Secret_phase

2020年5月5日 20:25

容易找到入口处在phase_defused中：

```
0000000000004015c4 <phase_defused>:
  4015c4: 48 83 ec 78           sub    $0x78,%rsp
  4015c8: 64 48 8b 04 25 28 00  mov    %fs:0x28,%rax
  4015cf: 00 00
  4015d1: 48 89 44 24 68        mov    %rax,0x68(%rsp)
  4015d6: 31 c0                 xor    %eax,%eax
  4015d8: 83 3d 81 21 20 00 06  cmpl   $0x6,0x202181(%
rip)        # 603760 <num_input_strings>
  4015df: 75 5e                 jne    40163f <phase_defused+0x7b>
  4015e1: 4c 8d 44 24 10        lea    0x10(%rsp),%r8
  4015e6: 48 8d 4c 24 0c        lea    0xc(%rsp),%rcx
  4015eb: 48 8d 54 24 08        lea    0x8(%rsp),%rdx
  4015f0: be 19 26 40 00        mov    $0x402619,%esi
  4015f5: bf 70 38 60 00        mov    $0x603870,%edi
  4015fa: e8 f1 f5 ff ff        callq  400bf0 <__isoc99_sscanf@plt>
  4015ff: 83 f8 03              cmp    $0x3,%eax
  401602: 75 31                 jne    401635 <phase_defused+0x71>
  401604: be 22 26 40 00        mov    $0x402622,%esi
  401609: 48 8d 7c 24 10        lea    0x10(%rsp),%rdi
  40160e: e8 25 fd ff ff        callq  401338 <strings_not_equal>
  401613: 85 c0                 test   %eax,%eax
  401615: 75 1e                 jne    401635 <phase_defused+0x71>
  401617: bf f8 24 40 00        mov    $0x4024f8,%edi
  40161c: e8 ef f4 ff ff        callq  400b10 <puts@plt>
  401621: bf 20 25 40 00        mov    $0x402520,%edi
  401626: e8 e5 f4 ff ff        callq  400b10 <puts@plt>
  40162b: b8 00 00 00 00        mov    $0x0,%eax
  401630: e8 0d fc ff ff        callq  401242 <secret_phase>
  401635: bf 58 25 40 00        mov    $0x402558,%edi
  40163a: e8 d1 f4 ff ff        callq  400b10 <puts@plt>
  40163f: 48 8b 44 24 68        mov    0x68(%rsp),%rax
  401644: 64 48 33 04 25 28 00  xor    %fs:0x28,%rax
```

0x402619: "%d %d %s"

0x603870 <input_strings+240>: "7 0 "

这是可以看出这是phase_4的输入，结合sscanf的第一个参数我们应当紧跟着输入一个字符串

0x402622: "DrEvil"

可得到要求的字符串

可以猜出输入了六个字符串后才不会跳过中间部分

随后我们进入了secret_phase:

```
0000000000401242 <secret_phase>:
  401242: 53                    push   %rbx
  401243: e8 56 02 00 00        callq  40149e <read_line>
  401248: ba 0a 00 00 00        mov    $0xa,%edx
  40124d: be 00 00 00 00        mov    $0x0,%esi
  401252: 48 89 c7              mov    %rax,%rdi
  401255: e8 76 f9 ff ff        callq  400bd0 <strtol@plt>
  40125a: 48 89 c3              mov    %rax,%rbx
  40125d: 8d 40 ff              lea    -0x1(%rax),%eax
  401260: 3d e8 03 00 00        cmp    $0x3e8,%eax
  401265: 76 05                 jbe    40126c <secret_phase+0x2a>
  401267: e8 ce 01 00 00        callq  40143a <explode_bomb>
  40126c: 89 de                 mov    %ebx,%esi
  40126e: bf f0 30 60 00        mov    $0x6030f0,%edi
  401273: e8 8c ff ff ff        callq  401204 <fun7>
  401278: 83 f8 02              cmp    $0x2,%eax
  40127b: 74 05                 je     401282 <secret_phase+0x40>
  40127d: e8 b8 01 00 00        callq  40143a <explode_bomb>
  401282: bf 38 24 40 00        mov    $0x402438,%edi
  401287: e8 84 f8 ff ff        callq  400b10 <puts@plt>
  40128c: e8 33 03 00 00        callq  4015c4 <phase_defused>
  401291: 5b                    pop    %rbx
  401292: c3                    retq
```

读取一行输入转化为十进制数字

数字小于1001

需要fun7的返回值等于2，其第一个参数似乎是一个指针

再来看fun7：

```
0000000000401204 <fun7>:
  401204: 48 83 ec 08           sub    $0x8,%rsp
  401208: 48 85 ff              test   %rdi,%rdi
  40120b: 74 2b                 je     401238 <fun7+0x34>
  40120d: 8b 17                 mov    (%rdi),%edx
  40120f: 39 f2                 cmp    %esi,%edx
  401211: 7e 0d                 jle    401220 <fun7+0x1c>
  401213: 48 8b 7f 08           mov    0x8(%rdi),%rdi
  401217: e8 e8 ff ff ff        callq  401204 <fun7>
  40121c: 01 c0                 add    %eax,%eax
  40121e: eb 1d                 jmp    40123d <fun7+0x39>
  401220: b8 00 00 00 00        mov    $0x0,%eax
  401225: 39 f2                 cmp    %esi,%edx
  401227: 74 14                 je     40123d <fun7+0x39>
  401229: 48 8b 7f 10           mov    0x10(%rdi),%rdi
  40122d: e8 d2 ff ff ff        callq  401204 <fun7>
  401232: 8d 44 00 01           lea    0x1(%rax,%rax,1),%eax
  401236: eb 05                 jmp    40123d <fun7+0x39>
  401238: b8 ff ff ff ff        mov    $0xffffffff,%eax
  40123d: 48 83 c4 08           add    $0x8,%rsp
  401241: c3                    retq
```

又见指针套指针，指针指向数据，偏移0x8与0x10为另外的指针，一套套俩，似乎是树。这里列出内存：分别是地址，值，两个指针。仔细看是一颗搜索二叉树

0x6030f0	0x24
0x603110	0x603130

0x603110	0x8
0x603190	0x603150

0x603130	0x32
0x603170	0x6031b0

0x603190	0x6
0x6031f0	0x603250

0x603150	0x16
0x603270	0x603230

0x603170	0x2d
0x6031d0	0x603290

0x6031b0	0x6b
0x603210	0x6032b0

0x6031f0	0x1

0x603250	0x7

0x603270	0x14

0x603230	0x23

0x6031d0	0x28

0x603290	0x2f

0x603210	0x63

0x6032b0	0x3e9

最后我们得到fun7：

```
struct node{
    int n;
    struct node *left,*right;
};
int fun7(int n,struct node *ptr){
    if(0==ptr)
        return 0xffffffff;
    else{
        if(n<=ptr->n)
            if(n==ptr->n)
                return 0;
            else
                return 2*fun7(n,ptr->left);
        else
            return 2*fun7(n,ptr->right)+1;
    }
}
```

如果值与节点相等就返回0，小于就返回2*rax,大于就返回2*rax+1

2==((2*0)+1)*2

故是根节点的左子树的右子树，就是0x16(22)