**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八章内容 |
| **本周实际完成进度：**了解定理证明的原理、方法实现和典型工具。 |
| **详细内容：**   1. **定理证明基本原理**   定理证明与前一章的模型检测同属于基于逻辑的检测方式，原理也极为相似。定理证明的本质在于在形式化规约的基础上，分析系统是否具有所期望的性质。主要流程：首先对于需要安全性证明的程序进行分析，以数学名义给出其形式化定义然后以定理公式的形式描述程序的安全属性，并转接为可求解的规范表示；最后通过定理证明器对程序和程序安全属性对应的定理进行证明，以判定是否满足安全属性。  **1. 基本概念**  它采用逻辑形式来描述程序及其安全性质，其中的逻辑由一个具有公理和推理规则的形式化系统给出，然后使用公理和规则来证明程序是否是一个合法的定理，以此来判断待分析程序是否满足制定的安全属性。如果程序合法，那么它便满足我们所要求的安全属性，**在实际中，证明程序是否合法的过程一般是由定理证明器自动实现的。**  **该技术中用到的概念和定义：自动定理证明、形式化规约、形式化验证、命题逻辑、Hoare逻辑、BAN逻辑、布尔公式、布尔可满足性。（P185）**  **2. 技术框架**  在基于定理证明的漏洞分析过程中，程序和程序的安全属性均以逻辑形式描述，定理的证明由传统的手工验证发展到目前定理证明器实现定理证明过程。  ①程序转换：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。  从以上的分析中我们可以发现其与模型检测的区别，定理证明的安全属性是基于源程序需要的安全为基础进行形式化的，而模型检测中的程序安全缺陷属性是基于常见的漏洞缺陷属性进行集成的，所以构建的状态转换系统是基于程序安全缺陷属性的。   1. **定理证明的方法实现** 2. **程序转换**   在程序形式化转换过程中，首先需要对代码中的各种数据类型、指针、结构体和循环以及分支结构进行转换，将其转换为一种定理证明系统可接受的规范形式。转换方式很多，主要包括：传统的证明程序正确性方法，如Floyd归纳断言法，将程序转换成流程图，并建立断言进行推导；PVS定理证明器根据生成的程序验证条件；基于SAT的定理证明工具把程序转换成布尔公式然后对其可满足性求解。  ①流程图  流程图用于描述软件的控制信息流，由图框和流程线组成。其中图框表示各种操作的类型，图框中的文字和符号表示操作的内容，流程线表示操作的先后次序。传统的定理证明方法一般把程序转换成流程图的方式予以处理，例如Floyd归纳断言法和良序集法，Manna子目标断言法等。  ②验证条件  验证条件源于Floyd和Hoare程序验证思想，通过逻辑断言形式对程序规约，是程序转换后的一种逻辑形式。在该逻辑形式中，前置假设条件指明程序如何被调用，后置断言条件表明程序结果状态应该满足的关系，循环不变量制定程序执行过程中其属性应该满足的条件，以此来验证程序的安全性和可靠性。**验证条件的过程是为定理证明属性描述的基础。**  ③布尔公式  布尔公式是程序转换的一种形式。在基于SAT定理证明方法中，程序首先需要对代码中的各种数据类型、指针、结构体和循环以及分支结构进行转换，成为布尔公式形式，然后将其代入SAT求解器中予以证明。基于SAT的程序源代码—布尔公式的转换方法包括以下五个步骤：对程序源代码进行预处理、循环展开、函数调用展开、数组处理、从程序代码得到布尔公式。这时得到的布尔公式与定理证明属性描述得到的定理命题进行比对检查即可。   1. **定理证明属性描述**   定理证明以数学方法来分析程序安全属性，并以形式化方法对其进行精确描述和表达，其中不仅包括程序对象的安全属性，还包括**逻辑证明部分**。在属性描述过程中，**根据指定的规则，把程序安全属性转换为定理命题的表达形式，即转换为定理证明器可以识别的规范逻辑公式子句构成的定理命题。**任何一个定理命题均可转换为一个子句集，通过各种可满足性方法判定该子句集是否可满足，从而判定该命题是否成立。关于属性的描述，有多种形式。  ①断言  对程序属性的刻画，可以采用断言形式。断言可表示为布尔表达式，然后在程序中的某个特定点测定该表达式值是否为真。断言有三种形式：前置条件断言，表示代码执行之前必须满足的条件；后置条件断言，代码执行之后必须满足的条件；循环不变式断言，即反映循环变量的变化规律，且在每次循环体的执行前后均为真的逻辑表达式。在证明程序漏洞中，断言主要用于刻画程序的安全属性。  ②注释  在编写程序代码的过程中，一般都会用到注释，注释就是对代码的解释和说明。程序注释一般包括序言性注释和功能性注释。序言性注释主要包括模块的接口、数据的描述和模块的功能。功能性注释主要包括程序段的功能、语句的功能以及数据的状态。   1. **定理证明**   程序的正确性证明包括部分正确性、程序终止性和完全正确性三个方面，一个程序的完全正确，等价于该程序是部分正确的，同时又是可终止的。证明部分正确的方法有：Floyd的不变式断言法、Manna子目标断言法和Hoare的公理化方法等。终止性证明的方法有：Floyd良序集方法、Knuth的计数器方法和Manna等人的不动点方法。完全正确性证明的方法有:Hoare公理化方法的推广、Burstall断言法和Dijkstra弱谓词变换方法以及强验证法。目前，定理的证明一般由定理证明器完成。  Floyd不变式断言法：Floyd不变式断言法证明过程为首先建立程序的输入、输出断言，如果程序中有循环出现的话，在循环中选取一个断点，在断点处建立一个循环不变式断言。然后将程序分解为不同的道路，为每一个通路建立一个检验条件，最终用数学工具证明所有的检验条件，如果每一条检验条件都为真，则程序是部分正确的。  这几种定理证明器大同小异，用的算法和架构不同，产生了面向不同方面的功能和结构。   1. **实例分析**   **我应该做个关于看过实例之后与看实例之前的分析。看过实例分析后，定理证明是将程序转换而来的断言与属性描述而来的断言与注释有机结合，也就是插入到合适的地方，然后用定理分析器进行运行。**   1. **典型工具**   **1. Saturn**  Saturn是基于SAT问题求解框架的静态程序漏洞检测工具，主要面向大型C语言软件程序，可分析上百万行代码。它自动验证多种安全属性，能够并行分析多个函数，具有可扩展性，能准确检测程序漏洞。Saturn的实现主要基于函数摘要，即每一个函数都被分别分析，得到其行为的摘要。分析程序中函数调用时只对其对应的摘要分析，函数摘要中包括数据类型、全局变量以及其他相关信息。同时也基于条件约束，即利用约束条件描述程序状态的迁移过程，并把迁移过程表示为一个整体的系统。Saturn中约束条件以布尔可满足性描述语言表示，其布尔表达式中每一位分别对应一个过程或布尔变量。最终Saturn中程序分析部分以逻辑程序语句Calypso表达，能够对函数摘要和条件约束提供有效的支持。  **2. ESC/Java**  ESC/Java是由康柏计算机公司系统研究中心在2002年研发的开源工具，用来检测Java程序漏洞。ESC/Java中包含精确分析程序语义引擎和定理证明器Simplify,可以对Java程序进行自动的定理证明和精确的语义分析，能够静态地自动检测出Java程序源代码中的程序缺陷。ESC/Java的实现原理基于模块化检查思想，由用户对每一个例程进行给出相应规格条件的注释，在程序执行过程中ESC/Java检测各例程的执行是否满足其规格条件。 |
| **相关文件及参考资料：** 软件漏洞分析技术的第八章内容 |
| **遇到的问题及该阶段学习的想法：**第八章介绍了定理证明这种形式化的检测方法。定理证明是众多静态分析方法中最准确的，可以处理无限状态空间问题。但由于其证明过程过于抽象，致使程序运行过程的相关信息缺乏，分析程序漏洞的产生过程的能力也较差。 |
| **接下来的进度计划：** 阅读第九、十章 |