DevSecOps 的六大支柱

协作与集成

CSA GCR cloud security GREATER CHINA REGION alliance®

CSA cloud security alliance®

DevSecOps 工作组的永久官方网址

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

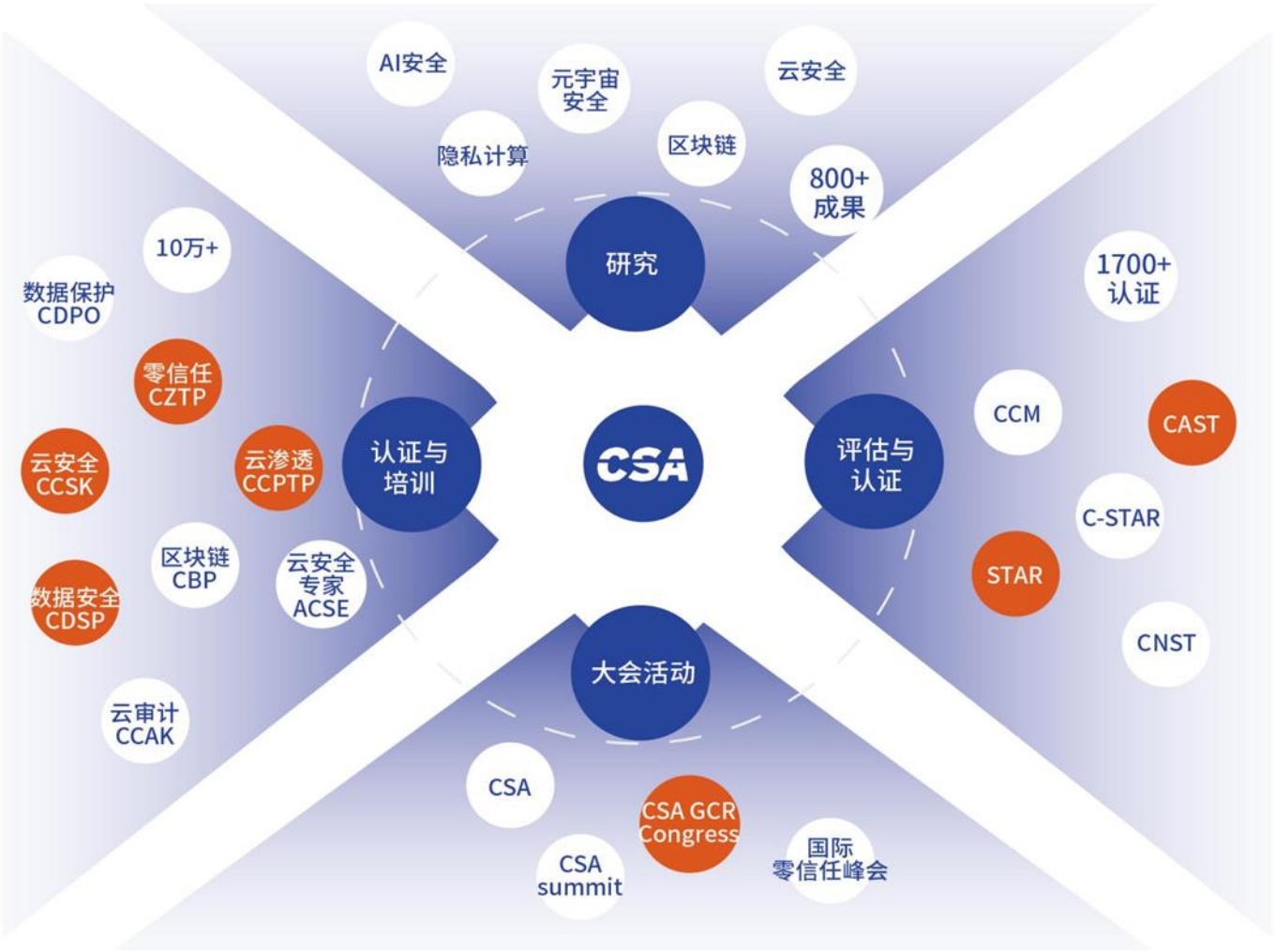
联盟简介

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 的六大支柱：协作与集成（The Six Pillars of DevSecOps: Collaboration and Integration）》由 Abdul Rahman Sattar 编写，并由 CSA 大中华区专家组织翻译并审校。（以下排名不分先后）：

中文版翻译专家组

组长：

李　岩

翻译组成员：

何伊圣　欧建军　王 彪　王贵宗　伏伟任　谢绍志

审校组成员：

苏泰泉　李　岩　江 楠　罗智杰　卜宋博

研究协调员：

闭俊林　易利杰

贡献单位：

北京天融信网络安全技术有限公司

英文版本编写专家

主要作者：

Abdul Rahman Sattar

贡献者：

Aristide Bouix

Amit Butail

Ivan De Los Santos

Darien Hirotsu

Nitin Kulkarni

Alexandria Leary

Michael Roza

审稿人：

Kapil Bareja

Ram Reddy

Udith Wickramasuriya

CSA 分析师：

Josh Buker

目　录

序言

《协作与集成》报告充分体现了组织中的每个部门都同样负责在软件开发周期的每个阶段集成安全性。DevSecOps 是一种文化取向、自动化方法和平台设计方法，将安全性作为整个 IT 生命周期的共同责任。

DevSecOps 是基于 DevOps 的安全敏捷化的一场变革，其中 DevOps 名著《加速：企业数字化转型的 24 项核心能力》中第 22 个能力要求即是协作能力 ，协作是支持和促进团队之间的合作，反映了传统上孤立的团队在开发，运营和信息安全方面的互动程度。其中第 3 个能力要求即是集成能力,集成能力是实现持续交付的第一步。这是一种开发实践。

本白皮书以在 DevOps 中引入安全性为起点，由浅入深地介绍基于 DevOps 的安全开发生命周期，需要在每个阶段充分地理解软件生命周期各阶段的安全都需要人员、文化、流程和技术的组合推进。安全的本质是一项团队运动，需要各种组织角色之间的全面协作，包括业务领导者、领域专家、安全人员、架构师、软件开发人员、渗透测试人员、SOC 分析师和产品/项目经理。

DevSecOps 也是 CSA 顶级云安全专家课程(CSA ACSE)的核心内容,DevSecOps 是践行共享安全责任的协作表现，DevSecOps 意味着从一开始就考虑应用程序和基础设施的安全性。，需要在每个阶段充分地理解软件生命周期各阶段的安全都需要人员、文化、流程和技术的组合推进。

李雨航 Yale Li

CSA 大中华区主席兼研究院院长

前言

云安全联盟坚定致力于提高软件安全成果。作者于 2019 年 8 月发表的论文《DevSecOps 的六大支柱》提供了一套高级方法和成功实施于快速构建软件并最大限度地减少与安全相关的漏洞的解决方案。这六大支柱是：

支柱 1：集体责任（2020 年 2 月 20 日发布）

支柱 2：协作与集成（2024 年 2 月 21 日发布）

支柱 3：务实的实现（2022 年 12 月 14 日发布）

支柱 4：建立合规与发展的桥梁（2022 年 2 月 8 日发布）

支柱 5：自动化（2020 年 7 月 6 日发布）

支柱 6：测量、监控、报告和行动（2024 年 5 月发布）

为支持六大支柱，云安全联盟和 SAFECode1 联合发布了一系列更详细的成功解决方案。本文是此系列出版物的第二篇。

引言

云安全联盟 DevSecOps 工作组 (WG) 在云安全联盟《通过反思性安全进行信息安全管理：安全、开发和运营集成的六大支柱》1 中发布了高级指南，介绍了一种称为“反思性安全”的新应用程序安全方法。对于任何对实现反思性安全或 DevSecOps 感兴趣的组织来说，这六个支柱被认为是关键关注领域。

其中支柱之一是“支柱 2：协作与集成”，可以概括为“安全只能通过合作而不是对抗来实现”。安全是一项团队运动，需要各种组织角色之间的全面协作，包括业务领导者、领域专家、安全人员、架构师、软件开发人员、渗透测试人员、SOC 分析师和产品/项目经理。需要关键利益相关者和各个组织角色之间的协作，以确保充分了解与业务部门相关的威胁态势，并确保有组织的 IT 活动（包括软件开发生命周期）的实践遵循适当的安全习惯。各个利益相关者还需要协作来确保组织支持持续的基于角色的安全培训。安全拥护者必须与组织中的其他团队合作，以确保安全实践得到充分记录并经常在整个组织内进行沟通。领导层和安全团队需要合作，以确保业务连续性风险纳入相关网络风险，并确保企业制定适当的网络事件响应策略。

1 https://cloudsecurityalliance.org/artifacts/information-security-management-through-reflexive-security/page7-9

目标

本文的主要重点是强调将 DevSecOps 集成到组织流程中以及促进成功实施 DevSecOps 所需的协作的重要性。本文首先列出了 DevSecOps 沟通的指导原则，然后深入研究在组织中实施持续的基于角色的安全培训计划的细节。接着，本文将介绍不同组织角色如何在端到端 DevSecOps 交付流水线中进行协作。接下来的一节介绍了不同组织角色之间所需的沟通和协作，以将新的采购集成到组织现有的 DevSecOps 流程中。本文最后有一节介绍了 DevSecOps 与零信任、AIOps 和 MLSecOps 等其他技术领域之间的融合，概述了如何利用 DevSecOps 实现零信任、MLSecOps 中与 DevSecOps 有一些相似之处的各种问题，以及 DevSecOps 如何利用 AIOps。

受众

本文档的目标受众包括参与风险、信息安全、信息技术和知识管理的管理和运营职能的人员。这包括 CISO、CIO、COO 以及参与以下职能领域的个人：安全工程、产品管理、解决方案和应用程序架构师、自动化、DevOps、质量保证、信息安全、应用程序安全、治理、风险管理和合规性、人力资源和培训。

1．成功的 DevSecOps 协作和集成的指导原则

DevSecOps 是一种将安全性集成到 DevOps 流程中的方法。这确保了软件从开发过程的一开始就是安全、可靠和高效的。“左移”（Shift Left）是一个经常用来描述这种集成的术语。

要成功实施 DevSecOps，建立协作、沟通和持续改进的文化非常重要。领导层、产品、项目、开发人员、安全专业人员和运营团队无缝协作，确保业务连续性、IT 安全性，并创建和部署安全软件。以下是为 DevSecOps 打造有效的跨团队协作和集成文化的一些技巧。

领导层主动和有效的沟通

为了有效，文化转变必须自上而下进行。领导层的大力支持将促进实施 DevSecOps 文化所需的协作。领导层最终负责制定业务连续性目标并将其传达给组织的其他部门。这反过来又会推动组织的资金和努力，以帮助实现领导层设定的这些目标 确保面对网络事件时业务连续性的规划至关重要，实施安全第一的架构需要多个团队之间的协作，以了解企业面临的各种内部和外部网络威胁，以及这些威胁对业务的影响。一旦充分理解了威胁模型，团队就需要制定策略，确定如何识别和保护组织资产、持续监控和检测网络威胁，以及在发现网络漏洞时响应和恢复业务资产。使用 DevSecOps 原则将安全性集成到整个 IT 生命周期是保护和检测功能的关键。

没有孤岛

DevSecOps 的目的是将安全性作为一项共同责任整合到整个 IT 生命周期中。这需要业务所有领域之间完全透明，愿意从开发过程的一开始就嵌入安全性，并坚信安全与质量一样，不是一个团队或个人的唯一职责，而是由组织的所有部门共同承担。领导层应努力确保有一个持续的安全反馈循环，并鼓励团队之间公开、频繁的沟通，努力将安全性完全集成到业务流程中。工程、安全和运营应携手合作，共同拥有 DevSecOps 流程的成果。获得最大投资回报的方法之一是最

大程度地实现安全自动化。工程团队和安全团队之间的合作是确保这些自动化正常运行并正确实施的好方法。

自动化

利用工具在软件开发生命周期（SDLC）2 中实现安全优先设计的自动化可以节省时间并实时提供安全反馈。自动化可用于各种与安全相关的活动，例如：资产识别、持续安全监控、代码分析、漏洞扫描、渗透测试、事件响应和恢复以及在简化网络入侵期间的通信。

以人为本的方法

由于 DevSecOps 文化只有在许多不同团队的充分参与下才能成功，因此必须关注人员方面。持续学习的文化是确保您的团队能够保持 DevSecOps 技术和流程前沿的最佳方式。

需要创造衡量成功的标准，需要庆祝取得的成就！这应该在人员、学习以及技术方面实施。团队的学习目标可以包括培训和认证，这些目标应包含在绩效计划和员工评审中。请记住，您需要为这些学习计划的实施提供时间和激励。还应该庆祝技术里程碑，以保持团队的积极性并确保 DevSecOps 计划步入正轨。技术成功衡量的一些示例包括交付时间、部署频率、可用性以及安全漏洞或攻击的频率。

组织背景的理解

各行各业的组织都是独一无二的，每个组织都有其独特的文化、结构和特点和运行的方法。因此，DevSecOps 方法的采用需要根据每个组织的具体情况进行精心定制。这不仅涉及采用合适的工具，更重要的是培养符合 DevSecOps 原则的文化。围绕 DevSecOps 转型的讨论，过度关注工具选择和实施是很常见的。然而，为了让组织获得 DevSecOps 的真正好处，重点应该放在其基本原则上。

明确的所有权和责任

对于成功的 DevSecOps 计划，需要有明确的利益相关者 RACI（责任分配矩阵）。对于每项任务，需要明确利益相关者对谁负责该任务，哪些团队负责完成任务，以及需要咨询哪些利益相关者并告知任务进度和完成情况。例如，软件工程师负责处置在其应用程序或产品的应用层发现的风险。基础设施工程师负责操作系统层面发现的风险的处置等。

就如何衡量进展达成一致

组织需要达成一致，确定如何衡量质量、性能、可用性、安全风险和管控效果。这将有助于使所有相关方达到相同的目标。

就将采用何种方法对工作优先级排序达成协议。

组织需要就如何评估和优先考虑风险达成一致。关于如何优先考虑风险的分歧将不利于组织的合作。

将失败视为机遇

组织必须接受失败是不可避免的事实。他们应该把这些挫折看作是增长和提高的宝贵机会，而不是消极看待。

采用这些原则将为成功的 DevSecOps 项目创造合适的环境，帮助组织的安全软件开发实践成熟，同时为他们的客户提供价值。

2 另请参阅：支柱 3：务实的实现 和 支柱 5：自动化

2. 基于角色的安全培训项目的原因和计划

安全是一项团队运动。每个组的成员都需要根据他们的角色进行培训。世界是一个复杂多样的地方，人们有各种各样的能力、技能和态度。在公司中，这种多样性通常被用来创建利用这些不同能力和技能的特定角色。

DevSecOps 包含了一个多维度的方法，适用于组织和其产品提供的多个层次。鉴于个体在组织中有不同的角色，因此必须不断提供与他们的特定职责和职责相一致的培训。

表 1：组织中有不同的角色

参与者	角色定义
业务关键人员	
C 级高管	• 了解 DevSecOps 操作、最关键绩效指标（响应和修复平均时间的减少 MTTR...[3]）和关键风险指标。 • 为基于角色的安全培训分配资源和预算。
投资者，创始人和其他业务所有者	• 了解 DevSecOps 操作，最关键绩效指标（响应和修复平均时间的减少 MTTR）和关键风险指标。
工程经理，产品所有者，产品经理：	• 咨询实施 DevSecOps 方法在他们的工作方式和产品生命周期。
安全利益干系人	
CISO / CSO 或 CIO	• 对整个组织的 DevSecOps 计划负责。 • 负责定义 DevSecOps 计划目标。 • 使 DevSecOps 计划与商业目标和目标保持一致。
安全副总裁	• 对他们相应部门的 DevSecOps 计划负责。

	• 负责定义 DevSecOps 实施功能要求。
安全团队领导	• 负责监控 DevSecOps 过程并升级潜在问题。 • 负责定义 DevSecOps 实施功能要求。
安全运营（渗透测试人员，IR 团队，威胁情报）	• 负责将 DevSecOps 输入纳入他们的运营。 • 受 DevSecOps 实施者咨询。
DevOps/平台利益相关者	
VP /工程总监：	• 对 DevSecOps 流程实施负责。 • 咨询定义 DevSecOps 实施功能要求。 • 咨询 DevSecOps 在各部门的实施。 • 负责衡量 DevSecOps 实施对部门生产力的影响。 • 咨询评估 DevSecOps 实施对部门生产力的影响。
平台架构师和所有者	• 对在他们相应的应用程序中实施 DevSecOps 流程负责。 • 咨询 DevSecOps 在他们相应的应用程序中的实施。 • 负责衡量 DevSecOps 实施对他们相应的应用程序的影响
DevOps 工程师和团队领导	• 负责在开发和发布（CI / CD）工作流程中实施和维护 DevSecOps 流程。
软件开发工程师和团队领导	• 负责遵循 DevSecOps 流程。 • 咨询 DevSecOps 流程实施和持续改进。

Scrum Masters 和项目经理	• 负责在开发流程中通知和推广 DevSecOps 流程。 • 咨询 DevSecOps 流程实施和持续改进。
质量保证利益相关者	
质量保证副总裁/总监	• 咨询定义 DevSecOps 实施功能要求。
质量保证经理	• 对 DevSecOps 在 QA 流程中的实施负责。 • 咨询以评估 DevSecOps 实施对应用程序性能的影响。
质量保证和团队领导	• 对 DevSecOps 在 QA 流程中的实施负责。 • 咨询 DevSecOps 流程实施和持续改进。

安全培训计划的实施

在作为路线图的一部分推出安全计划时，必须记住，很多人可能没有足够的时间专注于安全。因此，确定您的计划目标是至关重要的。例如，该计划可以让产品所有者在新产品功能的发现阶段内置威胁建模。

一旦这些目标到位，建立组织的基线培训就至关重要。这将使成员熟悉安全和平台团队引入的工具和流程。可以发送定期的电子邮件通信或内部新闻简讯，介绍影响公司的最新安全趋势，以保持组织的更新。

预测某些团队成员会寻求更深入的见解，并主动提问或提出关于安全主题的改进建议。接受并视每个查询都来自“安全客户”至关重要。接受 DevSecOps 意味着赋予开发人员安全地发布他们的工作的能力，而不是阻碍他们。

对于那些希望深入研究的安全爱好者或“冠军”，可以考虑组成一个专门的工作小组。使用公司的协作工具来促进关于安全相关主题的频繁讨论，并至少每季度安排定期的内部研讨会或聚会。

如何获得安全培训计划的认可和支持

从多元化的业务负责人那里获得支持可能会很有挑战性，尤其是在很少有员工有专门时间处理他们组织内部的安全问题的情况下。然而，这不应该翻译成独立的努力。强调对领导层的必要性，即定期解决安全问题并与其他技术债务一起解决，以防止重大的安全事件发生。应向领导层强调，网络攻击可能对组织产生的数据丢失，客户流失，业务停机等影响。为获得安全培训计划的领导批准，向领导展示安全培训计划在减少组织中的网络事件方面的影响，这是通过网络意识的劳动力，健全的 devsecops 计划，以及组织应对网络攻击的更好准备。

另一个挑战在于保持员工对安全培训计划的参与。一种经过验证的方法是定期举办安全会议，并每月留出 30 分钟与每个安全冠军进行 1 对 1 的互动。这些时刻提供了公开承认安全冠军贡献的机会，并在他们的角色的其他方面提供支持，例如让他们参与关键的安全决策，例如选择和实施新的工具，流程或政策。

如何衡量安全培训计划的成功

衡量培训计划成功的一个直接方法是跟踪完成安全培训的参与者数量，以及测量与新闻简讯的互动。此外，您可以进行现场和线下的安全测验，桌面演练和模拟培训课程，在这些课程中您可以评估他们的知识以及参与者在这种情况下会如何行动或思考。

在建立安全活动工作小组并培养 DevSecOps 协作文化后，您可以引入其他指标，例如：

- 安全冠军（教练）数量的增长
- 工作坊参与度
- 报告的安全问题数量

- 在定义的内部服务等级协议内解决的安全问题数量
- 安全事件频率的变化
- 补救平均时间
- 安全问题数量上升或下降
- 检测安全问题的平均时间
- 修复安全问题的平均时间

3．交付流水线中的协作和集成

下图列举了构成端到端软件开发生命周期（SDLC）的关键阶段。每个阶段都需要利益相关者之间保持沟通与协作，以帮助实现该阶段的目标。本节将逐一介绍每个具体阶段，并针对每个阶段确定参与该阶段的关键利益相关者，以及这些利益相关者之间为确保成功执行所需的必要沟通和协作。

虽然这种视觉效果给人一种从一个阶段到另一个阶段的线性流动的印象，但阶段之间存在双向反馈循环。

图 1：CSA DevSecOps 交付流水线

安全设计和架构阶段

这个阶段是产品团队和架构师之间的跨团队协作，其中包括系统和安全架构师、开发人员和项目团队。 架构师和开发人员合作，将产品团队开发的产品愿景和产品需求文档 (PRD) 作为输入，并生成理想的系统高级设计 (HLD)。 实现理想的系统高级设计的步骤还应包括系统的渗透测试、红队、蓝队和紫队练习的计划工作。 架构师和开发人员在此阶段合作评估各种设计选择，并提出理想的设计来满足各种功能和非功能要求。 安全架构师为系统开发威胁模型，以了解每种设计选择的威胁向量、攻击面、安全风险和暴露半径。 安全架构师还评估并考虑重用组织中其他团队成功使用的现有安全设计模式。

安全编码阶段

在此阶段，项目经理和开发人员合作，确保项目包含从安全设计和架构阶段实现理想架构所需的高级里程碑。 开发团队使用敏捷开发方法将这些里程碑分解为史诗、用户故事和任务。 每日站会和项目回顾确保开发步入正轨并向前推进，并且团队成员保持一致。 在代码开发过程中，开发人员使用安全编码实践、OWASP 等安全标准、组织编码标准和代码审查流程来协作和开发安全代码。 早期阶段的系统高级设计用于开发身份验证、授权、审计以及与构建系统、混沌测试、安全测试和监控工具集成的代码。

持续构建、集成和测试

在此阶段，开发人员、安全测试人员、质量保证测试人员、运营和架构团队协作开发测试自动化并将其集成到系统 CI 流水线中，以便持续测试系统的性能、质量、可用性、可用性和安全性 作为构建周期的一部分。 开发团队使用来自代码覆盖率分析、QA 测试、SAST、DAST、IAST、容器镜像扫描、混沌测试、可扩展性和压力测试以及渗透测试的反馈和测试结果来解决代码和设计中的差距。 由于这些不同的测试练习而在系统设计中发现的任何差距都需要开发人员和架构师在系统高级设计中协作进行设计调整和修改。 敏捷用于让产品和项目团队随时了解所需的任何其他设计变更和错误修复。

持续交付和部署

在此阶段，开发人员和运营人员协作为项目开发持续部署 (CD) 流水线，并将其集成到组织使用的持续部署工具中。 团队共同努力，为系统设置持续监控和监控仪表板，以跟踪关键绩效指标 (KPI) 和警报。 该团队还合作开发项目事件响应手册，以确保项目与事件响应平台集成，并制定第 2 天支持的待命时间表。

运行时防御和监控

在这个阶段，开发人员和运营团队协作，确保系统在第二天的生产中顺利运行。运营和开发团队始终掌握项目 KPI 仪表板和安全事件，并共同解决系统警报和安全事件。运营团队通常是事件发生时的第一道防线。如果无法解决问题，事件将上报给开发团队。 事件发生后，通常会有一个事件的事后分析过程，架构、开发和运营团队聚集在一起，向高层领导简要介绍事件的根因分析 (RCA)、事件的详细信息以及这一事件是否可以避免。 然后，此事件的事后处理是否需要修复系统中的设计和编码差距提供反馈。4.新收购项目与 DevSecOps 交付流水线的集成过程

当一家新公司被收购时，安全团队需要了解其软件开发流程。这一点很重要，有两个原因：一是衡量其流程的成熟度，二是学习如何将其最佳地集成到现有流程中。收购方可以采取以下步骤。

收购后 60 天内

1.编制一份所有源代码管理平台（Bitbucket、GitHub、GitLab、Subversion 等）的列表。包括这些系统的管理员的名称。

2.列出所有活跃的代码存储库 URL。整理每个代码存储库支持的每个应用程序和其他工件。包括像适用于每个应用程序和产品的产品经理和工程师负责人。

3.列出被收购组织使用的编程语言及其比例。这有助于识别控制（扫描、代码

审查等）方面的差距。

4.确定使用了哪些工具/流程（如有）：

- 隐密扫描
- 静态代码分析
- 动态分析
- 容器/图像扫描
- 作为代码扫描的基础设施
- 漏洞管理
- 威胁建模
- 配置风险/基线扫描
- 软件即服务风险管理
- 身份和访问管理治理
- 变更管理
- 故障管理
- 风险管理
- 凭证平台集成

组织中使用的安全工具类型将很好地了解其当前 DevSecOps 的成熟度。

5.列出所有云帐户（AWS、Azure、GCP 等）和内部部署环境资源相应的所有者，以便与收购方的组织共享。

6.安置必要的设备，将所有安全扫描结果和可用日志作为新的数据源添加，发送到收购方机构的安全运营中心（SOC）。

7.与被收购方的高管、产品经理和软件工程师讨论，从工作优先级的角度来看，如何处理安全问题。

是否在每次冲刺中都分配了时间、金钱和人员来满足安全和架构现代化要求？

根据上述信息，确定与收购方流程和优先级方面的差距。

注：现阶段，不应对被收购方现有的软件开发流程进行任何变更，除非该现有流程的风险超过收购方的风险偏好。

收购后 180 天内

1.使用 DevSecOps 成熟度模型来衡量被收购方的各方面。应使用与收购方相同的模型。流行的 DevSecOps 成熟度模型如下：

- OWASP DevSecOps 成熟度模型
- 云原生计算基础成熟度模型
- 软件保证成熟度模型

2.根据成熟度模型的结果来制定计划，纳入收购方的 DevSecOps 计划并促其达成。

至少每年一次

1.评估并继续完善整个组织的 DevSecOps 计划。

每个团队都处于不同的成熟度水平，并有独特的途径来改进其 DevSecOps 实践。改变现有的软件开发过程应该在每个团队的实际基础上进行探索。

2.评估和报告不同的 DevSecOps 成熟度水平（团队、产品、部门等）

4. DevSecOps 案例研究

本节快速概述了一些不同规模、处于业务生命周期不同阶段的组织的真实案列，这些组织成功地实施了 DevSecOps，作为其数字化转型之旅的一部分，以及支撑其 DevSecOps 实施的关键方面。

拥有传统组件的大型老牌公司

Capital One 开始了 DevSecOps 转型之旅，以增强安全性、简化流程并加快软件交付。以下是他们如何实现 DevSecOps 的概述。

文化转型：Capital One 专注于在开发、运营和安全团队之间建立协作和分担责任的文化。他们培养了开放的沟通渠道，鼓励跨职能合作，以打破孤岛，促进责任共担。

自动化安全控制：Capital One 在整个开发流水线中实现了自动化安全控制。他们将静态应用程序安全测试（SAST）和动态应用程序安全检测（DAST）等安全扫描工具集成到其持续集成和部署过程中。这有助于在早期阶段识别和解决安全漏洞。

持续合规：Capital One 通过自动化合规检查并将其纳入其 CI/CD 流水线，强调持续合规。他们利用配置管理工具和政策执行机制来确保遵守行业法规和内部安全标准。

威胁建模与风险评估：Capital One 进行了全面的威胁建模和风险评估，以确定其遗留组件中的潜在安全风险。他们分析了攻击表面，识别了漏洞，并实施了适当的安全措施，以有效降低风险。

传统组件现代化：Capital One 通过将其分解为更小、更易于管理的单元，逐步实现其传统组件的现代化。他们采用了微服务架构和容器化，以提高可扩展性、可维护性和安全性。这使他们能够有效地将 DevSecOps 实践应用于现代化的组件。

安全自动化和编排：Capital One 利用安全自动化和编制工具来简化安全流

程和响应。他们实施了安全事件和事件管理（SIEM）系统、安全编排与自动化响应（SOAR）平台以及高级威胁检测机制，以增强其安全操作

参考文献：

•Capital One 成功的 DevOps 之旅秘密|开始、快速失败、重复

•DevSecOps-DevX

•云与运营团队如何将云应用转化为成功

•AWS re:Invent 2020：Capital One 的云“全能”之旅

风险偏好较高，规模快速发展的企业

大型公司成功实施 DevSecOps 的一个例子是 Netflix。他们采用 DevSecOps 实践，以确保其应用程序和基础设施的安全，同时保持高度灵活性和快速部署周期。以下是 Netflix 如何实现 DevSecOps 的概述。

拥抱 DevOps 文化：Netflix 培养了开发、运营和安全团队之间的协作和共享责任的文化。他们鼓励在整个软件开发生命周期中跨团队协同工作。

自动化安全流程：Netflix 自动化安全流程，将安全检查和控制集成到其持续集成和持续部署（CI/CD）流水线中。他们实施了一系列自动化安全工具和扫描仪，以便在开发过程的早期识别漏洞和配置问题。

持续安全测试：Netflix 在整个开发生命周期中执行持续安全测试，包括静态应用程序安全测试（SAST）、动态应用程序安全检测（DAST）和软件组成分析（SCA）。这使他们能够快速检测和解决安全问题。

基础设施即代码（IaC）：Netflix 使用基础设施即代码原则来管理其云基础设施。他们使用了 AWS CloudFormation 和 Netflix 的开源工具 Spinnaker 等工具来安全地自动化基础设施资源的供应和配置。

安全监控和事件响应：Netflix 部署了强大的安全监控系统，用于实时检测和响应安全事件。他们结合使用安全信息和事件管理（SIEM）工具、日志分析和威胁情报，及时识别和缓解安全事件。

协作和知识共享：Netflix 强调团队之间的协作和知识分享。他们定期举办安全培训课程，组织黑客马拉松，并鼓励公开讨论，以提高整个组织的安全意识和做法。

参考文献：

•DevSecOps-DevX

•Netflix 的全周期开发人员——运营您构建的内容

•AWS re:Invent 2015|（SEC310）让开发者和审计人员在云中保持快

成长为规模化企业的私人高成长型创业公司[5]

一个初创公司成长为可扩展 DevSecOps 实现的例子是特殊商品市场 Catawiki。他们采用 DevSecOps 实践，以确保其市场和微服务的安全，而不妨碍其部署状态和平台可用性。以下是 Catawiki 如何实现 DevSecOps 的概述。

拥抱开发人员和安全性之间的合作：Catawiki 的 DevSecOps 第一个里程碑是打破了“你负责安全，我负责编码”的心态。作为这一过程的一部分，安全部门的作用从质量的把关者演变为在整个软件开发生命周期中提供持续反馈和输入的推动者。这种新方法促进了安全、开发和基础设施团队之间的信任和更密切的合作。

标准化组件：考虑到开发团队的成熟度和技能水平各不相同，Catawiki 认识到创建一个所有人都可以依赖的共同基础的重要性，并提出改进建议。他们建立了标准框架、模板和基础设施即代码（IaC）工作流程，从而可以轻松实现安全性。

内置安全：Catawiki 自动化安全流程，将安全监控集成到其持续集成和部署（CI/CD）流水线中。他们明智地选择了与现有框架和技术兼容的自动化安全工具，保持与当前的工作方法一致，防止技术分散。

实施持续评估和单一窗口可见性：为了在整个软件开发生命周期（SDLC）中最大限度地采用安全技术，Catawiki 优先考虑向开发人员可以提供安全工具和报告。他们还开发了简单的流程，以简化工具的采用，并从开发人员的角度实现价值最大化。

实施漏洞奖励计划并鼓励负责任的披露：Catawiki 认识到安全性只有从外部角度才能进行真正的验证，因此定期邀请独立的安全研究人员检查他们的平台。安全和开发团队根据通用漏洞评分系统（CVSS）和内部服务水平协议（SLA），根据漏洞和补丁管理政策，共同确认解决漏洞的优先级。

游戏化漏洞管理：除了强制执行内部 SLA 外，Catawiki 还通过游戏化服务安全评分来激励非安全员工修复漏洞。他们为解决漏洞而奖励团队积分和奖励。还

通过额外的培训和研讨会加深对安全的理解。

5 Source: https://medium.com/catawiki-engineering/catawikis-devsecops-journey-c826fe7a9030

初创银行拥抱 DevSecOps

尽管许多银行仍在使用过时的传统技术，但受益于云原生系统具有更安全和性能的优势，银行业正在发生变化。在这种演变中，初创公司 10x Future Technologies 正逐渐崭露头角。他们率先开发了专为大型银行设计的云原生银行平台，有效解决了传统系统带来的成本和安全挑战。

以下是 10x Future Technologies 如何将 DevSecOps 融入其整体战略的概述。

通过使用工具集，促进安全团队与开发团队之间的合作： 在 10x，通过让每个人都能访问，用于跟踪工具发现漏洞的实时仪表板，促进安全团队与开发团队之间的合作。这些工具集可以对存在的漏洞发出警报，确保对所有团队成员的透明度和可访问性。透明度是一种重要机制，可确保每个团队都了解自己作为团队成员在交付安全的软件中扮演的角色。

安全责任共担： 在开发云原生银行平台的过程中，该组织采用的沟通策略非常重要，以加强对安全的实践。通过提高透明度、促进高效沟通、推动日常任务中的团队合作以及积极实施安全措施，10x 将安全整合为所有团队成员的共同责任。

实施漏洞安全持续审查流程：合作不再仅仅局限于现有漏洞。10x 通过召开由整个组织不同团队的代表参加的每日站会，确保在持续解决安全问题时考虑到广泛的观点。所有团队在每日站会上对漏洞进行审查，是确保组织做好准备应对不断变化的威胁环境的关键策略。

在组织文化中植入安全观念：沟通大大提高了团队的效率，并且团队之间开放的沟通渠道也有助于将安全融入到组织文化。这种方法进一步强化了安全软件的集体责任观念，确保将其融入组织的内在精神。通过这种宝贵的方法，在 10x 公司培养了内部不同角色之间更多的共鸣和理解。

通过对人员、流程和技术三大支柱的专注，10x 能够围绕软件交付建立起的安全文化实施独特实践。 10x 展示了这三大支柱如何以合作的方式，共同确保

交付高效、可靠和安全的软件。

参考

- Integrating security into development
- DevSecOps Examples | Successes and Lessons Learned | Snyk

5. DevSecOps 和其他技术实践融合的实践

本章将简要概述本文中阐述的一些原则和与其他技术实践的交互流程，以及 DevSecOps 与这些技术实践之间如何协作以实现更好的安全成果。本节将介绍的技术实践包括零信任、AIOps 和 MLSecOps 以及数据网格。

DevSecOps 和零信任

DevSecOps 概述

DevSecOps 是一种将安全实践融入 DevOps 流水线的方法，强调安全是开发、运营和安全团队的共同责任。其主要原则包括自动化、协同工作、持续集成、持续交付（CI/CD）和反馈循环。

零信任概述

零信任是一种安全战略。它假定无论是在组织网络内部还是外部的任何实体，都不应被默认信任。无论其位于何处，每个试图访问资源的用户、设备和应用程序的身份和可信度都需要验证。零信任的关键原则包括身份识别与访问管理、微隔离和持续监控。

DevSecOps 与零信任的交集。

DevSecOps 和零信任的交集成为整个软件开发生命周期的安全基础要素。

安全设计和架构

在安全设计和架构阶段，零信任团队和 DevSecOps 团队之间的合作至关重要。零信任提倡 "绝不信任，始终验证"的原则。它鼓励设计不信任任何实体（包括组织内的实体）的网络和应用架构。这意味着从一开始就进行微隔离、严格的访问控制和用户身份验证。零信任团队可以指导如何实施有效的微隔离控制，而 DevSecOps 可以确保将这些控制集成到架构中。这种协同工作可确保设计既安全又符合零信任的 "绝不信任，始终验证 "理念。

以微隔离为例：

零信任架构师：指导 DevSecOps 工程师为新的云应用程序设计微隔离控制。这种指导可

能包括定义微隔离、确定需要微隔离的应用程序和服务，以及推荐微隔离工具和技术。

DevSecOps 工程师：与开发团队合作，将微隔离控制集成到应用程序架构中。这可能涉及配置基于云的安全工具、开发自定义微隔离解决方案以及测试和验证微隔离控制措施。

安全编码

在安全编码阶段，零信任团队和 DevSecOps 团队之间的合作至关重要。零信任强调需要验证和确认用户和应用程序的身份，这与 DevSecOps 对安全编码实践的关注是一致的。DevSecOps 团队应与零信任团队合作，定义并执行基于身份的安全策略，确保只有经过身份验证并获得授权的实体才能访问代码。这种协作方法可确保将安全编码实践和身份验证无缝的集成到开发流程中。

以用户身份和应用程序为例

零信任架构师： 指导 DevSecOps 工程师定义和执行基于身份的代码安全策略。该指导可能包括定义用户访问代码所需的角色和权限，确定需要访问代码的应用程序和服务，以及推荐的身份验证与访问控制技术。

DevSecOps 工程师：与开发团队合作，实施基于身份的代码安全策略。这可能涉及配置源代码控制系统以执行基于身份的访问控制，与身份管理系统集成以验证用户和应用程序的身份，以及使用代码分析工具识别和修复与身份验证与访问控制相关的安全漏洞。

持续构建，集成及测试

在持续构建、集成和测试阶段，零信任团队和 DevSecOps 团队之间的协作对于持续的身份验证与访问控制至关重要。零信任对持续验证与 DevSecOps 对使用自动化安全测试工具的重视程度是一致的。当 DevSecOps 则将持续验证用户和应用程序身份集成到 CI/CD 流水线中时，零信任团队可以指导如何验证这些原则。协作可确保针对每个请求的访问控制重新评估，并使自动化安全测试工具与持续验证的零信任模型保持一致。

零信任架构师： 指导 DevSecOps 工程师定义和执行基于身份的代码安全策略。这种指导可能包括定义用户访问代码所需的角色和权限，确定需要能够访问代码的应用程序和服务，以及推荐身份验证和访问控制技术。

DevSecOps 工程师： 与开发团队合作，实施基于身份的代码安全策略。这可能涉及配置源代码控制系统以执行基于身份的访问控制，与身份管理系统集成以验证用户和应用程序的身份，以及使用代码分析工具识别和修复与身份和访问控制相关的安全漏洞。

持续交付和部署

在持续交付和部署阶段，零信任团队与 DevSecOps 团队之间的协作可确保对部署资源的访问受到严格控制，并且只允许访问必要的资源。零信任对访问控制和身份验证的重视与 DevSecOps 对安全检查和自动响应的使用是一致的。零信任团队可以指导访问控制策略的实施，而 DevSecOps 则将其集成到部署流水线中。协作可确保对每个访问请求的可信度进行验证，并对可疑行为触发自动响应。

- 零信任架构师：指导 DevSecOps 工程师在 CI/CD 流水线中实施持续身份验证和访问控制。这种指导可能包括验证用户和应用程序身份的推荐方法，确定可用于持续身份验证与访问控制的实施工具和技术，以及将持续身份验证与访问控制集成到 CI/CD 流水线中的指导：与开发团队合作，在 CI/CD 流水线中实施持续身份验证与访问控制。这可能涉及配置 CI/CD 工具用于执行身份验证，与身份管理系统集成以验证用户和应用程序的身份，以及使用自动安全测试工具识别和修复与身份和访问控制相关的安全漏洞。

运行时防御和监视

零信任团队和 DevSecOps 团队之间的合作对于运行防御和监控阶段的持续监控和行为分析至关重要。零信任原则与 DevSecOps 对持续监控和实时反馈的关注是一致的。零信任团队可以为实施网络分段、微隔离和行为监控提供见解，而 DevSecOps 则确保这些措施得到有效维护。这种协作可确保任何可疑行为都会触发调查和自动响应，从而与持续监控原则和零信任的主动防御保持一致。

零信任架构师:指导 DevSecOps 工程师为部署的应用程序实现持续监控和行为分析。该指导可能包括用于持续监控和行为分析所推荐的工具和技术，提供关于配置持续监控和行为分析工具用于检测可疑活动的指导，并帮助定义响应可疑活动的策略活动。

DevSecOps 工程师：与运维团队合作，对部署的应用程序实施持续监控和行为分析。这可能涉及配置监控工具以收集有关应用程序活动的数据，部署安全工具以检测所收集数据中的可疑活动，以及与安全编排自动化与响应 (SOAR) 工具集成，以自动响应可疑活动。

DevSecOps 和零信任总结

当考虑到软件开发生命周期的各个阶段时，DevSecOps 和零信任 的交叉点尤其强大。DevSecOps 将安全融入到软件开发的每个阶段，而零信任强调了即使在日益动态和分散的环境中也需要不断验证和核实访问。它们共同创建了一个整体而强大的安全框架，从设计到部署和运行时防御，为应用程序和基础架构提供保护。

MLSecOps and AIOps

MLOps 是成熟 DevOps 模式中的一个演进分支，对于充分发挥机器学习模型的潜力非常重要。MLOps 在优化的环境中协调运行过程，包括良好的基础设施配置、充分的模型开发、自动部署和持续的性能监控，从而确保机器学习模型的效率和有效性。与此同时，MLSecOps 还将安全措施和隐私因素纳入工作流程，确保数据安全、已部署模型的保护以及底层基础设施免受各种威胁。

AIOps 一词来源于 Gartner。AIOps 是人工智能功能的应用，旨在提高运营工作流程的效率。AIOps 利用大数据和机器学习来：

1.收集由基础设施、应用程序和服务生成的大量数据

2.对数据应用各种转换以提取特征

3.训练机器学习模型，建立行为基线

4.应用机器学习模型生成信号

5.使用机器学习、符号 AI、基于 AI 的规划和本体推理来关联信号。

6.应用机器学习模型来关联这些信号和根本原因分析(RCA)。

MLSecOps 和 MLOps 的整合意味着向 构建熟练且安全可靠的机器学习系统迈出了一大步。虽然 DevSecOps 的原则已被证明可以适应和转移到 AIOps 和 MLSecOps 等新兴领域，但在其实施过程中还存在一些细微差别，需要做进一步的探讨：

在本文中，我们将考虑如何将 DevSecOps 与 AIOps 相结合，以提高 IT 运营的效率、安全性和可靠性，并在稍后重点介绍 MLSecOps 如何实施控制和保障措施。

融合 DevSecOps 和 AIOps

潜在威胁检测和预防： AIOps 擅长筛选大量数据流和安全事件。这可确保迅速发现异常模式和潜在威胁。通过应用机器学习算法，AIOps 可以准确定位异

常行为、安全漏洞并预见风险，从而在潜在安全事件破坏生产环境之前采取积极措施加以缓解。

智能安全监控： AIOps 的功能扩展到监控和分析与安全相关的事件、日志和网络流量。人工智能驱动的异常检测和入侵检测大大缩短了安全团队从检测到采取必要行动之间的滞后时间。

风险缓解预测分析： 通过分析以往的安全事件和威胁情报，AIOps 可以洞察未来的潜在风险。这为 DevSecOps 团队提供了必要信息，帮助他们确定任务的优先级、有效分配资源并主动修复漏洞，从而降低安全漏洞发生的概率。

安全流程自动化： AIOps 将其自动化功能扩展到 DevSecOps 流水线中的安全流程。漏洞扫描、安全测试、合规性检查和配置管理等任务都可以通过 AIOps 实现自动化。

强化事件响应和补救： 如果不幸发生安全事件，AIOps 可促进事件响应和补救。AI 可生成安全事件摘要，从而简化安全分析师的切入时间，实时分析安全事件、事件相关性，并生成可操作的见解，为安全团队的响应工作提供指导。

辨别 MLSecOps 的差异因素

虽然基础是一致的，但机器学习的独特性质带来了特定的挑战，需要量身定制的安全措施。以下是 MLSecOps 与 DevSecOps 不同的五个关键领域。

模型证明（Model Provenance）： 模型证明（Model Provenance）提供全面的模型历史记录，涵盖开发、部署和使用过程。这些数据通过跟踪对模型的修改，包括贡献者、方法、时间和理由，有助于找出潜在的漏洞。

在人工智能法规不断发展的背景下，模型证明（Model Provenance）在评估过程中展示合规性方面的作用日益突出。通过揭示机器学习模型的开发、部署和使用过程，模型证明带来了透明度和责任感。这促进了负责任的数据使用，并在不断升级的网络威胁中保持模型的可信度。

治理、风险与合规： 机器学习材料清单（MLBoM）作为增强风险、治理和合

规性的一种方法，受到了广泛重视。该框架可应对 MLSecOps 中的挑战，包括 GRC [备注 1]和供应链漏洞，并提供机器学习模型开发材料的详尽清单，如算法、数据集和框架。它有助于发现供应链中可能危及模型完整性的潜在漏洞，如恶意代码或被篡改的组件。

MLBoM 可确保模型符合法律、法规和道德标准，保护数据隐私。强大的 "材料清单 "有助于追踪数据、模型和算法的来源，确保它们是可审计和可解释的。采用全面的 MLBoM 可以提高机器学习系统的安全性、透明度和问责制。

可信人工智能：偏见、公平和可解释性： 可信人工智能是指为实现公平、无偏见和透明而设计的人工智能系统。其目的是不考虑个人特征，实现公平决策。要实现这一目标，需要采用和使用一个定义明确的框架，如德勤可信人工智能（Deloitte TrustworthyAI）。该框架有助于管理与人工智能伦理和治理相关的常见风险和挑战，包括

- 公平公正的使用检查： 为确保人工智能的公平公正，公司必须积极识别并减少算法和数据中的偏见，从而解决编码过程中引入的偏见。这包括定义公平标准和实施控制措施，以防止出现意想不到的结果。

- 实现透明和可解释的人工智能:为了建立值得信赖的人工智能，确保透明度和可解释性至关重要。每个参与者都应该了解数据使用和人工智能决策。组织必须准备好使算法、训练数据、属性和相关性易于检查。

- 责任与问责： 创建值得信赖的人工智能需要制定明确的制度，确定对系统结果负责的责任方。

- 增加适当的安全控制：要确保人工智能值得信赖，就必须防范各种风险，包括可能造成伤害的网络安全威胁。公司必须全面评估和降低风险，同时以透明的方式将风险告知用户。

- 监控可靠性： 为使人工智能得到广泛应用，其可靠性应与传统系统和流程相匹配。公司必须验证人工智能算法对不同数据集产生的预期结果，并制定措施来解决可能出现的问题或差异。

- 保护隐私： 要建立对人工智能的信任，就必须按照 GDPR 和其他框架的规定，通过遵守数据法规和限制数据使用来维护隐私。各组织必须尊重消费者的隐私，避免使用未经授权数据，并为数据共享提供"加入/退出 "选项。

对抗式机器学习： 该领域涉及保护机器学习模型免受恶意攻击。恶意攻击可能涉及篡改输入数据以误导预测，或改变模型本身以降低准确性。对抗式机器学习旨在创建针对这些攻击的防御措施，增强模型的稳健性和安全性。为了实现这些目标，研究人员通常采用威胁建模和威胁影响分析、多分类器系统、实时攻击检测、合成训练数据的生成模型以及训练中的对抗性示例等保障措施。

6. 参考

云安全联盟，DevSecOps 的六大支柱:支柱 3，务实实施，2022 年 12 月，链接如下@

https://cloudsecurityalliance.org/artifacts/six-pillars-devsecops-pragmatic-implementation

云安全联盟，DevSecOps 的六大支柱:支柱 4 -桥接合规性和开发，2022 年 2 月，链接如下@

https://cloudsecurityalliance.org/artifacts/devsecops-pillar-4-bridging-compliance-and- development/

云安全联盟,DevSecOps 的六大支柱:支柱 5 -自动化,2020 年 7 月，

https://cloudsecurityalliance.org/artifacts/devsecops-automation/

云安全联盟，DevSecOps 的六大支柱:支柱 1 -责任共担，2020 年 2 月，

https://cloudsecurityalliance.org/artifacts/devsecops-collective-responsibility/

云安全联盟，《通过反身安全进行信息安全管理:DevSecOps 的六大支柱》，2019 年 8 月，

https://cloudsecurityalliance.org/artifacts/information-security-management-through-reflexive- security/

云安全联盟，DevSecOps 的六大支柱，通过集成安全、开发和运营实现自反式安全，2019 年 8 月，https://cloudsecurityalliance.org/artifacts/six-pillars-of-devsecops/

整合零信任和 DevSecOps，软件工程学院|卡内基梅隆大学，2021 年 7 月，

https://apps.dtic.mil/sti/pdfs/AD1145432.pdf

NSTAC 向总统提交的关于零信任和可信身份管理的报告，2023 年 2 月，https://www.cisa.gov/sites/default/files/2023-04/NSTAC Strategy for Increasing Trust Report %282-21-23%29 508 0.pdf

RSA 大会，通过专用 DevSecOps 流水线实现零信任，2023 年 4 月，https://www.youtube.com/watch?v=4DREGC-Z_FO

NIST 特别出版物 NIST SP 800-204D ipd 在 DevSecOps CI/CD 管线中集成软件供应链安全的策略，2023 年 8 月 30 日草案，意见征求截至于 2023 年 10 月

https://csrc.nist.gov/pubs/sp/800/204/d/ipd

MLSecOps 的定义，德勤新闻稿

Deloitte Introduces Trustworthy AI Framework - Press release | Deloitte US

利用 AIOps 扩展 DevsecOps：人工智能在提高效率和安全性方面的力量

Scaling DevSecOps with AIOps: The Power of Artificial Intelligence in Driving Efficiency and Security |

备注 1：GRC（即 *治理、风险与合规*）是一种用于管理治理、风险管理以及行业和政府法规合规性的组织策略。 GRC 还指用于实施和管理企业 GRC 计划的整套集成软件功能。 GRC 的整套实践和流程为使 IT 与业务目标保持一致提供了一种结构化的方法。 GRC 可帮助公司有效管理 IT 和安全风险、降低成本并满足合规性要求。 通过利用综合视图展现组织的风险管理状况，它

还有助于改进决策和绩效。

https://www.ibm.com/cn-zh/topics/grc

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