321802482736007398055062035379208452076737230434713253578487567102656316058359838677371858611103482655568960282456317246344692428707257038671271987034886290493637089469510830249086782609435207213269674311674163511186020504912971794852053427092483431870424499969053243194124890525788034756122115184197898224019139736403849025093060421125638592549665862075558205875337632858300131284650829531928694183722052256372921592811116427404289069677182075985679099446194420926973276926955925760844068671320662211164927589842604093130100571144605581970770408620135771295992281430006790753616184125553317180531314933238371299709178036814262549905514980604323805703740051019725536447168581500415435704987420588468232244339137402016911483372261685125789536964847204811632026654856078773376412628110264547425201371450701457762045081645915384827908491045728854919 1
'''
```

给了n和密文c，可以看到n是用p的r次幂，r为2~10的随机数，分解一下n试试，如下：



Result:		
status (?)	digits	number
FF	925 (show)	(1663177830...31<309>)^3 = (1663177830...31<309>)^3

发现可以成功分解，并且知道r为3，这样就很容易求出d了，解密脚本如下：

```python
import base64
from Crypto.Util.number import bytes_to_long
import gmpy2
import libnum

n = 46006168088915908178849461170094140835480136104690763811065684819487205214670732180248273600739805506203537920845207673723043471325357848756710265631605835983867737185861110348265556896028245631724634469242870725703867127198703488629049363708946951083024908678260943520721326967431167416351118602050491297179485205342709248343187042449996905324319412489052578803475612211518419789822401913973640384902509306042112563859254966586207555820587533763285830013128465082953192869418372205225637292159281111642740428906967718207598567909944619442092697327692695592576084406867132066221116492758984260409313010057114460558197077040862013577129599228143000679075361618412555331718053131493323837129970917803681426254990551498060432380570374005101972553644716858150041543570498742058846823224433913740201691148337226168512578953696484720481163202665485607877337641262811026454742520137145070145776204508164591538482790849104572885491 91
e =65537
c ="apxy3z3DgGnzaEedcUy3A49wAsqyyn9sqx6eYZL5iDrCq0Wjs8BOY2Ofza5wuaFigm32PVpO5jpuDgw9b6oX8KM2ZB9/dDmwQc7JKnAKhCQrIc1v9qt7iQbnTK0DTQj/xvQkz/IBeSjoWBmHOx4s0tDxZRAjOPui5wwAywNM3ynULEPczv+xN2v+6HBeoS2YuyfF5mq/pIAMPwZs+QpkuwxSbNQ6xPNP9Ox1IeKz/41F7/D2fDsGB5CcFdAiQq+r95BhVeGzeaiQBpzwAXAPKIyO+fP6/M9XmpSJwjaMSiAUnksp9KfVOXgEG9Z0FmxP6rgqPl0vU+rVeJ2RsTUYCSP8Vy+PD3PGwDDdUtNzvcEXKr2BKiNoOUxprBAtyvcsmGqRLgDl1ZVgzSZ1U4MAmJ9x42mIU0XvolqaOCJZzaym1kJoBlw7/7+Nej4owEtan/c3TIkDkr/gCenUD/8MSlvnfTUMGdQLkSht2BZiuiHxVVRVzY5ETG6v+w9AtDMC"
p = 16631778300856146161980935433814936995552950080487778469613539444556283756439226347837899675276602476947231103493005853597662495202279644971165076615530735950828972426718055175850342791227121671707461009028363513162261243515289813501164805400451185795535150672271221387718007498729219890507322208460963347183 1
r = 3
phin = pow(p, 3) - pow(p, 2)
d = gmpy2.invert(e, phin)
c = bytes_to_long(base64.b64decode(c))
m = pow(int(c), d, n)
flag = libnum.n2s(m)
print(flag)
```

运行得到flag：

```
PS C:\Users\Lethe\Desktop\RSA> python3 .\exp.py
flag{7422e7ed91c8089a1f2aa323a6a0a6f9}
```