CLASS 5

Buffer Overflow Lab

吴瑞欣-E41614059

1、阅读 Basic Integer Overflows 这篇文章，大概描述整数溢出的原因和危害。

http://www.phrack.org/issues.html?issue=60&id=10#article

原因：将大数放到了存储空间小的内存中。

种类：1：符号位引起溢出

2：运算超出范围引起的溢出

危害：整数溢出的危害在文档中并没有直接点明，但是，通过自己理解我觉得整数溢出的危害和数据溢出的危害是完全相同的。可以参考数据溢出的危害，得出结论。

数据溢出会造成：crash（崩溃），divert（恶意代码执行），inject（注入）

翻译：

https://blog.csdn.net/u012133341/article/details/79957377

```
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <setjmp.h>

void catcher(int a)
{
        setresuid(geteuid(),geteuid(),geteuid());
        printf("WIN!\n");
        system("/bin/sh");
```

```
        exit(0);
}

int main(int argc, char **argv)
{
        if (argc != 3 || !atoi(argv[2]))
                return 1;
        signal(SIGFPE, catcher);
        return atoi(argv[1]) / atoi(argv[2]);
}
```

可以通过构造 argv[2]，使得 argc[2]参数整数溢出，覆盖掉参数 argv[1]与 argc，这样 argv[2]不仅使得 SIGFPE 信号量为 1，而且 if 不会执行

2、阅读堆溢出的文章 Once upon a free()...

http://www.phrack.org/issues.html?issue=57&id=9#article。

比较堆溢出和栈溢出的异同。

想要比较堆溢出和栈溢出的异同，首先要明确堆与栈的区别与联系。

堆是在程序运行时，而不是在程序编译时，申请某个大小的内存空间。即动态分配内存，对其访问和对一般内存的访问没有区别。

栈后放进去的先拿出来，它下面本来有的东西要等它出来之后才能出来。（后进先出）

总结：堆是指程序运行时申请的动态内存，而栈只是指一种使用堆的方法(即先进后出)。栈是堆中的特例，非先进后出的数据溢出称为堆溢出。

3、(一定要做）上网搜索利用 jmp esp 或者 call esp 来进行栈溢出的文章并且仔细阅读，解释这种技术的优点。

shellcode 在内存中的起始地址往往不固定，导致漏洞利用不一定成功，可以通过 jmp esp 的方式来解决这个问题。

call esp 可以通过跳转，进行漏洞利用。

我对他们的理解还不够深刻，望通过学习得以加强。