GitHub

企业架构师的DevSecOps指南

简介

企业DevOps已深入人心。但是企业的架构师其实早于DevOps的领导者明白安全性的意义，不仅仅只是交付，而是共同承担责任。

如今，运维团队使用协作、自动化和容器来加快软件交付速度。尽管这些DevOps最佳实践帮助他们找到了新方法来更快地构建，但是旧的安全实践仍然使许多组织减慢了速度。

走进DevSecOps。DevSecOps将IT安全性引入开发运维团队，以确保安全性在软件开发生命周期的每个步骤中都获得优先考虑。只需进行一些更改，贵组织就可以交付更好、更安全的软件，并且不会出现延迟交付或成本增加。

为什么需要DevSec-Ops？

降低安全成本

DevSecOps包含高效团队所遵循的所有DevOps最佳实践，以及大型组织需要的安全性。通过在DevOps管道中构建安全性，就有可能在发布之前发现漏洞，并且修复漏洞也会更容易，成本更低。

更有效的团队合作

就像开发人员和运维人员都要对DevOps的可靠性和质量负责一样，DevSecOps让安全性成为团队工作的一部分，而不仅仅只出现在最后一步。开发、运维和安全团队一起工作，确保应用程序从第一行代码到最终产品的安全性。

策略驱动自动化

良好的DevSecOps计划也能增强贵组织对整个软件交付过程的信心。自动化检查是以一种策略驱动的方式实现安全性，而不是使用各种令人困惑的手动工具来实现，这些工具会降低每个人的开发速度。

然后：

孤岛式安全性

部署前进行测试

静态测试和动态测试在交付周期最后阶段（即发布之前）执行。

安全知识孤岛

开发、IT运维和安全团队独立工作。

手动安全测试

组织部署频率较低，并仅根据需要单独进行安全检查。

现在：

DevSecOps

从构思到生产的全程测试

静态和动态测试与安全编码实践、质量关检查和安全漏洞修复同时进行。

共享的安全专业知识

开发、IT运维和安全团队在工作中都遵循共享的安全准则。

自动化安全性测试

组织部署频率更高，并在其持续集成/持续交付管道中增加自动化安全性检查。

开始DevSecOps的三个建议

1 使用共享、安全的协作平台

像DevOps一样，DevSecOps依赖于协作，并以协作结束。共享平台有助于开发、IT运维和安全团队共同构建，并标准化他们的工作方式。优先考虑具有内置安全机制的平台，这样整个组织就可以共享最佳实践、查找和复用代码，并从一开始就实现协作。

GITHUB TIP

良好的安全性从登录开始。您找到的合适的协作平台还应该支持身份管理功能，如双重身份认证、单点登录、自动组织同步等。

2 端到端开发安全性

最近发布的应用程序中，多达99%都包含开源代码，这意味着开源依赖项已成为代码库的一部分。*将代码安全工具集成到您的持续集成/持续交付管道中，可以主动识别开放源代码和内部源代码中的安全漏洞。

* 2019年开源安全性和风险分析报告

GITHUB TIP

开源软件随处可见。LGTM变体分析、WhiteSource和Snyk等自动化安全工具可轻松发现和消除您的团队无法手动跟踪的错误和漏洞。

3 产品发布后安全性跟踪

提交代码后，安全性工作并不会终止，您的DevSecOps管道也一样。部署完成后，通过持续监控漏洞来保证代码和客户的安全。寻找能够在软件发布后和潜在黑客可利用前，跟踪和更新易受攻击的依赖项的工具。

GITHUB TIP

尽管安全漏洞警报增强了项目的安全性，但行业数据显示，超过70%的漏洞在30天后（很多长达一年）仍未打补丁。使用集成，不仅可以识别易受攻击的依赖项，还可以自动修复它们。

对安全软件开发有疑问？

我们可以提供帮助。

访问github.com/enterprise了解更多信息