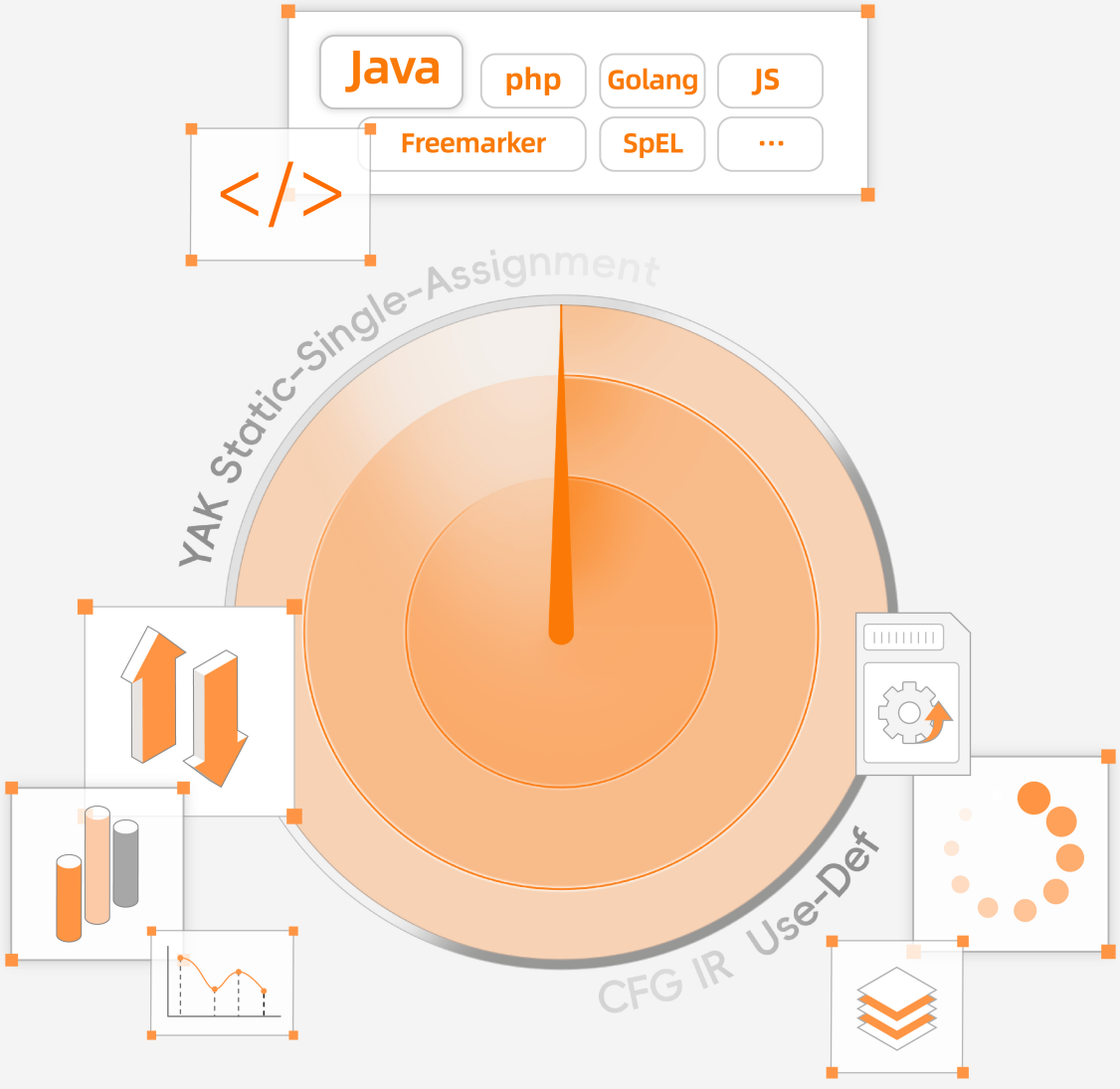
YAK | IRify

IRify 产品白皮书



四维创智（北京）科技发展有限公司

行业现状

随着企业数字化转型的深入，软件开发已经成为企业核心竞争力的关键组成部分。随着敏捷开发和DevOps的广泛应用，软件交付速度不断加快，但这也给代码安全带来了前所未有的挑战。传统的安全测试方法往往在软件开发后期才介入，不仅增加了漏洞修复成本，还可能延误项目进度。

近年来，随着供应链安全事件的频发和国家合规要求的提高，企业对于在开发早期发现并解决安全问题的需求愈发迫切。静态应用安全测试(SAST)技术越来越收到重视，它能够在代码编写阶段就识别潜在的安全漏洞，实现安全能力的"左移"。

静态应用安全测试（Static Application Security Testing，SAST）是一种通过静态分析技术对源代码或编译后的中间代码进行扫描、检测、分析，以发现代码中违反安全编码规则、语法缺陷、运行时缺陷和安全漏洞的技术。SAST工具无需运行程序即可在早期阶段识别安全风险，对提升代码质量和安全性至关重要。

1.1 传统 SAST 工具的痛点

尽管市场上已有多种静态应用安全测试（SAST）产品，安全人员和开发人员对其原理较为熟悉，但大多数工具仍存在多方面的问题。例如，现有产品在解析复杂逻辑结构时表现出不足，尤其在处理嵌套调用和闭包函数时经常出现遗漏或误报。此外，传统工具在跨函数调用的数据流分析方面表现不佳，导致覆盖率不足等等：

- **误报率高：**产生大量误报，使漏洞排查工作负担沉重。
- **结果难以理解：**部分报告缺乏清晰直观的描述，开发人员难以快速定位问题。
- **缺乏整改指引：**未提供直接的修复方案，开发人员需额外花费时间查找并发出整改方案。
- **扫描速度较慢：**在大规模代码库环境中，扫描耗时较长，难以满足敏捷开发需求。
- **标准化问题：**难以判断代码是否符合企业安全编码规范。

随着软件开发流程逐步向S-SDLC（安全软件开发生命周期）、DevOps、DevSecOps 模式演进，企业对 SAST 工具提出了更高要求，包括高性能、高可扩展性、低误报率，以及对多语言、多框架代码的支持能力等。

IRify 产品概述

2.1 产品定位

IRify 是一款领先于传统SAST产品的创新型安全分析平台，其基于静态单赋值形式（SSA）技术，旨在通过构建统一的中间表示（SSA IR）来实现高效的静态分析。模块化架构设计可深度解析复杂代码结构，涵盖多语言场景并提供精细的数据流和控制流分析能力。

IRify 为开发者和企业提供从本地快速检测到企业级自动化集成的全场景安全解决方案。相比传统 SAST 工具，IRify 在以下方面表现出显著的优势：

深度代码分析

采用统一的 SSA 语义模型，全面覆盖复杂控制流和数据流路径，显著提高漏洞检测精度。

误报率更低

通过上下文敏感分析，区分真实漏洞与无害路径，避免不必要的安全提示。

实时反馈与性能优化

提供个人客户端以进行本地快速检测，企业版集成 CI/CD 流水线，实现毫秒级反馈。

灵活规则配置

内置 SyntaxFlow 自定义规则引擎，支持复杂语义查询，适配不同项目需求。

2.2 产品理念

IRify 以“高效检测、精确定位、实时反馈”为核心理念，旨在将安全检测前置到开发流程中，实现实时反馈与持续集成。通过 SyntaxFlow 规则引擎和 SSA 中间表示技术，提供高度自动化的漏洞检测能力，满足 DevSecOps 环境下的高效需求。采用模块化架构，通过 SSA 技术构建统一的语义模型，使代码分析具备高一致性和可扩展性，为不同规模的项目和团队提供定制化检测方案。产品涵盖本地客户端和企业级集成模式，满足多场景需求。

2.3 产品亮点

灵活的本地与企业方案

支持个人用户进行快速扫描，企业用户实现自动化管控。

多语言多框架技术支持

可解析 Java、Python、JavaScript、PHP 等语言，支持其子语言特性，如 Freemarker 模板、EL 表达式等。

先进的分析技术

基于静态单赋值（SSA）形式，实现双向数据流分析、路径敏感的全局分析及数据流与控制流深度结合的上下文敏感过程间分析。

IR 数据库

采用高性能数据库存储结构，支持懒加载与懒存储机制，确保在大规模项目中的高效性能。

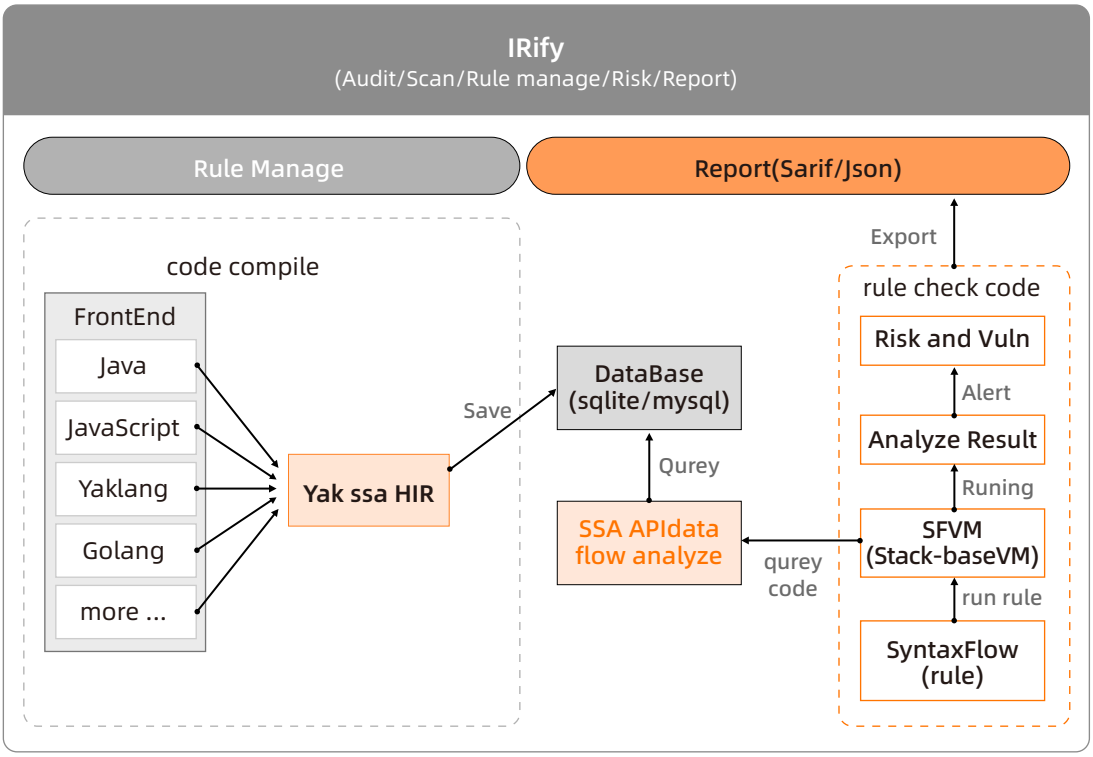
SyntaxFlow 分析引擎

通过专用 DSL 语言直接对中间表示（IR）进行扫描，实现高效规则匹配和漏洞检测。

规则编写系统

内置规则引擎将分析经验转化为可复用规则模板，方便用户根据项目需求快速配置专属检测方案。

技术架构



IRify 的架构设计采用模块化分层方案，以高效的静态分析流程为基础，确保在多种开发环境下保持出色性能和兼容性。架构主要由以下核心流程组成：

编译阶段

IRify 利用基于标准 eBNF g4 语法文件的多语言解析器，将 Java、Golang、PHP、JavaScript 及其模板语言（如 Freemarker、SpEL）等源代码统一转换为抽象语法树（AST），并进一步编译成静态单赋值形式（SSA）中间表示（IR）。在 SSA 中，每个变量和操作节点都拥有全局唯一标识，闭包与面向对象结构的定义域与使用域被精确映射，从而彻底剥离语言差异，为后续的数据流和控制流分析提供了高一致性和高可扩展性的基础。

扫描阶段

系统在 SSA IR 上开展双向、路径敏感的数据流追踪：正向定位从不可信输入到敏感操作的污点传播路径，逆向溯源敏感函数或调用点的潜在风险来源；同时结合控制流图（CFG）与数据流图（DFG）生成全局调用图（CG），确保跨文件、跨包场景下的漏洞路径无遗漏。基于 SyntaxFlow 的规则引擎则负责加载并执行 DSL 规则，对 IR 中的模式进行动态匹配，生成分级告警。所有分析结果均采用高性能 SQLite 数据库存储，通过懒加载与索引优化策略，既支持并行扫描，也能实现亚秒级的数据查询。

用户接口

告警产生后，IRify 会根据严重级别自动分级分类，并通过高性能 SQLite 数据库采用懒加载与索引优化策略存储结果。 IRify 提供了统一的命令行工具（CLI）与 Web 控制台 / REST API 两大入口，开发者可通过 yak ssa 与 yak scan 命令在终端一键完成编译与扫描，也可借助 REST API 将安全分析能 力嵌入自有 DevSecOps 流水线。

Web 控制台以集中式仪表盘展示扫描趋势、告警分布与修复进度，并支持基于角色的访问控制与审计日志。报表展示同样在此层面完成，用户可在线生成并导出 PDF、HTML、Excel 等多种格式的可视化安全报告，满足团队评审与合规审计需求。

核心技术

4.1 基于 SSA（静态单赋值形式）的高效数据流分析

IRify 构建在静态单赋值形式 (SSA) 基础上，每个变量仅赋值一次，从而简化了变量定义和使用之间的关系。通过 Phi 函数在控制流合并点上的运用，它高效解决了数据流在复杂路径上的追踪问题，特别是在处理循环、分支和多路径场景时，提供了精确的数据流建模能力。

4.2 SyntaxFlow 专用 DSL

SyntaxFlow 是 IRify 的核心规则定义语言，专为静态代码分析设计。它支持路径敏感的代码模式匹配和灵活的条件分析，能以声明式方式捕捉复杂的漏洞模式。SyntaxFlow 的规则具有高度的可定制性，使得用户能够根据需求高效地定义和扩展漏洞检测策略。

4.3 对现代高级语言特性的深度支持

IRify 针对高级语言的特性进行了优化，涵盖闭包、继承、多态等复杂语言概念。例如，它能够将闭包中的自由变量捕获为隐式参数，同时对捕获变量的副作用进行分析。此外，IRify 通过原型委托和对象组合，模拟传统 OOP 的行为，以无类系统 (Classless System) 的方式解析复杂对象关系。

4.4 深度数据流与路径分析

IRify 能够执行数据流敏感和路径敏感分析，结合控制流图 (CFG) 和数据流图 (DFG) 建模，追踪变量从源 (source) 到敏感点 (sink) 的完整路径。它支持跨过程和跨模块分析，可以有效发现复杂场景下的隐蔽漏洞，如 SQL 注入和变量污染等。

4.5 集成化的高效分析框架

IRify 提供从 SSA 编译到 SyntaxFlow 规则执行的完整集成工作流。其分析结果可以存储到 SQLite 数据库，并支持导出 DOT 格式的可视化图表，用于数据流展示和代码漏洞定位。工具链设计注重用户体验，便于快速上手和高效协作。

4.6 工程化优化与性能提升

为应对复杂代码和大型项目，IRify 在 SSA 表示中引入了全局唯一的指令 ID，简化指令管理和数据存储。通过定义域辅助系统，IRify 能精确处理闭包、作用域遮蔽等复杂语言特性，结合高效的符号表和索引技术，优化了静态分析的性能与准确性。

产品核心功能

5.1 多语言与框架支持

IRify 支持解析多种主流编程语言及其生态系统，包括 Java 系列（涵盖 Freemarker、SpEL 等子语言）、PHP、JavaScript/EcmaScript 和 Golang 等，兼容其主流版本，并针对 Spring Boot 等常见框架进行了深度优化。

5.2 双向数据流分析

基于静态单赋值形式（SSA）和 Phi 函数，支持自顶向下与自底向上的数据流分析，构建完整的数据流链路，提高漏洞检测覆盖率。

5.3 全局路径敏感分析

实现路径敏感分析，结合数据流图（DFG）与控制流图（CFG），支持跨模块、跨文件的全局分析能力，确保复杂调用关系的完整解析。

5.4 上下文敏感的过程间分析

支持自由变量捕获和闭包分析，追踪异步调用、多层嵌套函数及函数上下文间的依赖关系，确保对复杂逻辑的精准分析。

5.5 多项目并发扫描

支持同时部署多个扫描引擎，启用并发扫描能力，成倍提高大规模项目的检测效率，减少排队时间，特别适合 SaaS 部署的云端检测服务。

5.6 编码合规性检测

提供编码规范检测能力，分析源代码是否符合企业的编码规范要求，帮助避免因编码不规范导致的潜在漏洞，提高代码质量。

5.7 支持第三方库和开源组件扫描

IRify 提供对第三方库和开源组件的扫描能力，能够识别项目依赖中的已知漏洞，提供安全性评分和更新建议，保障项目使用的外部资源安全。

5.8 SyntaxFlow 专用分析引擎

提供专用 DSL 分析语言，可直接对中间表示（IR）进行高效扫描，支持规则模板编写、模糊匹配和方法调用链追踪。

5.9 规则管理与扩展

内置规则库支持用户根据行业经验自定义规则模板，快速适配特定项目需求，将安全检测经验转化为可复用的规则资产。

5.10 报告展示

支持多维度的安全检测报告展示，报告包含漏洞分类、数据流路径、修复建议等信息，支持 PDF、HTML 和 Excel 格式，方便不同团队共享和分析。

应用场景

6.1 S-SDLC 解决方案的核心支撑

S-SDLC（安全软件开发生命周期）涵盖安全培训、需求分析、设计、开发、测试、部署和运维的全流程安全实践。IRify 提供基于客户端与 Web 控制台的双重形态支持，能够灵活适配企业的不同使用场景。无论是本地化部署，还是通过浏览器实现跨平台管理，IRify 均能快速完成源代码和中间代码的静态分析，有效提升开发与测试阶段的安全扫描效率。通过精准的漏洞检测和修复建议，帮助企业在源头上减少安全缺陷，为业务连续性提供全面保障。

6.2 敏捷开发与 DevOps 平台集成（流水线自动化扫描）

IRify 的双形态架构（客户端和 Web 端）使其能够轻松适配敏捷开发和 DevOps 流水线中的多种场景需求。开发团队可通过客户端本地触发扫描，快速检测代码问题；也可通过 Web 控制台集中管理多个流水线任务。

IRify 提供丰富的 API 和插件支持，可无缝集成到 Jenkins、GitLab CI/CD 和 Azure DevOps 等主流平台，实现自动化触发扫描和实时同步检测结果。系统会在代码提交后自动完成静态安全分析，将漏洞报告推送至任务管理工具（如 JIRA）。这种闭环式的自动化能力显著降低了调度成本，提高了 DevOps 流水线的安全可控性。

6.3 开发阶段的代码安全扫描

IRify 为开发阶段提供多种灵活的扫描方式，满足不同团队的需求：

客户端方式	Web 控制台	IDE 集成
开发者可通过客户端工具直接拉取代码仓库（如 GitHub、GitLab）进行扫描，也可上传本地源代码。扫描结果通过本地界面展示，支持细化到具体代码行的修复建议。	项目管理者可通过浏览器访问 Web 控制台，发起团队级的安全扫描任务，实时查看各项目的漏洞趋势分析和修复进展。	持主流开发工具（如 IntelliJ IDEA、Visual Studio Code 等）的插件，开发者可在编程环境中一键触发扫描，扫描结果直接显示在 IDE 界面，便于开发者及时修复问题。

此外，IRify 提供定时扫描功能，适应代码库持续更新的场景；同时支持编码规范检查，确保代码命名、注释和格式的规范性，降低维护成本。

6.4 软件供应链安全审计

随着外部依赖的日益复杂，软件供应链安全已成为企业安全管理的重点。IRify 支持对各种主流语言的项目，包括自研、外包或第三方采购项目进行全面审计。其双形态架构使得审计任务既可通过客户端实现快速检测，也可通过 Web 控制台集中管理供应链安全分析。

IRify 提供了强大的开源依赖库和第三方组件扫描能力，可深入分析 Jar 包等字节码文件中的安全问题，精准识别已知漏洞（如 CVE）和潜在隐患，从源头构建供应链防护体系。此外，对于外包项目，IRify 支持企业定义个性化安全规则，严格保障交付代码的质量和合规性。

6.5 外包项目的安全评估与合规检查

在外包项目验收过程中，企业需特别关注代码的安全性与合规性。IRify 的双形态架构使得外包项目的审计工作更加灵活高效。

客户端模式：企业安全团队可在本地运行独立的扫描任务，对外包代码进行深度安全审计，快速定位问题并生成详细的修复建议，外包人员也可通过客户端进行自测。

Web 模式：项目经理可通过 Web 控制台远程管理外包项目的代码扫描任务，监控进展并生成合规性报告。

IRify 支持自定义检测规则，企业可根据自身的安全需求设置专属编码规范和漏洞排查策略，确保外包团队交付的代码质量符合预期。

6.6 软件安全测评支持

IRify 提供全面的静态代码分析能力，为安全测评机构和企业内审团队提供可靠支持。其客户端和 Web 控制台结合的方式，可高效处理多个测评任务：

客户端：适合小规模项目的快速扫描，直接生成本地化的安全报告。

Web 控制台：支持高并发的分布式任务管理，适应大规模项目的测评需求。

IRify 的静态数据流分析技术能精准定位漏洞发生的代码行，生成详细的调用关系图，帮助测评团队快速完成整改。其灵活的扩展机制可满足测评机构对特殊安全场景的分析需求。

客户价值

7.1 风险预防与代码质量提升

IRify 能够在开发早期阶段高效识别代码中的安全漏洞和潜在风险，避免高成本漏洞进入生产环境，显著降低修复成本和潜在声誉风险。同时，IRify 还通过检测不良编码实践，帮助开发团队优化代码质量，为企业建立稳固的安全基础。

7.2 节约成本与提高效率

IRify 提供高度自动化的静态代码扫描功能，无需运行应用程序即可快速定位漏洞，大幅减少人工审查和分析的时间成本。此外，IRify 可深度集成到 CI/CD 流水线中，实现从代码提交到部署全流程的实时安全检测，避免返工并大幅提升开发效率。

7.3 法规合规性保障

IRify 提供丰富的漏洞报告与修复建议，并支持常见的安全标准（如 OWASP Top 10、CWE、GDPR、PCI DSS、ISO 27001 等），帮助企业快速满足合规性要求，减少因合规失败导致的法律风险。

7.4 开发与安全团队的协作桥梁

IRify 独特的集成式解决方案，结合清晰易懂的报告和详细修复路径，使复杂的安全问题转化为开发人员可操作的任务。同时，与 Yak 平台的无缝集成，进一步促进开发与安全团队之间的高效协作，推动安全能力的团队化和可持续化。

7.5 持续安全保障与智能优化

IRify 借助其内置的强大 AI 模型“万径千机”，能够动态分析新兴安全威胁并提供智能化修复建议。此外，通过与 Yakit 平台的深度融合，企业能够持续监控代码库，建立动态适应性强的安全开发环境。

7.6 现代化开发流程适配

IRify 专为现代开发流程设计，支持多种编程语言和框架，并提供对主流工具链（如 Jenkins、GitHub、GitLab 等）的无缝集成，全面支持 DevSecOps 的落地需求，为企业构建快速迭代的安全保障体系。

7.7 全面安全生态支持

作为 Yakit 平台的重要组成部分，IRify 不仅提供静态分析功能，还与平台上的其他工具（如 Web Fuzzer、MITM 控制面板、Yak Cloud IDE 等）协同工作，为企业提供端到端的安全保障和开发测试能力。

YAK | IRify

010-5945 6626
market@4dogs.cn
北京市海淀区上地街道金隅嘉华大厦F座804
https://www.yaklang.com