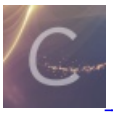
SWPU新生赛2021 Crypto部分WriteUp

原创

是Mumuzi 于 2021-10-11 18:15:25 发布 616 收藏 4

分类专栏： NSSCTF ctf 文章标签： 网络安全

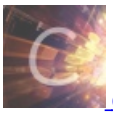
本文链接：https://blog.csdn.net/qq_42880719/article/details/120581904



NSSCTF 同时被 2 个专栏收录

6 篇文章 1 订阅

订阅专栏



ctf

75 篇文章 28 订阅

订阅专栏

第一波放题

crypto1

直接谷歌搜e1e2 = 3087 n = p*q，找到striving的博客，不愧是密码神

http://www.zbc53.top/archives/148/

```
c1= 46363407097182144969801282763157266530258921386852149185503896687900578439730938992292683802859812279518758436135914276165261995827309439842031492707300803108837589295717328091590430994971684215224980648602792013660324845494673796165025264166856262631003598334301870537007778387904758458281727121551759953127850730010456401114222994216038056352729138826083274980872747029133190290251819693292812810706711719870720962016990657579137379385477399956406053612139059368744988498893652236933173819952270026111649696586387068229585895795266153189447760395374249452663284139633838887919827091352357298057444079354357175727802053356562828571435881508330348909652431816407188813941243611296384561998151106123100161740681505698663468097514235219747602457580951497885703447768844323026376172903979785969794745481055100910803145729416484061115752471917334325948588108925293866445663767333736242444315001396118161944126792698184800910746657631468596147874835238845211404211589224327251424508160460779824381758673754666305973734468713088186135742308444802795989340244530329908960608193104121703595514393956745678210720344789834528473103815037772244732920207837587054152953984005141575943608338440820365961331353509434377223869139344747536480617159494",
c2= 13095953427570445321628233481503464726587563278179875090162777382681265733927436240624629792541129182219319148340984732331511039372902070052694671278679338099167500812856186363108109522222628578841297036251839875742370521611231353315539031520487551664545937062970627787621165675324798428237973185077044797853785507037932493528278932742862525994525006677404965095146504370008895896576205441861583804934072463937335124893349435559193423636050677849674105106415677109279800511253416205016509543006500082791609689340856975108555037962055828294225460697881903388553922141633584831908205480614885942771314428677751625172447431961396032779964372327820596925363651468475740905900334822915134120045178528839559648456348026121296311407106497955981232758247467481222526061675709989089690034000799058550147048476275236273496829753253365484619090057101763595938588394585833499588434176790561956750534175204758973181586848929569057410975882502138669844067061136112717089668901510843240849076372359467329947233606557530168105558308454784773316880103019126212213036968749723695976036687410604380154249339222742489092559573415048758675748430460994582792576238288959274370968248522926760477194453546955786012087849132998479244859710725632578334690404408",
n= 60930563709965447888275488090563812312491836411617305087486470099616509677623315552427741813267972785770273804378658838057748549057559102993015271882807597600007897198792210764553032335652512649656242349156336583649175347684079580404021901388096953915444438731302952256545689796220081702142370420407713300336114066003832745805789876485787264537723687075969158800966604718768565429767898743576905176212038853786849378977376668834772490391179674112423747682345250545070498945526007783382866055213071479488920829193905540629247684519448952521212963517328430178214161787848374078853299849240310132479572686586666178674034586263191679320803725027737694204690589234221366319775501031506099087114391938428330292546930977776998979819791304881394074748808719169790362466941577419802706399705870121712464008207478959159149410672685737672875966307473404075543862337268376285695888882637315181591462126286275049707824536968037803899542562846772841295339235909077573444067187438790572408322624658792471622651263167178659161158677494715665717865434309212311725537295479813126556630131603341431171209291349277498904805765062780199127786296317396135508808241909184856967568605858138354287798297969723582920644208778692793974580401745524431530511843 7",

from gmpy2 import *
from Crypto.Util.number import *
E = 3087
fac = [3, 3, 7, 7, 7]

factor = [3, 7, 3 * 3, 3 * 7, 7 * 7, 3 * 3 * 7, 3 * 7 * 7, 7 * 7 * 7, 3 * 7 * 7 * 7, 3 * 3 * 7 * 7]
for e1 in factor:
    assert E % e1 == 0
    e2 = E // e1
    g = gcd(e1, e2)
    print(g)
    _, s, t = gcdext(e1, e2)
    M = pow(c1, s, n) * pow(c2, t, n) % n
    for k in range(1000000):
        a = iroot(M + k * n, g)
        if a[1]:
            print(long_to_bytes(a[0]))
            break

#striving yyds!
#http://www.zbc53.top/archives/148/
```

> NSSCTF{d64dba66-b608-4255-b888-0b0f25c2f90e}

crypto2

共模攻击,直接套脚本

```
import gmpy2
import binascii
c1= 10015622147691092239350487036913994273203989948571504455391374334706588315913651378864948684177454427139669077827459179220005261466923548567553465335859636653507380230136139100732552097504332142397992456027276257982323378767168866941862250266350779664023382968948404453982900805868607584576297657345727814280
c2= 86203582128388484129915298832227259690596162850520078142152482846864345432564143608324463705492416009896246993950991615005717737886323630334871790740288140033046061512799892371429864110237909925611745163785768204802056985016447086450491884472899152778839120484475953828199840871689380584162839244393022471075
e1= 3247473589
e2= 3698409173
n= 10360670682981172015130996577767051960111287771331843539810327809934472545959722106486708995086712589254599750353155604861096884792630732203311732861470143210008457495370625977371141285336446395070346814279139012967109783487137112574156443471015119096238921389827002527291376106707839130888099559421800911031 3

s = gmpy2.gcdext(e1,e2)# 扩展欧几里得算法
m1 = gmpy2.powmod(c1,s[1],n)
m2 = gmpy2.powmod(c2,s[2],n)

m = (m1*m2)%n
print(binascii.unhexlify(hex(m)[2:]))
```

> NSSCTF{xxxxx******xxxxx}

crypto3

在striving的wp里找到这篇

http://www.zbc53.top/archives/41/

[NPUCTF2020]共 模 攻 击这道题

可以发现题目也是e1,e2 = q,p，而且那道题的hint是m<400

这里print了一下flag，发现才308

于是套一下脚本

sagemath使用

```
from Crypto.Util.number import *
c1= 17893542812755845772427795161304049467610774531005620109503081344099161906017295486868699578946474114607624347167976713200068059018517606363517478396368430072890681401898145302336139240273132723451063402106360810413024642916851746118524166947301681245568333254648265529408446609050354235727237078987509705857
c2= 95580409405085606847879727622943874726633827220524165744517624606566789614499137069562997931972825651309707390763700301965277040876322904891716953565845966918293178547100704981251056401939781365264616997055296773595435626490578886752446381493929807909671245959154990639046333135728431707979143972145708806954
n= 14045732358382416033898931768969810273834106196776815387964650542235854472060747614097706405362900576455133908212033722367233097929837365376678262097345409550748411856588488562332875164866037989459206343692490389498699474639450853972145935520018408947097777207572031948283992385697916631970047434904232689897 1

PR.<m> = PolynomialRing(Zmod(n))
f = m^2-(c1+c2)*m+c1*c2
x0 = f.small_roots(X=2^400)
print(x0)
```

```
sage: from Crypto.Util.number import *
....: c1= 178935428127558457724277951613040494676107745310056201095030
....: 746118524166947301681245568333254648265529408446609050354235727 2
....: c2= 955804094050856068478797276229438747266338272205241657445176
....: 886752446381493929807909671245959154990639046333135728431707979 1
....: n= 140457323583824160338989317689698102738341061967768153879646 5
....: 539721459355200184089470977772075720319482839923856979166319700 4
....:
....: PR.<m> = PolynomialRing(Zmod(n))
....: f = m^2-(c1+c2)*m+c1*c2
....: x0 = f.small_roots(X=2^400)
....: print(x0)
[19205354080073978342363933748920570676698656099634958455 01]
sage:
```


然后python再n2s

```
import libnum
print(libnum.n2s(19205354080073978342363933748920570676698656099634958455 01))
```

得到flag

> NSSCTF{why_gongmo_again}

crypto4

q = next_prime(p)可知，q和p相近，于是用yafu分解n
得到q和p，之后就是基操

```
import gmpy2
import binascii
c = 10227915341268619536932290456122384969242151167487654201363877568935534996454863939953106193665663567559506242151019201314446286458150141991211233219320700112533775367958964780047682920839507351492644735811096995884754664899221842470772096509258104067131614630939533042322095150722344048082688772981180270243
n = 52147017298260357180329101776864095134806848020663558064141648200366079331962132411967917697877875277103045755972006084078559453777291403087575061382674872573336431876500128247133861957730154418461680506403680189755399752882558438393107151815794295272358955300914752523377417192504702798450787430403387076153
p = 72212891714887278276735171395978445348693682894554196959649572390476926999190304058001161338058559681236014332470220900701143318427714175669288099560450 93
q = 72212891714887278276735171395978445348693682894554196959649572390476926999190304058001161338058559681236014332470220900701143318427714175669288099560440 21
e = 65537

L = (p-1)*(q-1)
d = gmpy2.invert(e,L)
m = gmpy2.powmod(c,d,n)

print(binascii.unhexlify(hex(m)[2:]))
```

> NSSCTF{no_why}

crypto5

c很小，考虑低指数攻击，直接套脚本。常见的3试试
①m^3<n,也就是说m^3=c。
②m^3>n，即(m^3+i·n)mod n=c（爆破i）

```
import gmpy2
import binascii
e = 3
c= 25166751653530941364839663846806543387720865339263370907985655775152187319464715737116599171477207047430065345882626259880756839094179627032623895330242655333
n= 13410948148270371321483802303541805256700087058716079693570858469413250739421136365242016093118533228040643729021051209066397763473086403237097740717973194006863453607928452802073998866571320081502134270036992251840696835645573639373894612801397364323522832797117071197968393196485456390498066985066062856141 9
i = 0
while True:
    if gmpy2.iroot((c+i*n),3)[1] == True:
        m = gmpy2.iroot((c+i*n),3)[0]
        break
    i += 1

print(binascii.unhexlify(hex(m)[2:]))
```

> NSSCTF{because_i_like}

crypto6

base16 base32 base64
得到5e110989-dc43-1bd3-00b4-9009206158fe

> NSSCTF{5e110989-dc43-1bd3-00b4-9009206158fe}

crypto7

cmd5解密即可

> NSSCTF{md5yyds}

crypto8

UUencode

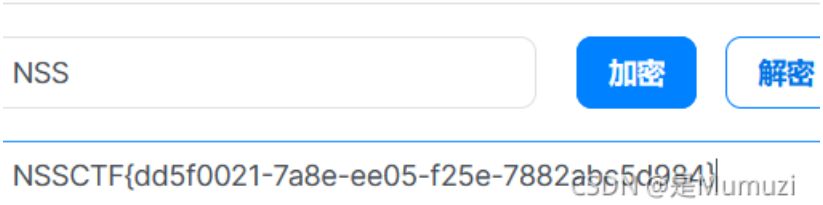
> NSSCTF{cheese_is_power}

crypto9

复制查了一下，是维吉尼亚
去手撸一下
首先密文是AKKPLX{qv5x0021-7n8w-wr05-x25w-7882ntu5q984}
A—N
K—S

第一个是N，然后发现第4 对的上号，说明秘钥长3
后面俩相同，所以只需要试一下，发现是S，再加一个S，正好出秘钥
秘钥就是NSS

AKKPLX{qv5×0021-7n8w-wr05-x25w-7882ntu5q984}



> NSSCTF{dd5f0021-7a8e-ee05-f25e-7882abc5d984}

crypto10

AFFPGS{pbatenghyngvbaf!!!} 凯撒(rot13)

> NSSCTF{congratulations!!!}

第二波放题

ez_caesar

base64解码+caeser

> NSSCTF{youhaveknowcaesar}

ez_rsa

真的基操，爱了

```
import gmpy2
p = 1325465431
q = 152317153
e = 65537
L = (p-1)*(q-1)
d = gmpy2.invert(e,L)
print(d)
```

当然用RSAtools也彳亍，记得md5一下

> NSSCTF{08bb8fb628da85923e5734a75ac19ffe}

pigpig

猪圈密码，对照着翻译就彳亍了
懒得再去翻译，自己翻译哈

traditional

二进制
直接搜八卦图

比如第一个 震坤艮
就是 001 000 100
合起来001000100，转10进制再10进制的ascii转成字符
总的如下：
001000100 001100001 000110000
000110001 001110011 001100001
001101100 001101100
68,97,48,49,115,97,108,108

```
s = [68,97,48,49,115,97,108,108]
print('NSSCTF{',end='')
for i in s:
    print(chr(i),end='')
print("}")
```

> NSSCTF{Da01sall}