

Secure Coding

MEMORY





```
class SomeClass
{
public:
    static int count;
    SomeClass() {}
    static SomeClass* GetNewInstance()
    {
        count++;
        return new SomeClass();
    }
};
```

静态变量count用于记录此类型生成的实例的个数，如果因new分配内存失败而抛出异常，那么其实例个数并没有增加，但count变量的值却已经多了一个，从而数据结构被破坏。

```
class SomeClass
{
public:
    static int count;
    SomeClass() {}
    static SomeClass* GetNewInstance()
    {
        count++;
        return new SomeClass();
    }
};
```

静态变量count用于记录此类型生成的实例的个数，如果因new分配内存失败而抛出异常，那么其实例个数并没有增加，但count变量的值却已经多了一个，从而数据结构被破坏。

```
class SomeClass
{
public:
    static int count;
    SomeClass() {}
    static SomeClass* GetNewInstance()
    {
        SomeClass* p = new SomeClass();
        count++;
        return p;
    }
};
```

如果new失败则直接抛出异常，count的值不会增加。