BLACKDUCK®

DevSecOps策略指南：

平衡速度与安全

随着现代软件彻底改变了业务运营和市场竞争，满足严格的上市时间要求的压力变得空前巨大。与此同时，对高度安全、可靠软件的需求也因快速生产的需要而增加。但是，快速的开发和发布周期以及日益复杂的网络攻击带来了一个核心挑战：如何在不牺牲安全的前提下，大规模开发和交付高质量的软件。

解决之道便是将应用安全测试 (AST) 纳入软件开发生命周期 (SDLC) 和DevOps工作流的各个阶段。这听起来简单，但实际操作起来仍有一些挑战，包括管理太多或重复的信息，避免可能会拖慢管道的不必要测试，客观地评估风险，以及各团队在安全意识和能力方面的差异。

安全、开发和DevOps团队的所有相关人员都必须明确自己的角色，以保护其创建、提取和在管道中传送的数字资产。只有以可随组织的发展而扩展的方式来开展这些活动，您才可以在未来取得成功，但前提是您要采用协调一致的应用安全方法 (AppSec)。传统的AppSec方法是在开发过程中“附加”测试、分类和修复措施，这无法应对现代网络罪犯带来的危险，也无法满足现代DevOps的工作流要求。

履行DevSecOps承诺：兼顾安全和速度

我们需要一种新的应用安全方法，在不妨碍公司发展的前提下处理业务风险，消除速度和安全之间神秘的权衡，履行DevSecOps承诺。

为了实现这一愿景，组织机构必须制定高效、有效的DevSecOps策略，这涵盖三个关键理念。

- 跨SDLC和CI管道建立应用安全性，以实现持续测试
- 在不牺牲安全性的前提下保持开发速度
- 优化工作流，以提高各团队的风险意识和安全能力

遵循该策略，您可将安全机制集成到SDLC的各个阶段，并获得统一DevSecOps所必需的可视性和可控性。它还允许安全和开发团队对风险和修复优先级进行客观评估，并且跨越工具和角色来构建顺畅的工作流。

那么，如何达成这一目标呢?首先，我们需要了解将安全性集成到DevOps中的最大挑战之一：每个团队都在使用多种不同的工具和系统。

将安全性转移到任何地方：持续快速测试

从智能手机上的应用，到支撑组织机构、行业和政府的系统，软件已经成为我们日常生活中不可或缺的一部分。但是，软件不是一体式产品，而是由多个组件构成的，这些组件通过协议和系统连接在一起，以实现操作和数据传输的统一执行。

贵公司的软件都是由您自己编写的代码、借用的软件和购买的软件组成的，无论是内部使用还是提供给外部客户和合作伙伴的软件。第一类是由内部开发人员构建的，旨在满足贵公司独特的个性化需求；第二类是指开源软件或来自第三方库的软件，旨在帮助开发人员加快开发速度，但需要根据许可限制使用、更新和分发。最后一类是组织机构从外部供应商手中购买的第三方许可的软件二进制文件，它们可以集成到您的项目和系统中，或与它们并行运行。

这三类软件都可以使用多种框架和技术来构建，以便将代码和组件转换为操作程序。例如用于隔离软件功能元素的容器，用于敏捷扩展的微服务，用于定义云资源设置的基础架构即代码 (IaC) 模板和API协议等。

测试和风险分析机制因技术而异，因此，随着新技术的不断出现，建立软件安全性变得越来越复杂。每当有新的测试周期“附加”到DevOps工作流中，都会加大在开发高质量软件和快速发布周期之间找到合适平衡点的难度。大多数组织都对AST设有基线，您应根据组织的基线制定DevSecOps策略。

- 检测专有代码中的不良或不安全的编码实践，并尽早在开发者桌面上进行检测，以加速修复并避免将问题推向下游
- 识别开源组件中的已知漏洞，包括在构建过程中无意间添加到项目中的传递依赖项
- 在运行时验证数据和应用功能的安全性，检测编译好的代码中是否存在潜在恶意活动

这些AST机制可以识别不同技术中的风险，为DevSecOps的参与者提供重要的洞察。然而，仅仅检测风险还不够，还需要采取有效的措施来修复风险。在过去，风险检测通常由AppSec团队负责，而风险修复则由开发团队来完成。

但在协同的DevSecOps项目中，这些责任是相互关联的，AST作为一个持续的过程，能够促进开发、安全和DevOps团队之间的闭环沟通，共同确定问题修复活动的优先级，并通过编写新代码或修改原代码来解决问题。要想有效做到这一点，需要将各种测试技术和安全机制统统集成到SDLC、CI管道和其他DevOps工作流中。这种集成的好坏将直接影响每个参与团队的业绩和成果。

消除摩擦：通过集成和自动化实现统一DevSecOps

如想完全实现DevSecOps，每项测试都需要集成到既有的工作流中，以确保能够触发相关测试，但不会触发可能导致软件交付延迟的不必要测试。有效的DevSecOps还需要根据客观安全标准自动执行基于风险的策略。这提供了一些关键能力，包括：

- 集成安全门，能够在源头或源头附近识别风险
- 安全防护能力，能够实时发现从无关或非法工作流和管道进入主项目的风险
- 将已确定优先顺序的风险分析结果快速清晰地传达给开发者，以进行修复
- 灵活可扩展的DevSecOps，能够随组织机构的发展而扩展

自动化是DevSecOps过程的重要组成部分。其主要目的是允许AppSec、开发和DevOps团队以最少的人工干预来执行日常和必要的任务。对于DevSecOps，自动化必须基于深思熟虑的、既定的策略，风险承受阈值和专门构建的安全门，以满足安全团队和开发团队的要求（如检测时间、修复时间、代码交付截止时间等）而专门构建的安全门。合理的AppSec策略是帮助您实现目标以及构建集成式自动化DevSecOps的基本机制。

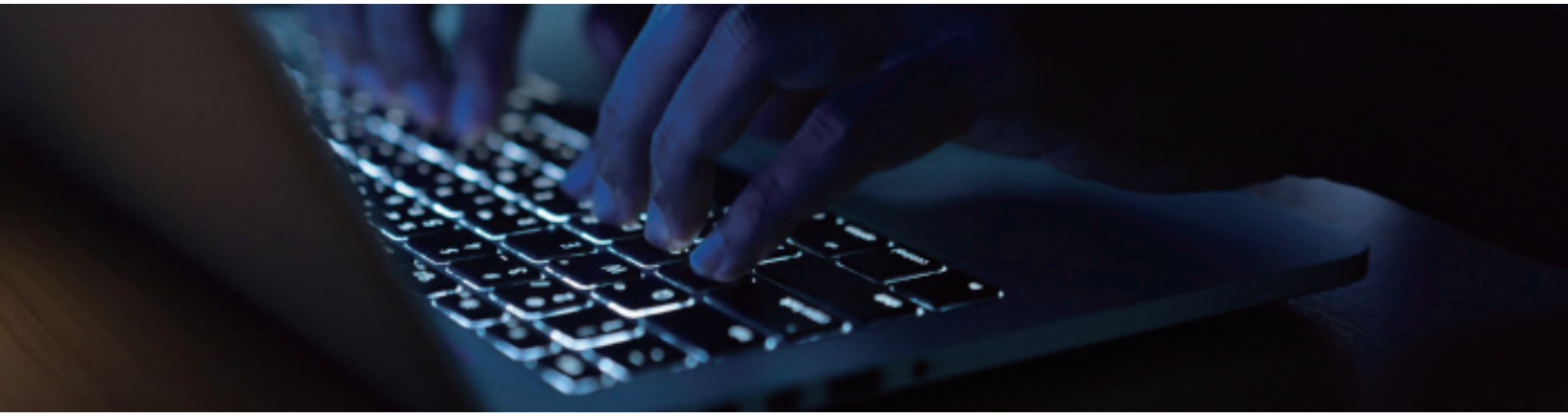
鉴于每个团队在定义与安全标准相关的策略时都发挥着独特的作用，因此，DevSecOps计划应该体现出一致、统一的组织愿景，即使在某些参与方缺席项目或冲刺的情况下也不例外。当违反策略时，利益相关者将自动收到警告，告诉他们项目或工作流的哪个部分出现了问题，并提醒他们需要采取预定义的行动。

一旦策略和标准建立起来，您就可以将AppSec集成到SDLC和CI管道中。在这里，“集成”应该被定义为将您的AppSec工具连接到现有的开发、安全和DevOps系统和工作流中。这包括：

- 基于管道活动、代码变更或新代码提交到源代码管理 (SCM) 和二进制仓库等来触发AST扫描，以加速风险检测
- 通过问题管理工作流和工具提供已确定优先顺序的风险分析结果和修复指导，以加速修复
- 利用DevOps工具（如GitHub Actions和GitLab Templates）的内置功能来支持自动测试、评论和修复拉取请求
- 阻止传送有漏洞的资产，或者打破违反安全策略的管道构建，以防止问题在下游出现

通过在现有的开发和DevOps工具、流程和软件之上集成这些类型的安全和DevSecOps措施，您可以让开发者更高效地工作，更快地修复问题，避免在后期重构代码，并交付更安全、更高质量的软件，从而降低贵组织的风险暴露。

当然，让开发者接受新的工具和平台可能具有挑战性，特别是当他们认为这些工具与其必要的日常任务无关时，这可能是许多AppSec测试工具遇到的情况。集成可使软件分析成为开发工作流的一部分，以便安全团队能够无缝分析软件，并且直接在开发者的IDE、协作工具和问题管理工作流中提供漏洞信息，从而大大降低将安全性纳入DevOps工作流的障碍，并消除主要参与者之间的摩擦。



根据开发者的具体情况来培养其安全能力

开发者需要快速工作，而且他们中的大多数人都根据自身需求定制了自己的IDE和工作流。尽管这些细微的开发环境和管道会增加DevSecOps项目的复杂性，但有一个有助于建立更统一安全原则的共同点：参与其中的开发人员。

如前所述，跨SDLC和CI管道集成自动化AST机制，有助于建立安全门和制衡系统，以尽可能在风险源头或其附近发现潜在的安全问题，无论它们来自哪个入口。您可以通过培训开发人员的安全能力来大幅提高贵组织保护数字资产的能力，因为开发者的安全编码技能往往存在很大差异。2023年的SANS报告显示，近21%的组织认为，即使他们拥有结构化的开发者安全培训计划，但他们为开发者提供的安全培训仍然不充分。然而，只有31%的受访者认为，开发者安全培训计划是DevSecOps项目取得成功的关键因素。

简言之，通过为开发者建立更严格的安全标准，并让他们将这些标准作为日常工作的一部分进行实施，您可以防止引入新的问题，并加速修复现有问题。作为DevSecOps计划的一部分，实现这一目标需要：

- 在开发者工作流中，提供清晰、有优先级排序的安全风险洞察，指出需要修复的最紧急问题及其在源代码和文件系统中的位置
- 详细修复指导，告诉开发者解决已知问题的最有效和最高效方法，即使他们缺乏安全知识或经验来独立完成修复工作
- 让开发者在他们的首选集成开发环境（IDE）中编写代码，像拼写检查一样实时获得风险意识和修复建议，即“安全拼写检查”体验，从而在合并或构建代码之前，写出更安全的代码
- 基于开发者的项目、技术和更大的业务需求，为其提供安全培训和安全编码教育，培养其与业务同步发展的安全能力

安全意识与安全能力相结合，可将集成式DevSecOps计划从条件触发的自动安全门禁系统提升为一种协调一致的方法，以实现敏捷的风险检测与快速风险处理，并且越来越高效。

安全意识与安全能力相结合，可将集成式DevSecOps计划从条件触发的自动安全门禁系统提升为一种协调一致的方法，以实现敏捷的风险检测与快速风险处理，并且越来越高效。这种基于贵组织的概况、技术和监管标准，专为贵组织定制的基于最佳实践的安全能力有助于跨项目和团队开展更客观的安全评估，取代开发人员对“安全”的主观评估。

当检测到问题时，应该给开发者提供明确的修复建议。这不仅有助于解决问题，而且能为开发者以后的工作规范提供参考。同时，还应根据开发者所用的技术、语言和框架安排相应的安全培训，让开发者的学习更有针对性和效果。这种培训应融入到现有的开发工作流中，并与问题管理工具和IDE集成，以便开发者能够直接获取与安全测试相关的易于使用的学习主题。

最后，应在开发者编写代码时激活这些安全功能，并通过IDE及时告知他们引入的新问题，这一点很重要。您可以通过在开发者终端进行本地安全测试来做到这一点，例如检查他们的代码是否存在编码缺陷（如跨站脚本错误，未定义超时等）以及是否包含存在已知漏洞的开源组件或库。

这种本地测试不应取代AppSec团队开展的基于管道的测试或集中安全测试。相反，它应作为一种提升开发者问题认知能力的机制，使他们及早注意到那些在开发初期更容易处理的问题，避免在后续的应用安全测试中影响或干扰安全团队的工作。因此，这样做的好处是双重的：既能加快修复速度、又能减轻AppSec团队的负担，并且所有这些都不需要偏离既定的开发工作流。

实现DevSecOps：人员、流程和规划

DevSecOps依赖于组织内部各种角色的成功协作与配合。虽然AppSec团队可能会继续负责软件的安全风险态势，但其实施安全措施和解决已知问题的能力却取决于开发和DevOps团队。许多组织机构在制定DevSecOps策略时之所以会遇到阻碍，都是因为他们未能针对每个参与团队确定成功标准和绩效指标并将其纳入到策略中。

对大多数组织来说，跨团队讨论都会得出一系列的标准和绩效指标，类似于以下列表：

- 检测时间
- 修复时间
- AppSec测试中的误报
- 修复成本

然而，对于现代DevSecOps来说，该列表并不能提供对安全规划状态的有用洞察，也没有考虑到集成式应用安全规划需要随组织的发展而不断变化的必要性。速度问题可以通过扩大AppSec集成在软件开发生命周期中的广度和深度，以及实施基于策略的自动化来解决。误报率取决于所用的AST工具以及测试点。修复成本会受到多个因素的影响，包括修复问题位置与问题产生位置之间的距离，或者负责修复问题的开发团队的熟练程度等等。

因此，明确每个参与团队的成功标准以及可以影响这些标准的DevSecOps机制很重要。一般来说，无论各团队重视的关键因素是什么，任何一名参与者都应具备评估DevSecOps规划的状况和表现的能力，只有这样，他们才有可能支持此类规划及其方案。

2023年的SANS调查显示，52%的受访者认为跨团队沟通是一个关键成功要素，而超过41%的受访者认为组织孤岛是成功实施DevSecOps的障碍，36%的受访者认为缺乏透明度是一个紧迫问题。显然，您能否获得参与团队的支持在很大程度上取决于您能否消除他们对DevSecOps规划可能无效或低效的担忧。

为了消除这种顾虑，您可以考虑组织级的DevSecOps方法，它能够反映每个团队的不同需求并支持一系列重要的成功指标：

- 平滑地集成安全测试，不仅能够避免开发团队无法按时完成任务，而且还能避免因安全测试失败而导致的后期返工，从而提高效率
- 开发人员能够在IDE和问题管理工作流中查看需要修复的新问题和既有问题，并按照优先级进行处理
- 在开发和安全团队之间建立闭环反馈，以及时通告问题解决情况
- 当发现可能影响现有软件安全性的新问题时，及时发出告警，以便运维和DevOps团队能够检查和确认生产资产的安全状况
- 清晰、可管理、具有优先级排序的风险报告，以最大限度地减少误报数量并减少漏洞积压
- 可以通过各种行动、流程和支持技术来积极地影响成功标准，无需从根本上改变对参与团队的要求。这是一个有效、灵活的DevSecOps规划的特征。

奠定坚实基础：随业务演变的基于平台的AST

贵组织如何在不牺牲安全性的前提下消除复杂性，降低成本并提高可扩展性?能否通过对开发者、安全人员和DevOps团队有益的方式来实现这些目标?鉴于云平台和环境具有价格优惠、可扩展、灵活和易于管理等特点，许多组织均已开始采用基于云的解决方案来构建开发工具链。组织机构希望从应用安全测试工具中获得这些好处，但一直以来，他们总是被迫在基本需求上做出一些牺牲。易用的平台可能无法检测出复杂应用中的安全隐患，而在本地快速运行的工具可能无法满足企业的扩展要求。托管式AST工具可能在静态应用安全测试 (SAST) 方面表现良好，但是在软件组成分析 (SCA) 方面表现不佳，反之亦然。

基于SaaS的AST平台可以提供多个分析引擎来帮助保护专有代码和第三方软件。

基于SaaS的AST平台可以提供多个分析引擎来帮助保护专有代码和第三方软件。使用这种方法，您可以消除冗余的工具或复杂的实施流程，选择由集中策略管理的统一平台，实现跨CI管道集成，并提供跨项目和测试的带有优先级排序的风险洞察。这对于正在执行DevSecOps规划，但没有资源、需求或意愿在所有开发、构建和测试环境中管理本地实施的组织来说尤其有价值。

此外，这还能确保您的DevSecOps规划从一开始就具有灵活性和可扩展性，不会受到不同AST工具的技术限制或偏差的影响。这种基于平台的AST即服务方法可以解决被许多人视为DevSecOps障碍的可视性和跨团队沟通问题，它可以跨越项目和团队提供风险和修复趋势的实时洞察，使您更容易识别重点领域并做出必要的调整。

构建有效的集成式DevSecOps：获得商业收益

应用安全的重要性远远不仅限于保护软件系统安全，而是已经演变成为关键的商业驱动力。如今，作为复杂软件供应链的参与者，您不仅需要确保引入到项目中的软件和组件的安全性，同时还要认识到客户和合作伙伴需要您提供合理的软件安全证明。这种安全需求的层层传递意味着客户、合作伙伴和最终用户越来越重视贵组织的风险态势和安全控制，包括：

- 检测问题的时间
- 解决问题的时间
- 准确的软件材料清单和相关安全证明

似曾相识?这是因为利益相关者对DevSecOps规划也提出了类似要求，将其作为关键标准。您的同事和参与团队之所以会关注这些问题，主要是因为他们也是数字资产的消费者，就像您的客户和合作伙伴一样。

随着您的战略愿景和明确定义的DevSecOps规划得到同事的支持和参与，它也将成为潜在的商业驱动力和竞争优势。合理定义和实施的DevSecOps规划应具有以下几个显著特征：

- 主动安全风险检测
- 快速风险修复
- 它所支持的应用具有高可靠性和可用性
- 通过软件和相连系统传递的敏感信息安全且不可访问

通过为所有参与团队集成DevSecOps实践、基于策略的自动化，以及风险和解决方案的端到端可视性，您可以快速将应用安全从成本中心转变为业务资产。这种情况下，应用安全将直接影响贵组织的销售能力，成为您在竞争中脱颖而出的关键因素。从本质上讲，在当前互联水平和安全意识日益增强的世界中，强大的安全实践已经开始与企业收入挂钩，因此，证明具备此项能力已经成为所有前瞻性组织的当务之急。

关于 Black Duck

Black Duck与众不同

Black Duck提供业界最全面、最强大、最值得信赖的应用安全解决方案组合。我们拥有无与伦比的专业知识和经验，来帮助世界各地的组织机构快速保护其软件，在其开发环境中高效集成安全性以及使用新技术进行安全创新。作为软件安全领域公认的领导者、专家和创新者，Black Duck拥有您构建可信软件所需的一切。如预了解更多信息，请访问www.blackduck.com。