360 软件安全开发管理整体解决方案

360 企业安全集团

1 方案概述

软件开发生命周期主要包括需求分析、设计、编码、测试、上线运行维护等阶段。软件安全开发生命周期（SDL)是在软件开发生命周期的各个阶段增加一系列的针对安全的关注和改进，以利于在开发过程中尽可能早的检测并消除安全隐患。

本文描述的软件安全开发管理解决方案主要从人员角色规划、安全开发培训、安全开发过程管理等方面展开，旨在帮助客户建立软件安全开发生命周期(SDL)管理的基本框架，采纳当前本领域中基础性的最佳实践，迈出软件“内建安全”的第一步。

2 解决方案

2.1 人员角色规划

软件安全开发管理涉及软件开发的全生命周期，需要各部门之间的紧密配合，因此在建立安全开发管理流程之前，需要对开发管理的现状进行梳理，确定人员角色和各自责任。

一般情况下，需要设置专门的“安全开发管理组”。“安全开发管理组”负责安全开发管理流程的落地实施，包括流程协调、代码检测、缺陷审计、缺陷修复支持等。人员角色规划示例如下：

角色名称	角色任务定义	部门	工作阶段	备注
安全顾问	● 负责安全开发生命周期管理推进中的咨询、培训、安全方案建议、安全技术支持、安全问题追踪等工作	安全开发管理组	全部	**新增的角色**
代码安全审计员	● 负责对完成开发的代码进行安全检测、审计等工作 ● 负责协助开发人员修复代码安全缺陷	安全开发管理组	开发 测试	**新增的角色**
渗透测试工程师	● 负责对开发完成的系统进行渗透测试	安全开发管理组	测试	**新增的角色**
开发人员	● 负责按照安全编码规范进行编码工作 ● 根据测试反馈，修正程序	开发部	开发	现有的角色
开发经理	● 负责软件开发项目组内的管理和协调工作 ● 同时兼任软件开发人员的角色，负责自己相关模块的软件开发工作 ● 代表项目组与其他部门进行工作对接	开发部	开发 测试	现有的角色
……	……	……	……	……

2.2 安全开发培训

只有软件开发人员具备了安全意识，熟练掌握了安全开发技能，安全开发管理流程才有可能得到真正的落地实施，因此安全开发培训在整个软件安全开发管理流程中的地位尤为重要。在安全开发管理流程建立之初，需要对开发管理者、开发人员进行一次全员培训，帮助其了解安全攻防现状、安全编程基本技能，为后续安全开发管理流程的实施奠定基础。在安全开发管理流程执行过程中，需要根据自身情况和需要，定期开展培训，同时最好建立企业内部的安全开发知识库，供开发人员随时查阅，安全开发知识库应紧跟安全攻防现状，保持安全开发知识

的及时更新。

安全开发培训主要包含软件安全基础培训及安全编程培训。基础培训主要对安全开发生命周期基本理念、软件安全等基础内容进行知识普及。安全编程培训详细讲解安全编程规范、主流安全缺陷机理分析、代码审计、漏洞修复等内容。培训内容示例如下：

项目	主要内容	参与人员
软件安全基础培训	软件安全开发生命周期基础、软件安全基础	技术管理者、开发人员
Java 安全编程	常见 Java 安全漏洞机理分析、漏洞修复、主流安全编程规范（CERT Java）培训	Java 开发人员
C/C++安全编程	常见 C/C++安全漏洞机理分析、漏洞修复、主流安全编程规范（CERT C/C++）培训	C/C++开发人员

2.3 安全开发过程管理

2.3.1 需求分析

任何一个应用软件或者应用系统的核心价值均与其企业使命/业务价值紧密联系。需求分析阶段，应针对来自于国家监管机构、行业的要求，结合具体应用系统所承载的业务，对应用系统进行安全需求分析。

安全需求分析一方面可参考各类合规要求和企业的业务实现，同时也可参考安全行业针对于安全功能的实现方法，如 ISO/IEC 15408（GB/T 18336）的评估方法和功能要求，将业务对应用软件/系统的安全需求以安全功能的形式展现出来。

2.3.2 系统设计

设计阶段，在满足系统的功能、性能需求基础上，应当依据安全设计的一般原则进行系统架构设计；在此阶段，还应根据系统的重要程度和预期应用环境，制定具体的代码安全目标，此安全目标将贯穿于后续的开发过程。

- **安全设计和威胁建模**

安全设计原则应包括权限分离、最小特权、最少公共机制等常见设计原则。对设计好的系统还应进行威胁建模，威胁建模的一般做法是将系统逐层分解，构建数据流图，标识应用程序入口点和信任边界，分析和归类可能存在的安全威胁，提出缓解和降低安全威胁的措施。威胁建模通常可基于微软的 STRIDE 模型进行。

- **安全目标设定**

在此阶段，“安全开发管理组”应根据项目特点，与开发团队共同设定代码安全目标，代码安全目标应包括违禁的安全缺陷列表、代码安全缺陷密度等指标。违禁的安全缺陷是指代码中不允许出现的严重安全缺陷类型，代码安全缺陷密度则通常通过每千行代码的缺陷数来衡量。

2.3.3 软件编码与代码安全检测自动化

- **软件安全编码**

开发人员需要依据安全编码规范进行程序开发工作。目前国际上有一些安全编码标准可做为参考，例如 CERT C/C++/Java 安全编码标准、MISRA C/C++ 安全编码标准等，我国的国家标准目前也正在制定过程中。通常情况下，我们建议结合企业自身实际情况制定适用于本企业的安全编码规范。安全编码规范针对开发人员提供包括基本安全开发要求、规避高危漏洞的编码方案、不推荐的开发函数列表、最佳安全编码实践等在内的通用的安全开发策略；同时还可根据应用

系统的功能需求、行业标准、公司策略、数据敏感级别等具体信息，为应用开发提供具体的安全开发指导。

- **代码安全检测自动化**

基于“360 代码卫士”系列产品搭建统一的安全开发与测试管理平台，实现代码安全检测的自动化，以最小代价与企业原有开发流程进行无缝整合，包括 IDE 整合、代码库整合、构建工具整合、缺陷管理系统整合等，平台同时提供代码安全数据的可视化管理，帮助管理者掌握代码安全整体状况，功能包括目标设定、差距分析、修复追踪、趋势分析等。代码安全检测自动化流程示意图如图 1 所示。

图 1 代码安全检测自动化示意图

一个典型的应用场景是：开发人员在管理平台中配置好任务计划（例如晚上 12 点以后执行检测任务），平台会根据任务计划的设定，自动从 SVN/Git 中获取代码进行检测。开发人员第二天上班时可通过平台查看和审计检测结果，按照高、中、低优先级别进行缺陷修复。平台对代码修复情况会进行自动统计，便于开发人员和测试人员全程跟踪。

“360 代码卫士”系列产品是目前唯一的国产商用级别代码安全产品，其功能和性能已完全具备替代国外产品的能力，且相比国外产品具备更好的灵活性、可定制性以及高质量的原厂商技术支持服务。“360 代码卫士”系列产品目前支持 Windows、Linux、Android、Apple iOS、IBM AIX 等平台上的代码安全检测，支持的编程语言涵盖 C/C++/C#/Objective-C/Java/JSP/JavaScript/PHP/Python/Cobol 等主流语言。在软件代码缺陷检测方面，支持 13 大类，600 多个小类代码安全缺陷的检测，兼容国际 CWE、OWASP Top 10、SANS Top 25 等标准和最佳实践；在软件编码合规检测方面，支持 CERT C/C++/Java 安全编码规范的检测，并可根据用户需求进行灵活定制；在开源代码溯源检测方面，支持 80000 多个开源代码模块识别，28000 多个开源代码漏洞的检测。“360 代码卫士”系列产品能力框架如图 2 所示。

图 2 “360 代码卫士”系列产品能力框架图

2.3.4 上线前测试

在系统上线前，要对全部源代码进行一次安全检测和审计，对软件源代码进行整体的把关，并修复发现的安全缺陷，代码各项指标达到最初设定的安全目标后方可上线发布。同时，采用黑盒漏洞测试手段，对软件系统进行渗透测试，渗透测试是源代码检测和审计的必要补充，渗透测试中发现的安全漏洞也必须在上线发布前修补。

2.3.5 运行维护

运行维护阶段是指系统正式上线运行后的阶段，本阶段的重要工作是建立安全漏洞响应机制，对日常安全事件进行通报并跟踪处理结果。安全响应机制的主要工作内容包括：

- 建立安全漏洞报告和沟通渠道（包括内部和外部），如电子邮箱、官方网站、即时通信工具等；
- 针对报告的漏洞进行严重级别评估，确定应对措施和修补计划；
- 漏洞补丁的开发、测试、发布等；
- 漏洞的分析与经验总结。