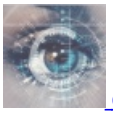
CTF常见编码

转载

周粥粥啊 于 2021-04-14 15:26:36 发布 795 收藏 2

分类专栏： ctf CS 信安基础 文章标签： unicode

原文链接：https://www.cnblogs.com/1go0/p/9977359.html


ctf 同时被 3 个专栏收录

10 篇文章 0 订阅

订阅专栏


CS

10 篇文章 0 订阅

订阅专栏


信安基础

5 篇文章 0 订阅

订阅专栏

原文链接：https://www.cnblogs.com/1go0/p/9977359.html

```
0.md5(只包含数字(0-9)和字母（a-f）,长度为16或32)
注意长度，字符和替换
#utf-7
    密文：+/v+ +ADwAcwBjAHIAaQBwAHQAPgBhAGwAZQByAHQAKAAiAGsAZQB5ADoALwAlAG4AcwBmAG8AYwB1AHMAWABTAFMAdABlAHMAdAAlAC8AIgApADwALwBzAGMAcgBpAHAAdAA+AC0-

    +Axxxxxxxxx- 是 UTF-7 编码的特殊格式。所以只取+ADwAcwBjAHIAaQBwAHQAPgBhAGwAZQByAHQAKAAiAGsAZQB5ADoALwAlAG4AcwBmAG8AYwB1AHMAWABTAFMAdABlAHMAdAAlAC8AIgApADwALwBzAGMAcgBpAHAAdAA+AC0-
    解码得到<script>alert("key:/%nsfocusXSStest%/")</script>-
http://toolswebtop.com/text/process/decode/UTF-7
00.Windows hash
    附件如下
    nick:1003:8EA9109FDA91D6BFC6EBE8776A153FEB:C6002D00F57F9F399B6263D714AF8C3A:::
    Tips：LMHASH，NTHASH，Ophcrack可知这是windows的hash密码
    Windows系统下的hash密码格式为：用户名称:RID:LM-HASH值:NT-HASH值，例如：
    nick:1003:8EA9109FDA91D6BFC6EBE8776A153FEB:C6002D00F57F9F399B6263D714AF8C3A:::
    8EA9109FDA91D6BFC6EBE8776A153FEB:C6002D00F57F9F399B6263D714AF8C3A解码后就是密码
    用户名称为：nick
    RID为：1003
    LM-HASH值为：8EA9109FDA91D6BFC6EBE8776A153FEB
    NT-HASH值为：C6002D00F57F9F399B6263D714AF8C3A
    在http://www.objectif-securite.ch/en/ophcrack.php解密的passwd为hacker521，登陆就能得到flag
    root:$6$HRMJoyGA$26FIgg6CU0bGUOfqFB0Qo9AE2LRZxG8N3H.3BK8t49wGlYbkFbxVFtGOZqVIq3qQ6k0oetDbn2aVzdhuVQ6US.:17770:0:99999:7:::
    用户名root
    RID 6
    LM-HASH值为26FIgg6CU0bGUOfqFB0Qo9AE2LRZxG8N3H
    NT-HASH  3BK8t49wGlYbkFbxVFtGOZqVIq3qQ6k0oetDbn2aVzdhuVQ6US
```

1.ASCII编码

ASCII编码大致可以分作三部分组成：

第一部分是：ASCII非打印控制字符（参详ASCII码表中0-31）；

第二部分是：ASCII打印字符，也就是CTF中常用到的转换；

#例题/：从一个二维码里扫到一串数字

45 46 45 46 32 45 32 46 46 45 46 32 46 45 46 46 32 46 46 46 32 45 46 46 46 32 46 46 45 45 46 45 32 45 46 46 46 32 46 46 46 32 46 45 46 46 32?

看起来像是ascii，根据ascii表解码得到了这些：

EFEF2E2FFEF2FEFF2FFF2EFFF2FFEEFE2EFFF2FFF2FEFF2

如果把2看成是断开，那么就得到

EFEF

E

FFEF

FEFF

FFF

EFFF

FFEEFE

EFFF

FFF

FEFF

看起来很像摩斯码吧，E是—，F是.　　于是解密就得到CTFnullSBnullBSL

2.Base64/32/16编码

编码原理：Base64编码要求把3个8位字节转化为4个6位的字节，之后在6位的前面补两个0，形成8位一个字节的形式，6位2进制能表示的最大数是2的6次方是64，

这也是为什么是64个字符(A-Z,a-z，0-9，+，/这64个编码字符，=号不属于编码字符，而是填充字符)的原因

#如果decode后发现变空白了，说明有问题，这种的一般都是=的问题，尝试增减=，发现加两个=时有变化：

Key1一看就是标准的base全家桶，but 两种base64解出来是乱码啊，base32总是解不出来，这里涉及到python语法（http://zhidao.baidu.com/link?url=d4lCkOsTA0RmtoKdb_-9DI2mnmPIJWPoWMvoL-jRANYzq4cZp1wPPw1yvviZo2jeY6Ki8Ds8B4834YShiTlmr_）

```
>>> a = 'DaoChangcloud01'
>>> import base64
>>> b = base64.b32encode(a)
>>> print b
IRQW6Q3IMFXGOY3MN52WIMBR
>>> c = base64.b32decode(b, True)
>>> print c
DaoChangcloud01
>>> c1='IRQW6Q3IMFXGOY3MN52WIMBr' #这里最后一个字母改成小写测试
>>> base64.b32decode(c1, True)
'DaoChangcloud01'
>>> base64.b32decode(c1, False)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python26\lib\base64.py", line 222, in b32decode
    raise TypeError('Non-base32 digit found')
TypeError: Non-base32 digit found
```

最终decod出来Comeon2012baby 提交后得到key2 :2tHEWinNer

```
    #base混合解密
        import base64
    file = open('base.txt','r')
    st = file.read()
    while True:
       try:
           st = base64.b16decode(st)
       except:
           try:
               st = base64.b32decode(st)
           except:
               st = base64.b64decode(st)
       if(st.find('flag') == 0):
           print(st)
```

```
            print(st)
```

3.shellcode编码

```
#!shell
\x54\x68\x65\x7f\x71\x75\x69\x63\x6b\x7f\x62\x72\x6f\x77\x6e\x7f\x66\x6f\x78\x7f\x6a\x75\x6d\x70\x73\x7f\x6f\x76
\x65\x72\x7f\x74\x68\x65\x7f\x6c\x61\x7a\x79\x7f\x64\x6f\x67
```

4.Quoted-printable 编码

```
#!shell
=E6=95=8F=E6=8D=B7=E7=9A=84=E6=A3=95=E8=89=B2=E7=8B=90=E7=8B=B8=E8=B7=B3=E8
=BF=87=E4=BA=86=E6=87=92=E6=83=B0=E7=9A=84=E7=8B=97
```

5.XXencode编码

它所选择的可打印字符是：+-0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz，一共64个字符。跟base64打印字符相比，就是XXencode多一个“-”字符，少一个”/”字符。

6.UUencode编码

UUencode所产生的结果刚好落在ASCII字符集中可打印字符（32-空白…95-底线）\

源文本： The quick brown fox jumps over the lazy dog

编码后： M5&AE('%U:6-K(&)R;W=N(&9O>"!J=6UP<R!O=F5R('1H92!L87IY(&1O9PH*

http://www.qqxiuzi.cn/bianma/uuencode.php

7.URL编码

在该字节ascii码的的16进制字符前面加%. 如 空格字符，ascii码是32，对应16进制是'20'，那么urlencode编码结果是:%20。

```
#!shell
%54%68%65%20%71%75%69%63%6b%20%62%72%6f%77%6e%20%66%6f%78%20%6a%75%6d%70%73%20%6f%76%65%72%20%74%68%65%20%6c%61%
7a%79%20%64%6f%67
```

8.Unicode编码

有如下四种编码方式

源文本： The

&#x [Hex]： The

&# [Decimal]： The

\U [Hex]： \U0054\U0068\U0065

\U+ [Hex]： \U+0054\U+0068\U+0065

9.Escape/Unescape编码

Escape/Unescape加密解码/编码解码,又叫%u编码，采用UTF-16BE模式， Escape编码/加密,就是字符对应UTF-16 16进制表示方式前面加%u。Unescape解码/解密，就是去掉"%u"后，将16进制字符还原后，由utf-16转码到自己目标字符。如：字符“中”，UTF-16BE是：“6d93”，因此Escape是“%u6d93”。

源文本： The

编码后： %u0054%u0068%u0065

10.HTML实体编码

详见http://www.w3school.com.cn/tags/html_ref_entities.html

结果	描述	实体名称	实体编号
"	quotation mark	"	"
'	apostrophe	'	'
&	ampersand	&	&
<	less-than	<	<
>	greater-than	>	>

11.敲击码

敲击码(Tap code)是一种以非常简单的方式对文本信息进行编码的方法。因该编码对信息通过使用一系列的点击声音来编码而命名，敲击码是基于5×5方格波利比奥斯方阵来实现的，不同点是是用K字母被整合到C中。

敲击码表：

```
#!shell
  1  2  3  4  5
1 A  B C/K D  E
2 F  G  H  I  J
3 L  M  N  O  P
4 Q  R  S  T  U
5 V  W  X  Y  Z
```

12.莫尔斯电码

摩尔斯电码(Morse Code)是由美国人萨缪尔·摩尔斯在1836年发明的一种时通时断的且通过不同的排列顺序来表达不同英文字母、数字和标点符号的信号代码，摩尔斯电码主要由以下5种它的代码组成：

点（.）
划（-）
每个字符间短的停顿（通常用空格表示停顿）
每个词之间中等的停顿（通常用 / 划分）
以及句子之间长的停顿
摩尔斯电码字母和数字对应表：

```
#!shell
A  .-    N  -.    .  .-.-.-  +  .-.-.    1  .----
B  -...  O  ---   ,  --..--  _  ..--.-   2  ..---
C  -.-.  P  .--.  :  ---...  $  ...-..-  3  ...--
D  -..   Q  --.-  "  .-..-.  &  .-...    4  ....-
E  .     R  .-.   '  .----.  /  -..-.    5  .....
F  ..-.  S  ...   !  -.-.--             6  -....
G  --.   T  -     ?  ..--..             7  --...
H  ....  U  ..-   @  .--.-.             8  ---..
I  ..    V  ...-  -  -....-             9  ----.
J  .---  W  .--   ;  -.-.-.             0  -----
K  -.-   X  -..-  (  -.--.
L  .-..  Y  -.--  )  -.--.-
M  --    Z  --..  =  -...-
```

---------- 表达的意思是调用信号，表示“我有消息发送”

#敲击码

敲击码(Tap code)是一种以非常简单的方式对文本信息进行编码的方法。因该编码对信息通过使用一系列的点击声音来编码而命名，敲击码是基于5×5方格波利比奥斯方阵来实现的，不同点是是用K字母被整合到C中。

#各种文本加密

文本加密可以将正常文本内容打乱为不可连读的文字或符号(汉字 数字 字母 音乐符号 国际音标 盲文 韩文 日文 傣文 彝文 箭头符号 花朵符号 俄文)，换行等格式信息也会被清除，达到加密的作用。在进行文本加密时可以设定一个密码，这样只有知道密码的人才能解密文本。密码可以是数字、字母和下划线，最多九位。

http://www.qqxiuzi.cn/bianma/wenbenjiami.php

例如 明文：我爱你

密文：耜懔第煨证喊==

明文：我爱你

密文：???←?↙??==

敲击码表：

```
#!shell
  1  2  3  4  5
1 A  B C/K D  E
2 F  G  H  I  J
3 L  M  N  O  P
4 Q  R  S  T  U
5 V  W  X  Y  Z
```

000000000000换位密码

1.栅栏密码

栅栏密码(Rail-fence Cipher)就是把要加密的明文分成N个一组，然后把每组的第1个字符组合，每组第2个字符组合...每组的第N(最后一个分组可能不足N个)个字符组合，最后把他们全部连接起来就是密文，这里以2栏栅栏加密为例。

明文： The quick brown fox jumps over the lazy dog

去空格： Thequickbrownfoxjumpsoverthelazydog

分组： Th eq ui ck br ow nf ox ju mp so ve rt he la zy do g

第一组： Teucbonojmsvrhlzdg

第二组： hqikrwfxupoeteayo

密文： Teucbonojmsvrhlzdghqikrwfxupoeteayo1

加解密 传送门

2.曲路密码(Curve Cipher)是一种换位密码，需要事先双方约定密钥(也就是曲路路径)。

明文： The quick brown fox jumps over the lazy dog

填入5行7列表(事先约定填充的行列数)

3.列移位密码

很有意思~~，见灵哥的文章

000000000000其他加密

000000000000替换加密

1.埃特巴什码

埃特巴什码(Atbash Cipher)是一种以字母倒序排列作为特殊密钥的替换加密，也就是下面的对应关系：

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ZYXWVUTSRQPONMLKJIHGFEDCBA

明文： the quick brown fox jumps over the lazy dog

密文： gsv jfrxp yildm ulc qfnkh levi gsv ozab wlt

加解密 http://www.practicalcryptography.com/ciphers/classical-era/atbash-cipher/

2.凯撒密码

凯撒密码(Caesar Cipher或称恺撒加密、恺撒变换、变换加密、位移加密)是一种替换加密，明文中的所有字母都在字母表上向后（或向前）按照一个固定数目进行偏移后被替换成密文。例，当偏移量是3的时候，所有的字母A将被替换成D，B变成E，以此类推，更多 参考 。

加解密http://planetcalc.com/1434/

也可使用米斯特的小工具哦~~

vbkq{ukCkS_vrduztucCVQXVuvzuckrvtZDUBTGYSkvcktv}

发现是凯撒加密，不过奇偶数移位方向不一样，发现偏移量是16，用脚本跑一下

```
str = "vbkq{ukCkS_vrduztucCVQXVuvzuckrvtZDUBTGYSkvcktv}"
for i in range(26):
    key = ''
    for x in str:
        s = ord(x)
        if (s not in range(97,123)) and (s not in range(65,91)):
            key = key + chr(s)
        else:
            #print chr(s)
            if s in range(97,123):
                if s % 2 == 0:
                    s = s - i
                    if s not in range(97,123):
                        t = 97-s
                        t = 123-t
                        key = key + chr(t)
                    else:
                        key = key + chr(s)
                else:
                    s = s + i
                    if s not in range(97,123):
                        t = s-122+96
                        key = key + chr(t)
                    else:
                        key = key + chr(s)
            else:
                #print chr(s)
                if s % 2 == 0:
                    s = s - i
                    if s not in range(65,91):
                        t = 65-s
                        t = 91-t
                        key = key + chr(t)
                    else:
                        key = key + chr(s)
                else:
                    s = s + i
                if s not in range(65,91):
                        t = s-90+64
                        key = key + chr(t)
                else:
                        key = key + chr(s)
    print key
```

3.ROT5/13/18/47

ROT5/13/18/47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。

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，下面以rot13以例。

明文： the quick brown fox jumps over the lazy dog

密文： gur dhvpx oebja sbk whzcf bire gur ynml qbt

加解密http://www.qqxiuzi.cn/bianma/ROT5-13-18-47.php

4.简单替换密码

简单换位密码(Simple Substitution Cipher)加密方式是以每个明文字母被与之唯一对应且不同的字母替换的方式实现的，

它不同于恺撒密码，因为密码字母表的字母不是简单的移位，而是完全是混乱的。 比如：

#!shell

明文字母 : abcdefghijklmnopqrstuvwxyz

明文字母 : phqgiumeaylnofdxjkrcvstzwb

明文： the quick brown fox jumps over the lazy dog

密文： cei jvaql hkdtf udz yvoxr dsik cei npbw gdm

（2）破解

当密文数据足够多时这种密码我们可以通过字频分析方法破解或其他方法破解，

比较好的在线词频分析网站 http://quipqiup.com/index.php (翻= =墙)，

这里推荐一篇通过"爬山算法"来破解简单替换密码 文章 ，基于文中的算法实现的工具来破解示例。

5.希尔密码--考察你的线性代数咯

解码 http://www.practicalcryptography.com/ciphers/hill-cipher/

#希尔密码

密文： 22,09,00,12,03,01,10,03,04,08,01,17 （c）

使用的矩阵是 1 2 3 4 5 6 7 8 10

请对密文解密.

一个希尔密码 老老实实写个Python解密

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
__Url__ = 'Http://www.purpleroc.com'
__author__ = 'Tracy_梓朋'
from numpy import *
Dic = {chr(i+97):i for i in range(26)}
def decode(pwd, org):
    temp = []
    result = []
    while True:
        if len(pwd) % 3 != 0:
            pwd.append(pwd[-1])
        else:
            break
    for i in pwd:
        temp.append(Dic.get(i))
    temp = array(temp)
    temp = temp.reshape(len(pwd)/3, 3)
    #print temp
    #print org
    xx = matrix(temp)*org
    for j in range(len(pwd)/3):
        for i in range(3):
            if (int(xx[j, i]) >= 26):
                result.append(chr(xx[j, i] % 26 + 97))
                #print xx[j, i] % 26
            else:
                #print xx[j, i]
```

```
                result.append(chr(xx[j, i] + 97))
    return result
def get_vmatrix(org):
    org_adjoin = org.I*linalg.det(org)
    print org_adjoin
    org_det = int(str(abs(linalg.det(org))).split('.')[0])
    print org_det
    for i in range(1, 26):
        if i * org_det % 26 == 1:
            break
    org_mod = -org_adjoin * i % 26
    org_mod = matrix(org_mod)
    temp = []
    for i in range(org_mod.shape[0]):
        for j in range(org_mod.shape[1]):
            temp.append(int(str(org_mod[i, j]).split('.')[0]))
    org_final = matrix(temp).reshape(org_mod.shape[0], org_mod.shape[1])
    #print org_final
    return org_final
if __name__ == '__main__':
    ''' for test
    pwd = list("act")
    org = matrix(array([[6, 24, 1], [13 , 16, 10], [20, 17, 15]]))
    result = decode(pwd, org)
    print "".join(result)
    deorg = matrix(array([[8, 5, 10], [21 , 8, 21], [21, 12, 8]]))
    result = decode(result, deorg)
    print "".join(result)
    '''
    pwd = "wjamdbkdeibr" #密文
    pwd = list(pwd)
    org = matrix(array([[1,2,3],[4,5,6],[7,8,10]])) #解密矩阵
    org_vm = get_vmatrix(org)
    print org_vm
    print "Your flag is :" + "".join(decode(pwd, org_vm))
```

6.猪圈密码(Pigpen Cipher或称九宫格密码、朱高密码、共济会密码或共济会员密码)，是一种以格子为基础的简单替代式密码。可以使用米斯特小工具试试，如果不行的话可以再试试其他变种

7.VBscript.Encode http://adophper.com/encode.html

#@~^TgAAAA==\'[6*liLa6++p'aXvfiLaa 6i[[avWi[[a *p[[6*!I'[6 cp'aXvXILa6 fp[:6+Wp[:XvWi[[6+XivRIAAA==^#~@

解完密之后是这样

Encode@decode

这个是网页用的一种不能算编码的东西，控制台可解，还有就是发布到百度也会自动转成ascii

控制台document.write("Encode@decode")就行了

8.

#小明某一天收到一封密信，信中写了几个不同的年份：辛卯，癸巳，丙戌，辛未，庚辰，癸酉，己卯，癸巳。信的背面还写有“+甲子”，请解出这段密文。key值：CTF{XXX}

从题目可以知道是 “干支数 + 甲子（60）”= 数字（ascii）-->对应的字母

辛卯，为干支之一，顺序为第28个 --> 88 --> X

癸巳，为干支之一，顺序为第30个 --> 90 --> Z

丙戌，为干支之一，顺序为第23个 --> 93 --> S

辛未，为干支之一，顺序为第8个 --> 68 --> D

庚辰，为干支之一，顺序为第17个 --> 77 --> M

癸酉，为干支之一,顺序为第10个 --> 70 --> F

己卯，为干支之一，顺序为第16个 --> 76 --> L

癸巳，为干支之一，顺序为第30个 --> 90 --> Z

古典密码，待解密字符串为： XZSDMFLZ

古典密码以换位、替换为主，试试常见的古典密码：换位栅栏、替换凯撒。

这题是这两种组合，没有一点提示，很尴尬、、、

首先，栅栏密码（两栏）--> 解密：XMZFSLDZ
再用凯撒 --> 解密：SHUANGYU
#
0x1 费纳姆密码
费纳姆密码其实是一种由二进制产生的替换密码。
这是一种二战期间德军使用的密码，一时令盟军难以破译，
其原理主要是把二进制数据每7位分出一组，再和密钥的每一位异或即可得到

```
import re

key = 'large'
s = '00000100001000001101000001100001010'
key = list(key)
s = re.findall('.{7}',s)
flag = ''
for i in range(len(s)):
    flag += chr(ord(key[i]) ^ int(s[i],2))

print flag
```

0x2 曼切斯特编码和差分曼切斯特编码
在曼彻斯特编码中，每一位的中间有一跳变，位中间的跳变既作时钟信号，又作数据信号；从高到低跳变表示"1"，从低到高跳变表示"0"。还有一种是差分曼彻斯特编码，每位中间的跳变仅提供时钟定时，而用每位开始时有无跳变表示"0"或"1"，有跳变为"0"，无跳变为"1"。
http://www.cnblogs.com/BinB-W/p/5045918.html

```
#!/usr/bin/env python
#coding:utf-8
import re
hex1 = 'AAAAA56A69AA55A95995A569AA95565556' # #  0x8893CA58
#hex1 = 'AAAAA56A69AA556A965A5999596AA95656'
def bintohex(s1):
    s2 = ''
    s1 = re.findall('.{4}',s1)
    print '每一个hex分隔:',s1
    for i in s1:
        s2 += str(hex(int(i,2))).replace('0x','')

    print 'ID:',s2
def diffmqst(s):
    s1 = ''
    s = re.findall('.{2}',s)
    cc = '01'
    for i in s:
        if i == cc:
            s1 += '0'
        else:
            s1 += '1'
        cc = i  # 差分加上cc = i
    print '差分曼切斯特解码:',s1
    bintohex(s1)
def mqst(s):  #只能算曼切斯特编码,无法算差分
    mdict = {'5': '00', '6': '01', '9': '10', 'A': '11'}
    a1 = ''.join(mdict[i] for i in s)
    a2 = ''.join(mdict[i][::-1] for i in s)
    print '曼切斯特解码:   ',a1
    print '曼切斯特解码2:  ',a2
    bintohex(a1)
    bintohex(a2)
if __name__ == '__main__':
    bin1 = bin(int(hex1,16))[2:]
    diffmqst(bin1)
```

```
    mqst(hex1)
```

0x3 轮盘密码 Wheel Cipher

也是二战期间的使用的一种密码，原理大概就是给定一个密码表和一个密钥，根据密钥对每一列旋转到第一位（和轮盘一样）， 然后看每一列的字符是否有规律
在ISCC2017上就有这样一个题，脚本如下:

```
#coding:utf-8
a ="""
ZWAXJGDLUBVIQHKYPNTCRMOSFE
KPBELNACZDTRXMJQOYHGVSFUWI
BDMAIZVRNSJUWFHTEQGYXPLOCK
RPLNDVHGFCUKTEBSXQYIZMJWAO
IHFRLABEUOTSGJVDKCPMNZQWXY
AMKGHIWPNYCJBFZDRUSLOQXVET
GWTHSPYBXIZULVKMRAFDCEONJQ
NOZUTWDCVRJLXKISEFAPMYGHBQ
XPLTDSRFHENYVUBMCQWAOIKZGJ
UDNAJFBOWTGVRSCZQKELMXYIHP
MNBVCXZQWERTPOIUYALSKDJFHG
LVNCMXZPQOWEIURYTASBKJDFHG
JZQAWSXCDERFVBGTYHNUMKILOP
"""

b="NFQKSEVOQOFNP"
c="2,3,7,5,13,12,9,1,8,10,4,11,6"
a=a.splitlines()
a.pop(0)
# print a
s = ''
t = []
c=c.split(',')
for i in range(0,len(c)):
    #print a[int(c[i])-1].index(b[i])+1
    index = a[int(c[i])-1].index(b[i])
    a[int(c[i])-1] = a[int(c[i])-1][index:]+a[int(c[i])-1][:index]
    print a[int(c[i])-1]
    t.append(a[int(c[i])-1])
    # print '交换',i,'和',int(c[i])-1

for y in range(len(t[0])):
    for x in range(0,len(t)):
        s += t[x][y]

    print '第'+str(y)+'列',''.join(s)

    s = ''
```

0x4 xxencode

XXencode将输入文本以每三个字节为单位进行编码。如果最后剩下的资料少于三个字节，不够的部份用零补齐。这三个字节共有24个Bit，以6bit为单位分为4个组，每个组以十进制来表示所出现的数值只会落在0到63之间。以所对应值的位置字符代替。它所选择的可打印字符是：+-0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz，一共64个字符。跟base64打印字符相比，就是多一个“-”字符，少一个”/”字符。

源文本：The quick brown fox jumps over the lazy dog

编码后：hJ4VZ653pQKBf647mPrRi64NiS0-eBKpkQm-jRaJm65FcNG-gMLdt64FiNkc+

解密地址:http://web.chacuo.net/charsetxxencode

0x5 UUencode编码

UUencode是一种二进制到文字的编码，最早在unix 邮件系统中使用，全称：Unix-to-Unix encoding，UUencode将输入文本以每三个字节为单位进行编码，如果最后剩下的资料少于三个字节，不够的部份用零补齐。三个字节共有24个Bit，以6-bit为单位分为4个组，每个组以十进制来表示所出现的字节的数值。这个数值只会落在0到63之间。然后将每个数加上32，所产生的结果刚好落在ASCII字符集中可打印字符（32-空白…95-底线）的范围之中。

源文本：The quick brown fox jumps over the lazy dog

编码后： M5&AE('%U:6-K(&)R;W=N(&9O>"!J=6UP<R!O=F5R('1H92!L87IY(&1O9PH*

解密地址:http://web.chacuo.net/charsetuuencode

0x6 Unicode编码

Unicode编码有以下四种编码方式：

源文本：The

&#x [Hex]：The

&# [Decimal]：The

\U [Hex]：\U0054\U0068\U0065

\U+ [Hex]：\U+0054\U+0068\U+0065

0x7 Escape/Unescape编码

Escape/Unescape加密解码/编码解码,又叫%u编码，采用UTF-16BE模式， Escape编码/加密,就是字符对应UTF-16 16进制表示方式前面加%u。Unescape解码/解密，就是去掉"%u"后，将16进制字符还原后，由utf-16转码到自己目标字符。如：字符“中”，UTF-16BE是：“6d93”，因此Escape是“%u6d93”。

源文本：The

编码后：%u0054%u0068%u0065

0x8 埃特巴什码

埃特巴什码(Atbash Cipher)是一种以字母倒序排列作为特殊密钥的替换加密，也就是下面的对应关系：

ABCDEFGHIJKLMNOPQRSTUVWXYZ
ZYXWVUTSRQPONMLKJIHGFEDCBA

明文：the quick brown fox jumps over the lazy dog

密文：gsv jfrxp yildm ulc qfnkh levi gsv ozab wlt

0x9 dvorak 键盘编码

Dvorak键盘是一种将常用字母都归在一起，以期提高打字速度键盘布局。1936年由美国人August Dvorak设计。键盘布局在硬盘里。现在通用的QWERTY键盘，以键盘第一排字母的左边6个字母而得名。
#仿射函数
密文mzdvezc是用仿射函数y=5x+12加密得到的，试对其解密。
直接一个Python解决问题
#!/usr/bin/env python

```
# -*- coding: utf-8 -*-
__Url__ = 'Http://www.purpleroc.com'
__author__ = 'Tracy_梓朋'
def affine(a, b):
    pwd_dic = {}
    for i in range(26):
        pwd_dic[chr(((a*i+b)%26)+97)] = chr(i+97)
    return pwd_dic
if __name__ == '__main__':
    pwd_dic = {}
    pwd = "mzdvezc"   #密文mzdvezc
    plain = []
    pwd_dic = affine(5, 12)   #5，12 分别为两个系数
    for i in pwd:
        plain.append(pwd_dic[i])
    print "You Flag is: " + "".join(plain)

#异或加密
#########################################################
小野入侵了一个台服务器，发现一个保存重要密码的程序代码，小野又发现了一些蛛丝马迹，他能破解么？
附件： 公司内部密钥程序.txt
0xBE, 0x2A, 0x28, 0x48, 0x7A, 0x5C, 0x2A, 0x21, 0xCB, 0x93, 0x0D, 0x2A, 0x70,
0x36, 0xD3, 0x4E, 0xC9, 0xB6, 0xCF, 0x3C, 0xB6, 0x71, 0x99, 0xF5, 0x46, 0x69, 0xA1,
0x24, 0xF9, 0x71, 0x70, 0x11, 0x2A, 0x37, 0x31, 0x27, 0x30, 0x16, 0x71, 0x90, 0x26,
0xC9, 0x18, 0x72, 0xC9, 0x09, 0x4E, 0xC9, 0x0B, 0x5E, 0xC9, 0x4B, 0xC9, 0x2B, 0x4A,
0xEF, 0x7F, 0x28, 0x48, 0x7A, 0x5C, 0x37, 0x47, 0xD7, 0xBD, 0x15, 0xBA, 0xD7, 0x22,
0xC9, 0x07, 0x7E, 0xC9, 0x0E, 0x47, 0x3A, 0x41, 0x8F, 0xC9, 0x1B, 0x62, 0x41, 0x9F,
0x71, 0xBD, 0x05, 0xC9, 0x76, 0xF9, 0x41, 0xB7, 0xDB, 0x4D, 0xFC, 0x44, 0x78, 0x86,
0x36, 0x4A, 0x83, 0x88, 0x45, 0x41, 0x92, 0x04, 0xA9, 0xB3, 0x79, 0x16, 0x66, 0x5E,
0x37, 0xA6, 0xC9, 0x1B, 0x66, 0x41, 0x9F, 0x24, 0xC9, 0x7E, 0x39, 0xC9, 0x1B, 0x5E,
0x41, 0x9F, 0x41, 0x6E, 0xF9, 0xD7, 0x1D, 0xE9, 0x15, 0x23, 0x7F, 0x28, 0x48, 0x7A,
0x5C, 0x37, 0xEB, 0x71, 0x99, 0x11, 0x2A, 0x35, 0x34, 0x36, 0x64, 0x2A, 0x14, 0x29,
0x68, 0x7A, 0xC9, 0x86, 0x11, 0x12, 0x12, 0x11, 0xBD, 0x15, 0xBE, 0x11, 0xBD, 0x15,
0xBA, 0xD2, 0xD2, 0xD2, 0xD2
企业加密方式.txt 公司内部所有的加密形式均为异或加密，且只使用异或加密，加密KEY：0x42 请牢记，保密！
异或完了后,把它放到一个exe的入口点,然后,直接运行exe就可以弹出flag.
#########################################################
#!/usr/bin/env python
# -*- coding: utf-8 -*-
__Url__ = 'Http://www.purpleroc.com'
__author__ = 'Tracy_梓朋'
pwd = "BE,  0x2A,  0x28,  0x48,  0x7A,  0x5C,  0x2A,  0x21,  0xCB,  0x93,  0x0D,  0x2A,  \
    0x70,  0x36,  0xD3,  0x4E,  0xC9,  0xB6,  0xCF,  0x3C,  0xB6,  0x71,  0x99,  0xF5,  0x46,  \
    0x69,  0xA1,  0x24,  0xF9,  0x71,  0x70,  0x11,  0x2A,  0x37,  0x31,  0x27,  0x30,  0x16,  \
    0x71,  0x90,  0x26,  0xC9,  0x18,  0x72,  0xC9,  0x09,  0x4E,  0xC9,  0x0B,  0x5E,  0xC9,  \
    0x4B,  0xC9,  0x2B,  0x4A,  0xEF,  0x7F,  0x28,  0x48,  0x7A,  0x5C,  0x37,  0x47,  0xD7,  \
    0xBD,  0x15,  0xBA,  0xD7,  0x22,  0xC9,  0x07,  0x7E,  0xC9,  0x0E,  0x47,  0x3A,  0x41,  \
    0x8F,  0xC9,  0x1B,  0x62,  0x41,  0x9F,  0x71,  0xBD,  0x05,  0xC9,  0x76,  0xF9,  0x41,  \
    0xB7,  0xDB,  0x4D,  0xFC,  0x44,  0x78,  0x86,  0x36,  0x4A,  0x83,  0x88,  0x45,  0x41,  \
    0x92,  0x04,  0xA9,  0xB3,  0x79,  0x16,  0x66,  0x5E,  0x37,  0xA6,  0xC9,  0x1B,  0x66,  \
    0x41,  0x9F,  0x24,  0xC9,  0x7E,  0x39,  0xC9,  0x1B,  0x5E,  0x41,  0x9F,  0x41,  0x6E,  \
    0xF9,  0xD7,  0x1D,  0xE9,  0x15,  0x23,  0x7F,  0x28,  0x48,  0x7A,  0x5C,  0x37,  0xEB,  \
    0x71,  0x99,  0x11,  0x2A,  0x35,  0x34,  0x36,  0x64,  0x2A,  0x14,  0x29,  0x68,  0x7A,  \
    0xC9,  0x86,  0x11,  0x12,  0x12,  0x11,  0xBD,  0x15,  0xBE,  0x11,  0xBD,  0x15,  0xBA,  \
    0xD2,  0xD2,  0xD2,  0xD2"
key = "0x42"
if __name__ == '__main__':
    cipher = pwd.split(',  0x')
    #print "".join(cipher)
    plain = []
```

```
    plain = []
    temp = []
    for i in cipher:
        open("../../DATA/BASIC05/basic5.bin","ab").write(chr(int(i, 16) ^ int(key, 16)))
    #print "Flag is: Vk*8wvt&"

##############################################
上次的用完了，我生日又快到了，再给我点流量当生日礼物吧。 http://script.iscc.org.cn/script03/
看上去像去年的神兽，可其实不是，也不知道是要考啥，有个提示转流量下限是1G。 所以让结果等于NAN，无穷小就行了
          输入：99e9999-99e1111 得到flag：
          你打算送我NANMB流量。 谢谢你给我这么多流量，flag给你了。 flag:{68d43d512ca7214e05acc96cc100515e}
##############################################
波利比奥斯方阵密码
  1  2  3  4   5
1 A  B  C  D   E
2 F  G  H  I/J K
3 L  M  N  O   P
4 Q  Q  S  T   U
5 V  W  X  Y   Z
加密实例：
明文： THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
密文： 442315 4145241325 1242345233 213453 2445323543 442315 31115554 143422
#普莱菲尔密码
加密解密实例(ps：这里加解密也是横向编制密码表)：
#!python
>>>from pycipher import Playfair
>>>Playfair('CULTREABDFGHIKMNOPQSVWXYZ').encipher('THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG')
#输出'UKDNLHTGFLWUSEPWHLISNPCGCRGAUBVZAQIV'
>>>Playfair('CULTREABDFGHIKMNOPQSVWXYZ').decipher('UKDNLHTGFLWUSEPWHLISNPCGCRGAUBVZAQIV')
#输出'THEQUICKBROWNFOXIUMPSOVERTHELAZYDOGX'
#维吉尼亚密码
明文： THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
密钥(循环使用，密钥越长相对破解难度越大)： CULTURE
（2）已知密钥加解密
#!python
>>>from pycipher import Vigenere
>>>Vigenere('CULTURE').encipher('THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG')
'VBPJOZGMVCHQEJQRUNGGWQPPKNYINUKRXFK'
>>>Vigenere('CULTURE').decipher('VBPJOZGMVCHQEJQRUNGGWQPPKNYINUKRXFK')
'THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG'
#格罗斯费尔德密码(Gronsfeld cipher)
#!python
>>>from pycipher import Gronsfeld
>>>Gronsfeld([2,20,11,45,20,43,4]).encipher('THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG')
'VBPJOZGMVCHQEJQRUNGGWQPPKNYINUKRXFK'
>>>Gronsfeld([2,20,11,45,20,43,4]).decipher('VBPJOZGMVCHQEJQRUNGGWQPPKNYINUKRXFK')
'THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG'
#自动密钥密码
#Porta密码--Porta密码(Porta Cipher)是一个由意大利那不勒斯的医生Giovanni Battista della Porta发明的多表代换密码，Porta密码具有加密解密过程的是相同的特点
密表
#!shell
KEYS| A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
----|-------------------------------------------------
A,B | N O P Q R S T U V W X Y Z A B C D E F G H I J K L M
C,D | O P Q R S T U V W X Y Z N M A B C D E F G H I J K L
E,F | P Q R S T U V W X Y Z N O L M A B C D E F G H I J K
G,H | Q R S T U V W X Y Z N O P K L M A B C D E F G H I J
I,J | R S T U V W X Y Z N O P Q J K L M A B C D E F G H I
K,L | S T U V W X Y Z N O P Q R I J K L M A B C D E F G H
```

M,N | T U V W X Y Z N O P Q R S H I J K L M A B C D E F G
O,P | U V W X Y Z N O P Q R S T G H I J K L M A B C D E F
Q,R | V W X Y Z N O P Q R S T U F G H I J K L M A B C D E
S,T | W X Y Z N O P Q R S T U V E F G H I J K L M A B C D
U,V | X Y Z N O P Q R S T U V W D E F G H I J K L M A B C
W,X | Y Z N O P Q R S T U V W X C D E F G H I J K L M A B
Y,Z | Z N O P Q R S T U V W X Y B C D E F G H I J K L M A
明文： THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
密钥(循环使用，密钥越长相对破解难度越大)： CULTURE
加密过程：明文字母'T'列与密钥字母'C'行交点就是密文字母'F',以此类推。
密文： FRW HKQRY YMFMF UAA OLWHD ALWI JPT ZXHC NGV
在线解码：http://www.practicalcryptography.com/ciphers/classical-era/porta/
#培根密码
#!shell
A = aaaaa I/J = abaaa R = baaaa

B = aaaab K = abaab S = baaab

C = aaaba L = ababa T = baaba

D = aaabb M = ababb U/V = baabb

E = aabaa N = abbaa W = babaa

F = aabab O = abbab X = babab

G = aabba P = abbba Y = babba

H = aabbb Q = abbbb Z = babbb
明文： T H E F O X
密文： baaba aabbb aabaa aabab abbab babab
#asp混淆加密，VBScript/JScript.Encode
 aspencode
密文#@~^EQAAAA==V Xlj4UmkaYAUmKN3bAYAAA==^#~@?
丢到http://adophper.com/encode.html解密即可。。
#aaencode 颜文字 直接开发者模式console运行
?ω??= / ‘m’) ? ~┻━┻   //*’? ‘*/ ['_']; o=(???) =_=3; c=(?0?) =(???)-(???); (?Д?) =(?0?)= (o^_^o)/ (o^_^o);(?
Д?)={?0?: '_' ,?ω?? : ((?ω??==3) +'_') [?0?] ,???? :(?ω??+ '_')[o^_^o -(?0?)] ,?Д??:((???==3) +'_')[???] }; (?Д?
) [?0?] =((?ω??==3) +'_') [c^_^o];(?Д?) ['c'] = ((?Д?)+'_') [(???)+(???)-(?0?)];(?Д?) ['o'] = ((?Д?)+'_') [?0?
];(?o?)=(?Д?) ['c']+(?Д?) ['o']+(?ω?? +'_')[?0?]+ ((?ω??==3) +'_') [???] + ((?Д?) +'_') [(???)+(???)]+ ((???==3)
 +'_') [?0?]+((???==3) +'_') [(???) - (?0?)]+(?Д?) ['c']+((?Д?)+'_') [(???)+(???)]+ (?Д?) ['o']+((???==3) +'_')
[?0?];(?Д?) ['_'] =(o^_^o) [?o?] [?o?];(?ε?)=((???==3) +'_') [?0?]+ (?Д?) .?Д??+((?Д?)+'_') [(???) + (???)]+((??
?==3) +'_') [o^_^o -?0?]+((???==3) +'_') [?0?]+ (?ω?? +'_') [?0?]; (???)+=(?0?); (?Д?)[?ε?]='\\'; (?Д?).?0??=(?Д
?+ ???)[o^_^o -(?0?)];(o???o)=(?ω?? +'_')[c^_^o];(?Д?) [?o?]='\"';(?Д?) ['_'] ((?Д?) ['_'] (?ε?+(?Д?)[?o?]+ (?Д
?)[?ε?]+(?0?)+ (???)+ (?0?)+ (?Д?)[?ε?]+(?0?)+ ((???) + (?0?))+ (???)+ (?Д?)[?ε?]+(?0?)+ (???)+ ((???) + (?0?))+
 (?Д?)[?ε?]+(?0?)+ ((o^_^o) +(o^_^o))+ ((o^_^o) - (?0?))+ (?Д?)[?ε?]+(?0?)+ ((o^_^o) +(o^_^o))+ (???)+ (?Д?)[?ε?
]+((???) + (?0?))+ (c^_^o)+ (?Д?)[?ε?]+(???)+ ((o^_^o) - (?0?))+ (?Д?)[?ε?]+(?0?)+ (?0?)+ (c^_^o)+ (?Д?)[?ε?]+(?
0?)+ (???)+ ((???) + (?0?))+ (?Д?)[?ε?]+(?0?)+ ((???) + (?0?))+ (???)+ (?Д?)[?ε?]+(?0?)+ ((???) + (?0?))+ (???)+
 (?Д?)[?ε?]+(?0?)+ ((???) + (?0?))+ ((???) + (o^_^o))+ (?Д?)[?ε?]+((???) + (?0?))+ (???)+ (?Д?)[?ε?]+(???)+ (c^_
^o)+ (?Д?)[?ε?]+(?0?)+ (?0?)+ ((o^_^o) - (?0?))+ (?Д?)[?ε?]+(?0?)+ (???)+ (?0?)+ (?Д?)[?ε?]+(?0?)+ ((o^_^o) +(o^
_^o))+ ((o^_^o) +(o^_^o))+ (?Д?)[?ε?]+(?0?)+ (???)+ (?0?)+ (?Д?)[?ε?]+(?0?)+ ((o^_^o) - (?0?))+ (o^_^o)+ (?Д?)[?
ε?]+(?0?)+ (???)+ (o^_^o)+ (?Д?)[?ε?]+(?0?)+ ((o^_^o) +(o^_^o))+ ((o^_^o) - (?0?))+ (?Д?)[?ε?]+(?0?)+ ((???) + (
?0?))+ (?0?)+ (?Д?)[?ε?]+(?0?)+ ((o^_^o) +(o^_^o))+ (c^_^o)+ (?Д?)[?ε?]+(?0?)+ ((o^_^o) +(o^_^o))+ (???)+ (?Д?)[
?ε?]+(???)+ ((o^_^o) - (?0?))+ (?Д?)[?ε?]+((???) + (?0?))+ (?0?)+ (?Д?)[?o?]) (?0?)) ('_');
#jsfuck
只包含[]()!+六个字符，可以直接在console控制台运行得到flag
http://www.jsfuck.com/
#jother
8个少量字符包括:!+()[]{}
直接在浏览器(IE可以)的控制台里输入密文即可执行解密

直接在浏览器(IE可以)的控制台里输入密文即可执行解密：
#brainfuck
只有八种符号，所有的操作都由这八种符号(> < + - . , [ ])的组合来完成。
明文：hello!
#!shell
```
+++++ +++++ [->++ +++++ +++<] >++++ .---. +++++ ++..+ ++.<+ +++++ +++++
[->++ +++++ ++++< ]>+++ ++++. <++++ +++[- >---- ---<] >--.< +++++ ++[->
----- --<]> ----- ----- .<
```
#ook
#和brainfuck是一个本地解密站点，解密即可
```
    Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
    Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook.
    Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
    Ook. Ook? Ook. Ook? Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook! Ook. Ook. Ook.
    Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook? Ook. Ook.
    Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook! Ook! Ook! Ook!
    Ook! Ook! Ook? Ook. Ook? Ook! Ook. Ook? Ook! Ook! Ook! Ook! Ook! Ook. Ook.
    Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook? Ook.
    Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook.
    Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook. Ook? Ook! Ook. Ook? Ook. Ook.
    Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook.
    Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook! Ook! Ook! Ook! Ook! Ook! Ook? Ook.
    Ook? Ook! Ook. Ook? Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook. Ook! Ook! Ook!
    Ook! Ook! Ook! Ook! Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook.
    Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook!
    Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook? Ook.
    Ook? Ook! Ook. Ook? Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook!
    Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook!
    Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
    Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook.
    Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook. Ook? Ook! Ook. Ook? Ook.
    Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook? Ook. Ook. Ook.
    Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook.
    Ook. Ook. Ook. Ook. Ook. Ook? Ook. Ook? Ook! Ook. Ook? Ook. Ook. Ook! Ook.
    Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook!
    Ook! Ook! Ook! Ook! Ook! Ook? Ook. Ook? Ook! Ook. Ook? Ook! Ook! Ook! Ook!
    Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook.
    Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook?
    Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook?
    Ook. Ook? Ook! Ook. Ook? Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook!
    Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
    Ook. Ook! Ook. Ook. Ook. Ook! Ook. Ook. Ook. Ook! Ook. Ook. Ook. Ook! Ook.
    Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook!
    Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
    Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook.
    Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook. Ook? Ook! Ook. Ook? Ook. Ook.
    Ook. Ook. Ook. Ook. Ook! Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook.
    Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
    Ook. Ook. Ook! Ook. Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook. Ook.
    Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook? Ook. Ook. Ook.
    Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook.
    Ook. Ook? Ook. Ook? Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
    Ook. Ook. Ook. Ook. Ook! Ook. Ook? Ook.
```
#DES
加密方式:DES
密钥:6XaMMbM7
密文:U2FsdGVkX18IBEATgMBe8NqjIqp65CxRjjMxXIIUxIjBnAODJQRkSLQ/+lHBsjpv1BwwEawMo1c=
如果解密不了，可以换个网站试试，毕竟Des加密时候的填充、编码方式都会影响结果（实测这个网站可以：http://encode.chahuo.com）
明文:ctf{67a166801342415a6da8f0dbac591974}
#SHA1加密

#当铺密码
就是一种将中文和数字进行转化的密码，算法相当简单:当铺用的十个数字的暗码是“由”、“中”、“人”、“工”、“大”、“王”、“夫”、“井”、“羊”、“非”十个字。这种暗码是以每一个字上下左右露出的笔画的字头多少来表示数字的。如“由”字上面一竖露出一个头，为数字“一”，“中”字上下露出二个字头，为数字“二”，“羊”字上下左右露出九个字头，为数字“九”，“非”字上下左右共露出十个字头，为数字“十”。如果说“夫井”，那就是在说“七八”这个数字了。
密文：羊由大井夫大人王中工
明文：9158753624
#ascii字符偏移
密文
gndk€rlqhmtkwwp}z
二、分析题目
gndk€rlqhmtkwwp}z这个格式像不像flag{*******}?
我们比较一下"gndk"与"flag"的ASCII码
gndk的10进制的ASCII码分别是：103 110 100 107
flag的10进制的ASCII码分别是 ：102 108 97 103
发现ASCII以此减少 1 2 3 4，所以以此类推
但是不知道怎么处理'€'这个字符，暂时换成别的字符，最后在换成'{'
用python解解决
######八进制，十六进制###小葵工具
反斜杠加两位或三位数字，应该是八进制转义序列，八进制转ascii，脚本跑一下
\x格式的，看来是16进制，写个脚本16进制转ascii
\u开头的16进制Unicode编码，在线Unicode转换字符
纯数字应该是ascii码十进制表示，转为字符（注意先把中间的逗号删掉）
html转义字符，可以用Python解码（先删掉中间的空格）详细教程可以见Python HTML编码解码

继续Python html解码
####残缺的hash值
小明一直将电脑密码的哈希值写在纸上，结果一不小心将墨水撒在了上面，只看得到前10位是c2979c7124，小明只记得密码是4位的数字加字母，你能帮小明恢复密码的哈希值吗？（提示：flag为密码的哈希值）

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-

import string
import hashlib
import itertools

def md5(cstr):
    m = hashlib.md5()
    m.update(cstr)
    return m.hexdigest().lower()

ls = list(string.lowercase) + list(string.uppercase) + list(string.digits)

s = 'c2979c7124'
for l in list(itertools.permutations(ls,4)):
    p = ''.join(l)
    if md5(p)[0:len(s)] == s:
        print md5(p)
        exit(0)
```

或者

```
<?php
$str='0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
$cou=strlen($str);
for($i=0;$i<$cou-1;$i++){
        for($j=0;$j<$cou-1;$j++){
                for($k=0;$k<$cou-1;$k++){
                        for($l=0;$l<$cou-1;$l++){
```

```
                                $ret=md5($str[$i].$str[$j].$str[$k].$str[$l]);
                                if(substr($ret,0,10)=='c2979c7124'){
                                        echo $ret;die();
                                }
                        }
                }
        }
}
?>
#RSA1
#openssl安装方法：https://www.youtube.com/watch?v=892_4UQy_L8


列置换
```