

安世加沙龙第二十九期

广州站（城市沙龙）

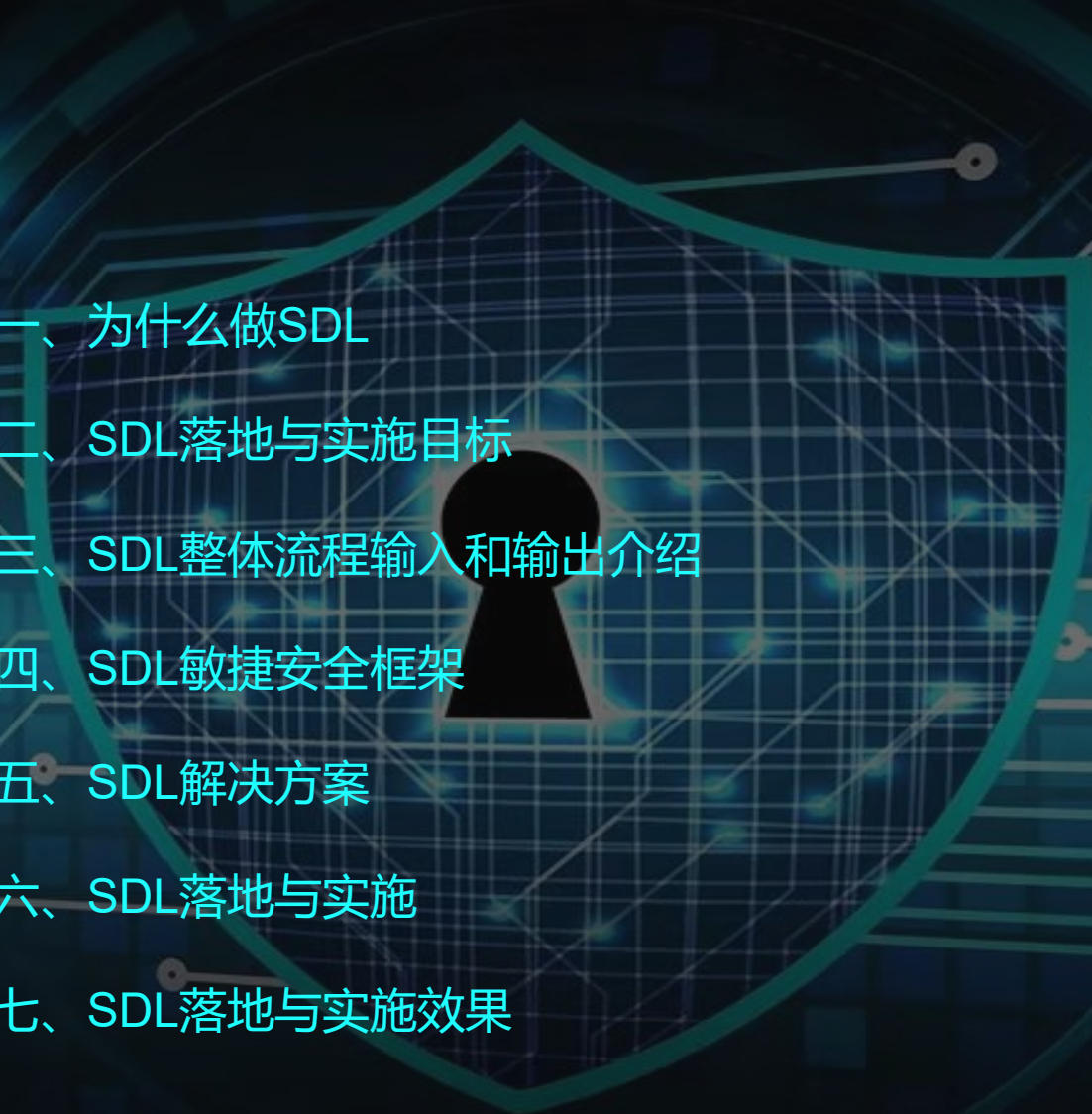
12月10日 / 周五下午

安世加

企业SDL安全落地与实践经验

东风日产乘用车公司-安全项目经理-潘陈粮

目录 Contents

- 
- 一、为什么做SDL
 - 二、SDL落地与实施目标
 - 三、SDL整体流程输入和输出介绍
 - 四、SDL敏捷安全框架
 - 五、SDL解决方案
 - 六、SDL落地与实施
 - 七、SDL落地与实施效果

一、为什么做SDL

企业传统开发场景中（瀑布式开发、敏捷开发）安全要素往往是在事后考虑，大部分的情况下安全人员还是属于安全事后处置，导致企业安全能力无法得到很好的提升。因此比较好的解决办法就是在整个软件开发生命周期流程中加入安全可以让安全更早的介入开发和测试、运维的过程，让安全的措施向前移动，使安全成为整个IT团队（包括开发、运维及安全团队）每个人的责任，安全贯穿从开发到运营整个业务生命周期的各个环节，让人人都能参与安全建设

SDL原则：

需要更多的关注研发流程的“左”边，在更早的环节中（设计、编码、自动测试）也要进行安全介入和管控，例如将源码扫描和开源组件安全检测融入其中，提升SDL各阶段的安全

二、SDL落地与实施目标

标准化 自动化

SDL所有阶段都按照事先制定的安全标准执行，通过SDL平台能实现统一管控，例如研发阶段能实现自动化代码扫描、进入到CICD阶段能自动化代码分支扫描、安全测试阶段实现自动化进行漏洞扫描

闭环 管理

实现动态跟踪威胁的处理流程，从需求提出、威胁发现到安全需求验证，提供全流程漏洞闭环管理

关联 分析

实现SDL平台全流程对接各阶段安全工具，并实现漏洞趋势、部门健康度、产品风险的综合关联分析

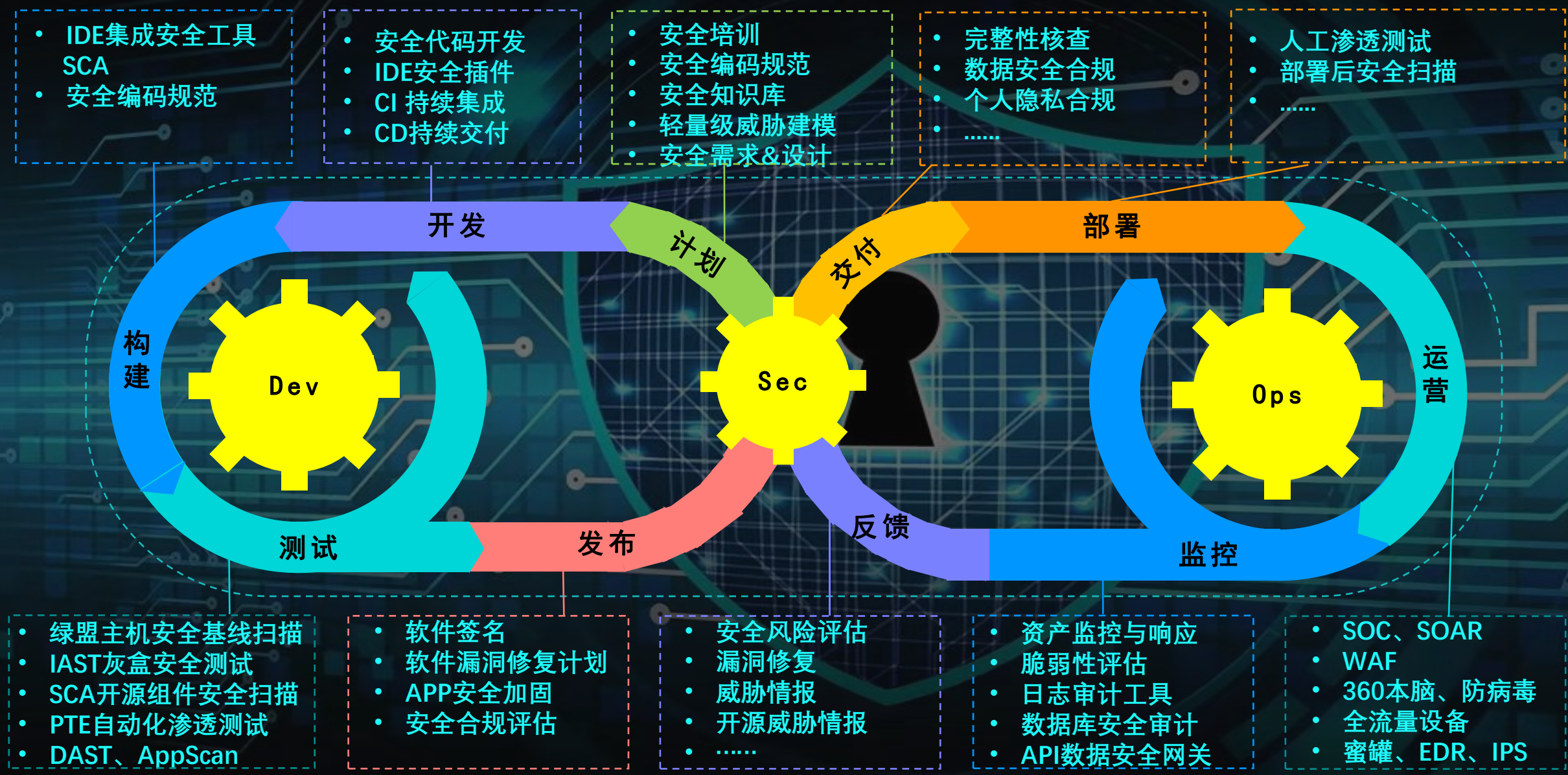
弹性 扩展

实现以漏洞管理为核心，插件化组件设计，根据组织架构的特点弹性调整处置流程，不受平台固定架构和流程约束

三、SDL整体流程输入和输出介绍



四、SDL敏捷安全框架敏捷安全框架（二）



五、SDL解决方案（1/3）

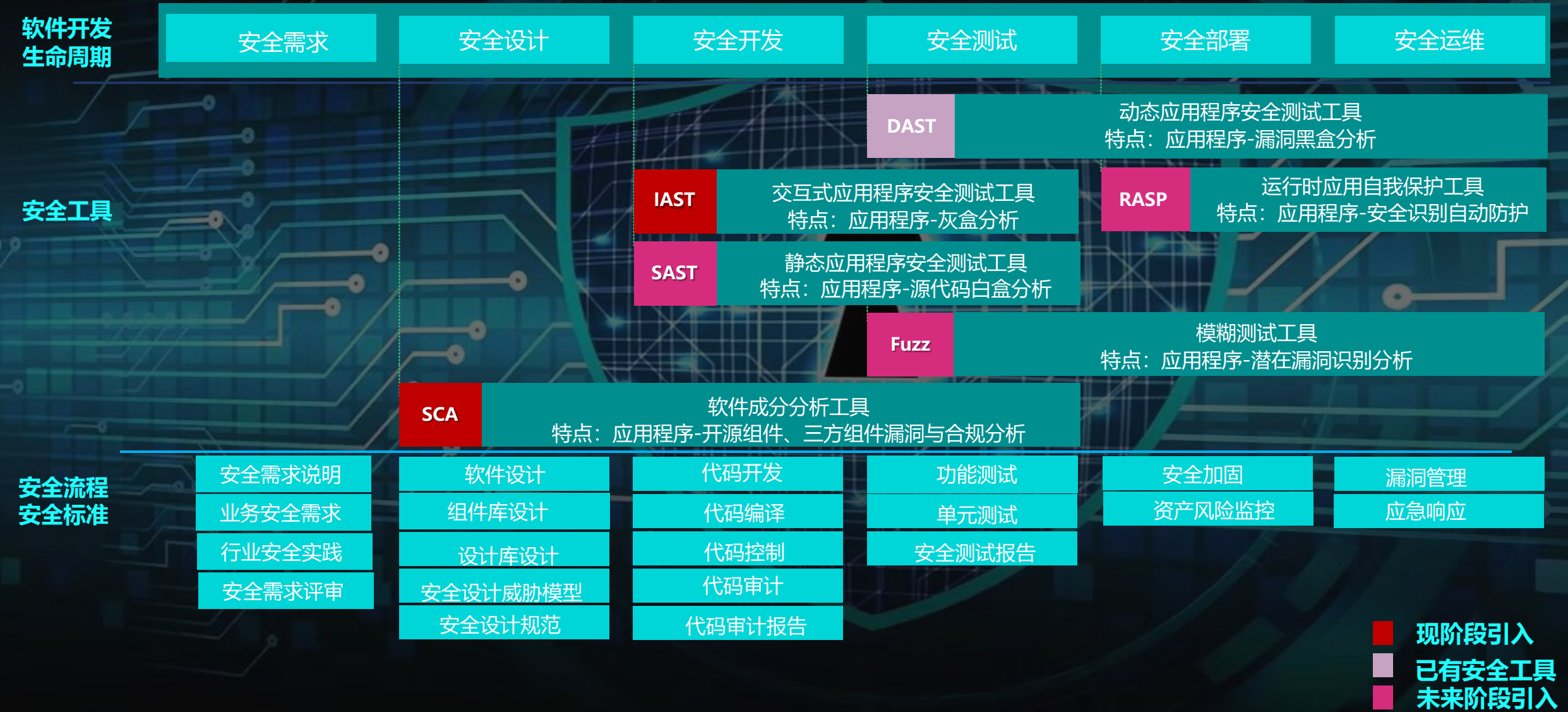
随着东风日产乘用车公司经过初步SDL咨询及体系化建设、流程标准制定与实施，目前企业SDL软件开发生命周期安全建设已彰显效果，有关安全规范流程体系均已逐步开展建立，安全能力得到显著提升

企业SDL流程体系目前尝试通过SDL平台安全评审和管理流程规范制定的方式进行推进。聚合企业内部有相关安全工具（SCA、IAST、PTE、DAST）将企业漏洞风险进行汇总分析，最终实现SDL流程落地

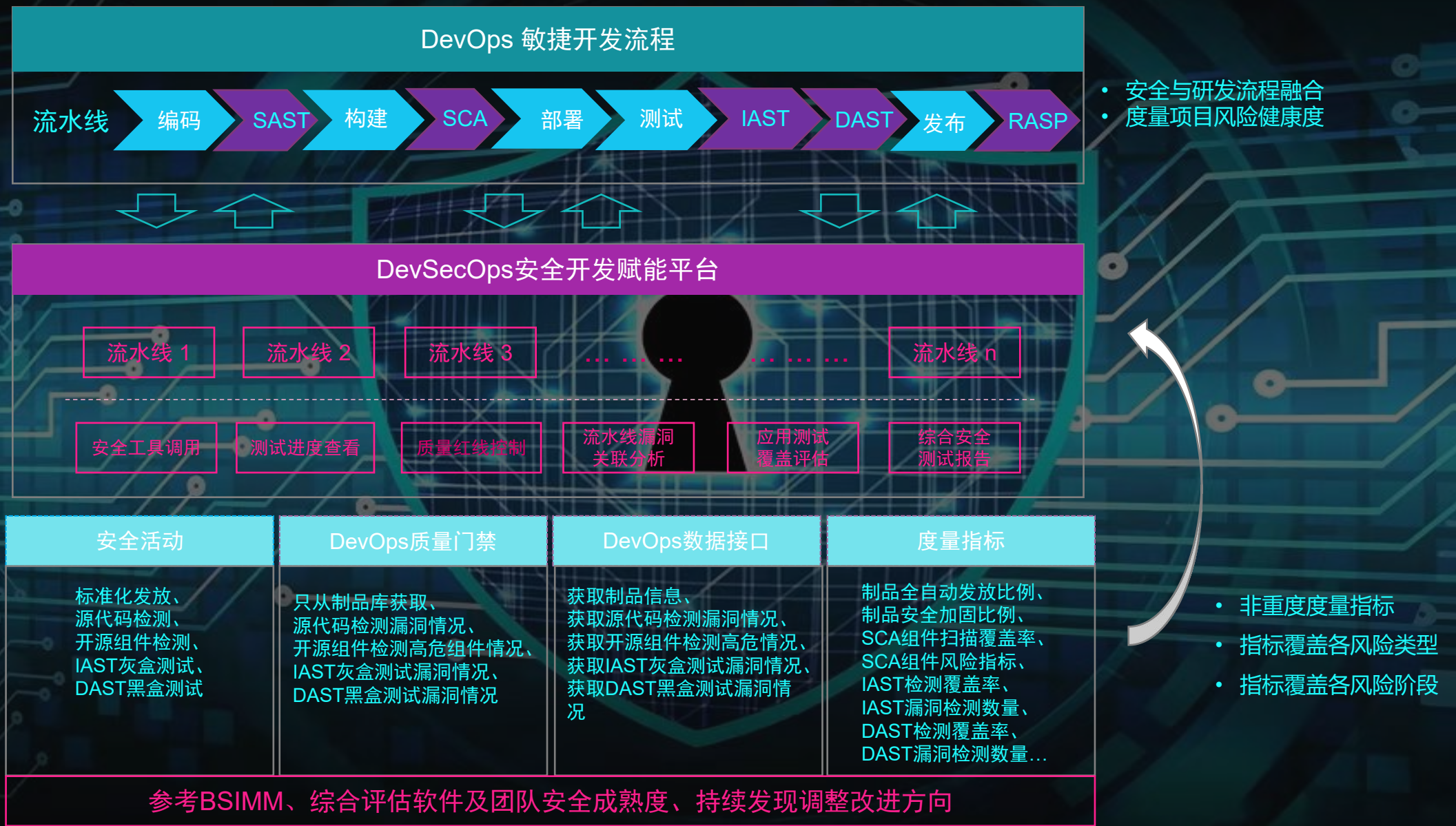


五、SDL解决方案 (2/3)

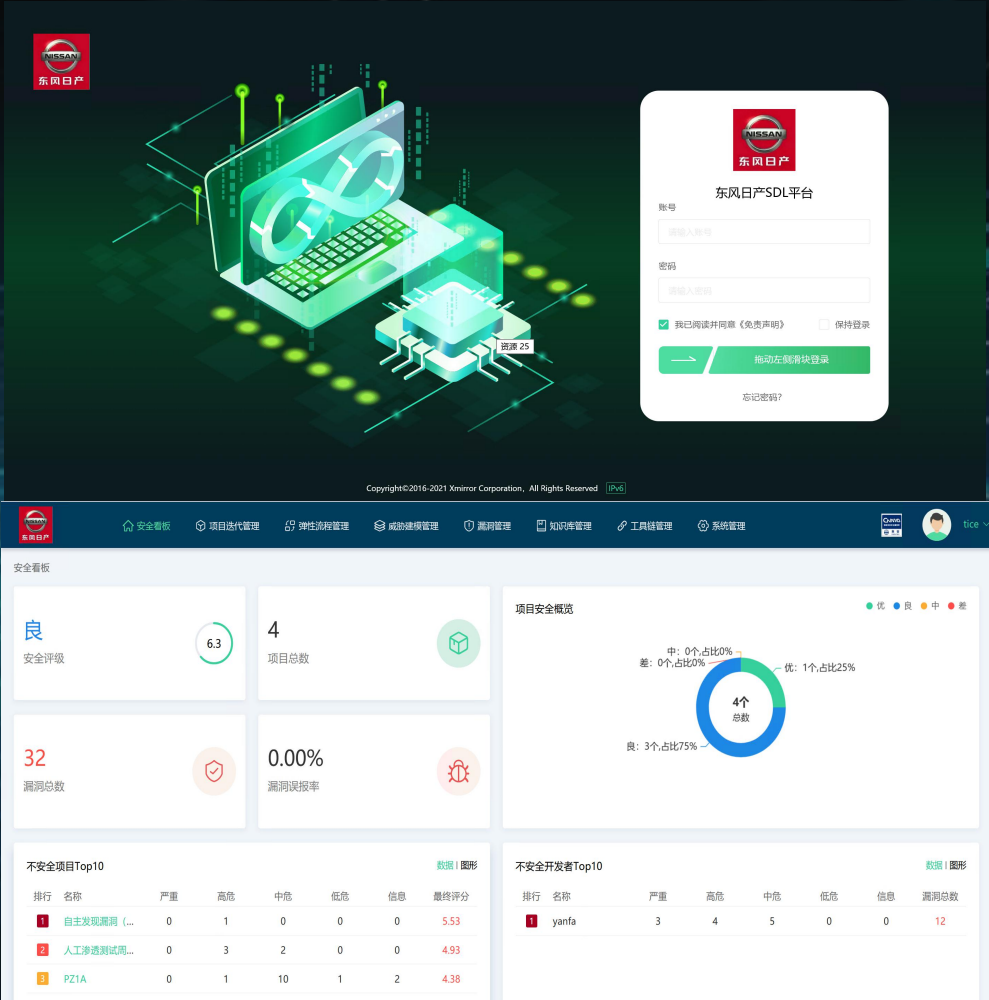
企业内部通过推行SDL安全开发平台建设，梳理SDL/DevSecOps项目管控流程，通过场景化威胁建模、安全需求与设计跟踪、安全工具对接与CI/CD流程更紧密的结合、漏洞关联管理等关键安全活动落地，实现企业内部项目的安全开发管控及度量分析



五、SDL解决方案 (3/3)



六、SDL落地与实施-漏洞风险态势管理（1）



东风日产SDL平台能够清晰展示项目安全开发的安全态势情况，从项目安全状态、漏洞情况、人员安全开发情况、部门安全态势等维度展示安全态势分析趋势，可视化展示企业安全开发状态，为企业安全开发提供整体分析，从侧面提高企业项目团队人员的安全开发意识，提供企业的安全开发能力

六、SDL落地与实施-项目迭代管理（2）

东微自户

安全看板

项目迭代管理

弹性流程管理

威胁建模管理

漏洞管理

知识库管理

工具链管理

系统管理

Comet

1.2.0

用户头像

项目迭代管理 / 项目列表

全部项目

立项阶段

需求阶段

设计阶段

编码阶段

测试阶段

部署阶段

运维阶段

项目列表

添加项目

批量管理

安全态势

未归档项目

项目名称

搜索

<input type="checkbox"/>	项目名称	当前安全态势	项目当前阶段	合规要求	项目周期	创建时间	创建人	操作
<input type="checkbox"/>	人工渗透测试周计划	<div><div></div></div>		App安全评估指南 (...)	2021-11-30 12:00:00-2021-12-...	2021-11-23 16:51:52	f	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	oneapp	<div><div></div></div>	V1 → 需求阶段	App安全评估指南 (...)	2021-11-23 12:00:00-2021-12-...	2021-11-23 16:17:31	f	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	SDC	<div><div></div></div>	V1 → 已完成 V2 → 已完成 V3 → 已完成 V4 → 需求阶段	等保2.0 三级要求	2021-11-23 12:00:00-2021-12-...	2021-11-23 14:32:28	f	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	自主发现漏洞 (合集)	<div><div></div></div>		JR/T 0213—2021 ...	2021-11-01 12:00:00-2021-12-...	2021-11-18 18:12:13	f	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	P21A	<div><div></div></div>	V1 → 已完成 V2 → 已完成 V3 → 已完成 V4 → 已完成 V5 → 设计阶段 V6 → 需求阶段 ...	App安全评估指南 (...)	2021-11-15 12:00:00-2021-11-...	2021-11-15 14:13:19		<div><div></div><div></div><div></div></div>

共 5 条

10条/页

上一页

1

下一页

前往 1 页

立项阶段

立项阶段进行项目的**整体安全评估管理**，从安全合规和安全标准等方面进行安全基线要求评审，将评审结果同步给研发阶段，为研发人员提供安全开发需求

需求阶段

需求阶段进行安全需求分析及评审，通过企业内部安全需求设计Checklist/场景化的调查问卷（未来考虑计划推广）进行分析，进行输出相关安全需求、安全设计、编码规范、安全测试等基线要求，并对相关内容进行评审活动，将相关评审结果同步至后面阶段

设计阶段

设计阶段对相关安全设计内容进行安全评审活动，将安全设计评审结果同步研发人员

编码阶段

编码阶段，关联安全需求产生的**安全编码规范要求**，辅助安全开发，同时辅以**代码安全检测**，利用Gitlab代码版本管理工具进行集成代码检测工具及软件成分分析工具-SCA自动化或人工分析代码安全风险

测试阶段

测试阶段为测试人员提供**安全测试要求**，并验证安全需求实现情况；同时在项目开发中辅助使用**安全测试工具**，利用灰盒-IAST和黑盒测试工具-PTE 进行项目安全性检测

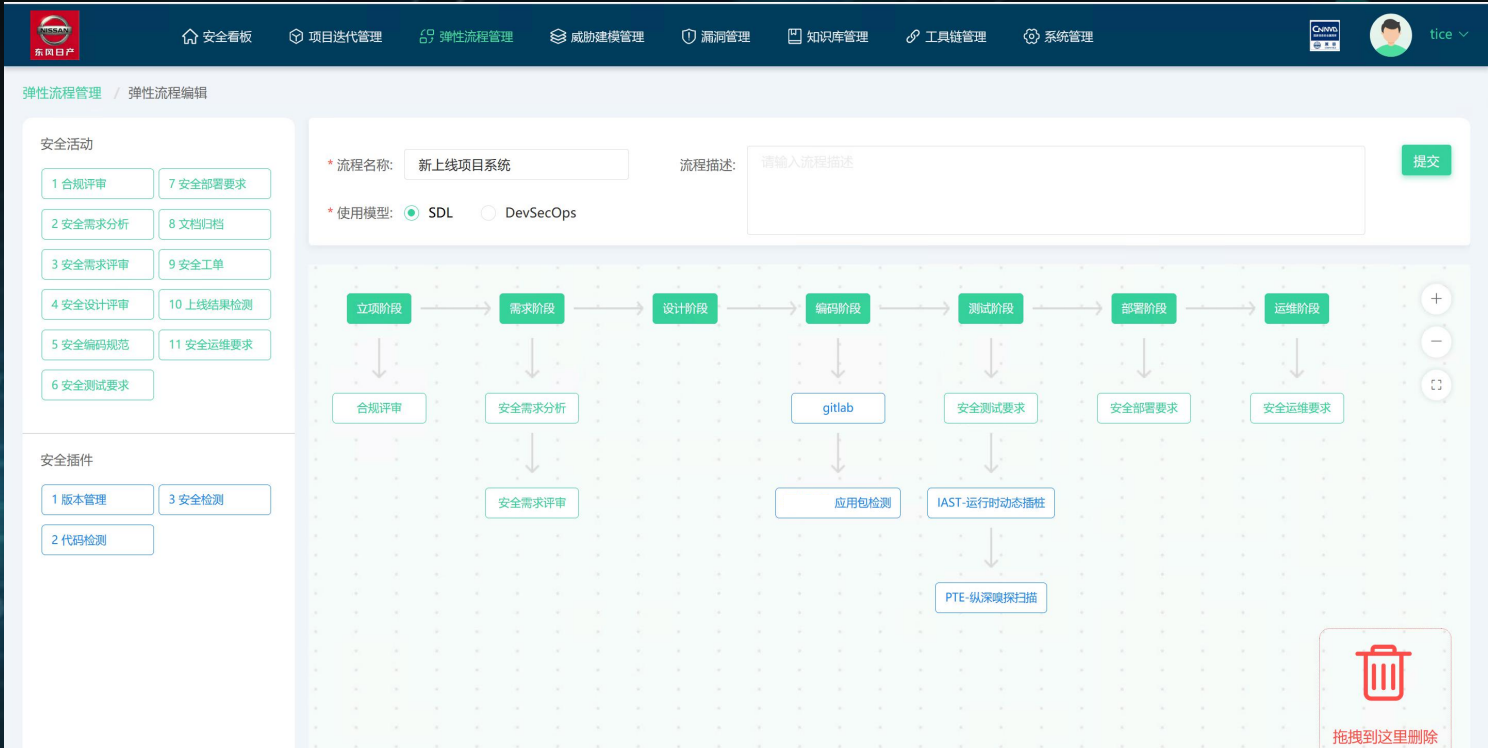
部署阶段

部署阶段，为项目提供**部署安全要求**，实现系统上线前通过安全规范要求与基础环境要求

运维阶段

运维阶段，为项目提供安全工单功能和**安全检测活动**，实现安全工具无缝嵌入项目安全运维流程中

六、SDL落地与实施-弹性流程管理（3）



- 安全负责人根据企业开发模式下的安全活动情况进行自定义编排，将安全检测工具插件化，集成在平台中
- SDL平台能够提供传统的SDL与DevSecOps的安全开发模式
- 在SDL模式下，人工自定义主要的安全活动与安全流程卡点
- 在DevSecOps模式下，根据DevSecOps流水线进行设置安全工具插件化的自定义编排，根据安全质量红线设置进行控制开发流水线

六、SDL落地与实施-安全需求分析（4）

通过情景式问卷的形式，以知识库基线内容进行自动化关联分析，形成安全需求、安全设计、安全测试、编码规范等内容，支持按照模板进行导出，为安全开发提供相关要求依据

全部项目

立项阶段

需求阶段

设计阶段

编码阶段

测试阶段

部署阶段

运维阶段

情景式分析

根据项目的初始需求业务场景进行情景式问题分析,获取项目的安全需求、安全设计、安全编码规范、安全测试要求等内容

关联场景 (请选择需要关联的场景)

请选择重点功能安全场景

动态口令令牌 个人信息维护 密码重置 (忘记密码) 客户信息采集 II、III类账户注册 注册 二维码/条码支付 互联网渠道端交易场景通用安全需求 其他机制 生物特征 文件证书 文件上传 文件下载 短信 图形验证码 手势密码 手机指纹密码 声纹 手机人脸识别 智能密码钥匙

请选择通用要求场景

DFN-应用系统分级安全 DFN-API接口安全要求 DFN-个人信息收集评估 DFN-APP安全开发要求 身份认证 会话安全 访问控制 输入校验 数据安全 安全审计 系统容错 应用部署 加密算法

请选择客户端安全场景

移动客户端基本要求 微信订阅号 微信服务号 微信企业号 微信小程序

请选择网络与通信安全场景

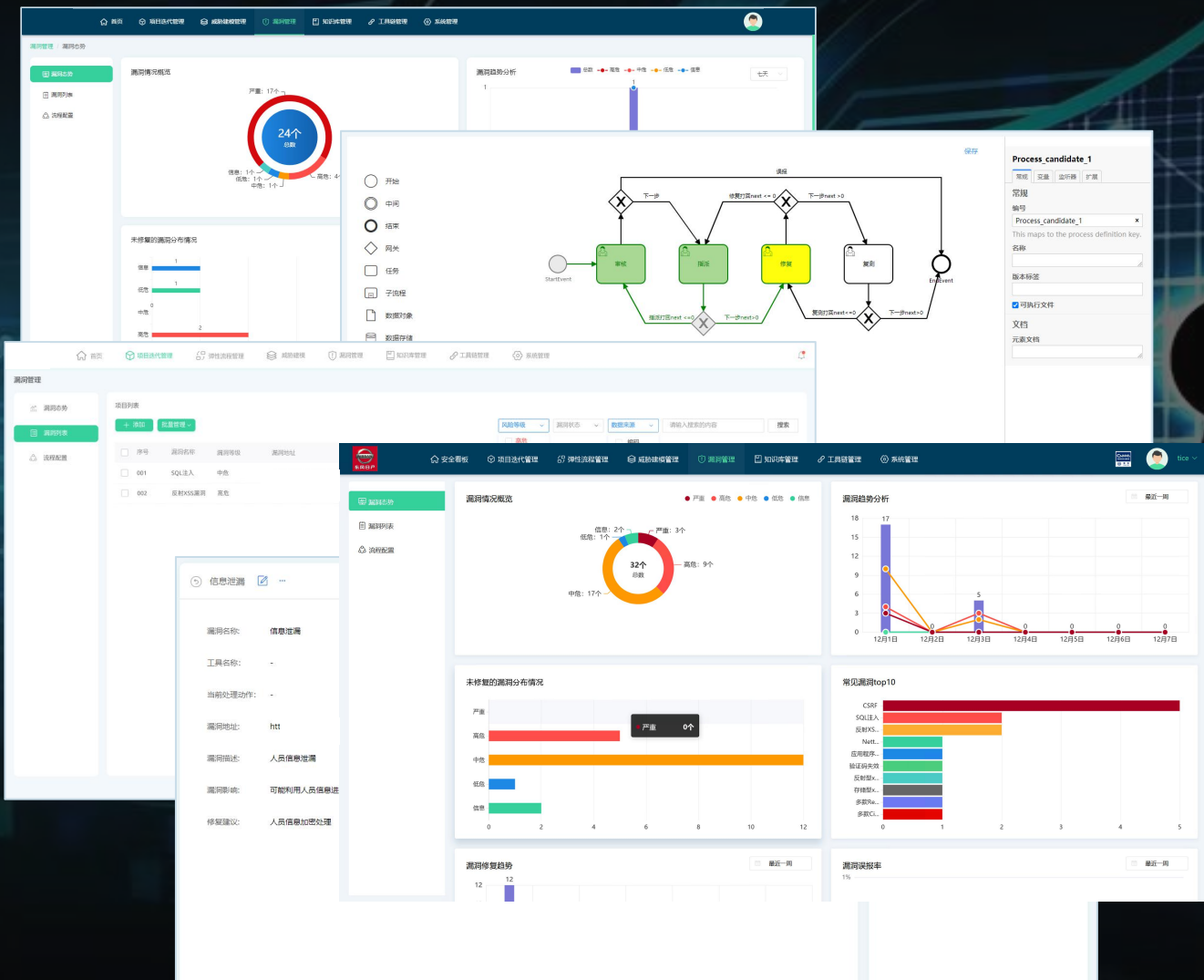
第三方对接 互联网访问 内部访问或系统对接 通讯协议 安全认证 通信链路

请选择服务器端安全场景

安全通信网络 等级保护要求 安全计算环境 虚拟化安全

```
graph LR; A([业务场景]) --> B([安全威胁]); B --> C([安全需求]); B --> D([安全设计]); B --> E([安全编码]); B --> F([安全测试]);
```


六、SDL落地与实施-漏洞闭环管理（5）



项目的安全漏洞进行闭环管理：

实现可视化展示企业漏洞整体情况，展示漏洞检测情况和人员漏洞修复情况，同时展示企业漏洞的分布、趋势等情况，从漏洞维度体现安全开发现状

集成企业内部工具检测结果同步漏洞管理模块，统一进行漏洞管理；

外部漏洞录入进行漏洞生命周期管理，如渗透测试结果录入、自主发现漏洞、EDR发现漏洞

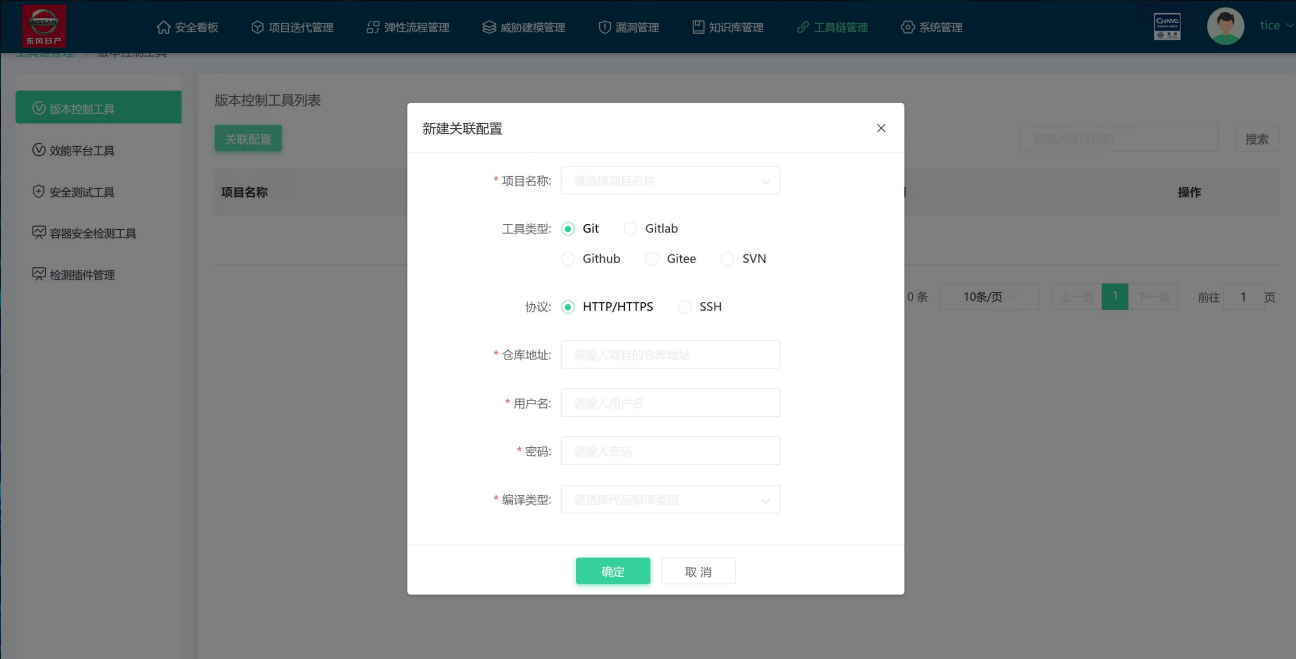
实现企业自定义配置漏洞闭环流程，漏洞处置可以根据设置的流程进行漏洞闭环管理，同时详细记录漏洞的每个状态变更情况，实现全流程跟踪

六、SDL落地与实施-工具链聚合管理（6）



PTE自动化渗透测试工具
软件成分分析工具
SCA

灰盒测试工具
IAST



SDL平台集成软件成分分析工具SCA，黑/灰盒安全检测工具链，同时集成版本控制工具如Gitlab，将相关工具集成后，在项目安全开发过程中使用安全检测工具实现自动化安全漏洞检测发现，同时将安全漏洞检测结果实时同步到漏洞管理模块，相关角色（业务部门、研发部门、测试部门、安全部门有关人员）进行安全检测结果处理（例如审核与卡点限制、指派）

集成Jenkins代码发布构建工具，在不同的阶段使用相关功能，例如在开发代码构建阶段能够自动化分析检测组件安全漏洞并及时阻断代码构建，实现安全开发流程开发配置管理

六、SDL落地与实施-知识库沉淀管理（7）

通过SDL平台建立企业内部合规要求知识库、安全威胁库、安全需求库、安全设计库、编码规范库、安全测试库、开源组件库、代码示例库等内容进行管理

知识库管理

业务场景库

安全威胁库

安全需求库

安全设计库

编码规范库

安全测试库

开源组件库

代码示例库

业务场景列表

列表

批量管理

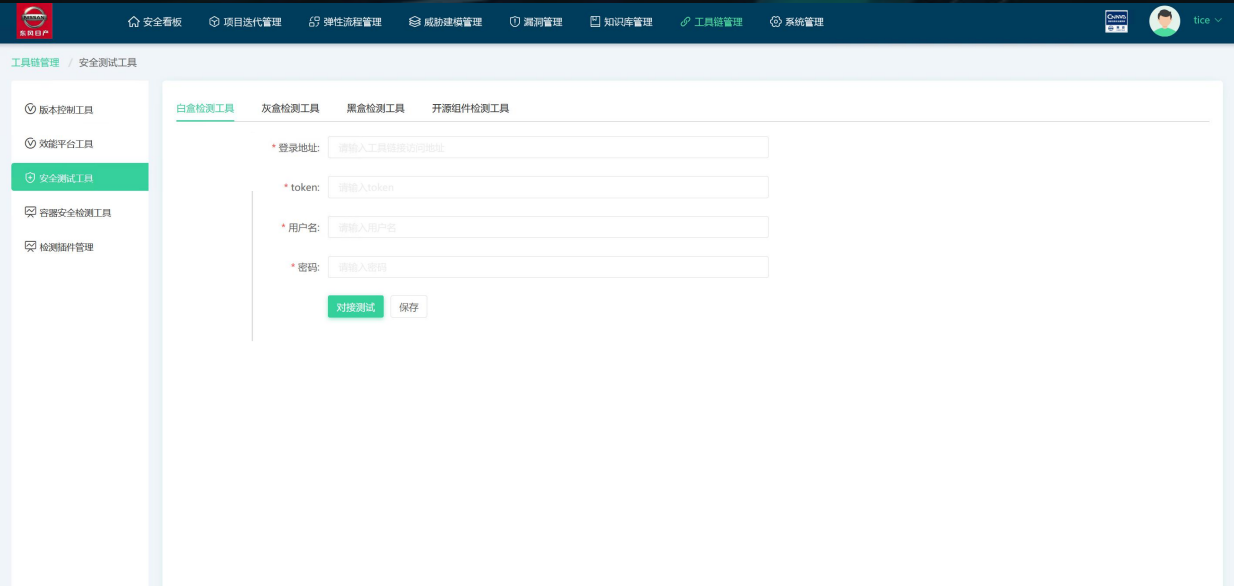
请输入关键词

搜索

编号	名称	描述	类型	标签	关联威胁	更新时间	操作
SBS-2021-000059	DFN-应用系统分级安全	DFN-应用系统分级安...	自定义	通用需求场景	系统上线阶段（S标准），安全测试阶段（S...	2021-12-02 16:27:29	<div>编辑</div> <div>删除</div>
SBS-2021-000061	DFN-API接口安全要求	DFN-web-API接口...	自定义	通用需求场景	身份验证	2021-12-02 16:04:22	<div>编辑</div> <div>删除</div>
SBS-2021-000062	DFN-个人信息收集评估	DFN-个人信息收集评...	自定义	通用需求场景	-	2021-12-02 16:03:55	<div>编辑</div> <div>删除</div>
SBS-2021-000060	DFN-移动APP安全开要求	DFN-移动APP安全开...	自定义	通用需求场景	身份验证	2021-12-02 16:03:47	<div>编辑</div> <div>删除</div>
SBS-2021-000037	II、III类账户注册	II、III类账户注册	系统提供	重点功能安全场景	账户认证要求、账户认证要求1、II、III类户交...	2021-07-02 15:58:18	<div>编辑</div>
SBS-2021-000042	安全通信网络	安全通信网络	系统提供	服务保障安全场景	安全通信网络搭建要求、安全通信网络基本要求	2021-07-02 15:58:18	<div>编辑</div>
SBS-2021-000038	客户信息采集	客户信息采集	系统提供	重点功能安全场景	防止垃圾数据、手机号码核实采集、采集信息输...	2021-07-02 15:58:18	<div>编辑</div>
SBS-2021-000039	密码重置（忘记密码）	密码重置（忘记密码）	系统提供	重点功能安全场景	密码重置功能限制重置他人登录密码、密码重...	2021-07-02 15:58:18	<div>编辑</div>
SBS-2021-000040	个人信息维护	个人信息维护	系统提供	重点功能安全场景	手机号码修改、存储型SQL注入...	2021-07-02 15:58:18	<div>编辑</div>



六、SDL落地与实施-难点问题解决（8）



为什么不建议企业在软件开发生命周期最初阶段采用安全左移-SAST工具

1、SAST工具落地实施阻力大。由于SAST工具在进行扫描代码时会产生大量的误报,大量误报的代码漏洞风险如果直接推送给业务方往往会造成业务方的忽视（可能由于项目进度等原因），这也将影响安全部门的权威性，因此我们认为：对于SAST工具在落地实施中的安全风险推送宁愿存在漏报也不要一开始就大量误报的情况产生

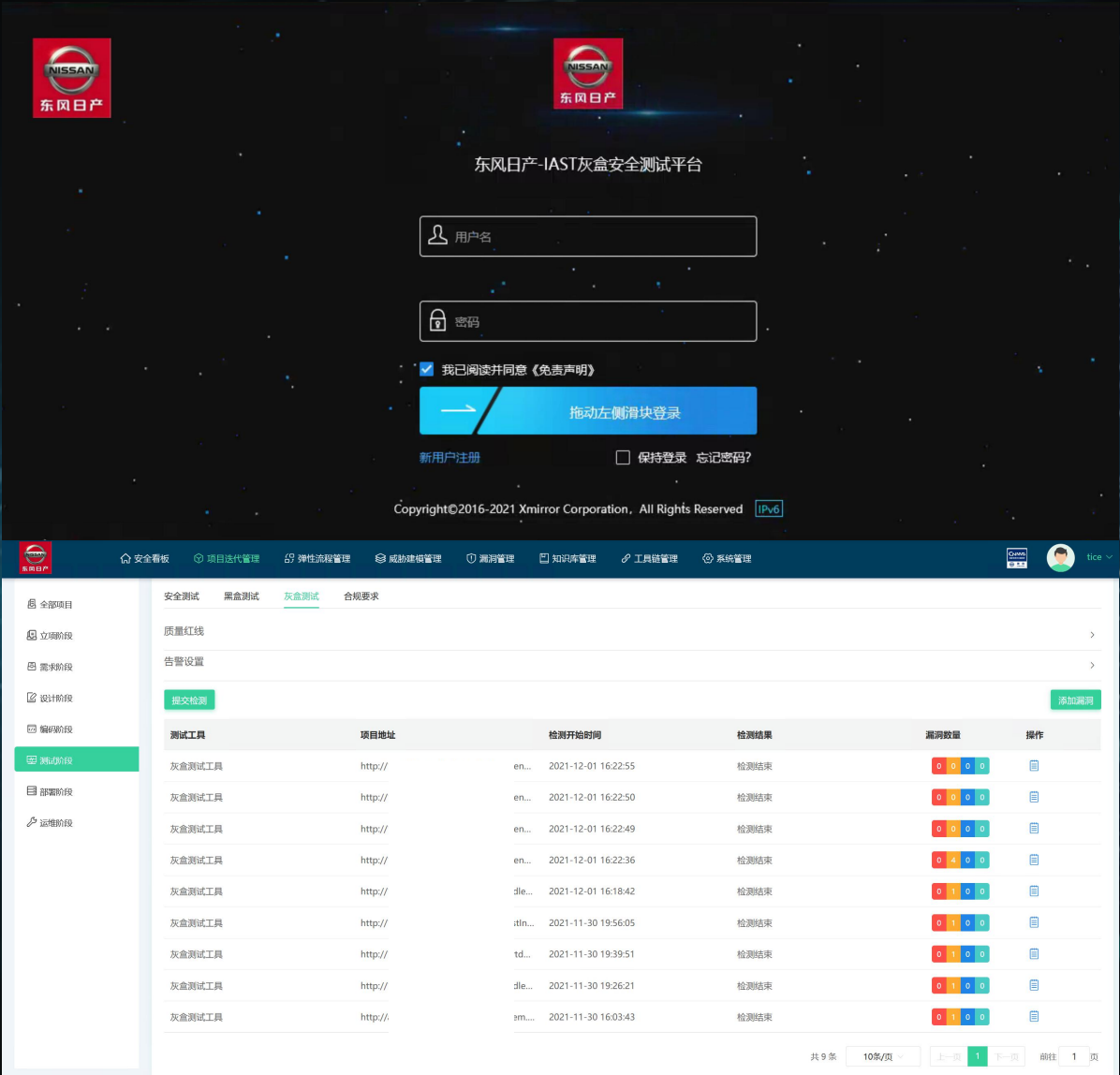
2、SAST工具对人员能力技术有一定要求。开发人员、安全人员需要对SAST工具中的语法语义较熟悉，同时还需要具备能力对SAST工具进行灵活的代码检测规则编写

SAST工具未来如何落地？

1、企业内部应制定统一的代码编写安全规范，例如详细列出有关错误API、高危函数以及安全操作方式，开发人员则尽可能遵循实施

2、SAST工具建议可以在以下开发编码的不同阶段引入。例如本地代码编写阶段：通过SAST-IDE插件实现自动扫描、自动修复错误功能代码。
CI/CD持续集成部署阶段：通过设置质量红线，开发人员在代码构建过程中如若触发质量红线，则无法实现代码构建

六、SDL落地与实施-难点问题解决（9）



IAST工具在落地实施过程中的难点问题：IAST工具其被动插桩技术原理存在实时不间断的漏洞识别检测状态，导致IAST工具将出现漏洞一直持续更新出现的情况（例如测试人员没有将项目应用所有功能点测试到位，后续再触发请求流量将出现新漏洞）

如果解决：

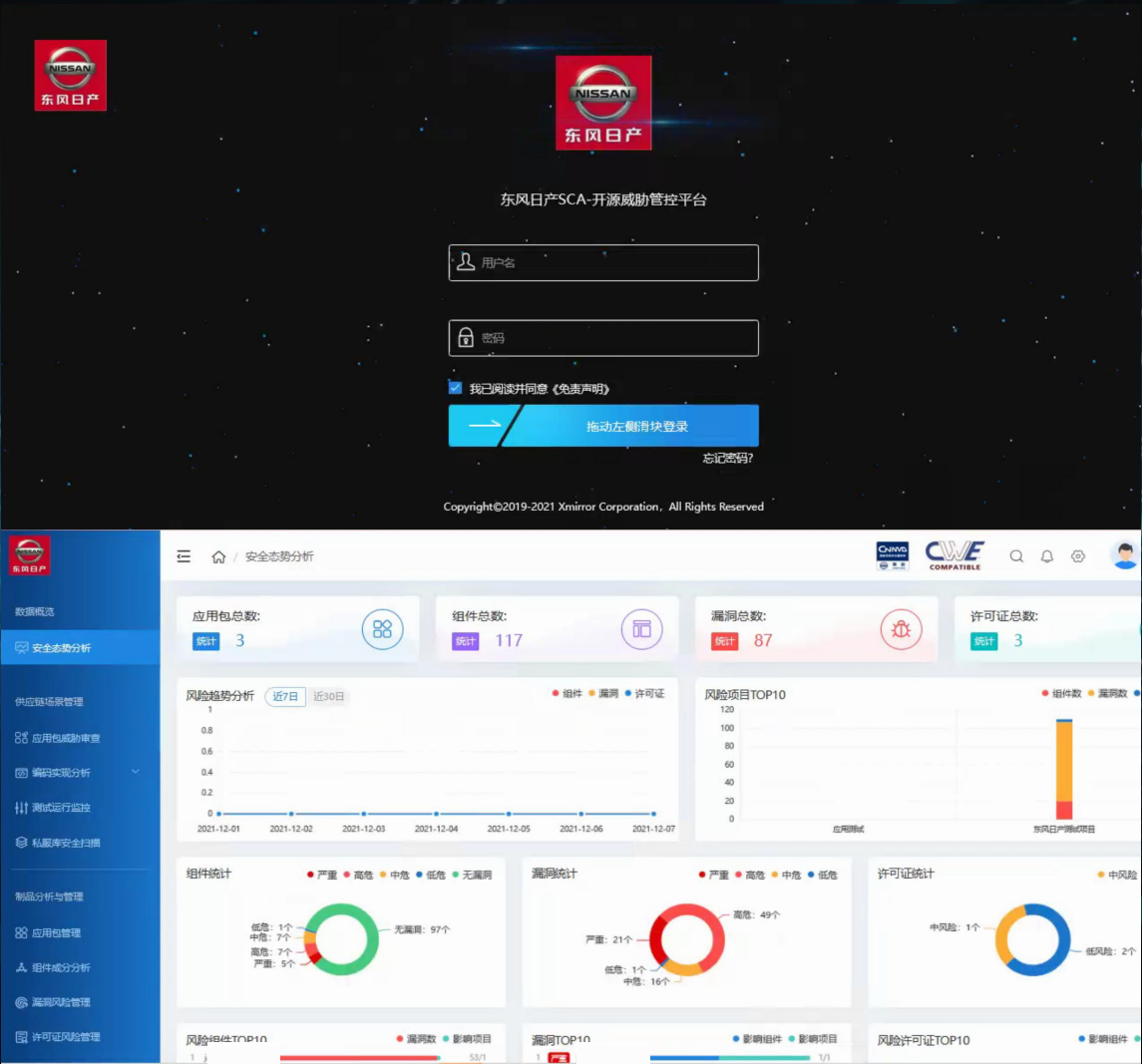
- 1、理想情况：企业内部统一制定安全开发管理流程标准并实施，开发人员编写符合标准的API接口开发文档，开发提供全量API接口开发文档给测试人员进行测试，应用系统IAST初测/复测时，测试人员通过使用API自动化测试平台对API进行高覆盖率一键扫描测试，触发IAST被动检测流量，实现安全漏洞识别与检测，IAST工具漏洞数据将实时同步至SDL平台
备注：漏洞复测时，为防止安全漏洞数据混杂，安全可以人为的把IAST工具后台的应用系统漏洞数据清理掉，最后再由测试人员通过一键API扫描测试
理想状态很难实现，因为目前企业很少能够做到完全标准的API开发和管控，同时也缺乏相对应的安全工具支持

- 2、企业目前现状解决办法：尽可能通过工具和流程进行管控。例如可以利用IAST的全量API自动扫描发现功能，设定API检测率的阈值作为临时对策，测试人员通过设定的不同API覆盖率进行分轮测试，例如初测我们设定为80%API覆盖率，复测设置70%API覆盖率或其它值

未来展望：对于API统一管控方面，第一可以通过企业内部安全开发管理流程标准不断推进去落地实施；第二是可以未来考虑部署API网关设备对企业所有应用的API进行统一检测与管理，助力企业内部开发部门形成类似全量API接口开发文档，最终交由测试人员进行统一的自动化API测试

六、SDL落地与实施-难点问题解决（10）

SCA工具在落地实施过程中的流程卡点设计问题



- SCA安全工具再SDL主线流程卡点方案设计：
- 1、研发阶段一：通过下载SCA工具插件，对企业内部私服库进行配置，实现研发阶段应用在本本地构建时的阻断（比如在build一个pom时，会进行私服安全的自动阻断，如果涉及到高危阻断下载，则不允许开发人员下载）。此流程卡点不仅效率高，且可控开发人员，开发人员必须修改问题组件才可实现成功构建
 - 2、研发阶段二：开发人员通过SCA工具插件，主动在IDE工具上进行扫描，通过返回在IDE上的结果直接修改问题组件。此卡点效率虽高，但并不可控，因为不是所有开发人员都会自觉的进行扫描和排错
 - 3、CI阶段：创建Git扫描项目，并配置定时扫描，在开发人员push代码至代码仓库后，Git项目会定时对Git的配置分支进行拉取和扫描
 - 4、CD阶段：在Jenkins平台中发起扫描，若扫描无匹配规则的组件，则扫描通过，可以进入下一阶段进行构建，若扫描结果出现匹配规则的组件，则停止构建，需要开发人员进行修整后重新上传代码
 - 5、安全测试阶段：若在测试阶段可获取应用源码包，那测试人员也可将源码包进行上传扫描，再根据扫描结果与开发人员对接。此卡点主要在于会浪费一定的资源，如若检测出组件有漏洞或许可问题，则可能涉及到组件的更换，组件更换的过程中，测试人员的各项测试无法进行，测试人员需确定无组件风险后才可进行测试，效率较低且浪费资源，所以此阶段做流程卡点一般不建议

七、SDL落地与实施效果

漏洞漏测
率降低

43%

非QA 测试发现的
漏洞数/QA 测试发
现漏洞数

漏洞复发
率降低

64%

已知漏洞再次发生
的占比降低

漏洞外部
曝光比例
降低

55%

外部上报漏洞占比
全部漏洞的比例降
低

SDL召回
率

75%

$(\text{漏洞总数} - \text{外曝漏} \\ \text{洞数}) / \text{漏洞总数}$

漏洞修复
率

90%

$(\text{QA 测试发现的} \\ \text{缺陷数} / (\text{QA 测试} \\ \text{发现的缺陷数} + \text{用} \\ \text{户使用发现的缺陷} \\ \text{数}))$

关注我们



安世加专注于网络安全行业领域，通过互联网平台、线下沙龙、峰会、人才招聘等多种形式，培养安全人才，提升行业的整体素质，助推安全生态圈的健康发展。

安世加