**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七章内容 |
| **本周实际完成进度：**了解模型检测的相关原理及方法实现过程；分析具体实例；两个典型工具 |
| **详细内容：**   1. **基本原理**   **1. 基本概念**  基本思想：用状态迁移系统表示软件系统的行为,用模态逻辑公式描述系统的性质。将“系统是否具有所期望的性质”就转化为数学问题“状态迁移系统是否是公式的一个模型”。  相关的定义  定义7.1:时序逻辑又称时态逻辑,由模态逻辑演变而来。 它是一种广义模态逻辑,主要用于对并发程序的推理,包括在考虑描述程序中单独语句行为的一组公理的情况下来证明程序的属性。时序逻辑由**命题变量**,**逻辑算子**,包括^(“合取”)、 V(“析取”)和 (“非”),以及**时态算子**,包括“总是”、 “终将”、“下一个时刻”和U(“直到”)等组成。 目前计算机存在多种时序逻辑,可分为线性时序逻辑和分支时序逻辑。  定义7.2:线性时序逻辑是时序逻辑的一种,此逻辑把状态树看成是有限条或无限条路径的集合,一条路径即代表着系统一饮可能的运行情况。  定义7.3:分支时序逻辑是时序逻辑的一种,可以描述状态的前后关系和分支情况。CTL表示的性质与当前状态相关,与选取该状态的路径无关  定义7.4:状态(state) 是描述一个系统的核心概念,表示程序在执行中某个确定时刻的一些信息。 状态可以简单地描述程序在某个给定点的抽象实体。  定义7.5:状态转换系统(state transition system)在计算机科学理论中是用来研究计算机的抽象机。它由一系列状态和状态之间转换组成，可由（S,S0,R）表示  定义7.6:自动机(automaton)作为一种计算模型,是基于完整的无穷语言上的有穷目动计算框架。无穷语言上的自动机分为“字自动机”和“树自动机”两大类。这两类自动机最初是用来解决二阶逻辑的判定问题。  目前,无穷语言上的自动机主要应用于以下几个方面:建模。自动机本身具有转换结构,因而可以用来描述变迁系统的转换行为。比如在SPIN中系统就被转化为一个非确定的Büchi自动机이,它允许使用同样的表示方法描述待验证系统和规约。判定。 许多时序逻辑的可判定性可转化为自动机判定问题-一这时自动机视为时序逻辑公式的一个有穷模型。此外,从逻辑到自动机的转化建立了模型和规约表示之间的一致性。规约。自动机本身也可以作为规约,同时自动机也可以作为规约的连接。  定义7.7:一个Büchi自动机(Bü chi automaton) 是一个定义在无限输入序列上的自动机,而不是处理有穷序列的标准有限状态机。 它接收一个无限输入序列作为它的语言的条件是:对于一个有穷状态自动机{S,So,L,T,F},其中S表示有限状态集合,so为初始状态,L为有限标签集合,T为状态转换关系,F为最终状态集合,无穷的运行序列被接收当且仅当F中至少有一个状态无限次地出现。  在漏洞分析工作中,**模型检测方法首先对给定的系统和安全缺陷属性进行建模,待检验的系统模型一般使用状态转换系统M来描述,而系统的安全缺陷属性则使用逻辑表达式s来表示**。**通过遍历状态空间自动地验证状态系统是否满足系统安全缺陷属性的问题即判断M=s?。**对于有限状态系统,这个问题是可判定的,可利用计算机程序在有限时间内自动确定。  **2. 技术框架**  通过对基本概念的了解，模型检测的框架包括：  ①对目标程序源代码进行建模：可被检查工具接受的描述模型成为状态转换系统，可用有限状态机表示。构建模型遵从最小原则和充分性原则  ②提取要验证的安全缺陷属性：通常由时序逻辑公式（如CTL、LTL）形式给出，描述系统的行为怎样随时间变化  模型检测，即匹配 ：根据相应的系统模型和安全缺陷属性进行检查,通过遍历有限状态机来检验系统是否具有时序逻辑所表达的性质。理论上检查的过程是完全自动化的,然而在实际应用中往往需要手工协助, 主要是分析检查的结果。 如果满足安全缺陷属性,检查工具给出一条错误路径,帮助使用者来查找错误在哪里以及是怎样发生的**。一个错误路径的产生原因之一**可能是因为**错误的模型**或者不**正确的安全缺陷属性**所导致的,称为伪错误。 这种情况下,使用者需要修改模型并重新开始检查。 但是,错误路径同样可以被用来修正模型或者验证安全缺陷属性。 还有一种可能是检查任务不能正确地终止,这通常是由于计算能力的限制或者模型的过大导致了计算机内存的溢出。 在这种情况下,需要调整验证工具参数或者修改模型(增加约束或者抽象)之后重新检查。  3．方法特点  模型检测用状态转换系统表示系统的行为,用时序逻辑公式描述系统的性质,然后用数学问题 “状态转换系统是否是该逻辑公式的一个模型”判定系统是否具有所期望的性质。  模型检测方法的优势主要体现在:1）自动化程度高,能够自动地执行,以批处理的方式一次性检验多条规约,能够很好地处理并发对象和线程的动态创建。2）模型检测可检验多种属性。 待验证的性质往往采用时序逻辑公式书写,时序逻辑能够非常灵活地表述诸如“安全性(safety)”“活性(liveness)”等时序性质。3）对于有限状态系统而言,该算法是可终止的。该问题本身是一个可判定的问题,在足够大的时空开销内,模型检测算法总会终止。4）模型检测能够给出诊断信息,当发现规约不被满足时,会生成一条反例路径~反例路径是模型中的一条执行序列,该序列能够揭示规约性质为何被违反。反例路径可以作为诊断信息,帮助设计者修正系统的错误。  该方法的不足主要表现在:1）它在检测程序漏洞过程中需要穷举所有可能的状态,当相应数据密集度较大时,存在着难以解决的状态空间爆炸问题,这就使得无穷系统的模型检测问题是不可判定的。2）在建模过程中,该方法对时序、执行路径等安全属性的边界处近似处理难度较大。 3)在软件系统中存在着值域无限的数据类型以及过程、递归结构,由此导致软件系统的数据无限性和控制无限性。 4)由于模型检测所针对的检查对象是模型而非程序本身,将程序向模型转化的过程中所使用的抽象技术可能使模型与程序不一致或者存在偏差,由此导致最终的检查结果无法准确反映实际程序中存在的漏洞情况。   1. **方法实现—基本原理的实现**   目前时序逻辑模型检测在模型检测中仍处于核心地位。模型检测中可综合利用程序静态分析和动态分析等技术,对程序源代码中漏洞进行有效的检测。 抽象解释技术能够把程序运行过程中无限状态空间转换为有限状态空间,动态分析技术可以准确定位程序缺陷代码,静态程序切片技术用于优化程序执行状态的搜索空间,最后通过模型检测算法分析程序漏洞。   1. **程序建模**   (1)程序代码抽象  基于抽象解释原理，通过去除与程序安全缺陷属性验证性质无关的信息，使得分析系统可以将其与缺陷属性进行匹配，从而检测程序中是否含有漏洞。  程序抽象的基本思想:用户定义要抽查的相关的部分数据的抽象函数、通过转换工具得到生成这些数据的状态流图或者基于抽象数据执行过程得到的抽象系统。过程以安全缺陷属性引导原则为指导。  对于抽象过程中存在的程序状态空间爆炸等问题国际上的优化技术：  a．状态合并的规约方法  b．数据抽象方法  (2)状态转换系统的构建  在抽象程序转换为状态转换系统时，需要进行系统状态图的可达性分析。  几个原则：  形式化构建程序状态模型：安全缺陷属性引导原则  最小原则  充分原则  精确原则：检测出满足的实例，与源程序中存在程序漏洞互为充分必要   1. **安全缺陷属性描述**   指对实际编程开发中发现的不良编程习惯 程序逻辑错误代仍等质量缺陷代码进行总结抽象,然后使用形式化语言描述建模形成用于检测的时序缺陷模式。  时序逻辑公式可以形式化定义：，如果执行过程满足安全缺陷属性，则表明程序存在相应的安全缺陷。也可以表示为自动机的形式，其中每一个转换关系分别表示一个安全相关操作的执行，违背设定的安全缺陷属性的操作序列将导致系统进入到自动机的终止状态。步骤：  (1)确定自动机中状态的组成元素。  (2)确定状态值,即给出状态中各个元素的取值。 主要包括两个方面:D给出自动机中错误操作序列,用以描述根据检测漏洞规则设定的安全缺陷属性的操作序列,自动机中必须包含一条或几条显式的错误操作序列。2确定自动机中正确操作序列,即不包含安全缺陷属性的操作序列,通常在描述安全规则的自动机中都会包含正确的操作序列。  (3)确定自动机的状态空间。 根据前面给出的每一条操作序列,确定该序列中的操作执行后系统应处于的状态。 将所有序列的状态相加,合并操作序列中相同的状态和最终的错误状态,可以得到自动机的状态空间。  (4)降低状态空间的复杂度,对状态空间降维优化处理。  (5)确定状态的转移。 每一条状态转移都由输入状态、触发条件和输出状态三个要素构成。 首先需要确定自动机中的操作序列的状态转移,然后考虑每个状态在各种操作执行时是否应该进行状态转移。 该步骤是自动机设计中的重点,需要对各个状态进行全面考虑。  (6)构建自动机。 根据状态转换关系绘制状态转换图,得到符合安全缺陷属性规则  的自动机模型。   1. **程序漏洞检查**   模型检测有多种算法，比如labeling算法和fixpoint算法。模型检测的过程可等价为系统状态  图的可达性分析过程  时序模型的检查结果需要进一步处理验证，以确定检查结果是否对应源程序中的真实漏洞，目前基于反例的抽象求精算法（CEGAR）   1. **典型工具**   **1. BLAST**  **2. MOPS** |
| **相关文件及参考资料：** 软件漏洞分析技术的第七章内容 |
| **遇到的问题及该阶段学习的想法：** 明显觉得到了这一章节之后多了很多名词和符号，代码示例也多了。 |
| **接下来的进度计划：** 阅读第九、十两章。 |