ICS 35.240.40
CCS A 11

团　　体　　标　　准

T/ZFIDA 0004—2024

金融业应用安全测试产品检测能力评估准则 第1部分：静态应用安全测试

Assessment criteria for application security testing products detection capability in financial industry Part 1: Static application security testing

2024-12-18 发布　　2024-12-18 实施

中关村金融科技产业发展联盟　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这些专利的责任。

本文件由中关村金融科技产业发展联盟归口。

本文件起草单位：北京国家金融科技认证中心有限公司、蚂蚁科技集团股份有限公司、中关村金融科技产业发展联盟、中关村互联网金融研究院、中国工商银行股份有限公司、中信银行股份有限公司、北京银行股份有限公司、浙江网商银行股份有限公司、杭州华为云计算技术有限公司、拉卡拉支付股份有限公司、上海蜚语信息科技有限公司、北京酷德啄木鸟信息技术有限公司、北京快手科技有限公司、北京国舜科技股份有限公司、深圳海云安网络安全技术有限公司、苏州棱镜七彩信息科技有限公司、北京车晓科技有限公司、中科聚信信息技术（北京）有限公司、中科软科技股份有限公司、北京海星科技产业服务有限公司

本文件起草人：李振、李博文、冯晓文、王雅仪、余瞰、边立忠、白晓媛、张利歌、李婷、华巍、时绍森、周翔宇、程岩、彭晋、尉郭晨、祖宇飞、李弈龙、陆碧波、曹琳虹、旷亚和、姜城、张然、张心、高佩明、魏辰、沈栋、董镇山、姜冰、束骏亮、吴兴川、杨文博、刘勇、曹亭亭、方蕊、赵浚雅、齐柏尧、杨小强、史明超、落红卫、谷晨、李晨曦、李会丽、丁延彪、范强、陈振、谢源、刘关萍、王俊、王辉、左春、王建林

引　言

静态应用安全测试SAST可以在软件开发生命周期早期发现源代码中的安全漏洞，是企业保障应用安全的重要工具，在金融行业中有着广泛的应用。相比于其他行业，金融行业一方面对网络安全要求高；另一方面系统架构也较为复杂，因此对SAST的能力有着更高的要求和期待。

SAST检测能力是SAST产品的核心能力，可以分成引擎能力和规则能力两大类。由于规则能力主要体现SAST对各类框架和漏洞规则的定制化能力，较难枚举，支持的成本也较低，一般可通过自定义配置实现。SAST引擎能力是SAST面向程序基本语言特性的识别和数据流跟踪的能力，是一款SAST产品的基础能力。本文件主要聚焦SAST引擎能力的评估。

由于不同语言的语言特性会略有不同，本文将主要涵盖Java/C/C++/JS/Go这五种语言的语言特性，其他语言可参考本文自行定义。各个评价项的测试样本可参考xAST开源项目https://github.com/alipay/ant-application-security-testing-benchmark。

金融业应用安全测试产品检测能力评估准则第1部分：静态应用安全测试

1 范围

本文件规定了静态应用安全测试产品检测能力评估框架、检测完整度评估和检测准确度评估等。

本文件适用于金融机构对静态应用安全测试产品的检测能力进行评估，也可为金融业静态应用安全测试产品提供商设计、开发产品时提供参考。

2 规范性引用文件

本文件没有规范性引用文件。

3 术语和定义

下列术语和定义适用于本文件。

3.1

静态应用安全测试 static application security testing

一组旨在分析应用程序源代码、字节码和二进制文件的技术，以找出可能存在安全漏洞的编码和设计条件。

3.2

语言特性 language characteristics

编程语言规范定义的各种特性，如基础数据类型、控制流语句等。

3.3

引擎能力 engine capability

面向程序的语言特性，不和具体的框架和漏洞规则相关，是一款SAST产品的基础能力。

3.4

规则能力 rule capability

对不同的框架、不同漏洞类型规则的支持程度。

3.5

上下文敏感 context-sensitive

分析考虑函数调用的上下文。

注：在上下文敏感分析中，不仅关注函数本身，还会考虑它是在何种上下文中被调用的，从而能够捕捉到不同上下文引起的不同行为。例如，对于递归函数，不同的调用上下文可能导致不同的执行结果。

3.6

流敏感 flow-sensitive

分析考虑程序中的数据流。

注：在流敏感分析中，程序变量的值会根据执行的顺序和流向变化，从而提供更精确的信息。这种分析考虑数据生成、使用和销毁的顺序。

3.7

对象敏感 object-sensitive

分析考虑对象的身份和状态，即不同对象之间的区别。

注：对象敏感分析可以更准确地处理对象之间的引用和状态变化，避免将不同对象的状态混为一谈。

3.8

域敏感 field-sensitive

分析考虑不同类型的变量或数据在不同域中的行为。

注：域可以是类型系统中的类型、内存区域、文件系统中的文件等。域敏感分析可以帮助更精确地理解程序中不同类型数据的流动和交互，从而提高分析的准确性。

3.9

路径敏感 path-sensitive

分析关注程序执行中可能的所有路径。

注：路径敏感分析考虑不同路径导致的不同程序行为，从而可以捕捉到控制流的变化。例如，对于条件语句，路径敏感分析会考虑根据不同输入而导致的不同执行路径。

3.10

抽象语法树 abstract syntax tree

源代码语法结构的一种抽象表示，以树状的形式表现编程语言的语法结构，树上的每个节点都表示源代码中的一种结构。

求解

通过符号执行、类型推导、约束求解等手段来计算和推导程序某个特定属性或状态的过程，从而计算表达式的结果，确定程序的执行路径或状态。

3.11

符号执行

一种程序分析技术，通过使用符号变量而不是实际输入值来模拟程序执行。

注：这种方法允许探索程序的不同路径，生成条件表达式来代表路径约束，并帮助识别潜在的运行时错误。

3.12

类型推导

在不显式声明变量类型的情况下，自动确定程序中表达式或变量的类型。

注：类型推导基于程序上下文中的信息，通过一套规则和算法分析表达式的结构和使用方式，来推断出每个表达式或变量的具体类型。

3.13

约束求解

给定一组约束条件，寻找满足这些条件的解决方案的过程。

注：这些约束可以是线性的、非线性的、布尔的、或更复杂的逻辑公式。

3.14

二阶数据流分析 second order data flow analysis

支持对通过数据库/会话/文件/缓存等存储后，再次异步触发的数据流的分析。

4 缩略语

下列缩略语适用于本文件。

SAST：静态应用安全测试（static application security testing）

AST：抽象语法树（abstract syntax tree）

5 评估框架

5.1 概述

SAST检测能力评估框架如图1所示。

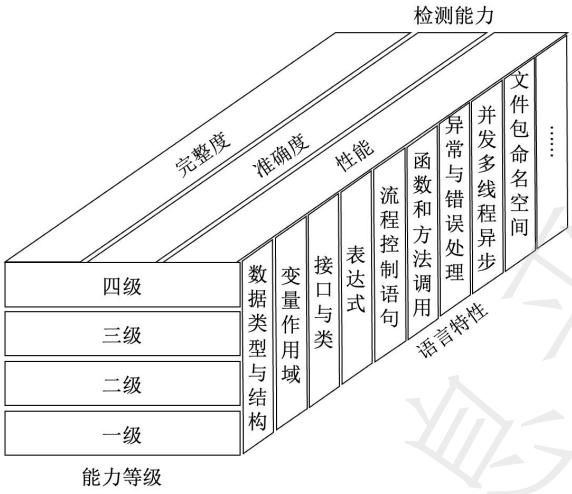

图1 SAST检测能力评估框架

SAST检测能力评估框架由以下三个维度构成：

a) 检测能力维度

检测能力维度给出评估SAST产品检测能力的主要指标，包括完整度、准确度和性能。

b) 语言特性维度

语言特性维度是评估SAST产品检测能力的主要方向，包括数据类型与结构、变量作用域、接口与类、表达式、流程控制语句、函数和方法调用、异常与错误处理、并发多线程异步、文件包命名空间等，不涉及产品功能方面的评价。

c) 能力等级维度

将SAST产品检测能力由低到高划分为四级，一级能力最低，四级能力最高。

5.2 检测能力

5.2.1 完整度

完整度主要用于评估SAST对语言特性的支持程度，体现的是SAST产品的检出能力，检出能力的不足将导致漏报，影响用户对产品使用的信心。检测完整度可以从语言特性识别完整度、单应用数据流跟踪完整度、跨应用跟踪完整度、动态特性完整度和三方包跟踪完整度等方面进行评价，具体包括：

a) 语言特性识别完整度：体现对不同语言特性的识别能力，不涉及数据流跟踪能力。

b) 单应用数据流跟踪完整度：包括污点对象跟踪完整度和污点链路跟踪完整度，其中污点对象跟踪完整度用于评价SAST产品对各类不同类型的污点对象的分析能力，如污点类型为数组、List和字符串等；污点链路跟踪完整度用于评价SAST产品对污点对象在污点传播链路上的数据流分析能力，如污点传播过程涉及for循环、try catch和Lambda表达式等场景。

c) 跨应用跟踪完整度：是一类特殊的污点跟踪，即污点的来源和污点的触发属于不同的服务或应用，用于评价SAST对跨应用场景下污点数据流的分析能力。传统静态程序分析技术只能分析单应用的场景，但跨服务的场景在微服务架构下已经越来越普遍。

d) 动态特性跟踪完整度主要评价SAST对于动态特性的污点数据流分析能力。

e) 三方包跟踪完整度主要评价SAST对于有源码的三方包函数调用场景的污点数据流分析能力，对于无源码的三方包，在技术上通常需要规则建模实现，不纳入引擎能力的评价范畴。

5.2.2 准确度

准确度的高低直接影响SAST产品的实际可落地性。SAST的准确度主要通过流敏感分析、对象敏感分析、路径敏感分析、域敏感分析和上下文敏感分析等五方面的能力得以体现。每一种敏感性都能在特定情况下提高分析的准确性，但同时也可能增加分析的复杂度和计算成本。具体包括：

a) 路径敏感：路径敏感分析能够考虑到程序中不同路径的特定条件。它通过考虑程序执行中可能遇到的条件分支，使得分析能够根据是否满足特定条件来区分和跟踪不同的执行路径。这种分析方式有助于识别那些只在特定路径上发生的问题。

b) 域敏感：域敏感分析能够区分和跟踪对象内部的不同字段。这意味着，当分析程序状态时，分析器能够识别出不同字段的赋值和使用。这种精确度允许分析器更精确地模拟对象的实际使用场景，提高了分析的准确性。

c) 上下文敏感：上下文敏感分析可以理解为分析器能考虑到函数调用的上下文，如一个函数被调用时，可以区分出不同参数或不同调用路径下的调用实例。

d) 对象敏感：对象敏感分析特别关注于程序中的对象实例。在对象敏感分析中，即使两个对象属于同一类型，也会被当作独立的实体来对待，以便更精确地跟踪它们各自的状态变化和相互作用。

e) 流敏感：流敏感分析指的是分析过程能够考虑到程序的执行顺序，通过关注程序执行的具体顺序，来区分不同语句对变量状态的影响，从而增加了分析的精确性。

5.2.3 性能

SAST的性能由于受应用场景、对完整度和准确度的支持程度不一，难以进行客观的评价，本文件将只定义出衡量性能的一些基准指标。

5.3 能力等级

SAST 检测能力根据实现难度和语言特性的流行度，从低到高分成四个等级，其中下一层的能力要求需要包括上一层的所有能力要求。

每一个语言特性评价项包含若干个测试样本,SAST 产品能覆盖该评价项 70%以上的测试样本即可被认为是达到了该语言特性评价项的要求。。

如表 1 所示为 SAST 产品检测能力分级细则。

表1 SAST产品检测能力分级细则

等级	等级描述	适用场景	完整度					准确度				性能
			语言特性识别	单应用数据流	三方包	动态特性	跨应用	区分上下文	区分执行顺序	跟踪粒度	区分执行路径	明确测试基准指标
一级基础级	具备语言特性识别能力 无数据流分析能力	一般应用于编码规范的检测	✓									

等级	等级描述	适用场景	完整度					准确度				性能
			语言特性识别	单应用数据流	三方包	动态特性	跨应用	区分上下文	区分执行顺序	跟踪粒度	区分执行路径	明确测试基准指标
二级标准级	具备单应用程序内的数据流分析能力； 检测精度较低，分析的最小粒度是对象，无法区分执行路径，只能按顺序执行	适用于粗粒度要求的漏洞检测场景	✓	✓				✓	✓（只需满足顺序执行场景）	✓（只需满足对象级）		
三级增强级	在二级基础上，具备对有限场景下反射调用的数据流分析能力； 在有限场景下，支持能区分执行顺序和条件分支、跟踪粒度能达到字段级	适用于对完整度和准确度有较高要求的漏洞检测场景	✓	✓		✓（只需满足有限场景反射）		✓	✓（只需满足顺序+延迟执行场景）	✓（只需满足有限场景中的字段级）	✓（只需满足区分条件分支）	
四级卓越级	在三级基础上，具备跨应用和三方包的数据流分析能力； 检测精度最高，能通过求解技术，在多数场景下，区分执行顺序和路径，跟踪的粒度为字段级	SAST 产品未来的技术标杆，适用于极端复杂的漏洞检测场景	✓	✓	✓	✓	✓	✓	✓	✓	✓	

注 1：测试样本可参考 xAST 开源项 https://github.com/alipay/ant-application-security-testing-benchmark

注 2：✓表示对应等级需要无条件满足的能力项；✓（）表示对应等级有条件满足的能力项，（）中为具体的条件。

6 检测能力完整度评价

6.1 概述

针对具体级别的检测能力完整度评价的评价项包括必选项与可选项。未明确标注的均为必选项，即应满足所属等级下所有评价项的测试样本的70%，才能被判定达到所属等级；标注为可选项的所属等级下的评价项，可视具体场景和需求来选择是否进行评估测试。

6.2 语言特性识别完整度

通过正则表达式或 AST 匹配的方式，具备对语言特性的识别能力。语言特性识别完整度方面的评价项如表 2 所示。

表2 语言特性识别完整度评价项

评价项	评价项说明	涉及的语言	评价项所属等级
注释	支持对注释的识别	Java/Go/JS/C/C++	一级
6.3.1——6.3.10 所述语言特性	支持对 6.3.1——6.3.10 所述语言特性的识别	参照 6.3.1——6.3.10 所述语言特性对应语言	

6.3 单应用跟踪完整度

6.3.1 数据类型和结构

单应用跟踪完整度在数据类型和结构方面的评价项如表3所示。

表3 数据类型和结构评价项

评价项	评价项说明	涉及的语言	评价项所属等级
基础数据类型	支持对整型、浮点型、字符型、布尔型（及其包装类）和枚举型数据的数据流分析	Java/Go/JS/C/C++	二级
数组	支持对数组（含多维数组/切片）类型的数据的数据流分析；数组元素包括多种不同的数据类型，如基础数据类型、对象等	Java/Go/JS/C/C++	
集合	支持对 List/Set/Queue 等数据的数据流分析	Java/Go/JS/C++	
字典	支持对 Map 数据的数据流分析	Java/Go/JS/C++	
结构体	支持对结构体数据的数据流分析	Go/C/C++	
联合体	支持对联合体数据的数据流分析	C/C++	
字符串	支持对 String 的数据流分析	Java/Go/JS/C++	
指针	支持对指针类型的数据流分析	C/C++	

6.3.2 变量作用域

单应用跟踪完整度在变量作用域方面的评价项如表 4 所示。

表 4 变量作用域评价项

评价项	评价项说明	涉及的语言	评价项所属等级
静态变量	支持对 static 变量的数据流分析	Java/Go/JS/C/C++	二级
private 变量	支持对 private 变量的数据流分析	Java/Go/JS/C++	
protected 变量	支持对 protected 变量的数据流分析	Java/C++	
public 变量	支持对 public 变量的数据流分析	Java/Go/JS/C++	

6.3.3 接口与类

单应用跟踪完整度在接口与类方面的评价项如表 5 所示。

表 5 接口与类评价项

评价项	评价项说明	涉及的语言	评价项所属等级
简单对象	支持对单层简单对象及其字段的数据流分析	Java/Go/JS/C++	二级
复杂对象	支持对多层复杂对象及其字段的数据流分析	Java/Go/JS/C++	
子类对象	支持对子类对象及其字段的数据流分析	Java/Go/JS/C++	
接口的实现类	支持对接口的实现类对象的数据流分析	Java/Go/C++	
抽象类的实现类	支持对抽象类的实现类对象的数据流分析	Java/Go/C++	
匿名对象	支持对匿名对象的数据流分析	Java/Go/JS/C++	

6.3.4 表达式

单应用跟踪完整度在表达式方面的评价项如表 6 所示。

表6 表达式评价项

评价项	评价项说明	涉及的语言	评价项所属等级
基础表达式	支持对算术、逻辑、位、关系和赋值运算的数据流分析，包括二元算术运算符（+=）以及自增自减运算	Java/Go/JS/C/C++	二级
Lambda 表达式	支持对 Lambda 表达式的数据流分析	Java/Go/JS/C++	
条件表达式	支持对三元条件表达式的数据流分析	Java/Go/JS/C/C++	
类型转换	支持对类型转换的数据流分析	Java/Go/JS/C/C++	

this 表达式	支持对 this 表达式的数据流分析	Java/Go/JS/C++	
语言特有的表达式	支持对某些语言特有的表达式的数据流分析，如 JavaScript 中的扩展运算符/剩余参数、可选链操作符、空合并运算符、指数运算符、模板字面量和解构赋值表达式等	/	视表达式在特定语言中的语言特性流行度自行确定

6.3.5 流程控制语句

单应用跟踪完整度在流程控制语句方面的评价项如表 7 所示。

表7 流程控制语句评价项

评价项	评价项说明	涉及的语言	评价项所属等级
条件语句	支持对 if/switch 等语句的数据流分析	Java/Go/JS/C/C++	二级
循环结构	支持对 for/while/do-while 等语句的数据流分析	Java/Go/JS/C/C++	二级

6.3.6 函数和方法调用

单应用跟踪完整度在函数和方法调用方面的评价项如表 8 所示。

表8 函数和方法调用评价项

评价项	评价项说明	涉及的语言	评价项所属等级
参数传递	支持对参数传递的数据流分析，包括值传递和引用传递两种方式	Java/Go/JS/C/C++	二级
返回值传递	支持对返回值传递的数据流分析	Java/Go/JS/C/C++	
匿名函数/闭包	支持对匿名函数/闭包的数据流分析	Java/Go/JS/C++	
箭头函数	支持对箭头函数的分析	Java/Go/JS/C/C++	
静态方法	支持对调用静态方法的数据流分析	Java/Go/JS/C/C++	
方法重写	支持对方法重写过程的数据流分析	Java/Go/JS/C++	
方法重载	支持对方法重载过程的数据流分析	Java/C++	
高阶函数	支持对高阶函数的数据流分析	Java/Go/JS/C/C++	
链式调用	支持对函数链式调用的数据流分析	Java/Go/JS/C++	
装饰器函数	支持对装饰器函数的数据流分析	Go/JS	二级（可选项）
Native 方法	支持对调用 Native 方法的数据流分析	Java/Go/C	
内联函数和宏替换	支持对函数和宏替换的数据流分析	Java/Go/JS/C/C++	

生成器函数	支持对生成器函数的数据流分析	JS/C++	
语言特有的函数调用	支持对某些语言特有的函数调用的数据流分析，如指针函数，默认方法，标签函数	/	视函数调用在特定语言中的语言特性流行度自行确定

6.3.7 注解

单应用跟踪完整度在注解方面的评价项如表 9 所示。

表9 注解评价项

评价项	评价项说明	涉及的语言	评价项所属等级
注解	支持对特定语言原生注解的数据流追踪，例如 @Inherited，@Override 等	Java	视注解在特定语言中的语言特性流行度自行确定

6.3.8 异常与错误处理

单应用跟踪完整度在异常与错误处理方面的评价项如表 10 所示。

表10 异常与错误处理评价项

评价项	评价项说明	涉及的语言	评价项所属等级
异常抛出和捕获	支持对 try/catch/finally 的数据流分析	Java/Go/JS/C++	二级（可选项）
断言	支持对 assert 的数据流分析	Java/Go/JS/C/C++	

6.3.9 别名

单应用跟踪完整度在别名方面的评价项如表 11 所示。

表11 别名评价项

评价项	评价项说明	涉及的语言	评价项所属等级
别名	支持对多个引用指向同一块内存时的数据流追踪	Java/Go/JS/C/C++	二级

6.3.10 并发、多线程、异步

对于完整度的评价，这里只考虑对并发、多线程和异步语法结构的支持程度，不涉及具体异步语义的实现，如表 12 所示。

表 12 并发、多线程、异步评价项

评价项	评价项说明	涉及的语言	评价项所属等级
多线程	支持对多线程的数据流分析	Java/Go/C/C++	二级

评价项	评价项说明	涉及的语言	评价项所属等级
多进程	支持对多进程的数据流分析	Java/Go/C/C++	
同步原语	支持对同步原语的数据流分析，包括锁、信号量、条件变量等	Java/Go/JS/C/C++	
延迟执行异步	支持对延迟执行异步的数据流分析，如回调函数/Promise/async/await	Java/Go/JS/C/C++	

6.3.11 文件、包、命名空间

单应用跟踪完整度在文件、包、命名空间方面的评价项如表 13 所示。

表13 文件、包、命名空间评价项

评价项	评价项说明	涉及的语言	评价项所属等级
跨文件	支持对跨文件方法的数据流分析	Java/Go/JS/C/C++	二级
跨包（package）	支持对跨包方法的数据流分析	Java/Go/JS	
跨命名空间	支持对跨命名空间方法的数据流分析	C++	
跨模块	支持对跨模块的数据流分析	Java/Go/JS/C++	
跨编译单元	支持对跨编译单元的数据流分析	C/C++	

6.4 动态特性跟踪完整度

动态特性跟踪完整度评价项如表14所示。

表14 动态特性跟踪完整度评价项

评价项	评价项说明	涉及的语言	评价项所属等级
反射调用	反射调用的方法和属性为字符串常量	Java/Go/JS	三级
	反射调用的方法和属性需要求解运算得到	Java/Go/JS	四级
动态类型推断	支持根据程序上下文动态推断变量类型	/	视动态类型在特定语言中的语言特性流行度自行确定

6.5 三方包跟踪完整度

三方包跟踪完整度评价项如表15所示。

备注：三方包跟踪完整度仅考虑有源码的三方包函数调用场景，对于无源码的三方包，在技术上通常需要规则建模实现，不纳入评价范畴。

表15 三方包跟踪完整度评价项

评价项	评价项说明	涉及的语言	评价项所属等级

三方包库函数	支持对三方包库函数的数据流分析（源码）	Java/Go/JS	三级

6.6 二阶数据流跟踪完整度

二阶数据流跟踪完整度评价项如表16所示。

表16 二阶数据流跟踪完整度评价项

评价项	评价项说明	涉及的语言	评价项所属等级
二阶数据流	支持对二阶数据流的跟踪分析能力	Java/Go/JS/C/C++	四级

6.7 跨应用跟踪完整度

跨应用跟踪完整度评价项如表17所示。

表17 跨应用跟踪完整度评价项

评价项	评价项说明	涉及的语言	评价项所属等级
跨应用数据流	支持对跨应用的数据流分析	Java/Go/JS/C/C++	四级
跨应用二阶数据流	支持对跨应用的二阶数据流分析能力	Java/Go/JS/C/C++	四级（可选项）

7 准确度评价项

7.1 概述

准确度评价项主要包括区分上下文、区分执行顺序、区分执行路径和跟踪粒度等维度。根据 SAST 通常的实现原理，可以将区分上下文对应上下文敏感分析技术，区分执行顺序对应流敏感分析技术，区分执行路径对应路径敏感分析技术，跟踪粒度对应对象敏感和域敏感分析技术。

针对具体级别的检测能力准确度评价的评价项包括必选项与可选项。未明确标注的均为必选项，即应满足所属等级下所有评价项的测试样本的 70%，才能被判定达到所属等级；标注为可选项的所属等级下的评价项，可视具体场景和需求来选择是否进行评估测试。

7.2 流敏感分析

流敏感分析评价项如表18所示。

表18 流敏感分析评价项

评价项	评价项说明	涉及的语言	评价项所属等级
常规顺序执行语句	能够区分程序常规语句的执行顺序（从上至下）	Java/Go/JS/C/C++	二级
循环顺序执行语句	能够区分循环语句结构的执行顺序	Java/Go/JS/C/C++	
延迟执行	能够区分程序语句中延迟执行的语句执行顺序	Java/Go/JS/C/C++	三级

评价项	评价项说明	涉及的语言	评价项所属等级
异步执行	能够区分异步执行场景下语句的实际执行顺序	Java/Go/JS/C/C++	四级

7.3 上下文敏感分析

上下文敏感分析评价项如表19所示。

表19 上下文敏感分析评价项

评价项	评价项说明	涉及的语言	评价项所属等级
参数/返回值传递	能够区分不同函数调用参数/返回值的状态	Java/Go/JS/C/C++	二级
多次调用	能够区分函数多次调用的不同状态	Java/Go/JS/C/C++	
多态	能够区分不同类型对象调用函数的不同状态	Java/Go/JS/C/C++	

7.4 对象敏感与域敏感分析

对象敏感与域敏感分析评价项如表20所示。

表20 对象敏感与域敏感分析评价项

评价项	评价项说明	涉及的语言	评价项所属等级
区分不同的类对象、结构体/联合体和字典/列表/数组	能够区分不同类对象、结构体/联合体、字典/列表/数组的状态	Java/Go/JS/C/C++	二级
	类对象包括子类对象、接口的实现类、抽象类的实现类对象和匿名类等	Java/Go/JS/C/C++	
区分不同类对象的不同字段	能够区分不同类对象不同字段的状态	Java/Go/JS/C++	三级
区分不同结构体的不同字段	能够区分不同结构体的不同字段的状态	Go/C/C++	
区分不同联合体的不同字段	能够区分不同联合体的不同字段的状态	C/C++	
区分一维字典/列表/数组的不同元素	索引值为数字的场景，能够区分不同索引上特定元素的状态（无需求解）	Java/Go/JS/C++	
	索引值非数字的场景，能够区分不同索引上特定元素的状态（需要求解）	Java/Go/JS/C++	四级
	指针偏移量访问的场景，能够区分不同索引上特定元素的状态	C/C++	
区分多维字典/列表/数组的不同元素	能够区分不同索引上特定元素的状态	Java/Go/JS/C++	

7.5 路径敏感分析

路径敏感分析评价项如表21所示。

表21 路径敏感分级评价项

评价项	评价项说明	涉及的语言	评价项所属等级
异常抛出和捕获	能够区分异常抛出对执行路径的影响	Java/Go/JS/C++	三级
条件语句、条件表达式和循环结构	无需通过对不同的条件进行求解，即能够区分不同的执行路径的状态	Java/Go/JS/C/C++	
	能够对上下文条件进行求解，以区分不同执行路径的状态	Java/Go/JS/C/C++	四级
跳转语句	能够区分跳转语句对执行路径的影响，如goto	Java/Go/JS/C/C++	四级（可选项）

8 性能评价

性能评价需要在明确的测试基准环境下进行，测试基准环境由以下测试基准指标确定，即性能测试需要明确给出以下测试基准的指标值。如表 22 所示。

表 22 性能评价指标

测试基准指标	指标说明
服务器 CPU 配置	型号、核数与线程数、主频
服务器内存配置	容量、类型（如 DDR4）、速度
服务器存储配置	硬盘类型（如 SSD）、容量、读写速度
操作系统	操作系统类型、发行版、内核版本
虚拟化软件	如 VMware
容器平台	如 Docker
测试数据集	测试数据集名、代码库链接或代码库文件、代码库大小、代码总行数、开发语言
并发扫描数	
SAST 扫描配置	SAST 产品开启了哪些功能和规则
扫描耗时	单位：秒，同一时间段连续测试 5 次，取耗时中位数

9 评估流程

SAST产品检测能力评估流程具体包括：

a) 确定待测产品和语言；

b) 确定该语言对应的语言特性评价项，并初步选定目标等级；

c) 根据语言特性评价项和目标等级，确定测试样本；

注1：测试样本可参考xAST开源项https://github.com/alipay/ant-application-security-testing-benchmark。

d） 执行测试用例；

e） 统计测试结果，并确定等级，输出SAST产品检测能力评价表。

注2：SAST产品检测能力评价表参加附录A。

附 录 A
（资料性）
SAST 产品检测能力评价表

表A.1给出了SAST产品检测能力评价表示例。

表A.1 SAST产品检测能力评价表

待测产品		测试基准指标		操作系统		测试数据集	
评价语言		服务器 CPU 配置		虚拟化软件		并发扫描数	
测试用例		服务器内存配置		容器平台		SAST 扫描配置	
测试时间		服务器存储配置				扫描耗时	
列出 6.2——7.5 中所选择具体等级需满足的评价项及其评价结果							
评价结论							

参 考 文 献

[1] GB/T 25069—2022 信息安全技术 术语

[2] https://github.com/alipay/ant-application-security-testing-benchmark