20307130135实验6

20307130135 李钧

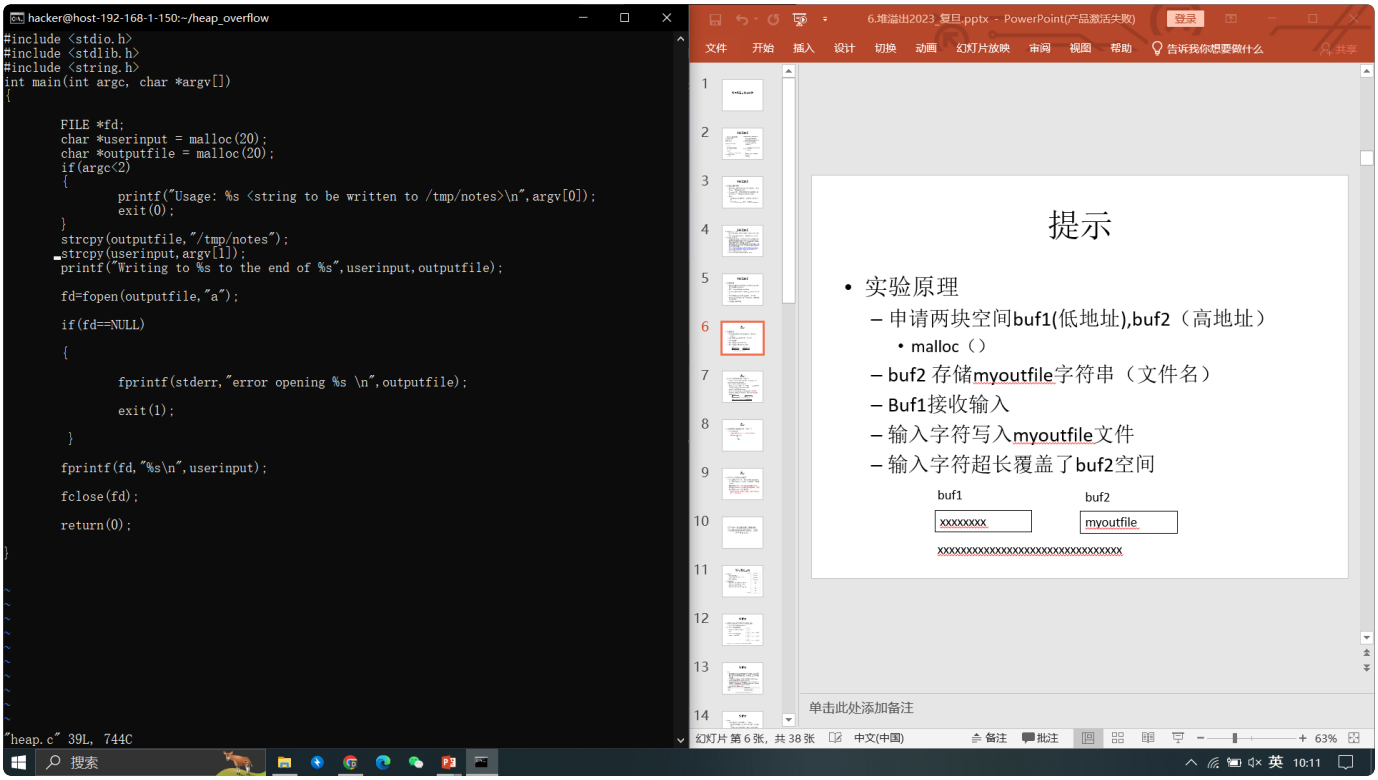
一、实验目的

1. 假定heap.c编译成为heap可执行程序后，属主为root，具备suid标志位
2. 以hacker登录，利用此程序的漏洞，进行堆溢出修改重要文件，实现获得root权限的目的

二、实验过程

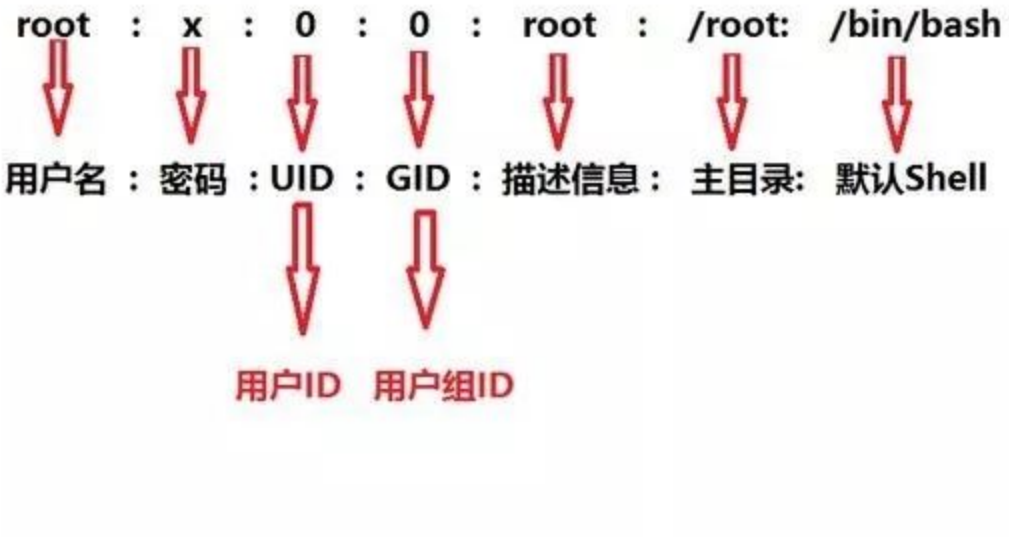
查看源代码

首先进入heap_overflow文件夹中查看heap.c文件，容易观察到该程序连续使用了两次malloc函数，并且连续使用了两次不安全的函数strcpy，该函数不会检查输入的范围。故当我们输入的userinput超过设置的范围一定长度时，将会覆盖到outputfile，改变fopen()打开的文件，并进行文件的写入。



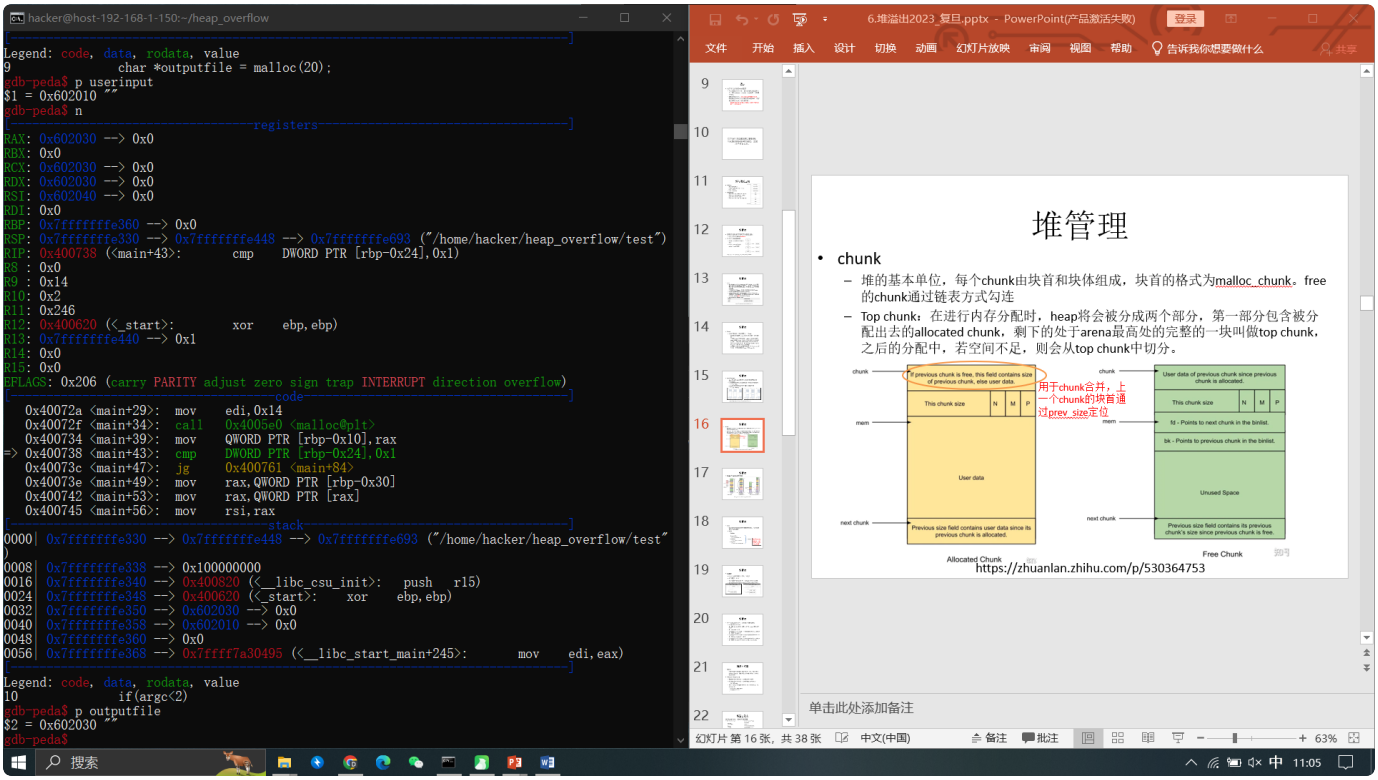
分析重要文件/etc/passwd

为了达到以hacker登录并利用堆溢出漏洞获取root权限，我们可以在/etc/passwd中写入内容，新增一个有root权限的用户，再通过switch user获取root权限。如下图所示，/etc/passwd中每一项都以 `root:x:0:0:root:/root:/bin/bash` 的形式记录，当x为空表示不需要密码，UID设置为0表示具有root权限。



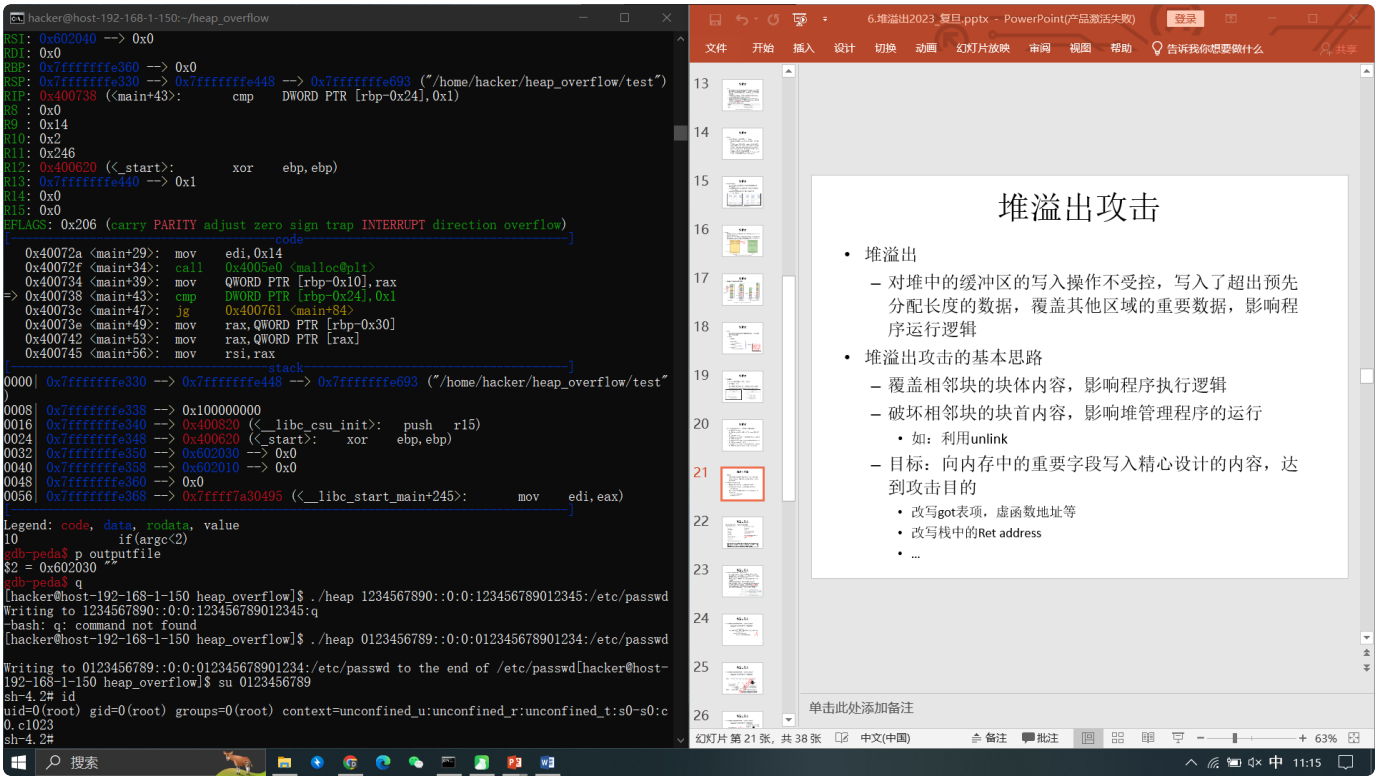
确定恶意输入的长度

为了得到能够成功攻击的构造输入长度，我通过 `gcc -g -o test heap.c` 指令重新编译了源文件，并通过gdb调试，在两次malloc的行位置设置断点，并一条条运行指令，程序执行完两次malloc操作之后直接通过 `p userinput` 和 `p outputfile` 指令打印出两次分配的堆地址起始位置。如下图所示，我们得到userinput的起始地址是0x602010，outputfile的起始地址为0x602030，即两者地址之差为0x20，相差了32个字节。为了达到攻击目的，我们需要构造32个字节长度的输入，并在恶意输入后跟上/etc/passwd来覆盖outputfile打开的文件。



构造输入进行攻击

在确定了需要构造的输入长度（32字节）之后，我们只需要在运行./heap时跟上足够长度的恶意输入作为参数，即可达到攻击目的，如下图所示，通过攻击可以确定我们已经获得了root权限，使用的输入参数为 `0123456789::0:0:012345678901234:/etc/passwd`



三、实验总结

为什么heap溢出不需要调用setuid函数?

1. Heap溢出攻击的目的：Heap溢出攻击的目的通常是修改程序的内存中的数据或控制程序的执行流程，以达到攻击者的目标，例如执行恶意代码或获取未授权的访问权限。它通常不是用于提升特权（如root权限），而是用于绕过安全措施或执行恶意操作。
2. setuid函数的作用：setuid函数用于修改进程的有效用户ID，以实现提权操作。它通常用于允许进程在运行时以不同于其启动用户的权限执行某些特定任务。通常情况下，只有具有特权的用户（例如root用户）才能成功调用setuid函数，以降低进程的权限级别。Heap溢出攻击的目的通常不是提升特权，而是攻击程序本身的漏洞。
3. Heap溢出的攻击方式：Heap溢出攻击通常包括向动态分配的内存区域写入超出分配大小的数据，以覆盖有关控制流程的信息或者修改数据。攻击者试图利用这种行为来实现他们的目标，而不是通过调用setuid函数来提权。