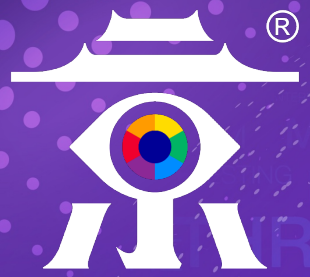


2020北京网络安全大会

2020 BEIJING CYBER SECURITY CONFERENCE

新一代灰盒安全测试技术实践分享

子芽@悬镜

软件工程安全的两个共识

系统一定有未被发现的安全漏洞

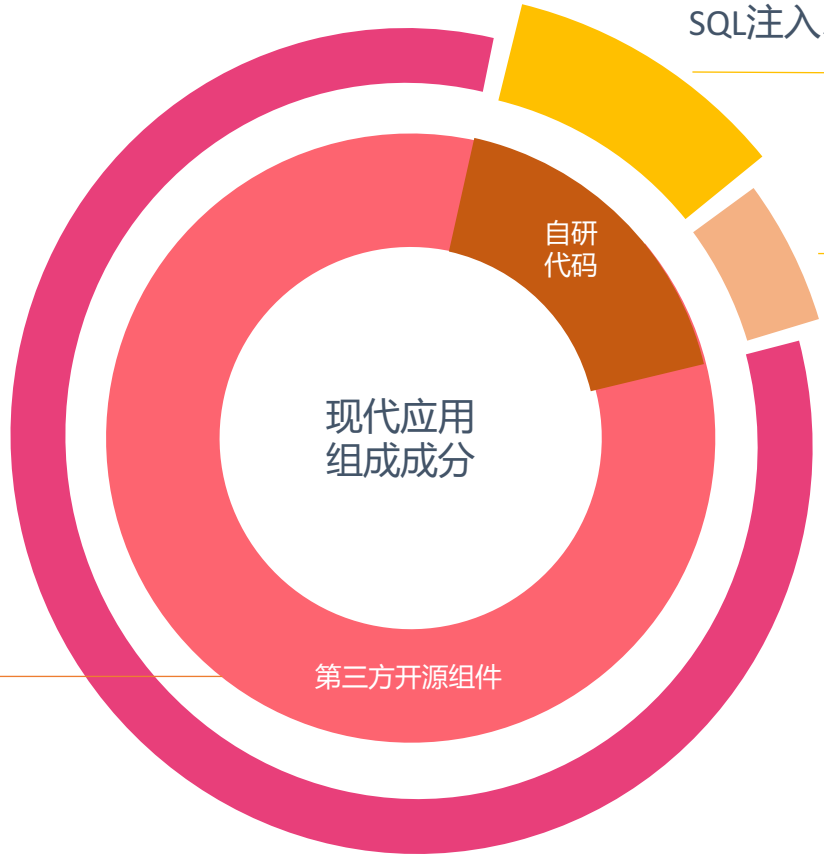
程序员每写1000**行代码**，就会出现1个逻辑性缺陷。每个逻辑性的缺陷，或者若干个逻辑性缺陷，最终导致一个漏洞；“缺陷是天生的，漏洞是必然的”。

现代应用都是组装的而非纯自研

78%-90%的现代应用融入了开源组件，平均每个应用包含147个开源组件，且**67%**的应用采用了带有已知漏洞的开源组件，软件供应链安全威胁迫在眉睫。

现代应用安全的风险面

针对现代应用全面风险审查应考虑从第三方开源组件、自研代码通用漏洞、自研代码业务逻辑漏洞等维度综合审计。



Web通用漏洞

SQL注入、命令执行、XXE、XSS等OWASP TOP10

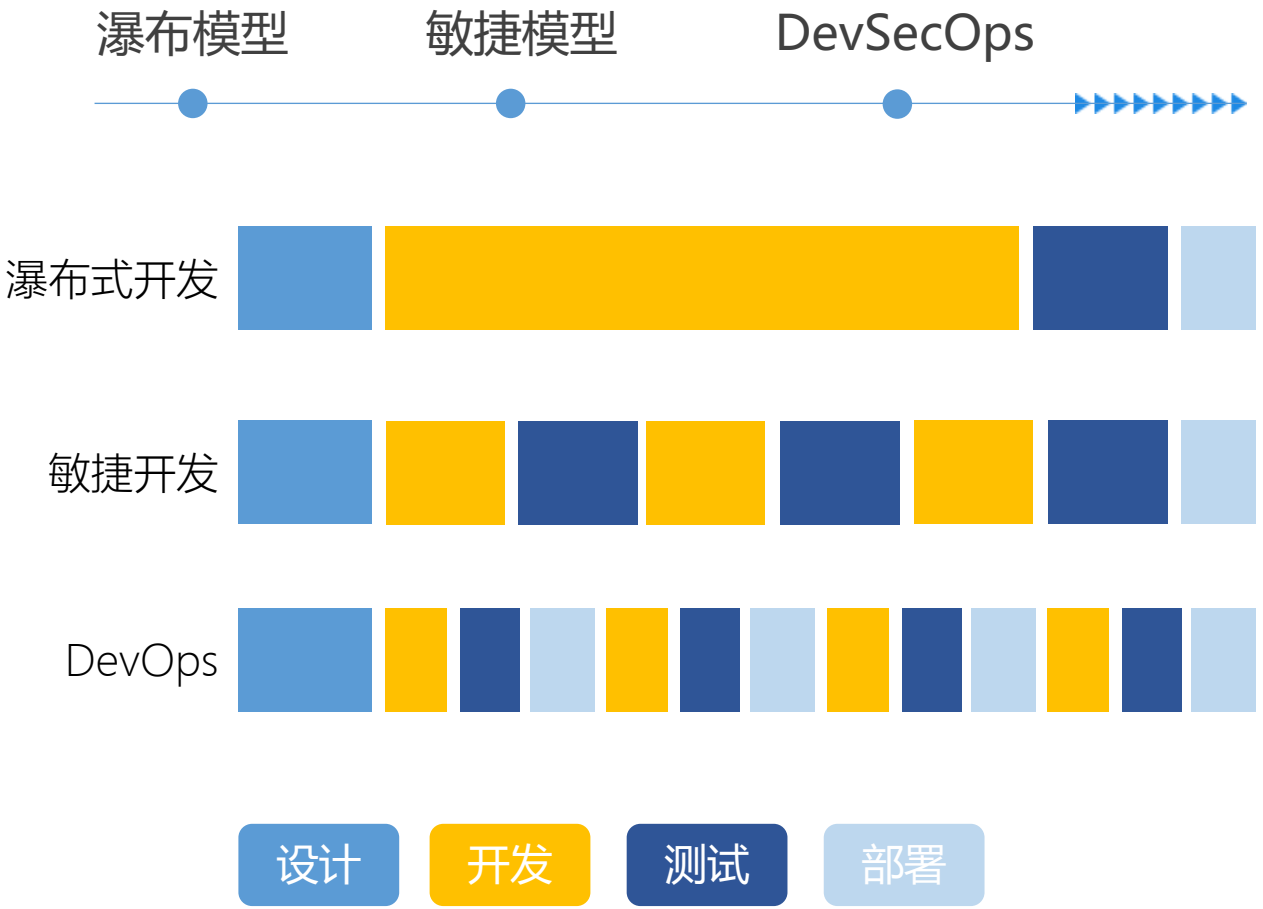
业务逻辑漏洞

水平/垂直越权、短信轰炸、批量注册、验证码绕过等

开源组件缺陷

CNNVD、CNVD、CVE等

现代应用开发模式的技术演进



传统SDL面临的安全挑战

- 业务先上线，安全问题后补救
- 安全责任过于依赖有限安全团队资源
- 安全较缓慢，常置于流程之外，当版本更新快时，传统安全手段影响业务交付

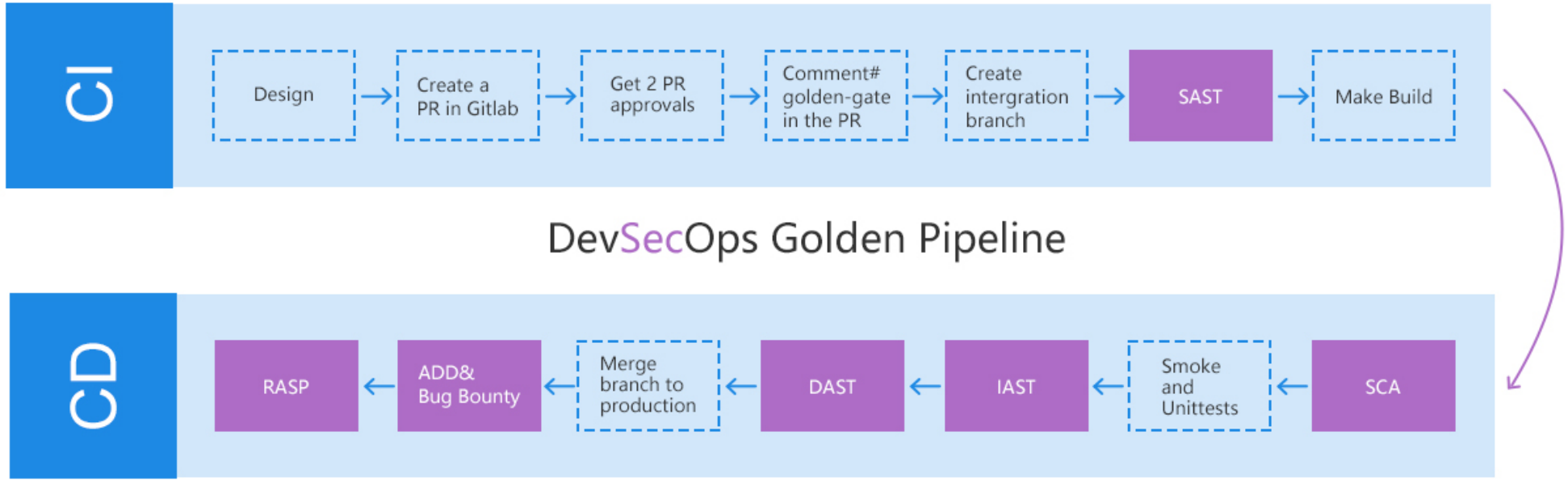
DevSecOps新特性

- 责任：安全是每个人的责任
- 嵌入流程：开发运维
- 柔和嵌入研发运维流程，自动化
- 自动化流程，人更趋向于运营反馈处理
- 适用于周期较短，迭代较快的业务

DevSecOps CI/CD黄金管道

RSAC2018正式提出 “Golden Pipeline” 软件流水线实践体系，强调 CI/CD 自动化工具链支撑

CI/CD黄金管道：



关键工具链技术：

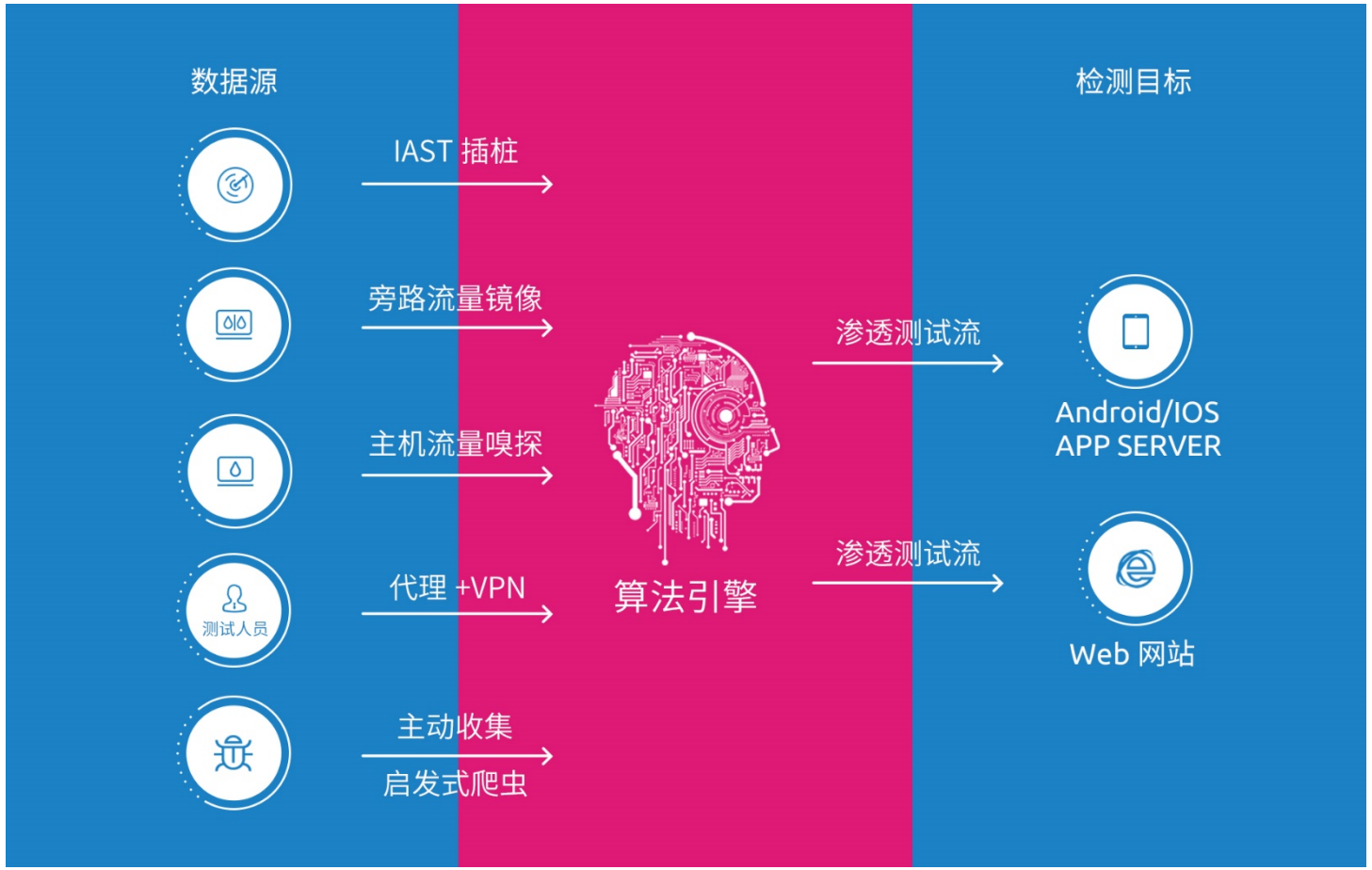
1. AST应用安全测试
2. SCA第三方组件成分分析
3. RASP运行时应用自保护
4. 红蓝对抗和SRC

AST技术优劣对比

	DAST	SAST	IAST
误报率	低	高	极低
检出率	中	高	高
检测速度	随URL、payload数量	随代码量	依赖点击流量实时检测
第三方组件漏洞	依赖payload、指纹	静态扫描支持	运行时支持
语言支持	不区分语言	区分不同语言	区分不同的语言
框架支持	不区分框架	一定程度区分	一定程度区分
漏洞验证利用	可验证利用	很难验证利用	可验证利用
使用风险	脏数据、大流量	无	无
使用成本	较低	高，人工排查误报	低，基本没有误报
漏洞详情	参数、请求响应	代码行数、执行流	请求响应、代码行数、数据流
CI/CD支持	低	高	高
漏洞种类覆盖	可发现配置、运维、运行时层面漏洞	更偏向应用代码漏洞	更偏向应用本身漏洞，难以回显带外也可发现

任何一项新兴技术出现，都有时代的背景和其适用的场景。

IAST灰盒安全测试技术



IAST主要关键技术

低门槛

无需专业技术背景，普通测试人员透明众测完成业务安全测试

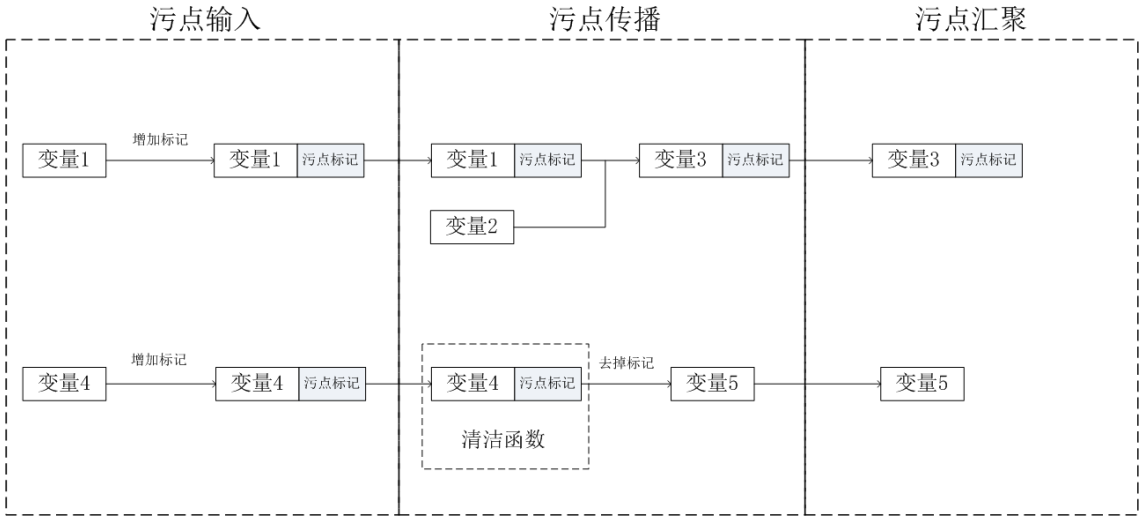
低侵入

旁路部署不影响正常业务流执行和通信效率。

低消耗

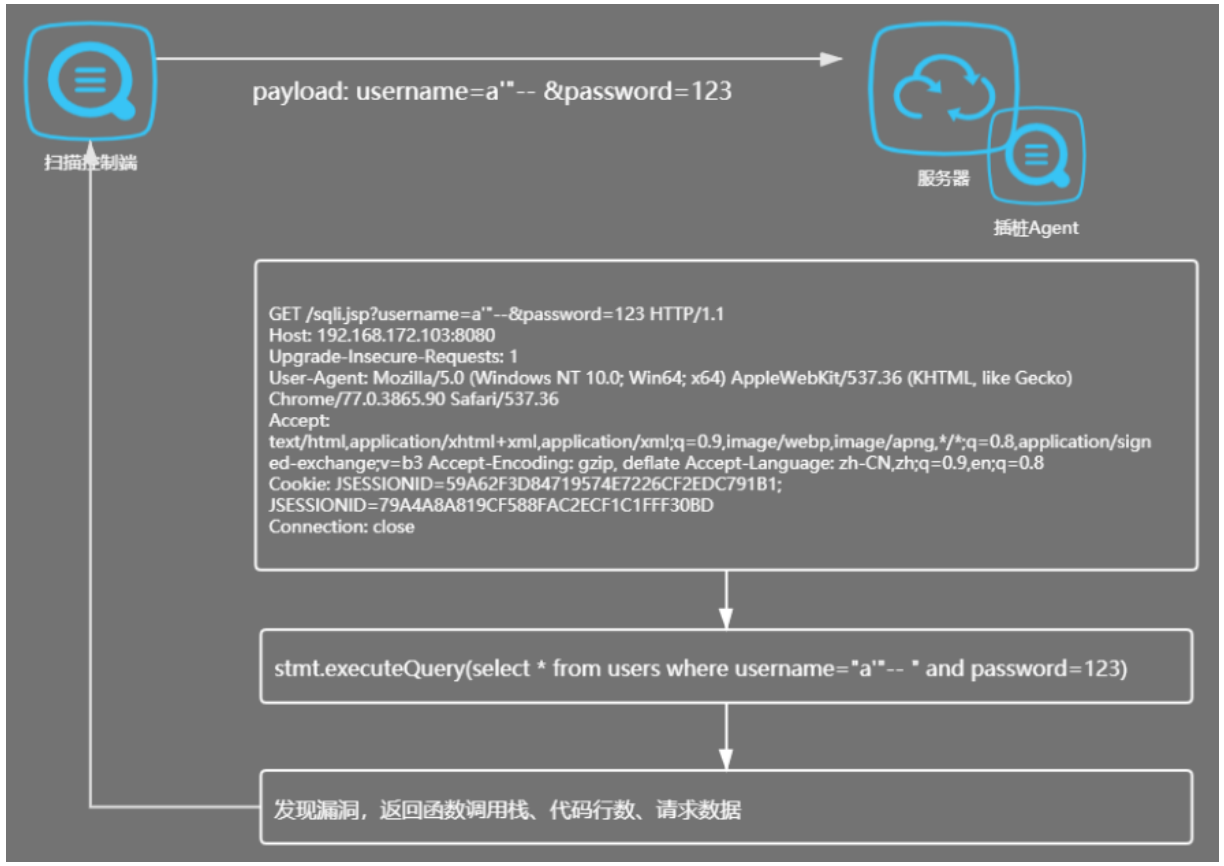
透明部署，不影响正常测试工作和用户使用流程。

被动IAST插桩-动态污点追踪技术



- 误报及漏报高于主动IAST
- 无数据重放、无脏数据
- 可处理签名、加密接口

主动IAST插桩-交互式缺陷定位技术



- 扫描端发送payload
- 插桩Agent在关键函数，获取上下文信息综合分析

  1）HTTP/HTTPS请求/响应

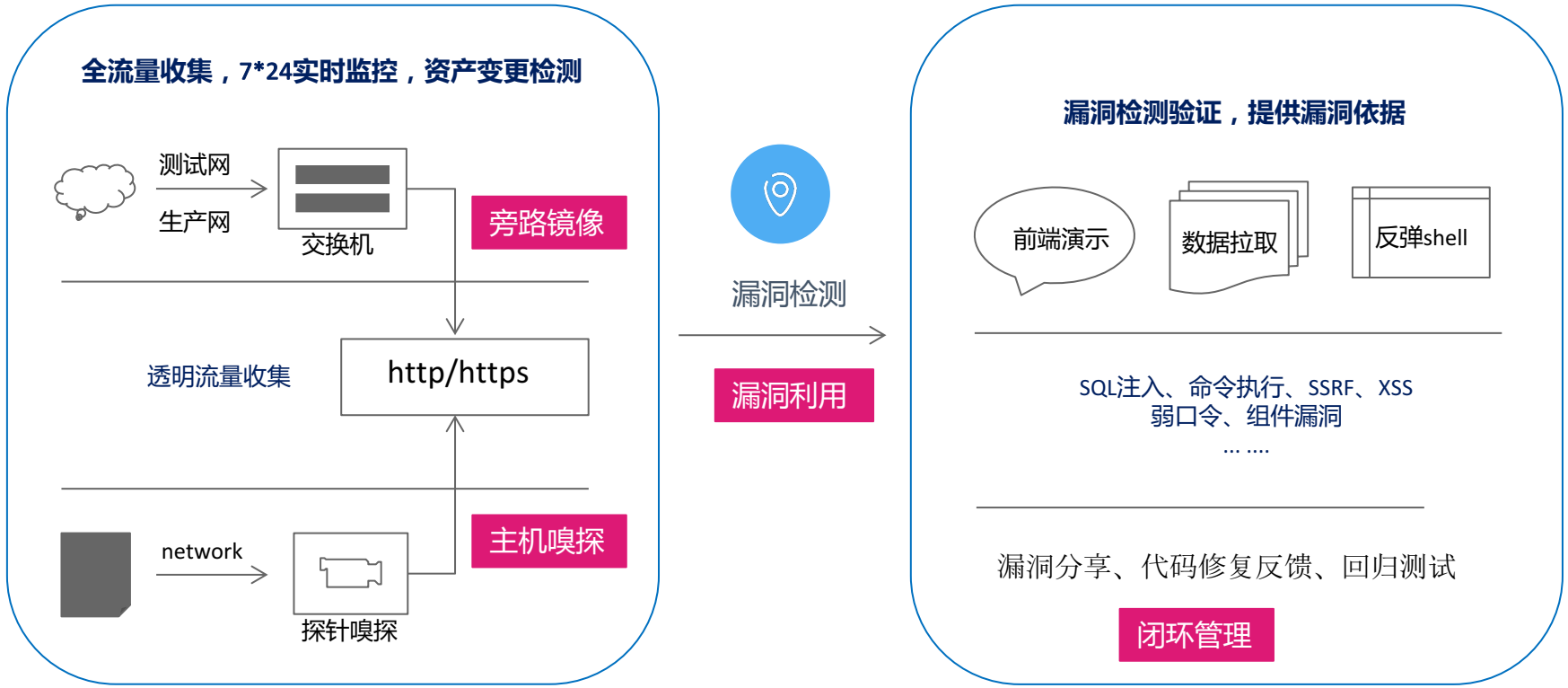
  2）URL是否触发相关插桩点

  3）payload是否进入了执行流程

- 精度更高，更易于指导研发修复
- 支持漏洞利用、漏洞复现
- 无法处理签名加密接口

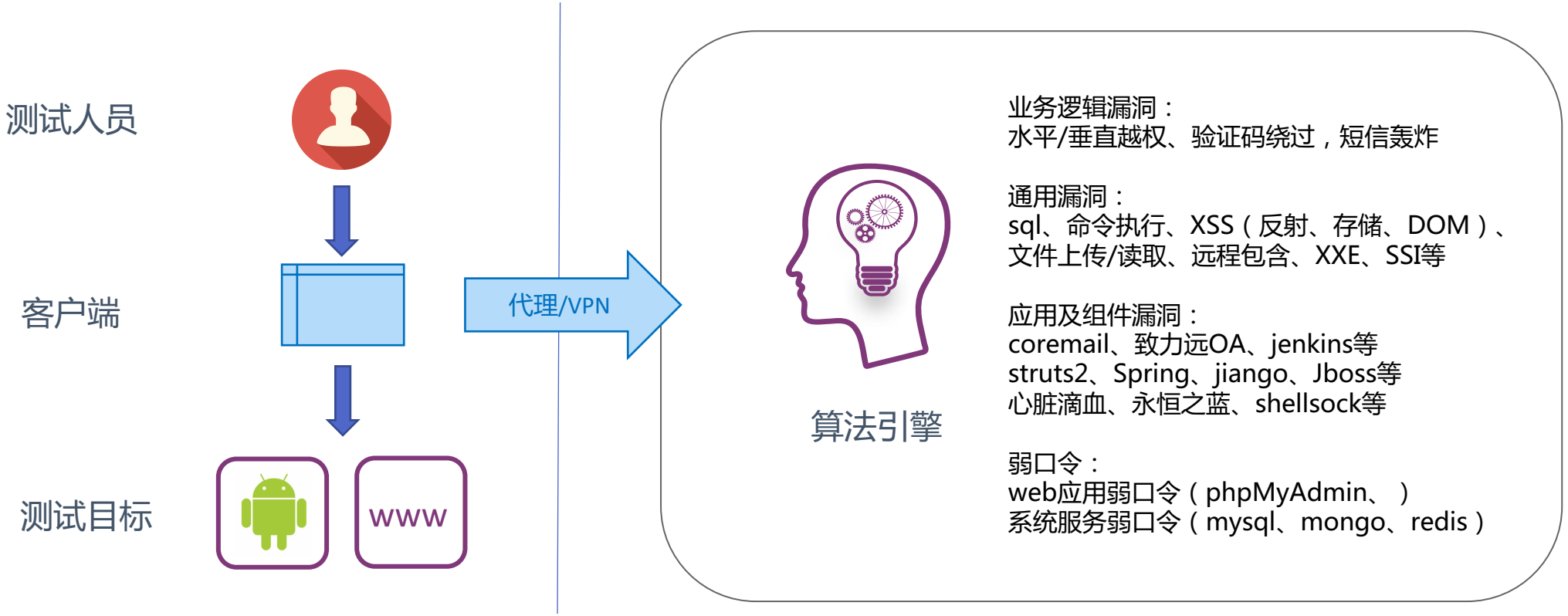
IAST流量镜像及嗅探技术

适合测试目标或部门之间存在隔离，无法清晰了解测试资产，通过在网络出口旁路部署流量镜像，透明无感知接入。



IAST流量代理及VPN技术

适合测试人员对单个网站进行深度渗透测试，不依赖测试系统语言环境，无须介入测试环境即可进行安全测试。



运行时OSS开源组件分析

运行时OSS分析，更加侧重应用系统实际运行过程中动态加载的第三方组件及依赖，在此基础上实时检测组件中潜藏的各类安全漏洞及开源协议风险。

IAST技术总结

IAST技术特点

- 相比传统SAST/DAST，IAST精准度更高；
- IAST可以解决签名接口问题；
- IAST获取信息全面和精细，有利于指导研发；
- IAST更易于整合到DevSecOps CI/CD流程；
- 主被动IAST融合，将使IAST技术优势更加突出。

落地重点考虑事项

- 同时支持应用插桩和多种流量追踪技术，应对业务场景丰富、开发语言众多、部署环境复杂等场景；
- 支持自动化安装，协调CI/CD完成批量部署；
- 支持热加载、性能异常熔断机制；
- 支持OSS第三方开源组件运行时监测分析；
- 支持Jenkins 、Jira、 CAS等第三方平台。

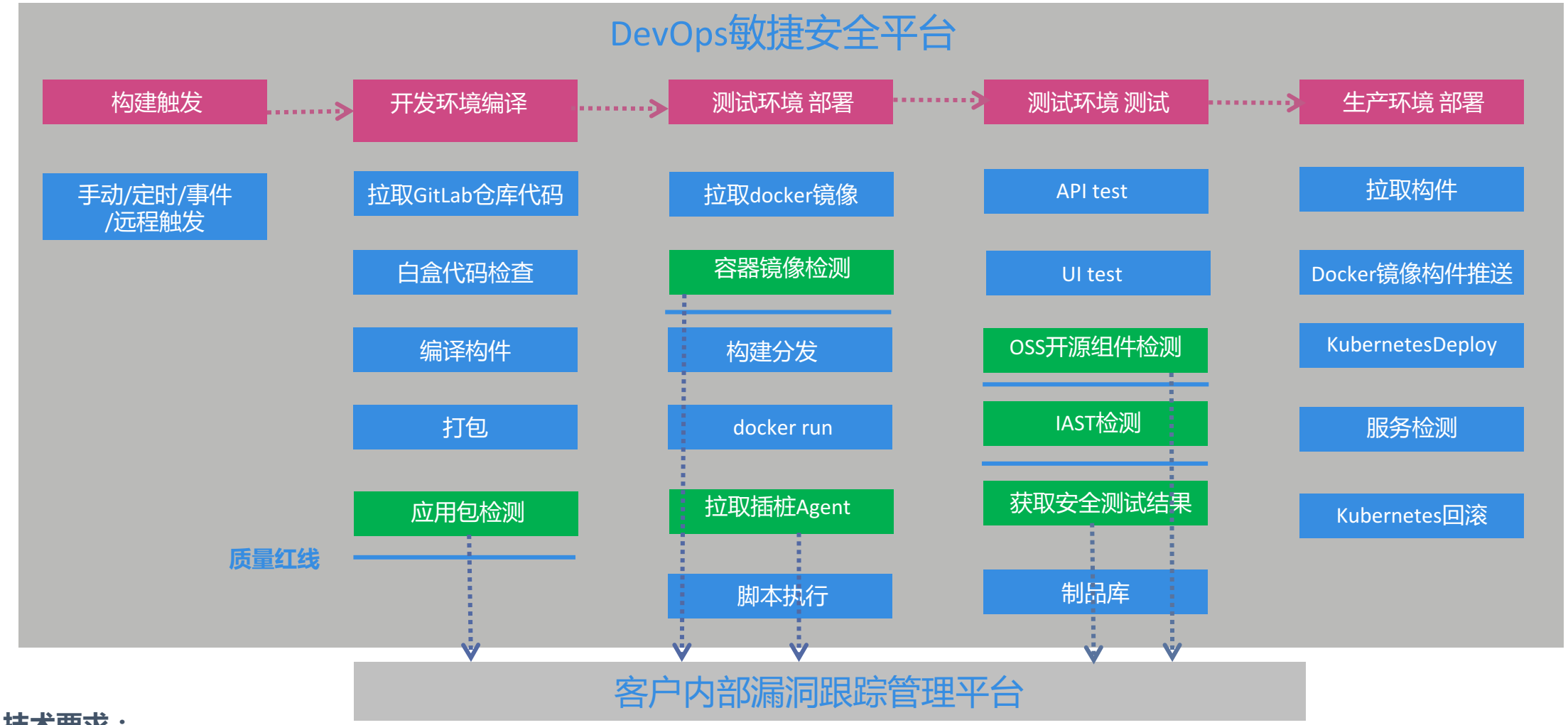
IAST方案实践期待

新一代灰盒安全测试平台

智慧闭环漏洞管理

- **漏洞发现**
  - 多智能验证引擎接近零误报
  - 自动化漏洞演示，提供详尽的漏洞判断依据
  - 测试用例覆盖度分析
- **问题沟通**
  - 邮件通知项目负责人
  - 单条/批量漏洞分享及确认
  - 双向消息同步
- **修复整改**
  - 漏洞代码行级定位
  - 提供优劣代码示例展示
  - 主动漏洞验证辅助
- **漏洞复查**
  - 单个漏洞重新检测
  - 整体项目回归测试
  - 项目整改对比分析

某银行DevSecOps模式下IAST实践



技术要求：

1）整个流程十几分钟，测试用例覆盖度可度量(类数量、请求数量)；
2）质量红线设定很关键（漏洞等级、对应CVE是否有POC、应用内外网属性等）；
3）要求批量自动化安装部署，支持热加载；
4）支持自研私有协议和开发框架。

IAST关键技术的演进方向

应用基础设施安全

随着云大物移的发展，应用的形态与需求正在经历变革，API、容器等软件基础设施安全愈发重要

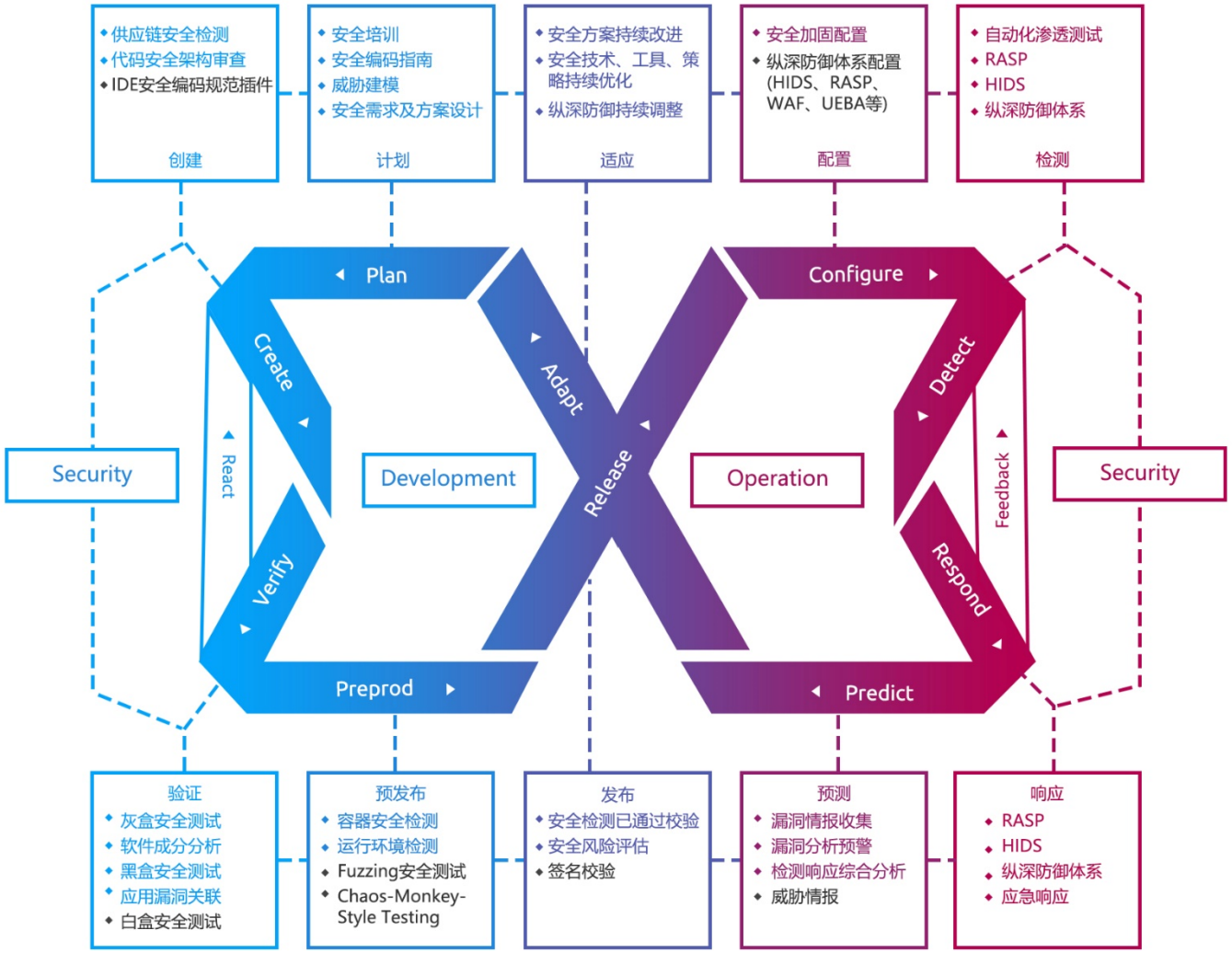
AST技术融合

AST白盒、灰盒及黑盒安全测试技术深度融合及检测手段编排，发挥各自技术优势，在精度和覆盖度上有质的提升。

基于CARTA的漏洞管理

关联分析各阶段的检测成果，结合应用业务价值与风险，综合评价漏洞优先级和治理措施

悬镜DevSecOps探索实践成果



安全是一门平衡艺术

- 本质是风险和信任的平衡
- 拥抱变化是安全建设的基石

人是安全的基本尺度

- 从源头做威胁治理
- 应用内生安全

悬镜落地实践方案

悬镜DevSecOps智适应威胁管理体系

安全开发服务

- SDL安全咨询
- 安全开发实训
- 安全编码指南
- 源代码审计

安全运营服务

- 渗透测试
- 攻防演练
- 风险评估
- 等保咨询


2020北京网络安全大会
2020 BEIJING CYBER SECURITY CONFERENCE
THANKS
全球网络安全 倾听北京声音