

babyRe-wp

出题人：D0UBL3SEV3N

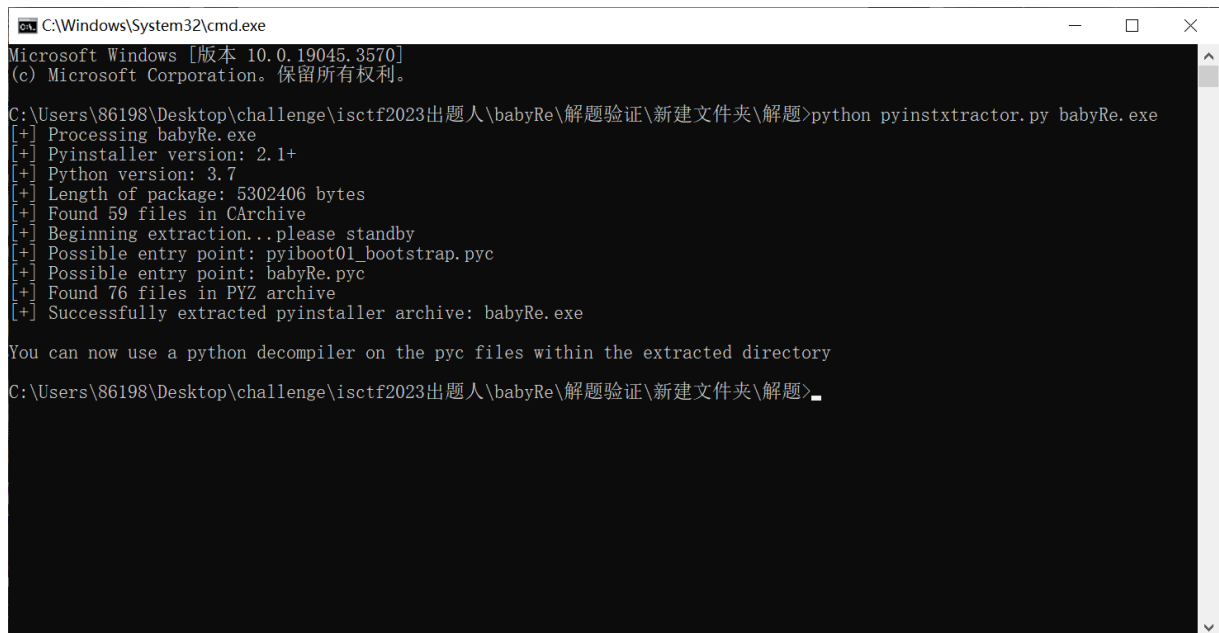
学校：大理大学

考点：pyinstxtractor, python 逆向

下载附件，是一个 txt 文件和一个 exe 文件。

exe 是 pyinstaller 打包 py 文件生成的 exe 文件。用 pyinstxtractor2.0 解 exe 文件。因为 pyinstxtractor1.0 解包的时候存在丢失头文件的情况。也可以手动在解 exe 下来的 题目名称.pyc 文件添加 struct.pyc 文件头。但是使用 pyinstxtractor2.0 就不会存在上述问题。具体细节请了解 pyc 逆向,这里不再赘述。

首先，使用 pyinstxtractor2.0 把 exe 程序解包。



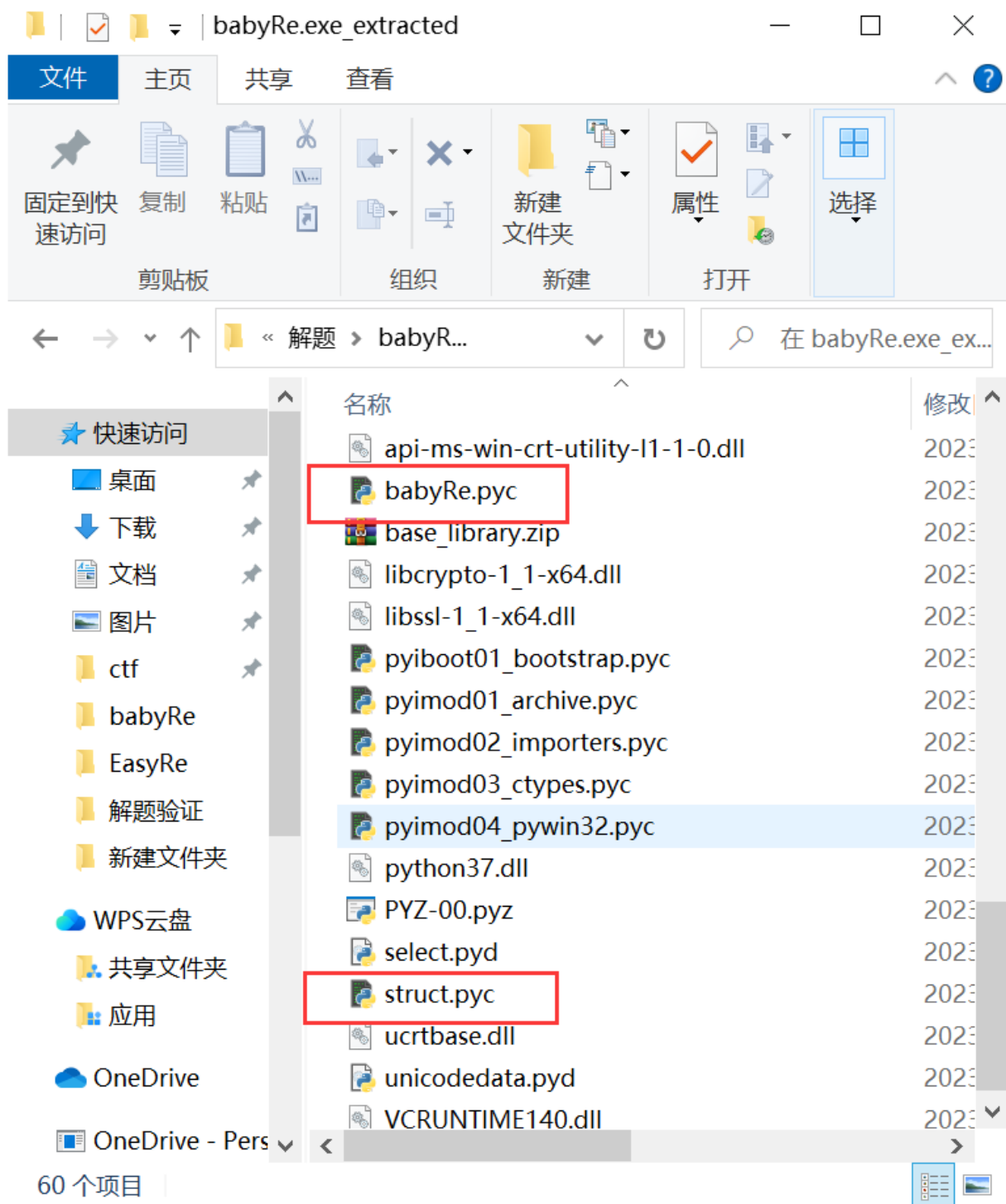
```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19045.3570]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\86198\Desktop\challenge\isctf2023出题人\babyRe\解题验证\新建文件夹\解题>python pyinstxtractor.py babyRe.exe
[+] Processing babyRe.exe
[+] Pyinstaller version: 2.1+
[+] Python version: 3.7
[+] Length of package: 5302406 bytes
[+] Found 59 files in CArchive
[+] Beginning extraction...please standby
[+] Possible entry point: pyiboot01_bootstrap.pyc
[+] Possible entry point: babyRe.pyc
[+] Found 76 files in PYZ archive
[+] Successfully extracted pyinstaller archive: babyRe.exe

You can now use a python decompiler on the pyc files within the extracted directory

C:\Users\86198\Desktop\challenge\isctf2023出题人\babyRe\解题验证\新建文件夹\解题>
```

得到下面这些内容。我们着重注意 struct.pyc 和 babyRe.pyc 文件



载入 010，对比 struct.pyc，看 babyRe.pyc 是否丢失文件头。对比之后没有。如果使用 pyinstxtractor1.0 就可能出现 babyRe.pyc 文件没有第一行的那些东西，那样的话需要手动添加。

010 Editor - C:\Users\86198\Desktop\challenge\isctf2023出题人\babyRe\解题验证\新建文...

文件(F) 编辑(E) 搜索(S) 视图(V) 格式(O) 脚本(I) 模板(L) 调试(D) 工具(T) 窗口(W) 帮助(H)

起始页 babyRe.pyc struct.pyc x

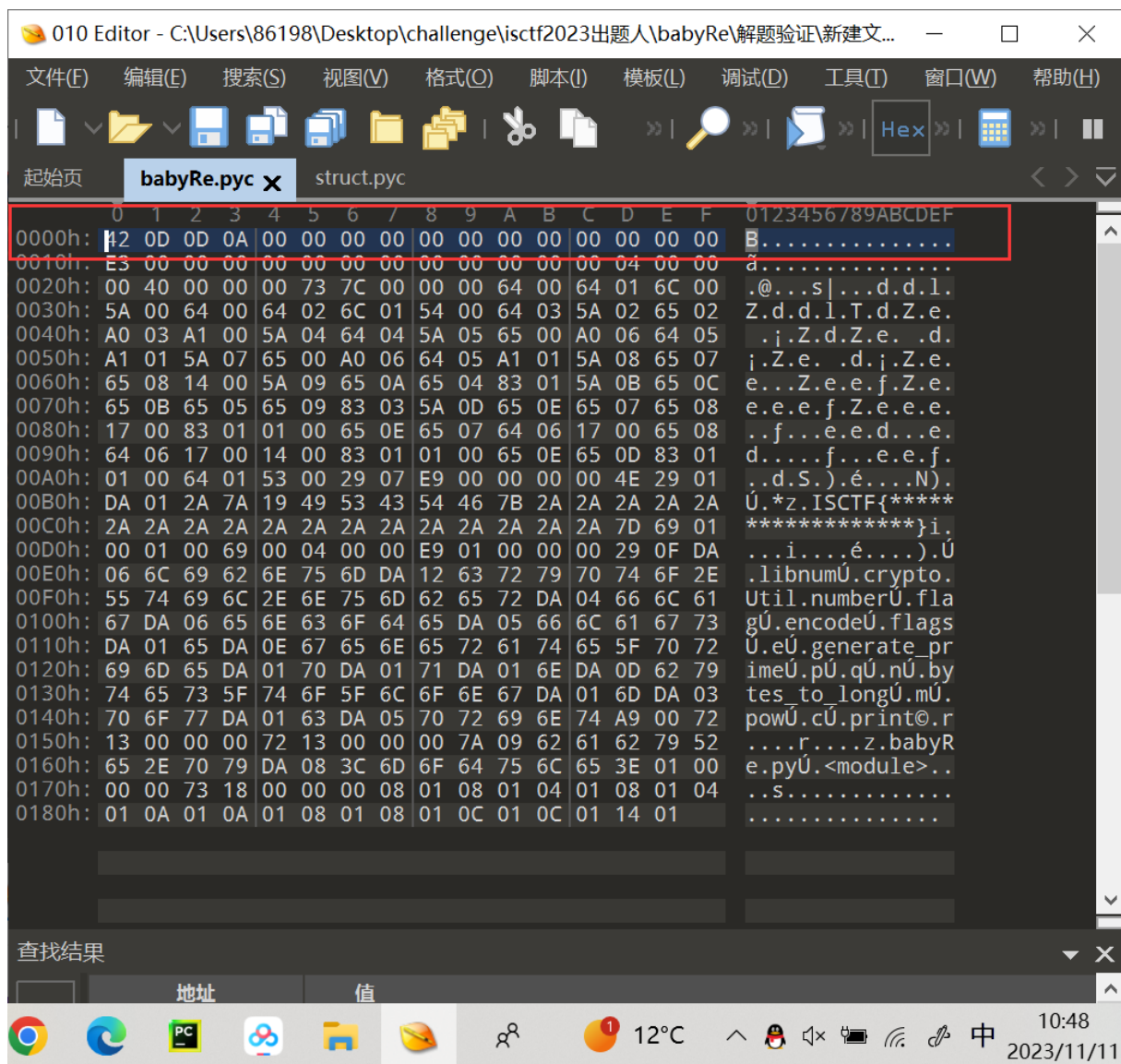
Hex

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000h:	42	0D	0D	0A	00	00	00	00	00	00	00	00	00	00	00	00	B
0010h:	E3	00	00	00	00	00	00	00	00	00	00	00	00	08	00	00	a
0020h:	00	40	00	00	00	73	38	00	00	00	64	00	64	01	64	02	.	@	.	.	.	s
0030h:	64	03	64	04	64	05	64	06	64	07	67	08	5A	00	64	08	d
0040h:	64	09	6C	01	54	00	64	08	64	0A	6C	01	6D	02	5A	02	d
0050h:	01	00	64	08	64	0B	6C	01	6D	03	5A	03	01	00	64	0C
0060h:	53	00	29	0D	DA	08	63	61	6C	63	73	69	7A	65	DA	04	S
0070h:	70	61	63	6B	DA	09	70	61	63	6B	5F	69	6E	74	6F	DA	pack	U
0080h:	06	75	6E	70	61	63	6B	DA	0B	75	6E	70	61	63	6B	5F	.	unpack	U
0090h:	66	72	6F	6D	DA	0B	69	74	65	72	5F	75	6E	70	61	63	from	U
00A0h:	6B	DA	06	53	74	72	75	63	74	DA	05	65	72	72	6F	72	k	U
00B0h:	E9	00	00	00	00	29	01	DA	01	2A	29	01	DA	0B	5F	63	e
00C0h:	6C	65	61	72	63	61	63	68	65	29	01	DA	07	5F	5F	64	lear	cache
00D0h:	6F	63	5F	5F	4E	29	04	DA	07	5F	5F	61	6C	6C	5F	5F	oc	_	N
00E0h:	DA	07	5F	73	74	72	75	63	74	72	0B	00	00	00	72	0C	U
00F0h:	00	00	00	A9	00	72	0F	00	00	00	72	0F	00	00	00	7A
0100h:	09	73	74	72	75	63	74	2E	70	79	DA	08	3C	6D	6F	64	.	struct
0110h:	75	6C	65	3E	03	00	00	00	73	0C	00	00	00	0A	01	02	ule	>
0120h:	03	02	03	06	03	08	01	0C	01							

查找结果

地址	值
----	---

12°C 10:47 2023/11/11



然后，就是把 pyc 文件反编译为可读的 py 文件。其实可以发现这个 pyc 文件是 python3.7 版本的，对于 3.8 版本以上的 pyc 文件，在线网站和 uncompyle6 反编译可能会不成功。

这里就有多种方法反编译，可以在线网站反编译也可以直接 uncompyle6 去反编译。命令就是

```
uncompyle6 -o babyRe.py babyRe.pyc
```

两种方法都可以获得源码。

```
C:\Users\S6198\Desktop\challenge\isctf2023出题人\babyRe\解题验证\新建文件夹\解题>uncompyle6 -o babyRe.py babyRe.pyc
babyRe.pyc --
# Successfully decompiled file
C:\Users\S6198\Desktop\challenge\isctf2023出题人\babyRe\解题验证\新建文件夹\解题>
```

```
# uncompyle6 version 3.9.0
# Python bytecode version base 3.7.0 (3394)
# Decompiled from: Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)]
# Embedded file name: babyRe.py
import libnum
from crypto.Util.number import *
flag = 'ISCTF{*****}'
flags = flag.encode()
e = 65537
p = libnum.generate_prime(1024)
q = libnum.generate_prime(1024)
n = p * q
m = bytes_to_long(flags)
c = pow(m, e, n)
output = open('output.txt', 'w')
output.write('p+q =' + str(p + q) + '\n')
output.write('(p+1)*(q+1)= ' + str((p + 1) * (q + 1)) + '\n')
output.write('c=' + str(c) + '\n')
output.close()
```

babyRe.pyc, 文件大小 : 532 B

反编译结果

```
1 # uncompyle6 version 3.9.0
2 # Python bytecode version base 3.7.0 (3394)
3 # Decompiled from: Python 3.6.12 (default, Feb  9 2021, 09:19:15)
4 # [GCC 8.3.0]
5 # Embedded file name: babyRe.py
6 import libnum
7 from crypto.Util.number import *
8 flag = 'ISCTF{*****}'
9 flags = flag.encode()
10 e = 65537
11 p = libnum.generate_prime(1024)
12 q = libnum.generate_prime(1024)
13 n = p * q
14 m = bytes_to_long(flags)
15 c = pow(m, e, n)
16 output = open('output.txt', 'w')
17 output.write('p+q =' + str(p + q) + '\n')
18 output.write('(p+1)*(q+1)= ' + str((p + 1) * (q + 1)) + '\n')
19 output.write('c=' + str(c) + '\n')
20 output.close()
21
```

```
output.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
292884018782106151080211087047278002613718113661882871562870811030
932129300110050822187903340426820507419488984883216665816506575312
384940488196435920320779296487709207011656728480651848786849994095
965852212548311864730225380390740637527033103610408592664948012814
290769567441038868614508362013860087396409860
212927890731602272957683197809979769913009236844149914320300773130
417623141447100937804683526164480475343392083245180897272107648436
551825159553593098136002869498872189165183463912881519545796929121
057877806041372763009570468994607966518559831546165837090959215326
393713110996596978348870645103513195319024333558336047526387571321
291367044581197672797767125168253797228370053809656868172297712526
937365343970632018808260102739307617676504386383950194111199791493
372607769652471447059159516746974255062368015954771594323698623773
783064618096698857646895260960876356352476583967806719766177168016
60025870405374520076160
520300554236132378034010366202314446850116178818393075997592479039
409799936706294460222859059805319400560149715418370060461464898095
895364359673251063546023336351720680326705497650605849559296478186
894361799224580846395795716110080015593610992834080875511209165161
925838520668403806360086466993445143963741056870047005736255404533
483609801330822851817590111323543625799839740138951192628873975926
808025137778235677962461654696623721373753525274892604208620360086
025155707444068587935416986620649096233120323401951648570096422792
466845218197596135291430435773176908138240694075026081754729955270
5287482926593175925396

第 1 行, 第 1 列    100%    Windows (CRLF)    UTF-8
```

观察题目 py 文件可以知道，这是一道 RSA，但是题目给出的是 $p+q$ 和 $(p+)(q+1)$

推导公式：

令 $x=p+q$

令 $y=(p+1)*(q+1)$

y 展开: $pq+p+q+1$, 也就是 $n+x+1$

那么, $n=y-x-1$

$\text{phi}=(p-1)*(q-1)=pq-p-q+1=n-x+1$

即 $\text{phi}=n-x+1$

解题脚本：

```
import libnum
from crypto.Util.number import long_to_bytes
import gmpy2

x =
29288401878210615108021108704727800261371811366188287156287081103
09321293001100508221879033404268205074194889848832166658165065753
12384940488196435920320779296487709207011656728480651848786849994
09596585221254831186473022538039074063752703310361040859266494801
2814290769567441038868614508362013860087396409860

y =
21292789073160227295768319780997976991300923684414991432030077313
04176231414471009378046835261644804753433920832451808972721076484
36551825159553593098136002869498872189165183463912881519545796929
12105787780604137276300957046899460796651855983154616583709095921
53263937131109965969783488706451035131953190243335583360475263875
71321291367044581197672797767125168253797228370053809656868172297
71252693736534397063201880826010273930761767650438638395019411119
97914933726077696524714470591595167469742550623680159547715943236
98623773783064618096698857646895260960876356352476583967806719766
17716801660025870405374520076160

c =
52030055423613237803401036620231444685011617881839307599759247903
94097999367062944602228590598053194005601497154183700604614648980
95895364359673251063546023336351720680326705497650605849559296478
18689436179922458084639579571611008001559361099283408087551120916
51619258385206684038063600864669934451439637410568700470057362554
04533483609801330822851817590111323543625799839740138951192628873
97592680802513777823567796246165469662372137375352527489260420862
03600860251557074440685879354169866206490962331203234019516485700
96422792466845218197596135291430435773176908138240694075026081754
7299552705287482926593175925396

e = 65537
n = y-x-1
phi = n-x+1
d = gmpy2.invert(e,phi)
m = pow(c,d,n)
print(long_to_bytes(m))
```



```
python babyRe解题脚本.py
C:\Users\86198\AppData\Local\Microsoft\Windows\
b'ISCTF{kisl-iopa-qdnc-tbfs-ualv}'

进程已结束 退出代码0
```

ISCTF{kisl-iopa-qdnc-tbfs-ualv}