




每天10道Crypto Day3

原创

宁嘉  于 2020-08-17 22:07:59 发布  326  收藏 1

分类专栏: [BUUCTF Crypto](#) 文章标签: [密码学](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/MikeCoke/article/details/107973533>

版权



[BUUCTF Crypto](#) 专栏收录该内容

34 篇文章 2 订阅

订阅专栏

1.救世捷径

2.坏蛋是雷宾

3.[ACTF新生赛2020]crypto-classic0

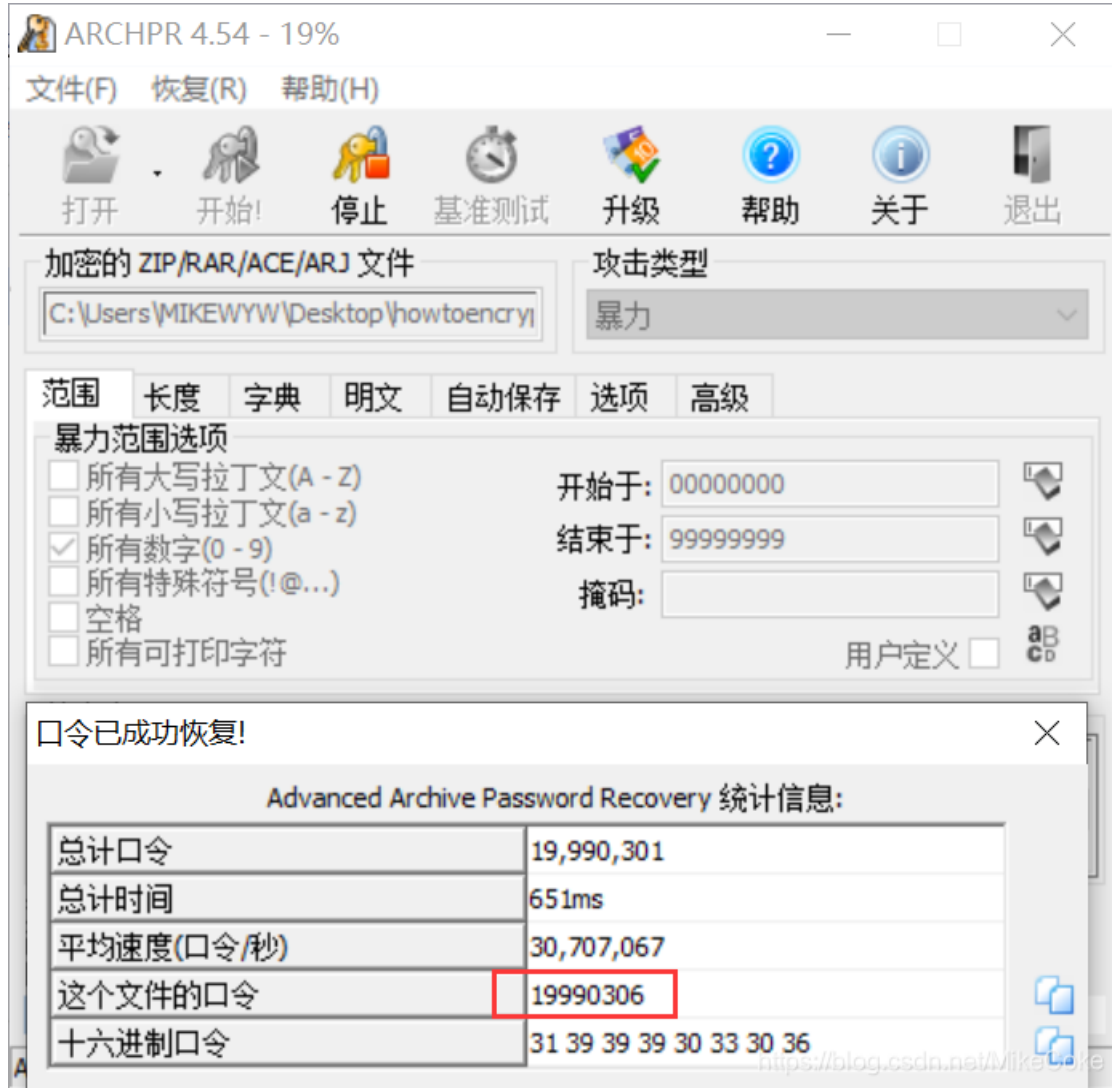
题目

hint.txt : 哼, 压缩包的密码? 这是小Z童鞋的生日吧==

cipher.txt: Ygvd mq[lYate[eIghqvakl}

以及一个howtoencrypt.zip的加密压缩包

题目提示压缩包密码是生日，那么通过爆破 8 位数的年月日。



打开加密文件得到代码

```
#include<stdio.h>

char flag[25] = ***

int main()
{
    int i;
    for(i=0;i<25;i++)
    {
        flag[i] -= 3;
        flag[i] ^= 0x7;    # ^表示异或
        printf("%c",flag[i]);
    }
    return 0;
}
```

解密python脚本

```
ciper = 'Ygvdmq[lYate[eIghqvakl]'\n\nfor i in range(0,23):\n    flag = ord(ciper[i])^0x7\n    flag+=3\n    print(chr(flag),end='')
```

flag{my_naive_encrytion}

4.[NPUCTF2020]这是什么觅

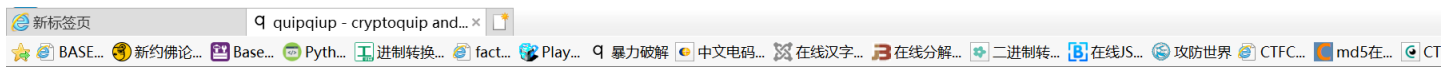
字母代表星期的首字母，其中 S 和 T 都出现两次，所以 S1 代表 SAT，S2 代表 SUN，每组最后一个数字即代表第几行，F1 W1 S2 S21 T12 S11 W1 S13对应得到 3 1 12 5 14 4 1 18，对照字母表 calendar。

flag{calendar}

5.[AFCTF2018]Single

直接词频分析爆破

Jmqrida rva Lfmz (JRL) eu m uqajemf seny xl enlxdomrexn uajiderc jxoqarereXnu. Rvada mda rvdaa jxooxn rcqau xl JRLu: Paxqmdyc, Mrrmjs-Yalanja mny oekay. Paxqmdyc-urcfa JRLu vmu m jxiqfa xl giaureXnu (rmusu) en dmnza xl jmraxzdeau. Lxd akmoqfa, Wab, Lxdanuej, Jdcqrx, Benmdc xd uxoarvenz afua. Ramo jmn zmen uxoa qxenru lxd atadc uxftay rmus. Oxda qxenru lxd oxda jxoqfejmrax rmusu iuimffc. Rva nakr rmus en jvmen jmn ba xqanay xncf mlrad uxoa ramo uxfta qdatexiu rmus. Rvan rva zmoa reoa eu xtad uio xl qxenru uvxwu cxi m JRL wenad. Lmoxiu akmoqfa xl uijv JRL eu Yaljxn JRL gimfu. Waff, mrrmjs-yalanja eu mnxrvad enradaurenz seny xl jxoqarereXnu. Vada atadc ramo vmu xwn narwxds(xd xncf xna vxur) werv tinfmdmbfa uadtejau. Cxid ramo vmu reoa lxd qmrijvenz cid uadtejau mny yatafaqenz akqfXeru iuimffc. Ux, rvan xdzmnehadu jxnnajru qmdrejeqmnrn xl jxoqarereXn mny rva wmdzmoa urmdu! Cxi uvxify qdxrajr xwn uadtejau lxd yalanja qxenru mny vmjs xqxnanru lxd mrrmjs qxenru. Veurxdejmffc rveu eu m ledur rcqa xl JRLu, atadcbxyc snxwu mbxir YAL JXN JRL - uxoarvenz fesa m Wxdfy Jiq xl mff xrvad jxoqarereXnu. Oekay jxoqarereXnu omc tmdc qxuebfax lxdomru. Er omc ba uxoarvenz fesa wmdzmoa werv uqajemf reoa lxd rmus-bmuay afaoanru (a.z. IJUB eJRL). JRL zmoau xlrax rxijv xn omnc xrvad muqajru xl enlxdomrexn uajiderc: jdcqrxzdmqvc, uraxz, benmdc mnmfcueu, datadua anzanaadenz, oxbefax uajiderc mny xrvadu. Zxy ramou zanadmffc vmta urdxnz useffu mny akqadeanja en mff rvaua euuiay. Iuimffc, lfmz eu uxoa urdenz xl dmnyxo ymrm xd rakr en uxoa lxdomr. Akmoqfa mljrl{Xv_l_xiny_er_neja_rDc}



quipqiup **BETA**

quipqiup is a fast and automated cryptogram solver by [Edwin Olson](#). It can solve simple substitution ciphers often found in newspapers, including puzzles like cryptoquips (in which word boundaries are preserved) and patristocrats (inwhi chwor dboun darie saren t).

Puzzle:

Oekay jxoqarereXnu omc tmdc qxuebfax lxdomru. Er omc ba uxoarvenz fesa wmdzmoa werv uqajemf reoa lxd rmus-bmuay afaoanru (a.z. IJUB eJRL). JRL zmoau xlrax rxijv xn omnc xrvad muqajru xl enlxdomrexn uajiderc: jdcqrxzdmqvc, uraxz, benmdc mnmfcueu, datadua anzanaadenz, oxbefax uajiderc mny xrvadu. Zxy ramou zanadmffc vmta urdxnz useffu mny akqadeanja en mff rvaua euuiay. Iuimffc, lfmz eu uxoa urdenz xl dmnyxo ymrm xd rakr en uxoa lxdomr. Akmoqfa mljrl{Xv_l_xiny_er_neja_rDc}

Clues: For example G=R QVW=THE

Already in China? We're hiring

flag{Oh_U_found_it_nice_tRy}

6.[WUSTCTF2020]B@se

看的师傅的博客

```
class base64:
    def __init__(self,alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"):
        self.alphabet = alphabet
    def _EnInsideManage(self, strlist):
        strflag = ""
        temp = ord(strlist[0]) >> 2
        strflag += self.alphabet[temp]
        temp = ((ord(strlist[0])&3)<<4)|(ord(strlist[1])>>4)
        strflag += self.alphabet[temp]
        temp = ((ord(strlist[1])&15)<<2)|(ord(strlist[2])>>6)
        strflag += self.alphabet[temp]
        temp = (ord(strlist[2])&63)
        strflag += self.alphabet[temp]
        return strflag

    def enbase64(self, charString):
        encode = ""
        for i in range(len(charString)//3):
            encode += self._EnInsideManage(charString[i*3:i*3+3])
        if len(charString)%3!=0:
            if len(charString)%3 == 1:
                encode += self._EnInsideManage(charString[-1:]+chr(0)+chr(0))[:2]+"=="
            if len(charString)%3 == 2:
                encode += self._EnInsideManage(charString[-2:]+chr(0))[:3]+'='
        return encode

    def TenToBin(self, tenum):
        binstr = ""
        for i in range(5,-1,-1):
            if 1 == (tenum//(2**i)):
                binstr += '1'
                tenum = tenum%(2**i)
            else:
                binstr += '0'
        return binstr

    def BinToStr(self, strbin):
        "Turn the binary string to a ASCII string"
        strten = ""
        for i in range(len(strbin)//8):
            num = 0
            test = strbin[i*8:i*8+8]
            for j in range(8):
                num += int(test[j])*(2**(7-j))
            strten += chr(num)
        return strten

    def debase64(self, base64string):
        binstr = ""
        for i in base64string:
            binstr += self.TenToBin(self.alphabet.find(i))
        return self.BinToStr(binstr)

from itertools import combinations, permutations
```

```

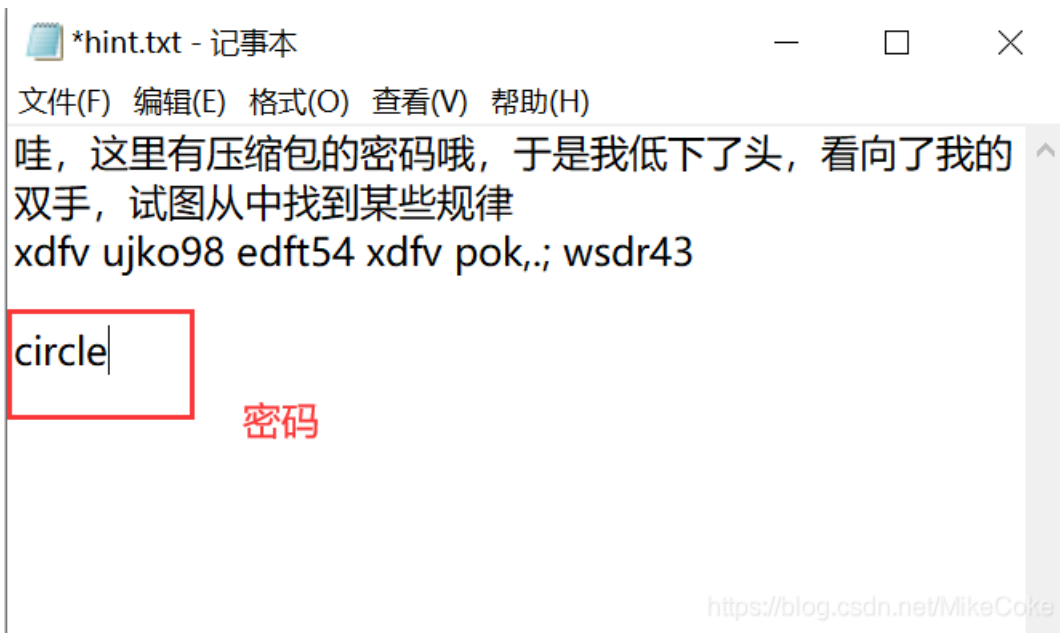
from itertools import combinations, permutations
for i in list(permutations(['3', '4', 'j', 'u'], 4)):
    try:
        password = 'JASGBWcQPRXEFbCDIImnHUVKTYZdMovwipatN0efghq56rs-{}kxyz012789+/' .format(''.join(i))
        print(password)
        newobj = base64(alphabet=password)
        print(newobj.debase64("MyLkTaP3FaA7KOWjTmKkVjWjVzKjdeNvTnAjoH9iZ0IvTeHbvD=="))
    except:
        pass

```

{base64_1s_v3ry_e@sy_and_fuN}

7.[ACTF新生赛2020]crypto-classic1

键盘加密 和 维吉尼亚密码



获取的维吉尼亚密文

```
SRLU{LZPL_S_UASHKXUPD_NXYTFTJT}
```

通过爆破获取 密钥key,我们可以猜到 SRLU的明文是ACTF

所以

```

#破解key python3
s='ABCDEFGHIJKLMNOPQRSTUVWXYZ'
s1='ACTF'
s2='SRLU'
key=''
for i in range(len(s1)):
    key+=s[(s.find(s2[i])-s.find(s1[i]))%26]
print(key)

```

得到 key = sp

CTF在线工具-在线维吉尼亚密... CTF|CTF工具下载|CTF工具包|C... CTFTools - BugKu

BASE... 纽约佛论... Base... Pyth... 进制转换... fact... Play... 暴力破解 中文电码... 在线汉字... 在线分解... 二进制转... 在线JS...

栅栏密码 凯撒密码 凯撒移位(中文版) 维吉尼亚密码 摩斯电码
 百度/Google/网页字符 MD5 置换密码 替代密码

清空 拼音 频率 去空格 每隔 2 个字符 加空格 横/竖 大写 小写 倒序 词倒序

替换 计算 十进制 > 十六进制 转换

维吉尼亚密码

在下面的文本框输入明文或密文，点加密或解密，文本框中即可出现所得结果

加密 解密 位移数(-25~25): 0

密钥:
SPSPSPSSSPSPSPSSSPSPSPS

密文框:
actf{what_a_classical_vigenere}

<https://blog.csdn.net/MikeCoke>

flag{what_a_classical_vigenere}

8.EasyProgram

参考文章

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 file.txt x 无标题1*

编辑方式: 十六进制(H) 运行脚本 运行模板

	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:	00	BA	8F	11	2B	22	9F	51	A1	2F	AB	B7	4B	D7	3F	EF	.	°	.	+	"	Ÿ	Q	j	/	«	·	K	×	?	i	
0010h:	E1	B5	13	BE	C4	D4	5D	03	D9	00	7A	CA	1D	51	A4	73	á	µ	.	¾	Ä	Ö]	.	Û	.	z	Ê	.	Q	α	σ
0020h:	B5	EF	3D	9B	31	B3											µ	i	=	>	1	³										

本地结尾 <https://blog.csdn.net/MikeCoke>

```

key = 'whoami'
s = []
t = []
j = 0

for i in range(256):
    s.append(i)
for i in range(256):
    t.append(key[i % len(key)])

for i in range(256):
    j=(j+s[i]+ord(t[i])) % 256
    s[i],s[j]=s[j],s[i]

i=0
j=0
f = [0x00,0xBA,0x8F,0x11,0x2B,0x22,0x9F,0x51,0xA1,0x2F,0xAB,0xB7,0x4B,0xD7,0x3F,0xEF,0xE1,0xB5,0x13,0xBE,0xC4,0x
D4,0x5D,0x03,0xD9,0x00,0x7A,0xCA,0x1D,0x51,0xA4,0x73,0xB5,0xEF,0x3D,0x9B,0x31,0xB3]
flag = ''

for m in range(38):
    i = (i + 1)% 256
    j = (j + s[i])% 256
    s[i],s[j]=s[j],s[i]

    x = (s[i]+(s[j] % 256)) % 256

    flag += chr(f[m] ^ s[x])

print(flag)

```

flag{f238yu28323uf28u2yef2ud8uf289euf}

9.[BJDCTF2020]Polybius

[参考文章 \(波利比奥斯方阵密码\)](#)


```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
oytnmkvwybsmfk
xsnzwiqrsbpwfi
sxntrkvwxborfk
kpzgislopdwiqs
iotgkxlpoeerkvx
kuzghoqsucwhlo
iytghpvxyrhlp
hsngkyqusemkvy
hxngiuvyxdmiqu
wmgzyslomdkyqs
rmtuxtlpmeiuvx
wrgzxoqsrckxlo
rwgtspxwcislp
mrgnpyqurehpvy
mwgnouvywdhoqu
qfatsmkhfxdspm
vfazxmihfsexom
lfanorkifycour
lfanpwikfucpyw
vfazyrhifoeyrs
qfatuwhkfpduxw
qlatrhpmlwdrkh
vlazwhomlrewih
lqanmiurqwcmki
lvanmkywvrcmik
vqazwisrqmewhi
qvatrkxwvmdrhh
flagispolybius
flagkxonlubkvx
fqaghousqxbhpo
fvaghpyxvsbhop
fqagkysuqpbkxy
fvagiuxyvobisu
vlazysmolieyrs
qlatuxmplkduwx
vqazxorsqhexmo
qvatspwxvhdsmp
lqanpyruqkcpwy
lvanouwyvicoru
>>>
```

得到 `flag{flagispolybius}`

10.[WUSTCTF2020]大数计算

关于宇宙终极问题的答案 x,y,z

$$42 = (-80538738812075974)^3 + 80435758145817515^3 + 12602123297335631^3$$

```
P1 = 1
for i in range(1,2021):
    P1 *=i
...

P2 = (520**1314 + 2333**666)
P3 = 80538738812075974 + 80435758145817515 + 12602123297335631
P4 = (22**2+36)*1314

print(P1)
print('-----')
print(P2)
print('-----')
print(P3)
print('-----')
print(P4)
...

a =38609695
b =67358675
c =17357662
d =683280

print(hex(a)[2:]+'-'+hex(b)[2:]+'-'+hex(c)[2:]+'-'+hex(d)[2:])
```

flag{24d231f-403cfd3-108db5e-a6d10}