第十六章 安全性工程

王美红

主要内容

- 软件的安全性
- 安全性需求分析
- 网络中的安全性和保密性
- 安全性工程分析
- 安全性保证
- 安全性风险分析
- 传统软件工程活动的作用
- 可信性系统验证

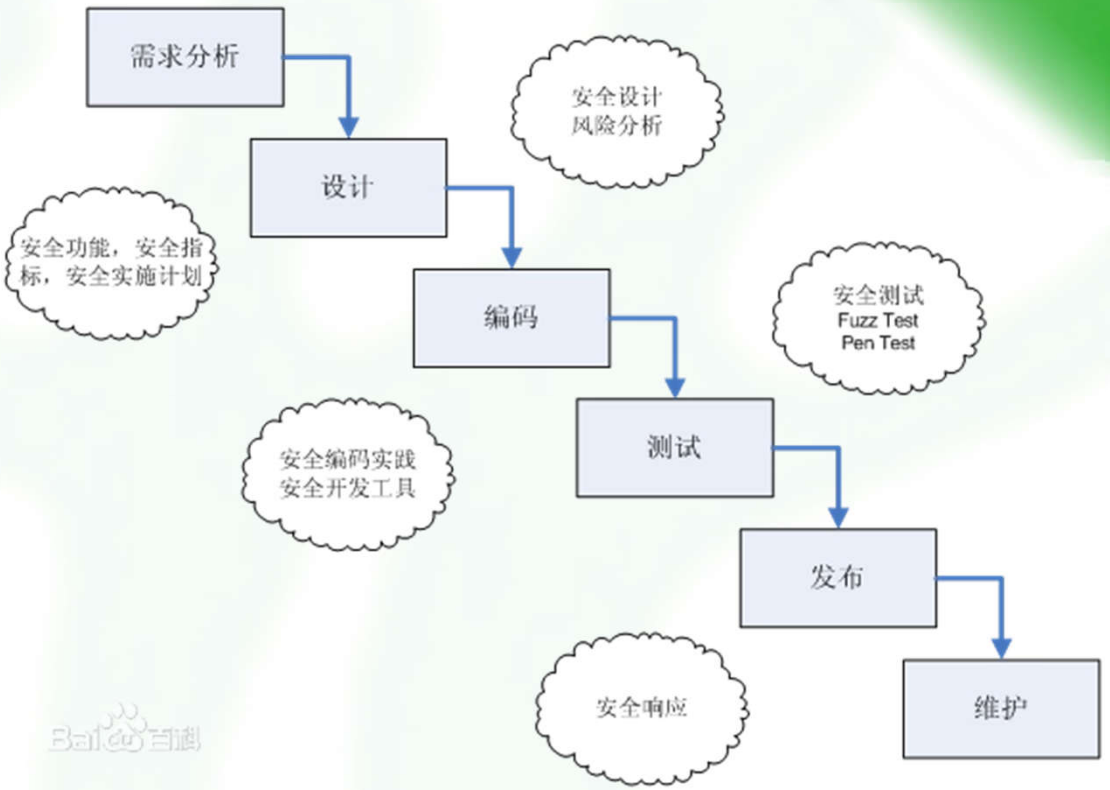
16.1 软件的安全性

- 几乎所有受软件控制的系统均会面临潜在对手的威胁。
  - 如用户隐私和个人信息可能丢失或被窃取的新的忧虑
- 软件工程师必须意识到这种威胁，并且要设计出具有可靠防卫性的系统，同时还要为客户提供有价值的产品。
- 软件安全性是软件质量保证的一个方面。
- 软件的安全性提供了使软件系统保护资产免于受到攻击的机制。
  - 资产包括数据信息、文件、程序、硬盘驱动器的存储空间、系统内存甚至处理器的容量。

16.2 安全生命周期模型SDL

- SDL即Security Development Lifecycle (SDL)，是微软根据多年实践经验提出的从安全角度指导软件开发过程的管理模式。
- SDL的核心理念就是将软件安全的考虑集成在软件开发的每一个阶段：需求分析、设计、编码、测试和维护。

SDL开发模式



微软SDL安全活动简图

SDL安全设计核心原则

- 攻击面最小化
- 基本隐私
- 权限最小化
- 默认安全
- 纵深防御
- 威胁建模
- http://blog.nsfocus.ne

降低默认执行的代码量；限制可访问到代码的人员范围；限定可访问到代码的人员身份；降低代码执行所需权限

履行法律规定和义务；增加客户的信赖；防止堵塞部署

普通管理员/系统管理员等角色管理；文件只读权限/文件访问权限等访问控制；进程/服务以所需最小用户权限运行

有利于更好的帮助客户掌握安全配置经验，同时也可以确保应用程序初始状态下处于较安全状态。

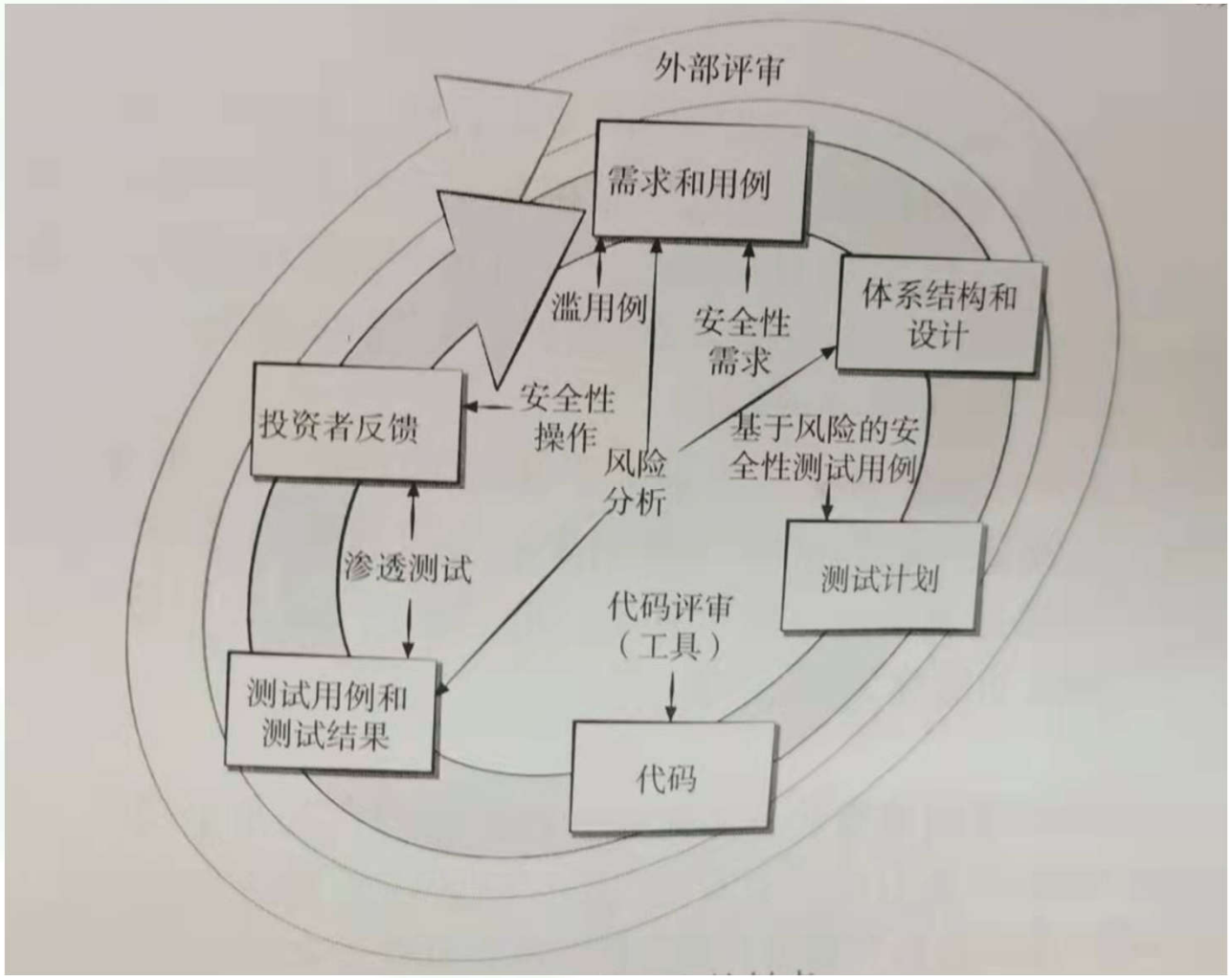
从不同的层面、不同的角度对系统做出整体的解决方案

与系统架构师及设计人员沟通，了解设计详情；使用成熟的威胁建模方法分析当前设计潜在的安全问题；提出安全建议及对潜在威胁的缓解措施；对安全设计进行验证并对整个设计方案进行回顾并再次确认

幻灯片 6

MOU1 Microsoft Office User, 2021-12-06

16.3安全开发生产周期活动



活动很重要，而不是模型。

16.4 安全性需求分析

- 软件的安全性需求由以下两个方面确定：
  - 一是与客户合作共同识别出的必须得到保护的资产；
  - 二是出现安全性漏洞时，这些资产受损的成本。
- 资产损失的价值被称为显露度。损失可用恢复或重建资产的时间和成本来度量。根据资产损失所造成破坏的修复成本进行优先排序。
- 在系统资产、系统漏洞和威胁识别出以后，可以制定控制措施，使系统即可避免受到攻击，又可缓解所遭受的破坏。

- 安全性需求的过程模型：
  - 核心安全需求工件
  - 软件成本压缩
  - SQUARE
  - 安全需求工程过程

建议在现有过程模型基础上，在术语表中添加安全性定义，识别潜在攻击，开发环节策略，并对候选安全需求进行分类和优先级排序。

SQUARE工程

表 18-1 SQUARE 过程

编号	步骤	输入	技术	参与者	输出
1	在定义上达成一致	IEEE 和其他标准中的候选定义	结构化访谈、小组座谈	利益相关者、需求团队	商定的定义
2	确定资产和安全目标	定义、候选资产和目标、商业驱动因素、政策和程序、示例	简易工作会议，调查、访谈	利益相关者、需求工程师	资产和安全目标
3	开发工件以支持安全需求定义	潜在的工件（例如场景、误用例、模板、表单）	工作会议	需求工程师	所需的工件：场景、误用例、模型、模板、表单
4	执行（安全）风险评估	误用例、场景、安全目标	风险评估方法、针对组织风险承受能力的预期风险分析，包括威胁分析	需求工程师、风险专家、利益相关者	风险评估结果
5	选择获取技术	目标、定义、候选技术、利益相关者的专业意见、组织风格、文化、所需的安全级别、成本收益分析等	工作会议	需求工程师	选定的获取技术
6	获取安全需求	工件、风险评估结果、所选择的技术	加速需求方法、联合应用开发、访谈、调查、基于模型的分析、清单、可重用需求类型列表、文档评审	需求工程师协助利益相关者	初始的安全要求
7	根据级别（系统、软件等）以及是需求还是其他类型的约束对需求进行分类	初始需求、体系结构	使用一组标准类别的工作会议	需求工程师、其他需要的专家	分好类的需求
8	对需求进行优先级排序	需求分类和风险评估结果	优先级排序方法，例如层次分析法（Analytical Hierarchy Process，AHP）、分诊、双赢等	需求工程师协助利益相关者	排好序的需求
9	检查需求	排好序的需求、候选正式检查技术	Fagan 和同行评审等检查方法	检验团队	初步选择的需求，决策过程和基本原理的文档

16.5 误用例、滥用例及攻击方式

- 误用（或滥用）例，可以帮助我们以与攻击者相同的方式观察软件。

16.5 网络中的安全性和保密性

- 为了使网站浏览器成为有效的用户界面，应该把保密、信任和安全当做最为重要的质量属性。
  - 社交媒体成为吸引恶意程序人员的目标
  - 移动APP外加了移动网络特有的新风险
  - 云计算服务商拥有全面访问和控制我们信息的能力，可能出现新型的内部恶意人员。
  - 物联网上任何实在的东西在互联网上都有其虚拟的实体，创新可能在不经意间侵犯人权。

16.4 安全性工程分析

- 安全性分析包括需求获取、威胁建模、风险分析、测度设计和正确性检查。除去商业理由以外，这些任务包括系统的功能性和非功能性细节的考量。

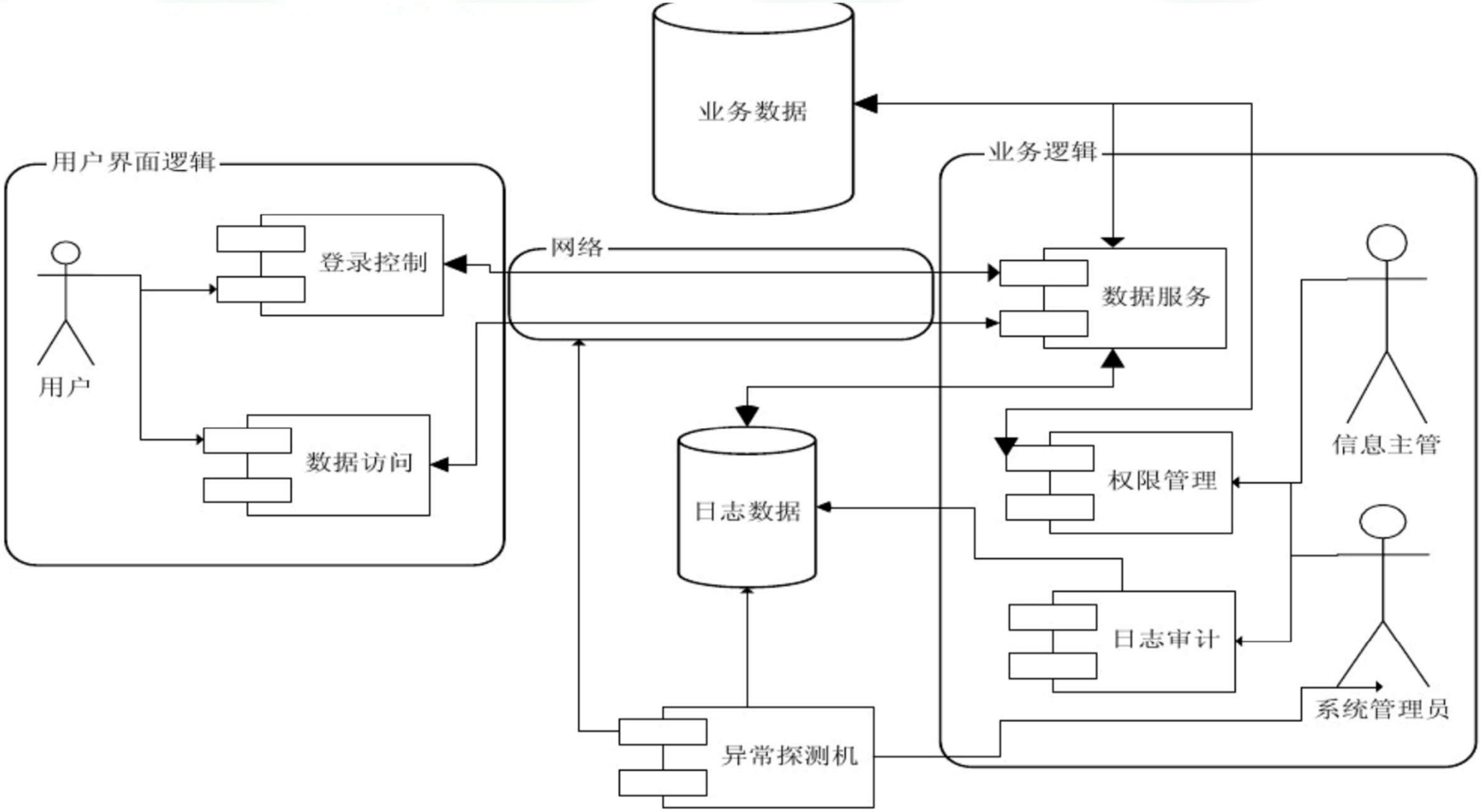
16.4.1 安全性需求获取

- 安全性需求获取对三个重要问题给出了回答：
  - 对于安全性软件，用户的要求是什么？
  - 如何设计安全体系结构，使其可提供良好的用户界面设计？
  - 如何设计良好的用户界面，使软件不仅安全性好，同时还能使它运行起来有效、高效，并且让用户满意？
- 在实施需求获取时，分析师应首先认清攻击模式。攻击模式是用于识别系统安全性缺陷的一种设计模式。通过为常用的安全性漏洞提出问题和解决方案，攻击模式可加速安全性分析。

16.4.2 安全性建模

- 建模是说明需求和分析需求的一个重要过程。
- 安全性模型是软件系统安全性策略的形式化描述，可以用文字或图形来表示。

安全模型举例


业务数据
用户界面逻辑
业务逻辑
登录控制
网络
数据服务
用户
数据访问
信息主管
日志数据
权限管理
日志审计
系统管理员
异常探测机

16.4.3 测度设计

- 安全软件必须具备三个属性：
  - 可靠性：软件可以在不友好的环境下运行
  - 可信性：系统不会在恶意的方式下运行
  - 存活性：在已妥协的情况下系统可继续运行
- 安全性度量应以测度为基础。
  - 三项测量：资产价值测度、威胁似然性测度和系统漏洞测度
  - 最佳测度是软件开发或运行期间现成的可用测度。如安全投诉数量、安全性测试用例的失效数量、攻击得逞的数量。

16.4.4 正确性检查

- 理论上，我们可以建立可证明的安全系统，但是实际上不可行。
- 安全性的正确性检查应该贯穿于整个软件开发周期。
- 安全标准的制定、安全性指南的开发，安全性测试用例的完备性等
- 内植于常规的软件工程任务的审核、审查和测试中。

16.5 安全性保证

- 安全性保证是为了向最终用户和其他利益相关者表明确已开发出一个安全产品，从而增强他们的信心。
- 由于着急把软件推向市场，使得项目经理往往更为关注项目的特性和功能，而把安全性列入次要地位。
- 安全性用例的三个目标：
  - 必须说明索赔要求对于该系统来说是适当的和可负担的。
  - 索赔要求是可以完成的。
  - 索赔的成果在风险要求的等级之内。
- 安全性用例支持索赔要求，因此说明这样的软件是安全的

16.6 安全性风险分析

- 威胁建模是一种安全性分析方法，可用于识别那些最有可能引发基于软件系统的破坏的威胁。

构建威胁模型的步骤

- 确认资产—列出所有的敏感信息和知识产权、存储位置、存储方式以及谁有访问权；
- 给出体系结构概述—写出系统用例并建立系统构建模型；
- 分解应用—保证在应用构件之间发送的所有数据都是有效的；
- 确认威胁—记录可能危及系统资产的所有威胁；
- 记录威胁—制定一个风险信息表，详细列出要监测和缓解的每项威胁；
- 评估威胁—根据影响大小和发生的可能性排序，以便区别对待。

16.7 传统软件工程活动的作用

- 忽略安全性问题，添加补丁的办法既是低效也是昂贵的
- 因不断的变更以及决策对安全性的影响，项目开始的时候，很难处理好所有安全性问题。
- 有效的软件过程包括一组合理的评审和调整措施。
- 制定计划时，项目预算和时间安排必须要把安全性问题考虑在内。

- 攻击面被定义为软件产品中一组可获取的和可利用的漏洞。先确定攻击面，再开发直接包含涉及攻击面的安全性规定在内的设计指南。
- 侧重于安全性问题的代码评审应作为实现活动一部分。应当根据设计活动中确定的相应安全性目标和威胁进行代码评审。
- 验证应包含安全性操作和资产归档规程的评审。安全性风险管理计划应作为维护过程度一部分进行定期的评审。

16.8 可信性系统验证

- “信任”表明一个系统实体（或组织）对另一个实体（或组织）相信的程度。
- 验证工作能确保使用基于测试、审查和分析技术的特定和可量化度量，以评估可信任系统的需求。
- 用来证明安全性用例的证据必须是可以接受的和有说服力的。
  - 比如验证登录，用户输入正常的值、一个不正确的值、一个空值和一个格式不正确的值，覆盖所有的逻辑路径，需要用256个测试用例。

- 在某种程度上，很多度量并没有考虑到一个现实问题，就是存在一些活跃的不良人员在不断地挖掘软件漏洞。
- 一些开发工具的使用可以帮助减少开发者引入系统漏洞的数量。
  - 代表性工具：PATS, ITS4, SLAM等。

作业

列出至少3～5种电子商务系统中的安全风险。