

NEW PLAN

软考高级架构师

• 一段新征程 •



安全架构设计



安全架构概述

- 对于信息系统来说，威胁可以是针对物理环境、通信链路、网络系统、操作系统、应用系统以及管理系统等方面。
 - 物理安全威胁是指对系统所用设备的威胁，如自然灾害、电源故障等；
 - 通信链路安全威胁是指在传输线路上安装窃听装置或对通信链路进行干扰；
 - 网络安全威胁是指通过技术手段窃取互联网信息，对网络形成严重的安全威胁；
 - 操作系统安全威胁是指对系统平台中的软件或硬件芯片中植入威胁，如“木马”和“陷阱门”、BIOS的万能密码；
 - 应用系统安全威胁是指对于网络服务或用户业务系统安全的威胁；
 - 管理系统安全威胁是指由于人员管理上疏忽而引发人为的安全漏洞，如人为的通过拷贝、拍照、抄录等手段盗取计算机信息。
- 具体来讲，常见的安全威胁有以下几种。
 - (1)信息泄露：信息被泄露或透露给某个非授权的实体。
 - (2)破坏信息的完整性：数据被非授权地进行增删、修改或破坏而受到损失。
 - (3)拒绝服务：对信息或其他资源的合法访问被无条件地阻止。
 - (4)非法使用(非授权访问)：某一资源被某个非授权的人或以非授权的方式使用。
 - (5)窃听：用各种可能的合法或非法的手段窃取系统中的信息资源和敏感信息。

安全架构概述

- (6)业务流分析：通过对系统进行**长期监听**，利用统计分析方法对诸如通信频度、通信的信息流向、通信总量的变化等态势进行研究，从而发现有价值的信息和规律。
- (7)假冒：通过欺骗通信系统(或用户)达到非法用户冒充成为合法用户，或者特权小的用户冒充成为特权大的用户的目的。**黑客大多是采用假冒进行攻击。**
- (8)旁路控制：攻击者利用系统的安全缺陷或安全性上的脆弱之处获得非授权的权利或特权。
- (9)授权侵犯：被授权以某一目的使用某一系统或资源的某个人，却将此权限用于其他非授权的目的，也称作“内部攻击”。
- (10)特洛伊木马：软件中含有一个察觉不出的或者无害的程序段，当它被执行时，会破坏用户的安全。
- (11)陷阱门(后门)：在某个系统或某个部件中设置了“机关”，使得当提供特定的输入数据时，允许违反安全策略。
- (12)抵赖：这是一种来自用户的攻击。例如：否认自己曾经发布过的某条消息、伪造一份对方来信等。
- (13)重放：所截获的某次合法的通信数据备份，出于非法的目的而被重新发送。
- (14)计算机病毒：所谓计算机病毒，是一种在计算机系统运行过程中能够实现传染和侵害的功能程序。一种病毒通常含有两个功能：一种功能是对其他程序产生“感染”；另外一种或者是引发损坏功能或者是一种植入攻击的能力。

安全架构概述

(15)人员渎职：一个授权的人为了钱或利益、或由于粗心，将信息泄露给一个非授权的人。

(16)媒体废弃：信息被从废弃的磁盘或打印过的存储介质中获得。

(17)物理侵入：侵入者通过绕过物理控制而获得对系统的访问。

(18)窃取：重要的安全物品，如令牌或身份卡被盗。

(19)业务欺骗：某一伪系统或系统部件欺骗合法的用户或系统自愿地放弃敏感信息。

- 安全架构是架构面向安全性方向上的一种细分，通常的产品安全架构、安全技术体系架构和审计架构可组成三道安全防线。
- 产品安全架构：构建产品安全质量属性的主要组成部分以及它们之间的关系。产品安全架构的目标是如何在不依赖外部防御系统的情况下，从源头打造自身安全的产品。
- 安全技术体系架构：构建安全技术体系的主要组成部分以及它们之间的关系。安全技术体系架构的任务是构建通用的安全技术基础设施，包括安全基础设施、安全工具和技术、安全组件与支持系统等，系统性地增强各产品的安全防御能力
- 审计架构：独立的审计部门或其所能提供的风险发现能力，审计的范围主要包括安全风险在内的所有风险。

安全模型

- 信息系统的安全目标是控制和管理主体(含用户和进程)对客体(含数据和程序)的访问。
- 安全模型是准确地描述安全的重要方面及其与系统行为的关系，安全策略是从安全角度为系统整体和构成它的组件提出基本的目标。安全模型提供了实现目标应该做什么，不应该做什么，具有实践指导意义，它给出了策略的形式。如下图是模型的分类：

基本模型: HRU

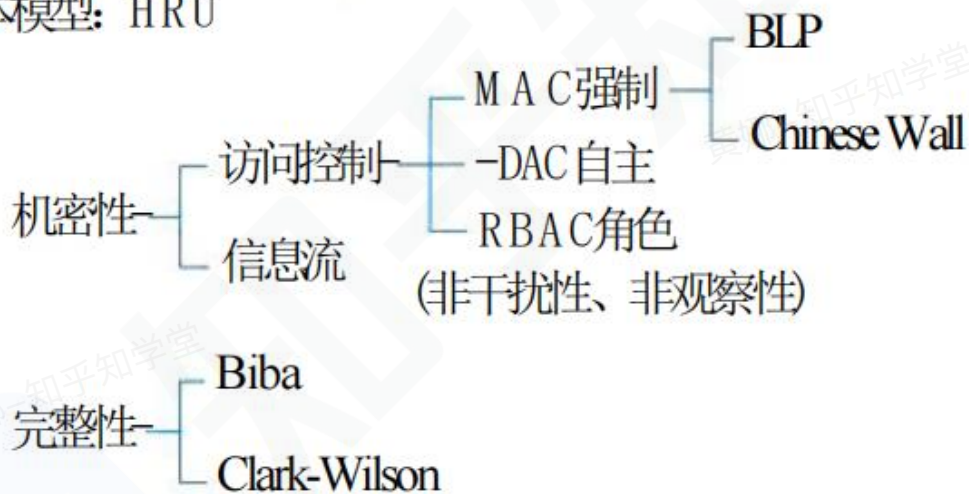


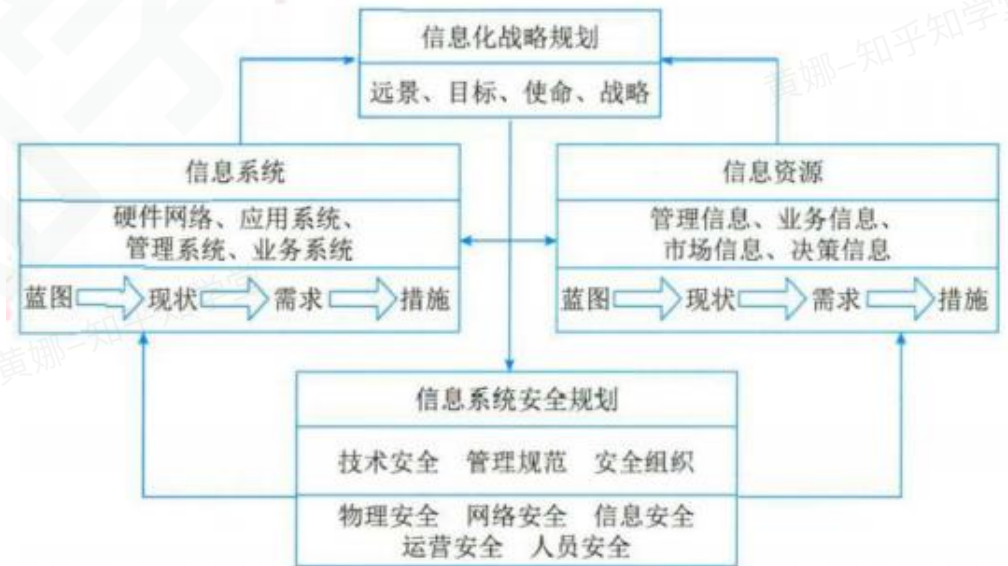
图18-3 安全模型的分类方法

系统安全体系架构规划框架

- 安全技术体系架构：对**组织机构信息技术系统的安全体系结构的整体描述**。安全技术体系架构的目标是建立可持续改进的安全技术体系架构的能力。根据风险威胁的存在实体划分出**5个层次**的实体对象：**应用、存储、主机、网络**和**物理**。
- 信息系统安全体系主要是由**技术体系(技术)、组织机构体系(人)和管理体系(制度)**三部分共同构成的。
 - 技术体系是**全面提供信息系统安全保护的技术保障系统**，该体系由物理安全技术和系统安全技术两大类构成。
 - 组织体系是信息系统的**组织保障系统**，由机构、岗位和人事三个模块构成。
 - 管理体系由**法律管理、制度管理和培训管理**三部分组成。

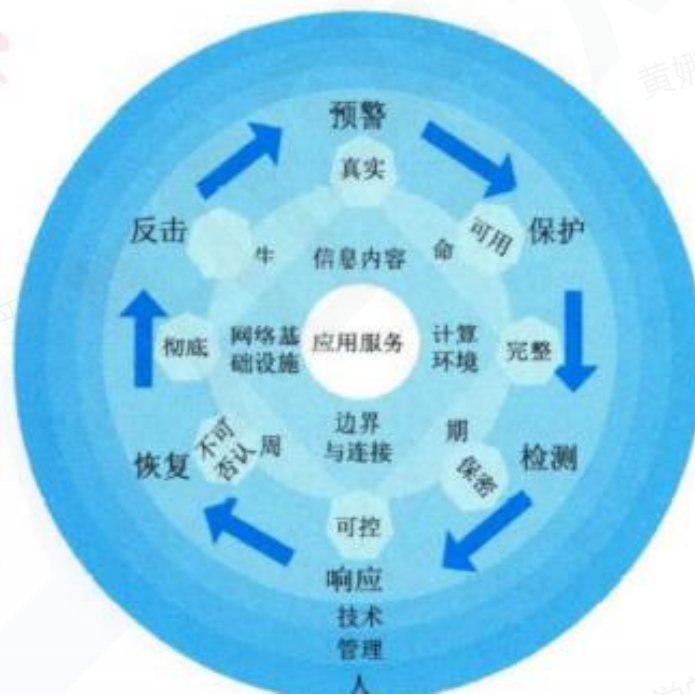
系统安全体系架构规划框架

- 信息系统安全规划依托企业信息化战略规划
 - 信息系统安全规划的目标应该与企业信息化的目标是一致的，而且应该比企业信息化的目标更具体明确、更贴近安全。
- 信息系统安全规划需要围绕技术安全、管理安全、组织安全考虑
- 信息系统安全规划以信息系统与信息资源的安全保护为核心，规划工作需要围绕着信息系统与信息资源的开发、利用和保护工作进行，要包括蓝图、现状、需求和措施4个方面。
 - 对信息系统与信息资源的规划需要从信息化建设的蓝图入手，知道企业信息化发展策略的总体目标和各阶段的实施目标，制定出信息系统安全的发展目标。
 - 对企业的信息化工作现状进行整体的、综合、全面的分析，找出过去工作中的优势与不足。
 - 根据信息化建设的目标提出未来几年的需求，这个需求最好可以分解成若干个小的方面，以便于今后的实施与落实
 - 要明确在实施工作阶段的具体措施与方法，提高规划工作的执行力度。



信息安全整体架构设计

- WPDRRC(Waring/Protect/Detect/React/Restore/Counterattack)是我国针对美国提出的PDRR的基础上新增了预警和反击功能的信息安全模型，它有6个环节和3大要素。
- 6个环节包括：预警、保护、检测、响应、恢复和反击，它们具有较强的时序性和动态性，能够较好地反映出信息系统安全保障体系的预警能力、保护能力、检测能力、响应能力、恢复能力和反击能力。
- 3大要素包括：人员、策略和技术。人员是核心，策略是桥梁，技术是保证，落实在WPDRRC 的6个环节的各个方面，将安全策预警略变为安全现实。





信息安全整体架构设计

- **W(Waring):** 预警主要是指利用远程安全评估系统提供的模拟攻击技术来检查系统存在的、可能被利用的薄弱环节，收集和测试网络与信息的安全风险所在，并以直观的方式进行报告，提供解决方案的建议，在经过分析后，分解网络的风险变化趋势和严重风险点，从而有效降低网络的总体风险，保护关键业务和数据。
- **P(Protect):** 防护通常是通过采用成熟的信息安全技术及方法来实现网络与信息的安全。主要内容有加密机制，数字签名机制，访问控制机制，认证机制，信息隐藏和防火墙技术等。
- **D(Detect):** 检测通过检测和监控网络以及系统，来发现新的威胁和弱点，强制执行安全策略。在这个过程中采用入侵检测、恶意代码过滤等技术，形成动态检测的制度，奖励报告协调机制，提高检测的实时性。主要内容有入侵检测，系统脆弱性检测，数据完整性检测和攻击性检测等。
- **R(React):** 响应是指在检测到安全漏洞和安全事件之后必须及时做出正确的响应，从而把系统调整到安全状态。为此需要相应的报警、跟踪、处理系统，其中处理包括了封堵、隔离、报告等能力。主要内容有应急策略、应急机制、应急手段、入侵过程分析和安全状态评估等。
- **R(Restore):** 恢复灾难恢复系统是当前网络、数据、服务受到黑客攻击并遭到破坏或影响后，通过必要技术手段，在尽可能短的时间内使系统恢复正常。主要内容有容错、冗余、备份、替换、修复和恢复等。
- **C(Counterattack):** 反击是指采用一切可能的高新技术手段，侦察、提取计算机犯罪分子的作案线索与犯罪证据，形成强有力的取证能力和依法打击手段。

信息安全整体架构设计

■ 信息系统安全设计重点考虑两个方面：**系统安全**保障体系和**信息安全**体系架构。

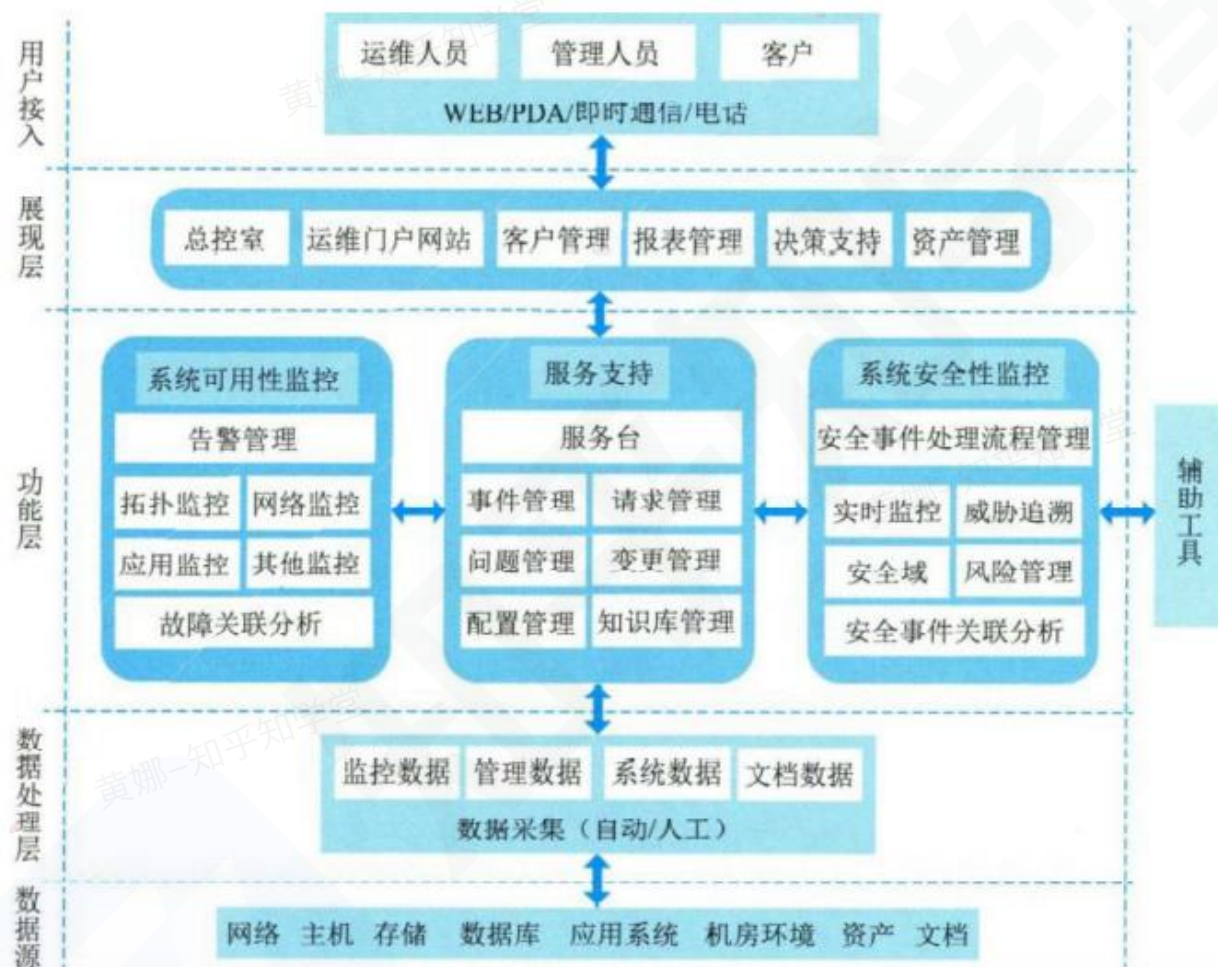
1.系统安全保障体系：是由**安全服务**、**协议层次**和**系统单元**等三个层面组成，且每个层都涵盖了安全管理的内容。系统安全保障体系设计工作主要考虑以下几点：

- **安全区域策略的确定**：根据安全区域的划分，主管部门应制定针对性的安全策略，比如涉及到金钱和隐私信息的区域的安全级别更高，涉及到外部人员的区域安全级别低一些。
- **统一配置和管理防病毒系统**：主管部门应当建立整体防御策略，以实现统一的配置和管理。
- **网络安全管理**：加强网络安全管理，制定有关规章制度。

2.信息安全体系架构：具体在**安全控制系统**，我们可以从下面5个方面开展分析和设计工作。

- **物理安全**：保证计算机信息系统各种设备的物理安全是保障整个网络系统安全的前提。包括：环境安全、设备安全、媒体安全等。
- **系统安全**：主要是指对信息系统组成中各个部件的安全要求。系统安全是系统整体安全的基础。它主要包括：网络结构安全、操作系统安全和应用系统安全。
- **网络安全**：是整个安全解决方案的关键。它主要包括：访问控制、通信保密、入侵检测、网络安全扫描系统和防病毒等。
- **应用安全**：主要是指多个用户使用网络系统时，对共享资源和信息存储操作所带来的安全问题。它主要包括资源共享和信息存储两个方面。
- **安全管理**：主要体现在三个方面。其一是制定健全的安全管理体制;其二是构建安全管理平台;其三是增强人员的安全防范意识。

信息安全整体架构设计-面向企业的安全控制系统安全架构



网络安全体系架构设计

- OSI定义了7层协议，其中除第5层(会话层)外，每一层均能提供相应的安全服务。实际上，最适合配置安全服务的是在物理层、网络层、运输层及应用层上，其他层都不宜配置安全服务。
- OSI开放系统互联安全体系的5类安全服务包括：鉴别、访问控制、数据机密性、数据完整性和抗抵赖性
- OSI定义分层多点安全技术体系架构，也称为深度防御安全技术体系架构，它通过以下三种方式将防御能力分布至整个信息系统中。
 - 多点技术防御：在对手可以从内部或外部多点攻击一个目标的前提下，多点技术防御通过对网络和基础设施、边界、计算环境这三个防御核心区域的防御达到抵御所有方式的攻击目的。
 - 分层技术防御：即使最好的可得到的信息保障产品也有弱点，其最终结果将使对手能找到一个可探查的脆弱性，一个有效的措施是在对手和目标间使用多个防御机制。
 - 支撑性基础设施：为网络、边界和计算环境中信息保障机制运行基础的支撑性基础设施，包括公钥基础设施以及检测和响应基础设施。

数据库系统的安全设计

- 数据库完整性是指数据库中**数据的正确性和相容性**。数据库完整性由各种各样的**完整性约束**来保证，因此可以说数据库完整性设计就是数据库完整性约束的设计。数据库完整性约束可以通过**DBMS**或应用程序来实现，基于**DBMS**的完整性约束作为模式的一部分存入数据库中。
- 在实施数据库完整性设计时，需要把握以下基本原则：
 - 根据数据库完整性约束的类型**确定其实现的系统层次和方式**，并提前考虑对系统性能的影响。一般情况下，静态约束应尽量包含在数据库模式中，而动态约束由应用程序实现。比如在购物车中，库存的数量是一个静态约束，应该包含在数据库模式中，以确保不会出现负库存。另一方面，用户的购物车中的商品数量则是一个动态约束，因为它根据用户的操作动态变化，应该由应用程序来实现。
 - 实体完整性约束、引用完整性约束是关系数据库最重要的完整性约束，在不影响系统关键性能的前提下需尽量应用。用一定的时间和空间来换取系统的易用性是值得的。比如在银行的数据库中，账户余额是一个非常重要的数据。应该应用实体完整性约束来确保账户余额不会出现负数，从而保护客户的资金
 - 要慎用目前主流**DBMS**都支持的触发器功能，一方面由于触发器的性能开销较大;另一方面，触发器的多级触发难以控制，容易发生错误，非用不可时，最好使用**Before**型语句级触发器。
 - 在需求分析阶段就必须制定完整性约束的命名规范，尽量使用有意义的英文单词、缩写词、表名、列名及下画线等组合，使其易于识别和记忆。
 - 要**根据业务规则对数据库完整性进行细致的测试**，以尽早排除隐含的完整性约束间的冲突和对性能的影响。
 - 要有**专职的数据库设计小组**，自始至终负责数据库的分析、设计、测试、实施及早期维护。
 - 应采用合适的**CASE**工具来降低数据库设计各阶段的工作量。

数据库系统的安全设计

■ 数据库完整性的作用：

- 数据库完整性约束能够防止合法用户使用数据库时向数据库中添加不合语义的数据。例如，不允许录入负分数或大于100分的分数。
- 利用基于DBMS的完整性控制机制来实现业务规则，易于定义，容易理解，而且可以降低应用程序的复杂性，提高应用程序的运行效率。比如在一个酒店预订系统中，数据库可以使用外键完整性约束，确保每个预订都关联到一个有效的客户。这有助于确保每个预订都与现有客户相关联，避免了无效的预订
- 合理的数据库完整性设计，能够同时兼顾数据库的完整性和系统的效能。
- 在应用软件的功能测试中，完善的数据库完整性有助于尽早发现应用软件的错误。
- 数据库完整性约束可分为6类：列级静态约束、元组级静态约束、关系级静态约束、列级动态约束、元组级动态约束和关系级动态约束。

■ 怎么进行数据库完整性设计：

- 首先需要在需求分析阶段确定要通过数据库完整性约束实现的业务规则。
- 然后在充分了解特定DBMS提供的完整性控制机制的基础上，依据整个系统的体系结构和性能要求，遵照数据库设计方法和应用软件设计方法，合理选择每个业务规则的实现方式。
- 最后，认真测试，排除隐含的约束冲突和性能问题。

系统架构的脆弱性分析

- 脆弱性分析：是分析信息系统中产生脆弱性的根源、脆弱性可能造成的影响、如何利用脆弱性进行攻击、如何修补脆弱性、如何防止脆弱性被利用、如何探测目标系统的脆弱性、如何预测新的脆弱性的存在等一系列问题。
- 从技术角度而言，漏洞的来源主要有以下几个方面：**软件设计时的瑕疵、软件实现中的弱点、软件本身的瑕疵、系统和网络的错误配置。**
- 软件脆弱性有其自身的特点，主要包括4个方面：
 - 脆弱性是软件系统中**隐藏的一个弱点**，本身不会引起危害，但被利用后会产生严重的安全后果；
 - 在软件开发过程中，**自觉或不自觉引入的逻辑错误**是大多数脆弱性的根本来源；
 - 与具体的系统环境密切相关，**系统环境的任何差异**都有可能产生不同的脆弱性问题；
 - 旧的脆弱性**得到修补或纠正的同时可能引入新的脆弱性**，因此脆弱性问题会长期存在。
- 软件脆弱性的生命周期：即每一种脆弱性都有其引入原因；在一种脆弱性引入之后，它会产生某种破坏效果，从而破坏系统的完整性或者可用性；针对已有的每一种脆弱性，人们可能会提出一些修补措施，在实施这些修补措施之后脆弱性将消失。

系统架构的脆弱性分析

脆弱性的**引入阶段**：引入软件脆弱性的原因有：

- 输入验证错误;
- 权限检查错误;
- 操作序列化错误;
- 边界检出错误;
- 软件设计时的缺陷;
- 其他错误。

产生破坏效果阶段：主要包括：

- 非法执行代码;
- 非法修改目标对象;
- 访问数据对象;
- 拒绝服务攻击。

修补阶段：主要包括：

- 删除伪造实体(如IP伪造、名字伪造等);
- 增加新的实体;
- 写该实体不正确的位置;
- 其他情况。

系统架构的脆弱性分析

■ 软件脆弱性分析可从三个方面考虑：

- 分析**软件故障现象**，分析故障的技术本质、总结脆弱性模式；
- 分析**软件开发**，发现安全管理和技术的薄弱环节，提高软件安全性；
- 分析**软件使用**，发现其脆弱性，采取相应措施，避免脆弱性转化为安全故障。

■ 软件脆弱性分析首先要明确分析对象，脆弱性分析对象可以分为两类：**脆弱性数据和软件系统**。由于软件本身具有自身的性质和特点，针对软件的脆弱性分析，我们也需要考虑软件本身的各种特点。主要考虑**软件结构和实现技术**两个方面。

■ 典型**软件架构的脆弱性**分析

1. **分层架构**的脆弱性主要表现在两个方面：

- **层间**的脆弱性。一旦某个底层发生错误，那么整个程序将会无法正常运行。
- **层间通信**的脆弱性。将系统隔离为多个相对独立的层，这就要求在层与层之间引入通信机制。本来“直来直去”的操作现在要层层传递，势必造成**性能下降**。

2. **C/S架构**的脆弱性主要表现在以下几个方面：

- **客户端软件**的脆弱性。因为在用户计算机上安装了客户端软件，所以这个系统就面临着程序被分析、数据被截取的安全隐患。
- **网络开放性**的脆弱性。目前很多传统的**C/S**系统还是采用二层结构，也就是说所有客户端直接读取服务器端中的数据，在客户端包括了数据的用户名，密码等致命的信息，这样会给系统带来安全隐患。
- **网络协议**的脆弱性。

系统架构的脆弱性分析

3.B/S 架构的脆弱性主要表现在：系统如果使用HTTP协议，B/S架构相对C/S架构而言更容易被病毒入侵，虽然最新的HTTP协议在安全性方面有所提升，但还是弱于C/S。

4.事件驱动架构的脆弱性主要表现在：

- 组件的脆弱性。组件削弱了自身对系统的控制能力，一个组件触发事件，并不能确定响应该事件的其他组件及各组建的执行顺序。
- 组件间交换数据的脆弱性。组件不能很好地解决数据交换问题，事件触发时，一个组件有可能需要将参数传递给另一个组件，而数据量很大的时候，如何有效传递是一个脆弱性问题。
- 组件间逻辑关系的脆弱性。事件架构使系统中各组件的逻辑关系变得更加复杂。
- 事件驱动容易进入死循环，这是由编程逻辑决定的。
- 高并发的脆弱性。虽然事件驱动可实现有效利用CPU资源，但是存在高并发事件处理造成的系统响应问题，而且，高并发容易导致系统数据不正确、丢失数据等现象。
- 固定流程的脆弱性。因为事件驱动的可响应流程基本都是固定的，如果操作不当，容易引发安全问题。

系统架构的脆弱性分析

5. MVC架构的脆弱性主要表现在:

- MVC架构的**复杂性**带来脆弱性。MVC架构增加了系统结构和实现的复杂性。比如说一个简单的界面，如果严格遵循MVC方式，使得模型、视图与控制器分离，会增加结构的复杂性，并可能产生过多的更新操作，降低运行效率。
- **视图与控制器间紧密连接**的脆弱性。视图与控制器是相互分离但确是联系紧密的部件，没有控制器的存在，视图应用是很有限的。反之亦然，这样就妨碍了它们的独立重用。
- **视图对模型数据的低效率访问**的脆弱性。依据模型操作接口的不同，视图可能需要多次调用才能获得足够的显示数据。对未变化数据的不必要的频繁访问也将损害操作性能。

6. 微内核架构的脆弱性主要表现在:

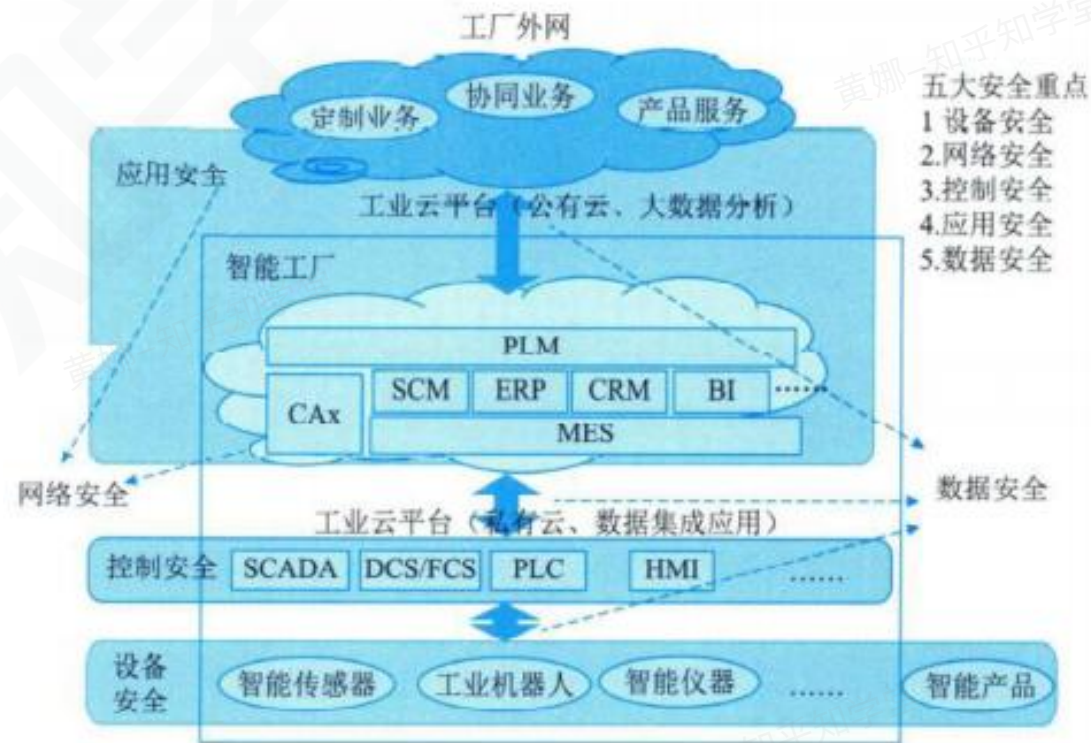
- 微内核架构**难以进行良好的整体化优化**。由于微内核系统的核心态只实现了最基本的系统操作，这样内核以外的外部程序之间的独立运行使得系统难以进行良好的整体优化。
- 微内核系统的进程间通信开销也较单一内核系统要大得多。从整体上看，在当前硬件条件下，微内核在效率上的损失小于其在结构上获得的收益。
- 通信损失率高。微内核把系统分为各个小的功能块，从而降低了设计难度，系统的维护与修改也容易，但通信带来的效率损失是一个问题。

7. 微服务架构的脆弱性主要表现在:

- 开发人员需要处理**分布式系统的复杂结构**。
- 开发人员要**设计服务之间的通信机制**，通过写代码来处理消息传递中速度过慢或者不可用等局部实效问题。
- **服务管理的复杂性**，在生产环境中要管理多个不同的服务实例，这意味着开发团队需要全局统筹。

安全架构设计案例分析

- 跨区域的安全生产管理是大型集团企业面临的主要生产问题。大型企业希望通过云计算平台实现异地的设计、生产、制造、管理和数据处理等，并确保企业内部生产的安全、保密和数据的完整。
- 混合云架构往往被大型企业所接受。混合云融合了公有云和私有云，是近年来云计算的主要模式和发展方向。我们知道私有云主要是面向企业用户，出于安全考虑，企业更愿意将数据存放在私有云中，但是同时又希望可以获得公有云的计算资源，在这种情况下混合云被越来越多地采用，它将公有云和私有云进行混合和匹配，以获得最佳的效果，这种个性化的解决方案，达到了既省钱又安全的目的。
- 下图给出了大型企业采用混合云技术的安全生产管理系统架构，企业由**多个跨区域的智能工厂和公司总部组成**，公司总部负责相关业务的管理、协调和统计分析，而**每个智能工厂负责智能产品的设计与生产制造**。智能工厂内部采用私有云实现产品设计、数据共享和生产集成等，公司总部与智能工厂间采用公有云实现智能工厂间、智能工厂与公司总部间的业务管理、协调和统计分析等。
- 整个安全生产管理系统架构由四层组成：
 - 设备层：主要是指用于智能工厂生产产品所需的相关设备；
 - 控制层：主要是指智能工厂生产产品所需要建立的一套自动控制系统，控制智能设备完成生产工作；
 - 设计/管理层：是指智能工厂各种开发、业务控制 and 数据管理功能的集合，实现数据集成与应用；
 - 应用层：主要是指在云计算平台上进行信息处理，有两个核心功能，一是“数据”，二是“应用”。



THE END

功不唐捐，玉汝于成！

• 开 启 新 征 程 •

