20307130135李钧实验5

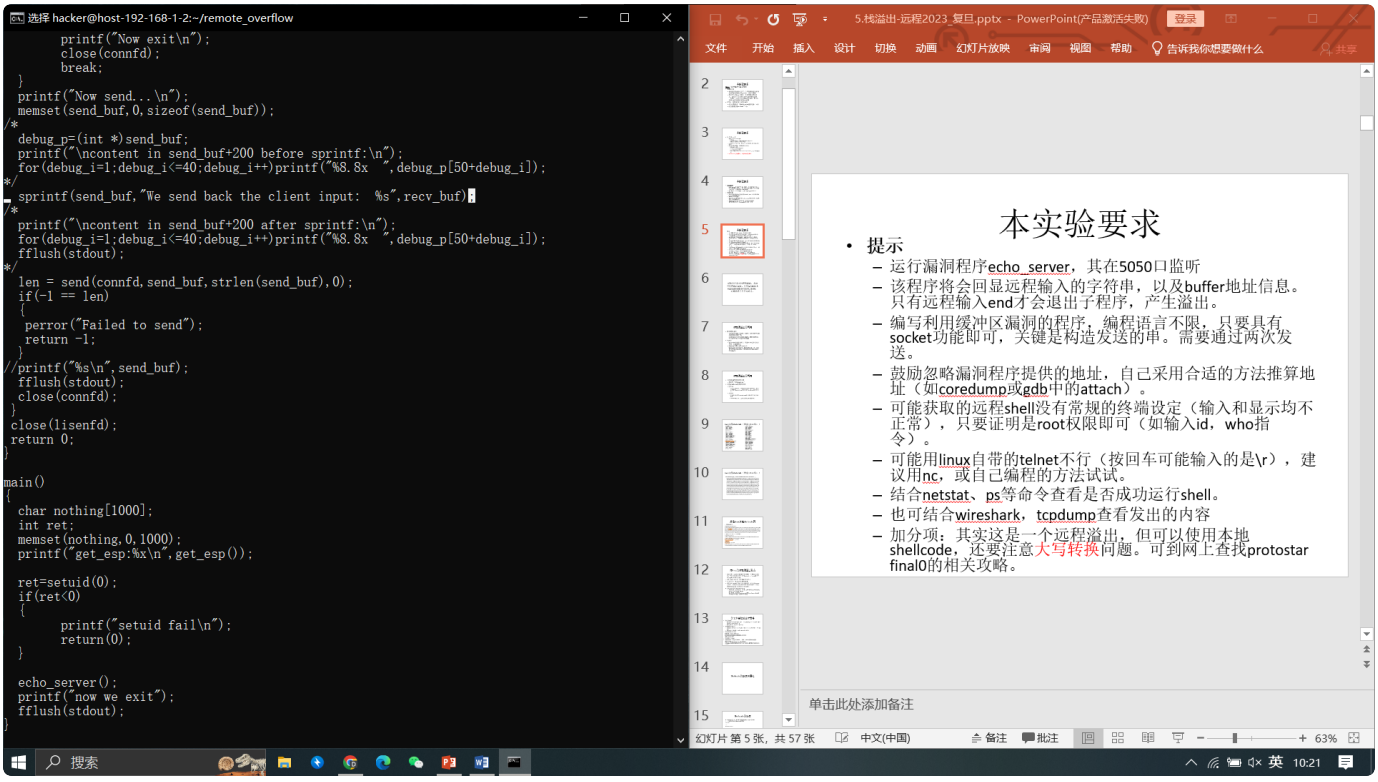
20307130135 李钧

一、实验目的

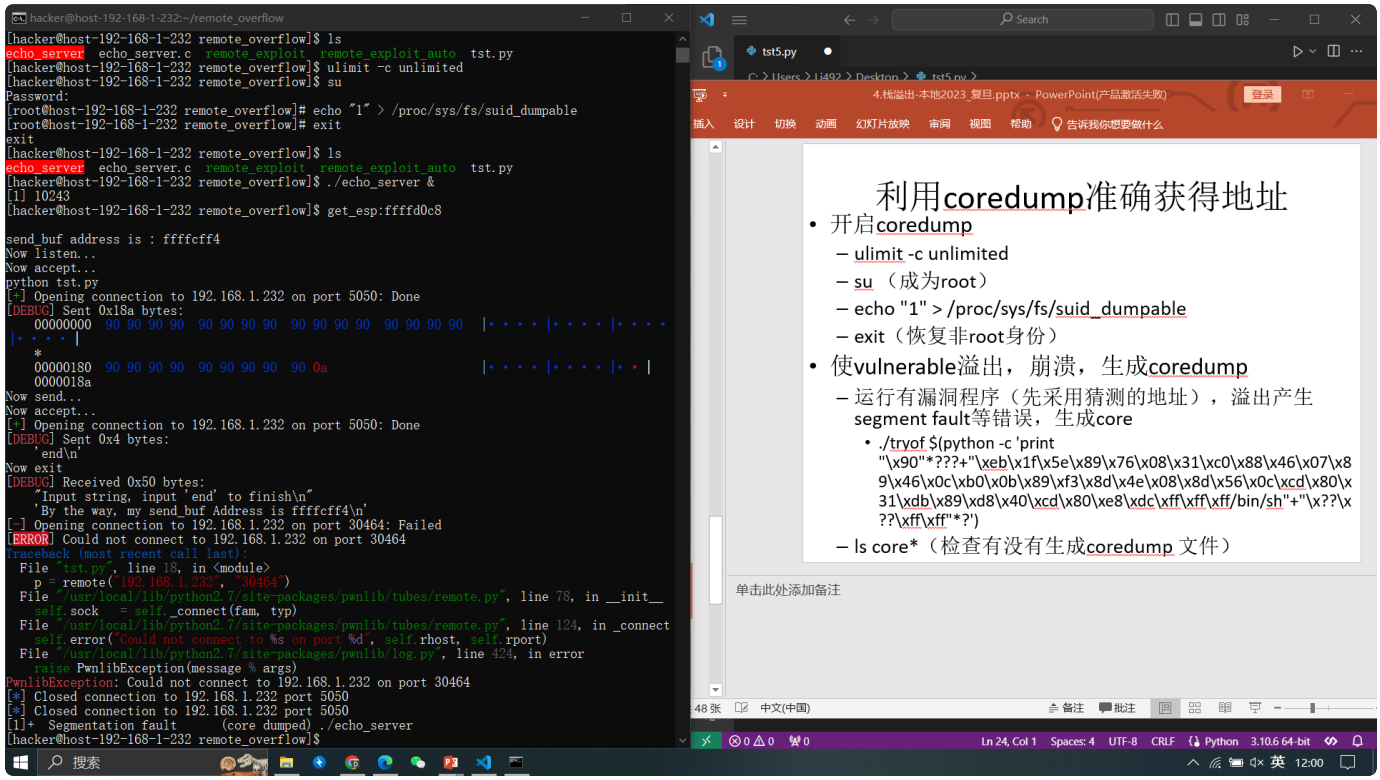
1. 本实验所用虚拟机上有一个具有缓冲区语出漏洞的可执行程序echo_server，具有's'属性
2. 要求学生在linux上编程，并利用相关调试工具，编写出一个利用缓冲区漏洞的exploit（远程），获得带有root权限的shell，要求在获得shell后证明已经是root权限
3. 成功远程溢出，使得echo_server崩溃退出：20分；成功获得远程rootshell：10分

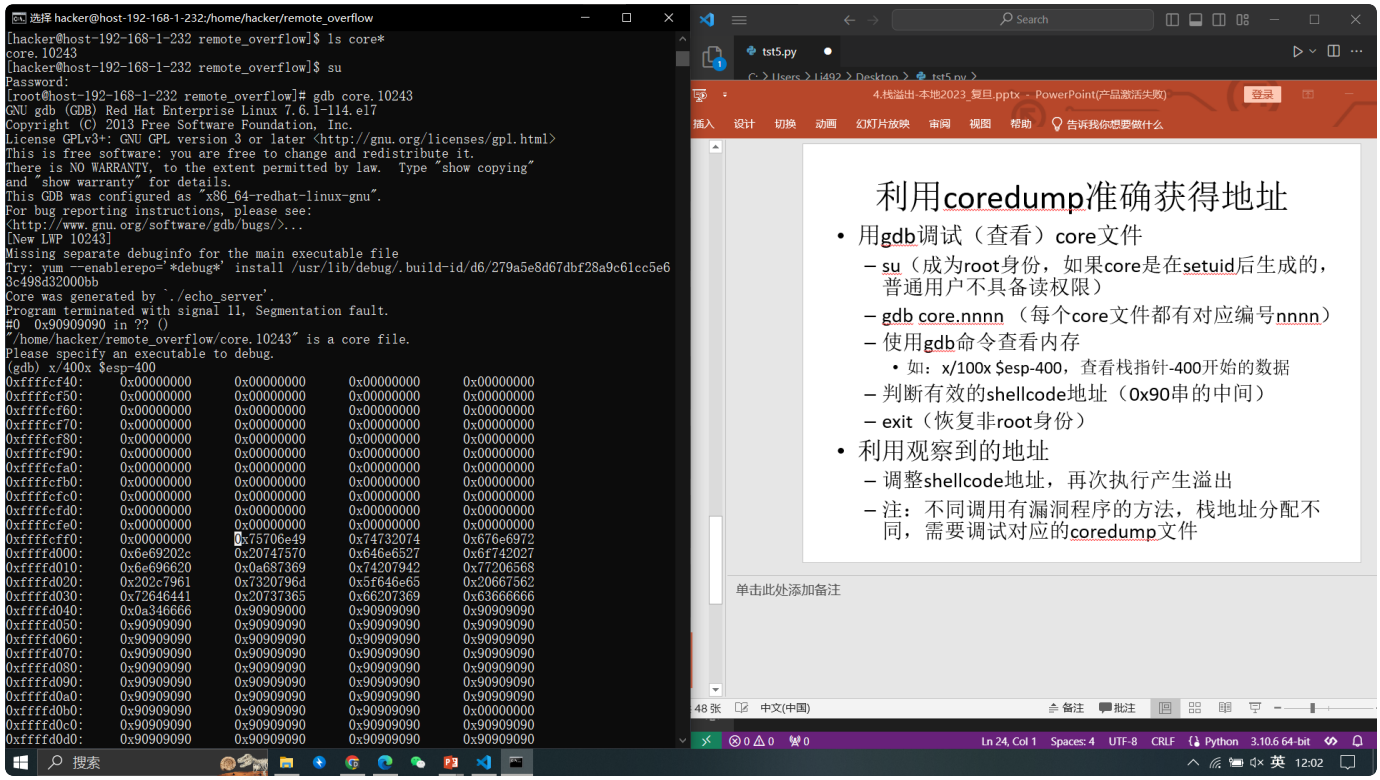
二、实验步骤

1）观察echo_server.c代码中的echo_server()函数，可以找到退出程序需要远程输入字符串"end"，还可以找到server返回后sprintf()函数打印了"We send back the client input: %s"，而且buf数组的长度设置成了200.



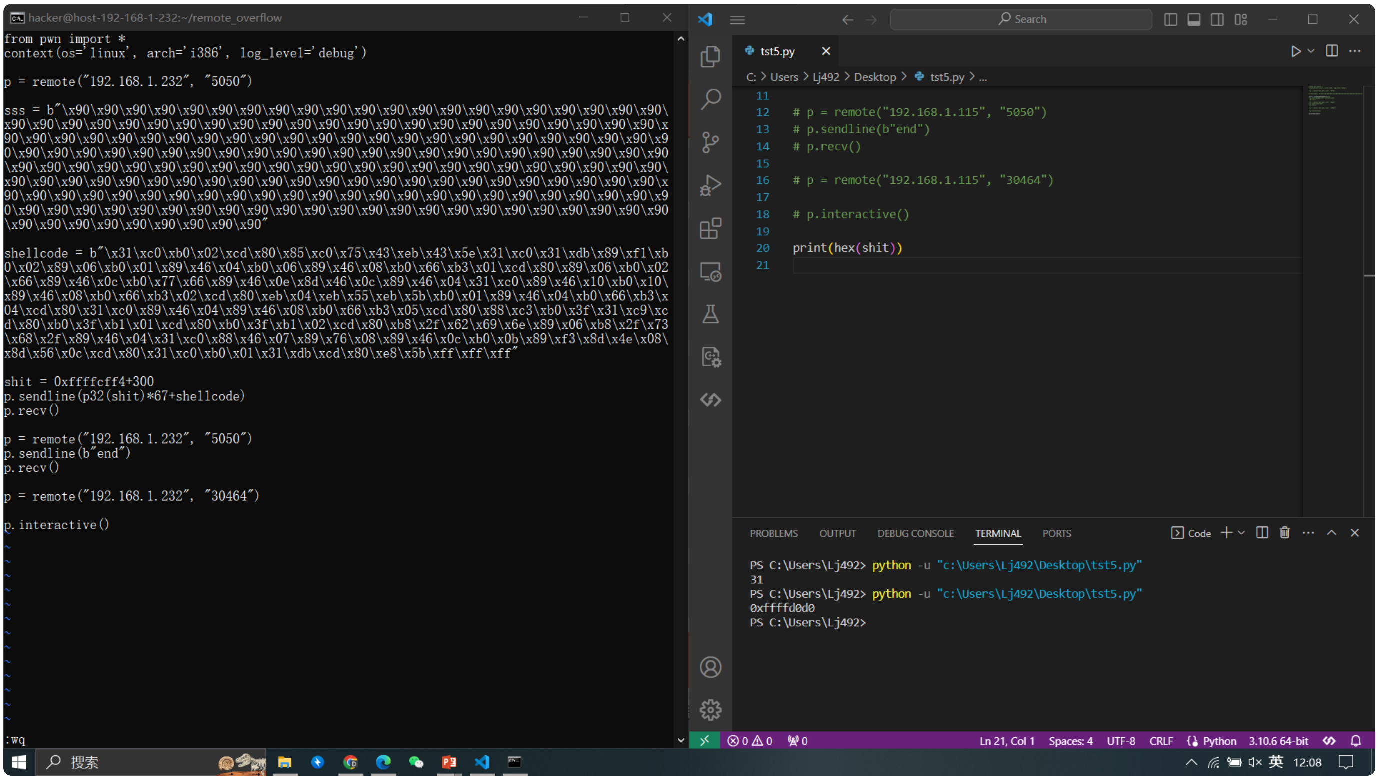
2）为了能够定位到构造的shellcode的位置，本人首先打算通过上次实验用到的coredump方法找到ret位置应该跳转到的地址，通过构造一串很长的\x90数据发送给server使其溢出崩溃，使用gdb调试由程序崩溃生成的core文件，如下图所示



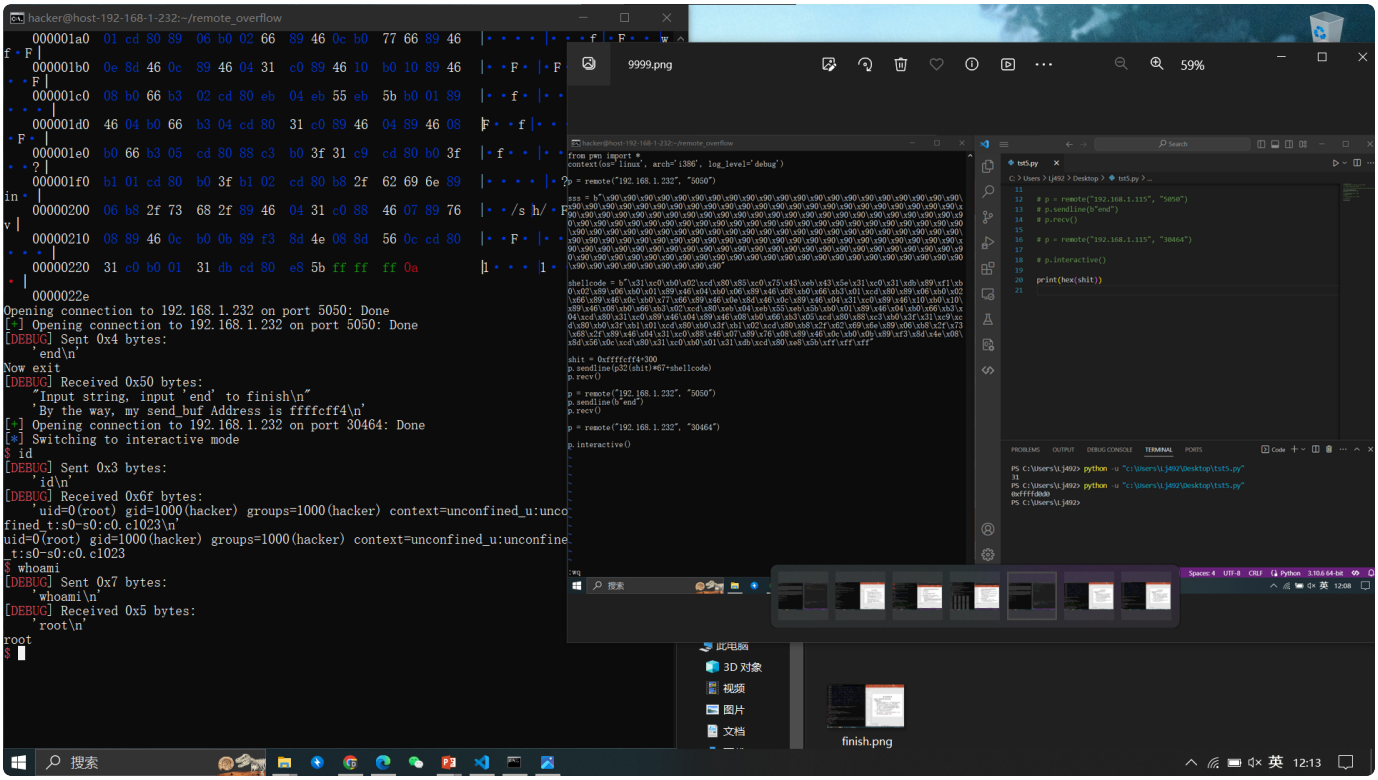


3）通过观察调试出来的数据，我们可以很容易得到buf起始地址为0xffffcff4，而其开头部分的32字节并非我们填充的\x90，而是填充了上述提到的"We send..."字符串。由于我们构造的\x90长度可以手动设置，紧随其后的shellcode地址就可以通过构造的填充长度和buf起始地址确定

例如本人想要在buf起始地址之后300字节位置填上shellcode，由于该长度远大于buf数组大小，故填充的内容一定可以覆盖到ret地址，能够实现跳转。为方便操作，直接将填充的内容设置为多个shellcode起始地址，填充长度由300/4–32/8=67确定，即需要填充67个0xffff----



4）通过指令./echo_server & 使得该服务器程序可以在后台运行，接着运行上图所示的python恶意程序，便可达到攻击成功的目的



三、实验总结反思

防范这类远程栈溢出应该防止远程用户控制服务端的send_buf，如果需要回显，可以通过strncpy这类限制长度的函数将recv_buf的内容复制到send_buf，避免栈溢出，不应不限长度。

此外还可以设置地址随机，栈不可执行和canary之类的手段避免栈溢出攻击。