ctf-攻防世界-crypto：幂数加密

原创

2021gracedoudou 于 2021-12-02 16:18:57 发布 198 收藏
分类专栏： # crypto 文章标签： unctf

本文链接：https://blog.csdn.net/m0_62619559/article/details/121678534

先看题，给出了flag格式和幂数加密的方式。

附件里的内容为884210122048022440401422420248012 2

正常的二进制幂数加密只有0，1，2，3，4，5，是不会出现8的。通过百度了解到这是云影密码、

简单说来就是以0为分隔符，将分割后的每个数字加起来，a-z对应1-26

（关于二进制幂数加密和云影密码的具体介绍放在文章最后）

那么可以手动计算，也可以写脚本。

手动计算如下：

```
884210122048022440401422420248012 2

88421 122 48 2244 4 142242 248 122
23      5    12 12    4 15       14   5
W       E    L  L     D O        N    E

flag: cyberpeace{WELLDONE}
```

CSDN @2021gracedoudou

脚本基础太差，先引用了别人的

原始脚本来源：攻防世界--CRYPTO--5.幂数加密--wp - 简书

https://adworld.xctf.org.cn/task/writeup?type=crypto&id=5120&number=5&grade=0&page=1（官方wp ch4ce）

```
a="884210122048022440401422420248012 2"
a=a.split('0')  #以0为分隔符，将a分割为列表类型
flag=''
for i in range(0,len(a)):   #从0开始循环a的元素个数-1，即0-7
    str = a[i]
    list = []
    sum=0
    for j in str:   #遍历str的每一个字母
        list.append(j)  #列表尾部加上j这个元素
        length = len(list)  #算出一共有多少个数字

    for k in range(0,length):
        sum+=int(list[k]) #把所有数字求和
    flag+=chr(sum+64)   #第一个英文字母的ascii码从65开始，所以要加上65-1，然后使用chr将ascii转化为字符
print(flag)
```

```
a="884210122048022440401422420248012 2"
a=a.split('0')
flag=''
for i in range(len(a)):
    str=a[i]
    sum=0
    for j in str:
        sum+=int(j)
    flag+=chr(sum+64)
print(flag)
```

关于二进制加密：二进制幂数加密法_百度百科

关于云影密码：https://gist.github.com/wh1t3p1g/ffd15270914492491e18ff9f070eab2b

```
【云影密码】
此密码运用了1248代码，因为本人才疏学浅，尚未发现有过使用的先例，因此暂归为原创密码，若有密码界前辈认为不妥，
请指出此密码或类似密码的普遍使用历史并附寄一份到我站内邮箱，我将以最快速度核查并改正。由于这个密码，我和片风云影初识，为了
原理很简单，有了1，2，4，8这四个简单的数字，你可以以加法表示出0-9任何一个数字，例如0=28，7=124，9=18。
这样，再用1-26来表示A-Z，就可以用作密码了。
为了不至于混乱，我个人引入了第五个数字0，来用作间隔，以避免翻译错误，所以还可以称“01248密码”。

题目：1240101180118021201140180 4
第一步，分割，即124  1  118  118  212  114  18  4
第二步，基本翻译，例如124可以表示7，也可以表示16（但不可能是34，因为不会超过26），所以可以放在一边，翻译其他没有异议的，
可得：124  a  s  s  w  o  18  d
第三步，推测得出明文。可以推测后面的18表示r，前面的为p最合适。
```