ICS 35.xxx
CCS Lxx

团 体 标 准

T/ICS XXXXX—XXXX

软件供应链安全能力评估规范

Security for software supply chain Inspection and evaluation specifications

征求意见稿

XXXX－XX－XX 发布 XXXX－XX－XX 实施

中国互联网协会 发布

目　　次

前　言

本文件按照GB/T 1.1—2020《标准化工作导则　第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件起草单位：中国信息通信研究院、深圳开源互联网安全技术有限公司。

本文件主要起草人：蒋阿芳、樊可欣、曾晨曦、张丽静、菅志刚、张磊、王晓龙等。

软件供应链安全能力评估规范

1 范围

本文件规定了的软件供应链安全能力评估要求。适用于组织机构对软件供应链安全的评估和改进，适用于第三方开展软件供应链安全检测评估认证。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有修改单）适用于本文件。

GB/T 25069—2022 信息安全技术 术语

GB/T 36637—2018 信息安全技术 ICT 供应链安全风险管理指南

GB/T 30279—2020 信息安全技术 网络安全漏洞分类分级指南

GB/T 34943—2020 信息安全技术 C/C++语言源代码漏洞测试规范

GB/T 34944—2020 信息安全技术 Java 语言源代码漏洞测试规范

GB/T 34946—2020 信息安全技术 C#语言源代码漏洞测试规范

3 术语和定义

GB/T 25069 界定的以及下列术语和定义适用于本文件。

3.1

软件供应链 software supply chain

为满足软件供应关系通过资源和过程将需方、供方相互连接的网链结构，可用于将软件产品和服务提供给需方。

3.2

软件供应链安全 software supply chain security

指软件供应链生命周期中各环节、过程涉及的软件产品和服务安全、供应关系安全、人员安全及软件供应链基础设施安全的总和。

3.3

软件供应链生命周期 life cycle of software supply chain

在软件供应链中，从软件的需求分析开始至软件的废止停用或者供需双方终止协议的整个时期，包括开发环节、交付环节和使用环节，划分为协商、生产、交付、获取、使用、运维、废止7个过程。

3.4

开放源代码社区 open source community

开源代码开发、维护的一种组织和运作方式。

3.5

第三方组件 third party component

由供方和需方以外的其他软件开发组织或人员开发的独立可用或可调用的软件模块,通常是由二进制程序文件或者源代码程序文件构成。

4 软件供应链安全体系模型

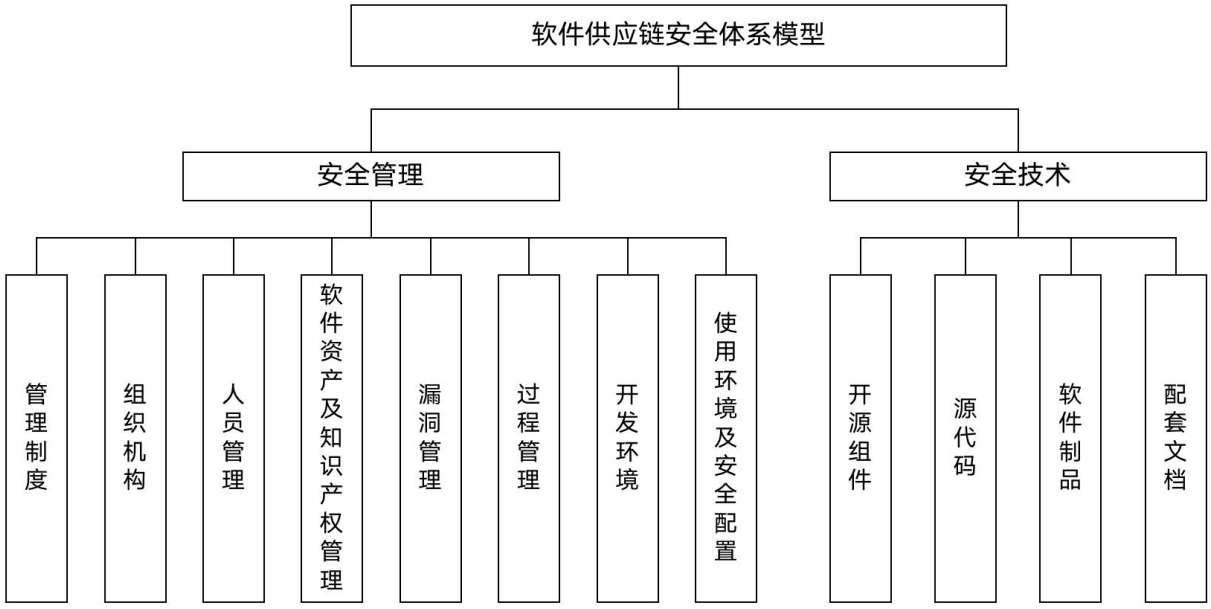

图 1 软件供应链安全体系模型

软件供应链安全体系主要包括安全管理和安全技术两部分。管理包括制度、组织、人员、过程、漏洞、资产、环境几个部分。技术包括源代码、开源组件、软件制品和配套文档几个部分。建立完整的软件供应链安全体系应包括以上内容，结合实际情况持续优化。在体系建设、运行、总结、评审等情况下，应逐条对照进行实施，在体系运行范围内贯彻落实。

5 安全管理要求

5.1 管理制度

本项要求包括：

a) 应制定软件供应链安全的总体方针和制度，明确本单位软件供应链安全基本要求；
b) 应制定软件供应链安全相关的人员管理制度；
c) 应制定软件资产及知识产权的安全管理制度，包括但不限于软件授权证书、专利、软件著作权、许可协议等内容；
d) 应制定软件安全漏洞管理制度，明确安全漏洞风险的防范、响应和处理要求和流程；
e) 应制定软件全生命周期的过程管理制度。

5.2 组织机构

本项要求包括：

a) 应组建软件供应链安全组织机构，明确及其职责范围，对于重要或核心业务场景，例如关键信息基础设施运营单位等组织，应设立专职部门或岗位开展软件供应链安全管理工作；
b) 应制定、实施各项软件供应链安全管理制度；
c) 应制定年度软件供应链安全保障计划，组织实施和监督，并在年底总结；
d) 应申请保障软件供应链安全所需的资源（如有关资金、场地、人力等），并在预算中予以考虑。

5.3 人员管理

本项要求包括：

a) 应划分人员的权限级别，采用最小授权机制，并建立操作规范，创建操作日志；
b) 应开展人员岗前的审查，考核人员的软件供应链安全意识和相关能力；
c) 定期（至少每半年一次）开展软件供应链安全和保密培训；

d） 应建立并执行离职离岗人员账号、权限、材料的交接和清理机制和规程。

5.4 软件资产及知识产权管理

本项要求包括：

a） 应充分掌握软件资产中的开源许可证情况，避免因此导致的法律风险；

b） 应长期维护软件资产库，随时更新软件资产记录，定期（至少半年一次）备份和资产盘点核查。

5.5 漏洞管理

本项要求包括：

a） 应制定漏洞管理机制，明确安全漏洞的防范、响应和处理流程；

b） 应建立内部安全漏洞库，并将权限开放给授权用户；

c） 应对所有软件资产进行检查，将检出的漏洞收录至安全漏洞库，并组织进行整改；

d） 应在软件供应链安全和保密培训时，集中分析讲解重要、典型、有代表性的安全漏洞。

5.6 过程管理（软件全生命周期）

本项要求包括：

a） 应对软件的需求、设计、开发、测试、部署、维护、废止等各阶段进行安全管理；

b） 应在需求分析阶段开展安全需求分析，提出安全需求，形成文档并组织评审；

c） 应在设计阶段考虑到安全性，选型和设计符合安全需求的技术架构，形成设计文档并组织评审；

d） 应在开发阶段执行安全编码规范，符合国标等相关要求；

e） 应在测试阶段开展安全性检测，包括但不限于源代码安全漏洞检测、开源组件安全漏洞检测、应用安全漏洞检测、模糊测试，对于检测出的问题及时修复；

f） 应在部署阶段实行安全配置原则，部署的软件应从受控的安全渠道发布（软件制品库），部署到环境中后，对于高风险或者不必要的端口、服务、数据等资源进行关闭，同时将有条件的安全功能配置打开（例如病毒防护、DDOS防护、RASP、加密、身份认证、访问控制、审计日志等）；

g） 应在维护阶段对软件的安全性进行实时监测，发现安全漏洞或攻击应及时响应修复。

5.7 开发环境

本项要求包括：

a） 应建立独立的开发环境网络，网络与互联网隔离，并配置入侵防范等措施；

b） 应对开发环境网络执行权限访问控制，未授权人员不得访问开发环境；

c） 应定期（至少半年一次）对管理开发环境中的软硬件系统、开发测试工具等进行安全检查，存在安全问题的及时响应处置；

d） 应建立软件资产库，软件资产库至少包括源代码库、开源组件库、软件制品库；

e） 应对软件资产库执行安全机制，如具备防止捆绑恶意代码、下载劫持、网络劫持、升级劫持的能力。

5.8 使用环境及安全配置

本项要求包括：

a） 应定期（至少一个月一次）对使用环境进行安全基线扫描，扫描对象包括但不限于网络设备、网络安全设备、服务器、终端、操作系统、数据库、中间件，并对发现的问题及时进行处置；

b） 应关闭使用环境各软硬件系统中的常见高危端口或服务，例如22端口、23端口、80端口等；

c） 应在重要或核心业务场景的网络中，配置软件系统的实时应用自我防护手段或产品；

d） 应在云环境中的各节点执行一致的安全配置。

6 安全技术要求

6.1 开源组件

本项要求包括：

a） 应选用不含安全漏洞（中危以上级别）的开源组件和开源代码；
b） 应选用不存在许可证冲突的开源组件和开源代码；
c） 应选用有替代品的开源组件和开源代码，防止停更、断供等风险；
d） 应开展软件成分分析，充分掌握软件中包含的开源组件和开源代码，以及它们之间的依赖关系；
e） 应对所有需要使用的开源组件和开源代码进行安全漏洞检测，然后收录到软件资产库集中管理；
f） 应只从内部软件资产库下载和引用开源组件和开源代码，例如Maven引用来源应配置为本地库；
g） 应检查发布的软件制品，对于缺少开源组件许可证信息或者使用开源许可证不规范的情况及时整改。

6.2 源代码

本项要求包括：

a） 应开展代码评审、代码检测、代码走查，识别并修复源代码安全漏洞；
b） 应将发现的源代码安全漏洞收录到内部漏洞库中，并执行后续的漏洞管理机制和流程；
c） 应在编译最终制品时时关闭不必要的选项，例如调试选项。

6.3 软件制品

本项要求包括：

a） 应对软件制品进行配置管理；
b） 应对软件制品进行恶意代码、脚本、病毒蠕虫木马等扫描；
c） 应对软件制品进行完整性检查，防止被恶意篡改；
d） 应对部署在重要或核心业务场景的软件实行数字签名。

6.4 配套文档

本项要求包括：

a） 应为关键供应活动编制文档，包括但不限于合同、保密协议、安全过程评审记录、软件验收单、测试报告、变更申请等；
b） 应在文档中记录时间、相关单位及负责人、主要内容，并加盖公章；
c） 应对关键文档进行管理，控制文档的流转、存储、备份和销毁；
d） 应对涉密文档设置权限，未授权人员不应获得、阅读、修改、复制、销毁。