**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三章内容 |
| **本周实际完成进度：大概**了解几种漏洞分析技术，对漏洞分析技术有一定的认识。 |
| **详细内容：**   1. **漏洞分析技术体系**   **理论描述**  从分析对象、漏洞形态等因素出发，可以将软件漏洞分析划分为软件架构安全分析技术、源代码漏洞分析技术、二进制漏洞分析技术和运行系统漏洞分析技术。根据软件模型、黑盒、或白盒分析等因素又可以继续划分，包含各种具体技术。这些技术之间的关系具体参见书本p.57。（这是全书的技术概览）   1. **四种分析技术** 2. **软件架构安全分析**   软件架构是软件开发生命周期中编码和开发的基础，该分析是为了减少在软件设计过程的缺陷。其基本原理是：首先对软件架构进行建模，并对软件的安全需求或者安全机制进行描述，然后检查架构模型是否满足所有安全需求，如不满足，需要根据相应信息重新设计架构，直至满足所有安全需求。在设计模型时应使用相同的标准。目前，已产生的技术分为两大类：形式化分析和工程化分析。  ①形式化架构分析技术  主要使用形式化方法统一描述软件架构和安全需求，然后检测软件架构是否满足安全需求。具体方法见于p61表3.2。  ②工程化架构分析技术  该技术从攻击者角度考虑软件面临的安全问题，实用性强但自动化程度较低。包括场景分析法、威胁建模法、错误用例分析法。具体原理详见p61表3.2。   1. **源代码漏洞分析技术**   该分析技术是直接针对高级语言编写的程序进行分析以发现漏洞的方法。源代码是软件的最初原始形态，其安全缺陷是导致软件漏洞的直接根源。通常使用静态分析法，静态分析过程不受程序输入或者执行环境的影响。源代码漏洞分析一般包括四个步骤见p62表3.4。已有的源代码分析技术划分为两类：基于中间表示的分析技术和基于逻辑推理的分析技术。  ①基于中间表示的分析技术  以编译原理为主要基础技术，基本思想是首先将源代码翻译为便于分析的中间表示，同时根据需要构建一些用于分析的数据结构，如控制流图、调用图，然后根据一些预定义的分析规则对中间表示进行遍历，以判断分析规则所描述的漏洞是否存在。漏洞分析的主要过程也就是对中间表示的检查过程，将中间表示和一些检查规则作为输入，按照一定方式分析中间表示并对比检查规则，判断漏洞。  包括数据流分析、符号执行和污点传播。具体见p63 。  ②基于逻辑推理的分析技术  该漏洞分析方法以数学推理为基础，是一类不分析程序代码的语义等价模型的方法。将源代码进行形式化的描述，并在形式化的基础上，利用推理、证明等数学方法验证或者发现形式化的描述的一些性质，以此推断程序是否存在某种类型的漏洞。为了获取更多性质，研究人员通常借用形式化方法来扩展基本分析技术，代表性的技术有模型检测、定理证明。  源代码漏洞分析技术对比图见于p65 表3.3 。   1. **二进制漏洞分析**   该技术是一种面向二进制可执行代码的软件安全性分析技术，通过对二进制可执行代码进行多层次、多角度、内部的分析，发现软件中的漏洞。对象是源代码编译后的二进制代码。分析涵盖的技术有反汇编逆向分析、汇编代码结构化、中间表示、漏洞建模、动态数据流分析/污点分析、控制流分析/符号执行等。  从书中的表中，我们可以从软件运行的角度进行简单的理解：静态分析是通过反汇编得到模拟的源代码，然后中间构造表示，进行基于模式的漏洞分析和二进制代码比对；动静结合分析：动的一方面是对二进制代码输入数据得到指导信息再结合模拟的中间表示，进行动态污点分析和智能灰盒测试；而单一的测试属于动态分析。三者的技术比较见于书本p69表3.4 。  **4. 运行系统漏洞分析**  该分析具有一定特殊性具体表现在：运行系统比单个软件更加复杂，难度更大，分析人员只能通过像运行系统输入具体的数据并分析和验证输出的方式来分析漏洞，并且运行系统内部的信息往往不公开，分析人员需要利用手工或者利用工具获取信息。。目前来说，分析人员通过信息搜集、漏洞检测和漏洞确认三个步骤对运行系统进行漏洞分析。具体见p71图3.9 。而这小节的技术对比是漏洞检测技术的对比，书上很详细，但没有给出实现方法。 |
| **相关文件及参考资料：** 软件漏洞分析技术的第三章内容 |
| **遇到的问题及该阶段学习的想法：**第三章对已有的软件漏洞分析技术的基本原理及具体分类进行了概括性的介绍。问题就是：在源代码漏洞分析里基于中间表示的分析技术里包含三个子技术，我觉得这三个子技术对中间表示并没有支持。 |
| **接下来的进度计划：** 下周阅读第五、六章 |