**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十四章内容 |
| **本周实际完成进度：**阅读完第十四章内容“软件架构安全分析”，了解软件架构安全分析的过程。 |
| **详细内容：**   1. **基本原理**   **1. 定义**  软件架构是每个软件系统的核心，它是关于软件系统的一系列设计决策。软件架构是一系列相关的抽象模式,用于指导大型软件系统各个方面的设计。软件架构描述的对象是直接构成系统的抽象组件。各个组件之间的连接则明确和相对细致地描述组件之间的通讯。在实现阶段,这些抽象组件被细化为实际的组件,比如具体某个类或者对象。  **2. 概述**  (1) 软件的安全设计原则  总体而言,为了实现软件的功能性、易用性与安全性的统一,软件的安全设计原则主要包括最低权限原则、默认失效保护原则、保障最弱环节原则(或加固最脆弱环节原则)、深度防护原则、权限分离原则、机制节约原则(或简单化原则)、通用机制最少化原则、不信任隐私安全原则、完全中立原则、心理可接受原则、机密性原则等。  (2) 架构安全策略  针对影响软件安全的信任问题（数据不可信、身份不可信、环境不可信），制定的软件架构安全策略包括：完整性策略、认证机制、机密性策略。  **3. 架构安全的分析过程**  进行软件架构分析时首先要对架构进行建模,并对软件的安全需求或安全机制进行描述,然后检查架构模型是否满足安全需求,如不满足,需要根据相应信息重新设计架构,如此反复直至满足所有安全需求。   1. **方法实现**   **形式化分析：使用形式化的方法描述软件架构和安全需求，最终分析结果精确、可量化、自动化程度高，但实用性较差。工程化分析：从攻击者角度考虑软件面临的安全问题，实用性较强，自动化程度低。**   1. **形式化分析技术** 2. **UMLsec建模描述分析方法**   利用UML这个统一建模语言，对软件需求、软件行为、执行时的时序状态或者交互关系进行表示。  除此之外UMLsec方法可以通过拓展标准UML来进行架构及安全性的描述，UMLsec通过在UML元模型中增加安全相关的构造型、标记值、约束等建模元素,来表达安全属性的语义和系统需求。其中,构造型是一种修饰,允许为建模元素定义新的语义;标记值是可以与建模元素相关联的键值对,允许在建模元素上标注任何值;约束是定义模型外形的规则,可表示为任何形式的文本,或者用更正式的对象约束语言表示。  UMLsec安全属性添加过程如下:使用UML建立系统未包含安全属性的PIM(platform independent mode,平台无关模型);将该模型保存为xml文件;选定预先定义的与安全相关的Profile文件(该文件对不同的UML图所能添加的构造型进行了约束);向UML模型中的组件添加Profile文件中的构造型;添加后的文件保存为xml格式,通过可视化机制将其显示为使用UML表示的包含安全属性的PIM。   1. **软件架构模型结构化分析方法**   基于Petri网的结构化分析方法（SAM）有两种建模方法：一种是低层(命题模型,使用库迁移( place-transition)网和分支时序逻辑CTL;另一种是高层(一阶)模型,使用谓词/迁移( predicate/transition,PrT网和一阶时序逻辑。目前,一般使用后者来进行软件架构安全建模与分析,即利用PrT网描述软件架构的组件、连接器等架构基本元素以及访问控制及信息流相关的安全模型,而利用一阶时序逻辑来表达对软件架构本身的安全约束或安全需求。   1. **离散时间马尔可夫转移链（DTMC）**   基于DTMC的转移过程,软件可靠性模型评估的一般步骤如下 确定软件架构,分解并划分系统的功能模块;  定义模块的失效行为,估计模块的可靠度及模块间的状态转移概率,结合软件可靠性模型与软件的体系结构对软件可靠性进行评估。   1. **ACME组件系统架构描述方法**   它具有以下基本特性:提供了基本的体系结构元素来描述软件的体系结构,并提供了扩展机制;提供了注解机制( annotation mechanism)来描述软件的非结构性信息;提供了可对软件体系结构风格进行重用的模板机制;提供了一种开放的语义框架,可以对体系结构描述进行形式化的推理。除了可利用ACME来描述软件系统的体系结构外,架构师也可以利用它来实现不同的描述语言之间的转换,以便于软件设计人员之间的交流   1. **工程化分析技术**   **基于工程化的软件结构分析技术主要包括场景分析法、错误用例分析法和威胁建模法。**   1. **场景分析法**   场景分析法使用场景描述与软件架构的静态结构和动态行为相关的安全属性,从用户的角度出发,从场景角度分析,建立相应的场景库和评价指标树,并采用评审会议的方式分析架构安全,是一种轻量级的分析方法。  场景分析方法针对具体项目在应用领域中的定位,展开安全需求分析,汇总系统预期安全性能并按对功能进行分类以确保每项功能都能够得到详细描述,并为每个功能定义相应的场景,建立功能场景库。在建立功能场景库的时候,也可以根据情况为各个功能和相应的场景设置优先级,在实际开发的不同阶段,可以选择应用所有场景库进行分析,或只取部分优先级高的场景进行分析。   1. **错误用例分析法**   错误用例分析法通过检查软件架构对每个错误用例(用户在与软件交互过程中,对其他用户、软件等造成损失的一系列行为)如何反应来判断架构是否满足安全需求。错误用例分析法能够提高安全问题的可视化程度,并且不依赖于某种特定的架构描述语言,但需要人工评审,不能进行自动化分析。  错误用例分析法主要实现过程如下:确定用户及其行为;找出用例和错误用例;根据用例来进行架构设计,并根据错误用例来分析和评估架构;由于根据用例可能产生多个备选架构,因此需要根据错误用例及用例分析每个架构的劣势和优势,形成一个最佳架构   1. **威胁建模法**   威胁建模法的特点：STRIDE建模方法通过建立分层数据流图、标识软件的入口点和信任边界来描述软件架构,并通过建立威胁模型来展现假冒、篡改、否认、信息泄露、拒绝服务和特权提升等六类威胁,最终使用 SDL TM工具辅助完成分析。威胁建模法可以借助工具自动完成部分分析工作,实用性较强,但对于安全属性的描述能力较弱。  威胁建模的输入主要有以下四种形式:用例和使用方案、数据流、数据架构、部署关系图。它的输出结果是一个威胁模型,其获取的主要项目包括:关系和约束的逻辑形式的直接映射。在这个框架里的主要内容包括威胁列表和漏洞列表。  主要包括以下5个重要步骤：   1. 确定安全目标 2. 创建应用程序概述 3. 分解应用程序 4. 确定威胁 5. 确定漏洞 6. **典型工具**   **1. SDL安全威胁建模工具**  **2. 软件架构分析工具（Rational Software Modeler）** |
| **相关文件及参考资料：** 软件漏洞分析技术的第十四章内容 |
| **遇到的问题及该阶段学习的想法：这章内容是架构安全和运行系统漏洞分析模块的第一章，我认为是非常基于理论的，交代了各有不同的软件架构安全分析。** |
| **接下来的进度计划：**阅读第十五章 |