

# 信息安全原理第二次作业报告

——缓冲区溢出攻击

## 1. 简单介绍

缓冲区溢出攻击是利用缓冲区溢出漏洞所进行的攻击行动。缓冲区溢出是一种非常普遍、非常危险的漏洞，在各种操作系统、应用软件中广泛存在。利用缓冲区溢出攻击，可以导致程序运行失败、系统关机、重新启动等后。

缓冲区溢出是指当计算机向缓冲区内填充数据位数时超过了缓冲区本身的容量，溢出的数据覆盖在合法数据上。理想的情况是：程序会检查数据长度，而且并不允许输入超过缓冲区长度的字符。但是绝大多数程序都会假设数据长度总是与所分配的储存空间相匹配，这就为缓冲区溢出埋下隐患。操作系统所使用的缓冲区，又被称为“堆栈”，在各个操作进程之间，指令会被临时储存在“堆栈”当中，“堆栈”也会出现缓冲区溢出。

## 2. 编程环境

Microsoft Visual Studio Professional 2017

版本 15.9.4

VisualStudio.15.Release/15.9.4+28307.222

Visual C++ 2017 00369-60000-00001-AA074

Microsoft Visual C++ 2017

### 3. 算法设计

由于代码较短，我直接粘贴进来，方便分析：  
这是一个认证程序，密码是“1234567”。

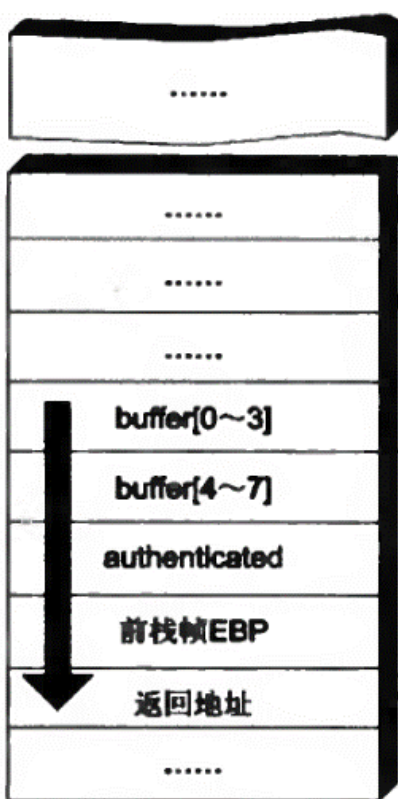
```
1.  /*verify.c*/
2.
3.  #include <stdio.h>
4.  #include <stdlib.h>
5.  #include <string.h>
6.
7.  #define PASSWORD "1234567"
8.  int verify(char *password)
9.  {
10.     int auth;
11.     char buffer[8];
12.     auth = strcmp(password, PASSWORD);
13.     strcpy(buffer, password);
14.     return auth;
15. }
16.
17. int main(void){
18.     int flag = 0;
19.     char pass[1024];
20.     while(1){
21.         printf("enter the password:\t");
22.         scanf("%s", pass);
23.         flag = verify(pass);
24.         if(flag)
25.             printf("password incorrect!\n");
26.         else{
27.             printf("congratulation!\n");
28.             break;
29.         }
30.     }
31.     system("pause");
32. }
```

其中我们看这部分函数：

```
1. int verify(char *password)
2. {
3.     int auth;
4.     char buffer[8];
5.     auth = strcmp(password, PASSWORD);
6.     strcpy(buffer, password);
7.     return auth;
8. }
```

正常情况，并不需要 `strcpy(buffer, password)` 这句话，但是当程序员并没有遵循安全编程的准则时，黑客就有可乘之机，利用该句就可以实现“密码破解”。

说实现之前，首先我们需要了解一个程序中的某个函数调用的堆栈的使用情况。如图所示，该部分我们可以看到当我们输入了大于 8 位的字符串并将其拷贝到 `buffer` 上时，就会覆盖 `authenticated` 这个判断是否输入正确的整型变量。



比如我们输入“aaaaaaaaaaaa”

(12 个'a')时惊奇发现通过认证了。

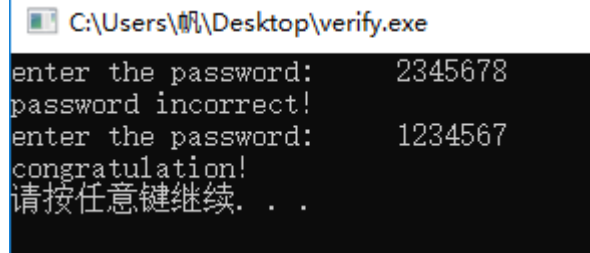
就是因为其覆盖了 `authenticated`，也可以尝试其他多个字符。我们发现，如果我们输入更多岂不是能更改返回地址了？比如返回到了重启计算机，强制系统关机，获取超级用户权限的地址等等，我们就能对计算机进行攻击了。这个需要一定的汇编、硬件知识。

这就是利用缓冲区溢出漏洞进行攻击的一个实例，如果我们利用精准精确一些的话，可以做到更多事情。

本次算法实现仅仅实现了若程序员写了 `gets()`、`scanf()`、`strcpy()`，等不安全的函数时，可以利用其缓冲区的溢出对进行计算机攻击。

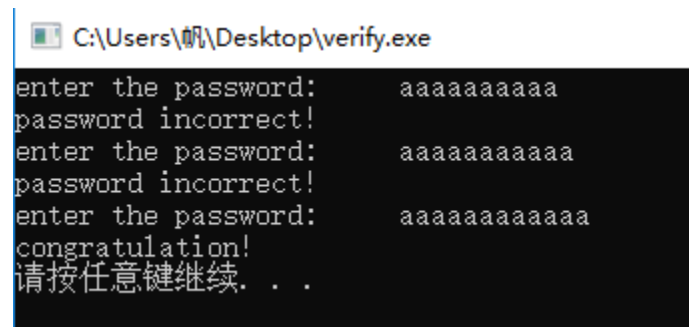
## 4. 实验结果

a) 正常输入密码时：



```
C:\Users\帆\Desktop\verify.exe
enter the password: 2345678
password incorrect!
enter the password: 1234567
congratulation!
请按任意键继续. . .
```

b) 利用缓冲区溢出“破解”密码时：



```
C:\Users\帆\Desktop\verify.exe
enter the password: aaaaaaaaaa
password incorrect!
enter the password: aaaaaaaaaa
password incorrect!
enter the password: aaaaaaaaaa
congratulation!
请按任意键继续. . .
```

## 5. 如何预防？

在程序员正式上岗以及产品正式发布前，必须进行程序员的安全编程规范的培训，并且要检查是否使用了不安全的函数，如在 VS 中就会有 SDL 安全检查，视 scanf() 等函数为 error 以防止使用。

## 6. 总结与经验

本次作业让我初步理解了程序调用函数的过程以及对于堆栈的使用，初步对计算机硬件、汇编有一定了解，并且对缓冲区溢出攻击的原理有了更加深入地了解！对于这门课、信息的安全，计算机的攻击与防守产生了浓厚的兴趣，以前从未涉猎该部分的我对于这部分有着强烈的兴趣。

感谢助教哥哥的批阅！

THANKS!