

中国科学技术大学

《网络安全》

实验报告

姓名：吴佳明

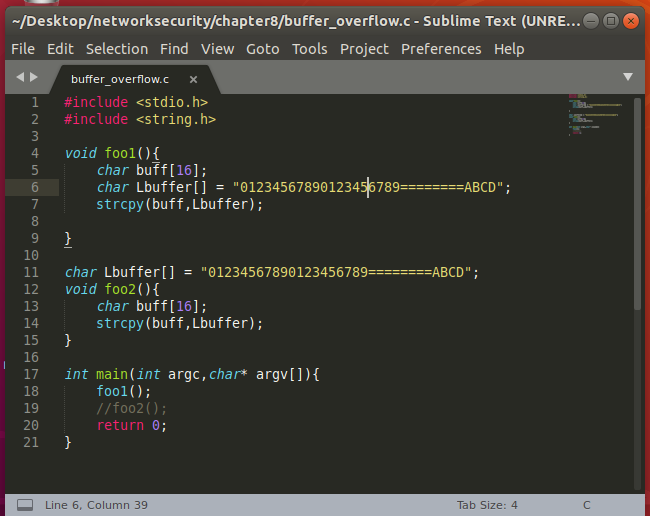
学号：SA19221027

年级：硕士研究生一年级

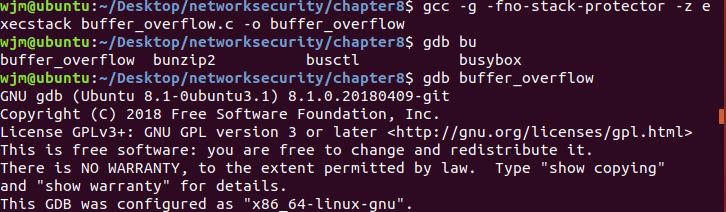
本实验基于ubuntu18 64bit

首先对foo1函数进行分析。

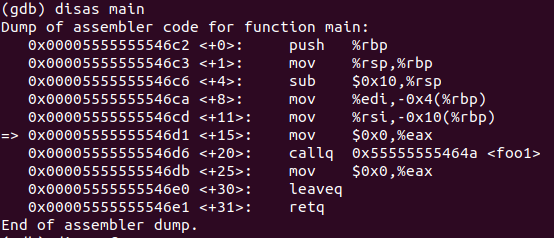
函数如下：

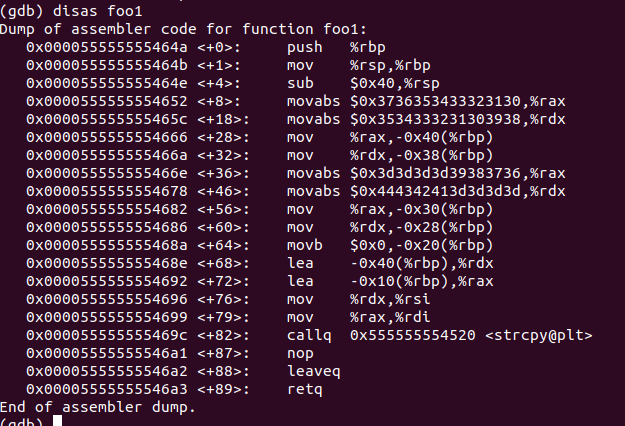


用gcc命令进行编译，使用gdb命令进行调试。

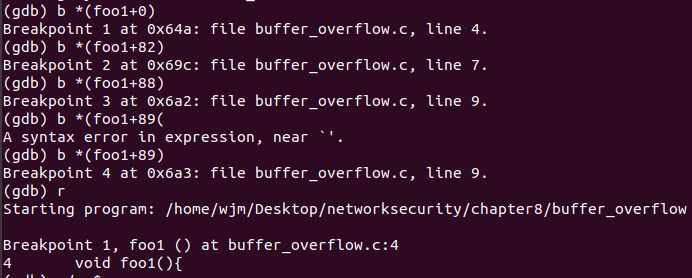


使用disas命令查看汇编代码（先运行一次）。





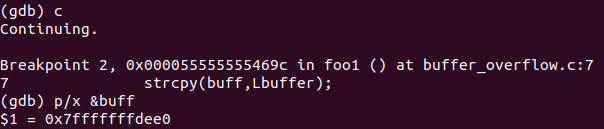
设置断点并开始运行。



在第一个断点处查看寄存器rsp中的值以及该值作为内存地址时内存单元的内容。



在第二个断点处查看buff的地址。

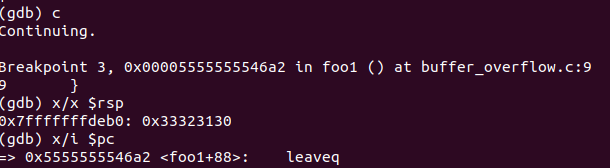


可以看到，buff的地址和之前查看的寄存器esp的值差0x18。

我们的Lbuffer字符串长度为32字节，值为01234567890123456789========ABCD，因此，在使用strcpy之后，从地址0x7fffffffdef8处开始，连续的8个内存地址的值为====ABCD。

下面我们看看是否正确。

在第三个断点处，查看接下来的汇编指令。

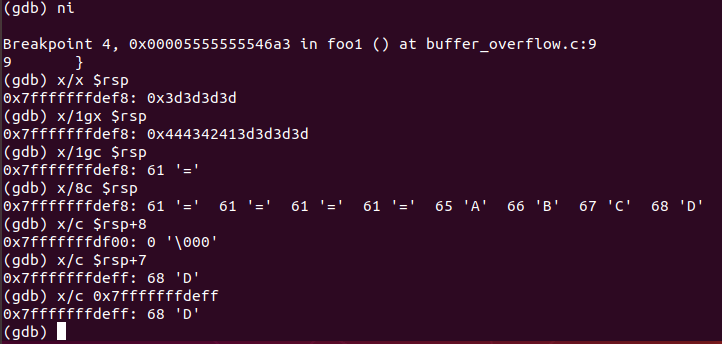


Leaveq汇编指令，相当于

mov %rbp, %rsp

pop %rbp

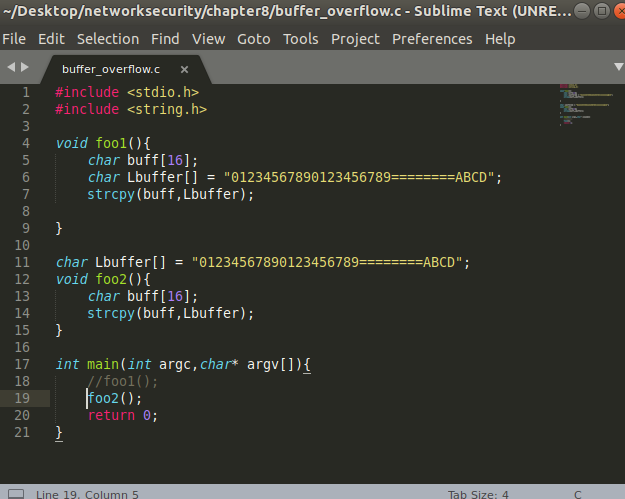
因此，执行完leaveq的指令后，寄存器rsp指向的值为foo1函数的返回地址。



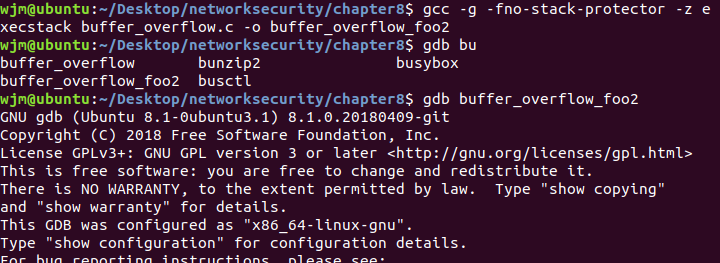
可以看到，现在地址0x7fffffffdef8到地址0x7fffffffdeff中的内容为====ABCD。

下面对foo2进行分析。

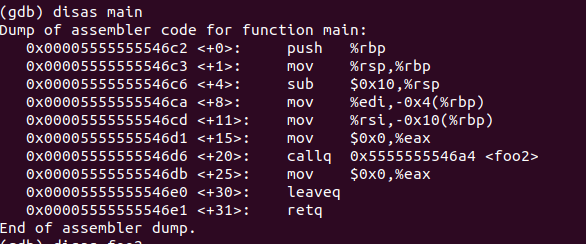
函数如下：

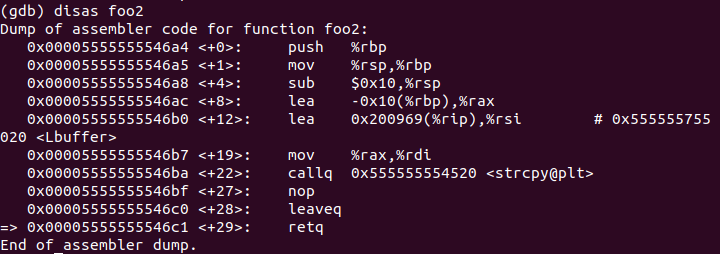


用gcc命令进行编译，使用gdb命令进行调试。

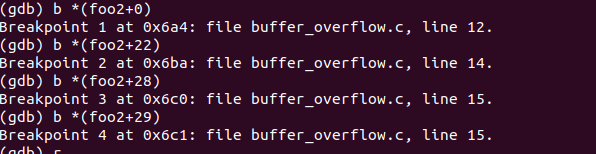


使用disas命令查看汇编代码（先运行一次）。





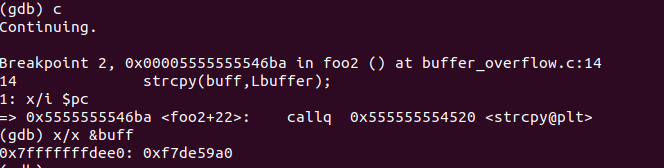
设置断点并开始运行。



在第一个断点处查看寄存器rsp中的值以及该值作为内存地址时内存单元的内容。



在第二个断点处查看buff的地址。

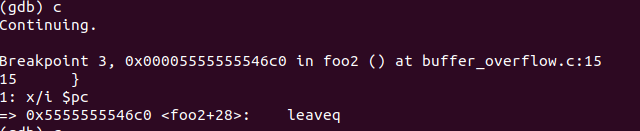


可以看到，buff的地址和之前查看的寄存器esp的值差0x18。

我们的Lbuffer字符串长度为32字节，值为01234567890123456789========ABCD，因此，在使用strcpy之后，从地址0x7fffffffdef8处开始，连续的8个内存地址的值为====ABCD。

下面我们看看是否正确。

在第三个断点处，查看接下来的汇编指令。

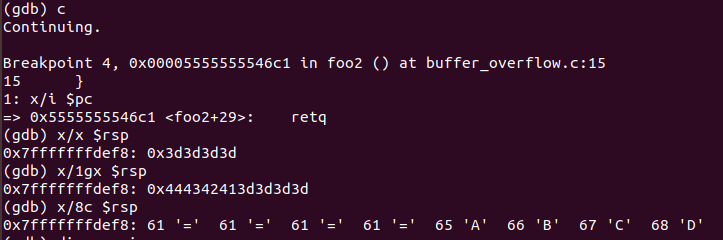


Leaveq汇编指令，相当于

mov %rbp, %rsp

pop %rbp

因此，执行完leaveq的指令后，寄存器rsp指向的值为foo2函数的返回地址。



可以看到，现在地址0x7fffffffdef8到地址0x7fffffffdeff中的内容为====ABCD。