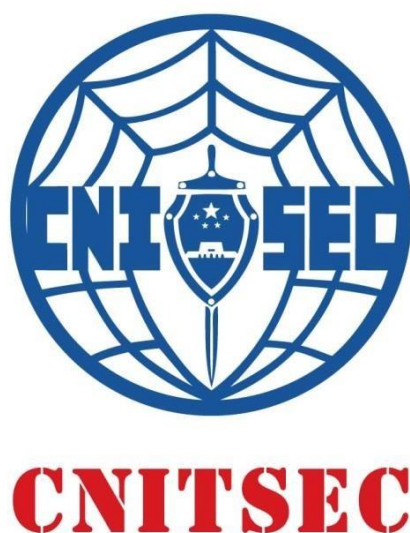


注册信息安全专业人员
软件安全开发专业人员
CISP-SSDP 知识体系大纲



发布日期：2025. 10. 11

生效日期：2025. 10. 11

中国信息安全测评中心

北京启明星辰信息安全技术有限公司 2025©版

目 录

第 1 章：前言	1
第 2 章：注册信息安全专业人员-软件安全开发知识体系概述	2
2.1 知识体系框架结构	2
2.2 考试试题结构	4
第 3 章：知识类：网络安全法律法规	6
3.1 知识体：网络空间安全基础	6
3.2 知识体：网络空间安全监管	8
第 4 章：知识类：软件安全基础概述	10
4.1 知识体：软件与安全概论	10
4.2 知识体：安全开发生命周期 (SSDLC)	12
第 5 章：知识类：软件安全开发核心	14
5.1 知识体：安全需求与架构设计	14
5.2 知识体：安全编码与代码审查规范	17
5.3 知识体：软件安全测试原理与方法	19
第 6 章：知识类：高级防护与 DevSecOps	20
6.1 知识体：高级软件安全测试与运行时保护	20
6.2 知识体：DevSecOps 与安全运维	22
第 7 章：知识类：软件安全治理与供应链防护	24
7.1 知识体：安全治理体系	24
7.2 知识体：软件供应链安全管理	26
第 8 章：知识类：软件安全开发实践	28
8.1 知识体：典型漏洞复现与防御实战	28
8.2 知识体：安全工具应用实战	32

第1章：前言

网络空间信息安全作为我国信息化建设健康发展的重要因素，关系到贯彻落实科学发展观、全面建成小康社会、构建社会主义和谐社会和建设创新型社会等国家战略举措的实施，是国家安全的重要组成部分。在网络空间信息系统安全保障工作中人是最核心也是最活跃的因素，人员的信息安全意识、知识与技能已经成为保障信息系统安全稳定运行的重要基本要素之一。

注册信息安全专业人员（CISP）是对我国网络基础设施和重要信息系统的信息安全专业人员进行资质评定的重要形式。多年来为落实我国有关政策“加快信息安全人才培养，增强全民信息安全意识”的指导精神，构建信息安全人才体系发挥了巨大作用。

本大纲从我国国情出发，结合我国网络基础设施和重要信息系统安全保障的实际需求，以知识体系的全面性和实用性为原则，明确规定了注册信息安全专业人员（软件安全开发）应当掌握的知识要点。主要面向软件开发人员、技术负责人、项目经理、产品经理、网络安全开发人员等角色岗位，是CISP-SSDP教材与课件编制，讲师授课，学员学习，以及考试命题的重要依据。

本大纲包含以下章节：

第3章 知识类： 网络安全法律法规

第4章 知识类： 软件安全基础概述

第5章 知识类： 软件安全开发核心

第6章 知识类： 高级防护与DevSecOps

第7章 知识类： 软件安全治理与供应链防护

第8章 知识类： 软件安全开发实践

第 2 章：注册信息安全专业人员—软件安全开发知识体系概述

注册信息安全专业人员—软件安全开发方向为：

- 注册信息安全专业人员软件安全开发专业人员，英文为 Certified Information Security Professional – Software Security Development Professional，简称 CISP-SSDP。证书持有人员主要从事软件开发领域或安全开发类工作，具有基本的安全需求分析，架构设计、安全编码与测试、DevSecOps、安全管理等知识和能力。

2.1 知识体系框架结构

知识体系使用组件模块化的结构，包括知识类、知识域、知识子域、知识点四个层次。

- 知识类：是对软件安全知识领域的总体划分，包含信息安全专业人员需要掌握的不同知识类别；
- 知识体：是知识类中由属于同一技术领域的知识内容构成的相对独立、成体系的知识集合；
- 知识域：是对知识体进一步分解细化形成的完整的知识组件；
- 知识子域：是构成知识域的基本模块，由一至多个具体知识要点构成。

本大纲规定了知识子域中每一个知识要点的内容和深度要求，分为“了解”“理解”和“掌握”三类

- 了解：是最低深度要求，学员需要正确认识该知识要点的基本概念和原理；
- 理解：是中等深度要求，学员需要在正确认识该知识要点的基本概念和原理的基础上，深入理解其内容，并可以进一步地判断和推理；
- 掌握：是最高深度要求，学员需要正确认识该知识要点的概念、原理，并在深入理解的基础上灵活运用。

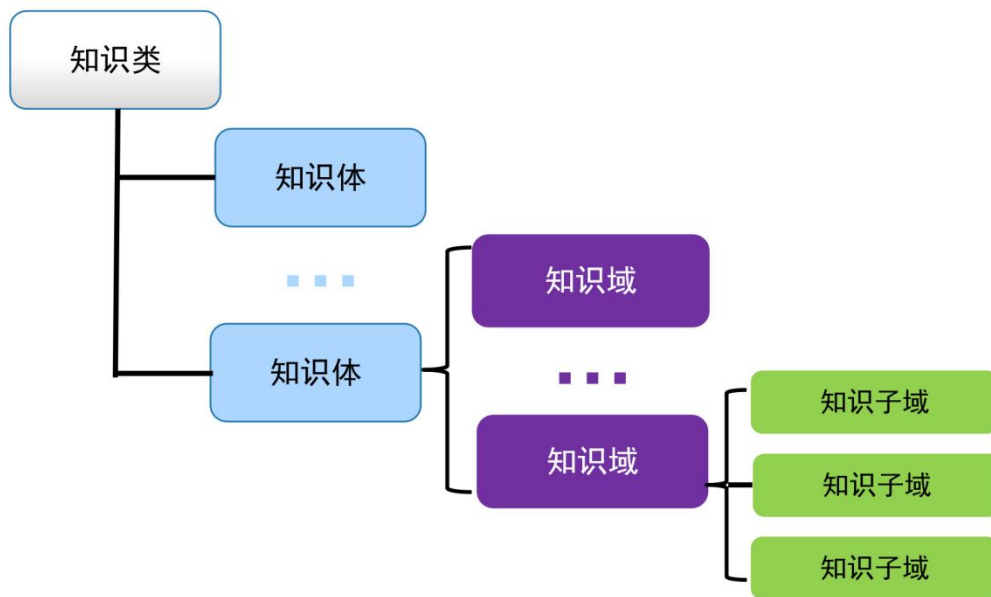


图 2-1:知识体系的组件模块结构

在整个知识体系结构中，共包括网络安全法律法规、软件安全基础概述、软件安全开发核心、高级防护与DevSecOps、软件安全治理与供应链防护、软件安全开发实践六个知识类，每个知识类根据其逻辑划分为多个知识体，每个知识体包含多个知识域，每个知识域由一个或多个知识子域组成。

知识体系结构共包含六个知识类，分别为：

- 网络安全法律法规：主要包括习近平强国思想、网络空间安全基础知识、信息安全保障、网络安全法律体系、我国网络安全战略发展、网络安全行业道德准则、信息安全行业标准。
- 软件安全基础概述：主要包括软件的本质与发展、软件安全危机与价值、企业安全风险、传统的SDLC与安全挑战、SDLC核心原则与活动。
- 软件安全开发核心：主要包括安全需求捕获与分析、威胁建模理论与方法、安全架构与设计模式原理、安全编码规范、代码安全审查原理与静态分析原理、动态应用安全测试原理、模糊测试原理。
- 高级防护与DevSecOps：主要包括交互式应用安全测试与运行时应用自我保护原理、渗透测试方法论、软件成分分析原理与方法、DevSecOps核心理念与持续安全集成方法论、安全部署与配置管理策略、安全监控

与应急响应理论。

- 软件安全治理与供应链防护：主要包括安全治理框架与策略、安全合规基础与审计、供应链安全威胁概论、供应链安全防护策略。
- 软件安全开发实践：主要包括Web应用注入类漏洞实践、Web 应用前端与鉴权漏洞实践、不安全配置与文件处理实践、后端服务与API安全漏洞实践、组件安全与供应链攻击实践、SAST 工具实战、DAST 工具实战、SCA 工具实战、CI/CD 安全集成等实践知识。本知识类是前几个知识类的相关实践过程，具体内容可查阅最后第八章节详细内容，在大纲体系结构图中不再显示。



注册信息安全专业人员-注册软件安全开发专业人员(CISP-SSDP)知识体系结构

图 2-2 CISP-SSDP知识体系结构框架

2.2 考试试题结构

CISP-SSDP 考试题型为选择题与实操题，总分共 100 分。其中选择题 20 分，实操题 80 分，得到 70 分以上（含 70 分）为通过。

表2-1：CISP-SSDP 试题结构

知识类别	占比	题型
网络安全法律法规	10%	客观
软件安全基础概述	10%	客观
软件安全开发核心	20%	客观+实操

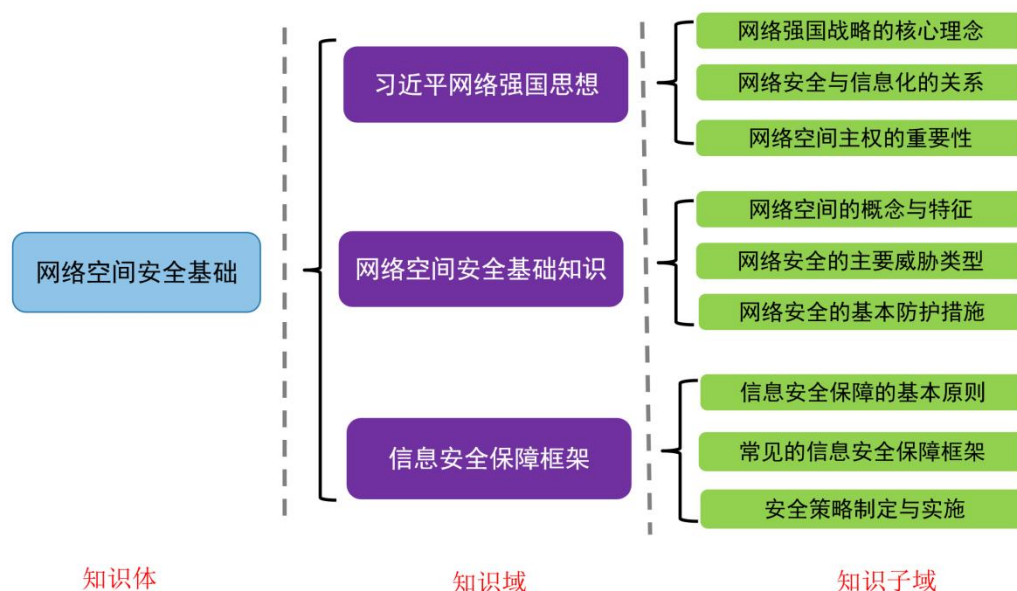
高级防护与 DevSecOps	20%	客观+实操
软件安全治理与供应链防护	10%	客观+实操
软件安全开发实践	30%	客观+实操

第3章：知识类：网络安全法律法规

网络安全法律法规是指引和规范组织、企业在网络空间安全行为、数据处理活动及个人信息保护的强制性法律体系，以及支撑这些法律实施、提供技术指导的国家标准和行业规范。它们共同构筑了保障国家网络安全、社会公共利益和公民合法权益的基石。通过本部分的学习，学员应当：

- 了解我国在网络空间安全的战略布局
- 理解我国的网络空间安全法律法规体系的建设过程和主要法律法规的关键内容
- 理解网络安全监管和保障的重要性
- 了解我国在网络空间安全的布局和产业发展趋势
- 掌握我国网络安全从业人员的职业道德要求
- 理解我国与国际主要信息安全标准的应用

3.1 知识体：网络空间安全基础



3.1.1 知识域：习近平网络强国思想

- 知识子域：网络强国战略的核心理念

- ◆ 理解国家在网络空间治理中的总体战略布局
- ◆ 掌握网络强国思想的理论支撑与实践路径
- 知识子域：网络安全与信息化的关系
 - ◆ 理解信息化发展的前提条件是网络安全保障
 - ◆ 理解网络安全在推动数字化转型中的作用
- 知识子域：网络空间主权的重要性
 - ◆ 理解网络空间主权的基本定义和内涵
 - ◆ 了解维护网络空间主权的战略意义

3.1.2 知识域：网络空间安全基础知识

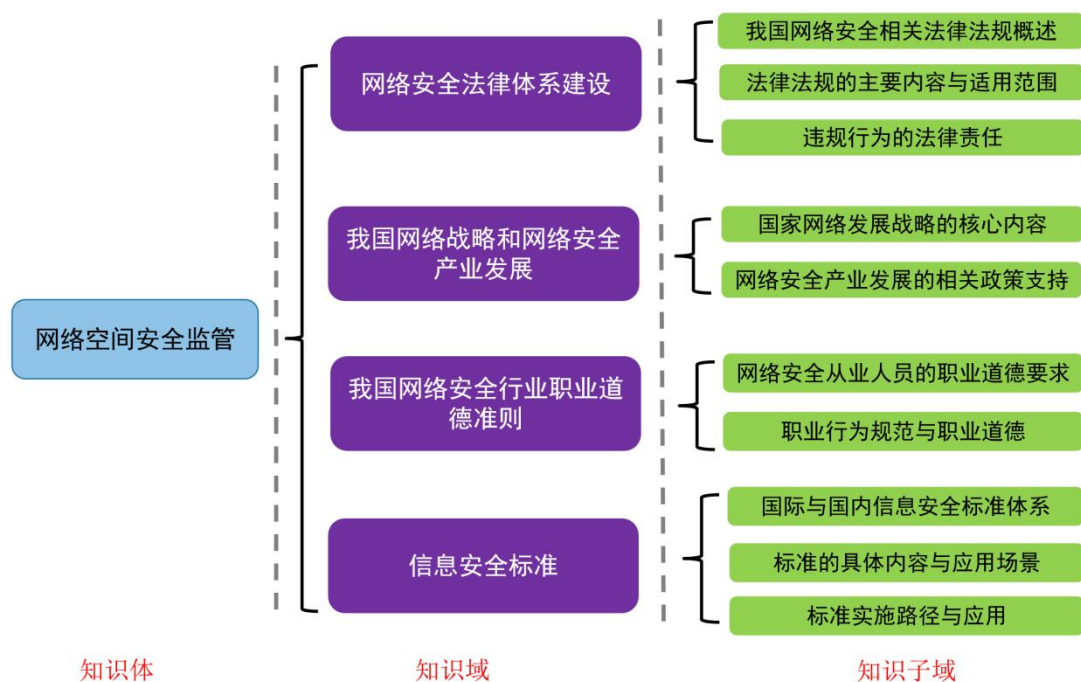
- 知识子域：网络空间的概念与特征
 - ◆ 理解网络空间作为第五大主权疆域的地位
 - ◆ 了解其虚拟性、全球性和动态性等特征
- 知识子域：网络安全的主要威胁类型
 - ◆ 掌握当前主要的网络安全威胁分类
 - ◆ 了解识别典型攻击手段及应对策略
- 知识子域：网络安全的基本防护措施
 - ◆ 了解基本的网络安全防护机制
 - ◆ 理解如何根据场景选择合适的防护手段

3.1.3 知识域：信息安全保障框架

- 知识子域：信息安全保障的基本原则
 - ◆ 掌握信息安全保障的 CIA 三原则
 - ◆ 理解 CIA 三原则的重要性
- 知识子域：常见的信息安全保障框架
 - ◆ 了解主流的信息安全管理体系框架
 - ◆ 理解如何能够根据组织需求选择合适框架
- 知识子域：安全策略制定与实施
 - ◆ 理解安全策略制定的步骤和方法

- ◆ 了解撰写符合组织实际的安全策略文档

3.2 知识体：网络空间安全监管



3.2.1 知识域：网络安全法律体系建设

- 知识子域：我国网络安全相关法律法规概述
 - ◆ 了解我国网络安全法律体系的基本结构
 - ◆ 了解我国关于 ICT 供应链与软件安全技术的政策法规
 - ◆ 理解主要法律法规的适用范围与责任划分
- 知识子域：法律法规的主要内容与适用范围
 - ◆ 理解各项法规的适用对象与法律责任
 - ◆ 理解企业如何满足合规要求
- 知识子域：违规行为的法律责任
 - ◆ 理解违反网络安全法规可能带来的行政、刑事后果
 - ◆ 了解企业如何规避法律风险

3.2.2 知识域：我国网络战略和网络安全产业发展

- 知识子域：国家网络发展战略的核心内容
 - ◆ 理解国家战略层面的网络安全布局
 - ◆ 了解未来发展方向与重点支持领域
- 知识子域：网络安全产业发展的相关政策支持
 - ◆ 了解国家对网络安全企业的扶持政策
 - ◆ 了解国家重点支持的网络安全技术方向

3.2.3 知识域：我国网络安全行业职业道德准则

- 知识子域：从业人员的职业道德要求
 - ◆ 理解网络安全从业人员应具备的职业操守
 - ◆ 掌握职业伦理的基本要求
- 知识子域：职业行为规范与职业道德
 - ◆ 掌握从业过程中应遵循的行为规范
 - ◆ 理解伦理责任对行业发展的影响
 - ◆ 理解违反职业道德可能导致的严重后果

3.2.4 知识域：信息安全标准

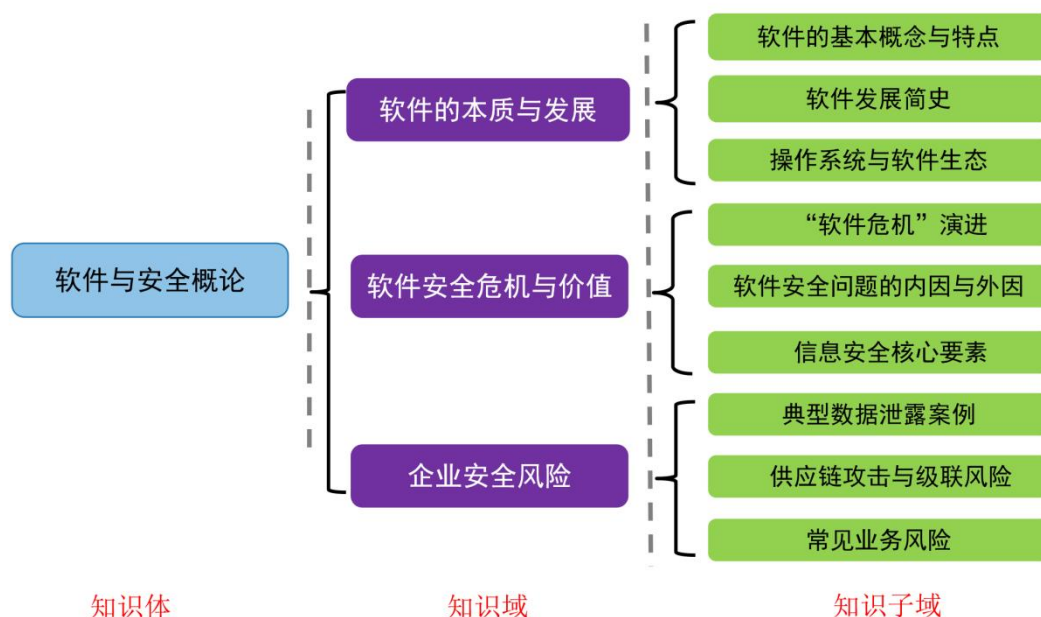
- 知识子域：国际与国内信息安全标准体系
 - ◆ 了解主流国际与国内信息安全标准体系
 - ◆ 理解标准之间的异同与互认机制
- 知识子域：标准的具体内容与应用场景
 - ◆ 了解标准的具体规范与实施要点
 - ◆ 了解如何结合实际业务场景应用标准
- 知识子域：标准实施路径与应用
 - ◆ 理解标准选型的方法与评估指标
 - ◆ 理解如何制定标准实施路线图

第 4 章：知识类：软件安全基础概述

软件的发展和软件的安全问题原因是开发安全软件的前提认知。通过本部分的学习，学员应当：

- 了解软件的本质与发展
- 理解软件漏洞危机与安全价值
- 理解企业安全风险的影响因素
- 理解传统 SDLC 与安全挑战
- 掌握 SSDLC 核心原则与活动

4.1 知识体：软件与安全概论



4.1.1 知识域：软件的本质与发展

- 知识子域：软件的基本概念与特点
 - ◆ 了解软件的基本组成
 - ◆ 理解软件的抽象性、非物质性、不可磨损性、依赖性、定制化和复杂性等特点

- 知识子域：软件发展简史
 - ◆ 了解机器语言、汇编语言到高级语言的诞生与演进
 - ◆ 理解编程范式（面向过程、面向对象、函数式编程等）对软件开发的影响
- 知识子域：操作系统与软件生态
 - ◆ 了解 UNIX、Linux、Windows 等操作系统发展历程
 - ◆ 理解软件生态系统（如云原生、移动应用生态）的形成与特点

4.1.2 知识域：软件安全危机与价值

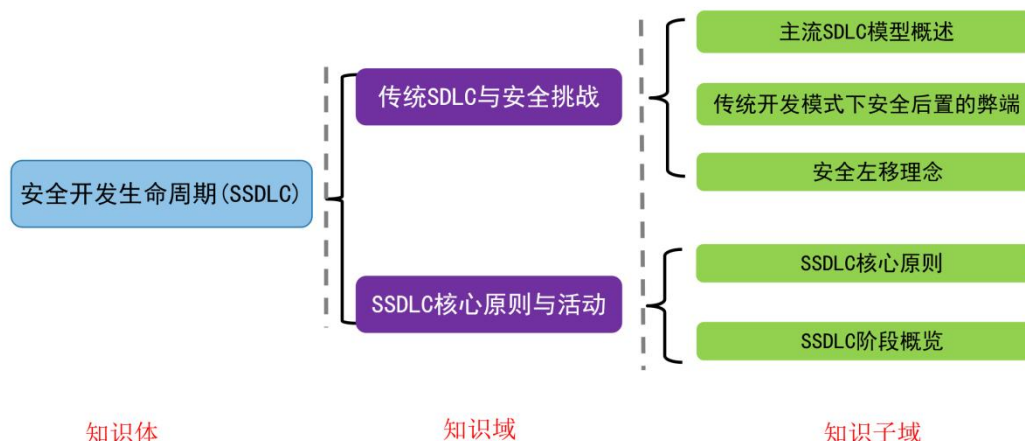
- 知识子域：“软件危机”演进
 - ◆ 理解第一次“软件危机”源于汇编语言的局限性
 - ◆ 理解第二次“软件危机”源于大型程序开发管理挑战
 - ◆ 理解第三次“软件危机”的根源是互联网普及、攻击专业化、软件复杂性增加及开发中对安全的忽视
- 知识子域：软件安全问题的内因与外因
 - ◆ 了解软件安全问题的内因（开发与管理视角）
 - ◆ 了解软件安全问题的外因（环境与攻击视角）
- 知识子域：软件安全属性保障
 - ◆ 掌握如何在软件生命周期中识别并保障软件安全属性，例如数据加密保障保密性，哈希校验保障完整性，高可用架构保障可用性
- 知识子域：软件安全与传统网络安全
 - ◆ 理解传统网络安全侧重边界防御的重点
 - ◆ 掌握并认识到随着云计算和 DevOps 的普及，形成更全面的安全防护体系

4.1.3 知识域：企业安全风险

- 知识子域：典型数据泄露案例
 - ◆ 了解近年来国内外典型数据泄露事件的发生过程和影响。
 - ◆ 理解数据泄露对企业声誉、财务及用户信任造成的深远影响。

- 知识子域：供应链攻击与级联风险
 - ◆ 了解国内外典型软件供应链攻击案例的背景和攻击链
 - ◆ 理解供应链攻击如何通过上游软件或组件缺陷导致大规模级联影响
 - ◆ 掌握如何识别企业自身在软件供应链中的潜在风险点
- 知识子域：常见业务风险
 - ◆ 理解勒索软件、服务中断、经济损失、法律合规风险、知识产权窃取、企业声誉受损等常见的软件安全引发的业务风险
 - ◆ 了解如何能够评估这些风险对自身业务运营的影响

4.2 知识体：安全开发生命周期 (SSDLC)



4.2.1 知识域：传统 SDLC 与安全挑战

- 知识子域：主流 SDLC 模型概述
 - ◆ 了解瀑布模型、敏捷开发与 DevOps 的基础概念及其特点
- 知识子域：传统开发模式下安全“后置”的弊端
 - ◆ 理解安全测试在开发末期进行，导致发现晚、修复成本高、迭代冲突等问题
 - ◆ 掌握并认识“安全债务”的积累及其对项目进度和质量的影响
- 知识子域：安全左移（Shift Left Security）理念
 - ◆ 理解安全实践前置到软件开发生命周期的早期阶段的战略意义
 - ◆ 掌握并深刻理解“安全左移”能显著降低安全风险和修复成本，提

升软件整体质量

4.2.2 知识域：SSDLC 核心原则与活动

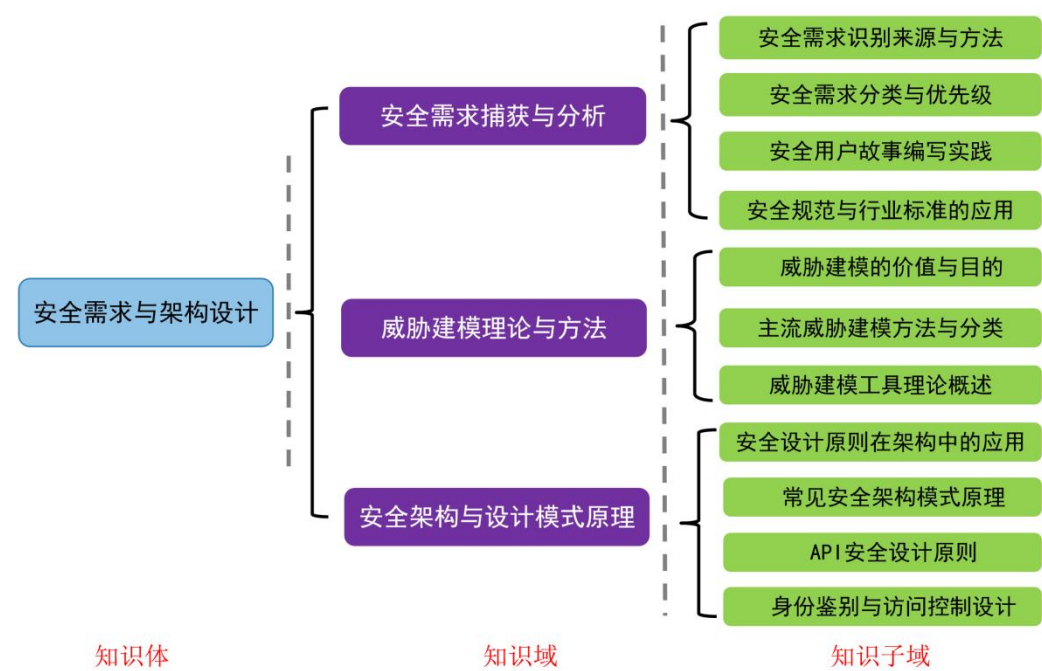
- 知识子域：SSDLC 核心理念
 - ◆ 理解安全左移理念的重要性及其在软件开发全生命周期中的体现
 - ◆ 理解安全原则是贯穿 SSDLC 全过程的指导思想，并能在概念层面阐述其在软件开发中的作用
- 知识子域：SSDLC 阶段与活动
 - ◆ 了解软件安全开发从安全需求、安全设计、安全编码、安全测试、安全部署到安全运维与响应的全生命周期中各阶段的活动
 - ◆ 理解各阶段安全活动的输入、输出和相互关系，以及如何在这阶段融入安全实践

第 5 章：知识类：软件安全开发核心

软件从概念到编码阶段，如何在设计、编码层面构建安全功能，并理解各种安全实践的理论、方法论和规范是构建安全软件的关键环节。通过本部分的学习，学员应当：

- 掌握如何从不同来源获取安全需求
- 掌握威胁建模的理论方法和常见工具的使用
- 理解安全架构与设计模式的原理
- 掌握常见的安全编码规范和原理
- 理解代码安全审查原理与静态分析技术的原理
- 理解动态应用安全测试与模糊测试方法的原理

5.1 知识体：安全需求与架构设计



5.1.1 知识域：安全需求捕获与分析

- 知识子域：安全需求识别来源与方法
 - ◆ 掌握从业务需求、用户案例、历史漏洞、威胁情报、合规性要求等

多种来源识别安全需求。

- 知识子域：安全需求分类与优先级
 - ◆ 理解功能性安全需求与非功能性安全需求的区别，并能够对其进行分类
 - ◆ 掌握能够根据风险等级和业务重要性对安全需求进行优先级排序
- 知识子域：安全用户故事（Security User Stories）编写实践
 - ◆ 掌握将抽象的安全需求转化为具体、可测试、可实现的用户故事（例如：“作为攻击者，我不能通过 SQL 注入绕过登录验证”）
- 知识子域：安全规范与行业标准的应用
 - ◆ 了解《GB/T 43848-2024 网络安全技术 软件产品开源代码安全评价方法》基本内容
 - ◆ 了解《网络安全技术 软件安全开发能力评估准则》（征求意见稿）标准的基本内容和意义
 - ◆ 理解如何利用这些标准来验证和完善自身项目的安全需求

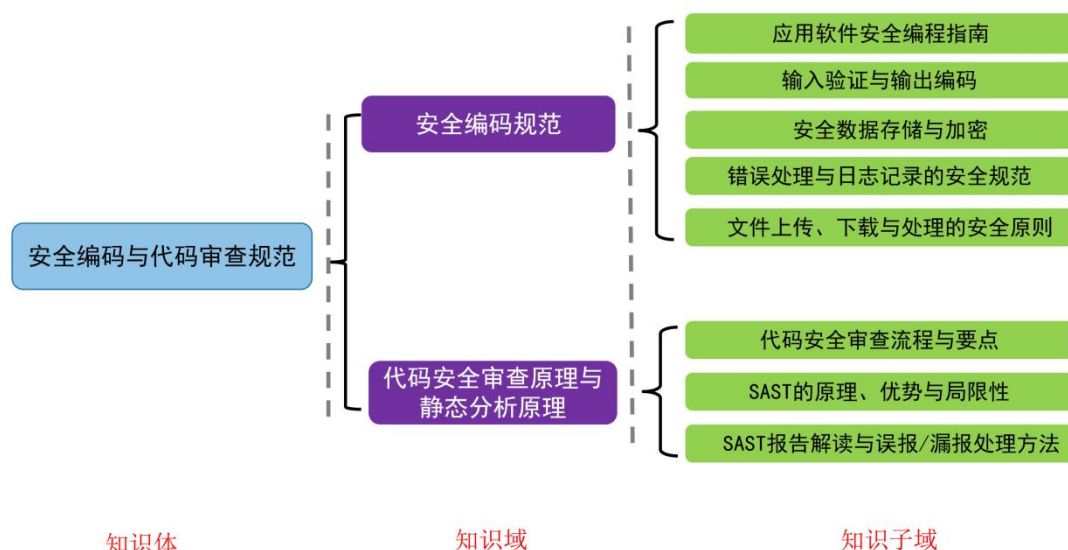
5.1.2 知识域：威胁建模理论与方法

- 知识子域：威胁建模的价值与目的
 - ◆ 理解威胁建模在软件生命周期早期发现设计缺陷、指导安全测试、提升系统整体安全性方面的核心价值
 - ◆ 掌握能够向团队或领导阐述威胁建模的必要性
- 知识子域：主流威胁建模方法与分类（STRIDE）
 - ◆ 掌握并熟练运用 STRIDE 威胁分类方法，并能识别相应威胁类型
- 知识子域：威胁建模实践步骤理论
 - ◆ 掌握能够独立完成数据流图（DFD）绘制、威胁识别、风险评估与缓解措施设计等威胁建模核心步骤
- 知识子域：威胁建模工具理论概述
 - ◆ 了解常见威胁建模工具的基本功能和应用场景
 - ◆ 理解如何借助工具辅助威胁建模过程

5.1.3 知识域：安全架构与设计模式原理

- 知识子域：安全设计原则在架构中的应用
 - ◆ 掌握如何在系统架构设计中主动运用和实践隔离、模块化、最小化暴露面、纵深防御、最小权限等安全原则，以构建弹性、高可用和安全的软件应用系统
 - ◆ 了解并能够分析不同架构模式下，这些安全原则的具体实现方式和挑战
- 知识子域：常见安全架构模式原理
 - ◆ 理解安全域划分、安全网关、统一身份认证中心、审计日志系统等常见安全架构模式的功能和适用场景
 - ◆ 了解微服务架构下的安全挑战与模式
- 知识子域：API 安全设计原则
 - ◆ 掌握对 API 设计进行认证、授权、传输安全、速率限制、输入参数校验等安全考虑和实现方法
 - ◆ 理解 RESTful API 和 GraphQL API 的安全差异
- 知识子域：身份鉴别与访问控制设计
 - ◆ 掌握并能够熟练应用认证(Authentication)、授权(Authorization)、会话管理、单点登录(SSO)、OAuth2.0/OIDC 等常见身份认证鉴别协议及实践
 - ◆ 理解基于角色的访问控制(RBAC)/基于属性的访问控制(ABAC)等访问控制模型的原理和适用场景

5.2 知识体：安全编码与代码审查规范



5.2.1 知识域：安全编码规范

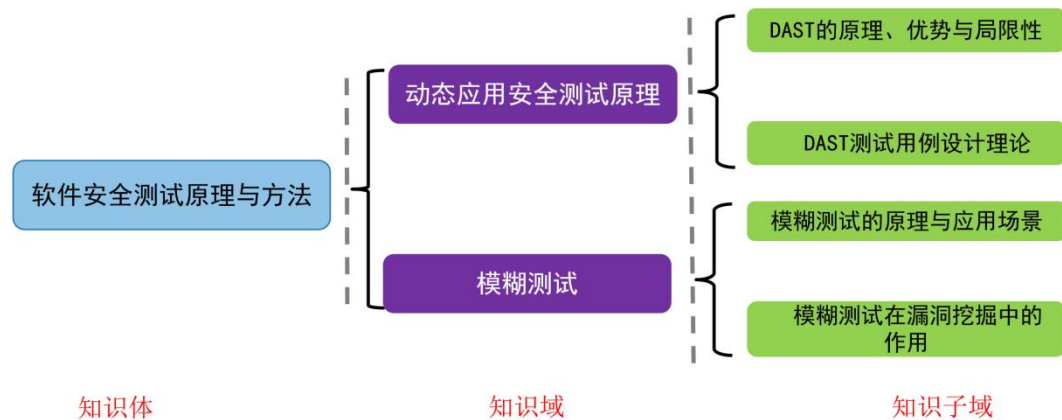
- 知识子域：应用软件安全编程指南
 - ◆ 了解《GB/T 38674-2020 信息安全技术 应用软件安全编程指南》基本内容
 - ◆ 理解应用软件安全编程框架的核心思想
- 知识子域：输入验证与输出编码
 - ◆ 掌握并能够熟练运用输入验证（如白名单、正则表达式校验）和输出编码（如 HTML 实体编码、URL 编码）等技术，有效防御 XSS、SQL 注入等常见漏洞
 - ◆ 了解如何在代码中正确实现输入验证和输出编码
- 知识子域：安全数据存储与加密
 - ◆ 掌握如何通过密码哈希（加盐、迭代哈希）、密钥管理（KMS）、敏感数据加密存储的最佳实践
 - ◆ 理解对称加密与非对称加密的基本原理和适用场景
- 知识子域：错误处理与日志记录的安全规范
 - ◆ 掌握避免在错误信息中泄露敏感系统或业务信息。

- ◆ 理解规范化日志记录的内容（如事件类型、时间、用户 ID、操作结果），确保日志的完整性和可审计性
- 知识子域：文件上传、下载与处理的安全原则
 - ◆ 掌握防止任意文件上传、路径遍历、文件包含等漏洞的防御策略（如文件类型校验、内容校验、大小限制、文件隔离存储、执行权限限制）
 - ◆ 了解如何在文件处理模块中应用这些安全措施

5.2.2 知识域：代码安全审查原理与静态分析 (SAST) 原理

- 知识子域：代码安全审查（Code Review）流程与要点
 - ◆ 理解代码审查在早期发现缺陷、提高代码质量、传播安全知识的重要作用
 - ◆ 掌握安全代码审查的常见要点（如凭证硬编码、弱加密算法、不安全配置）。理解：代码审查在早期发现缺陷、提高代码质量、传播安全知识的重要作用
 - ◆ 掌握安全代码审查的常见要点（如凭证硬编码、弱加密算法、不安全配置）
- 知识子域：SAST 的原理、优势与局限性
 - ◆ 理解 SAST 在代码编写阶段发现缺陷的能力，以及其无法发现运行时漏洞和业务逻辑漏洞的局限性
 - ◆ 了解 SAST 相对于 DAST、IAST 的特点
- 知识子域：SAST 报告解读与误报/漏报处理方法
 - ◆ 掌握能够有效地分析 SAST 扫描报告，区分真实漏洞和误报，并进行优先级排序和修复的方法论
 - ◆ 理解误报和漏报产生的原因，以及如何优化 SAST 规则。

5.3 知识体：软件安全测试原理与方法



5.3.1 知识域：动态应用安全测试（DAST）原理

- 知识子域：DAST 的原理、优势与局限性
 - ◆ 理解 DAST 模拟攻击者行为在运行时发现漏洞的能力，以及其可能无法覆盖所有代码路径的局限性
 - ◆ 了解 DAST 相对于 SAST 的优势（能发现配置错误、部署问题、某些业务逻辑漏洞）
- 知识子域：DAST 测试用例设计理论
 - ◆ 掌握能够设计有效的 DAST 测试用例的理论方法

5.3.2 知识域：模糊测试 (Fuzz Testing)

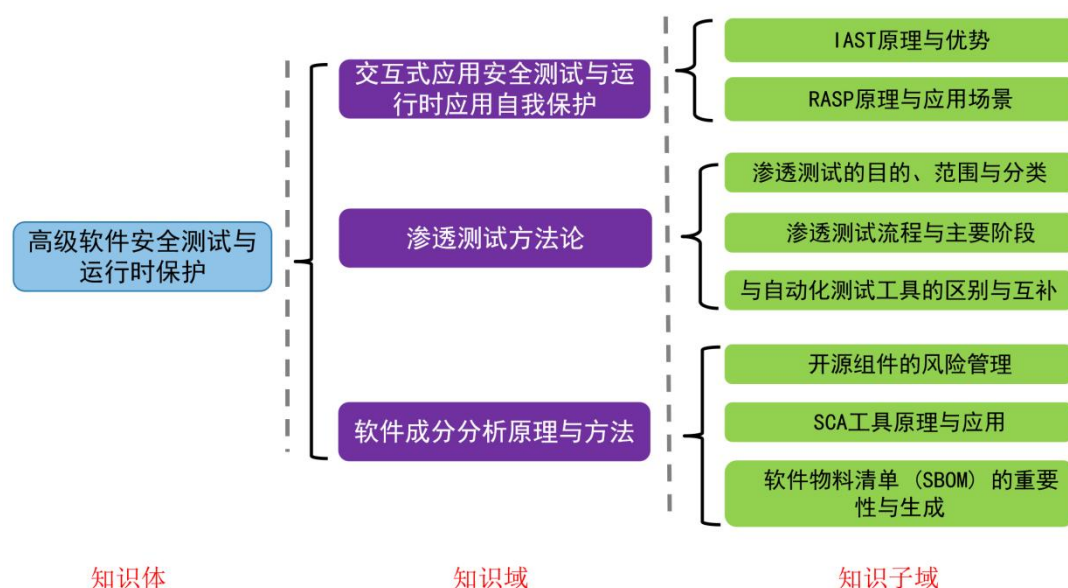
- 知识子域：模糊测试的原理与应用场景
 - ◆ 理解模糊测试通过向程序输入大量畸形、异常或随机数据，以发现可能导致程序崩溃或异常处理的漏洞的原理
- 知识子域：模糊测试在漏洞挖掘中的作用
 - ◆ 理解模糊测试在挖掘深层、未知缺陷方面的独特优势

第 6 章：知识类：高级防护与 DevSecOps

软件交付后，部署、运行和持续运维阶段，仍需要通过高级安全测试、自动化集成方法以及体系化的安全运维策略来保障软件的持续安全。通过本部分的学习，学员应当：

- 理解交互式应用安全测试与运行时应用自我保护方法原理
- 理解渗透测试方法论的核心活动和目的
- 掌握 DevSecOps 核心理念与各个阶段活动
- 理解安全部署与配置管理策略的重要性的方法
- 理解安全监控与应急响应理论的作用

6.1 知识体：高级软件安全测试与运行时保护



6.1.1 知识域：交互式应用安全测试与运行时应用自我保护

- 知识子域：IAST 原理与优势
 - ◆ 理解 IAST 结合 SAST 与 DAST 优势，在应用运行时进行实时分析的原理、优点（覆盖率高、误报少）与适用场景
 - ◆ 了解常见 IAST 工具及其部署方式

- 知识子域： RASP 原理与应用场景
 - ◆ 理解 RASP 在应用程序内部进行实时防护，无需修改代码的原理、部署方式与保护范围（如 SQL 注入、XSS 等）
 - ◆ 了解 RASP 与 WAF（Web 应用防火墙）的区别与互补

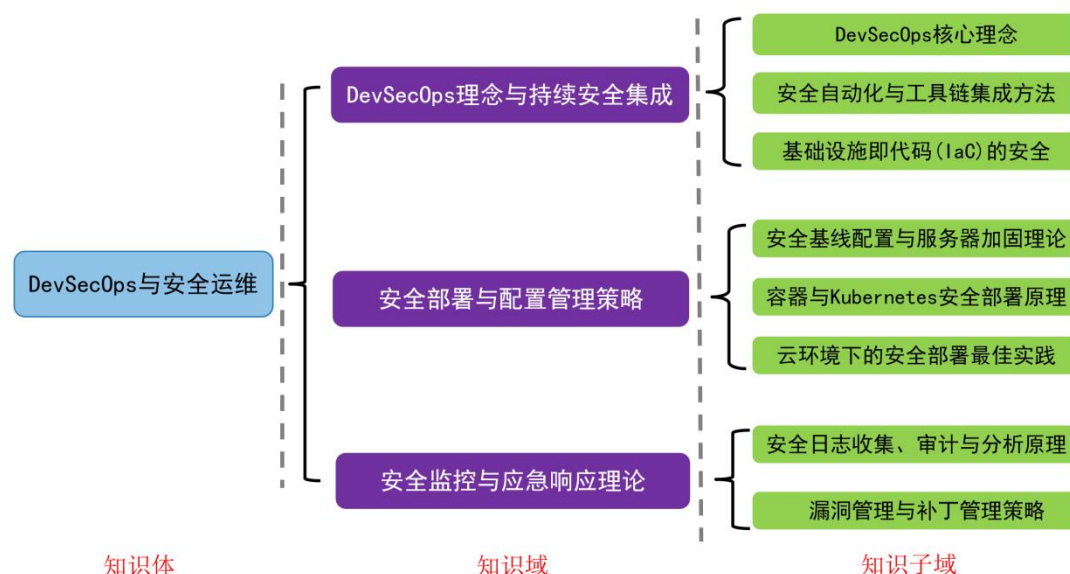
6.1.2 知识域： 渗透测试方法论

- 知识子域： 渗透测试的目的、范围与分类
 - ◆ 理解渗透测试作为一种模拟真实攻击，发现深层漏洞和业务逻辑漏洞的目的
 - ◆ 掌握黑盒、白盒、灰盒测试的区别与适用性
- 知识子域： 渗透测试流程与主要阶段
 - ◆ 掌握信息收集、漏洞扫描、漏洞利用、权限提升、痕迹清除、报告编写等渗透测试阶段的核心活动和常用工具
- 知识子域： 与自动化测试工具的区别与互补
 - ◆ 理解渗透测试的人工经验和对业务逻辑漏洞的发现能力是自动化工具无法替代的
 - ◆ 掌握如何将渗透测试结果反哺到 SSDLC 中

6.1.3 知识域： 软件成分分析 （SCA）原理与方法

- 知识子域： 开源组件的风险管理
 - ◆ 掌握如何识别开源组件引入的已知漏洞（CVE）、许可证合规性（GPL, MIT 等）风险
- 知识子域： SCA 工具原理与应用
 - ◆ 理解 SCA 工具如何识别项目中的第三方组件、追踪已知漏洞、分析许可证合规性
- 知识子域： 软件物料清单（SBOM）的重要性与生成
 - ◆ 理解 SBOM 作为软件供应链透明度的关键工具，能够帮助企业清晰了解其软件构成
 - ◆ 了解 SBOM 的组成要素和生成方法

6.2 知识体：DevSecOps 与安全运维



6.2.1 知识域：DevSecOps 理念与持续安全集成

- 知识子域：DevSecOps 核心理念
 - ◆ 理解 DevSecOps 如何将安全融入 DevOps 的各个环节，实现自动化、持续化的安全保障
 - ◆ 掌握 DevSecOps 与“安全左移”的紧密关系
- 知识子域：安全自动化与工具链集成方法
 - ◆ 掌握在 CI/CD 管道中集成自动化安全工具（SAST, DAST, SCA）的最佳实践方法
 - ◆ 理解如何实现安全扫描结果的自动化分析和告警
- 知识子域：基础设施即代码（IaC）的安全
 - ◆ 理解对 Terraform、Ansible 等 IaC 代码进行安全审计，避免引入配置漏洞
 - ◆ 了解静态分析 IaC 工具的原理

6.2.2 知识域：安全部署与配置管理策略

- 知识子域：安全基线配置与服务器加固理论

- ◆ 理解对操作系统（Unix/Linux/Windows）的安全基线配置和加固原则
- ◆ 理解对中间件（如 Nginx/Tomcat 等）的安全基线配置和加固原则
- ◆ 理解对数据库（如 MySQL 安全配置）的安全基线配置和加固原则
- 知识子域：容器与 Kubernetes 安全部署原理
 - ◆ 理解容器镜像安全（扫描、签名）、容器运行时安全、网络隔离、Pod 安全策略等容器化环境下的安全部署最佳实践
 - ◆ 了解 Kubernetes 集群的安全配置最佳实践
- 知识子域：云环境下的安全部署最佳实践
 - ◆ 理解云上身份与访问管理（IAM）、网络隔离（VPC、安全组）、数据加密（静态加密、传输加密）、日志审计等云安全措施。
 - ◆ 了解不同云平台的安全配置要点

6.2.3 知识域：安全监控与应急响应理论

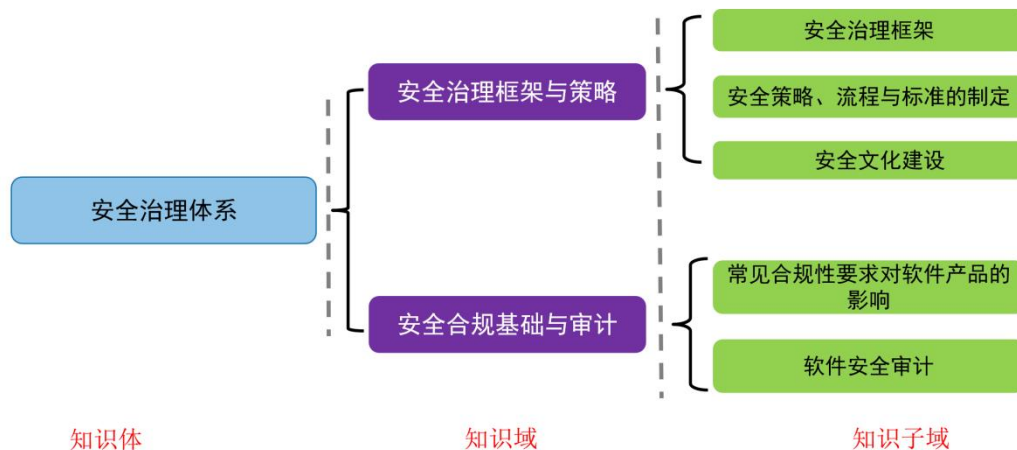
- 知识子域：安全日志收集、审计与分析原理
 - ◆ 理解 SIEM (Security Information and Event Management) 系统的作用，并掌握日志的规范化、集中化管理和分析方法
 - ◆ 理解如何能够配置应用程序的日志记录，使其符合安全审计要求
- 知识子域：威胁情报与安全告警管理
 - ◆ 理解如何利用威胁情报（如 CVE、CWE）提升防御能力
 - ◆ 掌握高效处理安全告警的流程，减少误报对开发运维的影响
- 知识子域：安全事件应急响应流程
 - ◆ 掌握安全事件应急响应的识别、抑制、根除、恢复、复盘的各个阶段，以及团队协作机制
 - ◆ 了解常见的应急响应框架
- 知识子域：漏洞管理与补丁管理策略
 - ◆ 掌握从漏洞发现、评估、优先级排序、修复到验证的完整生命周期管理
 - ◆ 理解自动化补丁分发机制和其在运维中的重要性

第 7 章：知识类：软件安全治理与供应链防护

企业软件安全治理体系的构建与前瞻性思考是企业安全建设的重要内容，它包括如何推动企业安全文化建设与安全策略与标准的制定等内容。通过本部分的学习，学员应当：

- 理解常见的安全治理框架的基本理念和作用
- 理解安全策略、流程与标准的制定的重要性和实现要求
- 理解安全左移理念的重要性
- 了解常见合规性要求对软件产品的影响
- 理解软件安全审计的目的和方法
- 了解软件供应链防护的基本防护策略与方法
- 了解 AI 技术在软件开发过程中的优势与挑战

7.1 知识体：安全治理体系



7.1.1 知识域：安全治理框架与策略

- 知识子域：安全治理框架
 - ◆ 了解 COBIT、NIST CSF 等安全治理框架的基本理念，以及它们在软

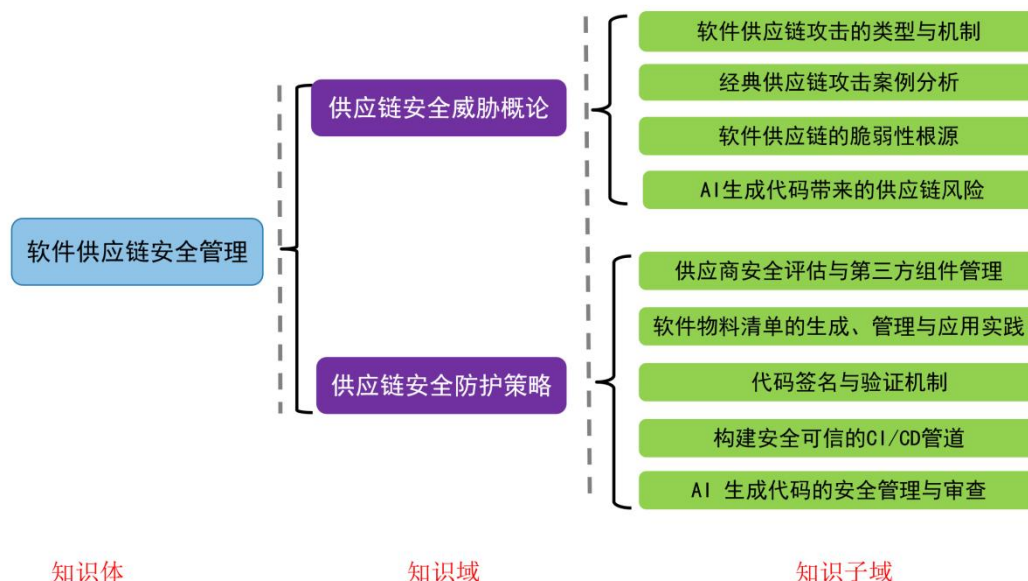
件安全中的应用

- ◆ 理解安全治理在组织层面协调安全资源、制定策略、管理风险的核心作用
- 知识子域：安全策略、流程与标准的制定
 - ◆ 掌握如何制定或理解企业内部的安全开发规范、代码评审流程、安全测试流程等标准
 - ◆ 理解如何将高层安全策略转化为具体的技术实现要求
- 知识子域：安全文化建设
 - ◆ 理解培养全员安全意识，提升安全投入与文化建设的重要性
 - ◆ 掌握和推动“安全左移”理念在组织内部落地的具体方法（如培训、奖励、考核）

7.1.2 知识域：安全合规基础与审计

- 知识子域：常见合规性要求对软件产品的影响
 - ◆ 了解数据安全法、个人信息保护法、GDPR 等国内外常见法规对软件数据处理、隐私保护的通用要求
 - ◆ 理解违反合规性可能带来的业务和法律风险
- 知识子域：软件安全审计
 - ◆ 理解软件安全审计的目的（验证合规性、发现缺陷）、方法（文档审查、技术测试）和范围
 - ◆ 了解如何准备审计材料，并应对外部审计

7.2 知识体：软件供应链安全管理



7.2.1 知识域：供应链安全威胁概论

- 知识子域：软件供应链攻击的类型与机制
 - ◆ 掌握如何识别代码投毒、恶意依赖包、供应商漏洞、分发渠道篡改等常见的供应链攻击手法及其攻击原理
- 知识子域：经典供应链攻击案例分析
 - ◆ 理解并深入分析典型供应链攻击事件，理解其对全球供应链的冲击和深层原因
 - ◆ 了解从案例中汲取经验教训，指导自身防护策略
- 知识子域：软件供应链的脆弱性根源
 - ◆ 理解供应链的复杂性、缺乏透明度、相互依赖带来的固有脆弱性
 - ◆ 了解识别自身软件供应链中的潜在脆弱环节
- 知识子域：AI 生成代码带来的供应链风险
 - ◆ 了解 AI 模型可能引入的已知漏洞、逻辑错误或不符合安全规范的代码片段（如缺乏输入验证、硬编码凭证）的风险
 - ◆ 理解 AI 生成代码可能存在的许可证冲突问题
 - ◆ 理解开发者在提示词中不慎输入敏感数据导致信息泄露的风险

7.2.2 知识域：供应链安全防护策略

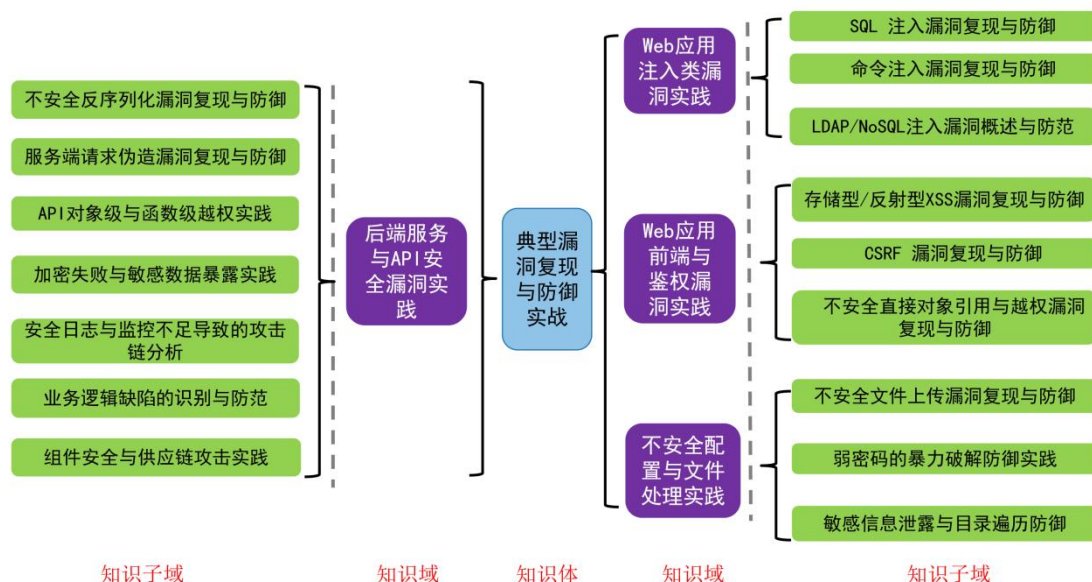
- 知识子域：供应商安全评估与第三方组件管理
 - ◆ 掌握建立对供应商进行安全审查的机制，并实施第三方组件白名单管理和版本控制
- 知识子域：SBOM（软件物料清单）的生成、管理与应用实践
 - ◆ 掌握如何正确管理 SBOM，确保软件组件来源可追溯，并及时发现已知漏洞
 - ◆ 理解 SBOM 在供应链透明度和风险管理中的核心作用
- 知识子域：代码签名与验证机制
 - ◆ 理解通过代码签名确保代码的完整性和来源可信的原理
 - ◆ 掌握如何在 CI/CD 中集成代码签名和验证
- 知识子域：构建安全可信的 CI/CD 管道（Securing the Pipeline）
 - ◆ 掌握实施自动化安全检查（如 SAST/DAST/SCA 集成）、安全配置管理、凭证管理等，防止恶意代码注入构建流程
 - ◆ 理解 DevOps 安全（DevSecOps）的核心理念
- 知识子域：AI 生成代码的安全管理与审查
 - ◆ 了解如何对 AI 生成代码进行初步的人工安全审查
 - ◆ 理解如何利用自动化安全工具对 AI 生成代码进行二次验证，并将其纳入 Code Review 流程
 - ◆ 了解编写安全的提示词的基本原则，以减少引入漏洞或泄露敏感信息的风险

第 8 章：知识类：软件安全开发实践

软件安全开发实践旨在将软件安全开发过程中的所学理论和方法转化为实际操作能力。通过本部分的学习，学员应当：

- 掌握典型 Web 应用漏洞（如 SQL 注入、XSS、CSRF、越权、文件上传等）攻击并能够进行复现与原理分析
- 掌握典型 Web 应用漏洞在代码层面实现相应的防御措施，验证防御效果
- 掌握静态应用安全测试、动态应用安全测试和软件成分分析的基本实践过程和工具的使用
- 掌握交互式应用安全测试、运行时应用自我保护的工具应用
- 掌握 CI/CD 的基本部署流程和工具使用
- 掌握 DevSecOps 中的各个阶段的安全实践方法与工具使用

8.1 知识体：典型漏洞复现与防御实战



8.1.1 知识域：Web 应用注入类漏洞实践

- 知识子域：SQL 注入漏洞复现与防御
 - ◆ 掌握并理解 SQL 注入（如联合查询、报错注入）的攻击过程

- ◆ 了解在代码层面实现参数化查询、ORM 映射，并演示如何避免直接拼接 SQL，验证防御效果
- 知识子域：命令注入漏洞复现与防御
 - ◆ 理解并理解命令注入攻击，获取目标系统信息或执行恶意命令的攻击过程
 - ◆ 掌握在代码层面实现输入过滤、严格白名单校验，并演示如何以最小权限执行外部命令，验证防御效果
- 知识子域：LDAP/NoSQL 注入漏洞概述与防范
 - ◆ 理解 LDAP 和 NoSQL 数据库的注入原理与潜在危害
 - ◆ 掌握针对非关系型数据库的参数化查询或输入校验方法

8.1.2 知识域：Web 应用前端与鉴权漏洞实践

- 知识子域：存储型/反射型 XSS 漏洞复现与防御
 - ◆ 理解存储型和反射型 XSS 攻击原理与危害
 - ◆ 掌握如何在代码层面实现输入验证（黑白名单）、输出编码（HTML 实体编码），并配置内容安全策略 CSP，验证防御效果
- 知识子域：CSRF 漏洞复现与防御
 - ◆ 理解 CSRF 攻击，理解其攻击原理和利用场景
 - ◆ 掌握如何在代码层面实现 Token 机制、配置 SameSite Cookie、进行 Referer 检查，验证防御效果
- 知识子域：不安全直接对象引用（IDOR）与越权漏洞复现与防御
 - ◆ 理解 IDOR 和越权（垂直越权、水平越权）攻击
 - ◆ 掌握如何在代码层面实现基于权限的严格控制，资源访问前强制校验用户权限，验证防御效果

8.1.3 知识域：不安全配置与文件处理实践

- 知识子域：不安全文件上传漏洞复现与防御
 - ◆ 理解不安全文件上传攻击（如上传 Webshell）
 - ◆ 实践：在代码层面实现文件类型校验（MIME Type、文件头）、内

容校验、大小限制，并配置文件隔离存储和限制执行权限，验证防御效果

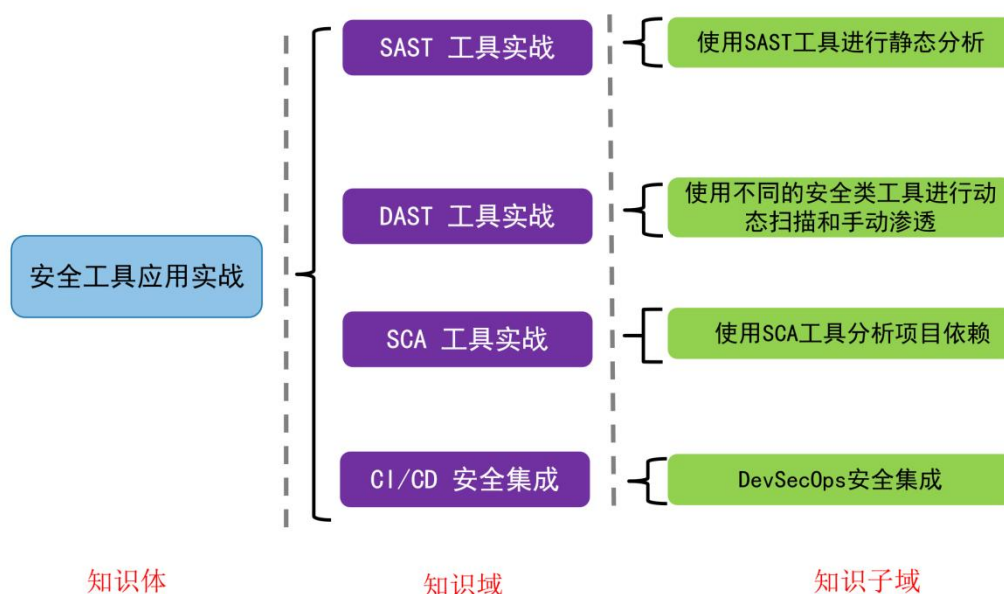
- 知识子域：弱密码的暴力破解防御实践
 - ◆ 理解弱密码攻击和暴力破解
 - ◆ 掌握在应用中强制密码复杂度要求、使用加盐哈希存储密码、实现登录限速和账户锁定策略，并集成多因素认证 MFA，验证防御效果
- 知识子域：敏感信息泄露与目录遍历防御
 - ◆ 理解通过不当配置（如默认配置、冗余文件、目录遍历）导致敏感信息（如配置文件、日志、备份文件）泄露的风险
 - ◆ 掌握如何加固 Web 服务器配置、删除不必要文件、正确配置权限，并避免路径遍历

8.1.4 知识域：后端服务与 API 安全漏洞实践

- 知识子域：不安全的反序列化漏洞复现与防御
 - ◆ 掌握基于 Java/Python 等常见语言的反序列化攻击，理解其原理和潜在的远程代码执行（RCE）危害
 - ◆ 掌握在代码层面实施反序列化黑名单/白名单、使用安全的反序列化库、对序列化数据进行数字签名验证等防御措施
- 知识子域：服务端请求伪造（SSRF）漏洞复现与防御
 - ◆ 理解 SSRF 攻击，理解如何利用 SSRF 访问内网资源、探测端口或绕过防火墙限制
 - ◆ 掌握在代码层面实现 URL 白名单校验、禁用重定向、限制协议与端口、对 URL 解析进行严格控制等防御措施
- 知识子域：API 对象级与函数级越权实践
 - ◆ 掌握 Broken Object Level Authorization (BOLA) 和 Broken Function Level Authorization (BFLA) 等 API 特有越权场景的攻击原理
 - ◆ 理解针对微服务和 API 网关环境下的鉴权与授权设计进行安全加固，确保所有 API 请求均进行严格的细粒度权限校验

- 知识子域：加密失败与敏感数据暴露实践
 - ◆ 掌握弱加密算法、硬编码密钥、敏感数据明文传输或存储导致的数据泄露场景的攻击原理
 - ◆ 理解在代码中实现符合行业标准的强加密算法、安全的密钥管理、强制 HTTPS 配置、敏感数据脱敏和敏感信息过滤等防御措施
- 知识子域：安全日志与监控不足导致的攻击链分析
 - ◆ 掌握通过不完整的日志或缺乏监控，模拟攻击者隐藏行踪，并分析其可能造成的危害
 - ◆ 理解更全面的日志记录（如敏感操作、异常登录、错误信息），并演示如何通过日志分析及时发现攻击。
- 知识子域：业务逻辑缺陷的识别与防范
 - ◆ 理解业务逻辑漏洞的特殊性，例如竞态条件如何导致绕过限制或重复操作
 - ◆ 理解简单的竞态条件场景，并演示防御策略（如加锁、事务处理、唯一性约束）
- 知识子域：组件安全与供应链攻击实践
 - ◆ 掌握已知高危漏洞的常见第三方库进行攻击复现
 - ◆ 理解如何通过升级组件版本、禁用不安全功能或应用补丁来防御此类攻击

8.2 知识体：安全工具应用实战



8.2.1 知识域：SAST 工具实战

- 知识子域：使用 SAST 工具进行静态分析
 - ◆ 掌握独立安装和配置不同的 SAST 工具，如 SonarQube Community Edition 或 Bandit for Python 等
 - ◆ 理解并能够对提供的目标代码库进行静态分析，并学习如何解读分析报告，识别真实漏洞并区分误报，提出代码修复建议

8.2.2 知识域：DAST 工具实战

- 知识子域：使用不同的安全类工具进行动态扫描和手动渗透
 - ◆ 掌握并能够使用不同的工具（如 OWASP ZAP 或 Burp Suite Community Edition 等）对靶场 Web 应用进行动态扫描，并进行基础的手动渗透测试，模拟攻击者行为，发现运行时漏洞

8.2.3 知识域：SCA 工具实战

- 知识子域：使用 SCA 工具分析项目依赖
 - ◆ 掌握如何使用 OWASP Dependency-Check 或类似工具，分析项目依赖
 - ◆ 理解生成 SBOM(软件物料清单)，识别开源组件中的已知漏洞(CVE)，并提出升级或替换建议

8.2.4 知识域：CI/CD 安全集成

- 知识子域：DevSecOps 安全集成
 - ◆ 掌握通过 Jenkins、Gitlab、Github 等产品实现 SAST、IAST、DAST、SCA 流水线过程，并将扫描结果作为构建质量门禁的基本 DevSecOps 流程
 - ◆ 掌握如何使不同类型的安全工具在 CI/CD 流程中应用