ICS 35.080

CCS L67

团　　体　　标　　准

T/ISC 0035—2023

软件成分分析（SCA）知识库
总体技术要求

Overall technical requirements for software composition analysis (SCA) knowledge base

（发布稿）

2023-11-15

2023 - 11 - 15 发布　　2023 - 12 - 01 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。

本文件由中国信息产业商会团体标准专业委员会提出并归口。

本文件起草单位：中国信息通信研究院 上海安势信息技术有限公司　墨菲未来科技（北京）有限公司 中国移动通信集团有限公司 中兴通讯股份有限公司 中国电信股份有限公司研究院　中移(苏州)软件技术有限公司 北京火山引擎科技有限公司　紫光展锐（上海）科技有限公司 湖南泛联新安信息科技有限公司 北京安普诺信息技术有限公司 工业和信息化部电子第五研究所 苏州棱镜七彩信息科技有限公司 西安奇科厚德信息科技有限公司

本文件主要起草人：沈滢 王峰 项曙明 王崇萍 张雷 柴思跃 王鑫辉 朱贤曼 张俊霞 李雪 谢竑申昊鑫 陈泳 孙振华 朱中华 覃子桐 庄表伟 陈泳 欧阳强斌 吴荣兵 龙文选 于金泽 胡滨 张涛 王雪松 但吉兵 王媛媛

软件成分分析（SCA）知识库总体技术要求

1 范围

本文件规定了软件成分分析（SCA）知识库系统设计的总体技术要求。

本文件适用于SCA知识库系统的设计、应用和评价。

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（包括所有的修改单）适用于本文件。

ISO/IEC 5230:2020 Information technology — OpenChain Specification

3 术语和定义

下列术语和定义适用于本文件。

3.1

开源项目 open source project

包含开源代码或开源软件的项目。简称“项目”。

3.2

组件 component

开源项目内具有独立的工作逻辑的功能模块。

3.3

软件成分分析 software composition analysis；SCA

通过对软件源码、二进制软件包等的静态分析，发现其所存在的开源合规、已知漏洞等合规性和安全性风险的开源组件应用管理方法。

3.4

知识库 knowledge base；K-base

包含推理规则以及有关某一领域中人类经验和专业知识的信息的数据库

[来源：ISO/IEC 2382:2015，2121399，有修改：删除注]

4 知识库技术要求

4.1 知识库一般要求

SCA知识库应满足如下功能和性能要求：

a）知识库内容全面准确；

b）知识库提供存储、更新与修改能力；

c）具备写入、存储、查询、管理数据的基本功能；

d）具备与主流外围软硬件系统集成和兼容的能力；

e）具备一定的管理能力，包括安装部署能力、配置管理能力及实时监控能力、用户管理能力、在线升级能力、元数据查看以及导入导出能力；

f）具备容错能力，以确保在发生故障时，不会影响到业务的运行，故障包括但不限于硬件故障、操作系统故障、数据库服务故障；

g）具备过载保护能力以及数据多副本能力；

h）具备扩展性，包括集群的在线扩容能力和缩容能力；

i）具备安全性，保证数据在传输和使用过程中的安全，包括对用户进行身份认证的能力、操作审计能力以及加解密能力；

j）具备较高的性能，需要考察写入性能、 查询性能、 数据导入性能及数据压缩能力。

4.2 知识库内容要求

4.2.1 SCA 知识库内容应至少包含源代码库、许可证库、漏洞库和密码算法库。

4.2.2 源代码库内容应满足如下要求：

a）代码来源广度：包含主流开源仓库/平台的开源源代码

b）单个项目的元数据齐全和完整：

1）元数据颗粒度：项目级别、组件级别、文件级别、代码片段级别；

2）元数据完整和准确：

• 项目或社区活跃度：最近更新时间，最近commit数量[1]，contributor数量[2]、star数量[3]、fork数量[4]，issue数量[5]，下载数量，所属开源社区/基金会等；

• 组件级元数据：基本元数据：组件名称、描述、创建时间、版本、许可证、URL、拥有者、程序语言活跃性数据：commit数量、PR数量、Issue数量、Star数量、Fork数量、Contributors数量、下载量、最新发布时间、资源库大小等 ；

• 文件级元数据：文件名称、URL、创建时间、最近更新时间、最近更新人等。

3）元数据关联关系：父子关系、依赖信息、许可证信息、版权信息、额外的许可要求等

c）数据的一致性：原样获取和经过处理的数据均需与来源一致。

4.2.3 许可证库内容应满足如下要求：

a）来源覆盖度：业界主要开源软件的许可证；

b）信息齐全：许可证SPDX简称、许可证全称、许可证原文URL、许可证原文文本、是否OSI认证、是否FSF认证；

c）许可证解读：权利、义务、约束条件；

4.2.4 漏洞库内容应满足如下要求：

a）来源齐全：至少包含NVD、CNNVD、CNVD、CVE和GitHub Advisory漏洞库；（更广泛的漏洞来源参见附录A）

b）漏洞时效性：在漏洞被发现第一时间能收录；

---

1) commit 数量是指代码提交次数
2) contributor 数量是指代码贡献者的数量
3) star 数量是开源项目被用户点击星按钮的次数
4) fork 数量是指开源项目被克隆的次数
5) issue 数量是指开源项目的使用者针对该项目所提出的问题的次数

c) 漏洞关联的准确性：能识别到所有受影响的版本；

d) 漏洞关联的颗粒度：关联到项目级别、组件级别、文件级别、函数级别、代码片段级别；

e) 漏洞分析：至少包含解决建议（修复或规避建议）、漏洞分级、影响范围。（体现更强漏洞分析能力的指标参见附录B)

4.2.5 密码算法库内容应满足如下要求：

a) 来源齐全：是否包含业界各常用密码算法，尤其是非标准密码算法

b) 密码算法特征：针对具体某个密码算法，其特征库是否涵盖各主流编程语言的特征，不因编程语言不同而影响识别；

c) 密码算法和组件关联关系：是否能准确方便地识别到使用密码算法的组件/版本；

d) 能区分标准密码算法和非标准密码算法。

4.3 知识库操作要求

4.3.1 知识库存储应满足如下要求：

a) 以合理的组织方式存储或压缩，尽量减少知识库大小，方便存储、更新和检索；

b) 数据格式方便转换，至少支持json、txt等目标格式。

4.3.2 知识库更新应满足如下要求：

a) 更新能力：出现新的组件和漏洞等知识数据，及时更新；

b) 更新方式：同时支持在线更新（不影响软件正常运行）和离线更新。

4.3.3 知识库修改应满足如下要求：

知识库具备可修改功能，包括不限于提供接口以供用户按照约定的格式进行内容修改与定制化。

附 录 A
（资料性）
常见公开漏洞库

常见公开漏洞库包括（但不限于）：

- NVD (National Vulnerability Database)
- CVE (Common Vulnerabilities and Exposures)
- CNNVD (国家信息安全漏洞库)
- CNVD (国家信息安全漏洞共享平台)
- GitHub Advisory Database
- GitLab Advisory Database
- OSV (Open Source Vulnerability)
- GSD (Global Security Database)

附 录 B
（资料性）
漏洞分析结果字段参考

漏洞分析结果字段通常包括（但不限于）：

- 漏洞原因分析
- 漏洞危害描述
- 漏洞分级
- 漏洞类型
- 影响范围（含影响软件及版本）
- 解决建议（修复或规避建议）
- 可利用的证明代码(PoC)
- 利用条件
- 利用成本
- 利用成熟度
- 处置优先级
- 漏洞可达性说明
- 参考链接