SOFTWARE TESTING

Software Testing

Software Safety & Security Testing

School of Computer Science & Engineering
Sun Yat-sen University

Instructor: Guoyang Cai
email: isscgy@mail.sysu.edu.cn

Approaches & Technologies

中山大學
SUN YAT-SEN UNIVERSITY

OUTLINE

- 9.1 软件的可靠安全性和保密安全性
- 9.2 软件安全性及测试
  - 软件安全性概述
  - 软件安全性分析
  - 软件安全性测试
- 9.3 软件安全及测试
  - 软件安全的概念
  - 软件安全漏洞
  - 软件安全技术
  - 软件安全测试

软件的可靠安全性和保密安全性

- 软件可靠安全性 (Software Safety) 简称软件安全性
  - 美国宇航局
    - 软件安全性是在整个软件生存周期运用系统安全性工程技术，确保软件采用以提高系统安全性的有效措施。它确保那些可能降低系统安全性的错误均已被排除或控制在可接受的风险水平。
  - 欧洲航空局 (欧洲太空总署 European Space Agency, ESA)
    - 软件安全性是软件在系统中执行而不致于在系统工作中造成不可接受的风险的能力。
  - GJB/Z 142-2004：军用软件安全性分析指南
    - 软件安全性是软件具有的不导致安全性事故发生的能力。
  - 上述定义的共同点：强调要在系统环境中讨论软件安全性；软件安全性是软件的一个质量属性或一种能力。

软件的可靠安全性和保密安全性

- 软件保密安全性 (Software Security) 简称软件安全
  - 软件保密安全性指软件在受到恶意攻击的情形下能够继续正确运行的能力以及确保软件能够在授权范围内合法使用的能力。
  - 网络空间安全

软件安全性概述

- 软件安全性的意义
  - 许多隐蔽性强的或非多发性的软件安全性错误很难被设计人员和测试人员所察觉，仅仅依靠设计技术的改进不足以解决安全性问题。
    - 需要一套严格的安全性分析程序和安全性分析方法，以预防系统发生安全性事故，或在发生安全性事故时降低其危害程度。
- 安全性关键软件
  - 软件应用于过程监控、实时控制、武器控制、航空航天、医疗、核反应等领域时，软件的错误能够通过软硬件接口使硬件发生故障或失效，从而造成应用系统的严重安全性事故。这类软件称为“安全性关键软件”。

软件安全性概述

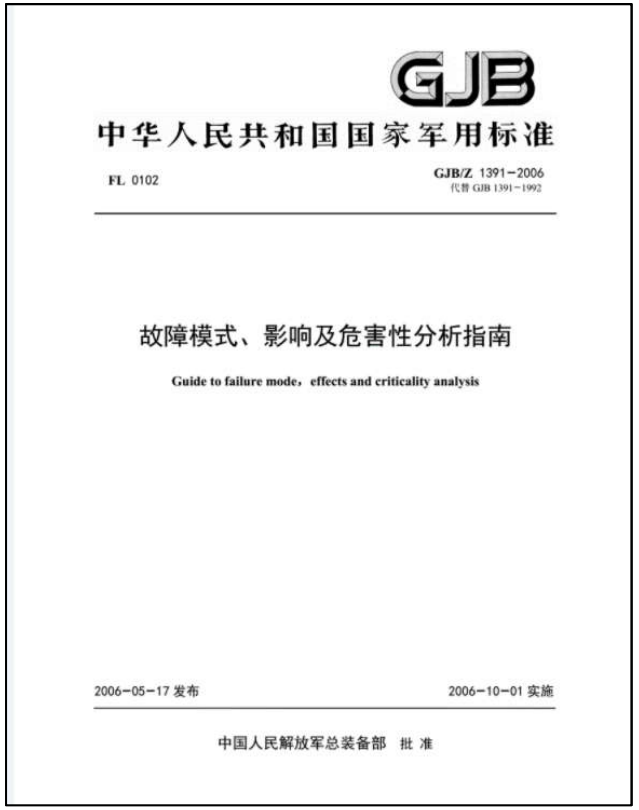
- 例：美军标-系统安全性 MIL-STD-882/1969, MIL-STD-882E/2012
  - 确定系统及系统中软件的安全性要求；
  - 将系统安全性说明中的要求准确转化为系统或分系统说明的要求，转化为软件需求说明的要求，并将这些要求在软件设计和编码中实现；
  - 在系统、分系统说明及软件需求说明中确定当可能发生安全性事故时的系统决策，这些决策包括失效安全、失效降级使用、失效容错使用等；
  - 确定软件系统中的安全性关键单元；
    - 安全性关键单元是指那些对系统安全性有关键性影响的程序、分程序和模块。
  - 对软件的安全性关键单元进行分析。
- 软件安全性工作内容包括软件安全性开发和软件安全性分析。

软件安全性开发

- 内容
  - 确定软件安全性需求，开展软件安全性设计、编程和测试。
- 目的
  - 全面、正确、合理地确定软件安全性需求，采用一系列的软件安全性设计和编程方法与准则，在软件设计和实现阶段加入软件安全性设计元素并实现。
- 方法
  - 确定软件一般安全性需求清单；
  - 制定软件安全性设计准则；
  - 定义软件编程语言的安全性子集或编程准则等。

软件安全性分析

- 例：GJB 1391A-2006：故障模式、影响及危害性分析指南



GJB

中华人民共和国国家军用标准

FL 0102

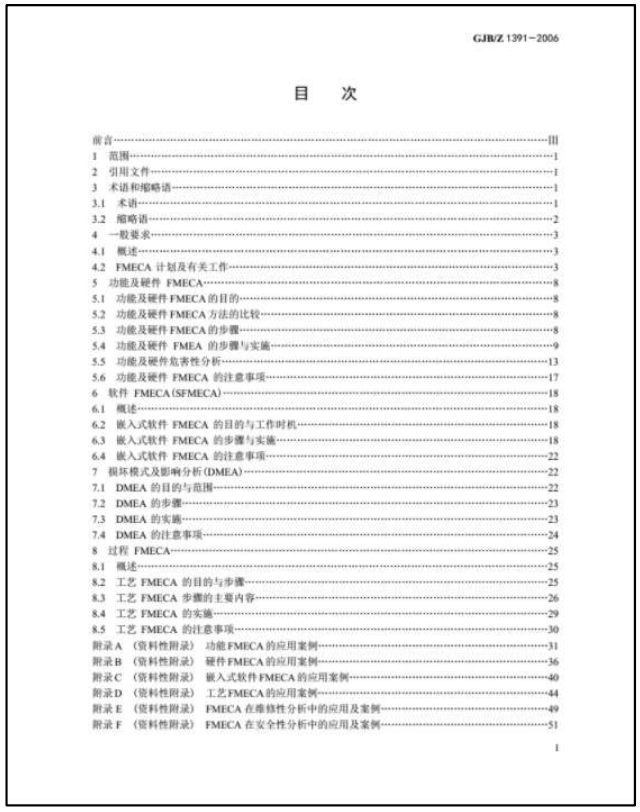
GJB/Z 1391—2006
代替 GJB 1391—1992

故障模式、影响及危害性分析指南

Guide to failure mode，effects and criticality analysis

2006—05—17 发布　　2006—10—01 实施

中国人民解放军总装备部　批 准



GJB/Z 1391—2006

目　次

I

软件安全性分析

软件安全性分析的内容

- 软件安全性分析采用一系列方法验证软件安全性需求的正确性、合理性和完备性，验证软件安全性设计和编程的正确性以及软件安全性测试的充分性。
- 例：MIL-STD-882E 规定的软件安全性分析内容
  - 软件需求危险分析
  - 概要设计及详细设计危险分析
  - 软件编程危险分析
  - 软件安全性测试
  - 软件与用户接口危险分析
  - 软件变更危险分析

- 软件安全性分析
  - 软件安全性分析的目的
    - 验证软件的全部安全性需求是否通过软件安全性的设计、编程和测试得到正确实现；
    - 发现软件安全性开发工作的不足，有利于进一步的完善。
  - 软件安全性分析的对象
    - 软件需求和设计文档；
    - 软件测试结果；
    - 与软件相关的接口、人员操作、硬件状态、硬件故障、系统时序等。

软件安全性分析

- 软件安全性分析的目标
  - 对安全性关键软件的运行环境、设计结构和测试结果等进行全面分析；
    - 从系统角度进行分析，并考虑软件使用过程中软件、硬件和操作人员的相互作用；
    - 分析软件可能的工作时序、适用条件、逻辑缺陷及其可能造成的不利影响。
  - 发现软件中与系统危险条件相关的设计缺陷及危险产生的条件，获得与软件相关的系统危险模式；
  - 分析危险的严重等级与发生概率；
  - 确认系统的危险风险指数；
  - 给出安全性评价。

- 软件安全性分析
  - 软件安全性分析的方法
    - 软件安全性流向分析
    - 时间、吞吐量和空间分析
    - 独立性分析
    - 设计逻辑分析
    - 设计数据分析
    - 设计接口分析
    - 测试覆盖分析
    - 测试结果分析
    - 软件失效模式及影响分析 (SFMEA)
    - 软件故障树分析 (SFTA)
    - 。 。 。

- 软件安全性分析
  - 软件安全性分析的特点
    - 软件安全性分析是系统安全性分析的一部分，必须在系统安全性分析的基础上进行。
      - 软件的逻辑、数据、时序等设计缺陷造成的不利影响，以及与软件相关的硬件故障和状态等都有可能引起软件失效，导致系统进人危险状态。
    - 软件安全性分析与软件测试的侧重点不同。
    - 软件与硬件的故障模式不同，因此分析的重点也不同。
    - 由于软件的特点和运行背景的多样化，软件安全性分析没有固定的分析模式和分析流程。

软件安全性分析

软件安全性分析的注意事项

- 每一个软件的功能、规模、实现逻辑、运行环境等均具有其特殊性，每一种分析技术也有其适用条件和局限性。
- 软件安全性分析要求不同领域的专家合作，从系统的角度考察软件、系统和操作人员之间的相互作用。
- 软件安全性分析是从系统顶级需求开始，自顶向下逐层逐级分析到软件功能层次，获得软件安全性需求，再从软件设计角度分析，自顶向下逐层分析软件设计缺陷对系统的不利影响。
- 分析内容的确定取决于软件本身的功能和系统设计要求。分析人员根据分析内容选择合适的分析技术，包括软件需求分析技术、软件可靠性分析技术和专门的安全性分析技术，对于比较复杂的系统甚至需要建立专门的仿真环境和分析工具进行计算机辅助分析。

软件安全性分析

软件安全性分析的常用技术

- 功能性危险分析 (FHA, Functional Hazard Analysis)
  - 自上而下确定系统功能故障状态，并对其影响进行评估。
    - 系统综合检查软件产品的各种功能，识别各种功能故障状态 (包括功能丧失和功能失效)，并根据其严重程度进行分类。
- 预先危险分析 (PHA, Preliminary Hazard Analysis)
  - PHA 也称初步危险分析，是一种对系统危险因素和危险程度进行定性分析评价的方法。
  - 在系统投入运行之前，对系统存在的危险类别、出现条件、可能造成的事故后果进行客观的概略分析。PHA 针对在程序设计、开发过程中需要跟踪、解决的危险及相关风险提供危险清单的初步框架，并记录通用的危险。

软件安全性分析

软件安全性分析的常用技术

软件失效模式与影响分析 SFMEA

- SFMEA 是传统的系统可靠性分析技术 FMEA (Potential Failure Mode and Effect Analysis) 的扩展。
  - FMEA 是一种自底向上分析技术，源于美国20世纪60年代的 Apollo 计划，由 MIL-STD-1629-1974 正式成型，并发展成为系统风险控制的主要手段之一。
  - FMEA 分析系统中所有可能产生的失效模式及其对系统造成的所有可能影响，按照每一个失效模式的严重程度、检测难易程度以及发生频度予以分类，进行归纳分析。

- 软件安全性分析
  - 软件安全性分析的常用技术
    - 软件失效模式与影响分析 SFMEA (续)
      - SFMEA 以软件故障模式为基础，以故障影响 (或后果) 为中心，根据分析层次和因果关系进行推理、归纳，以识别软件的薄弱环节，并提出改进措施建议。

软件安全性分析

软件安全性分析的常用技术 (续)

软件故障树分析 SFTA

- SFTA 是一种自顶向下的分析技术，来源于传统的系统分析技术 — 故障树分析技术 FTA (Fault Tree Analysis)，是一种重要的软件安全性与可靠性分析技术，特别适合在软件需求阶段进行分析。
- SFTA 选取对系统产生显著影响的失效事件作为顶事件，对引起系统失效的各种软件原因进行分析。

- 软件安全性测试
  - 软件安全性测试原则
    - 验证每一个软件安全性需求都有相应的软件安全性测试；
    - 证明每一个软件安全性需求都通过一个或多个测试得到满足；
    - 通过测试分析和软件工具对相关的危险和风险进行评估；
    - 判断给出的软件安全性测试已足够充分。
  - 软件安全性测试流程
    - 确认软件安全性测试规程；
    - 执行和监督软件安全性测试；
    - 整理和分析测试数据；
    - 检查测试失败的系统需求，必要时做回归测试；
    - 编写软件安全性测试报告。

- 软件安全性测试
  - 软件安全性测试方法
    - 基于可靠性分析法的软件安全性测试方法
      - 基于软件故障树 SFTA 的软件安全性测试方法
        - 利用 SFTA 的最小割集生成软件安全性测试用例。
      - 基于 Petri 网的软件安全性测试方法
        - 利用 Petri 网简洁、直观、潜在模拟能力强等特点，在因果关系作用下进行推演。
    - 基于软件测试方法的软件安全性测试方法
      - 基于猜错法的软件安全性测试方法
      - 基于接口的软件安全性测试方法

- 软件安全性测试
  - 软件安全性测试方法
    - 基于形式化模型的软件安全性测试方法
      - 形式化方法的基本思想是建立软件的数学模型，并在形式规格说明语言的支持下提供软件的形式规格说明。形式化方法可分为模型检验方法和定理证明方法。
      - 模型检验方法
        - 用状态迁移系统 S 描述软件的行为，用逻辑公式 F 表示软件执行必须满足的性质，通过自动搜索 S 中不满足公式 F 的状态来发现软件中的漏洞。
      - 定理证明方法
        - 将程序转换为逻辑公式，然后使用公理和规则证明程序是一个合法的定理。

- 软件安全的概念
  - 软件安全的焦点
    - 随着互联网和基于互联网的应用系统的发展，软件安全问题 (Software Security) 日益严重，其中的三个主要原因是：
      - 互联性
        - 软件系统与因特网的联系日益紧密。
      - 扩展性
        - 通过在线更新或者插入扩展件使软件系统升级。
      - 复杂性
        - 软件代码量增加；软件分布式开发、运行。
    - 软件安全关注能否保证软件在遭受恶意攻击时仍能正确运行。
      - 软件安全性 (Safety) 和软件安全 (Security) 的重要区别在于是否存在专业人员对系统进行恶意侵害、攻击和破坏。
    - 软件出现安全问题的根源是软件存在的安全漏洞。

软件安全的概念

软件安全的层次

软件安全一般分为两个层次：应用级别安全和系统级别 (操作系统) 安全。

(1) 应用级别安全

- 应用级别安全也称应用软件系统安全，它确保在预期的安全条件下，用户只能访问应用软件系统的特定功能和限制数据等。应用级别安全包括应用软件安全和数据安全。
- 应用软件安全
  - 应用软件本身的安全问题主要由应用软件的安全漏洞导致，这些漏洞可以是设计缺陷或者代码缺陷，甚至可能是软件开发人员预留的后门。
- 应用数据安全
  - 应用数据安全应包括数据存储安全和数据传输安全。

- 软件安全的概念
  - 软件安全分层

(2) 系统级别安全

- 系统级别安全也称操作系统安全。
- 确保只有具备操作系统平台访问权限的用户才能访问操作系统资源，包括对操作系统的登录或远程访问。

软件安全漏洞

- 安全漏洞的概念
  - 安全漏洞或系统脆弱性特指有关硬件、软件或协议的逻辑设计、系统实现或系统安全策略存在的缺陷或错误。
  - 安全漏洞一般是在系统具体实现和具体使用中产生的错误，但并不是系统中存在的错误都是安全漏洞，只有能威胁到系统安全的错误才是安全漏洞。
  - 从软件测试的角度看，不正确和不安全的开发过程可能导致产生软件安全漏洞。
    - 例如局部的执行错误、程序间的接口错误甚至更高级别的设计层面的错误。
  - 安全漏洞可被不法者利用，获得计算机系统的额外权限，在未授权或提高权限的情况下通过植入木马、病毒等方式攻击或控制计算机系统，窃取重要资料和信息，甚至破坏计算机系统。

- 软件安全漏洞
  - 安全漏洞产生的原因
    - 输入验证错误
      - 系统未对用户提供的输入数据的合法性作适当的检查。
    - 访问验证错误
      - 程序的访问验证部分存在某些可利用的逻辑错误，或用于验证的条件不足以确定用户的身份。
    - 竞争条件错误
      - 程序在处理文件等实体时在时序和同步方面存在问题，在处理的过程中可能存在一个机会窗口，使攻击者能够施加外来的影响。
    - 意外情况处置错误
      - 程序的实现逻辑没有考虑到一些应该处置的意外情况。
    - 其他错误如配置错误、环境错误、设计错误等

- 软件安全漏洞
  - 安全漏洞的危害
    - 系统的完整性 (Integrity)
      - 攻击者可利用漏洞入侵系统，对系统数据进行非法篡改，从而破坏数据完整性。
    - 系统的可用性 (Availability)
      - 攻击者利用漏洞破坏系统或者网络的正常运行，导致信息或网络服务不可用，无法满足合法用户的正常服务要求。
    - 系统的保密性 (Confidentiality)
      - 攻击者利用漏洞向非授权的个人和实体泄漏受保护信息。
    - 系统的可控性 (Controlability)
      - 攻击者利用漏洞对授权机构控制信息的保密性造成危害。
    - 系统的可靠性 (Reliability)
      - 攻击者利用漏洞对用户认可的质量特性造成危害。

- 软件安全漏洞
  - 安全漏洞的主要分类
    - 实现层面的错误
      - 缓冲区溢出
        - 当一个程序允许输入的数据大于已分配的缓冲区大小时，将会产生缓冲区溢出。
      - SQL 注入
        - 把 SQL 命令插入到 Web 表单递交或输入域名或页面请求的查询字符串，欺骗服务器执行恶意的 SQL 命令。
      - 跨站脚本
        - Web 服务器绑定的用户数据 (如在cookie中) 被泄露给恶意的第三方。

- 软件安全漏洞
  - 安全漏洞的主要分类
    - 实现层面的错误 (续)
      - 其他致命安全漏洞
        - 格式化字符串、整数溢出、命令注入、未能处理错误信息、未能保护网络流量、使用 Magic UPL 及隐藏表单字段、未能正确使用 SSL 和 TLS、使用基于弱口令的系统、未能安全存储数据、信息泄露、不恰当的文件访问、错误的网络域名解析、竞争条件、未认证的密钥交换、密码学强度随机数及不良可用性等。

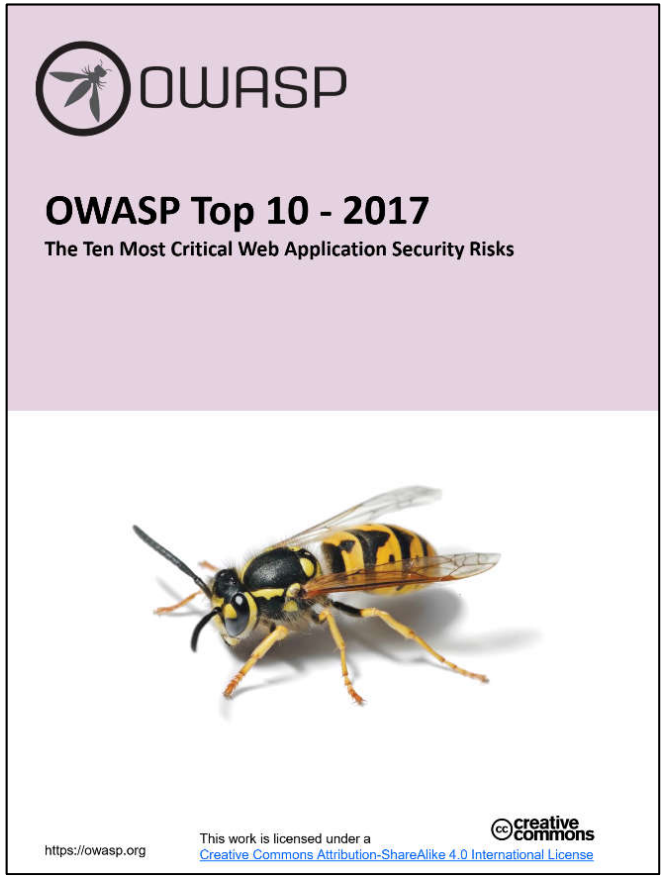
- 软件安全漏洞
  - 安全漏洞的主要分类
    - 设计层面的错误
      - 密码技术使用不当
      - 用户权限许可错误
      - 有缺陷的输入验证
      - 薄弱的结构性安全
      - 其他错误如：代码和数据混合、信任参数与外部环境相关、不安全的默认值、审计日志缺失等。
    - 设计层面的错误导致的漏洞是最普遍和最危险的，也是最难处理的缺陷类别。
      - 确定应用软件是否有设计层面的漏洞需要良好的专业知识背景支持，这使得发现这类缺陷很困难，而且难以实现过程的自动化。

- 软件安全漏洞
  - 安全漏洞的主要分类
    - 设计层面的问题实例
      - 面向对象系统中的出错处理
      - 对象共享和访问问题
      - 未保护的数据通道 (内部的和外部的)
      - 不正确或缺失的访问控制机制
      - 缺少审计或日志记录
      - 不正确的日志记录
      - 顺序错误和同步错误 (特别是在多线程系统中)

软件安全漏洞

安全漏洞的主要分类

- OWASP (the Open Web Application Security Project®) Top 10 (2017)
  https://owasp.org/www-project-top-ten/2017/



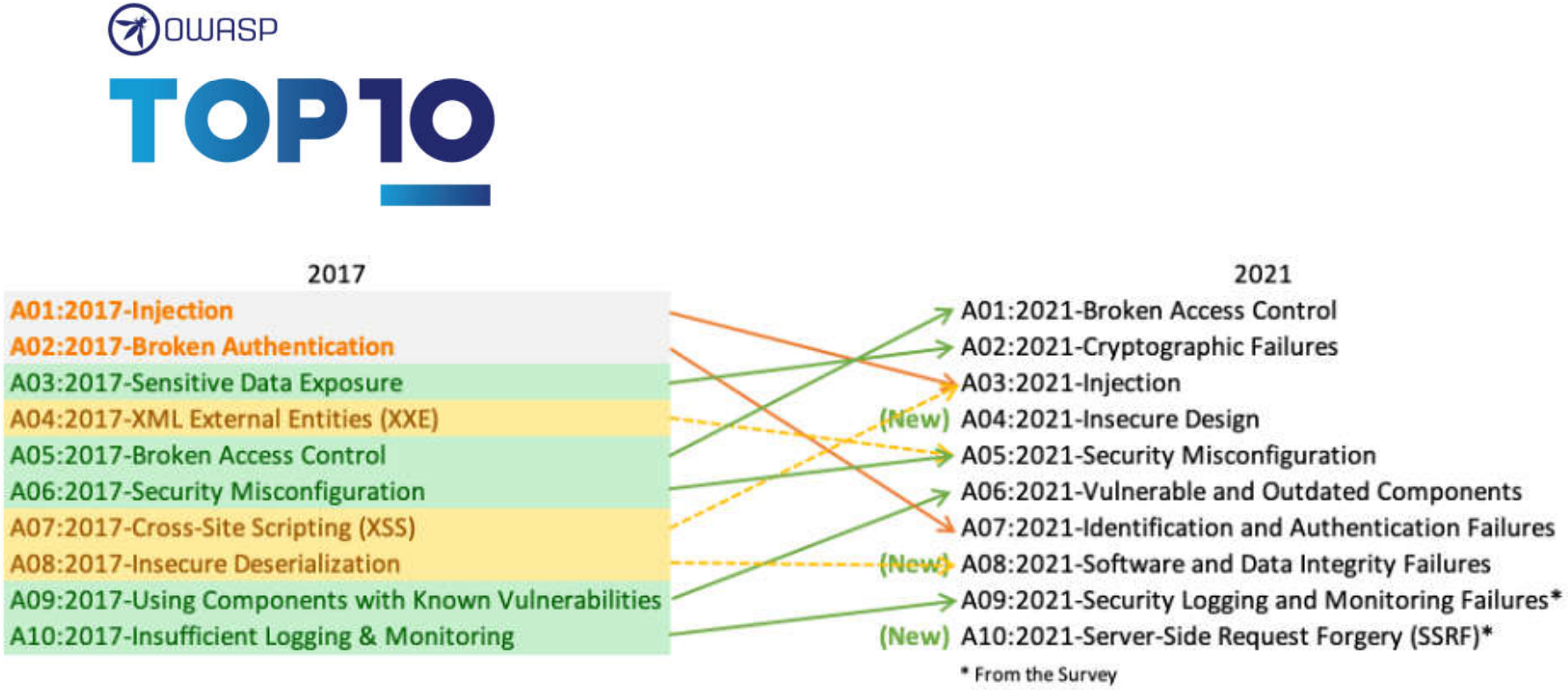
软件安全漏洞

安全漏洞的主要分类

OWASP (the Open Web Application Security Project®) Top 10 (2021)

https://owasp.org/www-project-top-ten/

Welcome to the OWASP Top 10 - 2021



软件安全技术

- 密码体系
  - 密码学研究数据的加密、解密及其变换，涵盖了数学、计算机科学、电子与通信等学科。
  - 密码技术主要包括：
    - 密码算法
    - 密码分析
    - 安全协议
    - 身份认证
    - 消息认证
    - 数字签名
    - 密钥管理
  - 密码技术不仅服务于信息的加、解密，还是身份认证、访问控制、数字签名等安全机制的基础，是信息安全的核心技术。

- 软件安全技术
  - 密码体系
    - 加密技术
      - 加密技术的基本思想就是伪装信息，使非法接入者无法理解信息的真正含义。信息借助加密技术以密文的方式进行存储，或通过数据通信网进行传输。
        - 即使发生数据被非法截取或因系统故障和操作人员的误操作造成数据泄漏，获得数据的未经授权者也不能理解数据的真正含义，从而达到信息保密的目的。
      - 网络数据常见的加密方式
        - 链路加密、节点加密、端到端加密。
      - 网络数据加密常见的算法
        - 古典密码算法、对称密钥算法、公钥制算法。

- 软件安全技术
  - 密码体系
    - 公钥体制
      - 公钥体制采用两个密钥将加密和解密能力分开，一个公钥作为加密密钥，一个私钥作为用户专有的解密密钥。
      - 通信双方无需预先交换密钥就可以实现保密通信。
    - 公钥基础设施 PKI
      - PKI 是一个基于公钥密码算法原理和技术的通用基础平台，提供全面的安全服务，如安全认证、密钥管理、数据完整性检验和不可否认性保证等。
      - 用户可以利用 PKI 平台提供的安全服务进行安全通信。
      - PKI 采用标准的密钥管理规则，能够为所有应用透明地提供加密和数字签名等密码服务所需要的密钥和证书管理 。

- 软件安全技术
  - 认证技术
    - 认证的概念
      - 防止消息被篡改、删除、重放和伪造的一种有效方法是使发送的消息具有被验证的能力。
      - 使接收者或第三方能够识别和确认消息的真伪，实现这类功能的密码系统称为认证系统 (Authentication System)。
      - 消息的认证性和消息的保密性不同。
    - 认证的内容和目的
      - 消息完整性认证
        - 验证信息在传送或存储过程中是否被篡改。
      - 身份认证
        - 验证消息的收发者是否持有正确的身份证据。
      - 消息序号和操作时间 (时间性) 等的认证
        - 防止消息重放或延迟等攻击。

- 软件安全技术
  - 认证技术
    - 认证技术的研究内容
      - 认证技术是防止不法分子对信息系统进行主动攻击的一种重要技术，它涉及的内容包括：
        - 认证技术的分层模型
        - 数字签名技术
        - 身份认证技术
        - 消息认证技术
        - 数字水印技术
        - 。。。

- 软件安全技术
  - 认证技术
    - 认证技术的分层模型
      - 安全管理协议
      - 认证体制
      - 密码体制
    - 一个安全的认证体制至少应该满足三个要求：
      - 接收者能够检验和证实消息的合法性、真实性和完整性；
      - 消息的发送者对所发出的消息不能抵赖，有时也要求消息的接收者不能否认收到的消息；
      - 合法的消息发送者的身份不能被其它方伪造。

- 软件安全技术
  - 认证技术
    - CA
      - 认证体制中通常存在一个可信中心或可信第三方 (TTP, Trusted Third Party)，称为认证机构 CA (Certificate Authority, 证书授权中心)。CA 仲裁、颁发证书或管理某些机密信息，通过数字证书实现公钥的分配和身份的认证。
      - CA 负责密钥的发放、注销及验证，所以 CA 也称密钥管理中心。
      - CA 为每个申请公开密钥的用户发放一个证书，证明该用户拥有证书中列出的公钥。
      - CA 的数字签名保证该证书不能伪造和篡改，因此，数字证书既能分配公钥，又实现了身份认证。

- 软件安全技术
  - 认证技术
    - 数字签名
      - 数字签名是信息发送者使用公开密钥算法技术所产生的其他人无法伪造的一段数字串。
      - 发送者用自己的私钥加密信息数据传给接收者，接收者用发送者的公钥解开数据后，可以确定消息来源合法，同时也是对发送者发送信息的一个证明。
      - 发送者对所发信息不能抵赖。
    - 身份认证
      - 身份认证又称身份鉴别，是指被认证方在没有泄露自己身份信息的前提下，能够以身份认证数据的方式来证明自己的身份，其目的是验证信息收发方是否持有合法的身份认证证据。

- 软件安全技术
  - 认证技术
    - 消息认证
      - 消息认证通过对消息或消息相关信息进行加密或签名变换进行认证，目的是防止传输和存储的消息被有意或无意篡改。
      - 消息认证包括消息内容认证 (消息完整性认证)、消息的源和宿认证 (身份认证) 及消息的序号和操作时间认证等。
    - 数字水印
      - 数字水印技术将特定的标记隐藏在数字产品中，用以证明原创者对产品的所有权，并作为起诉侵权者的证据。

- 软件安全技术
  - 漏洞扫描技术
    - 漏洞扫描对计算机系统或者其它网络设备进行安全相关检测，找出网络中存在的安全隐患和可能被黑客利用的漏洞，并针对每个具体漏洞给出一个详细的解决方案。
  - 防火墙技术
    - 防火墙技术是一种安全隔离技术，它通过在两个安全策略不同的网络之间设置屏障来控制两个网络之间的互访行为，阻止来自不明入侵者的所有通信。
    - 防火墙的功能
      - 过滤进出网络的数据包
      - 允许/禁止进出网络的访问行为
      - 记录通过防火墙的信息内容和活动
      - 对网络攻击进行检测和告警

- 软件安全技术
  - 入侵检测技术
    - 入侵是个广义的概念，不仅包括攻击者取得超出合法范围的系统控制权的事件，也包括收集漏洞信息、造成拒绝访问等对系统造成危害的行为。
    - 入侵检测 (Intrusion Detection, ID) 是对入侵行为的检测。
      - 入侵检测通过收集和分析网络行为、安全日志、审计数据、其它网络上可以获得的信息以及计算机系统中若干关键点的信息，检查网络或系统中是否存在违反安全策略的行为和被攻击的迹象。
    - 入侵检测作为一种主动安全防护技术，提供了对内部攻击、外部攻击和误操作的实时防护，在网络系统受损之前对攻击进行拦截和响应。
    - 入侵检测也被认为是防火墙之后的第二道安全闸门，它在不影响网络性能的情况下实现对网络的监测。

- 软件安全技术
  - 入侵检测技术
    - 入侵检测的主要内容
      - 监视、分析用户及系统活动
      - 系统构造和弱点的审计
      - 识别反映已知进攻的活动模式并向相关人员报警
      - 异常行为模式的统计分析
      - 评估重要系统和数据文件的完整性
      - 操作系统的审计跟踪管理，并识别用户违反安全策略的行为
    - 入侵检测系统 (IDS) 是入侵检测的软件与硬件的组合。

- 软件安全技术
  - 防病毒技术
    - 计算机病毒定义
      - 计算机病毒是编制或者在计算机程序中插入的破坏计算机功能或者毁坏数据、影响计算机使用、并能自我复制的一组计算机指令或者程序代码。
    - 计算机病毒特征
      - 潜伏性、传染性、破坏性、隐蔽性、多样性、触发性。
    - 计算机病毒的征兆
      - 磁盘文件数目增多，系统的可用内存空间变小，文件的日期/时间值被改变，可执行程序长度增加，磁盘上出现坏簇，程序加载时间比平时变长，程序执行时间较平时变长，硬盘读写时间明显增加，硬盘启动系统失败等。
    - 计算机病毒的主要危害

- 软件安全技术
  - 防病毒技术
    - 特征代码法
      - 采集已知病毒样本；
      - 在病毒样本中，抽取特征代码；
      - 在被检测文件中搜索病毒特征代码；
      - 特征代码法特点是速度慢、误报警率低、不能检查多形性病毒、不能对付隐蔽性病毒。
    - 校验和法
      - 计算正常文件内容的校验和，将该校验和写入文件中或写入辅助文件中保存。
      - 定期地或每次使用文件前，检查文件当前内容计算出的校验和与原来保存的校验和是否一致，从而判定文件是否感染病毒。

- 软件安全技术
  - 防病毒技术
    - 行为监测法
      - 利用病毒特有的行为特征来监测病毒。
    - 启发式扫描
      - 通过对病毒有关指令序列的反编译，逐步理解和确定其蕴藏的真正动机。
    - 虚拟机技术
      - 在虚拟机环境触发病毒。

- 软件安全技术
  - 虚拟专网技术
    - 采用隧道技术，将内部网络的数据加密封装后，通过虚拟的公网隧道进行传输，从而防止敏感数据被窃。
    - 虚拟专用网络 (VPN)
      - 将物理上分布在不同地点的局域网，通过公共网络构建成一个逻辑上的专用网络，实现安全可靠通信。
    - VPN 的安全技术包括
      - 隧道技术：在公用网建立一条数据通道—隧道，让数据包通过这条隧道传输。
      - 加密技术：包括加密、解密和密钥管理技术。
      - 认证技术：例如是使用者名称与密码或卡片式认证方式。
      - QoS技术：QoS 表示数据流通过网络时的性能，目的是向用户提供端到端的服务质量保证。

- 软件安全技术
  - 应用级别安全技术
    - 应用级别安全包括应用软件系统安全和数据安全两个方面。
    - 应用软件系统安全一般采用防火墙、防病毒及其他安全防范技术等措施实现，是属于被动型的安全措施。
    - 数据安全主要采用现代密码技术对数据进行主动的安全保护，如数据保密、数据完整性保护、身份认证等技术。

软件安全测试

软件安全测试的概念

- 一般的软件测试：确保软件能够实现预先设计的功能；确保软件不会去完成没有预先设计的功能。
- 软件安全测试：软件安全测试是在充分考虑软件安全问题的前提下进行的一类软件测试，用来验证集成在软件系统内部的安全保护机制是否能够在实际运行中保护系统免受非法的侵入，是验证应用软件的安全等级和识别潜在安全缺陷的过程。软件安全测试实际上是一轮多角度、全方位的软件安全攻击和反攻击过程。
- 软件安全测试采用两种不同的方法：
  - 测试安全机制，确保它们的功能被正确实现；
  - 模拟攻击者的攻击途径，促进基于风险的安全测试。

软件安全测试

软件安全测试的目的

- 软件安全测试的主要目的是查找软件设计和实现中存在的安全隐患，并检查软件对付非法侵入的防范能力。
  - 软件安全测试在系统受到真正的攻击之前尽可能多地找到软件中的安全漏洞，以降低软件发生安全事件的可能性。
  - 软件安全测试并非用于最终证明应用软件是安全的，而是用于验证所设立安全策略的有效性。这些策略基于威胁分析阶段所做的假设进行选择。
    - 例如，测试应用软件在出现非授权的内部或外部用户的访问或故意破坏等情况时的响应动作。
  - 不同的安全指标具有不同的测试策略。

软件安全测试

- 软件安全测试的内容
  - 全面检验软件在软件需求规格说明中规定的防止危险状态措施的有效性和在每一个危险状态下应该采取的反应；
  - 对软件设计中用于提高安全性的逻辑结构和处理方案进行针对性测试；
  - 在异常条件下测试软件，以表明不会因为可能的单个或多个输入错误而导致软件进入不安全状态；
  - 用错误的安全性关键操作进行测试，以验证系统对这些错误操作的反应；
  - 对安全性关键的软件单元功能模块单独进行加强测试以确认其满足安全性需求。
  - 安全测试主要关注的问题：
    - 缓冲区溢出、加密缺陷、错误处理不当、用户权限过大。

- 软件安全测试
  - 软件安全测试方法
    - 静态代码安全测试
      - 对源代码进行安全扫描，根据程序中数据流、控制流、语义等信息与其特有的软件安全规则库进行比对，从中找出代码中潜在的安全漏洞。
    - 动态渗透测试
      - 使用自动化工具或者人工方法模拟黑客的输入，针对应用系统可能存在的安全漏洞进行攻击性测试，最终获取访问关键隐私数据的权限，并评估系统的安全等级。
    - 程序数据扫描
      - 通常指内存测试。内存测试可以发现诸如缓冲区溢出之类的漏洞，这类漏洞使用其它测试手段一般难以发现。

软件安全测试

软件安全测试方法

Red-team testing 红队测试

- 红队测试类似 APT（高级可持续性威胁）攻击，对特定目标进行（相对）长时间的持续的综合性攻击，测试其安全防护体系的 PDR 能力 (prevention，detection and response)。
- 红队测试具有明确的测试目的，利用安全漏洞以及社会工程学、物理渗透、关联信息收集等综合手段规划攻击路径、制定攻击计划并实施攻击。
- 红队测试从开始准备到测试完成需要比较长的时间。

软件安全测试

软件安全测试内容分类

功能验证

- 采用黑盒测试方法，对涉及安全的软件功能如用户管理模块、权限管理、加密系统、认证系统等进行测试，验证上述功能是否有效。

漏洞扫描

- 漏洞扫描借助特定的漏洞扫描程序 (漏洞扫描器) 完成。
- 系统管理员通过漏洞扫描发现系统存在的安全漏洞，及时采取措施修补漏洞。
- 漏洞扫描可分为主机漏洞扫描和网络漏洞扫描两种类型：
  - 主机扫描：在系统本地运行检测系统漏洞的程序。
  - 网络扫描：基于网络运行远程检测目标网络和主机系统漏洞的程序。

软件安全测试

- 软件安全测试内容分类
  - 模拟攻击
    - 服务拒绝型攻击
      - 通过使服务器崩溃阻止提供服务，主要包括死亡之 ping 、泪滴、UDP 洪泛、SYN 洪泛、Land 攻击、Smurf 攻击 、Fraggle 攻击、电子邮件炸弹、畸形消息攻击等9种。
    - 漏洞木马型攻击
      - 模拟系统未及时对已知漏洞打补丁或者被植入木马等原因导致的非法人侵行为，主要包括口令猜测、特洛伊木马和缓冲区溢出等3种。
    - 伪装欺骗型攻击
      - 使目标配置不正确的消息，攻击技术主要包括 DNS 高速缓存污染、ARP 欺骗和 IP 欺骗。

- 软件安全测试
  - 软件安全测试内容分类
    - 侦听
      - 获取网络上传输的信息，用以管理网络、诊断网络问题、检查网络安全威胁。

- 软件安全测试
  - 软件安全测试过程
    - 反向安全测试过程
      - 反向安全测试过程从缺陷空间出发，建立缺陷威胁模型，通过威胁模型来寻找入侵点，对其进行已知漏洞扫描。
        - 目的是排除软件中已知类型的缺陷；成本较低。
        - 测试不完备，无法完整覆盖测试空间，无法发现未知的攻击手段，适合安全要求较低的软件。
    - 正向安全测试过程
      - 正向测试过程以测试空间为依据寻找缺陷和漏洞。
        - 测试比较充分，规避反向测试的不完备性；测试过的软件能够预防未知的攻击手段和方法；工作量较大。
    - 对于安全要求较高的软件，可以采用正向测试为主、反向测试为辅的方法进行安全测试。

- 软件安全测试
  - 软件安全测试工具
    - HP Fortify 软件安全中心
      - HP Fortify 套件提供三种软件安全测试能力：
        - 使用 HP Fortify Static Code Analyzer (SCA) 技术的静态应用安全测试；
        - 使用 HP WebInspect 技术的动态应用安全测试 (DAST)；
        - 实时混合分析：实时整合静态和动态应用的安全测试技术。

Lecture 27. Software Safety & Security Testing

End of Lecture