HGAME 2020 week1

原创

Slightwindsec 于 2020-04-23 01:14:58 发布 919 收藏 1
分类专栏： CTF 文章标签： python

本文链接：https://blog.csdn.net/qq_41956187/article/details/105697592

> 首发于我的博客：https://blog.slight-wind.com/

Web

Cosmos 的博客

1.看提示去 GitHub 上找这个网站的源代码，搜索 Cosmos Hgame 就可以找到，点开 3 commits，点开 new file 就可以看到：
base64 解码: aGdhbWV7ZzF0X2xlQGtfMXNfZGFuZ2VyMHVzXyEhlSF9
2.base64 解码得到 flag。

hgame{g1t_le@k_1s_danger0us_!!!}

Crypto

InfantRSA

题目

真 签到题

```
p = 681782737450022065655472455411;
q = 675274897132088253519831953441;
e = 13;
c = pow(m,e,p*q) = 27569846508236107014517368841149631154217290260855985901984 1
```

Writeup

1.真就“真 签到题”，拿到题赶快就搞了，还是二血…

```
p = 681782737450022065655472455411 q = 675274897132088253519831953441 e = 13 c = 27569846508236107014517368841149631154217290260855985901984 1 def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y) def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1:
        raise Exception('modular inverse does not exist')
    else:
        return x % m d=modinv(e,(p-1)*(q-1)) m=pow(c,d,p*q) print(hex(m))
```

hgame{t3Xt6O0k_R5A!!!}

Affine

题目

```
import gmpy2
from secret import A, B, flag
assert flag.startswith('hgame{') and flag.endswith('}')

TABLE = 'zxcvbnmasdfghjklqwertyuiop1234567890QWERTYUIOPASDFGHJKLZXCVBNM'
MOD = len(TABLE)

cipher = ''
for b in flag:
i = TABLE.find(b)
if i == -1:
cipher += b
else:
ii = (A*i + B) % MOD
cipher += TABLE[ii]

print(cipher)
# A8I5z{xr1A_J7ha_vG_TpH410}
```

Writeup

1.仿射加密解密，（A 为 A 对 MOD 的逆元）。

$c=(A*x+B)\ \%\ MOD$

$m=A\ *(x-B)\ \%\ MOD$

2.可以看出想解密要知道 A 和 B，我们已知 “hgame{” 被加密成了 “A8I5z{”，我们只需要取前两个字符就可以枚举出 A 和 B 了，从 12 变到46（ h 变成 A ），从 11 变到 33（ g 变成 8 ），模数 MOD 为 TABLE 的长度 62。

```
#include<iostream>
using namespace std;
int main(){
 for(int i=0;i&lt;62;i++)
  for(int j=0;j&lt;62;j++)
   if(((i*12+j)%62==46)&&((i*11+j)%62==33)&&((i*7+j)%62==43)){
    cout<<i<<' '<<j<<endl;
    break;
   }
 return 0;
}
```

3.得到 A 和 B 的值 13 14 就可以解密了。

```
import gmpy2
TABLE = 'zxcvbnmasdfghjklqwertyuiop1234567890QWERTYUIOPASDFGHJKLZXCVBNM'
cipher='A8l5z{xr1A_J7ha_vG_TpH410}'
def egcd(a, b):
    if a == 0:
      return (b, 0, 1)
    else:
      g, y, x = egcd(b % a, a)
      return (g, x - (b // a) * y, y)
def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1:
        raise Exception('modular inverse does not exist')
    else:
        return x % m
MOD = len(TABLE)
A=13
B=14
C=modinv(A , MOD)
flag = ''
for b in cipher:
    i = TABLE.find(b)
    if i == -1:
        flag += b
    else:
        ii = C*(i - B) % MOD
        flag += TABLE[ii]
print(flag)
```

hgame{M4th_u5Ed_iN_cRYpt0}

not_One-time

题目

```
import os, random
import string, binascii, base64

from secret import flag
assert flag.startswith(b'hgame{') and flag.endswith(b'hgame{')

flag_len = len(flag)

def xor(s1, s2):
#assert len(s1)==len(s2)
return bytes( map( (lambda x: x[0]^x[1]), zip(s1, s2) ) )

random.seed( os.urandom(8) )
keystream = ''.join( [ random.choice(string.ascii_letters+string.digits) for _ in range(flag_len) ] )
keystream = keystream.encode()
print( base64.b64encode(xor(flag, keystream)).decode() )
```

writeup

1.这题是我耗时最长的一题… nc 连接服务器的，所以可以得到很多的 flag 密文，而且已知flag以 'hgame{' 开始以 '}' 结束。

2.加密脚本就是用 flag 和随机等长字符串异或运算然后输出base64编码后的结果，一开始在网上各种找资料，流加密多次密钥攻击，发现都是这样:

> 如果攻击者可以拿到 A、B 的密文 E(A)、E(B)，以及攻击者自己的明文 B，就可以在无需知道密钥的情况下计算出 A 的明文：
> A = E(A) xor E(B) xor B

也在 GitHub 上可以找到 many-time-pad-attack 的脚本，也是需要有一段自己的字符串让脚本加密一次才能推出其他的明文。而这题显然不可以上传一个字符串让服务器上的脚本把 flag 和自己的字符串异或一下再发给自己...

3.然而想到一个集合内的字符相互异或的结果也是一个很有限的集合，反过来如果知道一个异或的结果，我们就可以大概知道这个结果可能是哪些字符异或出来的，称这些字符组成的集合为A，那么以此类推（多次nc下来一些密文）得到集合B，C，D...然后取这些集合的交集就可以极大地缩小这个位置可能的字符集合。

exp：

```
import gmpy2
import base64
yb=[
'XC8JlVMTKws0JmU/d0p6eEgAGAE/TGd+NkF0JUZ0W1YGGX8fXHsdXygPFA==',
'OikNXwJPCn8hBgQHDRl1Bmt+MVErax9eJhw9Ky0VdS4+AXljQ0UoRigARA==',
'XxlJXlYiGgU6G2lfDV4JYWxCUgkQZxRyDzRwHiExcjNqLwsIb11hYQhTSQ==',
'JQ0vBRQyInETGEBXclMCARBCVAgGGmpiHR52NDwlZSw1PkVuWAQZUHUOGw==',
'DFUxlgJDIVE9BmJZLRMCWxRdIyIkfEtHKksgIUExZFAZEIleVWY2dTspJQ==',
'HxE1IxUtI1EaO3ZbA0YbCxZcKyo9eWkUOyY/KUIlWREvU0ptfWQXfyoDMA==',
'DiwqBi9IFAQdAF8nPRp9ZFBbLCFXZx5OLzofJjN0ZTMyDFUsQAQ2Ww4wDA==',
'ACQiXyQRNEY/EgIaH2gedU17EDALRRJASQgzEwERXAs9EgIXbGQTYQgfEw==',
'DFUMPCsaAHwPG1wAARofRkoAL1Qcf1N/BCNwVyEkAgc6U3E3FAQZfBltPA==',
'OzQmBxclQn4vRkM9JBlrYGdiMAEueWpWRhwCMEluZi44Lmc0SUc4fy0jFw==',
'IwQ0LAsCEXs3MgI6PkkLSWxcCV8cSk9BFxsNDSUXAVURLQM6SgYqcgwDFg==',
'DxMDBF1MHlg9NEY+dn4gA0VeCy4RRBdDCQh2EzJ0UCNtG0YYXF9pexo0LA==',
'JAwZOiMYNHEBGAFYH1osfENeNTlLax9jDQIsKjt1WAUcHFQLY3E8VnUyBQ==',
'LgYVKD8UFI0mOFxZLmE+QUx0KAQySFBHHRh2ViY6Sj45XHc3ZwU+cBYpLQ==',
'BS8gOVUBHlxBC3s5DH5/Q21CFB4Wen8eTxE9UDYXfwgGD0krXQc0bi0wMA==',
'LDYzOAQQJ1xBCQkkLWA1UnQFMRcXV2IVJgsmCzQTWyAHWlgdfmQ6YjkNDg==',
'IBUKJjATJ0URJn89EEkeShUCJgU8SxNcEz0zPjcacTEeKkE1an0kdwUvPw==',
'XjMmARAXJ1U8G18CDkAKW35UMiw3QxVRRkV3AgRxAR4+OH4pSUlRUnoQHA==',
'PxMSlw40F0YmCwgeDUc0dBZxGS4Rd0x/SiUxFBAtZw9qKQQpYAc6ZxAXTg==',
'MDMMDyABOGE6EnUWF0Z1ZhBsKCwjRmoWPwEJLEwzZVIMPV44eQQaWDozKQ==',
'Xj8kNzYqAlAeR3YtKXlfWlxmUgZSWkJtHzYmHgI5Bg4IW2ocZmlxBHoXEA==',
'IBUXDCI1BnctHQQ9dn0MVkxsFSgEXXx+CSYCUSYpRjU6DUpgWVodDAEGGw==',
'DREYGQszP0oaEXAkN1k3QkteBVAWaktiOyQWVxg0AxVvI3cRHWYUWCsyMg==',
'LDMwG1cqIH0iKUQIE30hf2B7UjIffhRKS0M1Hiw0XC49CQYSZE8jX3JXMA==',
'IAYiWA8XAF4mHGNXPVI6B21NLwlcVVRLN0ZwNwc2fCAMGX8yZGAcDSQnMA==',
'Lhc0ClAUG0liOFYaM2x8dXRtUTBRdFNCESscUTY3VwsvJGAwFQ0bXSs2TA==',
'CgQEDzdLAVBNGkgiHXl8RRJNEC0UW0BzJiYDVQEWazlcJXQ/HnA0RToGRQ==',
'JFEQJDI/IQYCK1QaKWwfWFRSNA5WeBdfCEoENh5yfAcLKGMcH3EnWC5QFw==',
'PwgjIRcTBkMiRnY3Ek4EZ3RWBjYfHGJQLDwxLQwvSzwtHn0oamY4AhEKOQ==',
'HBNTCi0UOFAQNAEHFEUcUVBjDSxXRE0WNgd8IB0Nawc3DlE1R2YYAw8DNw==',
'ETNQDikdF2cdNnokDEk+BRRABDUNHkwWLCEuIxlJYTcZlUkSe1scYjsnJw==',
'CjZSKiYBF0E+GGI4KRo+BhV6VjMtW2pvTwA2J0Z1ZiNpAVQTW3wFeikmDA==',
'CwMRI1Q3AnoPHn9ZDWAaRWd2MxQKWxV8J0MHAD0NajYvG1QVWw0UXXUoSQ==',
'Wi4RJSkoCnYNRHMUME8eC0B5EQxWdx50LSQJDUZ7YzQaGXApYG0XTiJdCw==',
'CiMrOQIjEFIPIgQ3MGAiZncAOzAURUxNFEQmMiUrXgg7LEQTY0QCci4nKg==',
'KlcRVS8sCFAFOgQkA3EoV0EGBhAEfBd3KyoDHUU1ai8rJkRqTFMkQjlBHg==',
'WQs0DFVNAnU2NWc2cRp0RF5MWD4IYGcQEEcQJT8NXxldWFc2TGUGcywyJQ==',
'KywDNRlaRFdFRF4NEX0jXWthIB0jTFxKHwd3Dk0OZzMpOEYPZ0ApXzY9Cw==',
'HDRRGR0XJ34DQ1kaA058W2JiDi8IQXVHBhohMD02eyUwKF0eGmQUbgxSTQ==',
'WR0tLjBCCFYGJ3otF18eelZ0NQ0NW0NANQciEBwLSTc1O3oATH8eXSQ8Jw==',
```

```
'PAI4IVcDM1oBRAUHd0YMa1FgCSYMYU1fMTEmEgJ7fxQoUkcURncVWAYVNg==',
'Jy0tNyxPRwUyQWIeP0B0XUViVgMHTlBWL0QWKTg7YlMFLWoYR2Elc3sHKg==',
'DQoWFyEwCEUzJl0IcF4dYUh5Vh4/FWpzECM2ABx2Rg4UWWsMekYBTnQKCw==',
'AFUvJTwYJn0nAHcACnIoYHV5DRYlZHNNOUAMPgAgClVpH3pqZFJoBjsdHA==',
'XwJXKyZOM2l0R3ctEUYlSmtwAyYpGVxSLCYTlwAnYFYwG1wQf19gTStUSw==',
'WT8CXQ8fQGsyEl49CV8DBEZ9WTcwbm50TDEwlj06QTRuIXttYF0RVhRWDg==',
'OgovCQo1H1llOEc6cmp1akNfLQU2ZkpiC0txDAMZYQgUlUoTXnsWfBMDOQ==',
'OSlLWjw6CngZBAkdDmYcakFnWBAcYhB/TwAKAAEGYBAYB1VvTHAjbnYiRQ==',
'GTVQN1UdEHkSJX0cKUgefFBUliggQ0VEJAIADQYaZFZoMnQMV3QzYgVUOQ==',
'GAEvXVM3JAAiGgcLAF44YmEDNSohYxBgCUE/XD4FRCEUl2kNXFhjQnRROg==',
'ORQKLiEhBUs4S2AicEgmdFJSNC9Ud2xWOxovXTB2dlMSUwZte1xmBAQ0Tg==',
'P1BTOV0tFFsXNVMYNUc6ARJ/MAokG3FRDhofHiQyUAc0CGEXQ2QfVxc3BQ==',
'JhMZHgE0B0sSEH0EMEMXR3FnMgEiSmdtRkd2AwYwXAJuXH0QGwQEQ3AiEw==',
'DVMGKREaEwU+OEEqDGB5QktCNxQtSlNSBCE2FT8xAhRoDWsdSWYlWjQOFA=='
    ]
for i in range(0, 54):
    yb[i] = base64.b64decode(yb[i]).decode()
    yb[i] = bytes(yb[i], encoding='utf-8')
# KeySpace里面可以加其他的字符
KeySpace = b'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890{}-'
leng = len(KeySpace)


def maybe(Hex):
    ans = ''
    for i in range(0, leng):
        for j in range(i, leng):
            x = KeySpace[i] ^ KeySpace[j]
            if x == Hex:
                ans += chr(KeySpace[i])+chr(KeySpace[j])
    return ans


def PublicChr(aa, bb):
    anss = ''
    laa = len(aa)
    lbb = len(bb)
    for i in range(0, laa):
        for j in range(0, lbb):
            if aa[i] == bb[j]:
                if anss.find(aa[i]) == -1:
                    anss += aa[i]
    return anss


for i in range(0, 43):
    Ans = PublicChr(maybe(yb[0][i]), maybe(yb[1][i]))
    for j in range(0, 54):
        Ans = PublicChr(maybe(yb[j][i]), Ans)
    print(Ans)
```

hgame{r3us1nG+M3$5age-&&~rEduC3d_k3Y-5P4Ce}

Reorder

Whiteup

这题只有 nc 连接，随便输入会发现顺序被打乱后输出，最后输出打乱后的 flag ，所以只要按顺序输入字母数字然后看是怎样被打乱的，就说明 flag 也被用同样的方法打乱了，原样复原即可。

hgame{jU$t+5ImpL3_PeRmuTATi0n!!}

Misc

欢迎参加HGame!

题目

来来来，签个到吧～
Li0tlC4uLi0tlC4tLi4gLS4tLiAtLS0tLSAtLSAulC4uLS0uLSAtlC0tLSAuLi0tLi0gLi4tLS0gLS0tLS0gLi4tLS0gLS0tLS0gLi4tLS4tlC4uLi4gLS0ulC4tlC0tlC4uLi0t
注：若解题得到的是无hgame{}字样的 flag 花括号内内容，请手动添加hgame{}后提交。
【Notice】解出来的字母均为大写

Whiteup

base64 解码，然后摩斯电码解码。
hgame{W3LC0METO2020HGAM3}

壁纸

Whiteup

1.binwalk 提取出来个加密过的 zip，然后两秒爆破数字出密码 76953815 。

2.Unicode 转中文。

hgame{Do_y0u_KnOW_uNiC0d3?}
克苏鲁神话

Whiteup

1.题目解压出一个 Bacon.txt，一个 Novel.zip。

2.zip 爆破使用明文攻击（Plain-text），明文文件选择加密后的 Bacon.zip 。得到可以直接解压的 Novel.zip 。

3.解压Novel.zip发现加密的doc文件。Bacon.txt里的大小写转换为B和A，然后培根解密。
AABABABABBAAAAAAABBAAABBBABAAAAAABBAAABBAABAAABBABABAAAABBABAAABBABBBAAAABA

解密后：FLAGHIDDENINDOC

4.doc 里一篇文章，看不到 flag ，Word：「选项」-「显示」-「隐藏文字」。拉到文章底部看到 flag 。

hgame{Y0u_h@Ve_F0Und_mY_S3cReT}

签到题ProPlus

题目

Password.txt + OK.zip

Rdjxfwxjfimkn z ts wntzi xtjrwm xsfjt jm ywt rtntwhf f y h jnsxf qjFjf jnb rg fiyykwtbsnkm tm xa jsdwqjfmkjy wlviHtqzqsGsffywjjyynf yssm xfjypnyihjn.

JRFVJYFZVRUAGMAI

- Three fences first, Five Caesar next. English sentense first, zip password next.

Whiteup

1.按提示分别操作两段字符串，栅栏三个一组解密后，凯撒移位5个，上面的会变成一段话，下面的是密码：EAVMUBAQHQMVEPDT。

2.解压看到里面的 txt 内容是一堆 ook ，网站在线解码 ，然后 base32 ，接着 base64 。

3.base64 解出来的明文会乱码，只能查看16进制，开头是89 50 4e 47，png 的文件头，复制。

4.winHex：「文件」-「新建」-「1Bytes」-「确定」-「Ctrl+V 粘贴」- 选择「Ascii Hex」然后删去开头的“00”，然后另存为 xxx.png。

5.得到二维码，扫了就出 flag。

hgame{3Nc0dInG_@lL_iN_0Ne!}