Lecture12 Security Engineering

1. 介绍

什么是安全工程

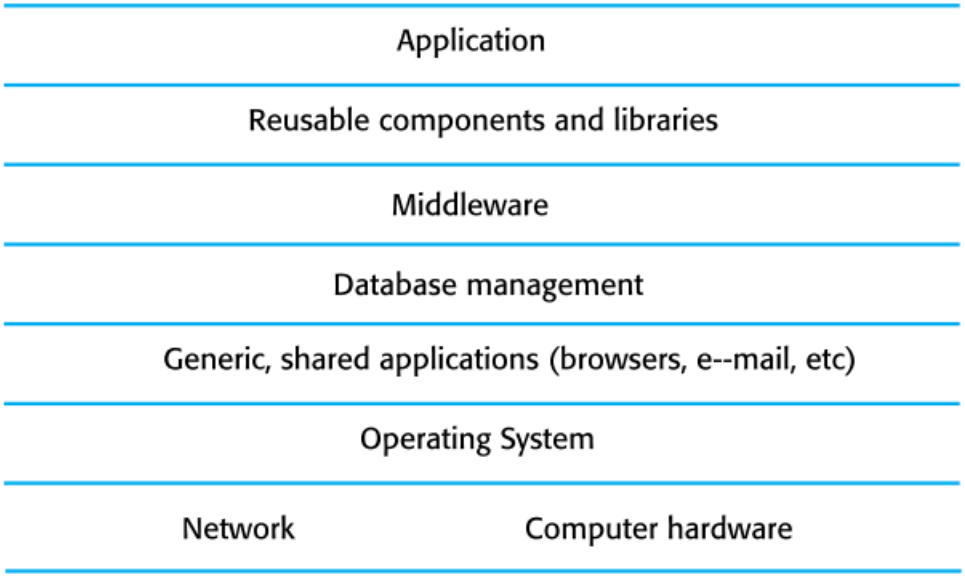
- 支持系统的开发和维护的工具、技术和方法，这些系统能够抵抗旨在破坏基于计算机的系统或其数据的恶意攻击
- 计算机安全这一更广泛领域的一个子领域

安全的维度

- **机密性 Confidentiality**
  - 系统中的**信息**可能会被**公开，或使未被授权访问该信息的人或程序可以访问**该信息
- **完整性 Integrity**
  - 系统中的**信息**可能**被损坏或损坏**，使其不寻常或不可靠
- **可用性 Availability**
  - **可能无法访问**正常情况下可用的系统或其数据

安全的层级

可能存在安全性系统层级



网路架构安全 Infrastructure security

- 维护为组织提供**网络架构**和一组**共享服务**的**所有系统和网络**的安全性
- 网络架构安全是一个系统管理问题，其中网络架构被**配置**为抵抗攻击

应用安全 Application security

- **单个应用**系统或相关系统组的安全
- 应用程序安全是一个软件工程问题，系统被**设计**用来抵抗攻击
- 用户及权限管理
  - 从系统中添加和删除用户，并为用户设置适当的权限
- 软件部署与维护
  - 安装应用软件和中间件，并配置这些系统，以避免漏洞
- 攻击监控、检测和恢复
  - 监控系统的未授权访问，设计抵御攻击的策略和开发备份和恢复策略

运行安全 Operational security

- 组织系统的安全操作和使用
- 主要是人类和社会问题
- 关心的是确保人们不采取可能危及系统本身的行动
  - 例如：告诉别人密码，让电脑保持登录状态
- 用户有时会采取不安全的行动，以便更容易地完成工作
- 因此，需要在系统安全性和系统有效性之间进行权衡

2. 安全性和可靠性

安全性

- 系统的安全性是一种系统属性，它反映了系统保护自身免受意外或蓄意外部攻击的能力
- 安全性是至关重要的，因为大多数系统都联网，以便外部访问系统
- 安全性是可用性、可靠性和安全性的基本先决条件

基本的安全性

- 如果一个系统是一个网络系统并且是不安全的，那么关于它的可靠性和安全性的声明是不可靠的
- 这些语句依赖于执行系统和开发的系统是相同的。然而，入侵可以改变正在执行的系统和/或其数据
- 因此，可靠性和安全性的保证不再有效

安全性术语

术语	定义	示例 Mentcare
资产 Asset	有价值的东西必须被保护起来。资产可以是软件系统本身或该系统使用的数据。	正在接受或已经接受治疗的每个病人的记录。
攻击 Attack	对系统漏洞的利用。通常，这是来自系统外部的蓄意破坏。	对授权用户的模拟
控制 Control	一种减少系统脆弱性的保护措施。加密是减少弱访问控制系统漏洞的控制的一个例子	一种密码检查系统，禁止使用字典中通常包含的专有名称或单词作为用户密码
暴露 Exposure	对计算系统可能造成的损失或损害。这可能会导致数据丢失或损坏，或者在出现安全漏洞后需要进行恢复时，可能会造成时间和精力的损失。	由于不相信诊所会维护他们的数据而不寻求治疗的未来患者的潜在经济损失。这位体育明星的诉讼造成的经济损失。声誉的损失。
威胁 Threat	有可能造成损失或伤害的情况。您可以将它们看作是受到攻击的系统漏洞。	未经授权的用户将通过猜测授权用户的凭证(登录名和密码)来访问系统。
脆弱性 Vulnerability	以计算机为基础的系统中的弱点，可能被利用来造成损失或伤害。	一个薄弱的密码系统，使用户很容易设置可猜测的密码

威胁类型

- **拦截威胁 Interception threats**
  - 允许攻击者访问资产
  - Mentcare系统可能面临的威胁可能是攻击者获得某个病人的记录
- **中断威胁 Interruption threats**
  - 允许攻击者使系统的一部分不可用
  - 一种可能的威胁可能是对系统数据库服务器的拒绝服务攻击，从而使数据库连接变得不可能
- **修改威胁 Modification threats**
  - 允许攻击者篡改系统资产
  - 在 Mentcare 系统中，修改威胁是攻击者改变或摧毁病人
- **捏造威胁 Fabrication threats**
  - 允许攻击者在系统中插入错误信息
  - 这在 Mentcare 系统中可能不是一个可信的威胁，但在银行系统中可能是一个威胁，在银行系统中，虚假交易可能会被添加到系统中，将钱转移到行凶者的银行账户。

安全保证

- **避免漏洞 Vulnerability avoidance**
  - 系统的设计是为了避免漏洞的出现。例如，如果没有外部网络连接，那么外部攻击是不可能的
- **攻击检测与消除 Attack detection and elimination**
  - 该系统被设计成能够检测到对漏洞的攻击，并在它们导致暴露之前消除它们。例如，病毒检查程序在病毒感染系统之前发现并清除病毒
- **暴露限制和恢复 Exposure limitation and recovery**
  - 系统的设计使成功攻击的不利后果最小化。例如，备份策略允许对损坏的信息进行恢复

安全性和可靠性

- **安全性和可靠性 Security and reliability**
  - 如果系统受到攻击，并且系统或其数据被破坏，那么这可能会导致系统故障，从而影响系统的可靠性
- **安全性和可用性 Security and availability**
  - 对基于 web 的系统的一种常见攻击是拒绝服务攻击，在这种攻击中，web 服务器会被来自各种不同来源的服务请求淹没。此攻击的目的是使系统不可用
- **安全性和防范性 Security and safety**
  - 破坏系统或其数据的攻击意味着关于安全性的假设可能不成立。安全检查依赖于分析安全关键软件的源代码，并假定执行的代码是该源代码的完全准确翻译。如果不是这样，可能会导致与安全相关的故障，为软件制定的安全案例是无效的
- **安全性和恢复性 Security and resilience**
  - 弹性是一种系统特性，反映了系统抵御和从破坏性事件中恢复的能力。网络软件系统最可能发生的破坏性事件是某种网络攻击，因此目前在恢复方面所做的大部分工作都旨在阻止、检测和恢复此类攻击

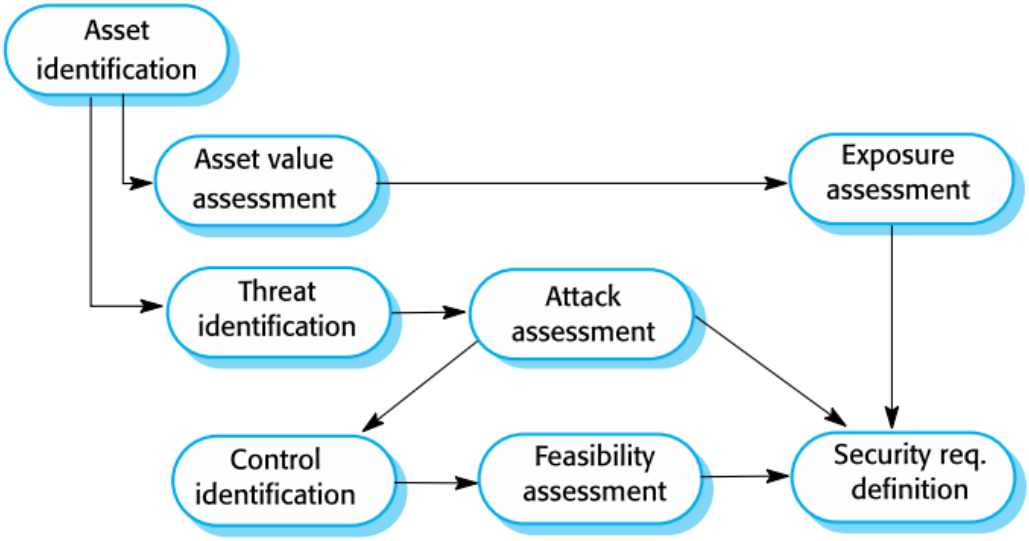
安全风险评估和管理

- 风险评估和管理涉及评估系统受到攻击可能导致的损失，并将这些损失与可能减少这些损失的安全程序的成本进行平衡
- 风险管理应该由一个组织的安全政策来驱动

评估管理内容

- **资产识别 Asset identification**
  - 确定必须保护的关键系统资产(或服务)
- **资产价值评估 Asset value assessment**
  - 评估已确定资产的价值
- **暴露评估 Exposure assessment**
  - 评估每个资产相关的潜在损失
- **威胁识别 Threat identification**
  - 确定对系统资产最有可能的威胁
- **攻击评估 Attack assessment**
  - 将威胁分解为对系统可能的攻击以及这些攻击发生的方式
- **控制识别 Control identification**
  - 提出可能用于保护资产的控制措施
- **可行性评估 Feasibility assessment**
  - 评估控制的技术可行性和成本
- **安全需求定义 Security requirements definition**
  - 定义系统安全需求，这些可以是基础设施或应用程序系统需求

初步风险评估 Preliminary risk assessment



- 初步风险评估的目的是识别适用于系统的一般风险，并决定是否可以以合理的成本达到适当的安全水平
- 风险评估应该集中于识别和分析系统的高层次风险
- 风险评估过程的结果用于帮助识别安全需求

设计风险评估 Design risk assessment

- 风险评估发生在系统开发生命周期期间，并由技术系统设计和实现决策通知
- 评估的结果可能导致安全性需求的更改和新需求的添加
- 识别已知的和潜在的漏洞，并使用这些知识来告知有关系统功能以及如何实现、测试和部署的决策制定

操作风险评估 Operational risk assessment

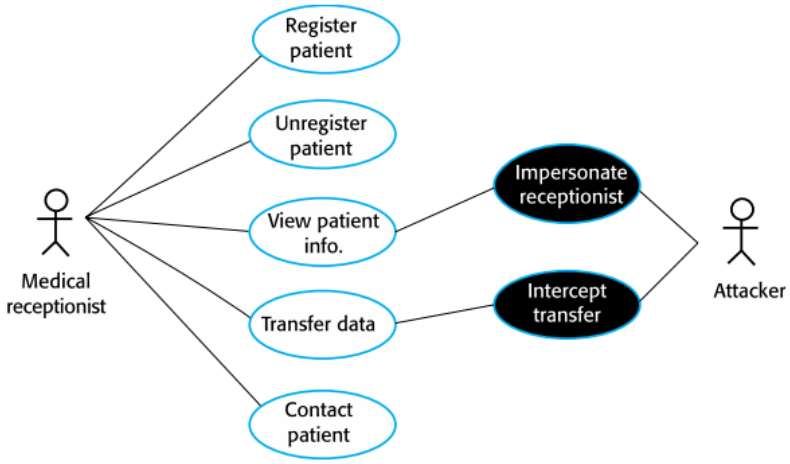
- 这类风险评估过程集中于系统的使用和可能由人类行为引起的风险
- 操作风险评估应在系统安装后继续进行，以考虑如何使用该系统
- 组织变更可能意味着系统以不同于最初计划的方式使用。这些变化导致新的安全需求，这些需求必须随着系统的发展而实现

Mentcare 系统的安全要求

- 患者信息应在门诊开始时下载到系统客户端上的一个安全区域，供临床工作人员使用
- 系统客户端的所有患者信息必须加密
- 患者信息应在门诊结束后上传至数据库，并从客户端计算机上删除
- 对数据库服务器所做的所有更改都必须维护在独立于数据库服务器的计算机上的日志

误用情况

- 误用情况是对系统造成威胁的实例



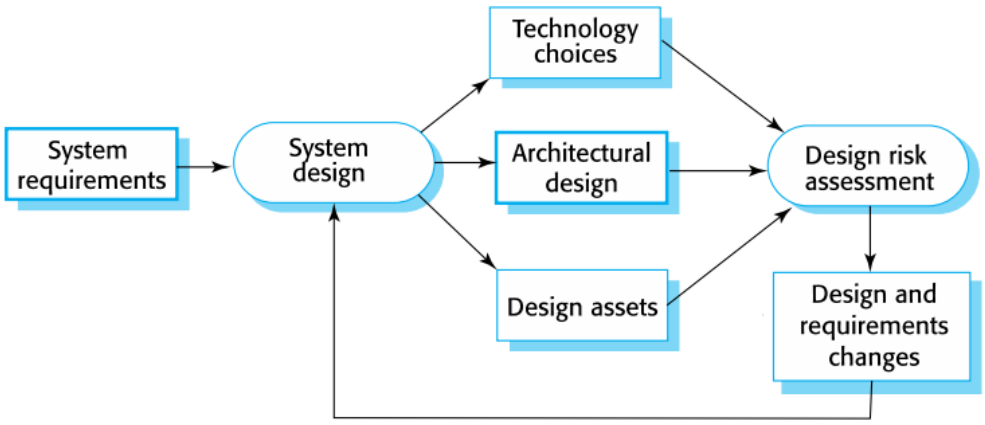
Mentcare 用例 —— 传输数据

流程	解释
Actors	医疗接待员，病人记录系统(PRS)
描述	接待员可以将数据从 Mentcare 系统转移到由卫生当局维护的一般患者记录数据库，传输的信息可以是更新的个人信息(地址、电话号码等)，也可以是患者诊断和治疗的摘要
数据	患者的个人信息，治疗方法
刺激	由医疗接待处发出的用户命令
回应	确认 PRS 已经更新
评论	接待员必须有适当的安全权限才能访问患者信息和 PRS

Mentcare 误用情况 —— intercept transfer

流程	解释
Actors	医疗接待员，病人记录系统(PRS)，攻击者
描述	接待员将数据从他或她的电脑传输到服务器上的 Mentcare 系统，攻击者拦截数据传输并获取该数据的副本
数据（资产）	患者个人信息，治疗总结
攻击	一个网络监控器被添加到系统中，并拦截从接待员到服务器的数据包。在接待员和数据库服务器之间设置一个欺骗服务器，使接待员相信他们正在与真实的系统交互
缓和	所有网络设备必须在锁着的房间内维护 使用设备的工程师必须经过认证 客户端和服务器之间的所有数据传输都必须加密 必须使用基于证书的客户端-服务器通信
需求	客户机和服务器之间的所有通信都必须使用安全套接字层(SSL)，https协议采用基于证书的认证和加密方式

3. 安全系统设计

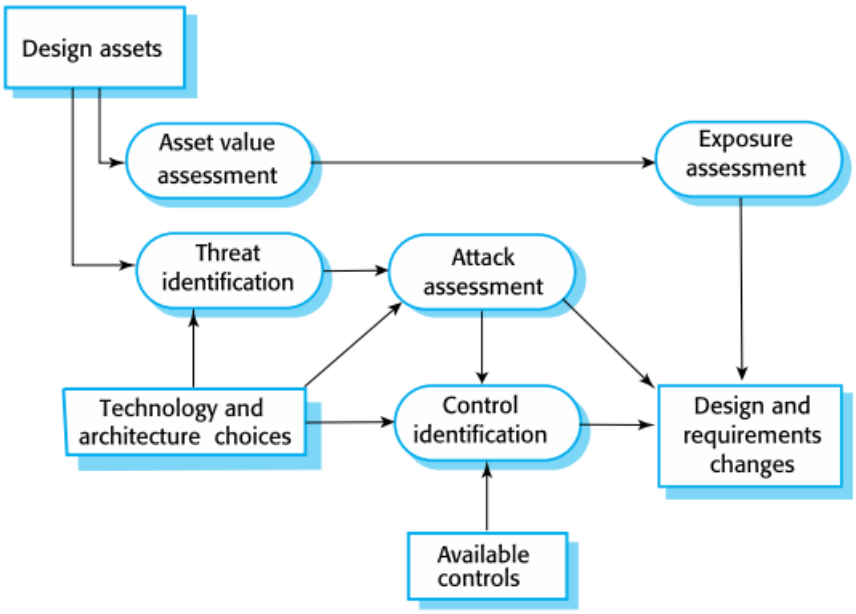


- 安全性应该设计到一个系统中——一个不安全的系统在设计或实现后很难安全
- 架构设计
  - 架构设计决策如何影响系统的安全性?
- 好的实践
  - 在设计安全系统时，什么是公认的良好实践

设计妥协

- 通过向系统中添加安全特性来增强系统的安全性，会影响系统的其他属性
- 性能
  - 额外的安全检查会降低系统的响应时间或吞吐量，因此可能会受到影响
- 可用性
  - 安全措施可能需要用户记住信息或需要额外的交互来完成事务，这会降低系统的可用性，并使系统用户感到沮丧

设计风险评估 Design risk assessment

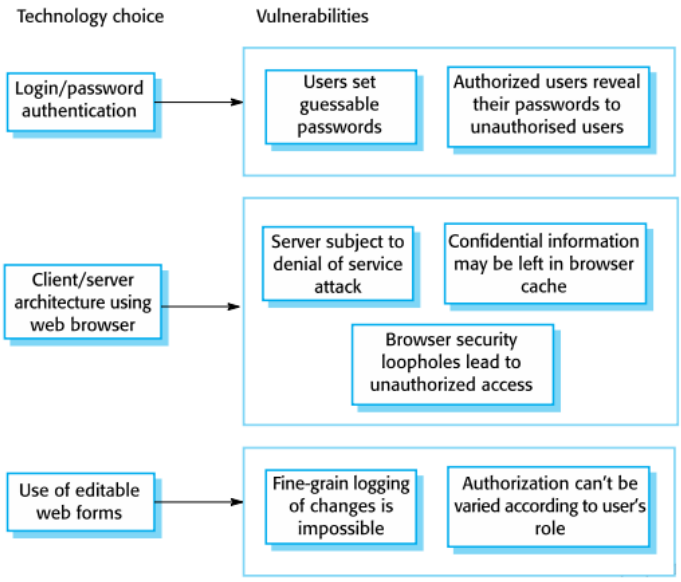


- 在系统开发期间和系统部署之后进行风险评估
- 更多的信息-系统平台，中间件和系统架构和数据组织
- 因此，可以确定由设计选择产生的漏洞

使用 COT 的设计决策

- 使用名称/密码组合进行身份验证的系统用户
- 系统架构是客户机-服务器，客户机通过标准的web浏览访问系统
- 信息显示为一个可编辑的web表单

与技术选择相关的漏洞



保护要求

- 当了解信息表示和系统分布时，可能会产生保护要求
- 分离患者和治疗信息限制了需要保护的信息量(患者个人数据)
- 在本地客户端维护记录的副本，可以防止对服务的拒绝服务攻击
    - 但这些可能需要加密

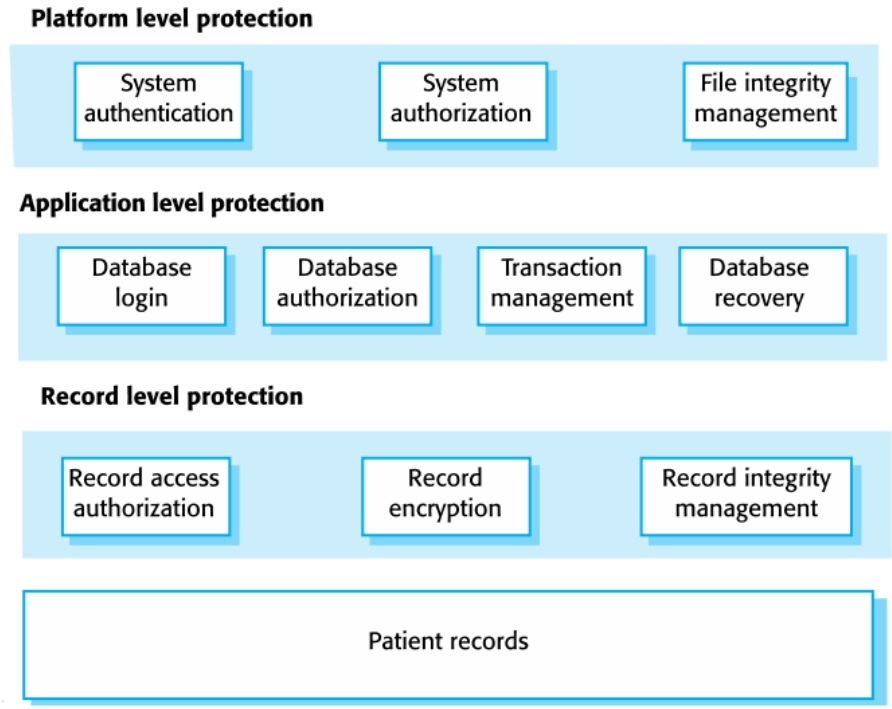
安全性要求

- 应提供密码检查器，并应每天运行。弱密码应报告给系统管理员
- 只有经过批准的客户端计算机才允许访问该系统
- 所有客户端计算机应该有一个单一的，由系统管理员安装的批准的网络浏览器

架构设计 Architectural design

- 在设计安全架构时，必须考虑两个基本问题
- 保护和分布
    - 如果资产是分布式的，那么保护它们的成本会更高
    - 如果资产受到保护，那么可用性和性能需求可能会受到影响

保护 Protection



如何组织系统以保护关键资产免受外部攻击

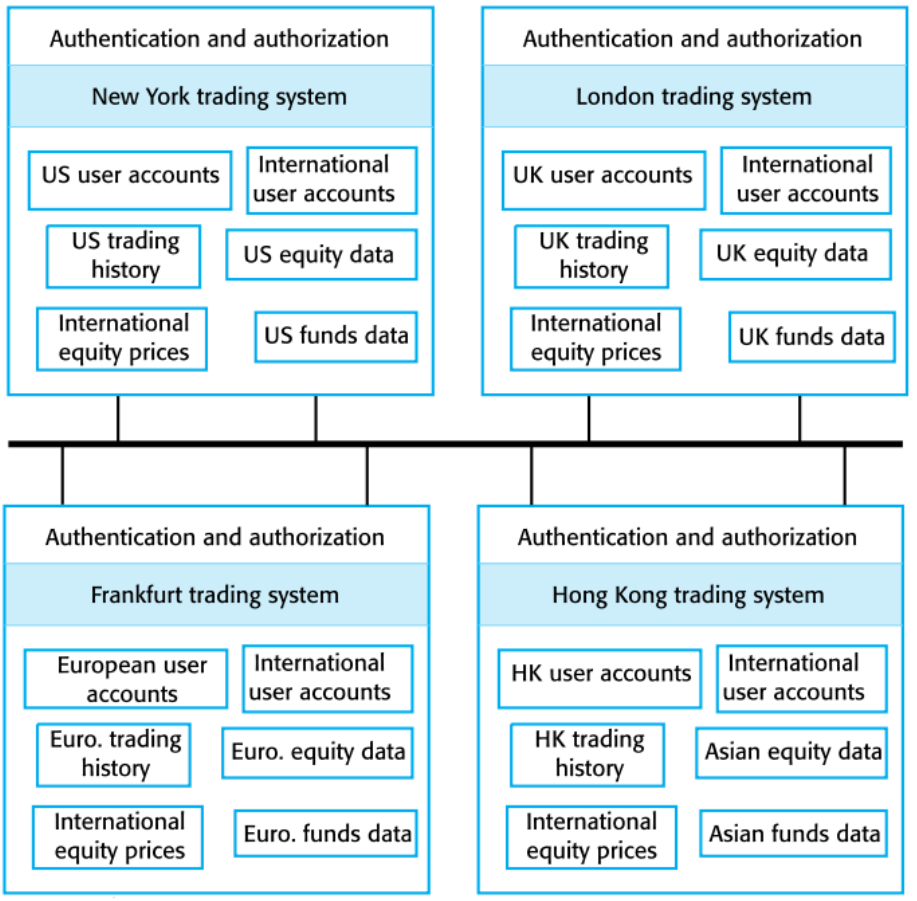
- 平台级保护 Platform-level protection
    - 系统运行平台上的顶级控件
- 应用程序级保护 Application-level protection
    - 特定的保护机制内置到应用程序本身
        - 例如额外的密码保护
- 记录级保护 Record-level protection
    - 当请求访问特定信息时调用的保护

分发 Distribution

如何分发系统资产，以使成功攻击的影响最小化

- 分布式资产意味着对一个系统的攻击不一定会导致系统服务的完全丧失
- 每个平台都有单独的保护特性，并且可能不同于其他平台，这样它们就不会共享一个共同的漏洞
- 如果拒绝服务攻击的风险很高，分发尤为重要

股票交易系统中的分布式资产



小结

设计指南 Design guidelines

- 设计指南封装了安全系统设计中的良好实践
- 设计指南有两个目的
  - 它们提高了软件工程团队对安全问题的认识，在做出设计决策时要考虑安全性
  - 它们可以作为系统验证过程中应用的检查清单的基础
- 这里的设计指南适用于软件规格说明和设计

设计指南	解释
基于明确的安全策略进行决策	为组织定义一个安全策略，该策略列出了应用于所有组织系统的基本安全需求
避免单点故障	确保只有在安全过程中有多个故障时才会导致安全故障，例如，有密码认证和问答认证
安全地失败	当系统因任何原因发生故障时，确保敏感信息不会被未经授权的用户访问，即使正常的安全程序不可用
平衡安全性和可用性	尽量避免使系统难以使用的安全程序，有时您必须接受较弱的安全性才能使系统更可用
记录用户操作	维护用户操作日志，可以通过分析来发现谁做了什么，如果用户知道这样的日志，他们就不太可能做出不负责任的行为
使用冗余和多样性来降低风险	保持数据的多个副本并使用不同的基础设施，这样基础设施漏洞就不会成为单点故障
指定所有系统输入的格式	如果已知输入格式，那么可以检查所有输入是否在范围内，以便意外输入不会导致问题
划分你的资产	组织系统，使资产位于不同的区域，用户只能访问他们需要的信息，而不是所有的系统信息
设计部署	设计系统以避免部署问题
设计的可恢复性	设计系统以简化攻击成功后的可恢复性

安全系统编程的各个方面

- 漏洞通常是特定于语言的
    - 数组绑定检查在 Java 等语言中是自动的，所以这不是一个可以在 Java 程序中利用的漏洞
    - 然而，数以百万计的程序都是用C和c++编写的，因为这两种语言可以开发更高效的软件，所以简单地避免使用这些语言并不是一个现实的选择
- 安全漏洞与程序的可靠性密切相关
    - 没有进行数组绑定检查的程序可能会崩溃，因此采取措施提高程序的可靠性也可以提高系统的安全性

可靠的编程指南

- 限制程序中信息的可见性
- 检查所有输入的有效性
- 为所有异常提供一个处理程序
- 尽量减少使用容易出错的结构
- 提供重新启动功能
- 检查数组边界
- 在调用外部组件时包含超时

- 命名所有代表真实值的常量

关键点

- **安全工程 Security engineering** 关心的是如何开发出能够抵御恶意攻击的系统
- **安全威胁 Security threats** 可以是对系统或其数据的机密性、完整性或可用性的威胁
- **安全风险管理 Security risk management** 涉及评估攻击可能造成的损失，并制定安全要求，将损失降至最低
- 要指定安全需求，您应该确定要保护的资产，并定义应该如何使用安全技术和技术
- **设计安全系统架构**时的关键问题包括组织系统结构以**保护**关键资产，以及**分发**系统资产以最大限度地减少成功攻击造成的损失
- 安全设计指导方针使系统设计人员对他们可能没有考虑到的安全问题更加敏感。它们为创建安全审查检查表提供了基础
- 安全验证之所以困难，是因为安全需求说明了系统中不应该发生什么，而不是应该发生什么。此外，系统攻击者是智能的，可能有更多的时间探测弱点，而不是用于安全测试