DevSecOps – 支柱 4

建立合规与发展的桥梁

SAFECode
Software Assurance Forum for Excellence in Code
Driving Security and Integrity

CSA GCR cloud security alliance® GREATER CHINA REGION

CSA cloud security alliance®

DeveSecOps 工作组的官方网址是：

https://cloudsecurityalliance.org/research/working-groups/devsecops/

联盟简介

云安全联盟(Cloud Security Alliance,CSA)是中立、权威的全球性非营利产业组织,于2009年正式成立,致力于定义和提高业界对云计算和下一代数字技术安全最佳实践的认识,推动数字安全产业全面发展。

云安全联盟大中华区(Cloud Security Alliance Greater China Region, CSA GCR)作为CSA全球四大区之一,2016年在香港独立注册,于2021年在中国登记注册,是网络安全领域首家在中国境内注册备案的国际NGO,旨在立足中国,连接全球,推动大中华区数字安全技术标准与产业的发展及国际合作。

我们的工作

联盟会刊下载地址
了解联盟更多信息

加入我们

CSA大中华区官网
(https://c-csa.cn)

点击会员

加入联盟

填写相关申请信息

成为CSA会员

JOIN US

致谢

《DevSecOps-支柱 4 建立合规与发展的桥梁（DevSecOps-Pillar4 Bridging Compliance and Development）》由 CSA 工作组专家编写，CSA 大中华区秘书处组织翻译并审校。

中文版翻译专家组（排名不分先后）：

组　长：李岩

翻译组：

车洵　　何国锋　　何伊圣　　贺志生　　黄鹏华　　江楠

苏泰泉　　余晓光

审校组：

何国锋　　江楠　　李岩

研究协调员：

卜宋博

感谢以下单位的支持与贡献：

中国电信股份有限公司研究院　　华为技术有限公司

腾讯云计算（北京）有限责任公司

英文版本编写专家

主要作者：

Souheil Moghnie　Theodore Niedzialkowski　Sam Sehgal

贡献者：

Michael Roza

CSA 分析师：

Sean Heide

特别感谢：

Ankur Gargi　Raj Handa　Manuel Ifland　John Martin　Kamran Sadique

Charanjeet Singh　Altaz Valani

在此感谢以上专家。如译文有不妥当之处，敬请读者联系 CSA GCR 秘书处给予雅正！ 联系邮箱 research@c-csa.cn； 国际云安全联盟 CSA 公众号。

序言

DevSecOps 是基于 DevOps 的安全敏捷化的一场变革，DevSecOps 的出现也改变了安全解决方案及安全合规的新思维。CSA 针对 DevSecOps 提出了六大支柱，分别为集体责任、培训和流程整合、实用的实施、建立合规与发展的桥梁、自动化、度量、监控、报告和行动等内容。

理想模式下 DevSecOps 世界中的合规意味着客户能够管理偏离安全基线的情况，并通过实时数据的自我修复功能。DevSecOps 在实现速度和安全优先的同时，实现具有更加高效、更安全的持续交付。在 DevOps 名著《DevOps HandBook》中就指出测量对 DevOps 实践合规性的重要性。

本白皮书以合规与发展为核心，提出在 DevSecOps 模式中的合规性目标是提高应用程序及环境的整体安全性，同时减少风险，以安全目标来验证持续交付。

DevSecOps 也是 CSA 高级云安全专家课程（CSA ACSE）的核心内容，DevSecOps 是践行共享安全责任的文化表现，实现满足企业对监管或行业合规标准的管理要求。尤其是很多企业通过了 ISO/IEC27001、CSA Star 等认证。通过学习 CSA DevSecOps 合规与发展，帮助企业提升数字化合规的能力，实现基于风险的安全合规的新方案。

李雨航 Yale Li

CSA 大中华区主席兼研究院院长

目录

前言

云安全联盟和 SAFECode 都致力于提高软件安全成果。2019 年 8 月发布的文章《DevSecOps 的六大支柱》[1]提供了一组高阶方法以及由其作者成功实施过的解决方案，以快速构建软件并将与安全相关的错误降至最低。这六大支柱是：

支柱 1：集体责任（2020.02.20 发布）[2]

支柱 2：培训和流程整合

支柱 3：实用的实施

支柱 4：建立合规与发展的桥梁 （2022.02.03 发布）

支柱 5：自动化（2022.07.06 发布）[3]

支柱 6：度量、监控、报告和行动

以支撑上述六大支柱的成功解决方案为主题，云安全联盟和 SAFECode 联合发布了一组更详细的出版物。本文是后续六份出版物中的第三篇。

[1] Cloud Security Alliance. (2019, August 7). Six Pillars of DevSecOps. https://cloudsecurityalliance.org/artifacts/six-pillars-of-devsecops/

[2] Cloud Security Alliance. (2021a, February 21). The Six Pillars of DevSecOps: Collective Responsibility. https://cloudsecurityalliance.org/artifacts/devsecops-collective-responsibility/

[3] Cloud Security Alliance. (2020b, July 6). The Six Pillars of DevSecOps: Automation. https://cloudsecurityalliance.org/artifacts/devsecops-automation/

1 简介

鉴于软件开发范式和实践的快速发展，整体安全合规活动与软件开发过程的结合已成为一项挑战。合规团队已经习惯于依靠流程和控制到位来证明（安全性）。然而，大多数认同 DevOps 观点的工程师认为证明应在代码中，而不在流程或文档内。

DevSecOps 实践旨在结合合规性与开发，需要安全团队和软件开发人员之间的协作（集体责任[4]）努力。DevSecOps 模式中的合规性目标是提高应用程序或环境的整体安全性，同时减少验证系统达到安全目标及合规性的所需工作量。

本文探讨的方法允许 DevSecOps 团队将安全性和合规性要求转化融合到开发周期中，达到以下目标：

- 软件开发人员可操作性
- 客观可度量
- 实用性的降低风险

本文还探讨了安全和开发团队进行系统性协作的要求、方法和建议，分为三个部分，如图 1 所示。合规性和开发功能（特性）应考虑以下要素：

- **评估：**一种划分和评估的方法，对运营影响最小
- **思维方式：**关于如何把合规性设计并实施到应用程序中的思想和实践转变
- **工具：**安全工具的不同实践，可以为合规性要求提供保证

本文和图 1 中对利益相关者的引用有两个角色，“合规”和“开发”：

- **“合规”**被确定为对监管或行业合规标准的管理，这些标准被下发到信息安全—以及法律、风险和审计—等团队，以形成塑造组织业务运营的要求和政策。
- **“开发”**是对应用程序的设计、配置和构建有影响的工程师和产品团队成员（包括开发人员、平台工程师、架构师和业务分析师）。

[4] Cloud Security Alliance. (2021a, February 21). The Six Pillars of DevSecOps: Collective Responsibility. https://cloudsecurityalliance.org/artifacts/devsecops-collective-responsibility/

图 1.连接合规与开发的框架

1.1 目标

本文提供了一些指导，即确保通过识别合规性目标，将它们转化成适当的安全措施，并通过将安全控制措施以自动化测量测试等方式清晰透明、易于理解地嵌入到软件开发生命周期中的关键节点，以弥合合规性与开发之间的差距。

本白皮书未确定合规性目标的来源，且不比较 DevSecOps 的标准或指南。（说明：合规性可以来自企业内部也可以来自企业外部）

1.2 读者群体

本文的目标读者包括涉及安全风险、信息安全和信息技术的管理和运营岗位工作人员。包括 CISO、CIO，尤其是涉及以下职能领域的个人：自动化、DevOps、质量保证、信息安全、治理、风险管理、内部审计和合规。

2 评估

评估通常是测量 DevSecOps 流程和控制的成熟度和有效性的第一步。组织评估其应用程序和 DevSecOps 应用情况的主要考虑因素是：

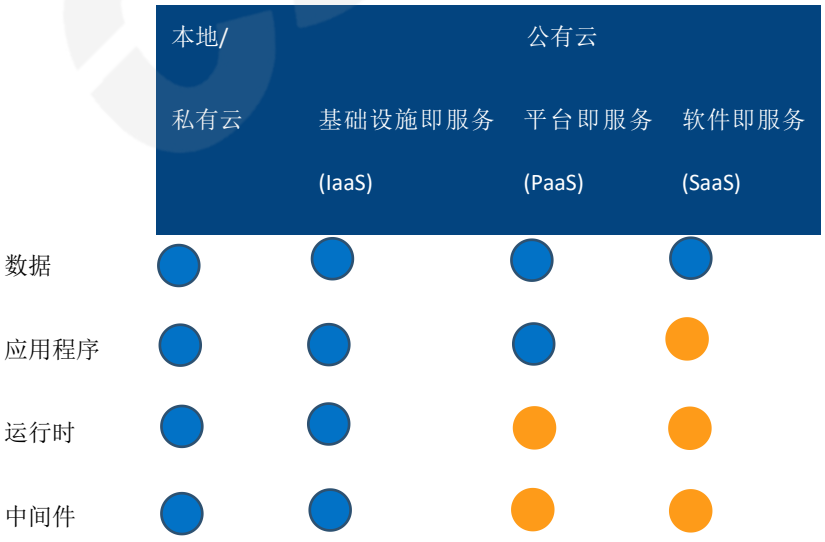
- 与云服务提供商的共担责任：确定风险转移的位置，明确云客户应在何处应用控制或寻求保障。
- 即时评估与持续评估：确定适当评估的方法，研究如何实现自动进行持续审查以提高调查结果的准确性。

2.1 与云服务提供商的共担责任

在大多数应用程序的部署使用中，云环境的设计和操作都对 DevOps 和站点可靠性工程 (Site Reliability Engineering) 至关重要。云安全取决于基础设施，因此云安全控制和责任共担管理在 DevSecOps 实践中势在必行。

当企业将合规性目标映射到安全要求时，了解云客户在选择解决方案和技术时的责任非常关键。安全工具和解决方案必须与技术保持一致，如容器化工作负载、虚拟机、云原生平台服务的配置状态。

云服务提供商 (CSP) 和云服务客户 (CSC) 应在服务级别协议 (SLA) 中同意并记录共同责任，这样 CSC 能够明确了解 CSP 的服务条款、优势和缺陷。 CSP 通常会公开提供此信息。图 2 中的通用模型展示了 CSC 和 CSP 之间从本地部署到 SaaS 的职责划分范围。



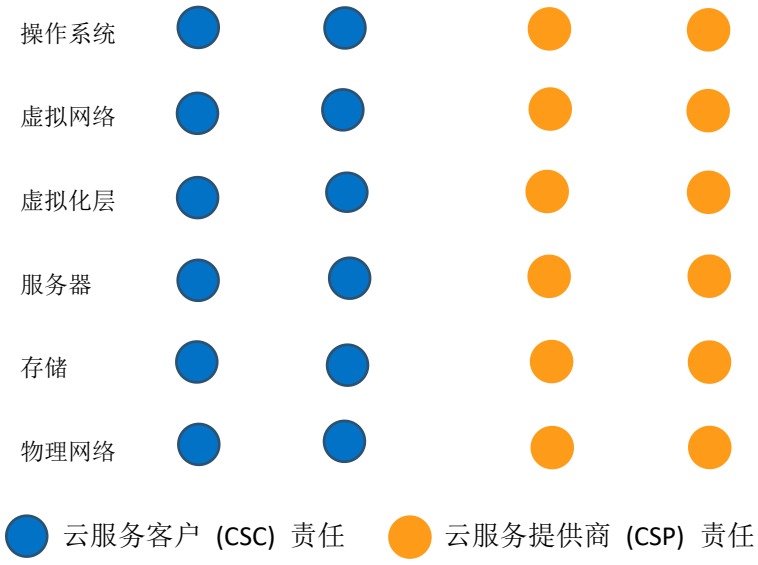


图 2 混合云中的责任共担[5]

DevSecOps 活动应反映出图 2 中的责任共担模型。例如，提供 IaaS 服务的 CSP 可以确保堆栈较低层（例如，hypervisor 及以下）的安全，而较高层（例如，网络层及以上）的安全则由 CSC 负责。

一旦组织能够识别他们所使用的服务和不同的云部署模型，他们就可以开始考虑“下一步”的安全活动。

2.1.1 下一步

在大多数情况下，云客户负责应用程序安全控制和云环境的管理。开发团队应该传达他们对使用云服务的应用程序的需求——应该确定依赖 CSP 的控制、可以评估 CSC 要求，并将其打包加入补救的活动/迭代阶段中。

如果责任属于 CSP，CSC 应获取第三方对 CSP 的评估保证，如 SOC 2[6]报告或 CSA STAR 认证。云客户应审视 CSP 提供的保证，以确定控制和流程的适当性，并在需要时实施补偿控制。

安全和合规职能团队应与开发团队合作，根据云部署模型准确地识别职责。安全控制和流程应该由安全和合规职能团队根据现有的组织策略、安全框架、检查表和模板周期性开展审查。如果责任属于 CSP，合规团队可以审查 SOC 2 报告或 CSA STAR，以确定 CSP 采取了的

[5] Cloud Security Alliance. (2020c, July 13). Hybrid Cloud and Its Associated Risks. https://cloudsecurityalliance.org/artifacts/hybrid-clouds-and-its-associated-risks/

[6] American Institute of CPAs. (n.d.-b). SOC for Service Organizations. AICPA. Retrieved December 22, 2021, from https://www.aicpa.org/interestareas/frc/assuranceadvisoryservices/socforserviceorganizations.html

适当控制措施。

2.2 即时评估和持续评估

评估是所有组织用于理解当前控制措施和流程的成熟度和有效性的通用方法。然而，伴随着敏捷或 DevOps 方法中快速的变化和部署速度，人们对评估的频率和相关性的提出了疑问。

即时评估[7]可能很慢，可能需要几个星期甚至几个月。从开始评估到完成报告的期间开发和部署环境可能不断变化。与此同时，即时评估可以提供一个应用程序及其基础设施安全状况的整体视图，并可以与其他活动，例如渗透测试和隐私数据审查等相结合。

为应用程序和基础设施引入安全开发人员工具可以帮助对组件进行分类，以便单独审查。在开发和部署阶段就可以识别和纠正问题，这消除了对即时评估的依赖。以下工具的使用，举例说明了如何为持续评估对应用程序进行分类和扫描：

- **软件组成分析：**识别、报告、修复和标准化高风险开源工具使用的机会。这可以在构建阶段进行应用及处理，并在即时评估之前达到合规要求。
- **静态分析：**一个扫描源代码的弱点和漏洞的机会。这将帮助开发人员在构建阶段编写安全的代码，并对大型代码库扫描发现的问题进行补救。
- **基础设施即代码（IaC）分析：**提供了根据云合规标准扫描代码的云构建（coded cloud builds）的机会。这有助于工程师在构建阶段解决云上的安全性问题，而不是依赖即时评估。
- **云态势管理：**一种针对行业标准和指南，全面而持续地对云构建和环境进行扫描的通用方法。云态势管理甚至可以取代即时评估。

要在构建时进行扫描和执行整体评估之间取得合适的平衡，取决于团队如何在不影响部署速度的情况下快速地完成完整扫描。

代码扫描（例如，IaC 分析）对于在构建阶段对实体（应用程序/平台/基础设施）的审查是很有效的，并且可以降低漏洞和安全弱点被写入代码的可能性。这将帮助组织实现安全

[7] SecurityScorecard. (2018, February 15). Limitations of Point-in-Time IT Security Risk Assessments. https://securityscorecard.com/blog/limitations-of-it-security-risk-assessment

左移，减少部署后的安全补救工作。然而，需要理解的是，与部署后扫描（例如，云态势管理）相比，在构建时进行扫描并不总能得到完整和全面的发现。

一个说明代码扫描完整性的例子是 Bridgecrew（一家 IaC 供应商）对其开源 IaC 分析工具“Checkov”的研究。Bridgecrew 透露，只有大约一半的行业云安全基准控制（由互联网安全中心 (CIS) 推荐）在 IaC 分析中得到了解决（见图 3）。这并非反映 “Checkov”的质量，而是反映了主要 CSP 在构建阶段的安全问题规模以及评估的完整性。

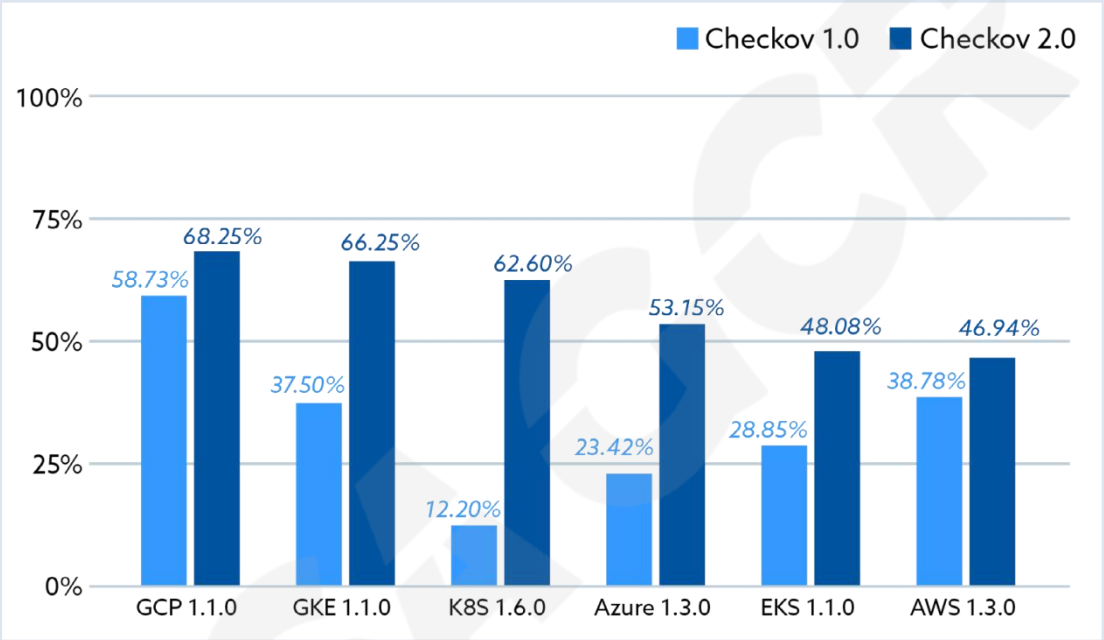


图 3. Checkov CIS 基准测试覆盖率[8]

代码扫描的分类并不能取代对即时评估的需求；然而，这确实减少了对即时评估的依赖。组织现在可以在构建时解决问题，比在部署到生产环境前评估中发现的严重安全问题更为有利。

当组织应用方法来执行持续的 DevSecOps 评估时，可以利用即时评估提供完整的、统一的、完全集成的审查，同时知道在构建过程中已经解决了大量安全问题。

[8] Johnson, M. (2021, April 8). Checkov 2.0: Deeper, broader, and faster IaC scanning. Bridgecrew. https://bridgecrew.io/blog/checkov-2-0-release/

3 心态

组织可以通过工具直接快速地解决风险，但是他们很容易忽略适当的心态（思维模式）在 DevSecOps 转型中的重要性。心态是一种方法，以一组活动的形式由安全和产品的利益相关方（例如，软件开发人员）所采用，通常可以使两个孤立的团队更紧密地联系在一起。有助于弥合安全和软件开发之间价值冲突的关键考虑因素有：

- **价值流映射（VSM）：**确定团队、交付时间和处理时间，以了解工作流如何从想法变成客户输出。这提供了一个通过手动或自动化来识别安全参与的机会。
- **将合规目标转换为安全措施：**如何将目标打包成供开发人员使用的安全措施。
- **流水线监控：**在不妨碍生产力的情况下对开发人员活动的控制措施进行监控和维护。

3.1 使用价值流映射的合规性

虽然合规和开发活动可以在它们的安全目标上保持一致，但传统的合规方式——冗长的即时评估——严重依赖于文档。传统方式与 DevSecOps 的快速、敏捷工作和持续交付的特性不能很好匹配。

因此，在考虑应用安全控制之前，了解工作在应用程序开发生命周期中的工作流程非常重要。价值流映射（VSM）是一项用于理解应用程序上下文的有用技术。如图 4 所示，VSM 可以包括以下组件：

- **利益相关方：**负责的团队和个人。
- **活动：**在某阶段执行的通用活动。
- **交付时间：**从一个工序接收一件工作到把该工作交给下一个下游工序的时间。
- **处理时间：**指完成一项工作所需的时间，前提是执行这项工作的人拥有完成这项工作所需的所有必要信息和资源，并且可以不间断地工作。
- **准确完成百分比（C/A）：**上游流程接收到可以使用且无需返工的内容的百分比。

图 4. VSM 示意[9]

通过了解 VSM 过程中应用程序的构建和部署活动，组织可以开始识别和规划适当的安全活动和工具。DevSecOps 是一种共同承担的责任[10]，建立在将安全实践集成到整个构建阶段和部署流水线的概念之上（如下图 5 所示）。

[9] Google. (n.d.). DevOps process: Visibility of work in the value stream. Google Cloud. Retrieved December 22, 2021, from https://cloud.google.com/architecture/devops/devops-process-work-visibility-in-value-stream

[10] Cloud Security Alliance. (2021a, February 21). The Six Pillars of DevSecOps: Collective Responsibility. https://cloudsecurityalliance.org/artifacts/devsecops-collective-responsibility/

虽然这个图像给人一种从一个阶段到另一个阶段的线性流动的感觉，但各阶段之间存在双向反馈机制

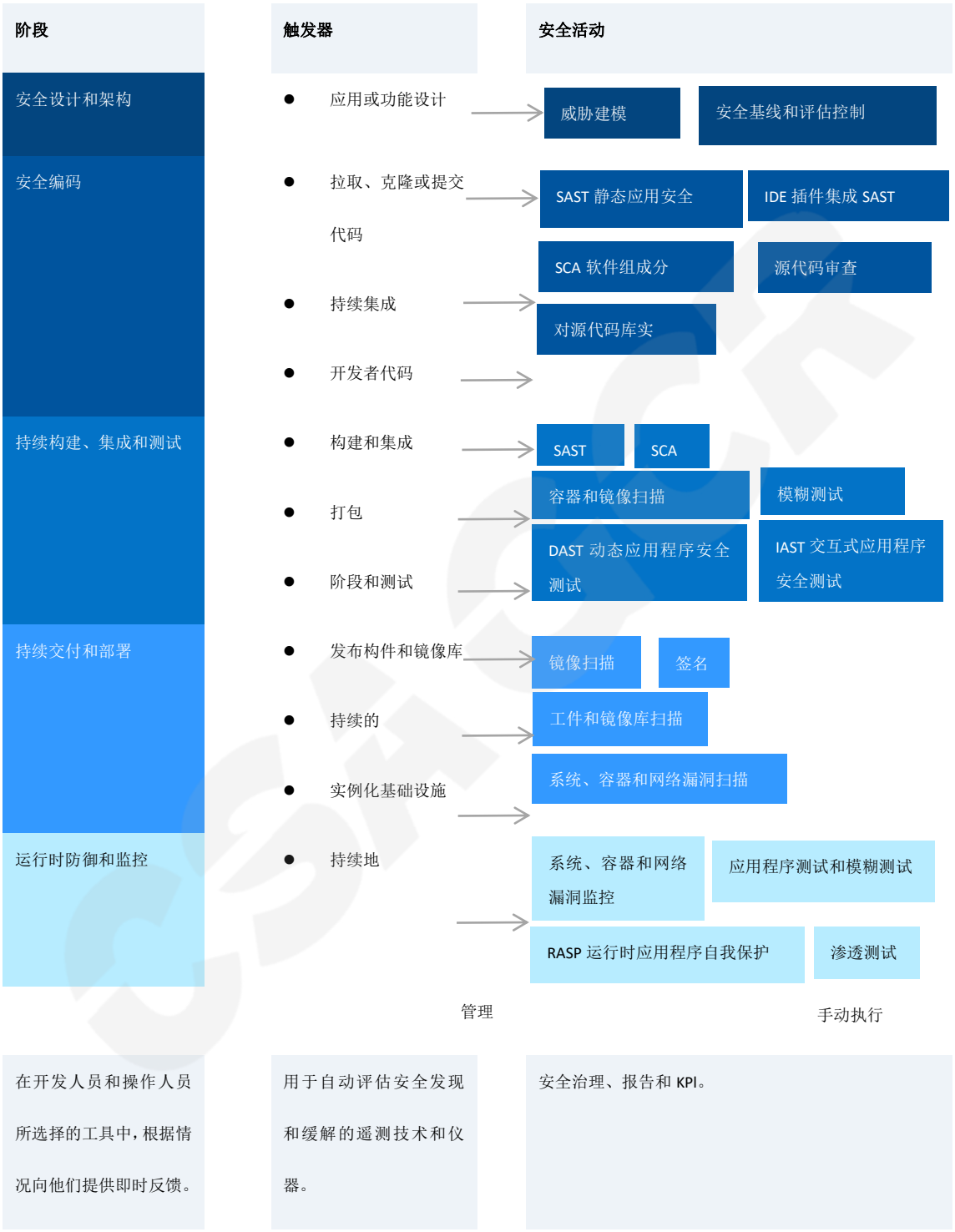


图 5. CSA 的 DevSecOps 交付流程[11]

通常安全和合规性活动是手动进行的，并形成安全质量门禁，这使得过程繁重并有可能

[11] Cloud Security Alliance. (2020b, July 6). The Six Pillars of DevSecOps: Automation. https://cloudsecurityalliance.org/artifacts/devsecops-automation/

影响到交付时间。为了应对这一挑战，合规性必须是“持续的”，这样才能不断审查安全性并将其集成到代码中。 作为交付流水线的一部分，证明符合共识的安全要求的能力将安全性集成到整个开发过程中。（注：短语“organized into gates”指的是软件开发中的一种做法，即将安全和合规性活动作为检查点或“门禁”进行组织，开发项目必须通过这些门才能进入下一个开发阶段。）

VSM 的过程可以借助测量每个安全措施的工作量和运营影响，来实现这一目标。通过识别 VSM 中的端到端流程，组织可以将合规性/安全性要求包含在需要改进、进一步控制和自动化的领域。

一旦价值流映射得到优化，组织就可以开始使用静态分析、策略即代码和依赖性检查等方法来实现自动化合规。本文探讨了将自主安全措施嵌入在交付流程中的不同方法，这可以帮助弥合合规和开发之间的鸿沟。

DevSecOps 交付流程本身是将安全应用于产品的基础。组织应该寻求在图 5 的 “安全活动”部分建立实践，持续支持架构工件（如加固指南和模式），以提供评估的参考点。

3.2 合规目标转化为安全措施

DevSecOps 的一个目标是“左移”安全，如将关键的安全控制作为应用程序开发过程的一个组成部分，而不是采取在交付点进行回顾性的审计活动的传统方法。

合规性的目标、政策、流程和控制通常源于合规框架、治理、风险和信息安全团队。对于大多数将合规与行业指南（例如 ISO2700[12]1 或 NIST CSF[13]）相匹配的具有成熟安全功能的组织来说，并不是新鲜事。

从传统的合规性过渡到更加动态的方法，需要打破安全合规和软件开发团队之间的隔阂（包括采用 DevOps 和网站可靠性工程的团队），将对 DevSecOps 的理解作为一种新的方法论融入文化和技术[14]。

---

[12] International Organization for Standardization. (n.d.). ISO/IEC 27001 — Information security management. ISO. Retrieved December 22, 2021, from https://www.iso.org/isoiec-27001-information-security.html

[13] National Institute of Standards and Technology. (2021, October 26). Cybersecurity Framework. NIST. https://www.nist.gov/cyberframework

[14] Cloud Security Alliance. (2021a, February 21). The Six Pillars of DevSecOps: Collective Responsibility. https://cloudsecurityalliance.org/artifacts/devsecops-collective-responsibility/

安全和合规利益相关者（例如信息安全）、架构和软件开发团队应紧密合作，定义明确的合规目标。在实践安全活动和达成合规目标的控制之间应有精确的映射。这些应被整合到交付流程中，并考虑生成自动和公开的报告。

可以让安全和合规性利益相关者参与到正在进行项目的迭代会议实现安全目标到安全措施的转化：

- 计划会议以确定整个应用开发过程中的必要工具和安全要求/控制；
- 提供对安全/控制变化迭代的更新建议，以帮助将合规要求引入应用程序的设计和构建中；
- 确定应用程序是否需要额外的安全措施。

为软件开发人员建立安全合规迭代和用户需求是将安全纳入软件开发的有效方法。现有的安全合规要求、组织政策、行业准则和标准可以作为参考，这需要项目利益相关者之间的合作才能实现。类似 INVEST[15]这样的方法可以帮助创建用户安全需求和任务，这些任务应该是：

- **独立**：应该是自成一体的
- **可商议**：应该留有讨论空间
- **有价值**：必须为利益相关者提供价值。
- **可估计**：应该能够估计规模。
- **足够小**：能够确定计划、任务和优先级
- **可测试**：应该能够被测试。

通过内部审计和治理，可以更好地向业务利益相关者展示安全合规用户需求、迭代和任务被设计到代码并配置到组织批准的安全控件中。

[15] Agile Alliance. (2021, July 26). What does INVEST Stand For? https://www.agilealliance.org/glossary/invest/

图 6. INVEST 用户需求[16]

在合规性和开发人员之间建立的桥梁不应该是单向的。安全策略应该映射到开发人员和操作人员可以完成的原子级技术需求。软件开发人员、架构师和 DevOps 工程师应及时提供完成状态的反馈，以便编写信息安全策略的安全合规团队能够了解 SDLC 当前阶段或迭代的安全态势。创建安全模式和加固指引将有力支持安全合规活动。（注： INVEST 用户需求指的是应用敏捷用户故事的方法针对用户提出的需求，在进行需求梳理时需要左移考虑安全的问题。）

3.2.1 流水线监测

软件开发人员组可以在短时间内生成大量代码和应用程序更新，并迅速将它们提交到生产环境。这些活动为经常部署软件功能的组织创造了巨大的业务优势，通常分类如下：

- **持续集成（CI）：**频繁提交和合并新代码和代码更改到编码存储库，这些存储库通过创建构建并针对该构建运行自动化测试来验证。
- **持续部署（CD）：**将代码自动部署到环境中的过程，包括自动扫描、测试和构建活动。

尽管执行 DevOps 的开发团队间独立自主和责任分担为质量和速度方面带来了益处，但

[16] S. (2019, June 7). How INVEST helps team write effective user stories. ... Medium. https://sabaimam.medium.com/investing-in-user-stories-c7cfb1fc5e85

如果缺乏对角色的控制则会带来安全风险。例如连续的快节奏活动增加了人为错误和恶意内部攻击的风险，例如，糟糕的编码实践或可利用的逻辑漏洞。这就是为什么 CI/CD 流水线应该包括所有操作的清晰和不可变更的审计跟踪。

审计跟踪通常在 CSP 环境中本地提供。在部署代码并嵌入度量以确保不可抵赖性时，证明操作的痕迹是至关重要的。

组织的目标应该是实现对应用程序和 CI/CD 流水线上的操作的监控。可以包括以下步骤/操作：

- 在构建和集成时，应建立接入控制，以防止未经授权合并到主分支或生产环境。用户不应有权限修改或覆盖接入控制策略。部署应该只引用被批准的主分支。
- 应商定并实施日志保留、轮换、存档和删除的策略。在可能的情况下，日志应归档在不同的环境中以便出于法律目的保留。
- 为了简化审查和管理审计跟踪，应根据事件标记日志并集中汇总，以减少人为错误并提高审查日志的速度。标签是根据事件的修改、部署和相关环境创建的。
- 为维护日志完整性，访问日志数据的用户账户应为只读且防篡改，并根据具体情况授予人员访问权限。识别和监控可以防止识别篡改日志的恶意活动。这也可以通过使用一次写入多次读取（WORM）兼容的存储机制来实现，该机制可以防止数据被删除或覆盖。

4 工具

工具通常是在应用安全控制和措施时发挥价值的。工具可以帮助引入安全检查、扫描和数据管理，并在部署流水线中使用触发器实现自动化——这些工具可以是专有的或开源的。工具采购时可以考虑以下关键因素：

- 采用“as-Code”模型；
- 采用 DevSecOps 方法进行测试；
- 跟踪开源风险；
- 安全护栏；
- 模式和模板。

4.1 拥抱即代码 as-Code 模型

代码质量不仅仅与软件开发人员有关；可以使用工具来提高质量并将策略检查应用于集成开发环境（IDE）中的代码，这些代码可以应用于源代码和基础设施即代码（IaC）。

基础设施即代码（IaC）消除了通过控制台和手动任务提供的传统基础设施，而是通过代码提供基础设施。IaC 可通过云服务提供商（CSP）（例如 AWSCloudFormation）的原生工具或独立第三方供应商 IaC 功能（例如 Chef、Ansible、Terraform）实现。IaC 的使用包括自动化、版本控制和治理。

此外，当与合规性即代码（CaC）/政策即代码（PaC）相结合时，组织可以将其合规性要求直接融入其 IaC 模板和清单中。这些合规性要求可能来自适用的监管框架、组织安全策略或两者的组合。

随着云部署变得愈发自动化和灵活，使用 Terraform、Puppet 和 Chef 等工具，在 IaC 上针对云的“合规即代码”实践已成为部署云资源的实际方法。

如图 7 所示，IaC 是使用可编程基础设施大规模自主构建云和本地环境的过程。可以将 IaC 代码视如源代码一样，这样它就可以在存储库中进行版本控制、测试和管理。

图 7.作为代码的基础设施工作操作流[17]

作为代码的合规性可以使用 IaC 的工具和插件来实现，其中云资源在脚本中配置并部署在多个环境中。IaC 安全插件可以识别基础设施（例如 Chef、Puppet、Terraform、Ansible）并部署相应的安全分析以促进正确的编码语法和安全代码（例如 ansible-lint、cfn_nag、cookstyle、Foodcritic、puppet-lint、Checkov、Terrascan）。

IaC 的开放策略代理（OPA）用于创建和修改云环境，并为 OPA 期望基础设施即代码的行为方式定义了一组策略。OPA 中设置的策略为软件开发人员、DevOps/DevSecOps 工程师和架构师提供了机会，以证明云上的代码符合记录的息安全策略（即，有机会在代码中执行自动检查而不是手动审查）。

通过将合规性构建到 DevOps 流水线中，产品团队可以提高灵活性，同时减少重复工作

[17] Velimirovic, A. (2020, September 9). What Is Infrastructure as Code? Benefits, Best Practices, & Tools. PhoenixNAP Blog. https://phoenixnap.com/blog/infrastructure-as-code-best-practicestools

以提供对合规性违规的持续检测和补救[18]。

CaC/PaC 也可以通过集成开发环境（IDE）在源代码级别实现。IDE 可以通过“linting”实现这一点——使用静态代码分析工具来标记编程错误——使用 SonarLint、DevSkim、OWASP FindSecurityBugs 和 Puma Scan 等工具。尽管在 IDE 中，组织的信息安全策略并不完全适用于检测工具，但在开发人员编写代码时，检测确实会执行质量审查并识别潜在的错误和重复。

IDE 安全插件旨在识别源代码开发过程中的安全漏洞和修复程序，让安全性向软件开发人员左移。图 4-2 表示 IDE 安全控制的门控方法，以及如何建立、审查功能分支（重复代码文件）并将其合并到主分支以进行部署。在这一点上，代码中的安全错误可以在开发的早期识别出来，而不是在它们更临近影响发布的时间窗口。

图 8.IDE 插件[19]

采用 PaC/CaC 有助于从开发生命周期一开始就引入合规性，实施安全护栏可确保基础设施和源代码是合规和安全的。这种方法还使开发人员和运营人员协作使合规活动更有效，进一步促进了 DevSecOps 的概念。

4.2 拥抱 DevSecOps 方法进行测试

DevSecOps 的连续性——持续测试——要求测试以比传统应用程序更加分段和频繁的方式执行。测试不再仅仅是应用程序发布到生产环境时的勾选项。

测试适用于应用程序的功能，通常被分解为需求。保持与应用程序的更改和部署一致的

[18] Cloud Security Alliance. (2020b, July 6). The Six Pillars of DevSecOps: Automation. https://cloudsecurityalliance.org/artifacts/devsecops-automation/

[19] SonarQube. (n.d.). SonarLint Integration. Retrieved December 22, 2021, from https://www.sonarqube.org/sonarlint/

测试频率非常重要。

提交到仓库的代码通常在持续集成（CI）工具中运行，并接受自动化测试。充分的自动化测试可以降低与部署新代码变更相关的风险，并减低代码回归或引入新缺陷的可能。

DORA[20]研究小组的四个关键指标表明，部署频率和变更准备时间与自动化测试套件的有效性相关。

- **部署频率（Deployment Frequency）：**组织成功发布到生产环境中的频率
- **变更的提前期（Lead Time for Changes）：**提交到生产环境所需的时间
- **变更失败率（Change Failure Rate）：**导致生产环境失败的部署百分比
- **服务恢复时间（Time to Restore Service）：**从生产环境中故障恢复时间

测试自动化提高了频率并缩短了变更的交付时间，这有助于降低回归缺陷的风险，并通过对攻击地图和发现漏洞威胁的更快响应来降低风险。

下表 4-1 中的方法和用例通常在软件开发中被考虑

测试类型	阶段	频率
安全冒烟和单元测试	在开发应用程序时，对源代码的小单元和应用集成部分进行测试，从而更早地发现缺陷，并以更低的成本进行补救。通常由开发人员作为一项活动执行，因为代码是在部署前生成。	开发人员审查和测试代码的持续活动。要求、迭代和任务应该分配时间来执行安全冒烟测试和单元测试。
静态应用程序安全测试（SAST）	一种通常由工具支持并集成到构建流水线中的方法，旨在在编译成最终二进制文件之前，分析应用的源代码找出可被利用的漏洞和脆弱性	在通过 IDE 进行开发期间，并在每次构建或部署时，进行持续的自动化活动。
动态应用程序安	自动化方法通常由工具支持并集成到构建流	每次部署都会持续自动启动测试，这

[20] Portman, D. G. (2020, September 22). Use Four Keys metrics like change failure rate to measure your DevOps performance. Google Cloud Blog. https://cloud.google.com/blog/products/devops-sre/using-the-four-keys-to-measure-your-devops-performance

全测试（DAST）	水线中，用来分析编译后的应用程序，识别可被利用的安全漏洞或脆弱性	通常是在非高峰时段的耗时扫描。可以作为非开发人员的互动“右移”，并在集成大部分应用程序组件后作为夜间扫描执行
交互式应用程序安全测试（IAST）	将 DAST 的元素和代码可见性结合起来，它作为测试运行时环境中的代理实现，以测试攻击操作并识别漏洞。	每次构建或部署以及应用程序的执行期间持续化的自动化运行。可以作为非开发人员的互动“右移”，并在集成大部分应用程序组件后作为夜间扫描执行
故障注入	一种混沌工程方法，通过对应用程序注入在生产中可能面临的故障和依赖性中断。故障注入是将故障引入应用程序设计，以验证其健壮性和错误处理能力。	通常需要所有应用程序相关方合作的计划活动（自动和手动）。需求、迭代和任务应该专门用于故障注入。
渗透测试	从内部或外部，使用白盒或黑盒的方法测试应用程序，以发现应用程序配置和设计上的脆弱性。通常测试由专业渗透测试人员在发布前实施	应用程序的可发布版本的计划活动。 需求、迭代和任务应专门用于渗透测试。

表 1：测试类型阶段和测试频率

表 1 展示了具有手动和自动活动的应用程序的理想测试参数。这些通常应在构建和部署阶段引入，并可以映射和对齐图 5 中的 CSA DevSecOps 交付流水线。

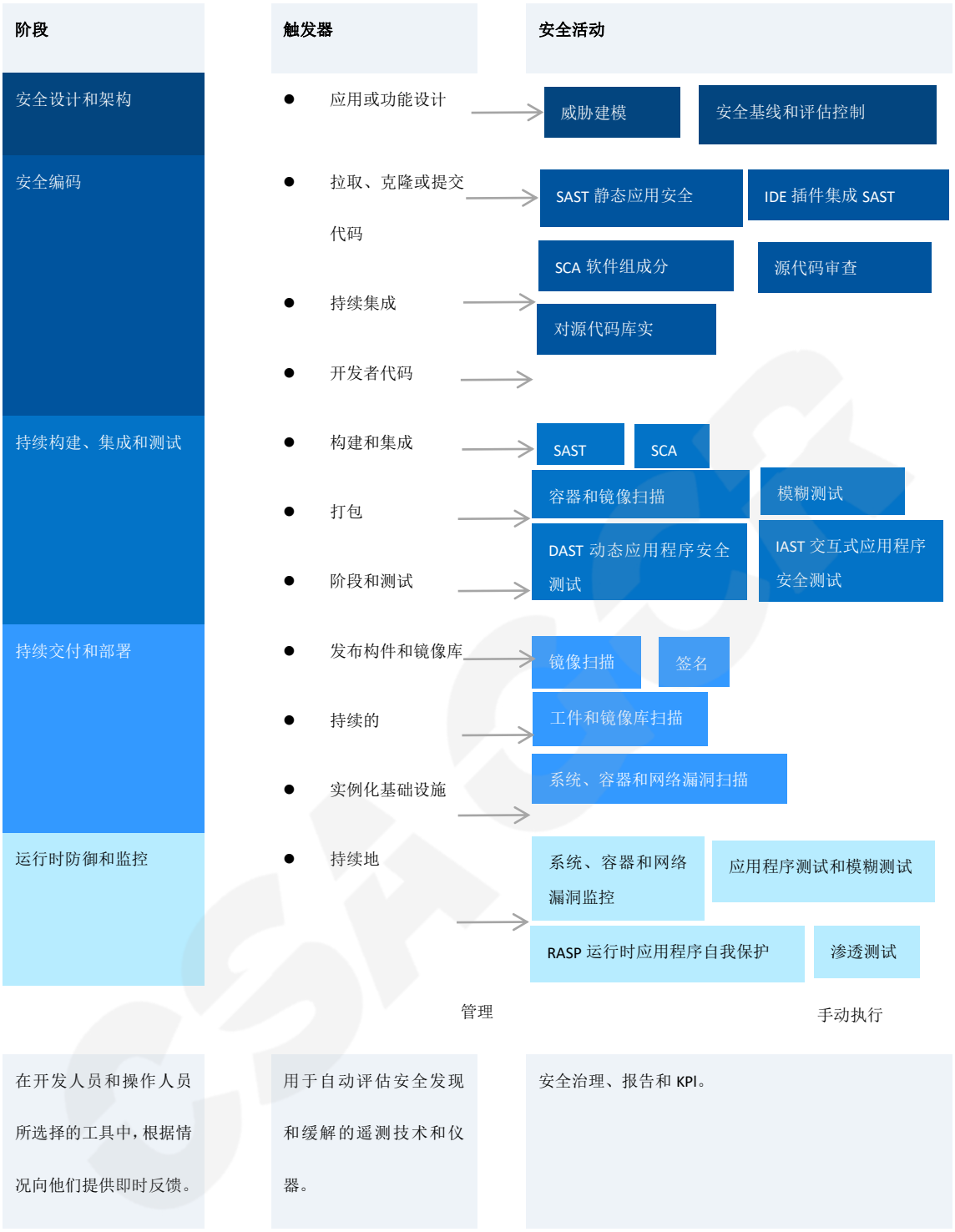


图 5.CSA DevSecOps 交付流水线[21]

4.3 追踪开源风险

组织越来越多地采用开源代码和库作为其环境和软件开发活动的一部分。虽然开源社区

[21] Cloud Security Alliance. (2020b, July 6). The Six Pillars of DevSecOps: Automation. https://cloudsecurityalliance.org/artifacts/devsecops-automation/

代表了巨大的价值和创新潜力，但也伴随着相关的风险和顾虑。

组织应审查连接到其代码库和环境的所有开源软件和组件，以确保它们不会引入不必要的风险和漏洞。开源工具和依赖关系的识别往往被忽视。依赖不明开源工具的应用程序将具有未知的供应链安全度量，如图 4-3 所示：

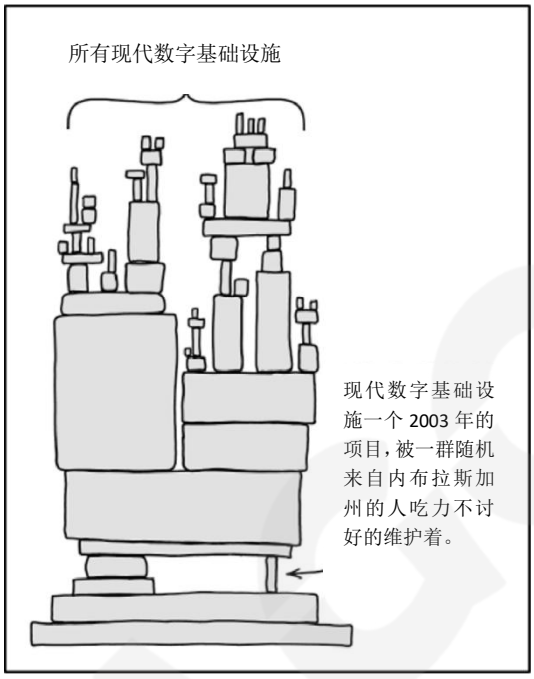


图 9.依赖风险示意图[22]

外部组件存在风险，应当被适当地审查。这可以通过以下流程和工具方法来实现：

- **流程：**组织应确保存在引入开源库的真正需求或业务案例。不必要的库可能会导致攻击面和暴露面增加。开源工具更新、发布和贡献率是衡量开源风险的指标。
- **工具：**组织机构应利用软件组成分析（SCA）工具扫描代码、项目和代码库，在技术方面寻找第三方库。SCA 可以帮助清点应用程序中使用的开源组件，并识别漏洞。

[22] Munroe, R. (n.d.). Dependency. Xkcd. Retrieved December 22, 2021, from https://xkcd.com/2347/

使用这种方法并不能消除风险，但考虑到开源软件和组件的广泛使用，它确实有助于限制攻击面。图 4-4 说明了第 3 阶段 SCA 可以适合 AWS 基础设施的阶段。图 10 还说明了在代码构建/安全编码阶段可以放置 SCA 的位置。SCA 是一种安全左移的有效方法，因为 SCA 是部署前构建阶段的安全控制。

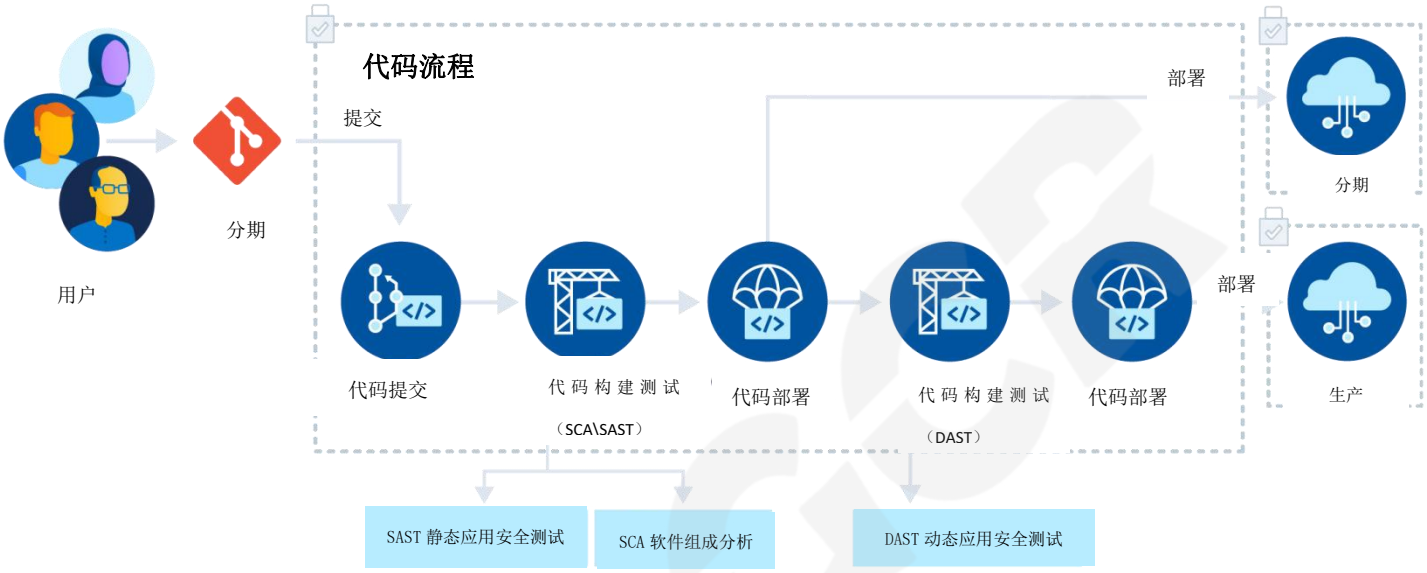


图 10.Cloud23 上的代码构建引入 SCA[23]

4.4 安全护栏

安全护栏（Guardrails）是集成工具，在最好的情况下，在软件开发过程中实现自动化，以确保与组织的目标和目的保持一致，包括合规性。安全护栏建立了一个可以持续监控部署的基线。这些基线可以表示为检测和预防策略的一组高级规则。

安全护栏可以作为合规性报告的一种方式（例如，在当前批准的镜像列表中运行操作系统的机器数量）或作为一组强制/自动修复的控制措施（例如，运行任何非批准的操作系统的设备自动关闭）。

专注于 DevOps 流水线内的内联安全护栏和反馈——通过紧密集成的工具和流程实现——可以将合规性方法从时间点转变为持续。必须考虑确保工具和技术的实施与商定的合规目

[23] Manepalli, S. (2021, January 21). Building end-to-end AWS DevSecOps CI/CD pipeline with open source SCA, SAST and DAST tools. Amazon Web Services. https://aws.amazon.com/blogs/devops/building-end-to-end-aws-devsecops-ci-cd-pipeline-with-open-source-sca-sast-and-dast-tools/

标保持一致。必须很容易地产生证据，以通过内部和外部治理实现对这些控制的外部验证。

“DevSecOps 的六大支柱：自动化“讨论了”安全自动化和程序化执行框架的实施以及安全控制的监控，以识别、保护、检测、响应和从网络威胁中恢复[24]，因为它涉及保护代码，应用程序和环境。这些自动化流程的输出应用于提供满足合规目标所需的证据。

在 DevSecOps 中，部署流水线可以使用甚至构建与合规性目标相一致的环境。这可能是监视和执行控制的组合，并且可能会因环境、数据分类和业务风险（即开发与生产）而有所不同。

通过在代码、模板和安全护栏中定义目标状态，安全目标应尽可能直接集成到部署流水线中。目标应该是一个易于审查的自动审计跟踪，以及相关的安全指标，供 DevOps 团队和合规团队持续审查以推动安全决策。下面的图 11 显示了如何将安全护栏应用于云环境的示例。在构建和配置账户和环境之前应用（检测和预防）云约束。

图 11.Cisco 的 AWS Guardrails 示例[25]

安全护栏可以根据组织的合规要求和风险偏好进行定制。安全护栏的成功标准依赖于以

[24] Cloud Security Alliance. (2020b, July 6). The Six Pillars of DevSecOps: Automation. https://cloudsecurityalliance.org/artifacts/devsecops-automation/

[25] Ramamoorthy, S. (2018, February 14). DevSecOps: Security at the Speed of Business. Cisco Blogs. https://blogs.cisco.com/security/devsecops-security-at-the-speed-of-business

下确定的三个基本原则，并由表 2 中的一组示例控制类别提供支持。对于一组较低级别的云安全控制，请参阅云控制矩阵（CSA CCM）v4[26]组织应该：

- 了解信息/反馈、漏洞报告或跟踪基础设施部署的合规性偏差（例如，使用经批准的 VM 镜像）到强制/阻止（阻止未经批准的 VM 镜像运行的安全护栏）的不同程度的保证。这有助于根据上下文（即开发或生产环境）和相关风险偏好在安全性和开发人员灵活性之间取得平衡。
- 定义所需的目标安全状态/关键绩效指标（KPI），例如补丁、服务水平协议（SLA）、漏洞评估和补救时间。所需的目标状态将取决于组织的安全和风险偏好、其云环境和应用程序环境以及监管要求。“支柱 6‘测量、监控、报告和行动’文件”将更详细地定义所需的目标安全状态。
- 为生产环境建立职责分离（SoD），其中至少需要有两个人对应用程序的关键功能进行更改，以防止欺诈和错误。对于执行概念验证（即开发和测试）的环境，可以删除 SoD 控制，以允许开发人员自由创建功能，减少可能抑制生产力的安全控制。这种开发者自由来自这样一个假设： 相关的环境不能访问生产数据。SoD 的实施可以从预生产和质量保证环境开始，以确保安全控制和应用程序功能按预期工作。应用程序所有者还需要确保在发生事件时可以使用 SoD 方法恢复应用程序的可用性。

控制类别	安全护栏控制
数据保护	● 信息分类和保护是自动化的。 ● 静态数据已加密。 ● 传输中的数据已加密。 ● 对数据进行分类以识别敏感信息（SPII）和个人身份信息（PII）信息。 ● IT 系统按处理的信息类型和业务关键性分类。 ● 云存储选项默认配置为“私有”。将存储配置为“公共”是一个例外。 ● 密钥和秘密应在密钥库中存储和管理。
治理	● 制定政策来管理云环境的安全性。

[26] Cloud Security Alliance. (2021b, June 7). Cloud Controls Matrix and CAIQ v4. https://cloudsecurityalliance.org/artifacts/cloud-controls-matrix-v4/

	● 适当地配置管理和库存。 ● 定义了适当的指标来跟踪合规性以及关键绩效指标（KPI）和关键风险指标（KRI）。 ● 进行控制测试审查。
访问控制	● 访问是根据最小特权原则授予的。 ● 实施了基于角色的控制访问（RBAC）。 ● 应用程序已指定负责访问控制的所有者。 ● 特权访问（即具有升级权限的访问，例如 root）应保存在安全数据库中，并且仅在例外情况下获得批准后才授予访问权限。 ● 集中管理对关键系统的访问（即安全运营或 DevOps 团队）。
物理环境安全	● 内部网络使用 VPN 或 VPC 来隔离和保护内部环境。 ● 使用零信任[27]安全模型。 ● Web 应用程序防火墙和反 DDOS 解决方案已到位。 ● 服务器、应用程序依赖项和容器经常打补丁并更新到最新版本。 ● 应验证第三方软件组件的使用。 ● 必须进行持续地漏洞扫描。 ● 环境分离到位（生产、暂存、开发、沙盒）。
应用程序变更	● 更改通过集中管理的 CI/CD 流水线进行。 ● 镜像和库集中管理和批准。 ● 新镜像经过加密签名。 ● 更改在部署前进行安全扫描。 ● 主要版本经过安全测试（请参阅“提高代码保证级别”一章）。 ● 更改经过代码审查、安全冒烟或单元测试。
弹性	● 环境具有内置的弹性，基于服务的价值和关键性（即故障转移、备份）。 ● 负载平衡到位以确保流量管理。 ● 执行定期备份以促进分配的服务级别协议内的可恢复性。 ● 高可用性环境和备份均已到位。 ● 业务连续性练习和故障注入（请参阅“提高代码保证级别”一章）包括生产环境。 ● 测试了备份的可恢复性。

[27] Cloud Security Alliance. (2020a, May 27). Software-Defined Perimeter (SDP) and Zero Trust. https://cloudsecurityalliance.org/artifacts/software-defined-perimeter-and-zero-trust/

监控	● 环境监测和基于事件的警报到位。 ● 警报规则是可配置的。 ● 关键事件和相关操作的日志记录到位。 ● 针对异常和新出现的威胁进行事件威胁检测。 ● 出于取证目的对端点进行远程调查。 ● 应根据保留政策保留日志，并应防篡改。

表 2.安全护栏类别和控制

像 AWS 等云服务提供商，可以通过在其环境中提供建议的护栏控制措施来提供帮助[28]。但是，在实施之前，建议执行识别与组织相关的控制的练习，如表 2 所示。

4.5 模式和模板

模式（部署和处理资源的标准化方法）和模板（已批准的供使用的资源构建）的使用可以协助构建安全的活动；因此，开发人员有权创建或消耗安全资源。

模式可以是创建用于支持护栏控制的文档。文档可以支持如何在云上创建资源的示例，如存储、虚拟私有云、计算服务和容器管理。模式可以用在开发人员自主创建和管理自己的资源。模式是将安全策略和合规要求映射到云资源的设计和配置的有效手段。它们可以作为开发人员和安全团队之间的共享责任项目，甚至可以属于安全的拥护者。

反模式使用（部署和处理资源的不良做法和方法）也是一个值得关注的点。然而，反模式有一个显著的缺点，即在识别、创建、维护和使用上的耗时，而且验证云资源或代码提取是否包含不良实践也是一个挑战。

模板是经过打包和认可的资源，如镜像、容器和 IaC 代码。模板可以在部署阶段生成，但在一个持续开发模型中，它是被设计为在构建期间中使用。开发人员创建的应用程序组件将利用用于编排托管基础设施（云）的模板，并知道该模板已被扫描和安全地配置。在创建模板（例如，docker 镜像、容器和 IaC 脚本）时，需要考虑的关键因素如下：

[28] Amazon Web Services. (n.d.). Guardrail reference - AWS Control Tower. Retrieved December 22, 2021, from https://docs.aws.amazon.com/controltower/latest/userguide/guardrails-reference.html

- 镜像（Images）
  - 识别可能用于每个组件的镜像需求和类型。这将有助于为每个用例生成单独的模板，并有助于加固操作，如有些组件可能需要 docker 写入权限。
  - 扫描资源中存在的漏洞并修复，包括常见漏洞和暴露（CVEs）亦或采用特定版本的软件如操作系统包和库。
  - 加固镜像，以满足所使用例的安全要求，亦或满足行业加固指南（例如，Red Hat、Palo Alto）.
  - 对于每个已扫描和加固镜像，应被打包和标记为模板存储在可访问的存储库中，供其他开发人员和团队使用。
  - 该模板应作为安全包的基线，并应持续或定期进行漏洞扫描和加固。
- 容器（Containers:）
  - 识别每种部署类型的容器用例—有些实例可能需要附加功能（例如卷/块存储）。
  - 使用业界加固指南（如，CIS、NAS 的 Kubernetes 安全加固指南）和开源工具（例如，Kubescape）扫描资源以找出针对不良设计和配置的弱点。
  - 对于每个已扫描和加固的容器，应被打包和标记为模板存储在可访问的存储库中，供其他开发人员和团队使用。该模板应做安全软件包的基线，并应持续或定期进行漏洞扫描和加固。
- **基础设施即代码（IaC）**
  - 识别基础设施即代码（IaC）用例并在所有 CSP 中部署资源。
  - 使用业界安全加固指南（例如，CIS）和开源工具（例如，Checkov、Terrascan）扫描资源以找出针对不良配置的弱点和漏洞。
  - 对于每个已扫描和加固的 IaC 脚本，应被打包和标记为模板存储在可访问的存储库中，供其他开发人员和团队使用。该模板应作为安全软件包的基线，并应持续或定期进行漏洞扫描和加固。

镜像、容器和 IaC 上模板的许可工具可以创建定制的策略，这可能有利于具有特定需求

的组织。对于拥有大量应用程序和产品团队的组织，模板可以帮助确保应用一致的安全控制级别。

5 总结

在 DevSecOps 之前，与风险相关的需求很难转化为安全活动；这些需求通常以安全门的方式被引入，开发和安全团队在不同的孤岛中工作。这造成了管理和沟通问题，导致安全需求在设计、构建和测试阶段传递不佳；在整个过程中，安全通常是事后才考虑的。如今，应用程序开发的部署速度和频率的不断提高，要求在不影响安全性和质量的情况下实现高效和自动化的解决方案。

DevSecOps 作为一种“现代方法”，用于保护应用安全和在整个安全软件开发生命周期（SSDL）中嵌入合规需求，允许安全“左转”。这种方法覆盖 SSDL 的五个应用阶段（设计和架构、编码、持续构建、集成和测试、持续交付和部署，以及运行时防御和监控[29]）以下原则和做法有助于弥补合规与发展之间的差距：

- 在实时审查的同时，持续进行评估。
- 引入价值流图来识别现有可以自动化来提高速度的实践做法。
- 识别合规性目标并转化为安全用户需求，供开发人员使用
- 将合规性作为代码/策略作为代码，编写云和基础架构构建中安全要求。
- 频繁使用各种安全测试方法（冒烟测试、SAST、DAST、错误注入、渗透测试）测试。
- 通过识别、审查、衡量风险和使用 SCA 工具来跟踪和控制开源代码。
- 定义和建立安全护栏用于监控部署情况和自动发现与期望基线的偏差。
- 利用模式和模板一致地扩大安全性。
- 了解监控 CI/CD 流水线中潜在恶意事件的方法.
- 使用 lint 工具、扫描器和插件来提高 IDE 中源代码和 IaC 的质量。

[29] Cloud Security Alliance. (2020b, July 6). The Six Pillars of DevSecOps: Automation. https://cloudsecurityalliance.org/artifacts/devsecops-automation/

解决合规和发展差距需要利益相关者对本文中建议的实践作出承诺。正确遵循 DevSecOps 的组织将意识到，在有效流程、文化和治理的支持下，工具和自动化将提高风险管理的质量和大规模应用安全控制的速度。

参考文献

Agile Alliance. (2021, July 26). *What does INVEST Stand For?* https://www.agilealliance.org/glossary/invest/

Amazon Web Services. (n.d.). *Guardrail reference - AWS Control Tower.* Retrieved December 22, 2021, from https://docs.aws.amazon.com/controltower/latest/userguide/guardrails-reference.html

American Institute of CPAs. (n.d.-a). *Segregation of Duties. AICPA.* Retrieved December 22, 2021, from https://www.aicpa.org/interestareas/informationtechnology/resources/value-strategy through-segregation-of-duties.html

American Institute of CPAs. (n.d.-b). *SOC for Service Organizations.* AICPA. Retrieved December 22, 2021, from https://www.aicpa.org/interestareas/frc/assuranceadvisoryservices/socforserviceorganizations.html

Cloud Security Alliance. (n.d.). *Security, Trust, Assurance and Risk (STAR).* Retrieved December 22, 2021, from https://cloudsecurityalliance.org/star/

Cloud Security Alliance. (2019, August 7). *Six Pillars of DevSecOps.* https://cloudsecurityalliance.org/artifacts/six-pillars-of-devsecops/

Cloud Security Alliance. (2020a, May 27). *Software-Defined Perimeter (SDP) and Zero Trust.* https://cloudsecurityalliance.org/artifacts/software-defined-perimeter-and-zero-trust/

Cloud Security Alliance. (2020b, July 6). *The Six Pillars of DevSecOps: Automation.* https://cloudsecurityalliance.org/artifacts/devsecops-automation/

Cloud Security Alliance. (2020c, July 13). *Hybrid Cloud and Its Associated Risks.* https://cloudsecurityalliance.org/artifacts/hybrid-clouds-and-its-associated-risks/

Cloud Security Alliance. (2021a, February 21). *The Six Pillars of DevSecOps: Collective Responsibility.* https://cloudsecurityalliance.org/artifacts/devsecops-collective-responsibility/

Cloud Security Alliance. (2021b, June 7). *Cloud Controls Matrix and CAIQ v4.* https://cloudsecurityalliance.org/artifacts/cloud-controls-matrix-v4/

CloudPassage. (2020, August 26). *Shared Responsibility Model Explained.* Cloud Security Alliance. https://cloudsecurityalliance.org/blog/2020/08/26/shared-responsibility-model-explained/

Google. (n.d.). *DevOps process: Visibility of work in the value stream.* Google Cloud. Retrieved December 22, 2021, from https://cloud.google.com/architecture/devops/devops-process-work visibility-in-value-stream33

International Organization for Standardization. (n.d.). *ISO/IEC 27001 — Information security management.* ISO. Retrieved December 22, 2021, from https://www.iso.org/isoiec-27001-information-security.html

Johnson, M. (2021, April 8). *Checkov 2.0: Deeper, broader, and faster IaC scanning.* Bridgecrew. https://bridgecrew.io/blog/checkov-2-0-release/

Manepalli, S. (2021, January 21). *Building end-to-end AWS DevSecOps CI/CD pipeline with open source SCA, SAST and DAST tools.* Amazon Web Services. https://aws.amazon.com/blogs/devops/building-end-to-end-aws-devsecops-ci-cd-pipeline-with-open-source-sca-sast-and-dast-tools/

Munroe, R. (n.d.). *Dependency*. Xkcd. Retrieved December 22, 2021, from https://xkcd.com/2347/

National Institute of Standards and Technology. (2021, October 26). Cybersecurity Framework. NIST. https://www.nist.gov/cyberframework

Portman, D. G. (2020, September 22). Use Four Keys metrics like change failure rate to measure your DevOps performance. Google Cloud Blog. https://cloud.google.com/blog/products/devops-sre/using-the-four-keys-to-measure-your-devops-performance

Ramamoorthy, S. (2018, February 14). DevSecOps: Security at the Speed of Business. Cisco Blogs. https://blogs.cisco.com/security/devsecops-security-at-the-speed-of-business

S. (2019, June 7). How INVEST helps team write effective user stories. . .. Medium. https://sabaimam.medium.com/investing-in-user-stories-c7cfb1fc5e85

SecurityScorecard. (2018, February 15). Limitations of Point-in-Time IT Security Risk Assessments. https://securityscorecard.com/blog/limitations-of-it-security-risk-assessment

SonarQube. (n.d.). SonarLint Integration. Retrieved December 22, 2021, from https://www.sonarqube.org/sonarlint/

Velimirovic, A. (2020, September 9). What Is Infrastructure as Code? Benefits, Best Practices, & Tools. PhoenixNAP Blog. https://phoenixnap.com/blog/infrastructure-as-code-best-practices-tools

词汇表

CSA STAR STAR 注册表记录了流行的云计算产品提供的安全和隐私控制。云客户可以使用这个可公开访问的注册表评估他们的安全供应商，以做出最佳的采购决策[30]。

时间点评估技术或流程集的成熟度和合规性的时限。 通常要持续 4 周或 4 个月，并在特定时间点审查当前状态。

条块化应用程序、平台、基础设施服务和资源的分离。

职责分离企业可持续风险管理和内部控制的基石。职责分离的原则基于关键流程的职责共担，将关键流程的基本功能分散给多个人或多个部门[31]。

责任共担当客户将应用程序、数据、容器和工作负载迁移到云上时，客户的安全团队要承担一定的安全责任。同时，提供商要承担部分责任，但不是全部。定义好客户和提供商的责任边界对降低漏洞引入公共、混合、多云环境带来的风险至关重要[32]。

SOC 2 报告组织对其系统的安全性、可用性、处理完整性、机密性和隐私的系统性控制措施[33]。

---

[30] Cloud Security Alliance. (n.d.). Security, Trust, Assurance and Risk (STAR). Retrieved December 22, 2021, from https://cloudsecurityalliance.org/star/

[31] American Institute of CPAs. (n.d.-a). Segregation of Duties. AICPA. Retrieved December 22, 2021, from https://www.aicpa.org/interestareas/informationtechnology/resources/value-strategy-through-segregation-of-duties.html

[32] CloudPassage. (2020, August 26). Shared Responsibility Model Explained. Cloud Security Alliance. https://cloudsecurityalliance.org/blog/2020/08/26/shared-responsibility-model-explained/

[33] American Institute of CPAs. (n.d.-b). SOC for Service Organizations. AICPA. Retrieved December 22, 2021, from https://www.aicpa.org/interestareas/frc/assuranceadvisoryservices/socforserviceorganizations.html

缩略语

AWS – 亚马逊 Web 服务

CAC – 合规即代码

CSA CCM – 云安全联盟 云控矩阵

CI/CD – 持续集成/持续交付

CSA STAR – 云安全联盟 云安全、信任、保障和风险

CSC – 云服务客户

CSP – 云服务提供商

DAST – 动态应用安全测试

DORA – DevOps 研究与评估

GDPR – 通用数据保护条例

IaaS – 基础架构即服务

IAC – 基础架构即代码

IAM – 身份管理与访问控制

IDE – 集成开发环境

INVEST – 独立、可协商、有价值、可估计、小、可测试

ISO – 国际标准化组织

KPI – 关键绩效指标

NIST – 美国国家标准与技术研究院

OPA – 开放策略代理

OS – 操作系统

OWASP – 开放 Web 应用安全项目

PaC – 策略即代码

PaaS – 平台即服务

PII – 个人身份信息

SaaS – 软件即服务

SAST – 静态应用安全测试

SCA – 软件组成分析

SLA – 服务水平协议

SOC – 系统和组织控制

SOC 2 – 系统和组织控制 2

SOD – 职责分离

SPII – 敏感个人身份信息

SSDL – 安全软件开发生命周期

STAR – 安全、信任、保障和风险

VPC – 虚拟私有云

VPN – 虚拟私有网络

VM – 虚拟机

CSA GCR cloud security alliance
GREATER CHINA REGION ®

Cloud Security Alliance Greater China Region

扫码获取更多报告

云安全联盟大中华区
官网：https://c-csa.cn 电话：0755-86548359
地址：上海市浦东新区东育路255弄5号3FB30（前滩国际经济组织集聚区）/
广东省深圳市南山区粤海街道北京航空航天大厦1号楼905