**软件漏洞分析技术读书报告**

姓名：仝志欣

|  |
| --- |
| **本周预期完成任务：**阅读软件漏洞分析技术的第五章内容 |
| **本周实际完成进度：**了解污点分析相关的概念以及基于污点分析的漏洞挖掘基本原理；了解两种污点分析的实现方法，分别是基于数据流的污点分析和基于依赖关系的污点分析；了解两款典型的利用污点分析技术的漏洞分析工具。 |
| **详细内容：**   1. **定理证明基本原理**   **1. 基本概念**  污点分析是一种跟踪并分析污点信息在程序中流动的技术。分析对象是污点信息流。这个污点信息可以是程序之外、进入程序的信息，也可以是程序接收的外部输入数据或者捕捉到的来自外界的信号如鼠标点击等。  过程：识别产生点（位置）并对污点信息进行标记，然后利用特定规则跟踪分析污点信息在程序中的传播过程，然后检查这些污点信息是否会影响一些关键操作。Source点规则和Sink点规则以及污点信息传播规则统称为污点分析规则。  污点分析本质上是一个跟踪输入数据流向的过程，在这种情况下污点分析过程与数据流分析中对数据流向分析很是相似。但分析过程中也会有基于依赖关系的分析，包含数据依赖关系和控制依赖关系，基于上述依赖关系传播的两种信息流分别被称为显示信息流和隐式信息流。  污点分析应用于漏洞挖掘中就是将程序漏洞问题转化为污点信息是否会被Sink点上的操作所使用的问题。污点类型的漏洞比如SQL注入漏洞、跨站脚本、命令注入漏洞等。  **2. 使用污点分析技术挖掘程序漏洞**  可以有两种方法实现：一种是利用中间表示、控制流图以及调用图分析污点信息在程序路径上流动的方法，称为基于数据流的污点分析方法；另一种是使用程序依赖图或者利用程序中的数据依赖关系或者控制依赖关系，分析敏感操作是否依赖于污点信息。  ①程序：软件程序根据指定规则由转换为一种逻辑形式，过程类似于程序编译，差别在于转换后的不是机器指令而是其他定理证明工具可接受的规范形式。例如Saturn采用SIL对程序建模，构造整数、指针、记录和条件分支语句的规范形式，最终转换成布尔公式。  ②属性转换：按照与程序转换中一致的转换规则在抽象转换器中将程序安全属性转换为约束条件。程序的安全属性通常以断言或注释形式给出，一些漏洞检测工具比如Saturn等即通过断言保证程序的安全属性，ESC/Java将通过添加注释验证其是否满足相应的安全属性。最后把断言和注释形式转换为布尔表达式。  ③定理证明：经过定理证明器处理后自动得出证明结果。然后，对证明结果进行分析，若程序满足指定的安全属性，则它是安全的；否则，则说明程序存在相应的漏洞。基于定理证明的漏洞分析工作流程中首先形式化地描述程序和程序安全属性，将其表达为定理证明器能够识别的形式。其中程序被转换为规范形式，安全属性被抽象为约束条件。在这一过程中，其核心问题即为程序和程序安全属性对应的定理如何生成，以及定理如何证明两部分内容。   1. **方法实现**   检测系统首先对程序代码进行解析获得中间代码表示，之后对代码进行控制流分析等辅助分析，获得那几种有用的结构。在辅助分析过程中，系统可以利用污点分析规则在中间表示上识别Source和Sink点，为接下来的分析做准备。   1. **基于数据流的污点分析**   需要的辅助分析：别名分析、取值分析。取值分析：静态的分析一些变量的取值，比如分析数组下标的取值有利于区分数组中元素的污染情况。通常让辅助分析和污点分析交替进行，好处：有针对性地对程序代码执行辅助分析，而不去考虑和污点信息可能不相关的代码，在一定程度上减少分析过程中的开销。  (1)过程内的分析  记录污点信息：通常使用污染标签进行记录，简单的污染标签如布尔型的变量，复杂的标签可以记录污染信息来自哪些Source点，甚至Source点接收的哪个部分。也可以不使用标签，而是使用对变量进行跟踪的方式来达到分析污点信息流向的目的。比如栈或者队列辅助完成。  ②程序语句的分析   1. 赋值语句。简单赋值、带有二元操作的赋值（其中需要注意的是当右端两个操作数都被污染后，复杂标签的表示）、和数组相关的赋值（下标的取值运算；很多情况下数组下标的取值不容易计算，分析过程省去对数组下标计算的过程）、包含字段的赋值和包含指针操作的赋值语句常常利用指向分析的结果。 2. 控制转移语句。对于条件控制语句，要考虑条件控制转移语句中的路径条件可能是包含对污点数据的限制。比如数组下标变量为污点信息时，当得出路径的限制是足够的时候就可以将相应的变量标记为污染的，而剩余的标记为为污染的。 3. 过程调用语句。   3 代码遍历：与第四章介绍的方法一样，只不过多了关注污点信息。   1. 流敏感分析 2. 路径敏感分析。为追求分析精确，还需要将分析路径条件对污染数据的限制进行记录，如果在一条程序路径上，这些限制足够保证数据不会被攻击者所利用，就可以将相应的变量状态记为为污染。   (2)过程间的分析  在分析中为每一个过程建立一个污点分析的摘要，并利用已有的摘要进行过程内的分析。此外还可以使用数据流过程间分析中的自顶向下的分析方法，利用队列记录待分析的过程，每次取一个过程，对该过程进行过程内的分析。   1. **基于依赖关系的污点分析**   分析程序依赖关系的过程可以看作是构建程序依赖图的过程。对于一定规模的程序常常无法完整计算出程序的依赖关系原因有二：空间上、静态分析的不可判定性。因此优先考虑和污点信息相关的程序代码。并且在实际应用中通常只关心程序上的数据依赖关系，因为隐式污点信息流很难被利用。  构建程序依赖关系，可能用到指向分析结果来分析程序中的数据依赖关系。  此外也可以利用程序切片技术将语句提取出来，例如p126中图5.7 。   1. **典型工具**   **1. Pixy**  **2. TAJ** |
| **相关文件及参考资料：** 软件漏洞分析技术的第五章内容 |
| **遇到的问题及该阶段学习的想法：** |
| **接下来的进度计划：** 阅读第七、八章 |