BUU Crypto解析

原创

LOL哦糯米藕 于 2020-10-20 14:24:34 发布 420 收藏

[GXYCTF2019]CheckIn

打开题目所提供的文件，发现内容有点像base64解密文件，解密后得到

看到解码出来的结果既有数字又有字母符号，这让我想到了rot,

ROT5、ROT13、ROT18、ROT47 编码是一种简单的码元位置顺序替换暗码。此类编码具有可逆性，可以自我解密，主要用于应对快速浏览，或者是机器的读取，而不让其理解其意。

ROT5 是 rotate by 5 places 的简写，意思是旋转5个位置，其它皆同。下面分别说说它们的编码方式：
ROT5：只对数字进行编码，用当前数字往前数的第5个数字替换当前数字，例如当前为0，编码后变成5，当前为1，编码后变成6，以此类推顺序循环。
ROT13：只对字母进行编码，用当前字母往前数的第13个字母替换当前字母，例如当前为A，编码后变成N，当前为B，编码后变成O，以此类推顺序循环。
ROT18：这是一个异类，本来没有，它是将ROT5和ROT13组合在一起，为了好称呼，将其命名为ROT18。
ROT47：对数字、字母、常用符号进行编码，按照它们的ASCII值进行位置替换，用当前字符ASCII值往前数的第47位对应字符替换当前字符，例如当前为小写字母z，编码后变成大写字母K，当前为数字0，编码后变成符号_。用于ROT47编码的字符其ASCII值范围是33－126，具体可参考ASCII编码。

这个跟rot47有点像，于是rot47解码（https://www.qqxiuzi.cn/bianma/ROT5-13-18-47.php）得到

即可提交flag.

[AFCTF2018]Morse

打开题目，发现是一段摩斯密码，进行解密，得到一串十六进制数，将其进行字符转换即可得到flag.

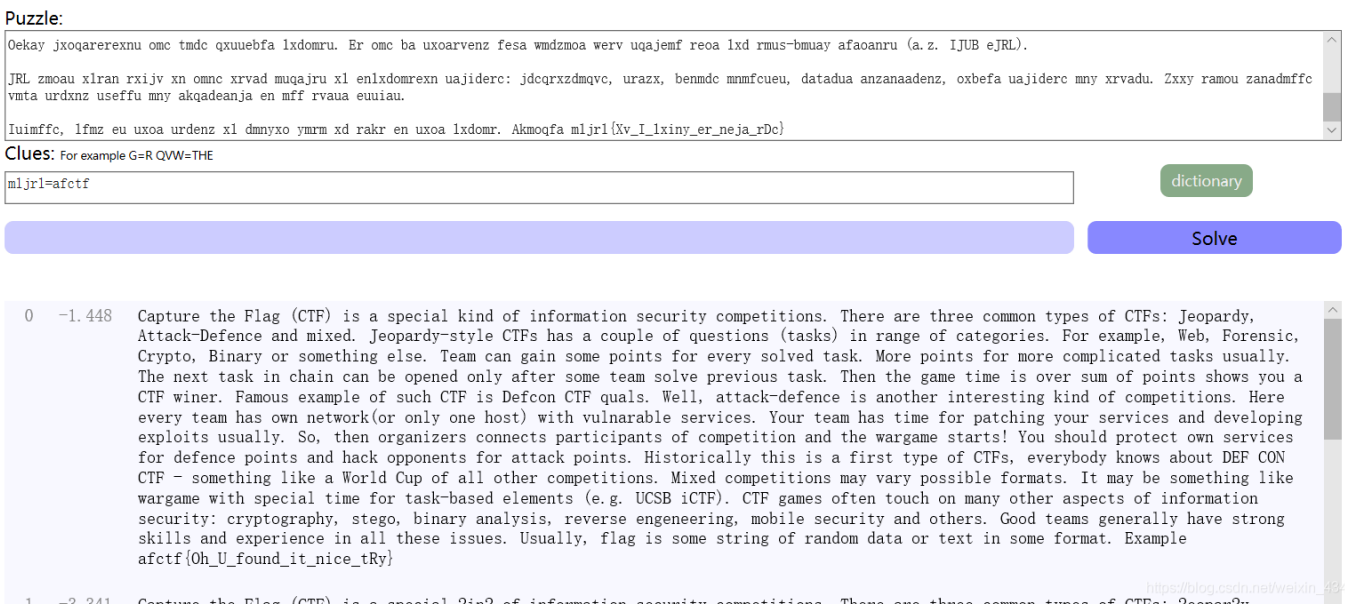
[BJDCTF2020]signin

打开题目，内容是16进制数，于是进行字符转换，得到flag.

[AFCTF2018]Single

文件中有密文，还有.c文件，查看代码，发现数组arr是打乱顺序的，无法逆向，也不能爆破，于是直接将密文中的内容进行词频分析

https://quipqiup.com/



[ACTF新生赛2020]crypto-classic0

打开文件发现需要密码，于是爆破，得到密码为19990306

输入打开文件是一个.c文件，对代码进行稍微的修改

```c
#include<stdio.h>

char flag[25] = "Ygvd mq[lYate[elghqvakl}";

int main()
{
    int i;
    for(i=0;i<25;i++)
    {

        flag[i] ^= 0x7;
        flag[i] += 3;
        printf("%c",flag[i]);
    }
    return 0;
}
```


得到flag{my_naive_encrytion}

[MRCTF2020]keyboard

文件是这样的

得到的flag用
MRCTF{xxxxxx}形式上叫
都为小写字母

6
666
22
444
555
33
7
44
666
66
3

刚看到这个时并没有什么思路，后来百度看了一下解析，思路一下就通了，就是手机中九键英文格，根据提供的信息，有几个数字，就代表着是九键格中每个格子的英文的第几个。最后得到flag:flag{mobilephone}

[MRCTF2020]天干地支+甲子

，有题目可以知道此题目是一个天干地支60轮回，下载文件，提示了：

得到得字符串用MRCTF{}包裹
一天Eki收到了一封来自Sndav的信，但是他有点迷希望您来解决一下
甲戌
甲寅
甲寅
癸卯
己酉
甲寅
辛丑

找出他们对应的数字，如图：

01 甲子	11甲戌	21甲申	31 甲午	41甲辰	51甲寅
02 乙丑	12 乙亥	22乙酉	32 乙未	42乙巳	52 乙卯
03 丙寅	13丙子	23 丙戌	33丙申	43丙午	53丙辰
04丁卯	14丁丑	24丁亥	34丁酉	44丁未	54丁巳
05戊辰	15戊寅	25戊子	35戊戌	45戊申	55 戊午
06己巳	16己卯	26己丑	36 己亥	46 己酉	56 己未
07庚午	17 庚辰	27庚寅	37 庚子	47 庚戌	57 庚申
08辛未	18 辛巳	28 辛卯	38 辛丑	48 辛亥	58 辛酉
09壬申	19壬午	29 壬辰	39 壬寅	49 壬子	59 壬戌
10癸酉	20癸未	30 癸巳	40 癸卯	50 癸丑	60 癸亥

写代码：

```
a=[11,51,51,40,46,51,38] sum='' for i in a:
sum+=chr(i+60) print(sum)
```

得到flag

[WUSTCTF2020]佛说：只能四天

打开题目有几个文件，看了一下，没有丝毫思路，只有去看看大佬的Writeup,突然茅塞顿开，

先要对文件的题目进行新约佛论禅解码新约佛论禅

然后再进行社会主义核心价值观编码核心价值观

查看解码的内容，发现最后有一段提示doyouknowfence，查阅了一下，发现是栅栏密码，于是在进行栅栏解密

[BJDCTF 2nd]rsa1

在虚拟机上nc 题目提供的信息

打开后看见题目提供了两个公式

p*p+q*q=数值和p-q=数值 以及c 和e

很显然是一个求m得题。首先我们得先解出q和p，解出后后面的步骤就简单了。写了求解代码

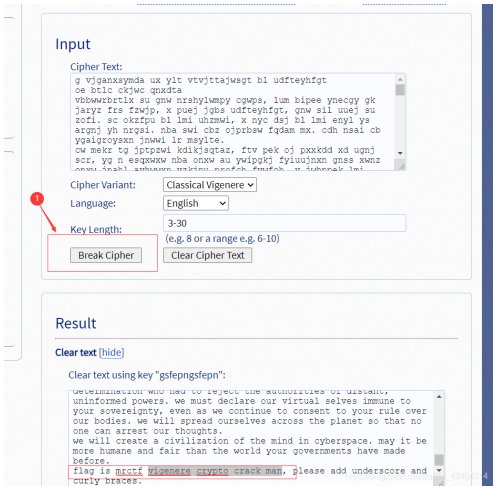
```
from z3 import * s = Solver() p = Int('p') q = Int('q')
s.add(p*p+q*q==145777678964392317179512141517513348955231058105961008209884844697524636074182075918239227919585208644108611626846193547420096576371805786622869202584453951898791631614240781105564972479768818391216085949275904501345106779163039717412004056274052208441906071844431198359257996810241446926750473781120669405938) s.add(p-q==107917896720396272661252633908039963932040144586975502982858189755518922659197708678912323507822928654929258007713418279460540380865054721602181775282717 4)
s.add(p>0) if s.check() == sat: print s.model() import libnum import gmpy2 from Crypto.Util.number import * q = 7980837892497312969272982465935889348875767198495654879554374645126472973151566117482365015476194443329466111649547173320327698324242697674006938644862993 p = 9060016859701275695885508805016288988196168644365409909382956542681662199743543204271488250554423729878758691726681356114933102132893244890028756397690167 c = 6985965861494807953194181513255121902054078424753244708109589557027094278911919053771192943639345960952388130163426722144654807463360762423179813559370712944439680749479744816811236764324251219758318189258999738442250706868834393586542734055430727532290667255667211047140720575028013924528532424228048397555300 e = 10058063 phin = (p-1)*(q-1) n=p*q d=gmpy2.invert(e,phin) # print d m = pow(c,d,n) print long_to_bytes(m)
```

flag{b6d27a35-7ef8-49ef-9a4a-c5f365b34976}

有关于python中z3的详细介绍，建议去z3了解，讲得很通俗易懂

[MRCTF2020]vigenere

是一个维吉尼亚题，下载文件后打开，发现是一堆乱序的单词，还提供了一个py文件，看了py文件，发现写不出代码，只得到网上找在线解密工具，功夫不负有心人，也是找到了vigenere解密，在不知道密钥的前提下，猜测一下密钥的大概范围，解密，在文章的最后也得到了flag



[GWCTF 2019]BabyRSA

文件提供的内容为提个txt文本文件，内容如下：

```
N=6365851495945747469090301601826908662229092564648472917830006518372279213372378996512879435977732709443840348589252957448807271016068414136400065276148731106514101558937765487
m1=900099743414522432169869380283712575286049432089411765187174635547749678781526945864693777652961131656594987260127122886704588843739714198427509292876586402662196866469569298
m2=4874439857574051734266281883756571176042355079369675229932579721088722836983052384544657232142268714142767889120581861970398212429127367428240806276809718025112069143946721 59
```

还提供了一个加密.py脚本；因此我们可以根据题目提供的信息分析其解题思路。

```
import hashlib
import sympy
from Crypto.Util.number import *
flag = 'GWHT{******}'
secret = '*******'
assert(len(flag) == 38)
half = len(flag) / 2
flag1 = flag[:half]
flag2 = flag[half:]
secret_num = getPrime(1024) * bytes_to_long(secret)
p = sympy.nextprime(secret_num)
q = sympy.nextprime(p)
N = p * q
e = 0x10001
F1 = bytes_to_long(flag1)
F2 = bytes_to_long(flag2)
c1 = F1 + F2
c2 = pow(F1, 3) + pow(F2, 3)
assert(c2 < N)
m1 = pow(c1, e, N)
m2 = pow(c2, e, N)
output = open('secret', 'w')
output.write('N=' + str(N) + '\n')
output.write('m1=' + str(m1) + '\n')
output.write('m2=' + str(m2) + '\n')
output.close()
```

代码加密过程说明了flag的字符长度为38位，将flag分为左右两部分，脚本提到了nextprime，指的是下一个质数，说明p,q是两个相邻的质数，那么可以对N进行开根号，有q=sympy.nextprime(p),可知q是p 的下一个质数。再可根据两个表达式求解c1,c2.对c1,c2进行加密即可得m1,m2。代码清楚了，就很清楚思路了，要求出F1,F2首先就得先求出c1,c2,而求解c1,c2的关键是求解p,q.前面已经介绍了该如何求得q,p后面的步骤就可想而知了。

```
import gmpy2 import sympy e = 0x10001
N=gmpy2.mpz(636585149594574746909030160182690866222909256464847291783000651837227921337237899651287943597773270944384034858925295744880727101606841413640006527614873110651410155893776548737823152943797884729130149758279127430044739254000426610922834573094957082589539445610828279428814524313491262061930512829074466232633130599104490893572093943832740301809630847541592548921200288222432789208650949937638303429456468889100192613859073752923812454212239908948930178355331390933536771065791817643978763045030833712326162883810638120029378337092938662174119747687899484603628344079493556601422498405360731958162719296160584042671057160241284852522913676264596201906163)
m1=900099743414522432169869380283712575286049432089411765187174635547749678781526945864693777652961131656594987260127122886704588843739714198427509292876586402662196866469569298721157821730939797429587451216719285687094685260987159271898296004972831180516411073051288526970320533681151812160696266061655034651257252048755787012377892929662118240027614818152766662368690051291388627824764859103086726091860497614883282949955023222414333243193268564781621699870412557822404381213804026685831221430728290755597819259339616650158674713248841654338515199405532003173732520457813901170264713085107077001478083341339002069870585378257051150217511755761491021553239
m2=48744398575740517342662818837565711760423550793696752299325797210887228369830523845446572321422687141427678891205818619703982124291273674282408062768097180251120691439467215924020691073585065199931610001469106729570813863936320359624469399556278028663711639473825077412975902108019732372480541466804231880601065281440507876973854891367546618155100552706530951536495061013720639325714835765966668709166274984856022545382636227170429269284759633953322908803882053208610942115857584107760126871317509787408353624900601894878941323878392284563349402360886525607196285658122989004389693902561360056428339132933145219906285893037456599163419149513793957453954 6
n2=gmpy2.iroot(N,2)[0] p=sympy.nextprime(n2) q=N//p phi=(p-1)*(q-1) d=gmpy2.invert(e,phi) c1=pow(m1,d,N)
c2=pow(m2,d,N) a=3*c1 b=-3*pow(c1,2) c=pow(c1,3)-c2
delta=gmpy2.iroot(pow(b,2)-4*a*c,2)[0] F2=(-b+delta)//(2*a) F1=c1-F2 print(hex(F2)[2:].decode('hex')+hex(F1)[2:].decode('hex'))
```

得flag：GWHT{f709e0e2cfe7e530ca8972959a1033b2}

[WUSTCTF2020]babyrsa

打开题目，给我们提供的内容为c,n,e

可行而知这是要求m，即解密

要求m首先得求出d,而要求得d得先求出p和q，由于题目提供的n不是很大，可以用大数分解(http://www.factordb.com/)直接求出p和q

脚本如下：

```
import gmpy2 from Crypto.Util.number import long_to_bytes c = 28767758880940662779934612526152562406674613203406706867456395986985664083182 e = 65537
p=189239861511125143212536989589123569301 q=3861231253719236511912198698112935864 59
d=gmpy2.invert(e,(p-1)*(q-1))
n=730698867716256428074357836610140626042647684817351458735088469257355216951 59 m=pow(c,d,n) print long_to_bytes(m)
```

[BJDCTF2020]easyrsa

提供的信息为一个脚本，如下：

```
from Crypto.Util.number import getPrime,bytes_to_long
from sympy import Derivative
from fractions import Fraction
from secret import flag
p=getPrime(1024)
q=getPrime(1024)
e=65537
n=p*q
z=Fraction(1,Derivative(arctan(p),p))-Fraction(1,Derivative(arth(q),q))
m=bytes_to_long(flag)
c=pow(m,e,n)
print(c,z,n)
'''
output:
792254786685776145980749150265421628301277617778951154935067295810181028134840228409831014779654943068925380351099487742013553726854941065265447962085869132411036718202564878840
321157486776232096674716228721852750702579247660150200728052673598390593932843165958829333722897321272740764345875193333001424730103446948038851685575488012024959332262154377633
153107451613368954134066900093247662007891792488969519420472354489016123511284593091458255475692984798211012490941618672076865376070474479687087589909501363809247473590525705495
'''
```

思路：

根据代码可以知道几个已知信息

n=p*q

z=p^2+q^2

难点就是要把z的这个公式推导出来，不熟悉的，百度搜索查找到

Fraction(a,b)=a/b;

Derivative(f(x),x):这是用来求导的

同时又知道acrtan和arth的导函数分别为（1/(1+x^2)）、(1/(1-x^2))

反正弦函数的导数公式：$(\arcsin x)' = \frac{1}{\sqrt{1-x^2}}$

反余弦函数的导数公式：$(\arccos x)' = -\frac{1}{\sqrt{1-x^2}}$

反正切函数的导数公式：$(\arctan x)' = \frac{1}{1+x^2}$

反余切函数的导数公式：$(arc\cot x)' = -\frac{1}{1+x^2}$

所以，很容易推导出z=p*p+q*q

然后我们根据n=p*q

联系上面两个公式，推导出p和q

首先设两个参数a,b

其中a=p^2+q^2+2*p*q=z+2*n

b=p^2+q^2-2*p*q=z-2*n

然后将a,b开平方，

p=(sqrt(a)+sqrt(b))//2

q=(sqrt(b)-sqrt(a))//2

后面就很容易了

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

已知了e

我们可以求得d

ed=1 mod phi

m=pow(c,d,n)

根据以上思路可写出py脚本：

```
from Crypto.Util.number import *
from sympy import Derivative
from fractions import Fraction
from gmpy2 import *


c=mpz(792254786685776145980749150265421628301277617778951154935067295810181028134840228409831014779654943068925380351099487742013553726854941065265447962085869132411036718202564
z=mpz(321157486776232096674716228721852750702579247660150200728052673598390593932843165958829333722897321272740764345875193333001424730103446948038851685575488012024959332262154
n=mpz(153107451613368954134066900093247662007891792488969519420472354489016123511284593091458255475692984798211012490941618672076865376070474479687087589909501363809247473590525
pqplus=iroot(z+2*n,2)[0]
pqminus=iroot(z-2*n,2)[0]
p=(pqminus+pqplus)//2
q=(pqplus-pqminus)//2
# p=getPrime(1024)
# q=getPrime(1024)
e=65537
phi=(p-1)*(q-1)
d=invert(e,phi)
m=pow(c,d,n)
#z=Fraction(1,Derivative(arctan(p),p))-Fraction(1,Derivative(arth(q),q))
flag=long_to_bytes(m)
print flag
```

持续更新中