第一章

什么是零日（0 day）漏洞?什么是零日（0 day）攻击?

零日漏洞是指未被公开披露的软件漏洞，没有给软件的作者或厂商以时间去为漏洞打补丁或是给出建议解决方案，从而攻击者能够利用这种漏洞破坏计算机程序、数据及设备。

利用零日漏洞开发攻击工具进行的攻击称为零日攻击。

为什么说面对当前的全球网络空间安全威胁，必须对软件安全给予强烈关注?

1. 软件是程序、数据和文档的集合体。
2. 软件无处不在
3. 软件规模日益庞大
4. 软件漏洞普遍存在，零日漏洞成为主要安全威胁。
5. 软件安全成为国家的一项竞争优势。
6. 100%安全的软件和系统不存在，应该尽力减少系统漏洞数量，提高整体安全性。

当前，黑客为了能够有效达到窃取数据、破坏系统的目的，常常通过挖掘或是购买零日漏洞，开发针对零日漏洞的攻击工具，零日漏洞威胁实际上反映了软件系统存在的一个什么问题？

100%安全的软件和系统是不存在的，软件产品存在漏洞是当前信息安全领域而临的最大困境。从理论上说，系统越庞大，代码越复杂，虽然对应地能够实现的功能越多，但是安全隐患也会随之增加。Windows 系统、Office 应用组件等系统，包括操作系统以及应用软件系统，都有有可能存在零日漏洞，从而成为被攻击的对象。软件漏洞是普遍存在的。

根据本书的介绍，软件安全威胁可以分为哪几类?软件面临哪些威胁？

1. 软件自身的安全（软件漏洞）
2. 恶意代码
3. 软件侵权

试谈谈对软件漏洞的认识，举出软件漏洞造成危害的事件例子。

软件漏洞通常被认为是软件生命周期中与安全相关的设计错误、编码缺陷及运行故障等。一方面，软件漏洞可能会造成软件在运行过程中出现错误结果或运行不稳定、崩溃等现象，甚至引起死机等情况。另一方面，软件漏洞会被黑客发现、利用，进而实施窃取隐私信息、甚至破坏系统等攻击行为。

软件漏洞造成的威胁，比如我们可以利用 MS08-067 漏洞渗透 Windows XP 系统获取系统较为高级的权限

什么是恶意代码?除了传统的计算机病毒，还有哪些恶意代码类型?

恶意代码（Malicious Software，Malware）是在未被授权的情况下，以破坏软硬件设备、窃取用户信息、干扰用户正常使用、扰乱用户心理为目的而编制的软件或代码片段。

恶意代码包括：

- 计算机病毒（Computer Virus）
- 蠕虫（Worm）
- 特洛伊木马（Trojan Horse）
- 后门（Back Door）
- 内核套件（Rootkit）
- 间谍软件（Spyware）
- 恶意脚本（Malice Script）
- 恶意广告（Dishonest Adware）
- 流氓软件（Crimeware）
- 逻辑炸弹（Logic Bomb）
- 僵尸网络（Botnet）
- 网络钓鱼（Phishing）、垃圾信息（Spam）等恶意的或令人讨厌的软件及代码片段。近几年危害甚广的勒索软件（Ransomware）也属于恶意代码范畴。

署名 复制 修改 发表 发行 翻译 出租 网络传播

针对软件的版权，有哪些侵权行为?

软件侵权主要指侵犯版权（著作权），常见软件侵权行为：

- 未经软件著作权人许可，发表、登记、修改、翻译其软件；
- 将他人软件作为自己的软件发表或者登记，在他人软件上署名或者更改他人软件上的署名；
- 未经合作者许可，将与他人合作开发的软件作为自己单独完成的软件发表或者登记；
- 复制或者部分复制著作权人的软件；
- 向公众发行、出租、通过信息网络传播著作权人的软件；
- 故意避开或者破坏著作权人为保护其软件著作权而采取的技术措施；
- 故意删除或者改变软件权利管理电子信息；
- 转让或者许可他人行使著作权人的软件著作权。

谈谈对软件安全概念的理解。

是软件工程与软件保障的一个方面，它提供一种系统的方法来标识、分析和追踪对危害以及具有危害性的功能（例如数据和命令）的软件缓解措施与控制。

1）保密性

2）完整性

3）可用性

4）可认证性

5)可审计性

6）不可否认性

7）可控性

8）可存活性

简述软件和软件工程的概念。

国标中对软件的定义为：与计算机系统操作有关的计算机程序、规程、规则，以及可能有的文件、文档及数据。

软件工程（ Software Engineering） 是指，采用工程的概念、原理、技术和方法来开发和维护软件，将正确的管理技术+最好的技术方法结合起来，从而经济地开发出高质量的软件并有效地进行维护。

对照一般软件工程的概念，软件安全工程主要增添了哪些任务?

系统安全工程是一项复杂的系统工程，需要运用系统工程的思想和方法，系统地分析信息系统存在的安全漏洞、风险、事件、损失、控制方法及效果之间复杂的对应关系，对信息系统的安全性进行分析与评价，以期建立-一个有效的安全防御体系，而不是简单的安全产品堆砌。

确切地说，系统安全工程是系统的安全性问题而不仅是软件产品的安全性问题，是一种普适性的信息系统安全工程理论与实践方法，可以用于构建各种系统安全防御体系。系统安全工程可以在系统生命周期的不同阶段对安全问题提供指导，例如，对于已经发布运行的软件，可以采用系统测试、风险评估与控制等方法构建安全防御体系；而对于尚待开发的系统，也可以应用系统安全工程的思想方法来提高目标系统的安全性。

谈谈软件安全与软件危机、软件质量和软件质量保证、软件保障、软件可靠性、应用软件系统安全、可信软件和软件定义安全等概念的区别和联系。

软件是程序、数据、文档的集合体。软件安全就是使软件在受到恶意攻击的情形下依然能够继续正确运行及确保软件被在授权范围内合法使用的思想。

软件危机（Software Crisis），也称为软件萧条（Software Depression）或软件困扰（Soft-ware Afliction），是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。这些问题绝不仅仅是不能正常运行的软件才具有的，可以说几乎所有软件都不同程度地存在这些问题。

软件质量（SoftwareQuality）就是“软件与明确的和隐含的定义的需求相一致的程度”。具体地说，软件质量是软件符合明确叙述的功能和性能需求、文档中明确描述的开发标准，以及所有专业开发的软件都应具有的和隐含特征相一致的程度。

通常软件保障包括软件质量（软件质量工程、软件质量保障和软件质量控制等功能）、软件安全性、软件可靠性、软件验证与确认，以及独立验证与确认等学科领域。长期以来，软件可靠性（SoftwareReliability）作为衡量软件质量的唯一特性受到特别重视。应用软件系统位于信息系统的上层，是在信息系统的硬件系统、操作系统、网络系统和数据库管理系统的支持下运行的，是构成信息系统的最重要部分，是信息系统中直接为用户提供服务的部分。为了确保业务应用的安全，首要的是确保应用软件系统的安全。

“可信性”是在正确性、可靠性、安全性、时效性、完整性、可用性、可预测性、生存性及可控性等众多概念的基础上发展起来的一-个新概念，是客观对象的诸多属性在人们心目中的一个综合反映。目前的可信软件研究是在软件正确性、可靠性、安全性和生存性等基础上发展起来的，SDS

是适应 SDN 复杂网络的安全防护新思想，基本原理是将物理及虚拟的网络安全设备预期接人模式、部署方式和实现功能进行解耦，底层抽象为安全资源池里的资源，顶层统一通过软件编程的方式进行智能化、自动化的业务编排和管理，以完成相应的安全功能，从而实现一种灵活的安全防护。

确保软件安全的基本思路是什么?软件安全涉及的技术主要有哪些方面?

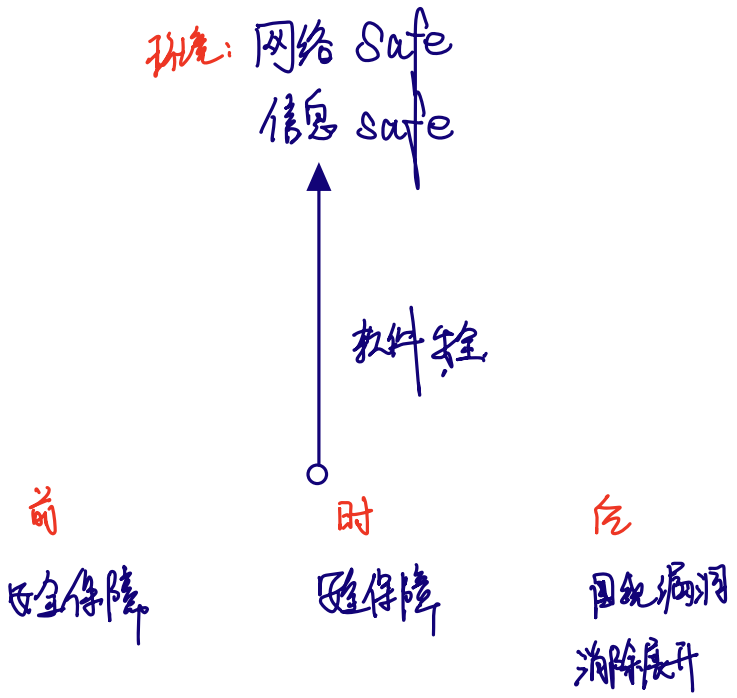
1、 网络空间安全、信息保障。

2、 软件安全是信息安全保障的重要内容。

3、 软件安全防护围绕漏洞消除展开。

4、 将安全保障措施置于软件发布运行之时。

5、 将安全保障的实施开始于软件发布之前，尤其强调从软件生命周期的早期阶段开始安全考虑。



缺陷 — 内部：开发和维护过程中存在的错误、毛病等各种问题
外部：对于某种功能的失效 or 违背功能

第二章

试述软件漏洞的概念，谈谈软件漏洞与软件错误、软件缺陷、软件 Bug 的区别与联系

环境 1、 漏洞是存在于评估对象中的、在一定的环境条件下可能违反安全功能要求的弱点。

自身 2、 漏洞是存在信息系统、系统安全过程、内部控制或实现过程中的可被威胁源攻击或者触发的弱点。

利用 3、 漏洞是能够被一个或多个威胁利用的资产或控制中的弱点。

小结，对漏洞认识的 3 个共同特点：

4、 漏洞是信息系统自身具有的弱点或者缺陷；

5、 漏洞存在环境通常是特定的；

6、 漏洞具有可利用性，若攻击者利用了这些漏洞将会给信息系统安全带来严重威胁和经济损失。

软件漏洞通常被认为是软件生命周期中与安全相关的设计错误、编码缺陷及运行故障等

为什么说安全缺陷或者说 Bug 是一个需要考虑具体环境、具体对象的概念?

需要说明的是，安全缺陷或者说 Bug 是-一个需 要考虑具体环境、具体对象的概念。举例来说，一般的 Web 应用程序没有使用 HTTPS 协议（超文本传输安全协议）来加密传输的状态并不能算作是 Bug，而对于网上银行或电子商务等应用，不采用 HTTPS 协议进行加密传输就应当算作一个 Bug。如同使用 HTTPS 来对传输内容进行加密那样，积极主动地加强安全性的措施，也就是增加安全性功能，可以尽可能地消除 Bug。安全性功能实际为软件系统的一种需求，所以也被称为安全性需求。是否将安全性功能加入到项目需求中，还需要根据项目的具体情况考虑，如项目经费等。

试分析软件漏洞的成因。为什么会出现软件漏洞

计算机系统结构决定了漏洞的必然性

软件趋向大型化，第三方拓展增多

新技术、新应用产生之初即缺乏安全性考虑

软件使用场景更具威胁

软件漏洞为什么要管理？如何管理？

漏洞是一种“武器”

让白帽的漏洞发现有章有法

漏洞管控势在必行

1）软件漏洞分类

通常可以从漏洞利用的成因、利用的位置、和对系统造成的直接威胁进行分类。

①基于漏洞成因的分类

内存破坏类、逻辑错误类、输入验证类、设计错误类和配置错误类。

②基于漏洞利用位置的分类

本地漏洞。

远程漏洞。

③基于威胁类型的分类

获取控制。获取信息。拒绝服务。

如何管理？

2）软件漏洞分级

①按照漏洞严重等级进行分级

②利用通用漏洞评分系统（CVSS）进行分级：基本度量、时间度量、环境度量

厂商发布漏洞信息的标准过程是怎样的？

1.设立部门

2.奖励白帽

3.社会面收集

第三章

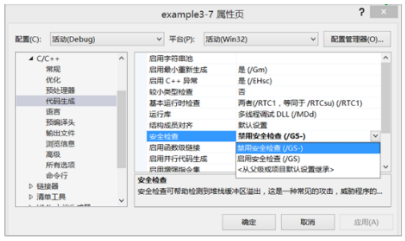
程序运行时的内存布局是怎样?

什么是缓冲区溢出漏洞?

缓冲区溢出漏洞就是在向缓冲区写入数据时，由于没有做边界检查，导致写人缓冲区的数据超过预先分配的边界，从而使溢出数据覆盖在合法数据上而引起系统异常的一种现象。

简述 Windows 安全漏洞保护的基本技术<---------->对抗技术

- **（1）栈溢出检测选项/GS**
- **调用函数时将一个随机生成的秘密值存放在栈上，当函数返回时，检查这个堆栈检测仪的值是否被修改，以此判断是否发生了栈溢出。**



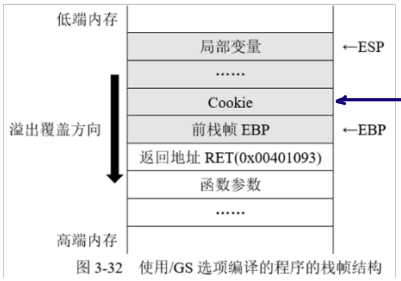


图 3-32 使用/GS 选项编译的程序的栈帧结构

- **（2）数据执行保护DEP（Data Execution Prevention）**
- **通过使可写内存不可执行或使可执行内存不可写来消除类似的威胁。**

- （3）地址空间布局随机化ASLR
- 通过对堆、栈、共享库映射等线性区域布局的随机化，增加攻击者预测目的地址的难度，防止攻击者直接定位攻击代码位置，达到阻止漏洞利用的目的。

- （4）安全结构化异常处理SafeSEH
- SafeSEH保护机制的作用是防止覆盖和使用存储栈上的SEH结构。

- （5）增强缓解体验工具包 EMET

本章介绍了 Windows 的 5 种典型保护机制，但是每一种保护机制仍然面临着缺陷和许多对抗的方法，这说明了什么问题?应当如何应对这一问题?

没有完美无缺的保护机制，需要加强防范意识，采取多种手段护

第四章

常用的 Web 三层架构是怎样的?

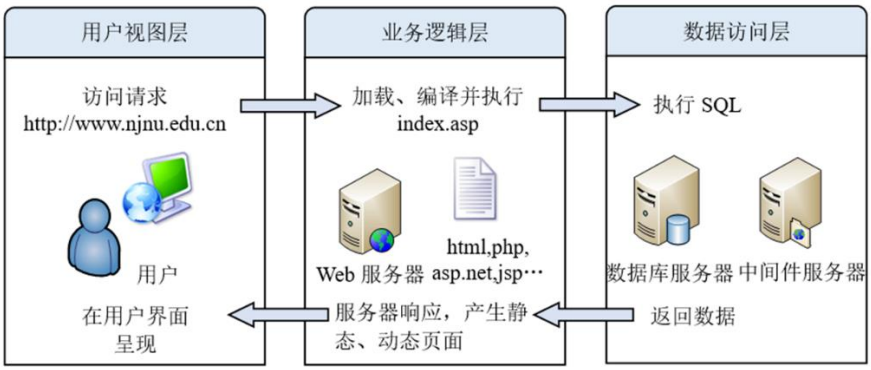


图 4-1　Web 三层通用架构

OWASP

2013年版《OWASP Top 10》	→	2017年版《OWASP Top 10》
A1 – 注入	→	A1:2017 – 注入
A2 – 失效的身份认证和会话管理	→	A2:2017 –失效的身份认证
A3 – 跨站脚本（XSS）	↘	A3:2017 –敏感信息泄漏
A4 – 不安全的直接对象引用 [与A7合并]	∪	A4:2017 – XML外部实体（XXE） [新]
A5 – 安全配置错误	↘	A5:2017 – 失效的访问控制 [合并]
A6 – 敏感信息泄漏	↗	A6:2017 – 安全配置错误
A7 – 功能级访问控制缺失 [与A4合并]	∪	A7:2017 – 跨站脚本（XSS）
A8 – 跨站请求伪造 （CSRF）	☒	A8:2017 – 不安全的反序列化 [新，来自于社区]
A9 – 使用含有已知漏洞的组件	→	A9:2017 –使用含有已知漏洞的组件
A10 – 未验证的重定向和转发	☒	A10:2017 – 不足的日志记录和监控 [新，来自于社区]

当在浏览器的地址栏中输人一个完整的 URL，再按 Enter 键直至页面加载完成，整个过程发生了什么?

1）域名解析：浏览器会依次查询浏览器的 DNS 缓存、系统缓存、路由器缓存，如果没有找到，则一直查询到根域名服务器缓存，找到域名所对应的的 IP 地址。

2）TCP 连接：通过 IP 地址找到 IP 对应的服务器后，要求建立 TCP 连接。

3）HTTP 连接：TCP 连接成功后，浏览器开始向这个服务器发送一个 HTTP 请求。服务器接收到请求后开始进行处理，处理结束，返回一个响应包。

4）浏览器接收和处理：浏览器接收到来自服务器的响应后，开始解析和渲染接收到的内容并呈现给用户。

5）TCP 断开连接：最后客户端断开与服务器的 TCP 连接。

试将 Web 典型漏洞根据客户端和服务器端来划分，并根据漏洞原理阐述这样划分的理由。

1， 客户端漏洞：

XSS（跨站脚本攻击）

CSRF（跨站请求伪造）

XXE（XML 外部实体注入）

2，服务端漏洞：

SQL 注入

文件上传漏洞

服务器请求伪造（SSRF）

反序列化

命令执行漏洞

文件包含漏洞

逻辑漏洞

越权漏洞

敏感信息泄露

简述 SQL 注人漏洞的原理?为什么 SQL 注人漏洞多年来一直名列 Web 安全漏洞的榜首

SQL 注入漏洞是指，攻击者能够利用现有 Web 应用程序，将恶意的数据插入 SQL 查询中，提交到后台数据库引擎执行非授权操作。SQL 注入攻击具有广泛性。相较于其他漏洞，对于 SQL 注入漏洞的防范要困难。

防范 SQL 注人漏洞的基本方法有哪些?重点谈谈在代码开发层面的安全措施。

1. 采用强类型语言，如 Java、 C#等强类型语言几乎可以完全忽略数字型注人。
2. 尽可能避免使用拼接的动态 SQL 语句，所有的查询语句都使用数据库提供的参数化查询接口。参数化的语句使用参数而不是将用户输人变量嵌人到 SQL 语句中。
3. 在服务器端验证用户输人的值和类型是否符合程序的预期要求。
4. 在服务器端对用户输人进行过滤。
5. 避免网站显示 SQL 错误信息

加固应用程序服务器和数据库，利用最低权限账户与数据库连接。

什么是 SQL 盲注?它与一般的 SQL 注人有什么区别?

盲注就是在 sql 注入过程中，sql 语句执行的选择后，选择的数据不能回显到前端页面。此时，我们需要利用一些方法进行判断或者尝试，这个过程称之为盲注。一般的注入攻击者可以直接从页面上看到注入语句的执行结果，而盲注时攻击者通常是无法从显示页面上获取执行结果，甚至连注入语句是否执行都无从得知

简述 XSS 跨站脚本漏洞的原理。

恶意攻击者在 Web 页面中插入恶意 javascript 代码（也可能包含 html 代码），当用户浏览网页之时，嵌入其中 Web 里面的 javascript 代码会被执行，从而达到恶意攻击用户的目的。

第五章

什么是软件的生命周期?软件生命周期通常包括哪几个阶段?

正如任何事物一样，软件也有其孕育、诞生、成长、成熟和衰亡的生存过程，一般称其为“软件生命周期”。软件生命周期由定义、开发和维护 3 个时期组成。共有问题定义、可行性研究、需求分析、总体设计、详细设计、编码、单元测试、综合测试以及维护八个阶段。

什么是软件过程?什么是软件开发（过程）模型?为什么从 20 世纪 90 年代以后，人们更多使用“软件过程”来替代传统的“软件开发模型”？

所谓软件过程，是指为了获得高质量软件所需要完成的一-系列任务的框架，它规定了完成各项任务的工作步骤。

以极限编程为代表的敏捷开发，具有对变化和不确定性的更快速、更敏捷的反应特性。软件过程更加适合当前的软件开发。

有哪些典型的软件开发模型?这些软件开发模型有什么区别与联系?

瀑布模型、快速原型模型、增量模型、螺旋模型、喷泉模型、Rational 统一过程、极限编程和敏捷开发、微软过程。瀑布模型是在 20 世纪 80 年代之前唯一被广泛采用的生命周期模型，是一个规范的、文档驱动的方法。快速原型模型是为了克服瀑布模型的缺点而提出来的。增量模型是把待开发的软件系统模块化，将每个模块作为一个增量组件，从而分批次地分析、设计、编码和测试这些增量组件。螺旋模型的基本做法是，在瀑布模型的每一个 开发阶段前引入非常严格的风险识别、风险分析和风险控制。它把软件项目分解成-一个个小项目，每个小项目都标识一个或多个主要风险，直到所有的主要风险因素都被确定。喷泉模型是一种以用户需求为动力，以对象为驱动的模型。

RUP 强调采用迭代和检查的方式来开发软件，整个项目开发过程由多个迭代过程组成。以极限编程为代表的敏捷开发，具有对变化和不确定性的更快速、更敏捷的反应特性。微软过程是 RUP 的精简配置版本，也是敏捷过程的一个扩充版本。

这些软件开发模型保障了用户需求和软件系统功能、性能的实现，但是从软件安全开发生命周期的角度来看，上述软件开发模型的安全性并没有得到系统、完整的重视和体现。

SD3+C 原则是 SDL 模型实施的基本原则，试简述其内容。

安全设计（ Secure by Design）。 在架构设计和实现软件时，需要考虑保护其自身及其存储和处理的信息，并能抵御攻击。

安全配置（ Secure by Default）。在现实世界中，软件达不到绝对安全，所以设计者应假定其存在安全缺陷。为了使攻击者针对这些缺陷发起攻击时造成的损失最小，软件在默认状态下应具有较高的安全性。例如，软件应在最低的所需权限下运行，非广泛需要的服务和功能在默认情况下应被禁用或仅可由少数用户访问。

安全部署（ Security by Deployment）。软件需要提供相应的文档和工具，以帮助最终用户或管理员安全地使用。此外，更新应该易于部署。

沟通（Communication）。 软件开发人员应为产品漏洞的发现准备响应方案，并与系统应用的各类人员不断沟通，以帮助他们采取保护措施（如打补丁或部署变通办法）。

什么是敏捷 SDL?敏捷 SDL 和经典 SDL 的主要区别是什么?

能够快速利用敏捷开发流程更好地实现安全需求区：

敏捷 SDL 不采用传统的瀑布模型，而是采用无阶段的迭代开发模型，以实现软件版本的快速更新和发布。如果开发团队采用瀑布式开发流程，那么更适合采用典型的 SDL 模型，而并不适合敏捷 SDL。在敏捷 SDL 中，并不是每个发布版本都需要达到所有的要求，这也是敏捷 SDL 与传统 SDL 之间最大的差别。

第六章安全需求分析

为什么要进行需求分析?通常对软件系统有哪些需求?

需求分析的任务不是确定系统怎样完成它的工作，而仅仅是确定系统必须完成哪些工作，也就是生成目标系统完整、准确、清晰、具体的要求的过程。它的基本任务是准确地描述“系统必须做什么”这个问题。

- 性能需求：指定系统必须满足的定时约束或容量约束，通常包括速度（响应时间）、信息量速率、主存容量、磁盘容量和安全性等方面的需求。
- 可靠性和可用性需求：可靠性需求定量地指定系统的可靠性。可用性与可靠性密切相关，它量化了用户可以使用系统的程度。
- 出错处理需求：说明系统对环境错误应该怎样响应。
- 接口需求：描述应用系统与它的环境通信的格式。
- 约束：描述在设计或实现应用系统时应遵守的限制条件。
- 逆向需求：说明软件系统不应该做什么。理论上有无限多个逆向需求，人们应该仅选取能澄清真实需求且可消除可能发生的误解的那些逆向需求。

将来可能提出的要求：明确地列出那些虽然不属于当前系统开发范畴，但是据分析将来很可能会提出来的要求。这样做的目的是，在设计过程中对系统将来可能的扩充和修改预做准备，以便一旦确实需要时能比较容易地进行扩充和修改。

为什么要进行安全需求分析?通常对软件系统有哪些安全需求?

软件安全需求分析的目的是描述为了实现信息安全目标，软件系统应该做什么，才能有效地提高软件产品的安全质量，减少进而消减软件安全漏洞。

1. 外部安全需求

法律、法规等遵从性需求

2. 内部安全需求

一是组织内部需要遵守的政策、标准、指南和实践模式，二是与软件业务功能相关的安全需求。

软件安全需求分析的主要工作是什么?它和软件需求分析有什么区别与联系?

描述为了实现信息安全目标，软件系统应该做什么区别：安全需求并不是从使用者的要求和兴趣出发，而是由系统的客观属性所决定的。因此，需求分析员将承担更多软件需求的分析工作。

软件安全需求分析不能只从软件本身出发，必须从系统角度进行分析。

软件安全的需求内容非常丰富，并不是所有的应用安全需求控制都要采纳和实施。

联系：

软件安全需求是软件需求的一个必要 组成部分。安全需求应该与业务功能需求具有同样的需求水平，并对业务功能需求具有约束力。

为什么说软件安全需求更多地来源于遵从性需求?

软件需求分析中，分析员和用户都起着至关重要的作用。然而，在软件安全性需求分析中，软件用户由于安全知识的缺乏，很难从专业角度提出安全需求。因此，软件安全需求更多地来自于对组织内部和外部的一些安全政策和标准的遵从。安全需求分析人员对这些政策需求和标准进行深入理解，并将它们转化为软件安全属性需求，是安全需求分析阶段要完成的艰巨任务。

安全需求遵从性标准有哪些类别，它们之间有何联系与区别?

信息安全标准从适用地域范围可以分为：国际标准、国家标准、地方标准、区域标准、行业标准和企业标准。

信息安全标准从涉及的内容可以分为：信息安全体系标准、信息安全机制标准、信息安全测评标准、信息安全管理标准、信息安全工程标准、信息系统等级保护标准和信息安全产品标准等类别。

我国为什么要实行网络安全等级保护制度?网络安全保护能力划分为哪些等级?具体每个等级有什么要求?

对信息安全分级保护是客观需求。信息系统的建立是为社会发展、社会生活的需要而设计、建立的，是社会构成、行政组织体系及其业务体系的反映，这种体系是分层次和分级别的。因此，信息安全保护必须符合客观存在。

等级化保护是信息安全发展规律。按组织业务应用区域，分层、分类、分级进行保护和管理，分阶段推进等级保护制度建设，这是做好国家信息安全保护必须遵循的客观规律。

第一级，属于一般网络，其-一旦受到破坏，会对公民、法人和其他组织的合法权益造成损害，但不危害国家安全、社会秩序和社会公共利益。

第二级，属于一般网络，其一旦受到破坏，会对公民、法人和其他组织的合法权益造成严重损害，或者对社会秩序和社会公共利益造成危害，但不危害国家安全。

第三级，属于重要网络，其一旦受到破坏，会对公民、法人和其他组织的合法权益造成特别严重损害，或者会对社会秩序和社会公共利益造成严重危害，或者对国家安全造成危害。

第四级，属于特别重要网络，其一旦受到破坏，会对社会秩序和社会公共利益造成特别严重危害，或者对国家安全造成严重危害。

第五级，属于极其重要网络，其--旦受到破坏，会对国家安全造成特别严重危害。

简述网络等级保护与信息安全管理体系的联系和区别。

信息安全管理体系是站在管理的角度上对信息进行管理，而等级保护则是管理体系中的一部分，是基础性的工作，两者在管理目标上具有一致性，而且还有相辅相成的作用。信息安全管理体系和等级保护的工作重点不同。

信息安全管理体系和等级保护所依据的标准不同。

信息安全管理体系和等级保护的实施对象不同。

软件安全需求获取过程中涉及哪些相关方人员?他们各自主要的职责是什么?

软件安全需求获取的相关方包括业务负责人、最终用户、客户、安全需求分析人员和安全技术支持等。

业务负责人、最终用户和客户在安全需求确定时应发挥重要作用，他们应当积极参与安全需求的采集和分析过程。业务负责人是业务风险的最终责任人，负责确定可接受的风险阈值，明确哪些残余风险是可以接受的，因此他们应该了解软件的安全漏洞，协助安全需求分析人员和软件开发团队考虑风险的优先顺序，权衡决定哪些风险是重要的。

安全需求分析人员要负责软件安全需求的收集和分析，并帮助软件开发团队将安全需求转化为功能说明。

运维小组和信息安全小组等安全技术支持也是软件安全需求获取相关方，安全需求分析人员、业务负责人、最终用户及客户应当积极与之保持联系和沟通，寻求他们的支持和帮助。

软件安全需求的获取方法有哪些?

一些最常见的安全需求获取方法包括头脑风暴、问卷调查和访谈、策略分解、数据分类、主/客体关系矩阵、使用用例和滥用案例建模，以及软件安全需求跟踪矩阵。

软件安全需求的获取方法中的策略分解是指什么?

社文策略分解是指将组织需要遵守的内部和外部政策，包括外部法律法规、隐私和遵从性命令分解成详细的安全需求。策略分解过程是一个连续的、结构化的过程。

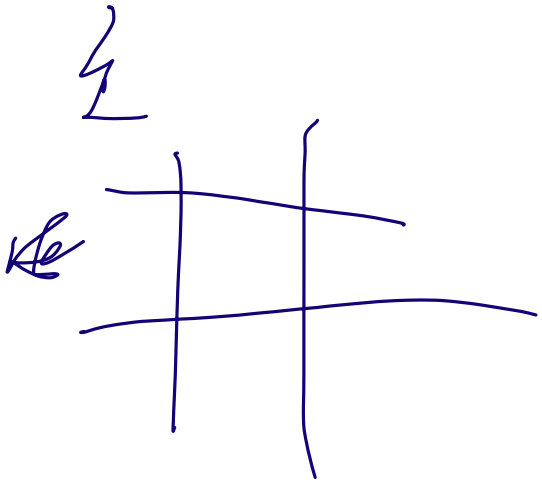
软件安全需求的获取方法中的数据分类是指什么?

数据分类是指，根据数据生命周期管理（ Data Lifecycle Management， DLM）对数据的分阶段划分来决定相应的安全需求；也可以根据数据的重要性对保护级别的划分来决定相应的安全需求。

针对信息系统中的数据生命周期，通常应当考虑的安全需求有哪些?

授权的级别、数据泄露保护、安全协议保护数据、数据存储时面临哪些安全威胁、如何应对频繁访问的关键数据存储介质的可靠性易失性问题、当数据归档时，需要遵循企业的数据保留政策，或是当地法律法规对数据存档的要求软件安全需求的获取方法中的主/客体关系矩阵是指什么?

采用主/客体关系矩阵来刻画一个基于使用用例的主/客体之间的操作关系。主/客体关系矩阵是角色和组件的二维表示，主体（角色）作为列，客体（对象/组件）作为行。当主/客体关系矩阵产生后，与主/客体关系矩阵所允许的对应动作相违背的事件就可以判定为威胁，在此基础之上可以确定安全需求。



第七章安全设计

软件设计阶段的主要工作是什么?

从生命周期的角度，软件设计可以看作是从软件需求规格说明书出发，根据需求分析阶段确定的功能，设计软件系统的整体结构、划分功能模块、确定每个模块的实现算法等内容，形成软件的具体设计方案，即从整体到局部，从总体设计（也称为概要设计）到详细设计的过程。

软件安全设计阶段的主要工作是什么?

将安全属性设计到软件架构中，以实现软件产品本质的安全性。

软件架构安全性设计的主要工作是什么?

软件架构安全设计首先需要进行系统描述，包括系统功能、安全要求、系统部署和技术需求，确定软件系统的安全级别。接着，设计软件网络、数据库等应具备的安全功能，根据软件具体安全需求的不同，设计的安全功能包括加密、完整性验证、数字签名、访问控制及安全管理等。在架构安全设计过程中，还需要解决软件安全功能的易用性、可维护性和独立性问题。

为什么要进行软件架构安全性分析?软件架构安全性分析的基本过程是什么?

为此，一旦软件架构设计或是软件架构安全性设计完成，在退出设计阶段进入开发阶段之前，需要对软件（安全）架构和设计方案进行检查，以确保设计能够满足软件的安全需求。这不仅包括功能方面的设计检查，也包括安全设计检查。检查可以帮助开发人员在编码之前对安全设计要素进行验证，提供一个识别和处理任何安全漏洞的机会，减少后续阶段重新设计软件的需要。

首先进行架构建模，然后根据软件的安全需求描述或相关标准，对架构模型是否满足要求进行检查，如果不满足则需要修改设计架构，如此反复，直至满足所有安全需求和相关标准。

软件受攻击面是指什么?举例说明软件设计时可以采取哪些策略来降低受攻击面。

软件受攻击面是指，用户或其他程序及潜在的攻击者都能够访问到的所有功能和代码的总和，它是一个混合体，不仅包括代码、接口和服务，也包括对所有用户提供服务的协议，尤其是那些未被验证的或远程用户都可以访问到的协议。一个软件的攻击面越大，安全风险就越大。减少软件受攻击面就是去除、禁止一切不需要使用的模块、协议和服务，其目的是减少攻击可以利用的漏洞。IOS 不支持 java 和 flash。IOS 不能处理.psd 文件，.mov 格式的文件也只被 IOS 部分支持。

什么是最小授权原则?试举例说明软件设计时哪些措施是采用了最小授权原则。

最小授权原则是指，系统仅授予实体（用户、管理员、进程、应用和系统等）完成规定任务所必需的最小权限，并且该权限的持续时间也尽可能短。最小授权原则可使无意识的、不需要的、不正确的特权使用的可能性降到最低，从而确保系统安全。

将超级用户的权限划分为一组细粒度的权限，分别授予不同的系统操作员/管理员。对管理员账户分配安全资源的访问权限也要设置为受限访问，而不是超级用户权限。

采用高内聚、低耦合的模块化编程方法，也就是模块之间的依赖关系是弱链接（低耦合），每一个模块只负责执行一个独立的功能（高内聚）。

试举例说明软件设计时哪些措施是采用了权限分离原则。

权限分离原则在软件设计中是指，将软件功能设计为需要在两个或更多条件下才能实现，以防止一旦出现问题，整个软件都可能面临风险。实际上这一原则也是最小权限原则的一种体现。

清晰的模块划分，将风险分散到各个模块中去。

不允许程序员检查自己编写的代码。

针对第 6 章介绍的核心安全需求，软件安全功能设计通常有哪些内容?

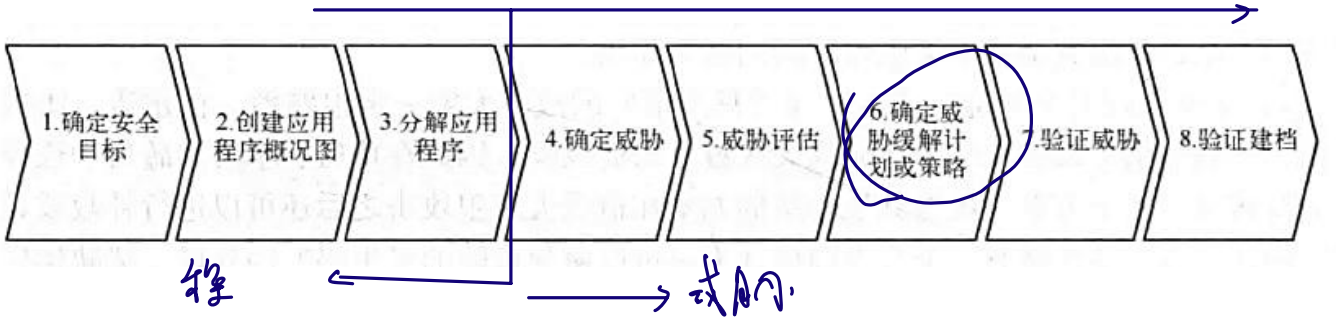
具体包括保密性、完整性、可用性、认证性、授权和可记账性等核心安全需求的设计，以及其他相关安全需求设计。

什么是安全模式?为什么说能够利用安全模式来快速、准确地进行软件安全设计?

安全模式是在给定的场景中，为控制、阻止或消减一组特定的威胁而采取的通用解决方案。安全模式封装了反复出现的系统问题的解决方案，同时精确地表述了系统要求和解决方案。采用模式的系统架构描述比较容易让人看懂，也为设计和分析提供了指南，还定义了使架构更安全的方法。安全模式使得不具备专业安全知识的应用开发人员也可以使用安全措施。还可以通过分析现有系统看它们是否包含特定的模式，进而评估它们的安全性。此外，可以在改造旧有系统时，利用模式来添加系统中缺失的安全特性。安全模式与威胁直接相关，特定的威胁可能是由一个或多个漏洞引起的。在软件设计阶段应用安全模式来控制、阻止或削弱威胁，可以从根本上消减软件安全漏洞。

什么是威胁建模?试简述威胁建模的过程。

软件威胁建模是指，通过抽象的概念模型对影响软件系统的威胁进行系统的识别和评价。



为什么说组织自身的威胁建模能力水平对提升组织的整体安全保障能力起到至关重要的作用?

一个完整的威胁模型是设计、开发、测试、部署和运营团队的代表性输入项。在设计阶段，应该由软件架构团队来识别威胁进而建立威胁模型；开发团队可以使用威胁模型来实现安全控制和编写安全的代码；测试人员不仅可以使用威胁模型生成安全测试用例，还需要验证威胁模型中已识别威胁的控制措施的有效性；最后，操作人员可以使用威胁模型配置软件安全设置，保证所有人口点和出口点都有必要的保护控制措施。

在威胁排序的几种计算方法中，为什么说相比 Delphi 法和平均排序法，PxI 排序方法更科学?

相比 Delphi 法和平均排序法，PxI 排序方法更科学。PxI 排序考虑业务影响（潜在损失和受影响的用户）和发生概率（可再现性、可利用性和可发现性）。PxI 排序法对事件发生概率、业务影响及它们合并的影响进行深入分析，使得设计团队能够灵活地掌握如何降低事件发生的概率、减小业务影响或二者同时降低；此外，PxI 排序方法还给出了更精确的风险图谱。

第八章安全编码

软件安全编码阶段的主要工作有哪些?

选择安全的编程语言、版本管理、代码检测、安全编译、代码分析、代码评审

什么是类型安全语言?哪些程序开发语言是类型安全的?

可以确保操作仅能应用于适当的类型，使程序员能够制定新的抽象类型和签名，防止没有经过授权的代码对特定的值实施操作。C#、java 强类型语言 Java和C#

安全编译是指在代码编译阶段采取的哪些安全措施?

1. 采用最新的集成编译环境，并选择使用这些编译环境提供的安全编译选项和安全编译机制来保护软件代码的安全性。
2. 代码编译需要在一个安全的环境中进行。
3. 对应用环境的真实模拟

试列举几条安全编码原则，并举例说明这些原则的重要意义。

1. 验证输入
2. 留意编译器警告
3. 安全策略的架构和设计保持简单性默认拒绝
4. 坚持最小权限原则
5. 清洁发送给其他系统的数据纵深防御
6. 使用有效的质量保证技术采用安全编码标准

为什么要避免使用 C 语言中原有的字符串函数?所谓的安全字符串函数解决了原有 C 字符串函数的什么安全漏洞?

不完善的字符串管理容易造成缓冲区溢出漏洞，缓冲区溢出

Java 提供的沙箱安全机制的核心思想是什么?

型的核心思想是：本地环境中的代码能够访问系统中的关键资源（ 如文件系统等），而从远程下载的程序则只能访问“沙箱”内的有限资源。

试谈谈 Java 提供的安全机制。

- **语言层安全**

通过某些关键字（如 private、 protected）定义代码的可见性范围

通过类型规则确保程序运行时变量的值始终与声明的类型一致，在函数或方法调用时形参与实参的类型匹配。

- **字节码层安全使用类加载器**

使用字节码验证器应用层安全

安全管理器用于说明一个安全策略并实施这个安全策略，描述了哪些代码允许做哪些操作。

第 12 章 p353

试解释以下与恶意代码程序相关的计算机系统概念，以及各概念之间的联系与区别:进程、线程、动态链接库、服务、注册表。

线程是执行任务，完成功能的基本单位，而进程则为线程提供了生存空间和线程所需要的其他资源，程序则是包含资源分配管理代码以及线程执行调度代码的一个静态计算机代码集合

动态链接库：动态链接库提供了一种方法，使进程可以调用不属于其可执行代码的函数。函数的可执行代码位于一个 DLL 中，该 DLL 包含一个或多个已被编译、链接并与使用它们的进程分开存储的函数。

服务：windows 系统的许多功能都是通过服务来实现的。简单来讲可以将服务理解为在后台完成系统任务的程序

注册表：注册表指在 Windows 中使用的中央分层数据库，用于存储一个或多个用户、应用程序和硬件设备配置系统所必须的信息。

从危害、传播、激活和隐藏 4 个主要方面分析计算机病毒、蠕虫、木马、后门、Rootkit 及勒索软件这几类恶意代码类型的工作原理。

破坏性。这是计算机病毒的本质属性，病毒侵人系统的目的就是要破坏系统的机密性、完整性和可用性等。计算机病毒编制者的目的和所人侵系统的环境决定了破坏程度，较轻者可能只是显示一些无聊的画面文字、发出点声音，稍重一点的可能是消耗系统资源，严重者可以窃取或损坏用户数据，甚至是瘫痪系统、毁坏硬件。

传染性。计算机病毒可以通过 U 盘等移动存储设备及网络扩散到未被感染的计算机。一旦进入计算机并得以执行，它就会搜寻符合其传染条件的程序，将自身代码插人其中，达到自我繁殖的目的。

潜伏性。大部分计算机病毒在感染系统或软件后不会马上发作，可以长时间潜伏在系统中，只在条件满足时才被激活，启动病毒的破坏功能。

隐藏性。计算机病毒不是用户所希望执行的程序，因此病毒程序为了隐藏自己，一般不独立存在(计算机病毒本原除外)，而是寄生在别的有用的程序或文档之上。同时，计算机病毒还采取隐藏窗口、隐藏进程、隐藏文件，以及远程

DLL 注入、远程代码注人和远程进程(线程)注人等方式来隐藏执行。

病毒程序与蠕虫程序的主要区别有哪些?

病毒的传播需要人为干预，而蠕虫则无需用户干预而自动传播。传统计算机病毒主要感染计算机的文件系统，而蠕虫影响的主要是计算机系统和网络性能。

什么是 Rootkit?它与木马和后门有什么区别与联系?

Rootkit 与木马、后门等既有联系又有区别。首先，Rootkit 属于木马的范畴，它用恶意的版本替换修改现有操作系统软件来伪装自己，从而掩盖其真实的恶意的目的，而这种伪装和隐藏机制正是木马的特性。此外，Rootkit 还作为后门行使其职能，各种 Rootkit 通过后门口令、远程 Shell 或其他可能的后门途径，为攻击者提供绕过检查机制的后门访问通道，这是后门工具的又一特性。Rootkit 强调的是强大的隐藏功能、伪造和欺骗功能，而木马、后门强调的是窃取功能、远程侵人功能。两者的侧重点不一样，两者结合起来则可以使得攻击者的攻击手段更加隐蔽强大。

什么是勒索软件?为什么勒索软件成为近年来数量增长最快的恶意代码类型?

勒索软件是黑客用来劫持用户资产或资源，并以此为条件向用户勒索钱财的一种恶意软件。

1. 加密手段有效，解密成本高。
2. 使用电子货币支付赎金，变现快，追踪困难。
3. 勒索软件即服务的出现，降低了攻击的技术门槛。

恶意代码防范的基本措施包括哪些?

1. 增强法律意识，2. 自觉履行恶意代码防治责任健全管理制度，3. 严格执行恶意代码防治规定

第 14 章

我国对于软件的知识产权有哪些法律保护途径?

《计算机软件保护条例》

《中华人民共和国专利法》商业秘密所有权保护

《中华人民共和国商标法》

《互联网著作权行政保护办法》

《信息网络传播权保护条例》

《移动互联网应用程序信息服务管理规定》

根据我国法律，软件著作权人有哪些权利?在日常学习和生活中，有哪些违反软件著作权的行为?

1.署名权、2.修改权、3.复制权 4.发表权(决定软件是否公之于众的权利)5.出租权、6发行权、7.信息网络传播权 8.翻译权等人身权和财产权。

未经软件著作权人许可，发表或者登记其软件的将他人软件作为自己的软件发表或者登记的未经合作者许可，将与他人合作开发的软件作为自己单独完成的软件发表或者登记的在他人软件上署名或者更改他人软件上的署名的未经软件著作权人许可，修改、翻译其软件的

试述软件版权的概念。针对软件的版权，有哪些侵权行为?有哪些保护措施?

软件版权保护旨在保护某个特定的计算机程序，以及程序中所包含信息的完整性、机密性和可用性。

1.软件盗版 2.逆向工程 3.信息泄露保护信息网络传播权。

保护为保护权利人信息网络传播权采取的技术措施。

保护用来说明作品权利归属或者使用条件的权利管理电子信息。

建立处理侵权纠纷的“通知与删除”简便程序。

软件版权保护的目标有哪些?它与软件保护的目标有什么联系与区别?

防软件盗版，即对软件进行防非法复制和使用的保护。

防逆向工程，即防止软件被非法修改或剽窃软件设计思想等。

防信息泄露，即对软件载体及涉及数据的保护，如加密硬件、加密算法的密钥等。软件版权保护的目标是软件保护目标的一个子集。软件保护除了确保软件版权不受侵害以外，还要防范针对软件的恶意代码感染、渗透、篡改和执行等侵害。

软件版权保护的许多措施同样可以应用于软件保护。