实验报告

实验名称	实验六：C/C++静态源码分析

1. 攻击面分析：
   1) AssignmentAddressToInteger：将地址值赋给 integer (int/long/etc)类型。
   2) arrayIndexOutOfBounds：数组访问越界。
   3) autoVariables：自动变量（也就是局部变量），存储空间在栈 stack 中，函数结束时空间被释放，如果此时变量地址被外部空间的函数使用，将会引起错误。
   4) bufferAccessOutOfBounds：数组访问越界。
   5) erase：erase 方法后迭代器的使用。
   6) memleak：内存泄漏。
   7) nullPointer：空指针引用。
   8) outOfBounds：缓冲区访问越界。
   9) resourceLeak：文件未关闭导致资源泄漏。
   10) syntaxError：定义宏时错误。
2. 设计思路：
   1) AssignmentAddressToInteger：直接返回 p+4，删除 int a = p;
   2) arrayIndexOutOfBounds：a[2]访问越界，删除语句 a[2] = 0;
   3) autoVariables：使用语句**a = b;获取局部变量 b 的值；
   4) bufferAccessOutOfBounds：数组大小为 2，for 循环中修改为 i<2；
   5) erase：保存返回的指针到迭代器 iter= items.erase(iter);
   6) memleak：程序结束前释放内存 free(a);
   7) nullPointer：仅当 p 不为空时可以引用。
      - 第一处修改为：

```
if (p) {//
        int x = p->x;
        do_something(x);
    }
```

      - 第二处修改为：

```
if (str[i] == ' ')
{
    p = str + i;
    return p[1];
    //break;
  }
```

      - 第三处修改为：

```
if (a == 1) {
        p = fred1;
        p->x = 0;
}
```

   8) outOfBounds：复制操作中源字符串修改为"0123"，符合大小范围。

9) resourceLeak：文件打开时程序结束应关闭 fclose(a);
10) syntaxError：修改为

```
#ifdef A
    int main()
{
}

#endif
```

3. 心得体会

本次实验，我了解到了静态分析在不执行程序的前提下就可以查出错误，此外也学会了 cppcheck 静态源码分析工具的使用方法；

了解了静态程序分析工具检查的类型：变量类型不匹配、变量在使用前未定义、不可达代码、死循环、数组越界、内存泄漏、空指针提取等。