**学生实验报告**

**开课学院及实验室： 电子楼416 2022年 4 月 13 日**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学院** | 网络空间安全学院 | **年级/专业/班** | 网安211 | **姓名** |  | **学号** |  |
| **实验课程名称** | 计算机系统基础实验 | | | | | **成绩** |  |
| **实验项目名称** | 实验4 缓冲区溢出实验 | | | | | **指导老师** |  |

**一、实验目的**

1、深入了解缓冲区溢出的隐患，了解如何利用缓冲区溢出这个漏洞对现有程序进行控制流劫持、执行非法程序代码，从而造成对程序进行攻击以及破坏的过程；

2、增强对程序机器级表示、汇编语言、调试器和逆向工程等理解。

**二、实验内容**

对目标程序实施缓冲区溢出攻击，通过造成缓冲区溢出来破坏目标程序的栈帧结构，继而执行一些原来程序中没有的行为。

本实验共完成以下2个实验小题：level1和level2。

1、准备工作

在所发目录中，一共3个和本实验相关的文件：

ctarget 代码注入攻击的目标文件

hex2row 将16进制数转化为攻击字符，因为有些字符在屏幕上面无法输入，所以输入该字符的16进制数，自动转化为该字符

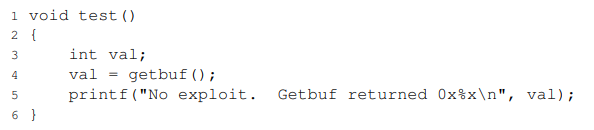
cookie.txt 一个8为16进行数，作为攻击的特殊标志符（level2使用）

其中，ctarget为所要分析处理的文件。其中包含3个函数：getbuf( )和touch1、touch2。可用objdump工具反汇编查看其代码，并存入文件中方便查看。命令为：

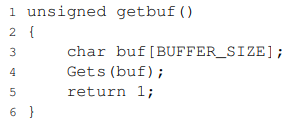
objdump -d ctarget > ctarget.txt

2、level1的内容和要求

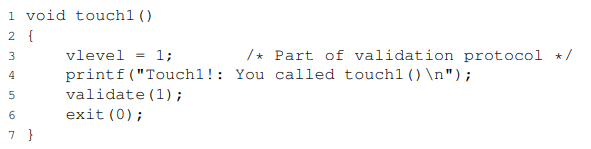
ctarget 的正常流程如下：



其中getbuf( )函数定义为：



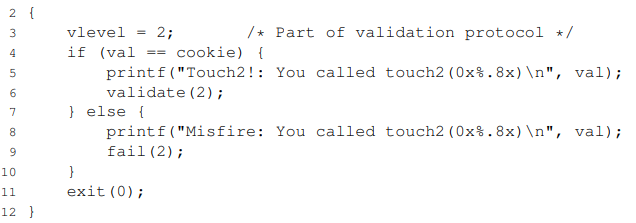
Gets( )的功能是输入字符串到buf中，但没有检查字符串的长度是否在BUFFER\_SIZE内。本题要求通过适当的输入，利用getbuf( )的漏洞使程序在test函数中重新定向到touch1函数。touch1函数定义如下：



3、level2的内容和要求

同level1的整体流程，即在getbuf中输入字符，拦截程序流，跳转到调用touch2函数。touch2函数定义如下：





这段程序验证传进来的参数val是否和cookie中值相等。本程序需要做的是在输入字符串中注入一小段代码，代码传入val参数到%rdi，其中val是和cookie一样的值，然后正确跳转到touch2。

**三、实验工具和环境**

软件工具：gdb调试器和objdump；

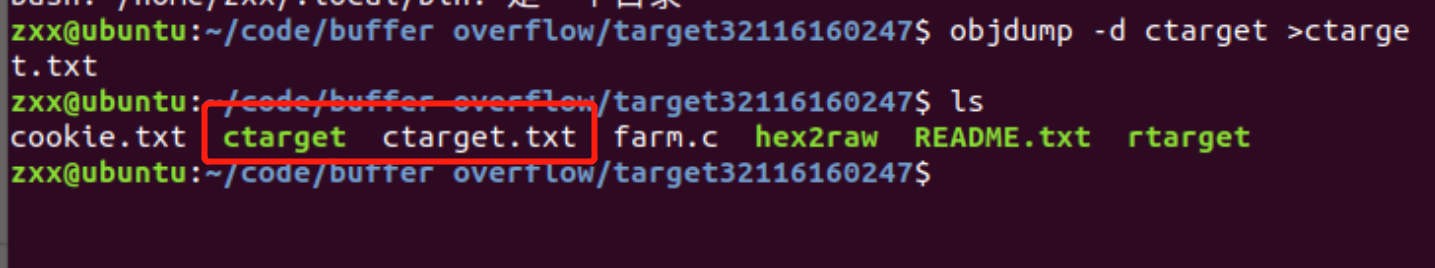
实验语言：C语言，AT&T汇编语言

实验环境： linux

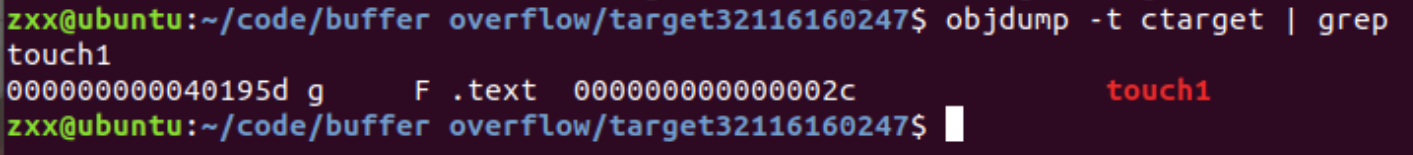
**四、过程分析**

Level 1：

1. 用objdump工具反汇编出ctarget.txt文件

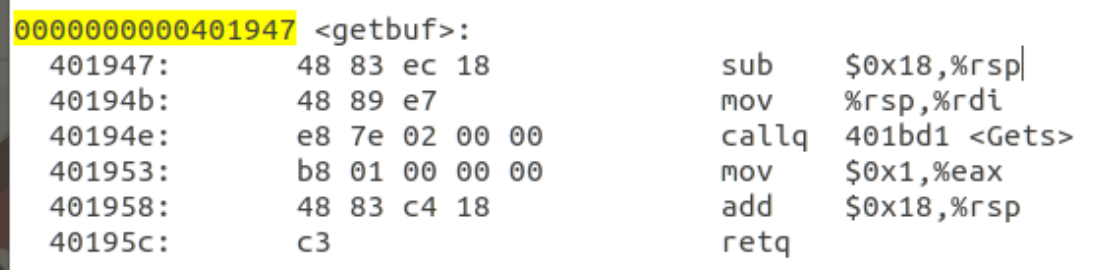


1. 用objdump工具找到touch1()，getbuf(),test()的地址





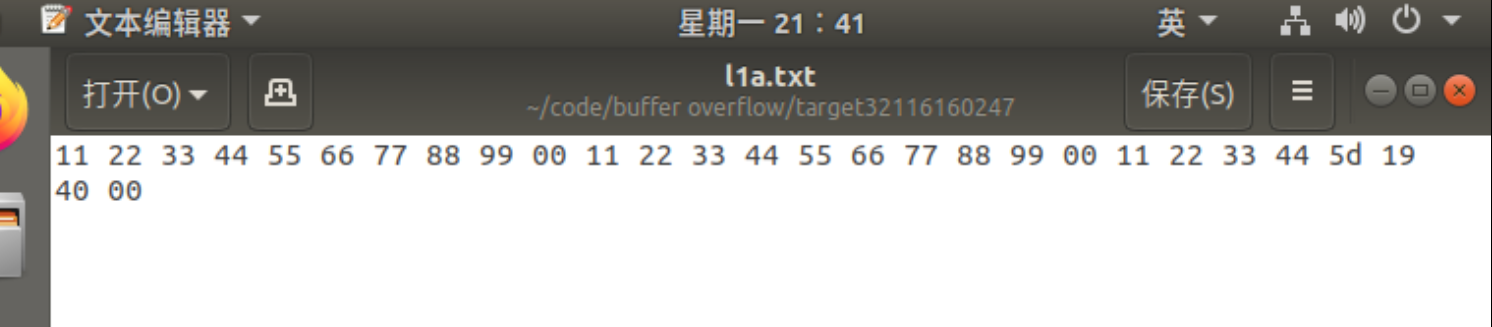
1. 算出getbuf()的缓冲区大小

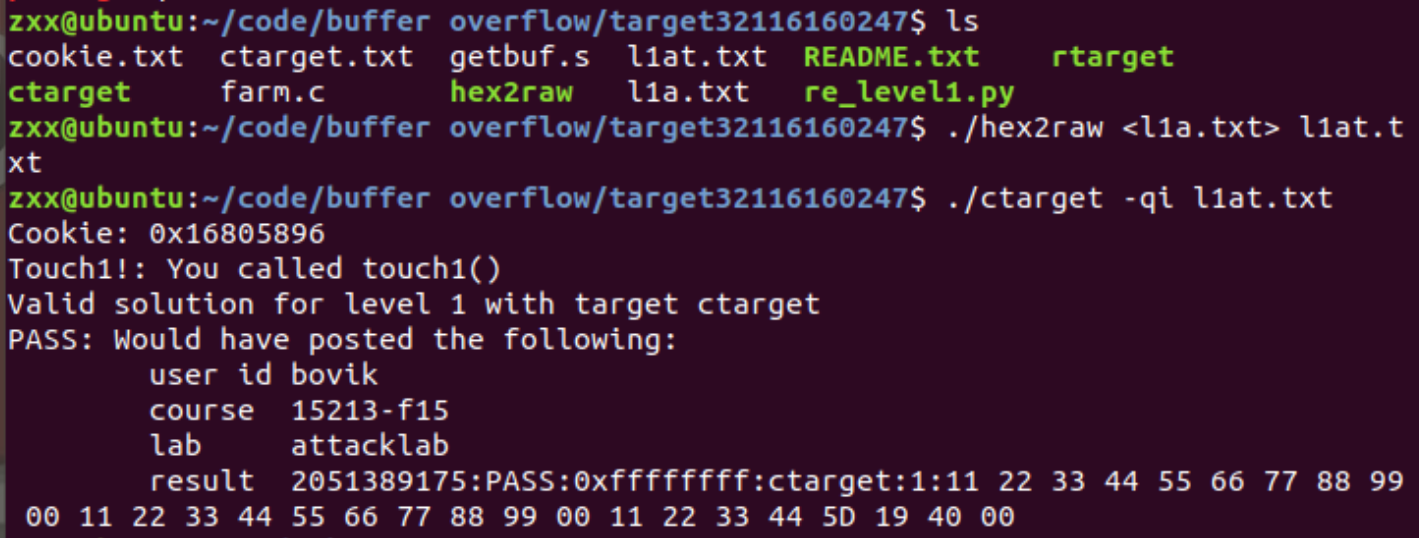


值 0x18以十六进制表示缓冲区的大小。将此值转换为十进制以获得缓冲区大小，24字节

1. 计算必要的输入长度：

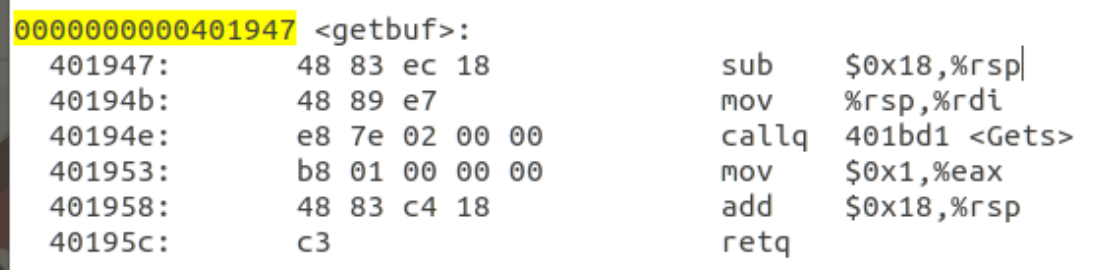
要覆盖返回地址，需要填充缓冲区和任何额外的空间，直到堆栈上的返回地址。额外的空间通常包括保存的基指针（8 字节）。计算到达返回地址所需的总输入长度：缓冲区大小 + 额外空间。小端序排





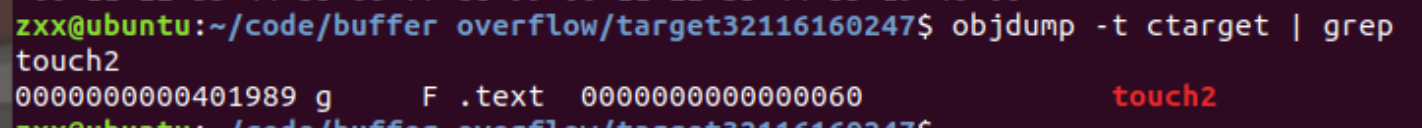
Level2：

1. 确认getbuf缓冲区大小：

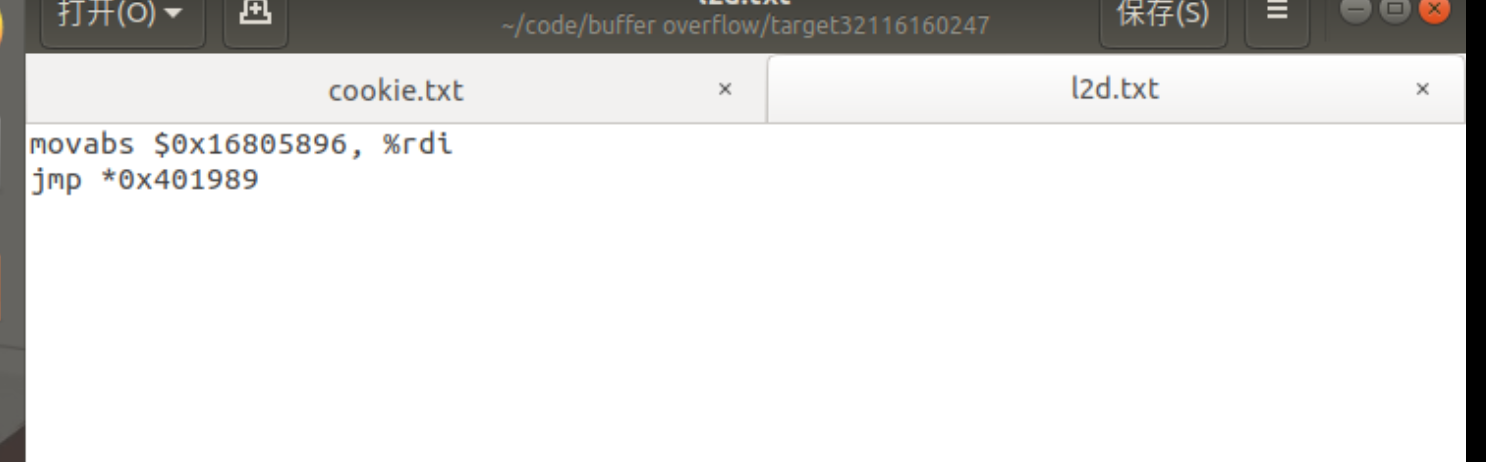


值 0x18以十六进制表示缓冲区的大小。将此值转换为十进制以获得缓冲区大小，24字节

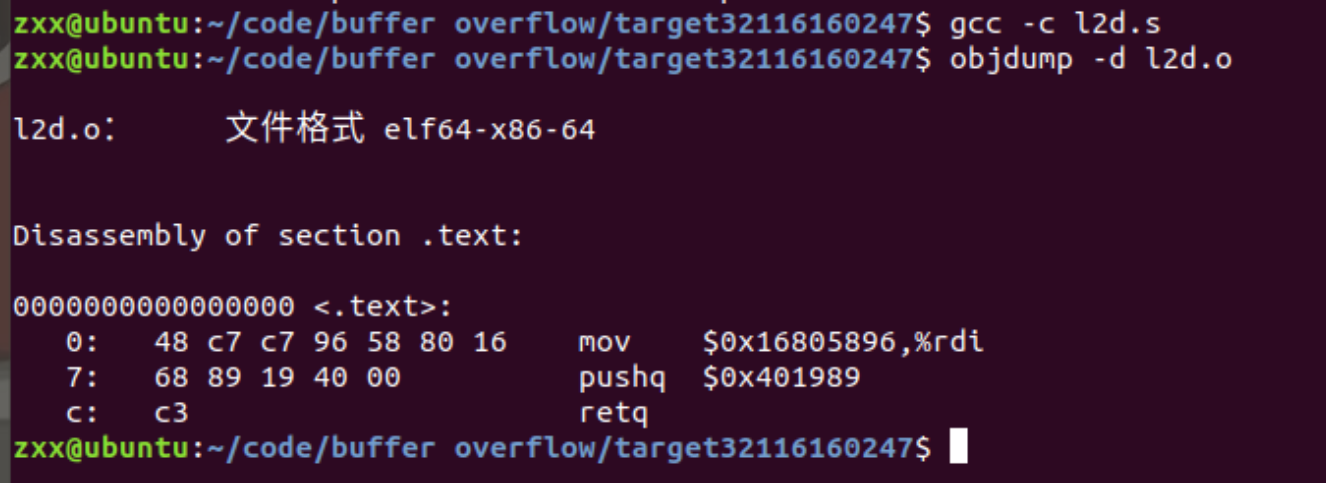
确认touch2地址：



2. 编写一个小的汇编代码片段：汇编代码应该将 cookie.txt 中的值加载到 %rdi 寄存器中，然后跳转到 touch2 函数。

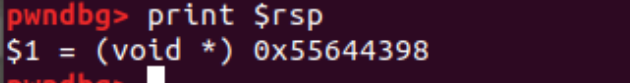


3.

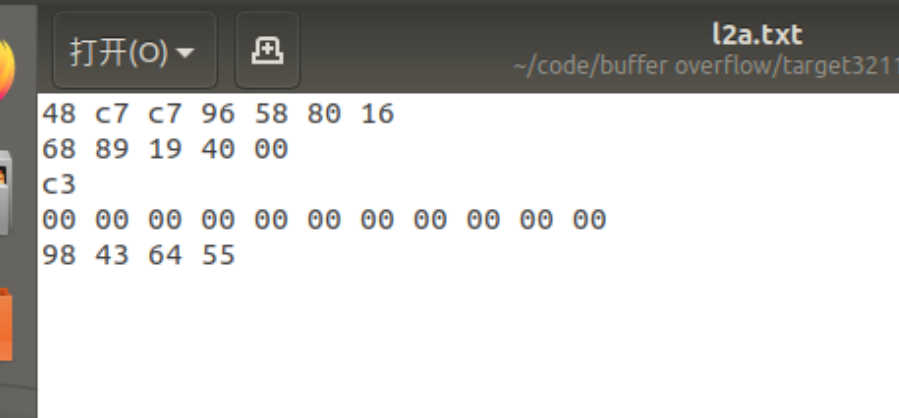


4.gdb找真正的缓冲大小：

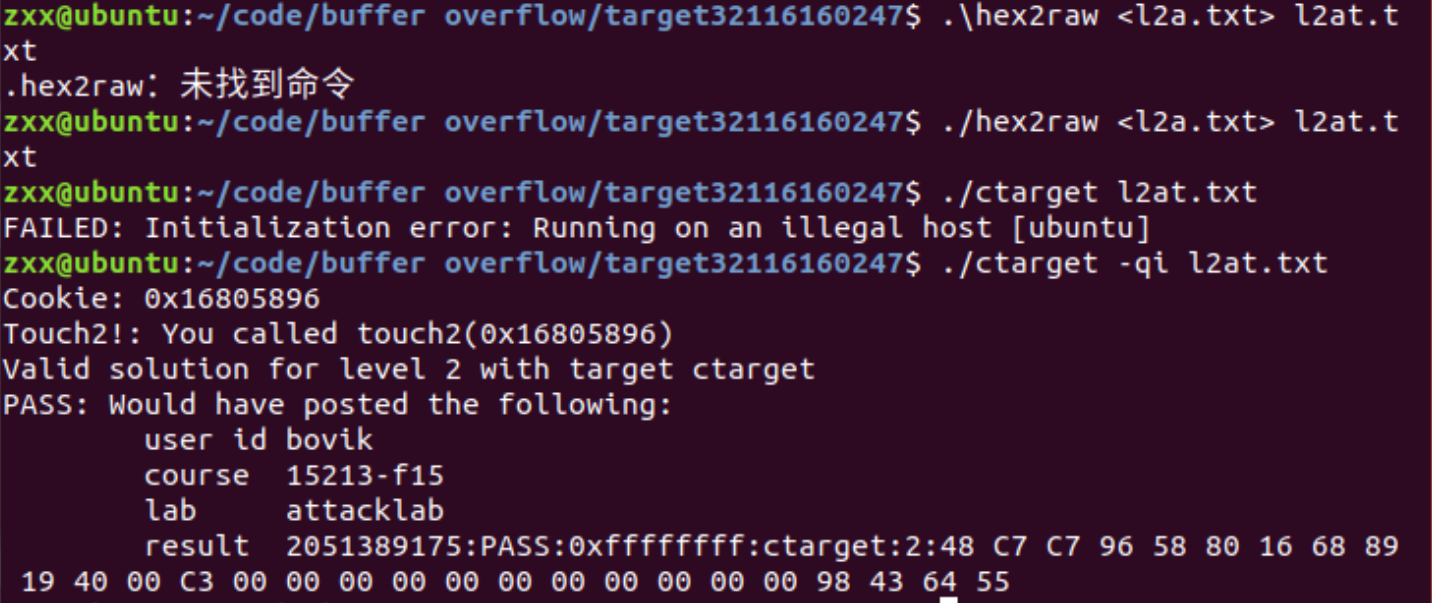




1. 写攻击文档

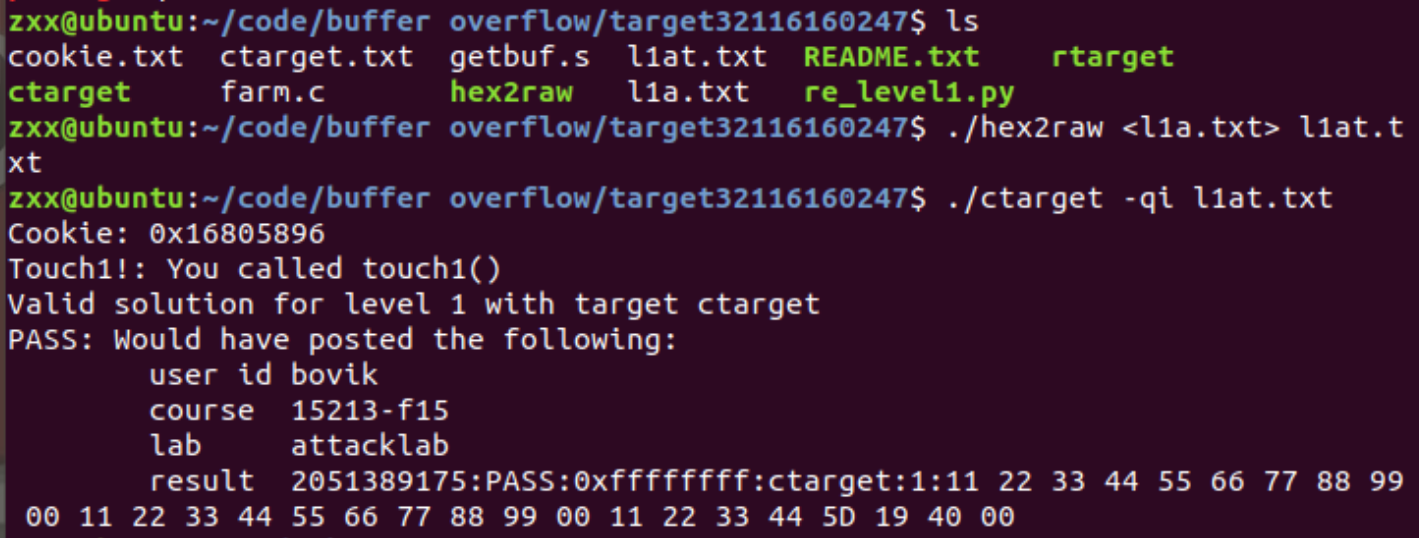


1. 转化后攻击

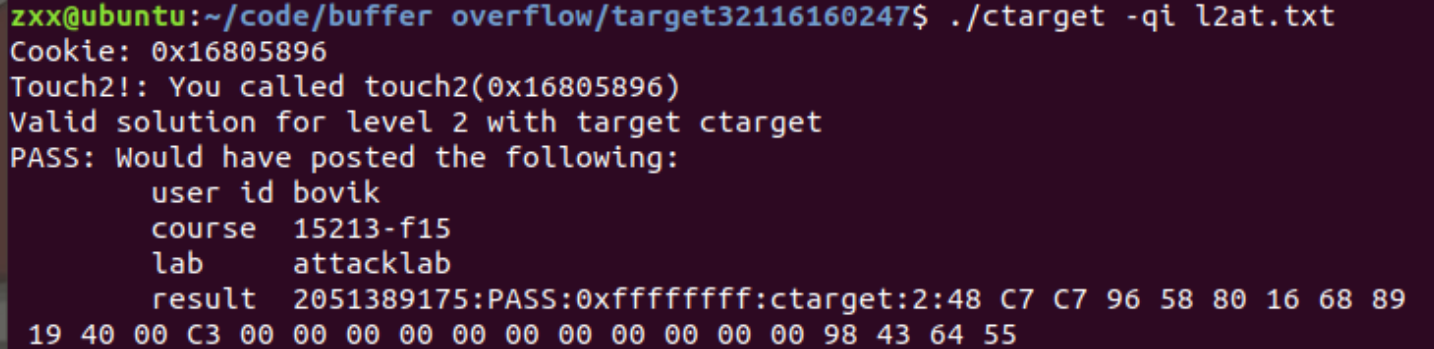


**五、程序运行结果截图**

Level1：



Level2：



**六、实验体会**

Level 1：

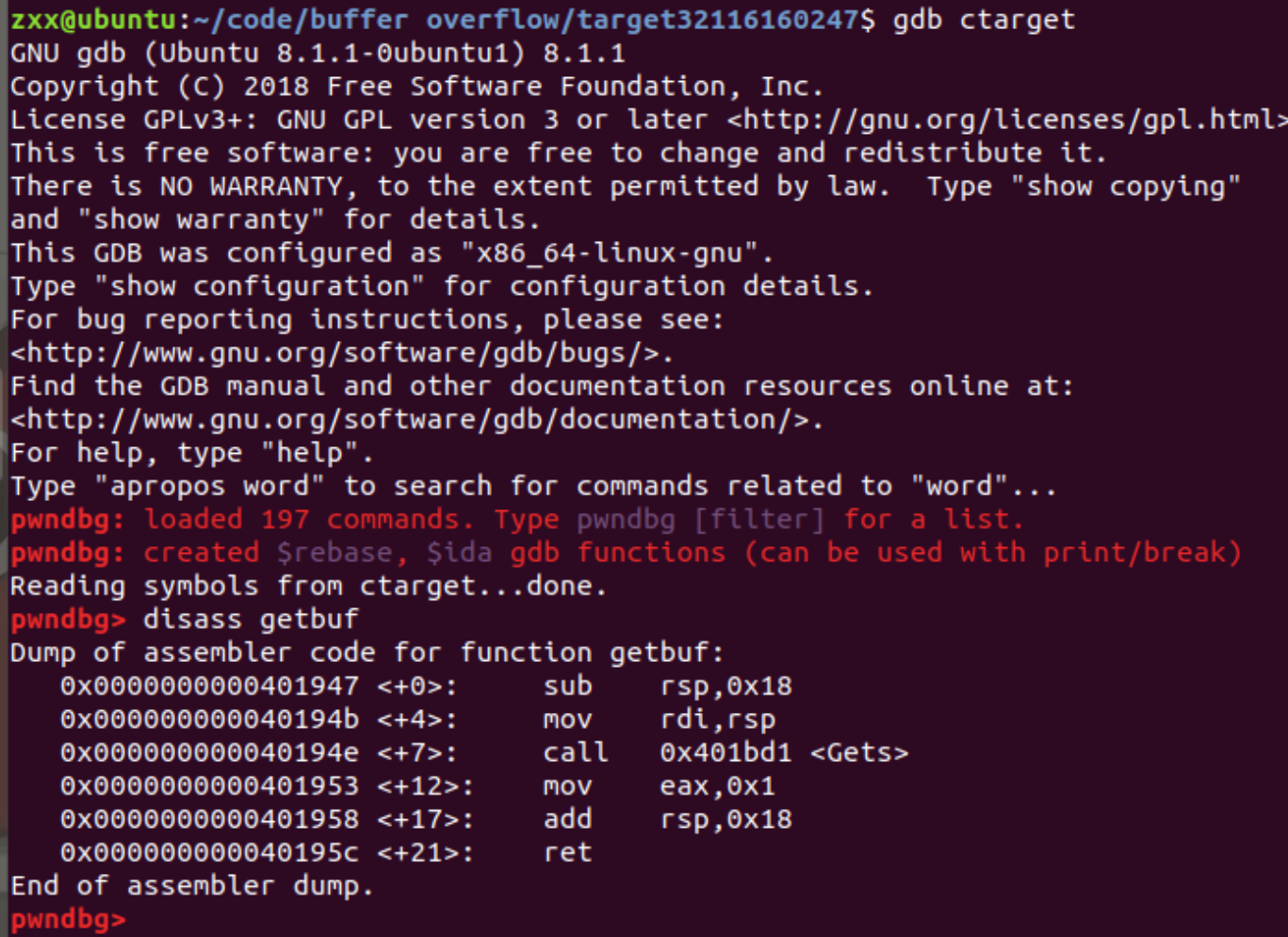
1. 我的想法：我用objdump反出了ctarget的汇编码并存进了ctarget.txt，正在从里面找touch1()的地址。我已经知道touch1()的C代码。我的想法是写出touch1()的汇编码,然后在txt中搜索，你觉得是否可行

ChatGPT回答：你的想法有一些问题。在反汇编代码中搜索 C 函数并不能保证一定会找到函数的地址。这是因为在编译器将 C 代码编译成汇编码时，编译器可能会将一个 C 函数分割成多个汇编码块，并将这些块分散在代码的不同位置。因此，在汇编代码中搜索函数名并不可靠。

改用：objdump -t ctarget | grep touch1

2.我的想法： unsigned getbuf(){ char buf[BUFFER SIZE];Gets (buf); return 1;}我将在ctarget.txt中找buf的开始和到堆栈中返回地址之间的字节数，我想检查getbuf()的汇编码，直接在ctarget.txt中搜索subq指令，可行吗？

ChatGPT: 在 ctarget.txt 中搜索 subq 指令是不可行的。因为在汇编代码中，subq 指令并不是唯一用于计算栈空间的指令，还有其它的指令用于栈空间的计算。因此，在汇编代码中搜索 subq 指令并不能保证能够准确地找到 getbuf() 函数内部计算栈空间大小的指令。

改进：找到了：

0x18=24字节