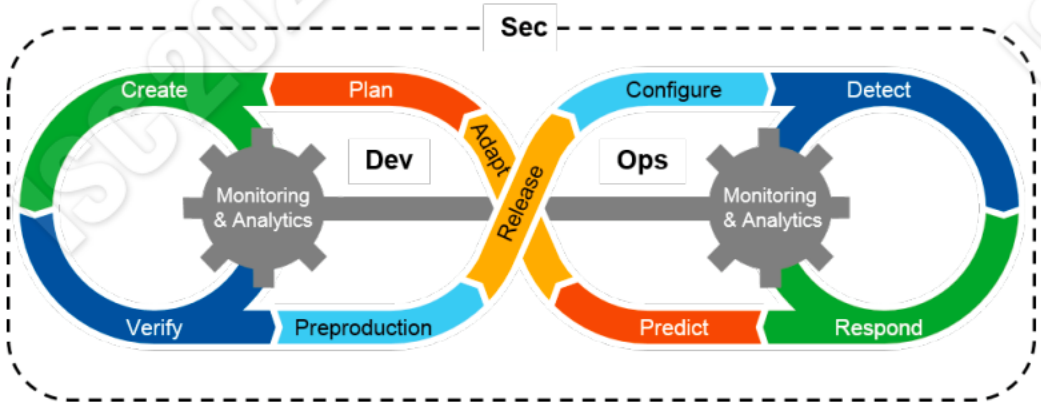
利用IAST推动应用安全测试自动化

IAST是DevSecOps实现自动化安全测试的最佳工具之一

分享人：徐锋（杭州孝道科技有限公司 CTO）

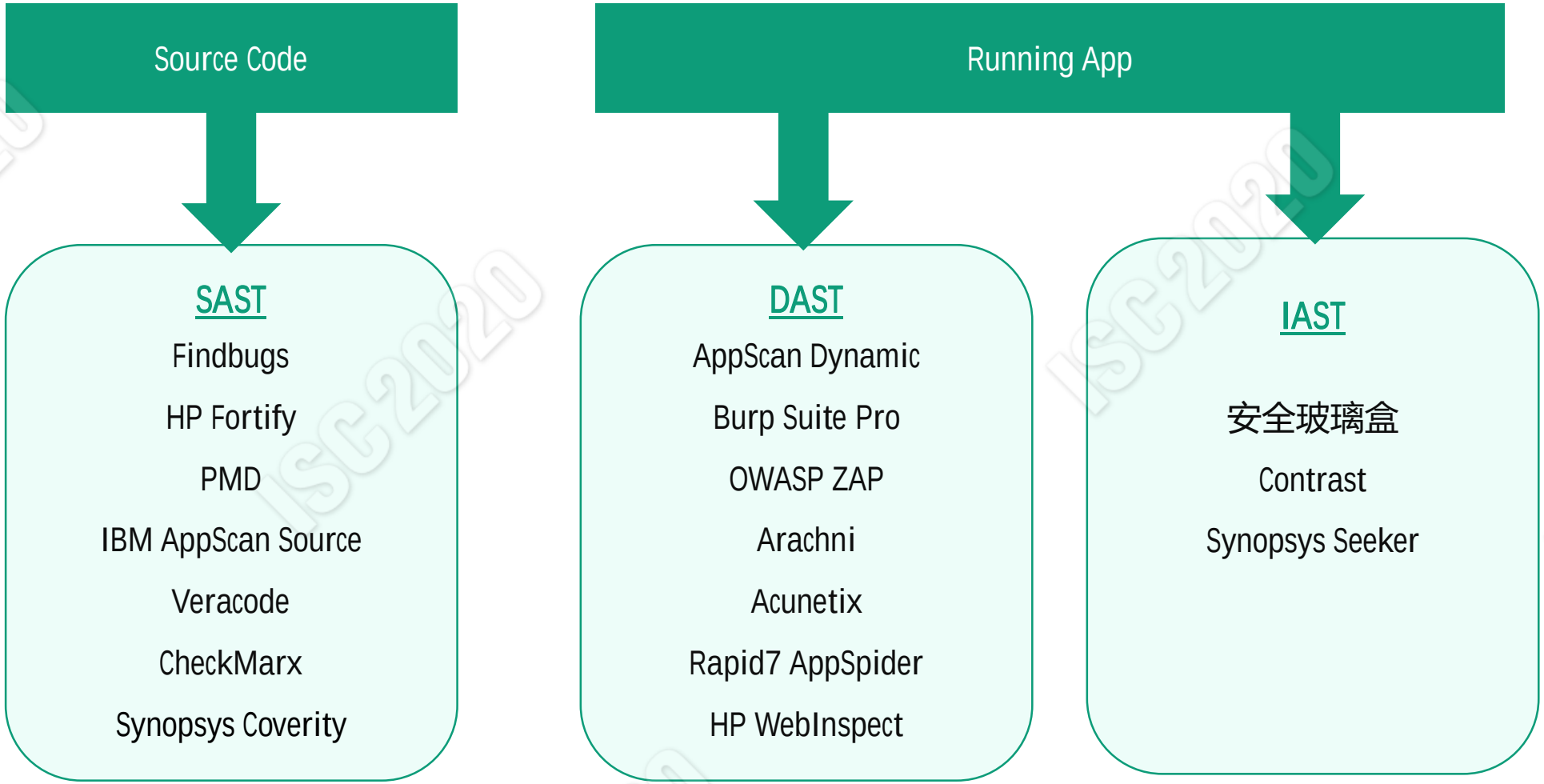
别给别人添麻烦

不要试图改变程序员和测试人员的工作方法，也不要去增加他们额外的工作负担。



Source: Gartner (September 2016)

应用安全测试常用工具


Source Code
Running App
SAST
Findbugs
HP Fortify
PMD
IBM AppScan Source
Veracode
CheckMarx
Synopsys Coverity
DAST
AppScan Dynamic
Burp Suite Pro
OWASP ZAP
Arachni
Acunetix
Rapid7 AppSpider
HP WebInspect
IAST
安全玻璃盒
Contrast
Synopsys Seeker

工作原理

词法分析、 抽象语法、语义分析、跟踪控制流、跟踪数据流、污染传播、规则分析…

不能敏捷的融入到 DevOps 中

SAST 因误报率高、效率低

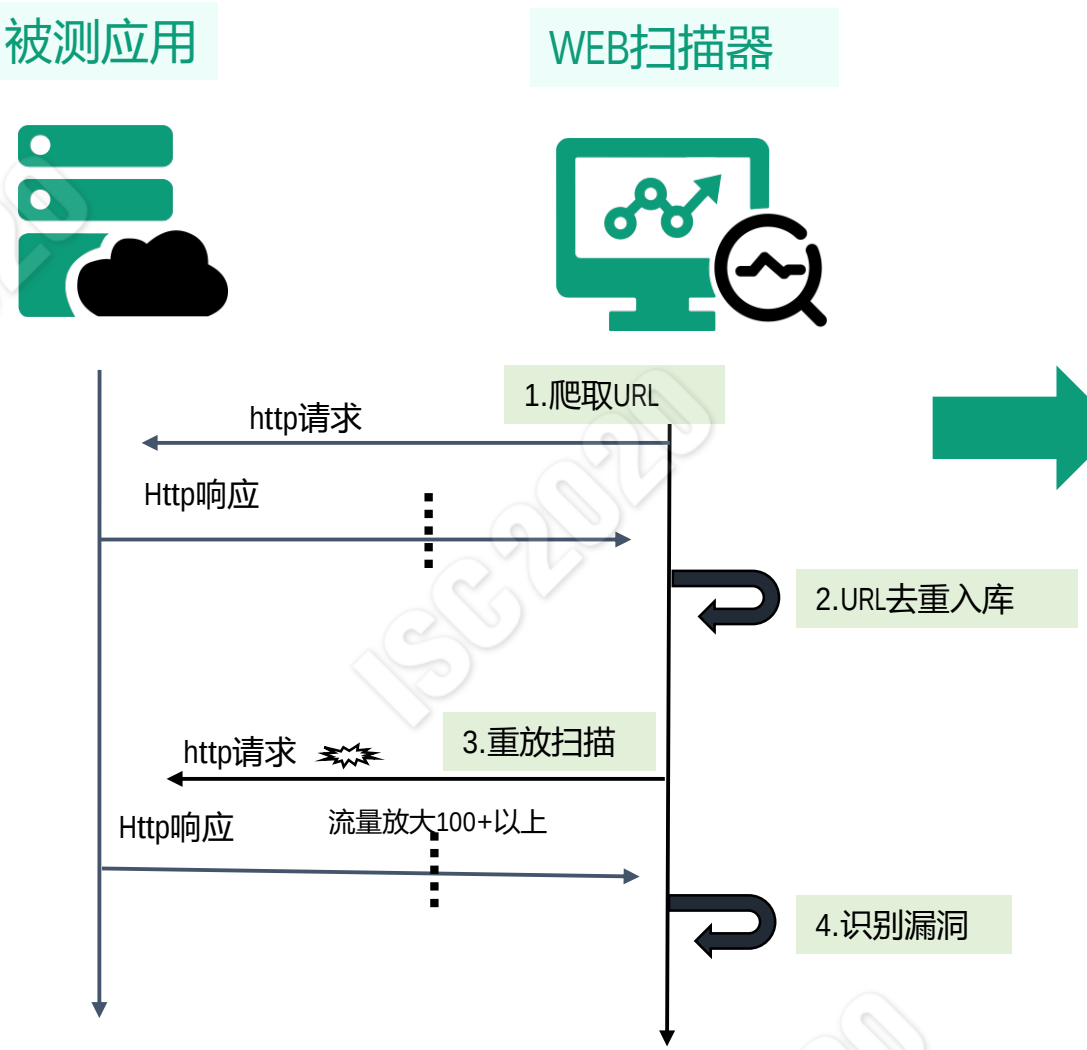
优势

- 覆盖率高
- 适用不同应用
- 问题定位到源码位置
- IDE集成

劣势

- 检测效率低
- 漏洞误报高
- 区分语言和框架
- 运营成本极高

DAST爬虫型WEB扫描器



优势

- 与开发语言无关
- 可执行远程扫描
- 使用简单

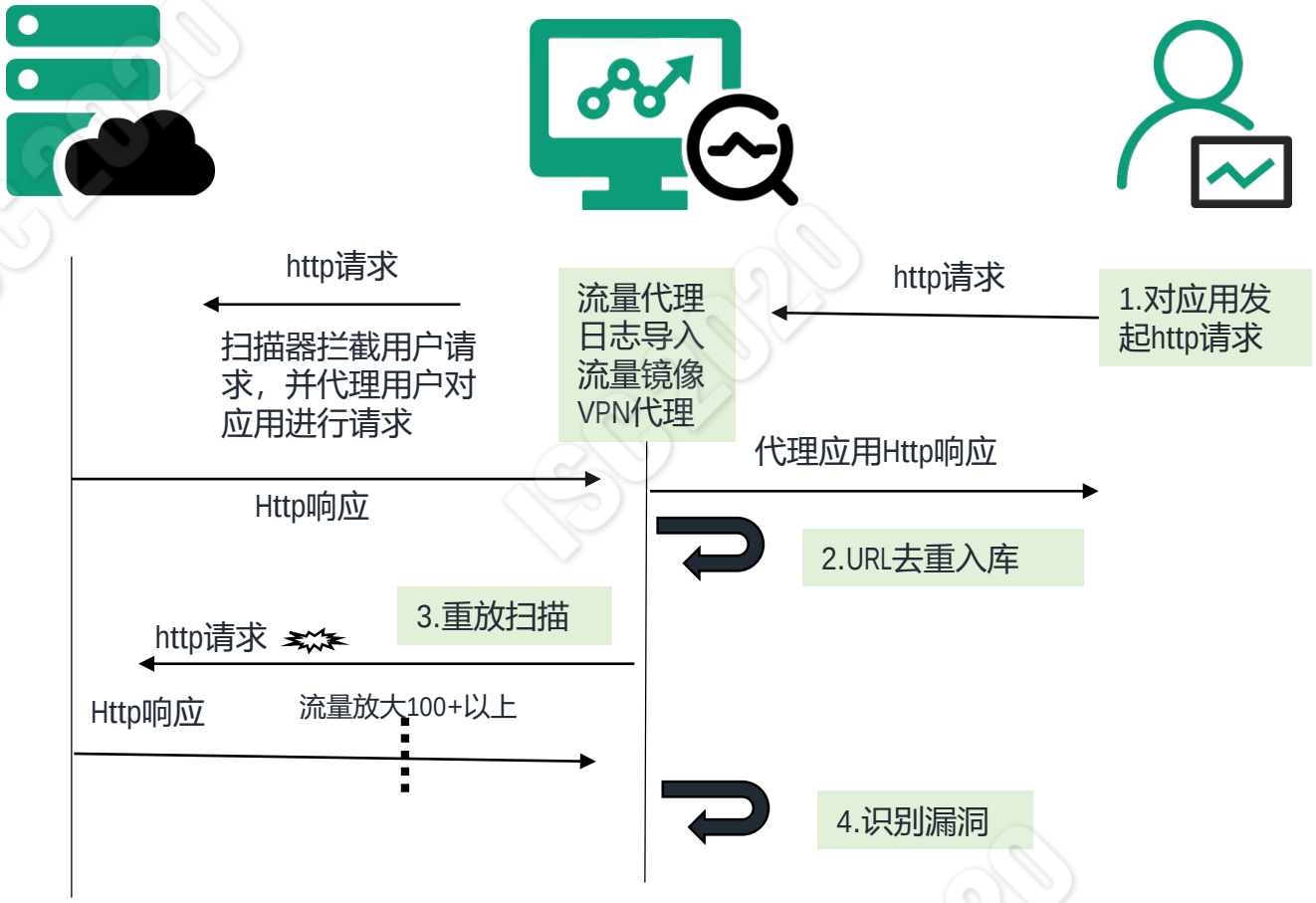
劣势

- 检测效率低
- 漏洞误报较高
- 覆盖率低：URL不全、加密流量、加签、一次资源等
- 产生“脏数据”、“脏操作”
- 漏洞详情粗略

DAST代理型WEB扫描器

被测应用

流量代理型WEB扫描器



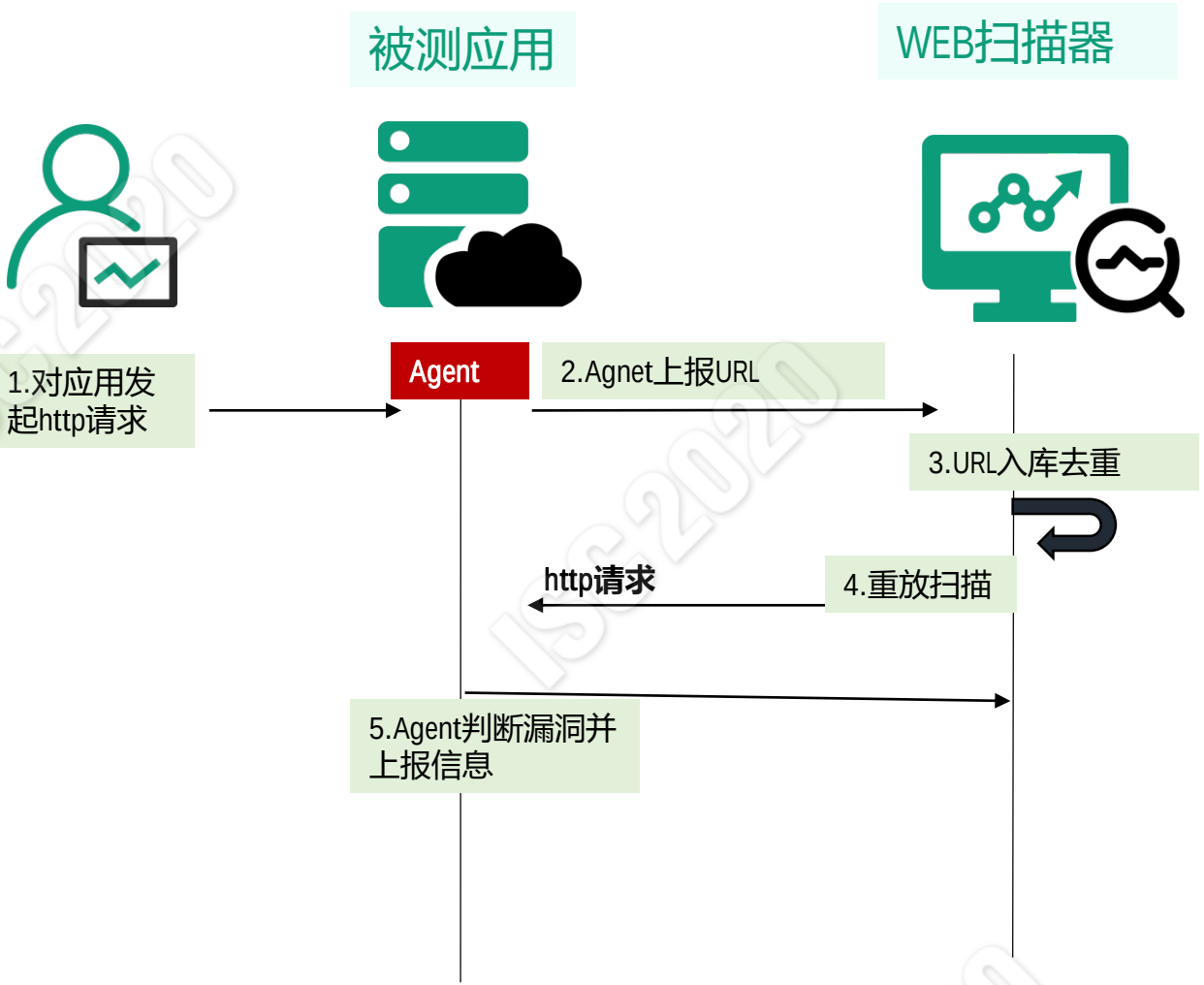
优势

- 与开发语言无关
- 可执行远程扫描
- 使用简单

劣势

- 覆盖率低：加密流量、加签、一次资源等
- 检测效率低
- 漏洞误报较高
- 无法透明融入DevSecOps
- 漏洞详情粗略

IAST插桩主动型IAST



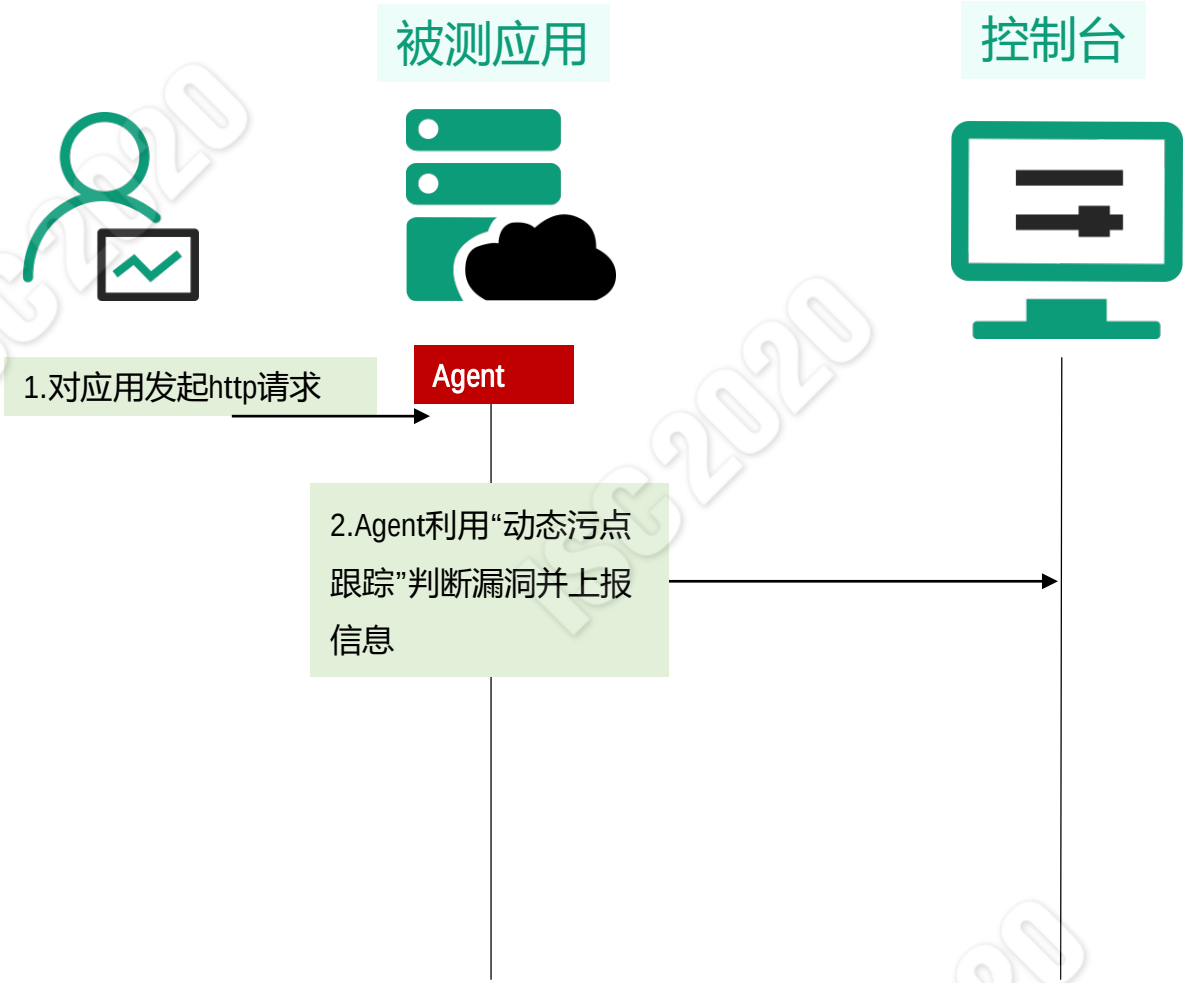
优势

- 检测效率有较高
- 可定位到漏洞代码
- 几乎没有误报

劣势

- 覆盖率低：加密流量、加签、一次资源等
- 与开发语言强关联性
- 需要部署Agent
- 无法透明融入DevOps

IAST插桩被动型(动态污点分析)IAST



优势

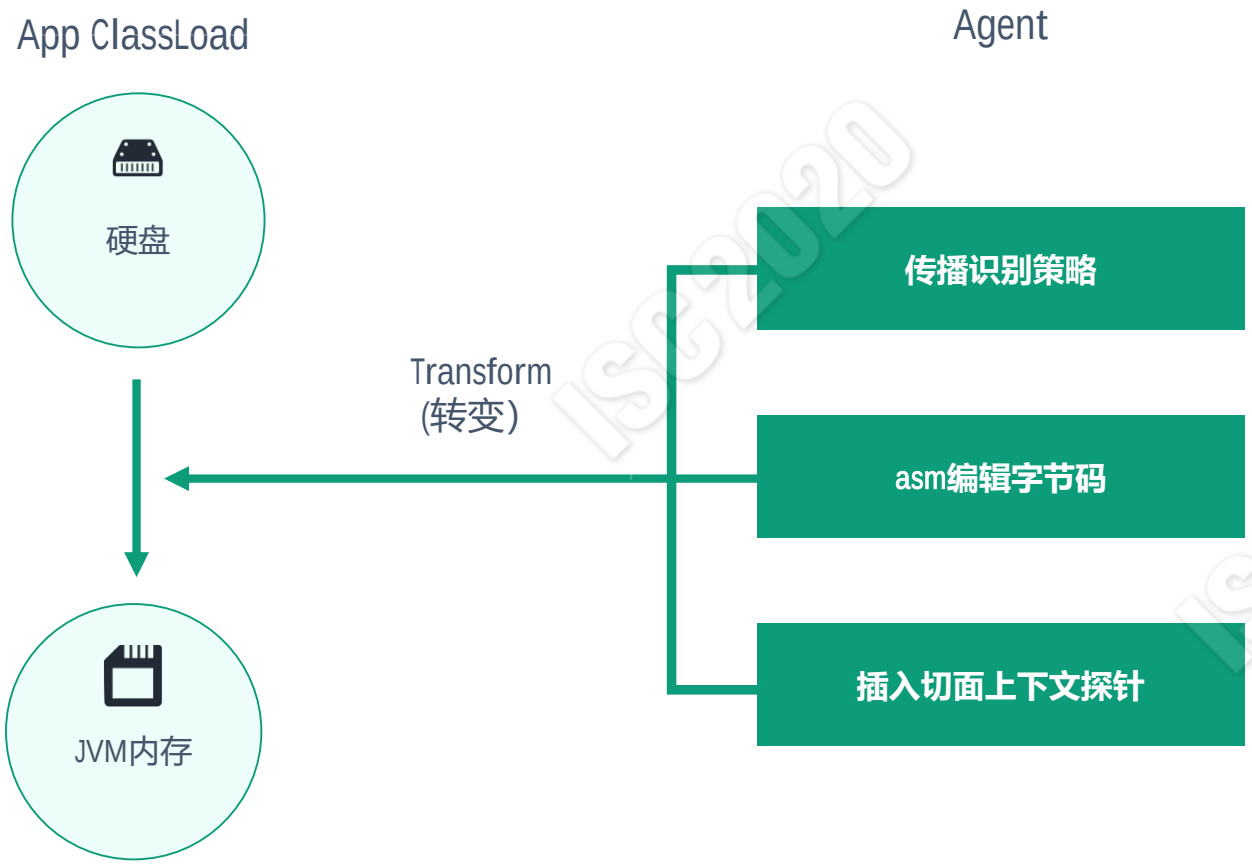
- 不产生脏数据、脏操作，可透明融入DevOps
- 检测效率极高实时检测
- 可定位到漏洞代码
- 误报率相对较低

劣势

- 需要部署Agent
- 与开发语言强关联性
- 无法有效识别安全过滤

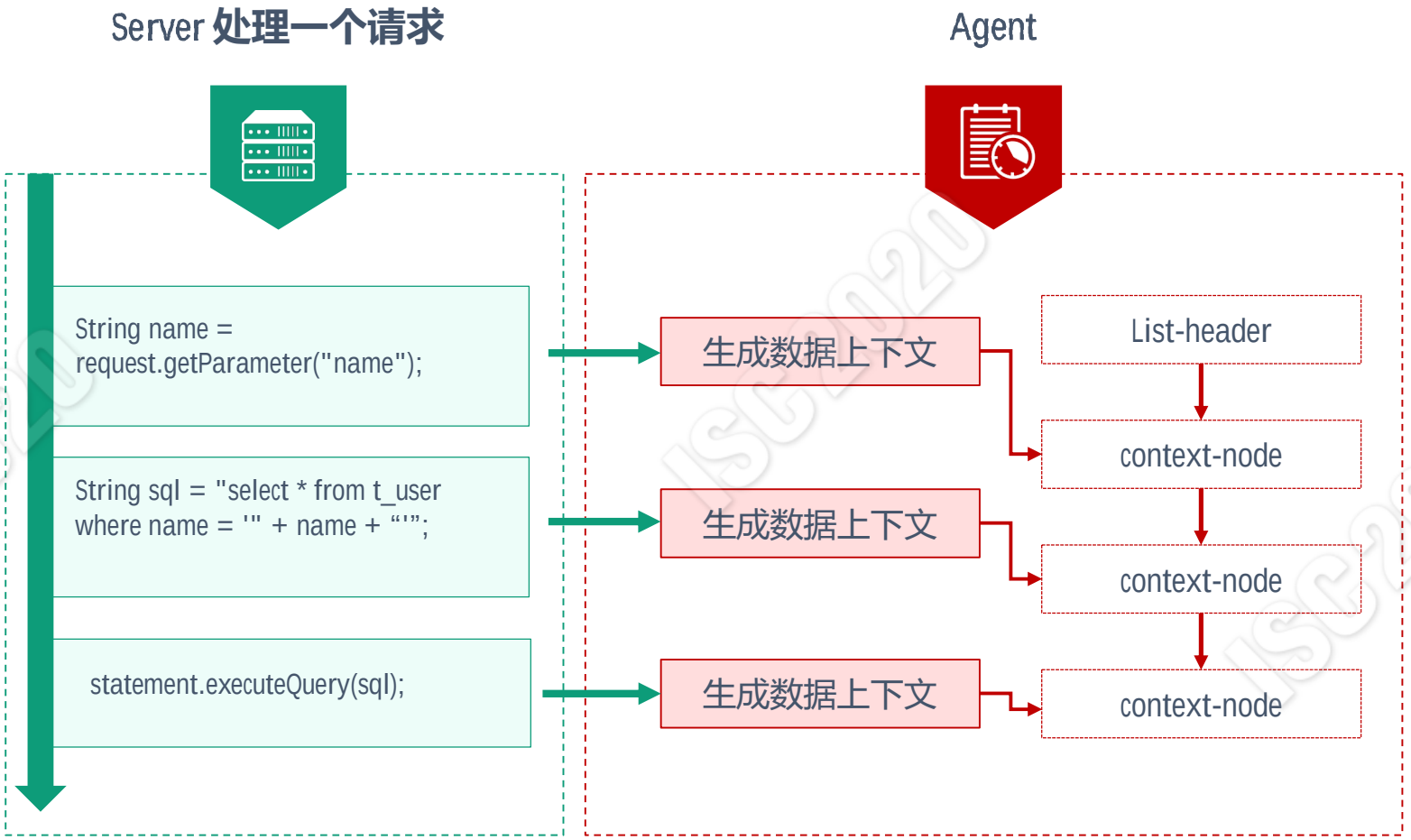
Agent动态修改字节码—插桩

- 在应用加载到JVM内存前提前加载Agent



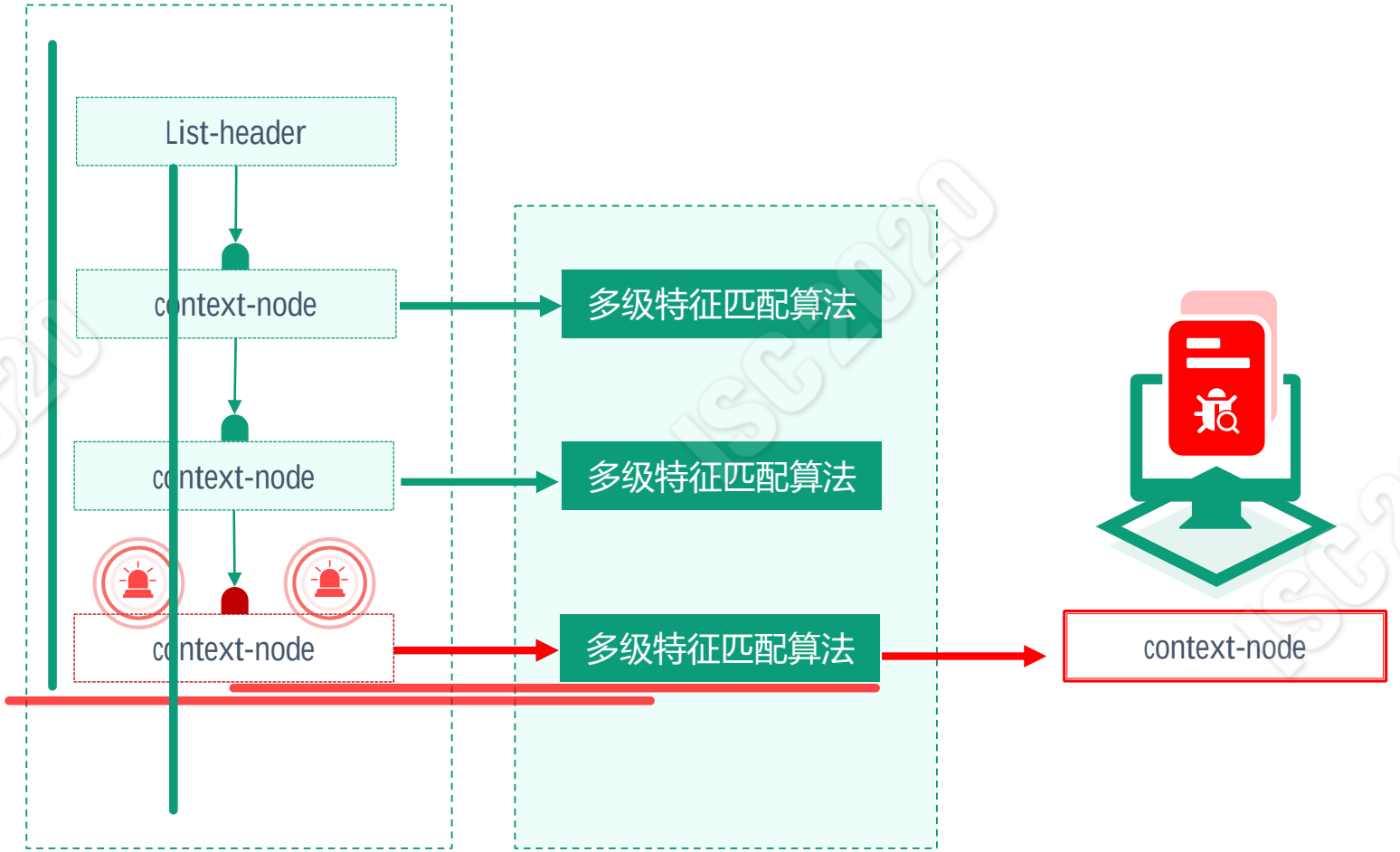
生成应用数据流—动态传播

- Agent会实时监控应用内部数据流上下文。
- 形成应用内部数据流图。



漏洞判断

- 外部输入参数
- 未经过安全过滤
- 传播到“风险方法OR函数”



THANKS

ISC 2020
第八届互联网安全大会
INTERNET SECURITY CONFERENCE 2020
数字孪生时代下的新安全
New Security in the Digital Twin Era