OWASP
Open Web Application
Security Project

DevSecOps最佳实践探索

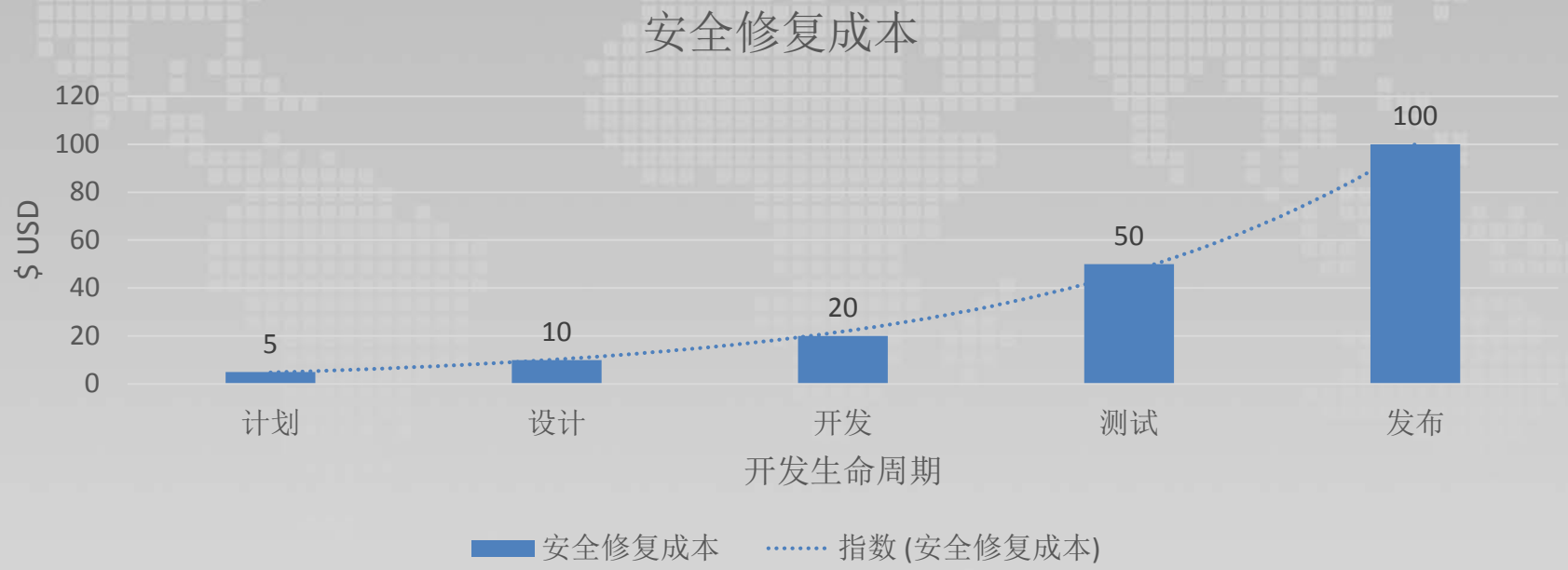
肖文棣 Mendick Xiao
安全架构师

目录

- DevSecOps漫话
- DevOps和DevDevOps
- DevSecOps和S-SDLC
- S-SDLC向DevOps演进的实践
- DevSecOps的未来

DevSecOps漫话

安全修复成本曲线

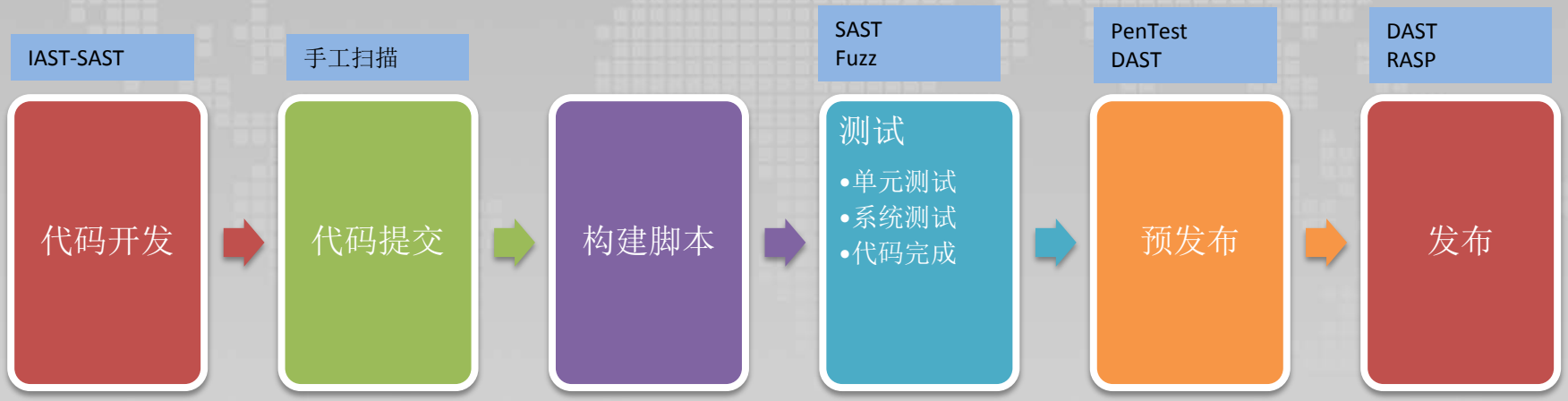


DevSecOps宣言

- **Leaning in** over Always Saying “No” 向前一步胜过说不
- **Data & Security Science** over Fear, Uncertainty and Doubt 数据和安全科学胜过害怕、不确定和质疑
- **Open Contribution & Collaboration** over Security-Only Requirements 开放性的贡献和合作胜过单纯的安全性需求
- **Consumable Security Services with APIs** over Mandated Security Controls & Paperwork 可用的基于API的安全服务胜过强制性的安全控制和文档
- **Business Driven Security Scores** over Rubber Stamp Security 业务驱动的安全分数胜过橡皮图章的安全审核
- **Red & Blue Team Exploit Testing** over Relying on Scans & Theoretical Vulnerabilities 红蓝团队漏洞测试胜过依赖扫描和理论上的漏洞
- **24x7 Proactive Security Monitoring** over Reacting after being Informed of an Incident 24x7主动安全监控胜过安全事件通告后的应急响应
- **Shared Threat Intelligence** over Keeping Info to Ourselves 共享威胁情报胜过保持自己的信息
- **Compliance Operations** over Clipboards & Checklists 合规运营胜过剪贴板和检查清单

http://www.devsecops.org/

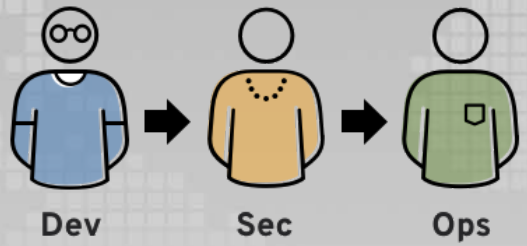
DevSecOps的过程

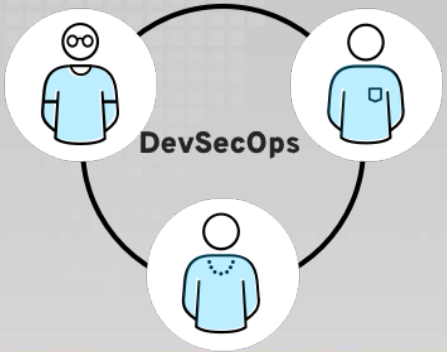


Gartner的DevSecOps九条建议

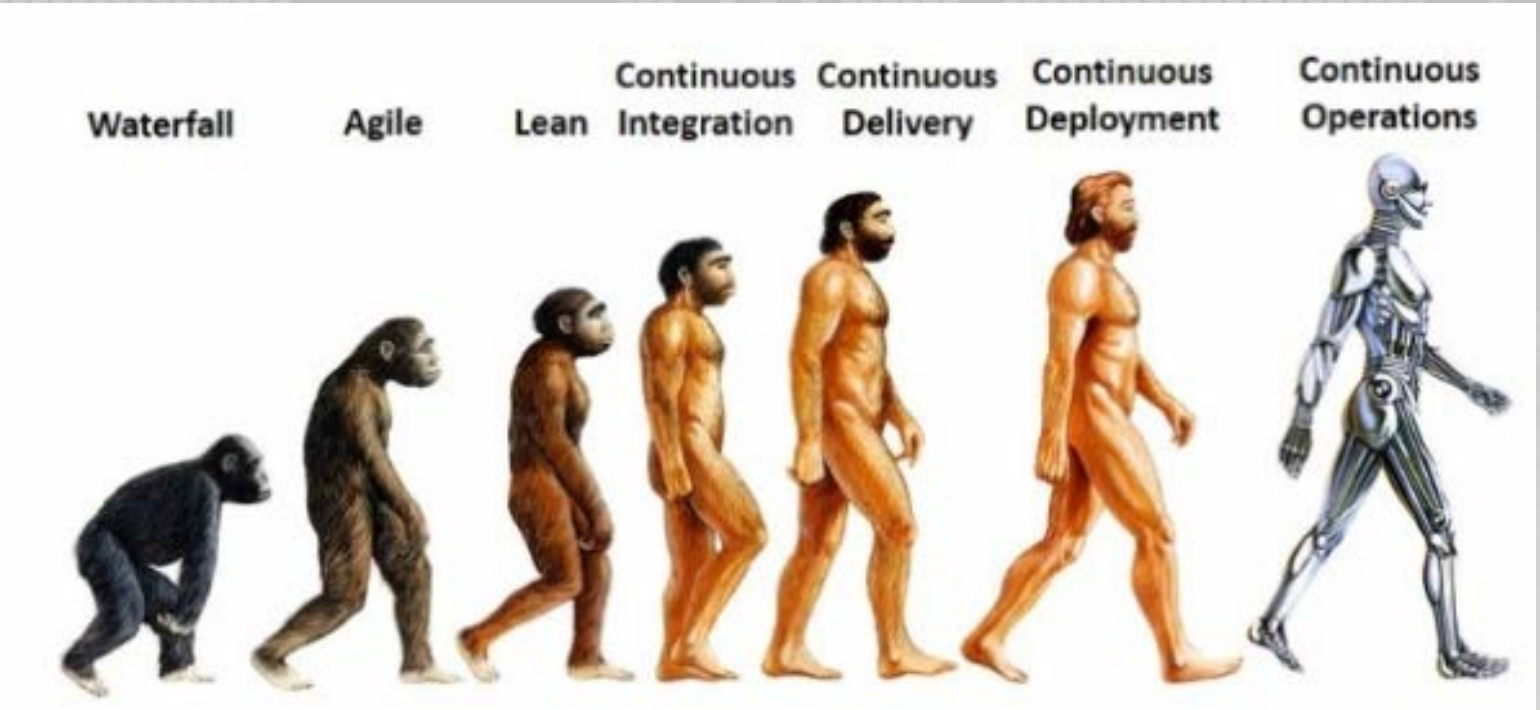
- Adapt your security testing tools and processes to the developers, not the other way around 让你的安全测试工具和流程适应开发人员，而不是相反
- Quit trying to eliminate all vulnerability during development 不要尝试消除开发过程中的所有漏洞
- Focus first on identifying and removing the known critical vulnerabilities 首先关注识别和删除已知的严重漏洞
- Don’t expect to use traditional DAST/SAST without changes 不要期望在没有变化的情况下使用传统的DAST/SAST
- Train all developers on the basic of secure coding, but don’t expect then to become security experts 培训所有开发人员基本的安全编码规范，但不要期望他们成为安全专家
- Eliminate the use of known vulnerable components at the source 从源头上消除已知的易受攻击组件的使用
- Secure and apply operational discipline to automation scripts 将安全操作规程变为自动化脚本执行
- Implement strong version control on all code and components 对所有的代码和组件进行严格的版本管理
- Adopt an immutable infrastructure mind-set 接受不可变的基础架构的思维模式

DevOps和DevSecOps





DevOps进化



DevSecOps意义

- 团队或社区effor，而不是个人'
- 自治和自动安全 - >规模安全
- DevSecOps是一种基于DevOps原则的IT安全方法
- DevSecOps跨越整个IT堆栈
- DevSecOps还涵盖整个软件生命周期

信息安全架构师必须以协作的方式将多个安全点集成到DevOps工作流中，这对开发人员来说非常透明，并且保护团队的工作，灵活性和DevOps和敏捷开发环境的速度，这就是“DevSecOps”

DevSecOps中的Sec

- 业务安全工作

构建安全控制（认证，鉴权，攻击阻断，加密）和合规要求

- 内部安全工作

执行威胁建模，风险管理，资产识别，安全架构，安全研究，漏洞评估，安全监控和分析，安全工具和基础架构安全

- 运营安全工作

打补丁，操作系统更新，漏洞修复意见，产生分析报告，监控告警，工作问题单

- 其他安全工作

安全救火，紧急安全漏洞处理，恢复系统和对外工作处理

DevSecOps的挑战

- DevOps合规性是IT领导者最关心的问题，但信息安全被视为DevOps敏捷性的抑制因素
- 安全基础设施在成为“软件定义”和可编程的能力方面落后，因此难以以自动，透明的方式将安全控制集成到DevOps风格的工作流程中
- 现代应用程序主要是“组装”而非开发，开发人员经常下载和使用已知的易受攻击的开源组件和框架。

DevSecOps的挑战解决建议

- 从安全开发和培训开始，但不要让开发人员成为安全专家或交换工具
- 接受以人为本的安全概念，并让开发人员承担个人责任，通过监控补偿安全性。 拥抱“信任和验证”的心态。
- 要求所有信息安全平台通过API公开全部功能以实现自动化。

DevSecOps的核心实践

分析

- 识别你最关键的安全挑战

安全

- 针对这个挑战实施防御策略

校验

- 对这个挑战实现自动化验证

防御

- 对这个挑战检测攻击和防御注入

如何识别安全过程

首先考虑威胁

考虑数据类型和敏感程度

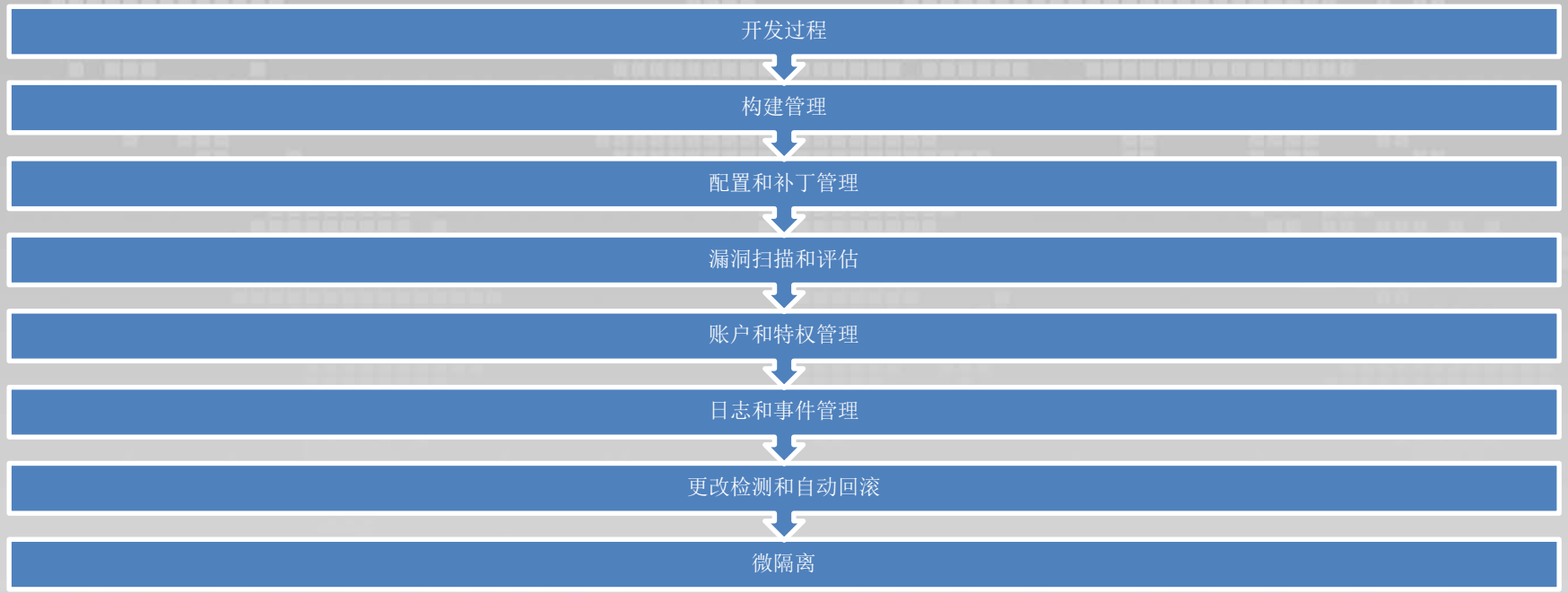
系统构建和控制

基于云的基础设施安全

退出控制

在云上遗忘的控制

把Sec集成到DevOps中



挑选安全工具的原则

- 政策范围
- 准确率
- 速度
- 弹性扩展
- 流程匹配度
- 集成

评估参考表格

项目	分数	项目	分数
CI/CD集成		支持多种格式的灵活报告	
自动化结果收集并且集成到运维工作		可以设置安全阀值终止流程	
安全专家提供支持		漏洞类型覆盖的广度	
通过检视结果降低误报率		目标语言覆盖的广度	
提供容易理解的结果报告			
减少识别漏洞和建议修复的时间			

DevSecOps和S-SDLC

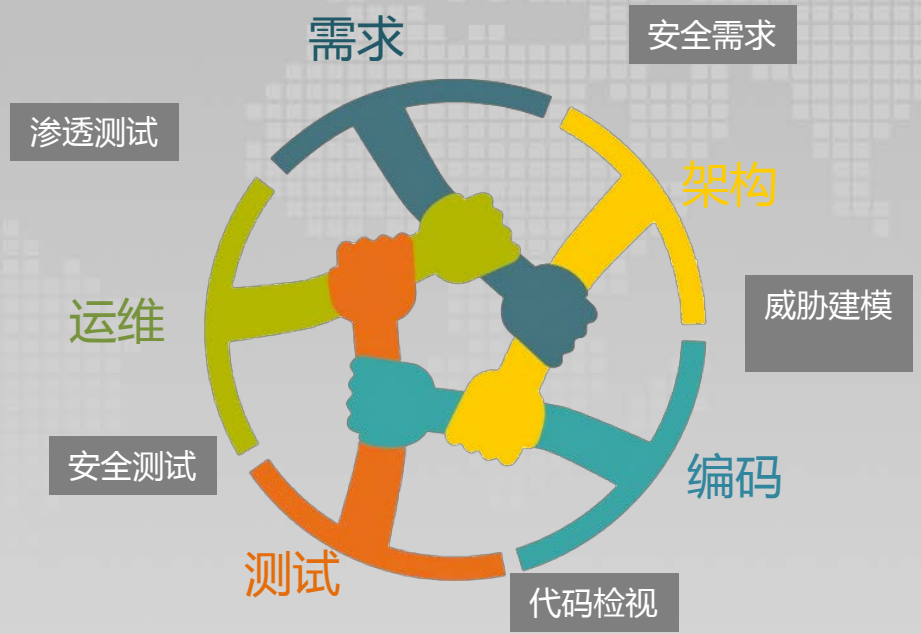
SDLC

S-SDLC

DevOps

DevSecOps

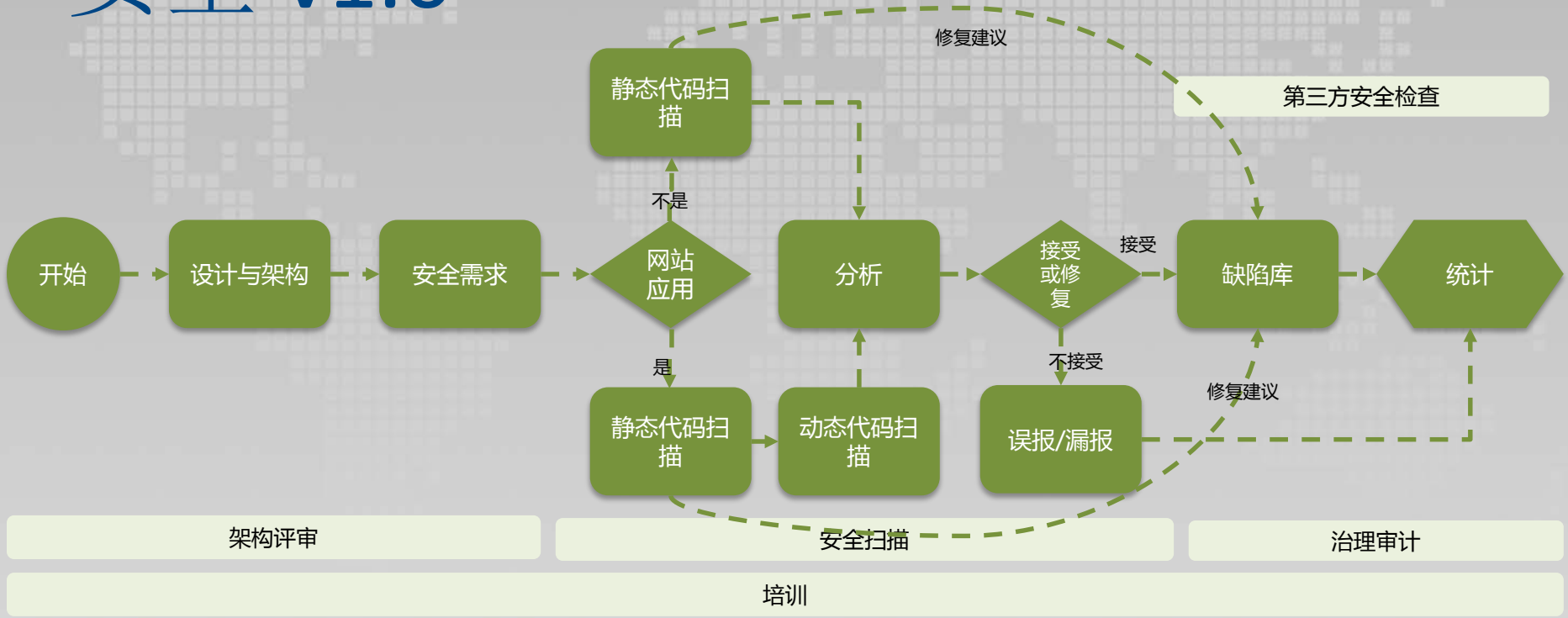
S-SDLC的具体实践



S-SDLC向DevSecOps演进

某跨国金融企业，研发中心遍及北美，中国，印度，欧洲，研发人员1000人左右，安全团队10-20人，研发产品包括客户端系统，移动端系统，网页端系统以及数据服务API。开发语言涉及C++，Java，.Net，Python，Scala。数据中心从自有向混合云架构部署。安全流程也从S-SDLC向DevSecOps过渡。

安全 v1.0


修复建议
静态代码扫描
第三方安全检查
不是
开始
设计与架构
安全需求
网站应用
分析
接受或修复
接受
缺陷库
统计
是
不接受
修复建议
静态代码扫描
动态代码扫描
误报/漏报
架构评审
安全扫描
治理审计
培训

架构评审

版本	描述	周期	缺陷	状态
1.0	架构评审文档Word版本	一个大的Release	很难分享	淘汰
1.1	架构评审文档Wiki版本	一个大的Release	比较容易分享	淘汰
1.2	专家系统 Json邮件版本	一个Sprint的周期	邮件比较麻烦	正在工作中
1.3	专家系统在线版本	一个Sprint的周期	需要安全专家检视	计划上线
2.0	专家系统API	实时		正在计划中

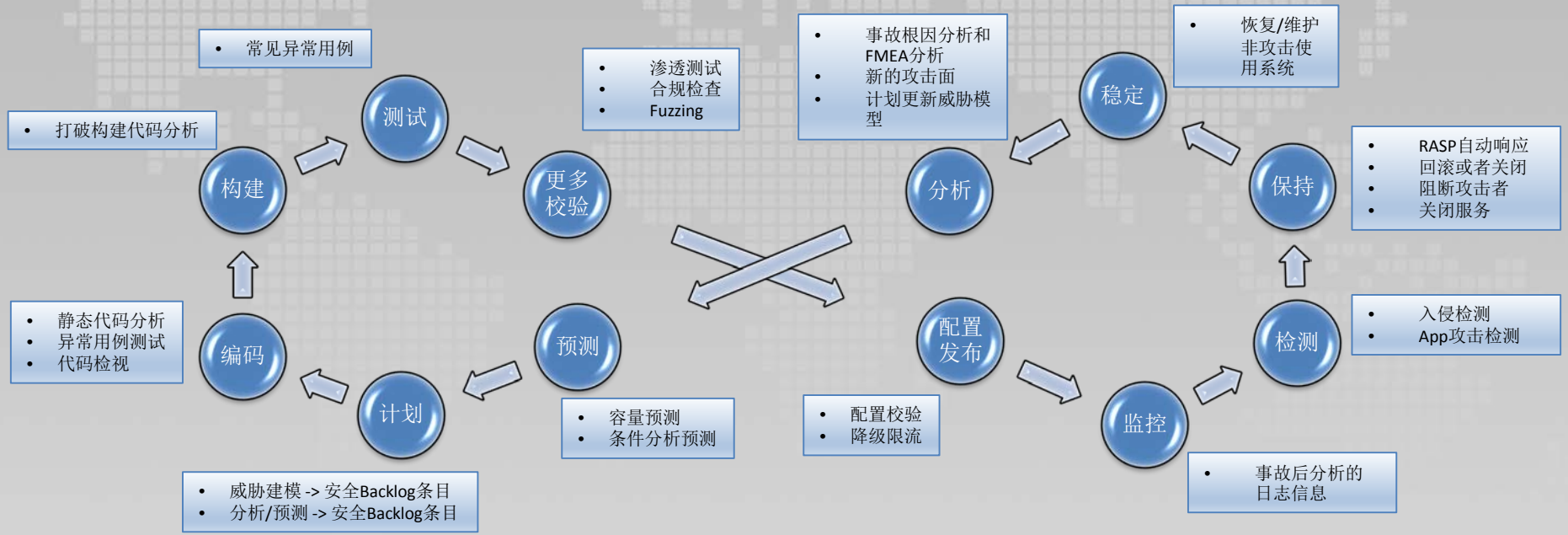
静态代码扫描

版本	描述	周期	缺陷	状态
1.0	Fortify	一个月一次扫描	基于代码编译，速度很慢，无法自服务	淘汰
1.1	Checkmarx	一周一次或者基于代码更新扫描	开发人员不能在本地扫描	正在使用
2.0	Checkmarx ++	实时		计划中

动态代码扫描

版本	描述	周期	缺陷	状态
1.0	Burp suite	一个Release之后	缺陷率比较高，与JIRA集成不好，无法自服务	淘汰
1.1	Whitehat	一个Sprint	没有API	正在运行
2.0	Whitehat ++	实时		正在计划

安全 v2.0



安全 v2.0进行中

阶段	状态	阶段	状态
计划	上个Sprint的问题会放到Blacklog中，出现的安全事故的解决方案会放到Blacklog中	监控	有完善的监控，有公司级的日志标准，日志数据会导入到Splunk平台
编码	使用Checkmarx进行静态代码扫描，代码采用Pull-Request模式，强制检视	检测	有入侵检测系统，有WAF防火墙
构建	无构建扫描，计划构建OSA扫描	保持	RASP正在计划中，通过WAF阻断攻击
测试	部分团队有进行安全测试，有使用Whitehat进行动态测试	稳定	有24x7团队恢复系统，根据ITIL流程建立起三级响应机制
更多校验	发布前使用GoASQ系统（已开源）进行专家检视	分析	对事故有根因分析，有必要会更新GoASQ的问卷系统
配置发布	有服务器扫描工具，会扫描服务器以及里面相关软件的配置，降级限流正在计划中	预测	有进行容量预测并及时扩容

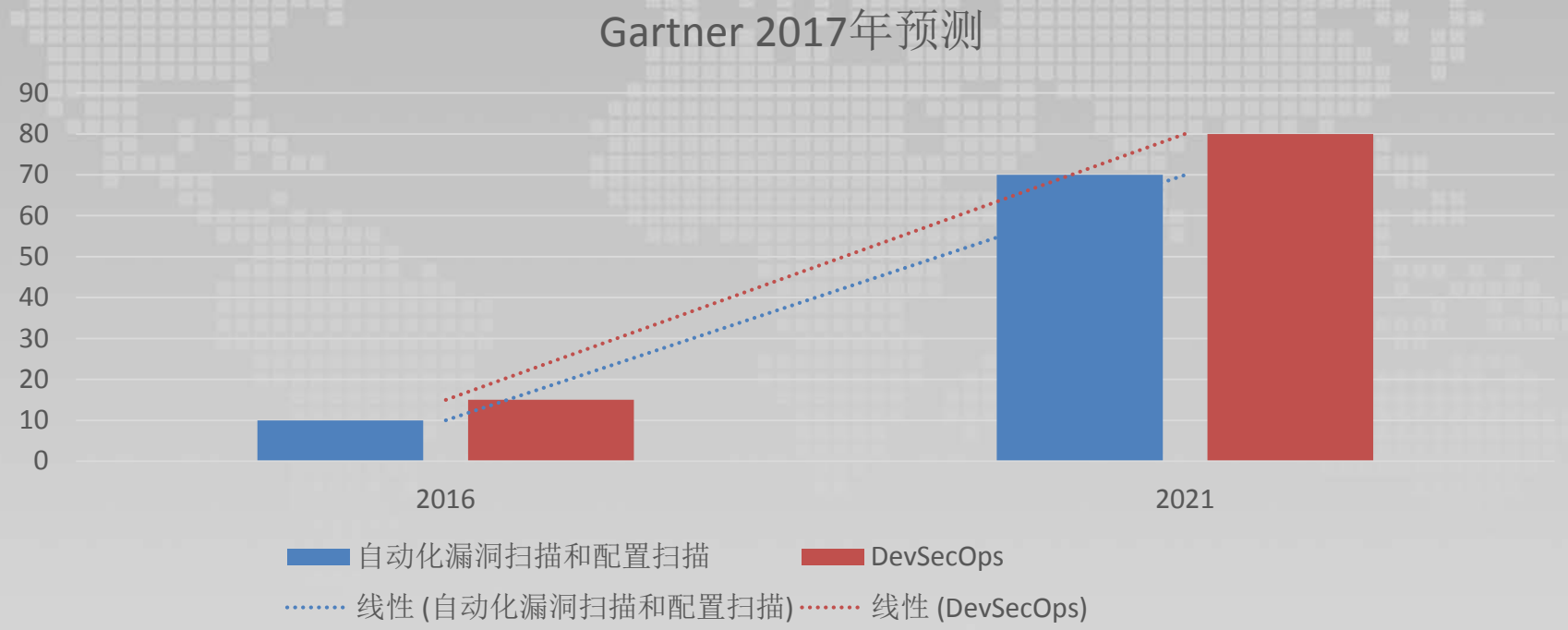
Gartner的九条建议实施情况

建议	状态	建议	状态
适应开发人员	共赢态度，安全开发一体，现在做得很多工作也是便利开发	从源头消除易受攻击组件	正在推行OSA扫描，对员工是信任的态度，没有做太多限制，但是上线前有足够的扫描进行保证
不追求过分完美	漏洞修复有优先级，Critical一个月内修复，High两个月内修复，非常合理	安全操作规程自动化	生成环境与开发环境分离，开发人员与发布人员分离，发布过程已经基本自动化和脚本化
关注识别删除已知漏洞	有团队跟踪已知漏洞，并及时打补丁，对于OWASP Top10问题，在开发阶段就会避免	对所有的代码和组件进行严格版本管理	代码是使用Git进行管理，组件管理放权到各个Scrum团队，但是有OSA扫描和其他扫描保证
DAST/SAST的使用	我们公司开发文化是敏捷文化，拥抱变化	接受不可变的基础架构思维	基于Amazon上推行Infrastructure-As-Code的思维，在自己的数据中心没有推行。
培训开发人员	新员工入职都会进行安全基础培训，另外有Whitehat的网上课程对开发人员进行培训		

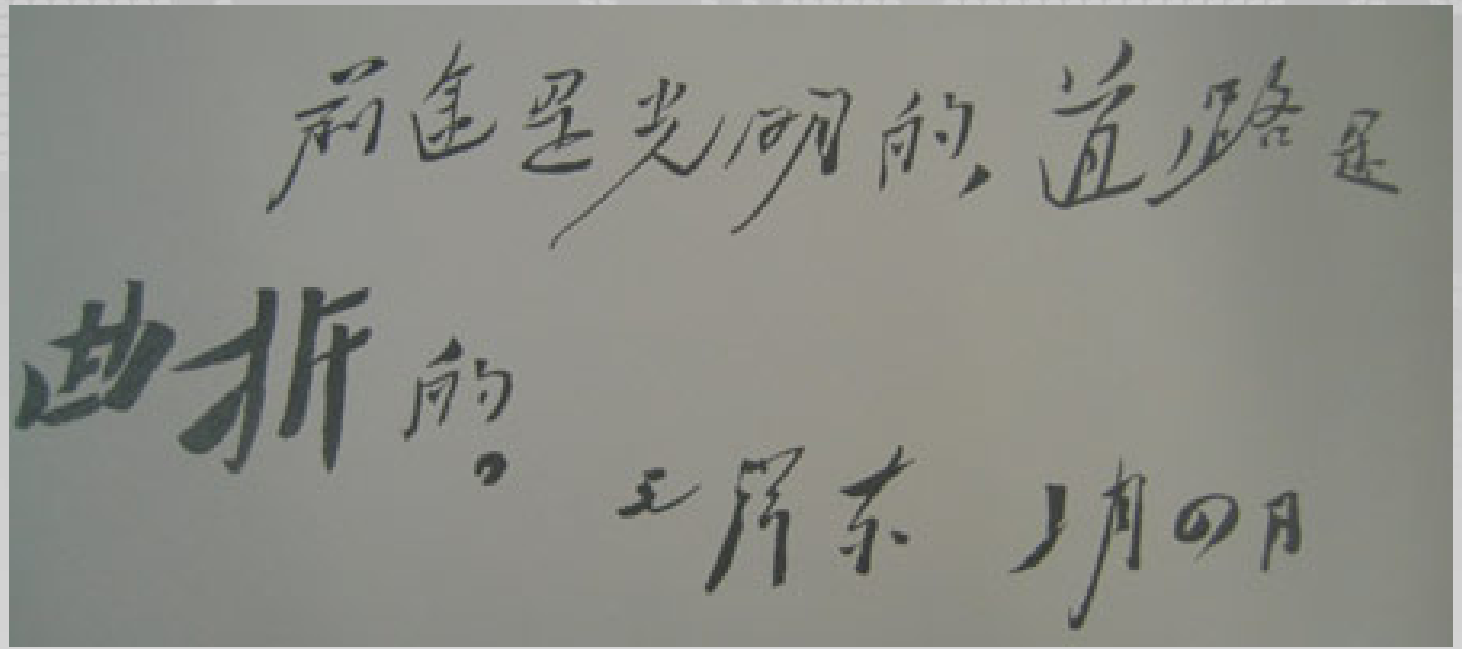
DevSecOps宣言检视

宣言	状态	宣言	状态
向前一步	公司是敏捷文化，拥抱变化	红蓝团队漏洞测试	现在没有做红蓝对抗，还是依赖于扫描以及专家系统检视
相信数据	公司建立了Dashboard以及评级标准，对全员公开，不害怕问题的暴露	24X7主动安全监控	有24X7团队监控系统，并且建立了WAF，IDP等一系列系统
贡献合作	强调共赢，开发，测试，运维，安全都一起努力做好产品	共享威胁情报	正在考虑建议威胁情报系统
基于API的安全服务	很多安全服务还没有完成API化，不能做到完全自助，还要努力	合规运营	公司遵循严格的ITIL流程，并定期审计
业务驱动的安全	安全是公司的生命线，行业有安全的核心需求，公司最高层非常重视安全		

Gartner关于DevSecOps的预测



DevSecOps的未来



晨星欢迎你


晨星Morningstar招聘
去这里
中国广东省深圳市福田区深南大
东海国际中心
车公庙
深航国际酒店
泰然四路
泰然工贸园
海松大厦
广华大厦
下沙


晨星资讯
“全球领先的独立投资研究机构”
投资/证券 / 500-1000人 / 海外、香港、深圳

附录

- GoASQ https://github.com/Morningstar/GoASQ
- 10 Things to Get Right for Successful DevSecOps

https://www.scribd.com/document/385486244/10-Things-to-Get-Right-for-Successful-DevSecOps