Software Security – Taint Analysis

Zhilong Wang

version：*1.0*

Last update：*May 14, 2019*

摘　　要

本文档是南京大学软件安全课程实验的说明文档。此文档介绍如何利用 Intel PIN 进行基本的染色分析。

1　实验目标

通过该实验掌握动态插桩工具 PIN 的简单使用，理解染色分析的的原理，熟悉染色分析的基本过程，并且能够清楚的认识到染色分析在软件安全领域的理论意义与实用价值。

2　实验环境

2.1　Intel PIN 的安装

我们我实验可以在任意 Intel PIN 支持的 Ubuntu 版本系统。首先你需要通过一下链接下载 Intel PIN：https://software.intel.com/en-us/articles/pin-a-binary-instrumentation-tool-downloads。Intel PIN 是免安装软件，下载并解压缩即可使用。

2.2　Intel PIN 的使用

Intel PIN 入门教程请参考：https://software.intel.com/sites/landingpage/pintool/docs/97619/Pin/html/。编写 PIN 所支持的相关函数使用方法请参考：https://software.intel.com/sites/landingpage/pintool/docs/71313/Pin/html/group__PIN__SYSCALL__API.html

Intel PIN 给提供了大量的样例代码，通过阅读学习这些样例代码可快速的掌握 PIN 的使用，PIN 的样例代码在文件夹：”source/tools/” 中，我们通过样例代码讲解 PIN 的使用。

编译一个文件夹下的所有代码：

```
$ cd source/tools/ManualExamples
$ make all TARGET=ia32
```

编译一份代码：

```
$ cd source/tools/ManualExamples
$ make obj-intel64/inscount0.so TARGET=ia32
```

运行 PIN tools:

```
$ ../../../pin -t obj-intel64/inscount0.so -- /bin/ls
```

该条指令在 PIN 提供的插桩环境下运行目标程序/bin/ls，通过插件 inscount0.so 实现指令计数。

总的来说，PIN 为我们提供了一个插桩和运行环境，任何基于 Intel 指令集的二进制文件都可以在 PIN 上运行，使用者通过编写 PIN Tools 调用 PIN 提供的 API 实现自己的功能。

3 染色分析实现缓冲区溢出检测

3.1 漏洞代码

Listing 1: 漏洞程序

```
struct MyType{
    char input[10];
    int offset;
    int BUFF[100];
};
struct MyType Data;
int vulfun1(){
    Data.offset = 10;
    readstr(Data.input);
    *(Data.BUFF + Data.offset) = Data.input[0]+Data.input[1]+Data.input[3]+Data.input[4];
    return 0;
}
int vulfun2(){
    char buff[10];
    readstr(buff);
    return 0;
}
int main(){
    vulfun1();
    vulfun2();
    return 0;
}
```

代码1中的函数 `vulfun1()` 和 `vulfun2()` 各存在一个缓冲区溢出漏洞:

- 通过函数 `vulfun1()` 中的缓冲区溢出漏洞可以篡改指针的偏移量，进而影响指针的解引用，实现对程序数据流的劫持。
- 通过函数 `vulfun2()` 中的缓冲区溢出漏洞可以溢出返回地址，实现对程序控制流的劫持。

3.2 触发漏洞

程序的运行接受两次输入，用于触发程序中的两个漏洞，分别在输入：“012345678912a”和“0123456789012345 67”即可触发两个漏洞。

3.3 实验要求

在 Intel PIN 平台下编写 PIN Tools 实现染色分析，跟踪上述漏洞程序的输入流向，以检测两次缓冲区溢出。

实验需要解决下列问题:

- 如何利用 PIN 提供的 API 函数截获程序的输入?
- 如何利用 PIN 实现对数据流传播的跟踪?
- 如何在指针解引用时确定指针是否被输入污染?
- 如何在程序返回时确定返回地址是否被篡改?