网鼎杯2020白虎组部分WriteUp

原创

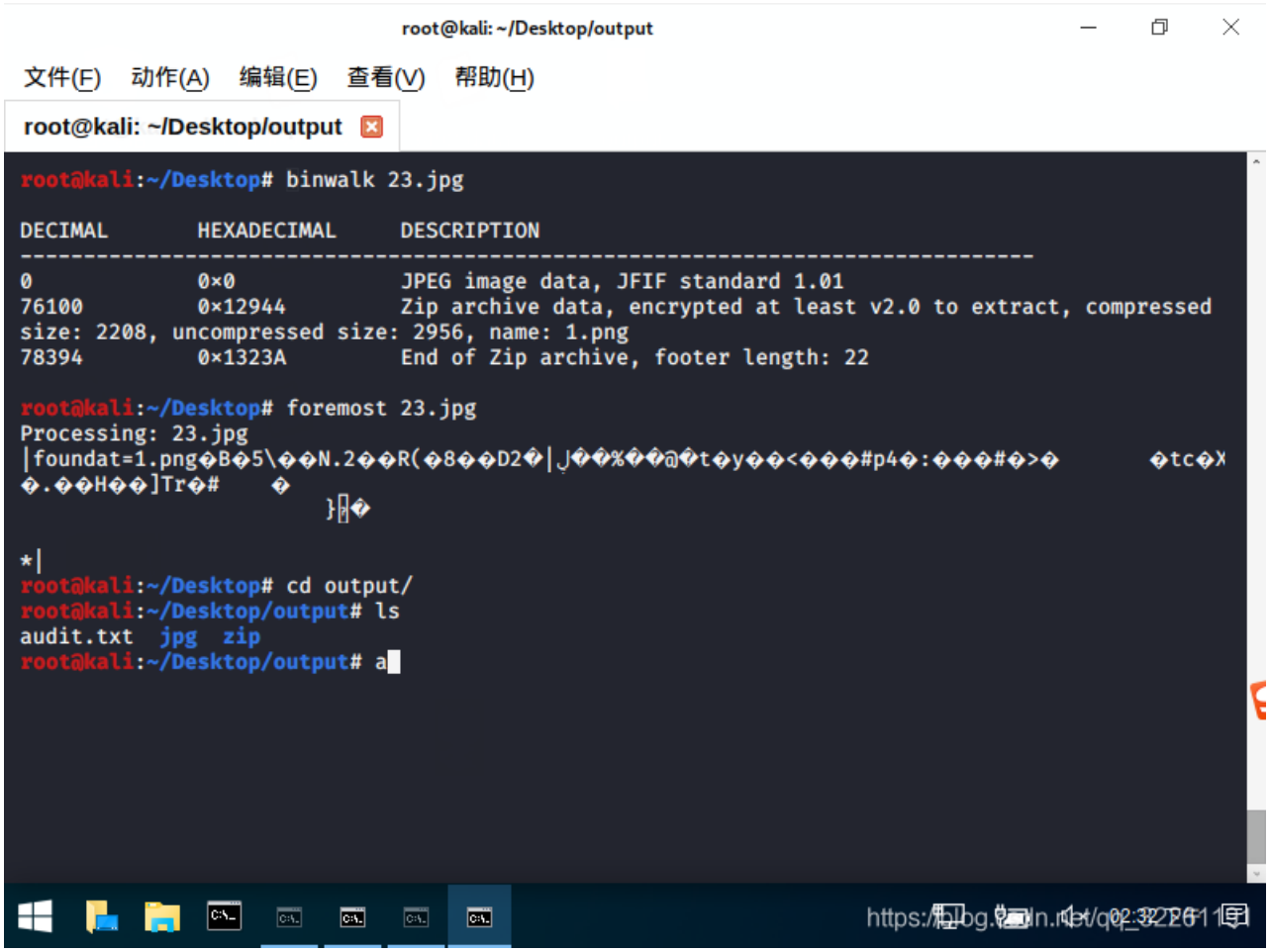
江左盟宗主 于 2020-05-15 12:32:29 发布
文章标签： CTF 网鼎杯 白虎组

部分题目：链接：https://pan.baidu.com/s/1_8iThteUQKj2jBg95nkzEQ

提取码：inrl

hidden

检测图片发现图片中包含隐藏的zip文件，然后将zip文件分离，如图：



解压发现需要输入密码，然后使用john进行枚举密码，然后解压得到另一个图片，如图：

```
root@kali:~/Desktop/output/zip# zip2john 00000148.zip >pass.txt
ver 2.0 00000148.zip/1.png PKZIP Encr: cmplen=2208, decmplen=2956, crc=364A320F
root@kali:~/Desktop/output/zip# john pass.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 3 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
1235             (00000148.zip/1.png)
1g 0:00:00:03 DONE 3/3 (2020-05-14 15:11) 0.3048g/s 148393p/s 148393c/s 148393C/s bowlart..liv2
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/Desktop/output/zip# unzip 00000148.zip
Archive:  00000148.zip
[00000148.zip] 1.png password:
  inflating: 1.png
error: invalid zip file with overlapped components (possible zip bomb)
root@kali:~/Desktop/output/zip# ls
00000148.zip  1.png  pass.txt
```


查看图片发现是二维码的一半，如图：

查看图片详细信息发现图片大小是133*67，对应十六进制85*43，将图片重命名为1.txt，使用sublime_text修改图片高度对应的值43为85，如图：

```
8950 4e47 0d0a 1a0a 0000 000d 4948 4452
0000 0085 0000 0043 0806 0000 0075 da3b
3c00 0000 0173 5247 4200 aece 1ce9 0000
0004 6741 4d41 0000 b18f 0bfc 6105 0000
0009 7048 5973 0000 0ec3 0000 0ec3 01c7
6fa8 6400 000b 2149 4441 5478 5eed 9281
6a23 c9b6 04df ffff f4be 9691 e14c 5452
a174 b5c4 ccc5 01e1 86ce ccd3 da61 ffef
e2bf 4f4a dabc 95a4 ce94 584e d8a7 a4cd
dfed f39b 397c 97a4 cd5b 49ea 4c89 e584
7d4a dafc dd3e bf99 c377 49da bc95 a4ce
9458 4ed8 a7a4 cddf edf3 9b39 7c97 a4cd
5b49 ea4c 89e5 847d 4ada fcdd 3ebf b9be
bc13 bbcf 9c1a d667 4e89 e5c4 fa96 1ba7
7b23 ddbf 9eeb cb3b b1fb cca9 617d e694
584e ac6f b971 ba37 d2fd ebb9 bebc 13bb
cf9c 1ad6 674e 89e5 c4fa 961b a77b 23dd
bf9e ebcb 3bb1 fbcc a961 7de6 9458 4eac
6fb9 71ba 37d2 fdeb b9be 9c30 3789 e584
7d4a 5267 4a52 a7d1 68fb 86dd 636e 9294
5fcf f5e5 84b9 492c 27ec 5392 3a53 923a
8d46 db37 ec1e 7393 a4fc 7aae 2f27 cc4d
6239 619f 92d4 9992 d469 34da be61 f798
9b24 e5d7 737d 3961 6e12 cb09 fb94 a4ce
94a4 4ea3 d1f6 0dbb c7dc 2429 bf9e ebcb
0973 9358 4ed8 a786 f52d 3fa5 bd6f fd36
3749 caaf e7fa 72c2 dc24 9613 f6a9 617d
cb4f 69ef 5bbf cd4d 92f2 ebb9 be9c 3037
89e5 847d 6a58 dff2 53da fbd6 6f73 93a4
fc7a ae2f 27cc 4d62 3961 9f1a d6b7 fc94
f6be f5db dc24 29bf 9eeb cb09 7393 584e
dabe 717a ef74 4fda 7bd6 676e 9294 5fcf
f5e5 84b9 492c 276d df38 bd79 ba27 ed3d
```


然后保存为png文件，打开之后显示完整的二维码，扫描二维码获得flag，如图：

hero

使用file查看发现是64位执行程序，然后首先运行程序，发现有三个选项：

1. battle with slime.
2. battle with boss.

3. go to the shop.

简单尝试发现slime的战斗力并不是固定的，如图：

```
root@kali:~/Desktop# ./hero
Day 0 , You want to:
+----------------------+
| 1.battle with slime. |
| 2.battle with boss.  |
| 3.go to the shop.    |
+----------------------+
Input the number of your chioce:1
A slime is refreshing , its Combat Effectiveness is 71
The slime is beated, you get 1 coin.
Day 1 , You want to:
+----------------------+
| 1.battle with slime. |
| 2.battle with boss.  |
| 3.go to the shop.    |
+----------------------+
Input the number of your chioce:1
A slime is refreshing , its Combat Effectiveness is 125
You die!root@kali:~/Desktop#
```


使用IDA静态分析，查看main函数发现存在outflag函数，如图：

```
1  // local variable allocation has failed, the output may be wrong!
2  int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
3  {
4    unsigned int v3; // [rsp+Ch] [rbp-4h]
5
6    init(*(_QWORD *)&argc, argv, envp);
7    eff = unief;
8    v3 = 0;
9    while ( bossexist )
10     dround(v3++);
11   outflag();
12   exit(1);
13 }
```


但是题目说不能通过batch获得flag，会影响flag值的准确性，暂时放弃直接执行outflag获取flag，查看outflag函数伪代码，如图：

```
1  int outflag()
2  {
3    byte_6034A8 = 0;
4    byte_603484 = 0;
5    byte_603489 = 0;
6    byte_603444 = 0;
7    byte_603449 = 0;
8    byte_603468 = 0;
9    return printf("flag{%s-%s-%s-%s-%s%s}", &unk_603445, flag3, flag1, flag2, flag4, &unk_603485);
10 }
```

发现程序会输出拼接的flag信息。main函数开始会调用init函数，因此查看init函数伪代码，发现flag4、unief、bossexit和coin变量的值，如图：

```
unsigned __int64 init()
{
  unsigned int v0; // eax
  signed int i; // [rsp+4h] [rbp-274Ch]
  FILE *stream; // [rsp+8h] [rbp-2748h]
  char v4; // [rsp+10h] [rbp-2740h]
  char v5; // [rsp+11h] [rbp-273Fh]
  char v6[16]; // [rsp+20h] [rbp-2730h]
  char s; // [rsp+30h] [rbp-2720h]
  unsigned __int64 v8; // [rsp+2748h] [rbp-8h]

  v8 = __readfsqword(0x28u);
  unief = 100;
  bossexist = 3;
  v0 = time(0LL);
  srand(v0);
  coin = 5;
  stream = fopen("hero", "r");
  if ( !stream )
    exit(1);
  fgets(&s, 9999, stream);
  fakemd5(&loc_401600, 214LL, v6);
  for ( i = 0; i <= 7; ++i )
  {
    snprintf(&v4, 0x20uLL, "%2.2x", (unsigned __int8)v6[i]);
    flag4[2 * i] = v4;
    flag4[2 * i + 1] = v5;
  }
  fclose(stream);
  return __readfsqword(0x28u) ^ v8;
}
```


结合main函数分析，eff=unief=100，bossexit=3，因此程序会进入while循环，并调用函数dround(v3++)，然后查看dround函数伪代码，如图：

```
unsigned __int64 __fastcall dround(unsigned int a1)
{
  int v2; // [rsp+14h] [rbp-Ch]
  unsigned __int64 v3; // [rsp+18h] [rbp-8h]

  v3 = __readfsqword(0x28u);
  printf("Day %d , You want to:\n", a1);
  puts("+----------------------+");
  puts("| 1.battle with slime.  |");
  puts("| 2.battle with boss.   |");
  puts("| 3.go to the shop.     |");
  puts("+----------------------+");
  printf("Input the number of your chioce:");
  __isoc99_scanf("%d", &v2);
  switch ( v2 )
  {
    case 1:
      slime();
      break;
    case 2:
      boss();
      break;
    case 3:
      store();
      break;
    default:
      puts("Nothing happens...");
      break;
  }
  return __readfsqword(0x28u) ^ v3;
}
```


这便是刚刚运行程序的初始信息，v2变量为输入的数字，选项1，2，3分别对应slime、boss和store函数，继续分析slime函数，发现v2的值是36到164之间的随机数，且只有当eff大于或等于v2时才会获得一个coin，如图：

```
__int64 slime()
{
  int v0; // ebx
  int v2; // [rsp+Ch] [rbp-14h]

  v0 = eff - 64;
  v2 = v0 + rand() % 128;
  if ( v2 == eff )
    ++v2;
  printf("A slime is refreshing , its Combat Effectiveness is %d\n", (unsigned int)v2);
  if ( v2 > eff )
  {
    printf("You die!");
    exit(1);
  }
  puts("The slime is beated, you get 1 coin.");
  return (unsigned int)(coin++ + 1);
}
```


接着分析boss函数，发现程序首先会判断bossexit的值，然后判断eff的值，当eff大于1000000时返回函数decrypt1的执行结果，当eff大于3000000时返回函数decrypt2的执行结果，当eff大于5000000时返回函数decrypt3的执行结果，根据提示这三个函数应该就是生成的flag的函数，如图：

```
__int64 boss()
{
  __int64 result; // rax

  if ( bossexist <= 2 )
  {
    if ( bossexist == 2 )
    {
      puts("The dragon appears, its Combat Effectiveness is 3000000.");
      if ( eff <= 3000000 )
      {
        printf("You die!");
        exit(1);
      }
      puts("The dragon is beated! flag is partly decrypted...");
      --bossexist;
      result = decrypt2("The dragon is beated! flag is partly decrypted...");
    }
    else
    {
      result = (unsigned int)bossexist;
      if ( bossexist <= 1 )
      {
        puts("The dragon appears, its Combat Effectiveness is 5000000.");
        if ( eff <= 5000000 )
        {
          printf("You die!");
          exit(1);
        }
        puts("The dragon is beated! combining flag and print...");
        --bossexist;
        result = decrypt3("The dragon is beated! combining flag and print...");
      }
    }
  }
  else
  {
    puts("The dragon appears, its Combat Effectiveness is 1000000.");
    if ( eff <= 1000000 )
    {
      printf("You die!");
      exit(1);
    }
    puts("The dragon is beated! flag is partly decrypted...");
    --bossexist;
    result = decrypt1("The dragon is beated! flag is partly decrypted...");
  }
  return result;
}
```


然而eff的值只有100，显然不能获得flag。然后查看decrypt1函数，发现调用des函数，并将结果按格式复制到变量flag1，如图：

```
int decrypt1()
{
  __int64 v0; // ST18_8

  v0 = des(-1030164995884238963LL, -1030164995884238963LL, 101LL);
  return snprintf(flag1, 0x10uLL, "%016lx\n", v0);
}
```

查看des函数应该是数据解密，发现没涉及到变量eff，然后查看decrypt2函数，也没有涉及到eff变量，如图：

```
int decrypt2()
{
  signed __int64 v0; // ST18_8

  v0 = des(0x5DD77756631856ADuLL, 0x5DD77756631856ADuLL, 101);
  return snprintf(flag2, 0x10uLL, "%016lx\n", v0);
}
```

查看decrypt3函数，发现涉及到变量unief，如图：

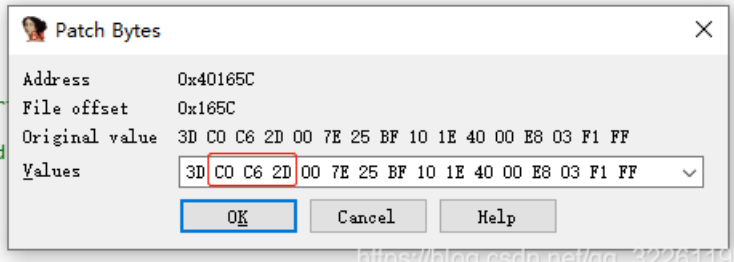
```
int decrypt3()
{
  signed __int64 v0; // ST18_8

  v0 = des(unief, unief, 101);
  return snprintf(flag3, 0x10uLL, "%016lx\n", v0);
}
```

因此也不能通过修改unief的值大于5000000来获得flag。既然不能通过修改unief的值来获取flag，那就修改和eff比较的值小于eff，从而达到获取flag的目的，如图：

```
__int64 boss()
{
  __int64 result; // rax

  if ( bossexist <= 2 )
  {
    if ( bossexist == 2 )
    {
      puts("The dragon appears, its Combat Effectiveness is 3000000.");
      if ( eff <= 3000000 )
      {
        printf("You die!");
        exit(1);
      }
      puts("The dragon is beated! flag is par
      --bossexist;
      result = decrypt2("The dragon is beated
    }
    else
    {
      result = (unsigned int)bossexist;
      if ( bossexist <= 1 )
```



使用Edit-->Patch program-->Change bytes修改3000000的值（十六进制位0x2dc6c0）为0，修改5000000和1000000的值也为0，然后使用Edit-->Patch program-->Apply patches to input file应用修改，然后运行程序获得flag，如图：

```
root@kali:~/Desktop# ./hero
Day 0 , You want to:
+----------------------+
| 1.battle with slime. |
| 2.battle with boss.  |
| 3.go to the shop.    |
+----------------------+
Input the number of your chioce:2
The dragon appears, its Combat Effectiveness is 1000000.
The dragon is beated! flag is partly decrypted...
Day 1 , You want to:
+----------------------+
| 1.battle with slime. |
| 2.battle with boss.  |
| 3.go to the shop.    |
+----------------------+
Input the number of your chioce:2
The dragon appears, its Combat Effectiveness is 3000000.
The dragon is beated! flag is partly decrypted...
Day 2 , You want to:
+----------------------+
| 1.battle with slime. |
| 2.battle with boss.  |
| 3.go to the shop.    |
+----------------------+
Input the number of your chioce:2
The dragon appears, its Combat Effectiveness is 5000000.
The dragon is beated! combining flag and print...
flag{0259-6430-726f077b-5959-40b7dde
```

但是获得的flag貌似还是不正确。

感谢评论区qq_26143017提供的新方法——整数溢出。

查看store函数伪代码发现程序使用scanf输入数字到v1变量，然后强制转换v1的数据类型为int型并与0比较，如果大于0就将coin - 2×v1的值赋值给v2，如图：

```
1  unsigned __int64 store()
2  {
3    unsigned int v1; // [rsp+0h] [rbp-10h]
4    int v2; // [rsp+4h] [rbp-Ch]
5    unsigned __int64 v3; // [rsp+8h] [rbp-8h]
6
7    v3 = __readfsqword(0x28u);
8    puts("2 coins to upgrade 1 point of Combat Effectiveness");
9    printf("Your coins:%d\n", (unsigned int)coin);
10   printf("Your Combat Effectiveness is :%d\n", (unsigned int)eff);
11   printf("input the points of you want to upgrade:");
12   __isoc99_scanf("%d", &v1);
13   if ( (int)v1 > 0 )
14   {
15     v2 = coin - 2 * v1;
16     if ( v2 < 0 )
17     {
18       puts("Your coins is not enough.");
19     }
20     else
21     {
22       eff += v1;
23       coin = v2;
24       printf("Your Combat Effectiveness upgraded %d points, now it is %d points.\n", v1, (unsigned int)eff);
25       printf("Your coins:%d\n", (unsigned int)coin);
26     }
27   }
28   else
000017AC store:28 (4017AC)
```

而int型能表示的十进制数据范围是-2147483648~2147483647，因此当v2 > 2147483647会造成整数溢出，v2会成为负数，所以当v2 mod 2147483648 ≤ 2147483647时，可以执行充值硬币和攻击力，即(5−2×v1) mod 2147483648 ≤ 2147483647，即当v1 mod 2147483648 ≥ 1073741827时可执行eff += v1，coin += v2。但是v1也不能太大，当eff + v1 > 2147483647时同样会造成溢出，导致eff成为负数，因此当(eff + v1) mod 2147483648 ≤ 2147483647时才可以使eff > 0，即v1 mod 2147483648 ≤ 2147483547时eff > 0，如图：

```
root@kali:~/Desktop# ./hero
Day 0 , You want to:
+----------------------+
| 1.battle with slime. |
| 2.battle with boss.  |
| 3.go to the shop.    |
+----------------------+
Input the number of your chioce:3
2 coins to upgrade 1 point of Combat Effectiveness
Your coins:5
Your Combat Effectiveness is :100
input the points of you want to upgrade:1073741827
Your Combat Effectiveness upgraded 1073741827 points, now it is 1073741927 p
oints.
Your coins:2147483647
Day 1 , You want to:
+----------------------+
| 1.battle with slime. |
| 2.battle with boss.  |
| 3.go to the shop.    |
+----------------------+
Input the number of your chioce:
```


然后battle with boss三次即可，如图：

```
Day 1 , You want to:
+----------------------+
| 1.battle with slime. |
| 2.battle with boss.  |
| 3.go to the shop.    |
+----------------------+
Input the number of your chioce:2
The dragon appears, its Combat Effectiveness is 1000000.
The dragon is beated! flag is partly decrypted ...
Day 2 , You want to:
+----------------------+
| 1.battle with slime. |
| 2.battle with boss.  |
| 3.go to the shop.    |
+----------------------+
Input the number of your chioce:2
The dragon appears, its Combat Effectiveness is 3000000.
The dragon is beated! flag is partly decrypted ...
Day 3 , You want to:
+----------------------+
| 1.battle with slime. |
| 2.battle with boss.  |
| 3.go to the shop.    |
+----------------------+
Input the number of your chioce:2
The dragon appears, its Combat Effectiveness is 5000000.
The dragon is beated! combining flag and print ...
flag{0259-6430-726f077b-5959-bf477a78c83b}
```


还有大佬给思路说强制调用decrypt函数然后将flag拼接即可，但目前还没搞定。