复习

> 课后习题，概念类的

第一章p32

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?

软件已经渗透到社会、经济与国防建设的方方面面，是信息时代所依赖的重要技术与手段，其安全直接关系到国计民生与国家安全，因此，软件安全关乎到国家竞争力。

安全漏洞是软件产生安全问题的根源，漏洞发现是软件安全的基础工作，软件安全体系的建立是以漏洞为核心展开的，对漏洞的掌控能力是衡量一个国家信息安全水平的重要因素。

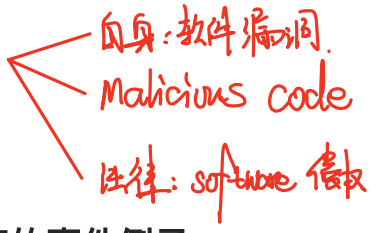
100%安全的软件和系统是不存在的，软件产品存在漏洞是当前信息安全领域而临的最大困境。

由于漏洞的产生、利用以及相互作用的机理复杂，因此，如何有效减少系统漏洞数量，提高信息系统整体安全性成为当前亟待解决的挑战性问题。

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?



软件自身的安全（软件漏洞）、恶意代码以及软件侵权

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。震网病毒便是利用软件漏洞实施的攻击，通过U盘和局域网进行传播，震网病毒对伊朗等国家的核设施造成的危害不亚于1986年发生的切尔诺贝利核电站事故，最终造成伊朗核计划拖后了2年，我国近500万网民及多个行业的领军企业也遭受了此病毒的攻击。

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）、恶意广告（ Dis-honest Adware）、流氓软件（ Crimeware）、逻辑炸弹（Logic Bomb）、僵尸网络（Botnet）、 网络钓鱼（Phishing）、恶意脚本（ Malice Script）及垃圾信息（ 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。

针对软件的版权，有哪些侵权行为?

- 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。

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。软件安全还包括保密性、完整性、可用性、可认证性、授权、可审计性、抗抵赖性、可控性和可存活性等多种安全属性。

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，把经过时间考验而证明正确的管理技术和当前能够得到的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。概括地说，软件工程是指导计算机软件开发和维护的一门工程学科，是技术与管理紧密结合形成的工程学科。

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。概括地说，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?

软件安全开发关注的是如何运用系统安全工程的思想，以软件的安全性为核心，将安全要素嵌人软件开发生命周期的全过程，有效减少软件产品潜在的漏洞数量或控制在一个风险可接受的水平内，提高软件系统的整体安全性。

软件安全开发方法抛弃了传统的先构建系统，再将安全手段应用于系统的构建模式，而是保留了采用风险管理、身份认证、访问控制、数据加密保护和人侵检测等传统安全方法，将安全作为功能需求的必要组成部分，在系统开发的需求阶段就引入安全要素，同时对软件开发全过程的每一个阶段实施风险管理，以期减少每一个开发步骤中可能出现的安全问题，最终提高软件产品的本质安全性。

第二章p53

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

漏洞是信息系统自身具有的弱点或者缺陷。漏洞存在环境通常是特定的。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。软件错误是软件开发生命周期各阶段中错误的真实体现。软件缺陷也称软件Bug，是指计算机软件或程序中存在的某种破坏正常运行能力的问题、错误，或者隐藏的功能缺陷。缺陷的存在会导致软件产品在某种程度上不能满足用户的需要。

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。

计算机系统结构决定了漏洞的必然性

软件趋向大型化，第三方扩展增多

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

对软件安全开发重视不够，软件开发者缺乏安全知识

软件漏洞如何分类分级管理?

基于漏洞成因的分类包括：内存破坏类、逻辑错误类、输人验证类、设计错误类和配置错误类。

基于漏洞利用位置的分类包括：本地漏洞和远程漏洞

基于威胁类型的分类包括：获取控制、获取信息和拒绝服务

按照漏洞严重等级进行分级

利用通用漏洞评分系统（CVSS）进行分级

软件漏洞管理应当遵循怎样的标准?

通用漏洞评分系统CVSS

通用漏洞和披露（ Common Vulnerabilities and Exposures， CVE）

通用缺陷枚举/评分系统

通用平台枚举

开放漏洞评估语言

软件漏洞买卖合法吗?软件漏洞应当如何管控?

不合法。国家通过政策法规和专业机构，形成一套管控体系。厂商成立安全应急响应部门（SRC），向社会收录旗下相关产品及业务的安全漏洞和威胁信息，并在第一时间进行处置，及时消除安全隐患。

厂商发布漏洞信息的标准过程是怎样的?

为了应对日益增加的漏洞，增加自身产品和服务的安全性，许多厂商纷纷成立安全应急响应部门（SRC），向社会收录旗下相关产品及业务的安全漏洞和威胁信息，并在第一时间进行处置，及时消除安全隐患。各厂商应急响应部门的迅速建立和发展，打通了厂商与“白帽”之间的正规渠道，相应的奖励也使得更多的“白帽”关注并协助厂商发现漏洞与风险，很大程度上提高了厂商的信息安全程度。

第三章p85

程序运行时的内存布局是怎样的?

在程序运行时，用来动态申请分配数据和对象的内存区域形式称为什么?

堆区

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人数据时，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及其存在的问题。

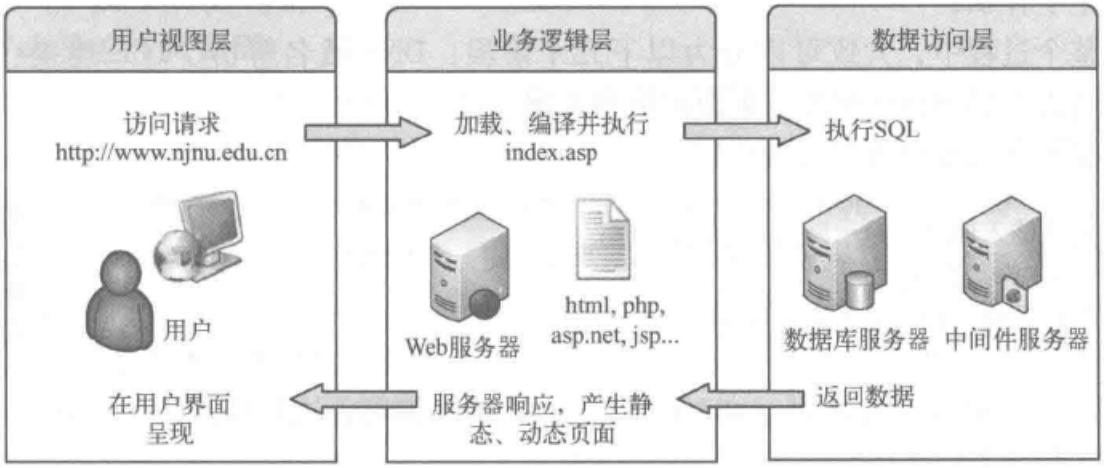
- /GS
    - 猜测cookie值
    - 通过同时替换栈中的Cookie和Cookie副本
    - 覆盖SEH绕过Cookie检查
    - 覆盖父函数的栈数据绕过Cookie检查
- DEP
    - 利用ret-to-libc执行命令或进行API调用，如调用WinExec实现执行程序。
    - 将包含Shellcode的内存页面标记为可执行，然后再跳过去执行。
    - 通过分配可执行内存，再将Shellcode复制到内存区域，然后跳过去执行。
    - 先尝试关闭当前进程的DEP保护，然后再运行Shellcode。
- ASLR
    - 对本地攻击者无能为力
    - 造成内存碎片的增多
    - 利用没有采用/DYNAMICBASE选项保护的模块作跳板
- SafeSEH
    - 利用未启用SafeSEH的模块作为跳板进行绕过
    - 利用加载模块之外的地址进行绕过
- 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，采取多种手段保护

第四章p121

常用的Web三层架构是怎样的?



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?

在这整个过程中，大致可以分为以下几个阶段： DNS 域名解析、TCP连接、HTTP请求、处理请求返回HTTP响应、页面渲染和关闭连接。

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。

客户端：

服务端：

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
6. 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

区别：

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p170

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。

- 功能需求：划分出系统必须完成的所有功能。
- 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。

外部安全需求

法律、法规等遵从性需求

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人理解，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p203

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。

为什么要进行软件架构设计?软件架构设计的主要工作是什么?软件架构安全性设计的主要工作是什么?

为了达到控制软件复杂性、提高软件系统质量、支持软件开发和复用的目的，开发人员提出了软件架构的概念。软件架构设计对于开发高质量软件具有较大作用。一般而言，软件架构的设计首先需要理清业务逻辑的功能要求，了解业务逻辑的变化性要求，包括可维护性和可扩展性，分离出概要业务逻辑层。接着，设计业务逻辑层和系统其他部分的接口与交互关系，按照职责分离原则设计包、类、方法和消息，设计业务逻辑算法。然后，使用自底向上和自顶向下相结合的方式，不断渐进地迭代架构设计。

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）。

什么是权限分离原则?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。

什么是软件设计模式?有哪些软件设计模式?

设计模式是对软件设计中普遍存在、反复出现的各种问题，根据多次处理的经验，提出的一套能够快速、准确响应此类问题的解决方案。设计模式描述在各种不同情况下，应解决共性问题。

工厂方法模式、抽象工厂模式、单例模式、建造者模式、原型模式、适配器模式、装饰器模式、代理模式、外观模式、桥接模式、组合模式、享元模式、策略模式、模板方法模式、观察者模式、迭代子模式、责任链模式、命令模式、备忘录模式、状态模式、访问者模式、中介者模式、解释器模式。

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，Pxl排序方法更科学?

相比Delphi法和平均排序法，Pxl 排序方法更科学。Pxl 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）。Pxl 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，Pxl 排序方法还给出了更精确的风险图谱。

第八章p223

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

安全编译是指在代码编译阶段采取的哪些安全措施?

- 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。
- 代码编译需要在一个安全的环境中进行。
- 对应用环境的真实模拟也是软件编译需要考虑的问题。
- 在安全编码阶段，多样化编译技术作为一-种提高软件安全性的方法已经得到了应用。

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。

验证输入

留意编译器警告

安全策略的架构和设计

保持简单性

默认拒绝

坚持最小权限原则

清洁发送给其他系统的数据

纵深防御

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。

第九章p250

什么是软件测试?软件测试的目标是什么?

软件测试是保证软件质量的关键步骤，它是对软件规格说明、设计和编码的最后复审。软件测试的目标是在软件投入生产性运行之前，尽可能多地发现软件中的错误。

软件安全测试的目标是什么?它与软件测试有什么区别?

目标：

- 验证软件系统的安全功能是否满足安全需求。
- 发现系统的安全漏洞，并最终把这些漏洞的数量降到最低。
- 评估软件的其他质量属性，包括可靠性、可存活性等。

软件测试主要是从最终用户的角度出发发现缺陷并修复，保证软件满足最终用户的要求。软件安全测试则是从攻击者的角度出发发现漏洞并修复，保证软件不被恶意攻击者破坏。通常普通用户不会去寻找软件漏洞，而恶意攻击者往往会想方设法寻找软件中的安全漏洞。安全测试和传统测试的最主要区别就是安全测试人员要像攻击者--样寻找系统的软肋。

软件安全测试的方法有哪些?

可以将软件安全测试分为白盒测试和黑盒测试。白盒测试可以分为功能测试和源代码分析;黑盒测试又可以分为漏洞扫描、模糊（ Fuzzing）测试和渗透测试等。

软件安全测试的一般流程是什么?

1. 制定安全测试策略。
2. 设计基于风险的安全测试计划。
3. 规范化的软件安全需求。
4. 软件结构风险分析。
5. 执行软件安全测试。
6. 测试环境管理。
7. 测试数据管理。

什么是软件安全功能测试?其测试内容主要有哪些?

软件安全功能测试主要针对需求与设计阶段的核心安全属性的实现状况进行验证，以保证软件安全设计目标的实现。

包括保密性、完整性、可用性、可认证性、授权及可记账性/审计等，软件安全功能测试的内容即围绕上述核心安全属性展开。

软件安全功能测试与软件安全漏洞测试的主要区别是什么?

软件安全功能测试主要针对需求与设计阶段的核心安全属性的实现状况进行验证，以保证软件安全设计目标的实现。

软件安全漏洞测试。安全漏洞测试是有关识别潜在的软件安全缺陷和验证应用程序安全性的过程。它站在攻击者的角度，以发现软件的安全漏洞为目标。

根据代码所处的状态，可以将代码分析分为代码静态分析和代码动态分析两种类型。什么是代码静态分析?什么是代码动态分析?两种分析技术各有什么优点和局限性?

代码静态分析是指在不运行代码的方式下，通过词法分析、语法分析和控制流分析等技术对程序代码进行扫描，验证代码是否满足规范性、安全性、可靠性和可维护性等指标的一种代码分析技术。

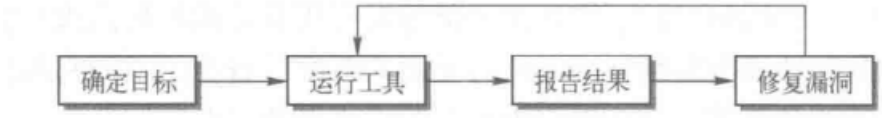
代码静态分析的优点在于，编码错误和漏洞能够尽早地被检测出来，以便于在软件部署之前解决问题。作为集成软件开发环境IDE的一部分，代码静态分析可以在软件开发阶段尽早地检测出安全Bug，并向开发人员及时反馈，使开发人员可以不断地迭代学习，开发出安全性更高的代码。另外，代码静态分析没有模拟产品环境的要求，可以在开发和测试两个阶段进行。

不过，代码静态分析工具还存在误判率和漏判率比较高的情况，代码静态分析结果显示编译结果没有任何错误，并不意味着它能够没有任何错误地运行。

代码动态分析是指对正在运行的代码（或程序）进行检查。代码动态分析可用于确保代码正常可靠地运行。

如果代码没有运行，则不会被分析。此外，动态代码分析不能执行静态代码分析工具的功能

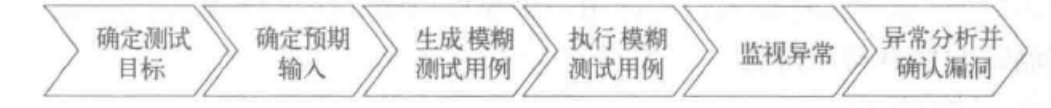
源代码静态分析的一般过程是怎样的?有哪些常用的源代码静态分析工具?



Fortify SCA、Coverity、LAPSE、Find Security Bugs、Flawfinder、RIPS、CodeXploiter、Seay源代码审计系统

什么是模糊测试?模糊测试的过程是怎样的?有哪些常用的模糊测试工具?

使用大量半有效的数据作为应用程序的输入，以程序是否出现异常作为标志，发现应用程序中可能存在的安全漏洞。



FileFuzz、SPIKEfile、notSPIKEfile、PaiMei、Sulley、SPIKE、Peach Fuzzer、Powerfuzzer、SPIKE Proxy、WebScarab、Web Inspect、COMRaider、Mangleme、Hamachi、CSSDIE、American Fuzzy Lop

什么是渗透测试?渗透测试的过程是怎样的?有哪些常用的渗透测试工具?

渗透测试（ Penetration Test）技术的核心思想是模仿黑客的特定攻击行为，也就是尽可能完整地模拟黑客使用的漏洞发现技术和攻击手段，对目标的安全性做深入的探测，发现系统最脆弱环节的过程。

过程：

前期交互、情报收集、威胁建模、漏洞分析、渗透分析、后渗透分析、报告

工具：

Kali Linux系统、Metasploit框架工具

第10章p260

什么是软件部署?软件部署的主要活动有哪些?软件部署的目标有哪些?

软件部署就是在特定平台上按照用户需求安装软件以满足需求的过程。

目标：

- 保障软件系统的正常运行和功能实现。
- 简化部署的操作过程，提高执行效率。
- 满足软件用户在功能和非功能属性方面的个性化需求

软件部署有哪些主要模式?

1. 单机软件的部署模式 2. 基于中间件平台的部署模式 3. 基于代理的部署模式

软件部署包括哪些主要过程?

为什么说软件安全部署是软件安全生命周期的一个重要环节?

软件部署作为软件生命周期中的一个重要环节，是软件生产的后期活动，主要覆盖由软件制品开发完成并交付到软件系统成功运行这一时间段，并随着软件部署向软件运行时管理的延伸，涵盖了运行时的部分时间段或时间点。

软件安全部署有哪些方面的重要工作?

软件安全开发生命周期将安全因素渗透到整个软件开发的生命周期中，以确保安全的软件得以成功实现。软件安全部署作为软件安全生命周期的一个重要环节，除了完成在特定平台上按照用户需求安装软件和功能实现以外，确保软件系统自身的安全和运行环境的安全是重要任务。

软件安装配置过程中，常采用的安全策略有哪些?

- 提供详细的安装手册。

- 可更改的软件安装目录。
- 设置默认安装模块。
- 提供安全功能。
- 启用最小权限用户身份。
- 开启应用日志审计。
- 记录部署过程。

为了确保软件的运行安全，常采取的安全策略有哪些?

- 强制修改默认口令。
- 重视数据备份。
- 注重软件运行期间的漏洞监测与处理。

软件运行基础环境的运行安全策略有哪些?

1. 安装适合本地系统环境的补丁版本。
2. 及时安装官方最新补丁。

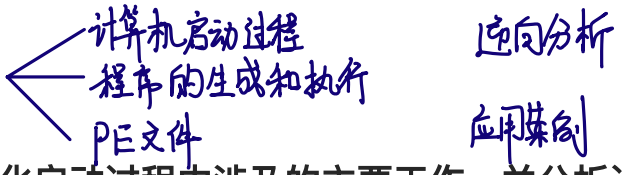
SSL/TLS协议在实现与部署中容易出现的安全问题有哪些?

在目前的应用环境中，OpenSSL、 JSSE等SSL组件的安全是SSL应用安全的基础，如果这些组件在实现时存在与SSL标准不一致的地方， 就会直接影响到上层应用SSL实施安全。

目前的SSL组件多通过提供接口的方式为应用提供SSL协议相关操作，之后应用通过调用这些接口来实现完整的SSL协议。但是在实际应用系统中，经常由于开发者缺乏安全基础而在调用接口时出现错误，造成某些关键步骤缺失或是使用不恰当的接口的情况，从而造成应用系统实现方面的安全问题。

除了SSL协议本身的实现安全以外，SSL协议的安全还依赖于证书、密钥等部分的安全管理和部署。因此如果这些部分存在安全问题，如使用了不安全的密钥或是未对证书和密钥进行妥善管理，也有可能造成整个SsL协议存在安全问题。

第11章p314



简述计算机初始化启动过程中涉及的主要工作，并分析计算机初始化启动过程中面临的安全问题。

电脑

- 按下电源开关，电源就开始向主板和其他设备供电
- BIOS的启动代码进行加电后自检
- BIOS的启动代码选择启动盘

由于BIOS芯片和COMS RAM芯片能够被改写，所以，通过改写BIOS可以加载病毒程序或者损坏BIOS内容，著名的CIH病毒就是这类恶意代码的代表。

简述操作系统启动过程中涉及的主要工作，并分析操作系统启动过程中面临的安全问题。

- 读取指定启动顺序中的存储设备的主引导记录
- 硬盘启动

- 操作系统启动

如果从硬盘引导系统，系统将频繁调用硬盘数据，由于硬盘属于非易失性存储介质，这就为病毒的存在提供了存储空间，因此，以后的每一个步骤都可能激活病毒。内核装载阶段是病毒随启动而加载的主要阶段，在这个启动过程中，内核装载主要与Smss. exe和Winlogon. exe等进程有关，因此，病毒也可能存在于其中。

一个C/C++程序从编写出来到运行，涉及哪些工具?涉及哪些主要环节?

一个C/C++程序从编写出来到运行，涉及的工具有：编辑器、编译器（含汇编器）、链接器和加载器。

1. 首先使用编辑器编辑程序源文件（.c 或. cpp）。
2. 源程序经过编译器被编译为等价的汇编代码，再经过汇编器产生出与目标平台CPU一致的目标代码（. obj），亦称机器语言代码（ 机器码）。
3. 尽管目标代码文件中包含的指令已经可以被目标CPU所执行，但其中可能还包含没有解析（Unresolved）的名称和地址引用等，因此需要链接器把目标代码文件和其他一些库文件和资源文件连接起来，产生出符合目标平台上的操作系统所要求格式的可执行程序（.exe），并保存在磁盘上。
4. 当用户执行.exe程序时，Windows操作系统的加载器会解读链接器记录在可执行程序中的格式信息（PE文件格式），将程序中的代码和数据“布置”在内存中，成为真正可以运行的内存映像。然后，生成一个进程，此后进程便开始运行。

CPU可以解析和运行的程序形式称为什么代码?

机器语言代码

编译器的主要功能有哪些?链接器的主要功能有哪些?

编译器(含汇编器)的基本功能是，将使用一种高级语言编写的程序(源程序)翻译成目标代码(机器语言代码)。

链接器的基本功能是，将编译器产生的多个目标文件合成为一个可以在目标平台下执行的文件。这里说的目标平台是指程序的运行环境，包括CPU和操作系统。其核心工作是符号表解析和重定位。

什么是静态链接?什么是动态链接?两者有什么区别?

- 静态链接。链接器将函数的代码从其所在地(目标文件或静态链接库中)复制到最终的可执行程序中，整个过程在程序生成时完成。静态链接库实际上是一个目标文件的集合，其中的每个文件含有库中的一个或者一 一组相关函数的代码，静态链接则是把相关代码复制到源代码相关位置处参与程序的生成。
- 动态链接。动态链接库在编译链接时只提供符号表和其他少量信息，用于保证所有符号引用都有定义，保证编译顺利通过。程序执行时，动态链接库的全部内容将被映射到运行时相应进程的虚地址空间，根据可执行程序中记录的信息找到相应的函数地址并调用执行。

为什么双击一个. exe程序文件（PE 文件）它就会被Windows运行?

Windows系统中，程序文件(PE文件)中除了存储文件的主体内容(比如.exe文件中的代码、数据等)外，还存储其他一些重要的信息。这些信息是给文件的关联程序用的，比如.exe文件的关联程序就是Windows系统。Windows系统可以根据这些信息知道把文件加载到地址空间的哪个位置，知道从哪个地址开始执行，以及加载到内存后如何修正一些指令中的地址等。

程序文件(PE 文件)中的这些重要信息就是由编译器和链接器完成加入的。针对不同的编译器和链接器，通常会提供不同的选项，让人们在编译和链接生成PE文件时，对其中那些Windows系统需要的信息进行设定。当然，也可以按照默认的方式编译链接生成Windows系统中默认的信息。例如，Window NT默认的程序加载基址是0x40000，可以在用.Visual C++链接生成. exe文件时使用选项更改这个地址值。

在不同的操作系统中可执行文件的格式是不同的，比如在Linux上常用文件格式是ELF。当然，它是由在Linux上的编译器和链接器生成的，所以，编译器和链接器是针对不同的CPU架构和不同的操作系统而设计出来的。在嵌人式领域中经常提到交叉编译器，它的作用就是在一个平 台下编译出能在另一个平台下运行的程序。例如，可以使用交叉编译器在Linux的机器上编译出能在ARM平台，上运行的程序。

为什么系统要把程序文件装载到内存后再执行呢?

内存直接由CPU控制，享受与CPU通信的最优带宽，然而硬盘是通过主板上的桥接芯片与CPU相连，所以速度相对较慢。再加上传统机械式硬盘靠电机带动盘片转动来读写数据，磁头寻道等机械操作耗费时间，而内存条通过电路来读写数据，显然，电机的转速肯定没有电的传输速度快。虽然现在使用固态硬盘大大提升了读写速度，但是由于控制方式依旧不同于内存，读写速度仍然不及内存。

因此，为了程序运行速率，程序在运行时，都通过加载器(Loader), 先将硬盘上的数据复制到内存，然后才让CPU来处理。加载器根据程序PE头中的各种信息，进行堆栈的申请和代码数据的映射装载，在完成所有的初始化工作后，程序从入口点地址进入，开始执行代码段的第一条指令。

当程序运行需要的空间大于内存容量时，加载器会将内存中暂时不用的数据写回硬盘;需要时再从硬盘中读取，并将另外一部分不用的数据写人硬盘。这样，硬盘中的部分空间会用于存储内存中暂时不用的数据，这一部分空间就称为虚拟内存。

什么是PE文件? PE文件有什么用?研究PE文件对于分析恶意代码有什么意义?

微软Windows环境下可执行文件的标准格式是PE文件，其目的是为所有Windows平台设计统一的文件格式，即为Windows平台的应用软件提供良好的兼容性和扩展性。

PE文件不仅包含了二进制的机器代码，还会自带许多其他信息，如字符串、菜单、图标、位图和字体等。PE文件格式规定了在可执行文件中如何组织所有的这些信息。在程序被执行时，操作系统会按照PE文件格式的约定去相应的地方准确地定位各种类型的资源，并分别装人内存的不同区域。PE文件数据资源定位采用链表与固定格式相结合的方式，前者利用链表管理资源，资源的具体位置灵活，后者要求数据结构大小固定，其位置也相对固定。

PE作为“可移植的执行体”意味着此文件格式可用于所有Windows操作系统平台和所有CPU上。对PE文件结构及相关技术的研究是恶意代码研究的基础，因为恶意代码的执行必将直接或者间接地依赖于PE文件。

简述PE文件的基本结构，以及PE文件执行的基本过程。

- DOS头。包括DOS MZ文件头和DOS插桩程序。
- PE头。
- 节表。
- 节。

PE文件在磁盘上就是按照上面的格式顺序存储的。当PE文件被执行时，PE装载器检查DOS MZ文件头里的PE头偏移量。如果找到，则跳转到PE头。PE装载器会检查PE头的有效性，确定该PE文件的总体信息，紧接着读取节表中的节信息，并采用文件映射方法将相应节映射到内存，PE 装载器将处理PE文件中最重要的导人表，从导入表中获取函数字符串名称信息、DLL名称信息及导人函数地址表项起始偏移地址等，最终完成PE文件的执行。

什么是虚拟内存? PE文件与虚拟内存之间是如何映射的?虚拟地址（VA）与相对虚拟地址（RVA）的转化计算式是什么?文件偏移地址（FOA） 与虚拟地址（VA） 的转化计算式是什么?

“ 虚拟内存”的概念，那是指当实际的物理内存不够时，操作系统会把“部分硬盘空间”当作内存使用，从而

使程序得到装载运行的现象。

VA= Image Base+RVA

某数据在PE文件中的偏移地址=某数据的RVA−节偏移

什么是逆向工程?什么是软件逆向工程?

逆向工程，源于商业及军事领域中的硬件分析。其主要目的是，在不能轻易获得必要的生产信息下，直接从对成品的分析入手，推导出产品的设计原理。即对一项目标产品进行逆向分析及研究，从而演绎并得出该产品的处理流程、组织结构和功能性能规格等设计要素，以制作出功能相近的产品。

软件逆向分析工程，是一系列对运行于机器上的低级代码进行等价的提升和抽象，最终得到更加容易被人们所理解的表现形式的过程。

逆向工程对于软件设计与开发人员、信息安全人员，以及恶意软件开发者或网络攻击者而言，都有什么作用?

对于软件设计与开发人员，为了保护自身开发软件的知识产权，一般不会将源程序公开，然而，他们又往往通过对感兴趣的软件进行逆向工程，了解和学习这些软件的设计理念及开发技巧，以帮助自己在软件市场竞争中取得优势。一些游戏玩家通过逆向工程技术来设计和实现游戏的外挂，国内的一些软件汉化爱好者也是通过对外文版的软件进行逆向工程，找到目标菜单的源代码，然后用汉语替换相应的外文，完成软件汉化的。

对于恶意软件开发者或网络攻击者，他们使用逆向分析方法对加密保护技术和数字版权保护技术进行跟踪分析，进而实施破解。他们还常常利用逆向工程技术挖掘操作系统和应用软件的漏洞，进而开发或使用漏洞利用程序，获取应用软件关键信息的访问权，甚至完全控制整个系统。

对于软件开发人员尤其是信息安全人员，可以使用逆向分析技术对二进制代码进行审核，跟踪分析程序执行的每个步骤，主动挖掘软件中的漏洞;也可以进一步对代码实现的质量和鲁棒性进行评估，这为无法通过查阅软件源代码评估代码的质量和可靠性提供了新途径;还可以对恶意程序进行解剖和分析，为清除恶意程序提供帮助。

如何正确应用逆向工程技术?

软件合法用户对软件进行反编译的行为，应不利用所获取的信息开发相似的软件，并不会与著作权所有人正常使用软件冲突，也不会对著作权所有人的合法权益造成不合理的损害。

简述动态逆向分析法和静态逆向分析法，并分析这两类方法各自的特点和应用中面临的困难。

动态分析是一个将目标代码变换为易读形式的逆向分析过程，但是，这里不是仅仅静态阅读变换之后的程序，而是在一个调试器或调试工具中加载程序，然后一边运行程序一边对程序的行为进行观察和分析。

- 动态分析的运行效果严重依赖于程序的输入，因此只能对某次运行时执行的代码进行分析，这就需要构造良好的测试输人集合来保证所有的代码分支都能够执行。
- 在实际分析中，一些场景下无法动态运行目标程序，比如软件的某一模块无法单独运行、设备环境不兼容导致无法运行等。
- 动态分析恶意程序时，虽然可以在虚拟机环境中进行观察和分析，但是目前的恶意程序已有很多具有了检测运行环境的能力，发现了虚拟机环境后不表现出恶意行为，这使得对恶意程序的动态分析失效。

静态逆向分析是相对于动态执行程序进行逆向分析而言的，是指不执行代码而是使用反编译、反汇编工具，把程序的二进制代码翻译成汇编语言，之后，分析者可以手工分析，也可以借助工具自动化分析。

- 程序加壳，这是对付静态分析的常用方法。
- 代码混淆甚至于被加密处理，这也是对付静态分析的常用方法。
- 汇编语言相对来说阅读仍然比较困难，它往往要求分析人员具备很强的代码理解能力，毕竟看不到程序如何处理数据，也看不到它是如何流动的。

软件逆向分析的一般过程是什么?

文本装载、指令解码、语义映射、相关图构造、过程分析、类型分析、结果输出。

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。

- 加密手段有效，解密成本高。
- 使用电子货币支付赎金，变现快，追踪困难。
- 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？

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

第13章p372

试从软件的权益处置角度，谈谈对商业软件、免费软件、共享软件、闭源软件、自由软件及开源软件概念的理解。

- 商业软件是作为商品进行销售获得收益的软件，商业软件是版权贸易的典型模式。通常通过销售软件的使用许可证及技术支持服务向用户收取费用。
- 免费软件是指不需要以金钱购买而免费得到或使用的软件，开源软件通常是免费软件，但通常在使用上会有一些限制
- 共享软件也称试用软件，是以“先使用后付费”的方式销售的享有版权的软件，是商业软件的一种特定形式。
- 闭源软件或专有软件是指源代码在获取、使用和修改上受到特定限制的软件，简单地说就是封闭源代码软件。
- 自由软件在一定程度上是对商业软件及其强化知识产权保护的批评与叛逆。自由软件是一类可以不受限制地自由使用、复制、研究、修改和分发的软件。

自由软件赋予软件使用者哪些“自由”？

- 不论目的如何，有运行(Run)该软件的自由。
- 有研究(Study) 该软件的自由，以及按需改写该软件的自由。
- 有重新发布(Redistribute)该软件的自由，所以每个人都可以借此来敦亲睦邻。
- 有改进(Improve)该软件的自由，以及向公众发布(Release)改写版的自由

试简述开源软件与自由软件的联系与区别。

开源软件由自由软件发展而来

自由软件是一个比开源软件更严格的概念，因此所有自由软件都是开放源代码的，但不是所有的开源软件都能被成为自由软件。只有遵守GPL和BSD许可的开源软件才符合自由软件定义。

在追求自由、分享精神的过程中，自由软件始终将自由作为道德标准，而开源软件则更加注重软件的发展。

所开发的软件中使用了带GPL许可证的开源软件，那么这个软件是不是就要开源?

是

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?

署名权、修改权、复制权、发表权(决定软件是否公之于众的权利)、出租权、发行权、信息网络传播权、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。

软件盗版、逆向工程、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。