AWS Well-Architected 框架

安全支柱

安全支柱: AWS Well-Architected 框架

Table of Contents

安全性支柱 – AWS Well-Architected Framework

发布日期：2024 年 11 月 6 日（文档修订）

本白皮书主要介绍 AWS Well-Architected Framework 的安全性支柱。该白皮书提供的指导，有助于您在安全 AWS 工作负载的设计、交付和维护过程中应用最佳实践和最新建议。

简介

AWS Well-Architected Framework 能够帮助您理解在 AWS 上构建工作负载时所做决策的权衡取舍。通过使用此框架，您将了解有关在云中设计和运行可靠、安全、高效、经济实惠且可持续的工作负载的最新架构最佳实践。该框架提供了一种方法，让您能够根据最佳实践持续衡量工作负载，从而确定需要改进的方面。我们相信，拥有架构完善的工作负载能够大大提高实现业务成功的可能性。

该框架基于六大支柱：

- 卓越运营
- 安全性
- 可靠性
- 性能效率
- 成本优化
- 可持续性

本白皮书重点介绍安全性支柱。这样可以帮助您遵循最新的 AWS 建议，从而满足您的业务和法规要求。本白皮书的目标读者是技术岗位的人员，例如首席技术官（CTO）、首席信息安全官（CSO/CISO）、架构师、开发人员和运营团队成员。

阅读本白皮书后，您将了解可在设计注重安全的云架构时使用的 AWS 最新建议和策略。本白皮书不提供实施细节或架构模式，但会针对此类信息提供适当资源参考。通过采用本白皮书中的实践，您可以构建能够保护您的数据和系统、控制访问并自动响应安全事件的架构。

安全基础知识

安全支柱介绍了如何利用云技术来保护数据、系统和资产，从而改善您的安全状况。本白皮书深度介绍了有关在 AWS 上构建安全工作负载的最佳实践指导。

设计原则

在云中，有很多原则可帮助您提高工作负载的安全性：

- 实施强大的身份验证基础：实施最低权限原则，并通过对每一次与 AWS 资源之间的交互进行适当授权来强制执行职责分离。集中进行身份管理，并努力消除对长期静态凭证的依赖。
- 保持可追溯性：实时监控和审查对环境执行的操作和更改并发送警报。为系统集成日志和指标收集功能，以自动调查并采取行动。
- 在所有层面应用安全措施：利用多种安全控制措施实现深度防御。应用到所有层面（例如网络边缘、VPC、负载均衡、每个实例和计算服务、操作系统、应用程序和代码）。
- 自动化安全性最佳实践：借助基于软件的自动化安全机制，您能够以更为快速且更具成本效益的方式实现安全扩展。创建安全架构，包括实施可在版本控制模板中以代码形式定义和管理的控制措施。
- 保护传输中数据和静态数据：按敏感程度对数据进行分类，并根据情况采用加密、令牌化和访问控制等适当机制。
- 限制对数据的访问：利用相关机制和工具来减少或消除对于直接访问或手动处理数据的需求。这样可以降低处理敏感数据时数据处理不当、被修改以及人为错误的风险。
- 做好应对安全性事件的准备工作：制定符合组织要求的事件管理和调查政策和流程，做好应对意外事件的准备工作。开展意外事件响应模拟演练，并使用具有自动化功能的工具来提高检测、调查和恢复的速度。

定义

云中的安全性包含七个方面：

- 安全基础知识
- 身份和访问管理
- 检测
- 基础设施保护
- 数据保护

- 事件响应
- 应用程序安全性

责任共担

安全性和合规性是 AWS 与客户共同承担的责任。这种共担模式可以减轻客户的运营负担，因为 AWS 会运营、管理和控制从主机操作系统和虚拟化层组件，一直到服务运营所在物理设施的安全性。客户负责管理来宾操作系统（包括更新和安全补丁）、其他关联应用程序软件以及 AWS 提供的安全组防火墙的配置。客户应慎重选择服务，因为他们所承担的责任因他们使用的服务、服务与其 IT 环境的集成以及适用法律法规而各异。这种责任共担的性质还赋予了客户足够的灵活性和控制能力来进行部署。如下图所示，责任的这种区分通常称为云“本身的”安全性与云“中的”安全性。

AWS“云的安全性”责任 – AWS 负责保护运行 AWS 云中提供的所有服务的基础设施。该基础设施由运行 AWS 云服务的硬件、软件、网络和设施组成。

客户“云中的安全性”责任 – 客户责任将由客户选择的 AWS 云服务确定。这决定了客户必须执行的作为其安全责任一部分的配置工作量。例如，Amazon Elastic Compute Cloud（Amazon EC2）等服务被归类为基础设施即服务（IaaS），因此，这需要客户执行所有必要的安全配置和管理任务。部署 Amazon EC2 实例的客户负责管理来宾操作系统（包括更新和安全补丁）、客户在实例上安装的任何应用程序软件或实用程序，以及每个实例上由 AWS 提供的防火墙（称为安全组）的配置。对于抽象服务（例如 Amazon S3 和 Amazon DynamoDB），AWS 会运营基础设施层、操作系统和平台，而客户访问端点即可存储和检索数据。客户负责管理他们的数据（包括加密选项）、对其资产进行分类以及使用 IAM 工具应用适当的权限。

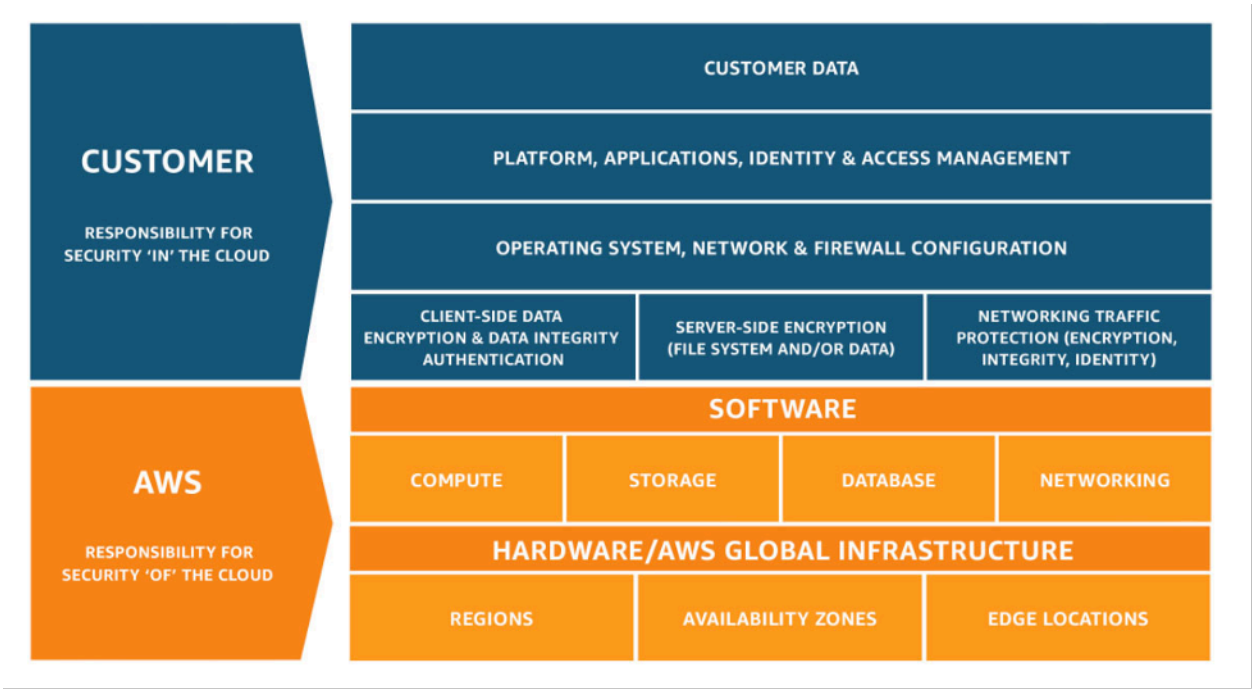


图 1：AWS 责任共担模式

此客户/AWS 责任共担模式还扩展到 IT 控制措施方面。正如 AWS 与客户共同行使控制 IT 环境的责任一样，管理、运营和验证 IT 控制措施的责任也是由双方共同承担。AWS 负责管理与部署在 AWS 环境中的物理基础设施相关的控制措施（此管理工作此前可能由客户承担），从而帮助客户缓解操作控制措施的负担。因为每个客户在 AWS 中的部署均不相同，所以客户可以藉此将管理特定 IT 控制措施的责任移交到 AWS，从而形成一个新型分布式控制环境。然后客户可以使用 AWS 控制和合规性文档来根据需要执行控制评估与验证程序。以下是由 AWS 和/或 AWS 客户管理的控制措施示例。

继承的控制措施– 客户完全从 AWS 继承的控制措施。

- 物理和环境控制措施

共享的控制措施 – 应用于基础设施层和客户层的控制措施，但在单独的上下文中或从单独的视角应用。在共享的控制措施中，AWS 提供了对基础设施的要求，客户必须对其使用 AWS 服务实施自己的控制措施。示例包括：

- 补丁管理 – AWS 负责修补和修复基础设施中的缺陷，而客户负责修补他们的来宾操作系统和应用程序。
- 配置管理 – AWS 维护其基础设施设备的配置，而客户负责配置他们自己的来宾操作系统、数据库和应用程序。
- 意识与培训 – AWS 培训 AWS 的员工，而客户必须对自己的员工进行培训。

客户特定 – 完全由客户负责的控制措施，基于客户在 AWS 服务中部署的应用程序。示例包括：

- 服务和通信保护或区安全，这可能需要客户在特定的安全环境中路由数据或将数据分区。

Governance

安全治理是总体方法的一部分，旨在通过制定策略和控制目标来帮助管理风险，从而帮助实现业务目标。通过遵循安全控制目标的分层方法来实施风险管理，其中每一层均基于前一层而构建。了解 AWS 责任共担模式是您的基础层。这方面的知识阐明了您需对客户承担的责任以及您从 AWS 得到了什么。AWS Artifact 是一种实用资源，让您可以按需访问 AWS 安全性与合规性报告以及选定的在线协议。

在下一层满足您的大部分控制目标。在该层提供了平台范围的功能。例如，该层包括 AWS 账户分配过程、与身份提供程序（例如 AWS IAM Identity Center）的集成以及常见的检测性控制。平台治理过程的一些输出也位于该层。在您希望开始使用新的 AWS 服务时，更新 AWS Organizations 服务中的服务控制策略（SCP）可为服务的初次使用提供防护机制。您可以使用其他 SCP 来实施常见的安全控制目标，这通常称为安全不变量。这些是您应用于多个账户、组织单位或整个 AWS 组织的控制目标或配置。典型示例是限制运行基础设施的区域，或防止停用检测性控制措施。该中间层还包含编码策略，例如配置规则或签入管道。

顶层是产品团队满足控制目标的地方，这是因为实施是在产品团队控制的应用程序中完成的。这可能是在应用程序中实施输入验证或确保身份在各项微服务之间正确传递。尽管产品团队负责配置，他们也仍能从中间层继承一些功能。

无论您在何处实施控制措施，目标都是一致的，即管理风险。一系列风险管理框架将应用于特定的行业、区域或技术。您的主要目标：根据可能性和后果来强调风险。这就是固有风险。紧接着，您可以定义控制目标，降低可能性和/或减少后果。随后，采用适当的控制措施后，您可以查看可能产生哪些风险。这就是剩余风险。控制目标可应用于一个或多个工作负载。下图显示了一个典型的风险矩阵。可能性基于以前发生事件的频率，而后果基于事件的财务、声誉和时间成本。

Likelihood	Risk Level				
Very Likely	Low	Medium	High	Critical	Critical
Likely	Low	Medium	Medium	High	Critical
Possible	Low	Low	Medium	Medium	High
Unlikely	Low	Low	Medium	Medium	High
Very unlikely	Low	Low		Medium	High
Consequence	Minimal	Low	Medium	High	Severe

图 2：风险等级可能性矩阵

AWS 账户管理和分离

我们建议您根据职能、合规性要求或一组通用控制措施，在单独的账户和组账户中组织工作负载，而不是完全沿用您企业的报告结构。在 AWS 中，账户是硬性边界。例如，强烈建议执行账户级分离，以使生产工作负载与开发和测试工作负载分离。

集中管理账户：AWS Organizations 会自动创建和管理 AWS 账户，并在创建之后控制这些账户。在通过 AWS Organizations 创建账户时，请务必考虑使用您的电子邮件地址，因为这将是允许重置密码的根用户。Organizations 允许您根据工作负载的要求和用途，将账户分组成代表不同环境的组织部门（OU）。

集中设置控制：在适当的级别，只允许特定的服务、区域和服务操作，以控制您的 AWS 账户能够执行的操作。AWS Organizations 允许您使用服务控制策略（SCP），在组织、组织部门或账户级别应用权限防护机制，此操作适用于所有 AWS Identity and Access Management（IAM）用户和角色。例如，您可以利用 SCP 禁止用户从您未明确允许的区域启动资源。AWS Control Tower 能够以一种简化的方式设置和管理多个账户。它会自动在您的 AWS Organization 中设置账户、自动预置、应用防护机制（包括预防和检测），并提供一个控制面板供您获得可见性。

集中配置服务和资源：AWS Organizations 可帮助配置能够应用于您所有账户的 AWS 服务。例如，您可以使用 AWS CloudTrail 配置集中日志记录功能，记录组织中执行的所有操作，也可以禁止成员账户禁用日志记录功能。您也可以使用 AWS Config 集中聚合自己定义的规则的数据，以便能够审计工作负载是否合规，并快速对变化做出反应。AWS CloudFormationStackSets 允许您在组织中跨账户和组织部门集中管理 AWS CloudFormation 堆栈。这样，您就可以自动预置一个新账户来满足自己的安全要求。

使用安全服务的委托管理功能，将用于管理的账户与组织计费（管理）账户分隔开。多项 AWS 服务（例如，GuardDuty、Security Hub 和 AWS Config）支持与 AWS Organizations 的集成，包括为管理功能指定特定的账户。

最佳实践

- SEC01-BP01 使用账户分隔工作负载
- SEC01-BP02 保护账户根用户和属性

SEC01-BP01 使用账户分隔工作负载

通过采取多账户策略，在环境（如生产、开发和测试）和工作负载之间建立共同的防护机制和隔离措施。强烈建议在账户层面进行分离管理，这样可为安全性、账单和访问提供强大的隔离边界。

期望结果：形成一种账户结构，可将云运维、无关工作负载和环境隔离到单独的账户中，从而提高整个云基础设施的安全性。

常见反模式：

- 将多个相互毫无关联，具有不同数据敏感度级别的工作负载放入同一账户中。
- 组织单位（OU）结构界定不清。

建立此最佳实践的好处：

- 即使不该访问的工作负载无意中被访问了，影响范围也会缩小。
- 能够对访问 AWS 服务、资源和区域进行集中治理。
- 可集中管理策略和安全服务，维护云基础设施的安全性。
- 实现账户创建和维护流程自动化。
- 集中审核基础设施状况，从而满足法规遵从性和监管要求。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

AWS 账户提供安全隔离边界，使不同敏感度的工作负载或资源相互分离。AWS 提供相应工具，以多账户策略来大规模管理云工作负载，从而利用此隔离边界。如要获得有关 AWS 多账户策略的概念、模式和实施的指导，请参阅《Organizing Your AWS Environment Using Multiple Accounts》白皮书。

如果需要集中管理多个 AWS 账户，账户应基于组织单位（OU）层建立层次结构。然后可以建立安全控制机制，并将其应用于 OU 和成员账户，从而为组织内的成员账户建立一致的预防性控制机制。安全控制机制是层层继承的，使您能够筛选位于 OU 层次结构较低层次的成员账户的可用权限。优秀的架构设计将能够利用这种层层继承的特性，减少设置安全策略，降低复杂性，并使每个成员账户的安全控制效果达到预期。

采用 AWS Organizations 和 AWS Control Tower 这两种服务，可在您的 AWS 环境中实施和管理多账户结构。AWS Organizations 使得您能够将账户建立成由一个或多个 OU 层定义的层次结构形式，每个 OU 均可包含若干成员账户。服务控制策略（SCP）使组织管理员能够对成员账户建立精细的预防

性控制机制，而 AWS Config 可用于建立对成员账户的主动式和检测性控制。许多 AWS 服务与 AWS Organizations 集成，可提供委派型管理控制，并在组织内的所有成员账户中执行服务特定的任务。

AWS Control Tower 位于 AWS Organizations 之上，为具有登录区的多账户 AWS 环境提供了一键式最佳实践设置。登录区是由 Control Tower 建立的多账户环境的入口处。与 AWS Organizations 相比，采用 Control Tower 具有若干好处。可以改进账户治理状况的三种好处为：

- 将强制安全防护机制集成于系统中，可自动应用于准入组织的账户。
- 有多种防护机制可供选择，还能开启或关闭给定 OU 组的防护机制。
- AWS Control Tower Account Factory 可在组织内部自动部署账户，设置好预先批准的基准和配置选项。

实施步骤

1. 设计组织单位结构：设计良好的组织单位结构减少了创建和维护服务控制策略及其他安全控制机制所需的管理负担。组织单位结构应与业务需求、数据敏感度和工作负载结构看齐。
2. 为多账户环境创建登录区：登录区提供了一致的安全性和基础设施基础，让组织可以从中快速开发、启动和部署工作负载。您可以使用定制的登录区或 AWS Control Tower 来编排环境。
3. 建立防护机制：通过登录区为环境实施一致的安全防护机制。AWS Control Tower 提供了可部署的必选和可选控制机制的列表。实施 Control Tower 时会自动部署必选控制机制。查看高度推荐和可选控制机制的列表，并实施适合您需求的控制机制。
4. 限制访问新添加的区域：对于新的 AWS 区域，诸如用户和角色之类的 IAM 资源将仅传播到您指定的区域。可以在使用 Control Tower 时通过控制台执行此操作，也可以通过调整 AWS Organizations 中的 IAM 权限策略执行此操作。
5. 考虑使用 AWS CloudFormation StackSets：StackSets 可帮助您通过已批准的模板将资源（包括 IAM 策略、角色和组）部署到不同的 AWS 账户和区域中。

资源

相关最佳实践：

- SEC02-BP04 依赖集中式身份提供程序

相关文档：

- AWS Control Tower

- AWS 安全审计指南
- IAM 最佳实践
- Use CloudFormation StackSets to provision resources across multiple AWS 账户 and regions
- Organizations 常见问题
- AWS Organizations 术语和概念
- Best Practices for Service Control Policies in an AWS Organizations Multi-Account Environment
- AWS Account Management Reference Guide
- 使用多个账户整理您的 AWS 环境

相关视频：

- Enable AWS adoption at scale with automation and governance
- Security Best Practices the Well-Architected Way
- Building and Governing Multiple Accounts using AWS Control Tower
- Enable Control Tower for Existing Organizations

SEC01-BP02 保护账户根用户和属性

根用户是 AWS 账户中权限最高的用户，对账户内的所有资源具有完全管理访问权限，在某些情况下不受安全策略的约束。停用对根用户的编程访问，为根用户建立适当的控制机制，并避免日常使用根用户，这样有助于降低无意中暴露根凭证以及随后破坏云环境的风险。

期望结果：保护根用户有助于减少因滥用根用户凭证而导致意外或故意损坏的可能性。建立检测性控制机制也可以在有人使用根用户执行操作时向适当人员发出警报。

常见反模式：

- 使用根用户执行各种任务，而非仅在必要时使用根用户凭证。
- 忽略定期测试应急计划，不验证关键基础设施、流程和人员在紧急情况下的运作情况。
- 只考虑典型的账户登录流程，而没有考虑或测试替代的账户恢复方法。
- 因为 DNS、电子邮件服务器和电话提供商要用于账户恢复流程，就不将其作为关键安全边界的一部分进行处理。

建立此最佳实践的好处：保护对根用户的访问可以建立信心，让账户中的操作受到控制和审核。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

AWS 提供许多有助于保护账户安全的工具。但由于其中一些措施默认情况下未启用，因此您必须采取直接行动来实施这些措施。请将这些建议视为确保 AWS 账户安全的基本步骤。实施这些步骤时，务必建立一个可持续评测和监控安全控制机制的过程，这非常重要。

当您首次创建 AWS 账户时，最初使用的是一个对账户中所有 AWS 服务和资源有完全访问权限的身份。此身份称作 AWS 账户根用户。您可以使用在创建账户所用的电子邮件地址和密码以根用户身份登录。由于授予 AWS 根用户的访问权限较高，您必须仅将 AWS 根用户用于执行特别需要它的任务。必须严格保护根用户登录凭证，并且应始终为 AWS 账户根用户使用多重身份验证（MFA）。

除了使用用户名、密码和多重身份验证（MFA）设备登录根用户的常规身份验证流程外，还可以使用账户恢复流程登录您的 AWS 账户 根用户，该用户可以访问与您的账户关联的电子邮件地址和电话号码。因此，保护发送恢复电子邮件的根用户电子邮件账户和保护与该账户关联的电话号码同样重要。还应考虑潜在的循环依赖关系，其中与根用户关联的电子邮件地址托管在同一 AWS 账户的电子邮件服务器或域名服务（DNS）资源上。

使用 AWS Organizations 时，有多个 AWS 账户（每个均有一个根用户）。将一个账户指定为管理账户，然后可以在管理账户下面添加几层成员账户。优先保护管理账户的根用户，然后解决成员账户根用户问题。保护管理账户根用户的策略可能与保护成员账户根用户的策略不同，您可以对成员账户根用户建立预防性安全控制机制。

实施步骤

建议使用以下实施步骤为根用户建立控制机制。在适用情况下，建议与 CIS AWS Foundations Benchmark 版本 1.4.0 交叉引用。除了这些步骤外，请参阅 AWS 最佳实践指导来确保 AWS 账户和资源安全。

预防性控制机制

1. 为账户设置准确的联系信息。
    a. 该信息用于丢失的密码恢复流程、丢失的 MFA 设备账户恢复流程，以及与您的团队进行关键的安全相关通信。
    b. 使用企业域托管的电子邮件地址（最好是通讯组列表）作为根用户的电子邮件地址。使用通讯组列表而不是个人的电子邮件账户可提供额外的冗余和连续性，以便在很长一段时间内访问根账户。
    c. 联系信息上所列的电话号码应该是为此目的而设置的专用安全电话的号码。电话号码不应列出或与任何人共享。

2. 不要为根用户创建访问密钥。如果存在访问密钥，请将其删除（CIS 1.4）。
   a. 消除根用户的任何长期编程凭证（访问密钥和私有密钥）。
   b. 如果已存在根用户访问密钥，您应将使用这些密钥的进程转换为使用 AWS Identity and Access Management（IAM）角色的临时访问密钥，然后删除根用户访问密钥。
3. 确定是否需要为根用户存储凭证。
   a. 如果您使用 AWS Organizations 创建新的成员账户，则新成员账户上根用户的初始密码将设置为一个不向您公开的随机值。如果需要，请考虑使用 AWS 组织管理账户的密码重置流程来访问成员账户。
   b. 对于独立 AWS 账户或管理 AWS 组织账户，请考虑为根用户创建并安全地存储凭证。为根用户启用 MFA。
4. 在 AWS 多账户环境中，为成员账户根用户使用预防性控制机制。
   a. 考虑为成员账户启用不允许为根用户创建根访问密钥预防性防护机制。
   b. 考虑为成员账户启用不允许以根用户身份执行操作预防性防护机制。
5. 如果需要根用户凭证，请执行以下操作：
   a. 使用复杂密码。
   b. 为根用户启用多重身份验证（MFA），特别是 AWS Organizations 管理（付款人）账户（CIS 1.5）。
   c. 考虑使用硬件 MFA 设备来提高韧性和安全性，因为一次性设备可以减少包含 MFA 代码的设备被重复用于其他用途的可能性。验证是否定期更换由电池供电的硬件 MFA 设备。（CIS 1.6）
      - 要为根用户配置 MFA，请遵循创建虚拟 MFA 或硬件 MFA 设备的说明。
   d. 考虑注册多个 MFA 设备用于备份。每个账户最多允许 8 个 MFA 设备。
      - 请注意，为根用户注册多个 MFA 设备将自动禁用在 MFA 设备丢失的情况下恢复账户的流程。
   e. 安全地存储密码，如果以电子方式存储密码，则考虑循环依赖关系。不要以需要访问同一 AWS 账户才能获得密码的方式存储密码。
6. 可选：考虑为根用户制定定期密码轮换计划。
   - 凭证管理最佳实践取决于您的监管和政策要求。受 MFA 保护的根用户并不依赖密码作为单重身份验证。
   - 定期更改根用户密码可降低无意中暴露的密码被滥用的风险。

侦测性控制

- 创建警报来检测根凭证的使用情况（CIS 1.7）。启用 Amazon GuardDuty 将通过 RootCredentialUsage 调查发现对根用户 API 凭证的使用进行监控和发出警报。

- 评估并实施适用于 AWS Config 的 AWS Well-Architected 安全性支柱合规包中包含的检测性控制机制，或者如果使用 AWS Control Tower，则评估并实施 Control Tower 内强烈建议的控制机制。

运营指导

- 确定组织中应该有权访问根用户凭证的人员。
  - 采用双人规则，以便不会出现一个人就能够访问所有必要凭证和 MFA 来获得根用户访问权限的情况。
  - 验证组织（而不是个人）对与账户关联的电话号码和电子邮件别名（用于密码重置和 MFA 重置流程）保持控制。
- 仅在例外情况下使用根用户（CIS 1.7）。
  - 不得使用 AWS 根用户执行日常任务，即使是管理任务也不可以。仅以根用户身份登录，以执行需要根用户的 AWS 任务。所有其他操作都应由代入适当角色的其他用户执行。
- 定期检查对根用户的访问是否正常，以便在出现需要使用根用户凭证的紧急情况之前对过程进行测试。
- 定期检查与账户关联的电子邮件地址以及备用联系人下列出的电子邮件地址是否有效。监控这些电子邮件收件箱，查看您可能从 <abuse@amazon.com> 中收到的安全通知。还要确保与该账户相关的任何电话号码都有效。
- 准备事件响应程序，应对根账户滥用情况。请参阅《AWS Security Incident Response Guide》以及《安全性支柱》白皮书“事件响应”部分中的最佳实践，了解有关为 AWS 账户构建事件响应策略的更多信息。

资源

相关最佳实践：

- SEC01-BP01 使用账户分隔工作负载
- SEC02-BP01 使用强大的登录机制
- SEC03-BP02 授予最低访问权限
- SEC03-BP03 建立紧急访问流程
- SEC10-BP05 预置访问权限

相关文档：

- AWS Control Tower

- AWS 安全审计指南
- IAM 最佳实践
- Amazon GuardDuty – root credential usage alert
- 通过 CloudTrail 监控根凭证使用情况的分步指导
- 获准与 AWS 一起使用的 MFA 令牌
- Implementing break glass access on AWS
- Top 10 security items to improve in your AWS 账户
- 发现我的 AWS 账户中存在未经授权的活动时该怎么办？

相关视频：

- Enable AWS adoption at scale with automation and governance
- Security Best Practices the Well-Architected Way
- Limiting use of AWS root credentials from AWS re:inforce 2022 – Security best practices with AWS IAM

安全地运营工作负载

安全地运营工作负载涵盖了从设计、构建、运行到持续改进的整个工作负载生命周期。为了增强您在云中安全运营的能力，其中一种方法是采用组织化的方法进行治理。治理是采用一致的方法来指导决策，而不是完全依赖于相关人员做出良好判断。您的治理模式和流程决定了您如何回答以下问题：“我如何知道给定工作负载的控制目标已实现并且适用于该工作负载？”采用一致的方法来制定决策可以加快部署工作负载，并帮助提高组织中的安全能力标准。

为了安全地操作您的工作负载，您必须将纲领性最佳实践应用于每个安全领域。将您在组织和工作负载级别的卓越运营中定义的要求和流程应用于所有领域。及时了解 AWS 和行业建议及威胁情报可以帮助您发展威胁模型和控制目标。实现安全流程、测试和验证的自动化可帮助您扩展安全运营。

利用自动化，可以实现流程的一致性和可重复性。虽然每个人都擅长做很多事情，但肯定不能始终如一地重复做同一件事而不出错。即使编写了良好的运行手册，您也会面临人员无法始终如一地执行重复任务的风险。当人员承担多种责任并且必须对不熟悉的提醒做出响应时尤为如此。不过，自动化每次都会以相同的方式响应。通过自动化部署应用程序是最佳选择。可以先测试运行部署的代码，然后将该代码用于执行部署。这增加了变更过程中的信心，同时降低了变更失败的风险。

要验证配置是否达到您的控制目标，请首先在非生产环境中测试自动化和部署的应用程序。这样一来，您就可以测试自动化，证明它正确地执行了所有步骤，还可以在开发和部署周期中获得早期反馈，从而

减少返工。要降低出现部署错误的几率，请通过代码而不是人员来进行配置更改。如果您需要重新部署应用程序，可以利用自动化更轻松地完成此操作。在定义其他控制目标时，您可以轻松地将它们添加到所有工作负载的自动化中。

让各个工作负载负责人使用常见功能和共享组件来节省时间，而不是将精力放在实现针对工作负载的安全性上。多个团队可使用的服务的一些示例包括 AWS 账户创建过程、人员的集中化身份、通用日志记录配置以及 AMI 和容器基础映像创建。此方法可以帮助构建者缩短工作负载周期并始终如一地达到安全控制目标。当团队的一致性更高时，您可以验证控制目标，并向利益相关方更好地报告您的控制态势和风险状况。

最佳实践

- SEC01-BP03 识别并验证控制目标
- SEC01-BP04 随时了解安全威胁和建议
- SEC01-BP05 缩小安全管理范围
- SEC01-BP06 自动部署标准安全控制措施
- SEC01-BP07 使用威胁模型识别威胁并确定缓解措施的优先级
- SEC01-BP08 定期评估并实施新的安全服务和功能

SEC01-BP03 识别并验证控制目标

根据合规性要求以及从威胁模型中发现的风险，获得并验证需要应用于工作负载的控制目标和控制措施。持续验证控制目标和控制措施可帮助您衡量风险缓解措施的有效性。

期望结果：针对业务明确定义了安全控制目标，并且这些目标符合合规性要求。通过自动化方法以及策略来实施和强制执行控制措施，并持续评估这些措施在达成目标方面的有效性。收集某个时间点以及一段时间内的有效性证据，并能够随时报告给审核人员。

常见反模式：

- 没有充分了解用于确保业务安全性的监管要求、市场期望和行业标准
- 网络安全框架和控制目标与业务要求不一致
- 虽然实施了控制措施，但没有与控制目标保持高度一致，也难于衡量
- 不使用自动化方法来报告控制措施的有效性

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

根据安全控制目标，您可以利用许多常见的网络安全框架来奠定基础。根据业务考虑监管要求、市场期望和行业标准，确定哪些框架能够很好地满足需求。这些框架的例子包括 AICPA SOC 2、HITRUST、PCI-DSS、ISO 27001 和 NIST SP 800-53。

对于所确定的控制目标，了解所使用的 AWS 服务如何帮助实现这些目标。使用 AWS Artifact 来查找与目标框架相符的文档和报告，这些文档和报告介绍了 AWS 承担的责任范围，并且提供了针对您负责的其余责任范围的相关指导。如需进一步了解与各种框架控制声明对应的服务特定指导，请参阅《AWS Customer Compliance Guides》。

在定义用于实现目标的控制措施时，请使用预防性控制措施来规范执行方法，并使用检测性控制措施来自动执行缓解方法。在您的 AWS Organizations 中，使用服务控制策略（SCP）来协助防范不合规的资源配置和操作。在 AWS Config 中实施规则，用于监控并报告不合规的资源，然后在确信规则的行为正确无误时，将规则切换为强制执行模式。要部署与网络安全框架相一致的预定义托管规则集，请优先评估使用 AWS Security Hub CSPM 标准。AWS 基础服务最佳实践（FSBP，Foundational Service Best Practice）标准和 CIS AWS 基础基准可作为很好的起点，用于制定与多种标准框架所共有的多个目标相一致的控制措施。如果 Security Hub CSPM 实质上没有所需的控制检测措施，则可以使用 AWS Config 合规包予以补充。

使用 AWS Global Security and Compliance Acceleration（GSCA）团队推荐的 APN 合作伙伴服务包，可以根据需要，从安全顾问、咨询机构、证据收集和报告系统、审核人员以及其他补充性服务那里获取协助。

实施步骤

1. 评估常见的网络安全框架，让控制目标与所选框架保持一致。
2. 使用 AWS Artifact 获取所选框架的相关指导和责任文档。了解在责任共担模式下，合规性的责任有哪些归属于 AWS，哪些由您承担。
3. 使用 SCP、资源策略、角色信任策略和其他防护机制，防止出现不合规的资源配置和操作。
4. 评估与您的控制目标相一致的 Security Hub CSPM 标准和 AWS Config 合规包的部署。

资源

相关最佳实践：

- SEC03-BP01 定义访问要求
- SEC04-BP01 配置服务和应用程序日志记录

- SEC07-BP01 了解数据分类方案
- OPS01-BP03 评估治理要求
- OPS01-BP04 评估合规性要求
- PERF01-BP05 使用策略和参考架构
- COST02-BP01 根据组织的要求制定各种策略

相关文档：

- AWS Customer Compliance Guides

相关工具：

- AWS Artifact

SEC01-BP04 随时了解安全威胁和建议

关注行业威胁情报出版物和数据源来获取行业动态，及时了解最新的威胁和缓解措施。评估根据最新威胁数据自动进行更新的托管服务产品。

期望结果：在行业出版物发布最新的威胁和建议更新时及时了解情况。 您可以使用自动化功能来检测潜在的漏洞和暴露情况，以及识别新的威胁。您对这些威胁采取了缓解措施。 您采用 AWS 服务来自动更新最新的威胁情报。

常见反模式：

- 没有可靠且可重复的机制来随时了解最新的威胁情报。
- 手动维护技术产品组合、工作负载和依赖项清单，这些清单需要人工审查来发现潜在的漏洞和暴露情况。
- 没有采取机制来更新工作负载和依赖项，未获得可提供已知威胁缓解措施的最新版本。

建立此最佳实践的好处：使用威胁情报来源来了解最新信息，可以降低错过可能影响业务的重要威胁形势变化的风险。 与手动替代方案相比，采取自动化功能来扫描、检测和修复工作负载及其依赖项中存在潜在的漏洞或暴露情况，有助于您快速且可预测地降低风险。 这样就可以控制与漏洞缓解相关的时间和成本。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

阅读可信的威胁情报出版物，随时掌握威胁形势。 有关已知的对抗策略、技巧和程序（TTP，Tactics, Techniques, and Procedures）的文档，请参阅 MITRE ATT&CK 知识库。查看 MITRE 的通用漏洞披露（CVE，Common Vulnerabilities and Exposures）列表，随时了解您依赖的产品中的已知漏洞。通过开放式全球应用程序安全项目（OWASP，Open Worldwide Application Security Project）的热门 OWASP Top 10 项目，了解 Web 应用程序面临的严重风险。

通过 CVE 的 AWS 安全公告，及时了解 AWS 安全事件和建议的修复措施。

为了减少保持最新状态所需的总体工作量和开销，您可以考虑使用 AWS 服务，这样就能随着时间的推移自动整合新威胁情报。 例如，Amazon GuardDuty 会及时了解行业威胁情报，从而检测账户中的异常行为和威胁特征。 Amazon Inspector 自动让其用于持续扫描功能的 CVE 数据库保持最新状态。 AWS WAF 和 AWS Shield Advanced 均提供了托管规则组，这些规则组会在新威胁出现时自动更新。

查看用于自动化实例集管理和修补的 Well-Architected 卓越运营支柱。

实施步骤

- 订阅与业务和行业相关的威胁情报出版物，了解最新动态。订阅 AWS 安全公告。
- 考虑采用自动整合新威胁情报的服务，例如 Amazon GuardDuty 和 Amazon Inspector。
- 部署符合 Well-Architected 卓越运营支柱最佳实践的实例集管理和修补策略。

资源

相关最佳实践：

- SEC01-BP07 使用威胁模型识别威胁并确定缓解措施的优先级
- OPS01-BP05 评估威胁形势
- OPS11-BP01 设置持续改进流程

SEC01-BP05 缩小安全管理范围

确定是否可以使用 AWS 服务，将某些控制措施的管理工作转移给 AWS（托管服务），从而缩小安全管理范围。这些服务有助于减少安全维护任务，例如基础设施预置、软件设置、修补或备份。

期望结果：在为工作负载选择 AWS 服务时考虑到安全管理工作的范围。在应该考虑的其他 Well-Architected 注意事项之外，将管理开销和维护任务的成本（总拥有成本，简称 TCO）与您所选择服务的成本进行权衡。在控制措施评估和验证流程中，可以结合考虑 AWS 的控制和合规性文档。

常见反模式：

- 在部署工作负载时，未充分了解所选服务的责任共担模式。
- 在虚拟机上托管数据库和其他技术，但没有评估具备相同功能的托管服务。
- 在与托管服务方案对比时，虚拟机上托管技术的总拥有成本中没有包括安全管理任务。

建立此最佳实践的好处：使用托管服务可以减轻管理运营安全控制措施的整体负担，从而降低您的安全风险和总拥有成本。原本会用在某些安全任务上的时间，可以重新投入到能够为业务创造更多价值的任务上。托管服务还可以将一些控制要求转移给 AWS，从而减少您为满足合规性要求的工作范围。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

您可以通过多种方式将工作负载的组件集成到 AWS 上。如果您在 Amazon EC2 实例上安装和运行各种技术服务，那么在总体安全责任中，通常需要承担更大的份额。为了减轻运营某些控制措施的负担，请找出可以减少您在责任共担模式中所承担责任范围的 AWS 托管服务，并了解如何在现有架构中使用这些服务。例如，使用 Amazon Relational Database Service（Amazon RDS）部署数据库，使用 Amazon Elastic Kubernetes Service（Amazon EKS）或 Amazon Elastic Container Service（Amazon ECS）编排容器，或者使用无服务器方案。在构建新应用程序时，请仔细考虑哪些服务有助于减少实施和管理安全控制措施的时间及成本。

在选择服务时，合规性要求也可能是需要考虑的因素之一。托管服务可以将一些合规性要求转移给 AWS。请与合规团队讨论，了解他们对审核您所运营和管理的服务各个方面的满意程度，以及接受相关 AWS 审核报告中控制声明的满意程度。您可以将AWS Artifact 中的审核构件提供给审核人员或监管机构，作为 AWS 安全控制措施的证据。您还可以使用一些 AWS 审核构件提供的责任指导，并结合 AWS Customer Compliance Guides 来设计架构。这些指导可以让您了解到，为了支持系统的具体应用场景，您还应落实的其他安全控制措施。

使用托管服务时，您需要熟悉将这些服务的资源更新到新版本的过程（例如，更新 Amazon RDS 管理的数据库版本，或者更新 AWS Lambda 函数的编程语言运行时）。尽管托管服务可能会为您执行此操作，但配置更新时间以及了解这些更新会对您的运营产生何种影响，仍然是您的责任。您可以利用 AWS Health 等工具在整个环境中跟踪和管理这些更新。

实施步骤

1. 评估工作负载中可以用托管服务取代的组件。
    a. 如果您将工作负载迁移到 AWS，则在评测是要重新托管、重构、更换平台、重新构建还是更换工作负载时，请考虑减少的管理工作（时间和开支）和降低的风险。从长远来看，在迁移开始时进行额外的投入，有时可以节省大量资金。
2. 请考虑实施 Amazon RDS 等托管服务，而不是安装和管理自己的技术部署。
3. 使用 AWS Artifact 中的责任指导来帮助确定应针对工作负载采取的安全控制措施。
4. 记录所使用资源的清单，及时了解新的服务和方法，以便发现减少责任范围的新机会。

资源

相关最佳实践：

- PERF02-BP01 为工作负载选择最佳计算方案
- PERF03-BP01 使用最能满足数据访问和存储要求的专用数据存储
- SUS05-BP03 使用托管服务

相关文档：

- Planned lifecycle events for AWS Health

相关工具：

- AWS Health
- AWS Artifact
- AWS Customer Compliance Guides

相关视频：

- How do I migrate to an Amazon RDS or Aurora MySQL DB instance using AWS DMS?
- AWS re:Invent 2023 - Manage resource lifecycle events at scale with AWS Health

SEC01-BP06 自动部署标准安全控制措施

在开发和部署 AWS 环境中的标准安全控制措施时，应用现代化 DevOps 实践。 使用基础设施即代码（IaC）模板定义和配置标准安全控制措施，收集版本控制系统中的更改，测试作为 CI/CD 管道一部分的更改，并自动将更改部署到您的 AWS 环境。

期望结果：使用 IaC 模板收集标准化的安全控制措施，并将其提交给版本控制系统。 在检测到变化的地方部署了 CI/CD 管道，并自动测试和部署 AWS 环境。 在继续部署之前，采取了防护机制来检查模板中的错误配置并发出警报。 工作负载部署到采用标准控制措施的环境中。 团队具有访问权限，可以通过自助服务机制部署经批准的服务配置。 制定了安全的备份和恢复策略，用于控制配置、脚本和相关数据。

常见反模式：

- 通过 Web 控制台或命令行界面手动更改标准安全控制措施。
- 依靠各个工作负载团队来手动实施中心团队定义的控制措施。
- 依靠中心安全团队，根据工作负载团队的要求来部署工作负载级别的控制措施。
- 允许相同的个人或团队开发、测试和部署安全控制措施自动化脚本，而没有采取适当的职责分离或制衡措施。

建立此最佳实践的好处：使用模板来定义标准安全控制措施，这样您就可以通过版本控制系统来跟踪和比较随时间发生的变化。 使用自动化功能来测试和部署更改，这样可以实现标准化程序及可预测性，增加成功部署的可能性，减少重复的手动任务。 为工作负载团队提供了自助服务机制来部署经批准的服务和配置，可减少配置错误和滥用的风险。这样还可以让团队在开发过程的早期融入控制措施。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

按照 SEC01-BP01 使用账户分隔工作负载中描述的做法，您最终将多个 AWS 账户用于您通过 AWS Organizations 管理的不同环境。 虽然这些环境和工作负载可能会需要不同的安全控制措施，不过您可以对整个企业内的一些安全控制措施进行标准化。 这样的例子包括集成集中式身份提供程序、定义网络和防火墙，以及配置用于存储和分析日志的标准位置。 就像使用基础设施即代码（IaC）将同样严格的应用程序代码开发要求应用于基础设施预置一样，您也可以使用 IaC 来定义和部署标准安全控制措施。

尽可能以声明式方式（例如在 AWS CloudFormation 中）定义安全控制措施，并将这些安全控制措施存储在源代码控制系统中。 使用 DevOps 实践来自动部署控制措施，从而获得可预测性更强的发布，

使用 AWS CloudFormation Guard 等工具自动进行测试，并检测已部署的控制措施与所需配置之间的偏差。 您可以使用 AWS CodePipeline、AWS CodeBuild 和 AWS CodeDeploy 等服务来构造 CI/CD 管道。请参考使用多个账户组织 AWS 环境中的指导，在每个服务自己的、独立于其他部署管道的账户中，配置这些服务。

您还可以定义模板来实现标准化的 AWS 账户、服务和配置的定义及部署。 利用这种技术，中心安全团队可以管理这些定义，并通过自助服务方法将这些定义提供给工作负载团队。 为此，您可以采取的一种方法是使用 Service Catalog，将模板作为产品发布，供工作负载团队整合到自己的管道部署中。 如果使用的是 AWS Control Tower，则可以使用一些模板和控制措施作为起点。 Control Tower 还提供 Account Factory 功能，让工作负载团队可以使用您定义的标准创建新的 AWS 账户。 在工作负载团队确定了需要新账户时，此功能可以避免需要依赖中心团队来审批和创建新账户。 您可能需要通过这些账户，根据工作负载提供的功能、所处理数据的敏感性或其行为等原因，来隔离不同的工作负载组件。

实施步骤

1. 确定如何在版本控制系统中存储和维护模板。
2. 创建 CI/CD 管道来测试和部署模板。 定义测试方法，用于检查配置是否有误，以及模板是否符合公司标准。
3. 构建标准化模板目录，供工作负载团队根据您的要求部署 AWS 账户和服务。
4. 为控制配置、脚本和相关数据实施安全的备份和恢复策略。

资源

相关最佳实践：

- OPS05-BP01 使用版本控制
- OPS05-BP04 使用构建和部署管理系统
- REL08-BP05 使用自动化功能部署更改
- SUS06-BP01 采用可以快速引入可持续性改进的方法

相关文档：

- 使用多个账户整理您的 AWS 环境

相关示例：

- Automate account creation, and resource provisioning using Service Catalog, AWS Organizations, and AWS Lambda
- Strengthen the DevOps pipeline and protect data with AWS Secrets Manager, AWS KMS, and AWS Certificate Manager

相关工具：

- AWS CloudFormation Guard
- Landing Zone Accelerator on AWS

SEC01-BP07 使用威胁模型识别威胁并确定缓解措施的优先级

执行威胁建模，以识别并维护一个针对工作负载的潜在威胁和相关缓解措施的最新登记表。确定威胁优先级并调整安全控制缓解措施，用于防范、检测和响应。根据工作负载以及不断变化的安全环境，重新审视和维护此登记表。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

什么是威胁建模？

“威胁建模可识别、沟通和理解威胁及缓解措施，用于保护重要资产。” – 开源 Web 应用程序安全项目（OWASP）应用程序威胁建模

为什么应该进行威胁建模？

系统是复杂的，也会随着时间的推移会变得越来越复杂、越来越强大，从而提供更多业务价值，提高客户满意度和参与度。这意味着 IT 设计决策需要考虑数量不断增加的应用场景。由于复杂性和使用案例排列组合的数量过多，非结构化方法将通常无法有效地发现和减轻威胁。因此，您需要采用一种系统方法来列举对系统的潜在威胁，制定缓解措施并确定这些缓解措施的优先级，确保贵组织利用有限资源，在改善系统的整体安全态势方面发挥巨大作用。

威胁建模旨在提供这种系统方法，目的是在设计过程的早期发现和解决问题。此时与生命周期的后期相比，缓解措施的成本和工作量相对较低。这种方法与左移安全实践的行业原则相一致。最终，威胁建模将与组织的风险管理流程相互集成，并通过使用威胁驱动的方法，帮助您决定实施哪些控制措施。

应在什么时候进行威胁建模？

应在工作负载的生命周期中尽早开始进行威胁建模，这使您能够更灵活地处理已识别的威胁。就像软件漏洞一样，越早发现威胁，解决威胁的成本效益就越高。威胁模型是一个动态文档，应随着工作负载的变化而不断发展。随着时间的推移（包括当发生重大变更、威胁形势发生变化或您采用新功能或服务时），重新审视您的威胁模型。

实施步骤

我们如何进行威胁建模？

可以采用许多不同的方式来进行威胁建模。就像编程语言一样，每种方式都有优点和缺点，您应该选择最适合自己的方式。一种方法是从 Shostack 的威胁建模 4 问题框架开始，该框架提出开放式问题，可为威胁建模工作提供结构：

1. 我们正在做什么？

   该问题旨在帮助您了解所构建的系统并达成一致意见，以及了解与安全性相关的系统细节。创建模型或图表是回答该问题的常用方法，因为这有助于对所构建的内容进行可视化，例如使用数据流图。写下关于系统的假设和重要细节也有助于定义范围内的内容。这使每个参与威胁建模的人员都能专注于同一件事，避免因在超出范围的主题（包括过时的系统版本）上走弯路而浪费时间。例如，如果您要构建一个 Web 应用程序，那么可能不值得花时间为浏览器客户端操作系统可信引导顺序进行威胁建模，因为您无法在设计中影响这一点。

2. 会出现什么问题？

   该问题可帮助您识别系统存在的威胁。威胁是指会产生不必要的影响，也可能影响系统安全的意外或故意行为或事件。如果不清楚哪里可能出现问题，您就无从应对。

   对于可能出现的问题，并没有一个规范的列表。创建此列表需要团队中的所有个人和参与威胁建模工作的相关角色集思广益并展开协作。您可以通过使用识别威胁的模型（如 STRIDE）来帮助集思广益，该模型建议了不同的评估类别：欺骗、篡改、抵赖、信息披露、拒绝服务和权限提升。此外，您可能希望通过回顾现有的列表和研究来帮助集思广益，寻找灵感，其中包括 OWASP Top 10、HiTrust 威胁目录和贵组织自己的威胁目录。

3. 我们要怎么做？

   与前面的问题一样，我们不可能得到包含所有缓解措施的规范清单。这一步骤需要考虑的是上一步中确定的威胁、威胁行动者和要改进的领域。

   安全性和合规性是您与 AWS 共同承担的责任。重要的是要明白，当您问“我们要怎么做？”时，您也在问“谁负责做这件事？”。了解您和 AWS 之间的责任平衡有助于将威胁建模工作的范围限定在您控

制的缓解措施范围内，这些缓解措施通常是 AWS 服务配置选项和您自己的系统特定缓解措施的组合。

对于共担责任中 AWS 应承担的部分，您会发现 AWS 服务在许多合规计划的范围内。这些计划可帮助您理解 AWS 用以维持云安全性与合规性的可靠控制机制。AWS 客户可以从 AWS Artifact 下载这些计划的审核报告。

无论您使用哪项 AWS 服务，客户始终要承担一部分责任，并且与这些责任相一致的缓解措施应包含在威胁模型中。对于 AWS 服务自身的安全控制缓解措施，您需要考虑跨域实施安全控制措施，包括身份和访问管理（身份验证和授权）、数据保护（静态和传输中）、基础设施安全性、日志记录和监控等域。每项 AWS 服务的文档都包含一个专门的安全章节，其中提供的安全控制机制指导可用作缓解措施。重要的是，需要考虑您正在编写的代码及其代码依赖项，并思考可以设置哪些控制机制来应对这些威胁。这些控制机制可以是输入验证、会话处理和边界处理等内容。通常，大多数漏洞都是在自定义代码中引入，因此请重点关注这一领域。

4. 我们做得好吗？

   该问题旨在随着时间的推移，让您的团队和组织提高威胁模型的质量并加快执行威胁建模的速度。通过将实践、学习、教学和回顾相结合可以取得这些改进。要想深入了解并亲身体验，建议您和团队完成“适合构建者的威胁建模方式”培训课程或讲习会。此外，如果您正在寻找如何将威胁建模集成到组织的应用程序开发生命周期中的指导，请参阅 AWS 安全博客上的《How to approach threat modeling》一文。

Threat Composer

为了协助并指导执行威胁建模，您可以考虑使用 Threat Composer 工具，缩短进行威胁建模时实现价值的时间。该工具有助于您执行以下操作：

- 撰写符合威胁语法、能够在自然非线性工作流程中发挥作用的有用威胁语句
- 生成人类可读的威胁模型
- 生成机器可读的威胁模型，允许您将威胁模型视为代码
- 使用 Insights 控制面板协助您快速确定质量和覆盖范围有待改进的方面

如需更多参考，请访问 Threat Composer 并切换到系统定义的示例工作区。

资源

相关最佳实践：

- SEC01-BP03 识别并验证控制目标
- SEC01-BP04 随时了解安全威胁和建议
- SEC01-BP05 缩小安全管理范围
- SEC01-BP08 定期评估并实施新的安全服务和功能

相关文档：

- How to approach threat modeling（AWS 安全博客）
- NIST: Guide to Data-Centric System Threat modeling

相关视频：

- AWS Summit ANZ 2021 - How to approach threat modeling
- AWS Summit ANZ 2022 - Scaling security – Optimise for fast and secure delivery

相关培训：

- Threat modeling the right way for builders – AWS Skill Builder virtual self-paced training
- Threat modeling the right way for builders – AWS 讲习会

相关工具：

- Threat Composer

SEC01-BP08 定期评估并实施新的安全服务和功能

评估并实施 AWS 和 AWS 合作伙伴提供的安全服务和功能，以此改善工作负载的安全态势。

期望结果：已经采取标准做法，可获取 AWS 和 AWS 合作伙伴发布的新功能及服务的信息。针对环境和工作负载评估了这些新功能对当前和新控制措施的设计有何影响。

常见反模式：

- 未订阅 AWS 博客和 RSS 源，无法及时了解相关的新功能和服务
- 依赖二手来源来了解有关安全服务和功能的新闻与动态

- 没有鼓励企业中的 AWS 用户随时了解最新动态

建立此最佳实践的好处：如果能够随时了解新的安全服务和功能，就可以针对云环境和工作负载中控制措施的实施，制定出明智的决策。大家可以通过这些来源提高对不断变化的安全形势的认识，以及了解如何使用 AWS 服务来防范新兴的威胁。

在未建立这种最佳实践的情况下暴露的风险等级：低

实施指导

AWS 通过多种渠道向客户告知新的安全服务和功能：

- AWS 的新功能
- AWS 新闻博客
- AWS 安全博客
- AWS 安全公告
- AWS 文档概览

您可以使用 Amazon Simple Notification Service（Amazon SNS）订阅 AWS 每日功能更新主题，获取全面的每日更新摘要。一些安全服务（例如 Amazon GuardDuty 和 AWS Security Hub CSPM）会提供自己的 SNS 主题，方便您随时了解这些特定服务的新标准、调查发现和其他更新。

每年在全球举行的会议、活动和网络研讨会上，同样会公布和详细介绍新的服务及功能。每年的 AWS re:Inforce 安全会议以及内容更全面的 AWS re:Invent 会议尤其需要注意。前面提到的 AWS 新闻频道会分享这些有关安全和其他服务的会议公告，您可以在 YouTube 上的 AWS Events 频道上在线观看深度探讨分组会议，其中提供了丰富的信息。

您也可以向 AWS 账户账户团队询问最新的安全服务更新和建议。如果没有这些团队的直接联系信息，您可以填写销售支持表与该团队取得联系。同样，如果您订阅了 AWS Enterprise Support，技术客户经理（TAM，Technical Account Manager）每周都会向您发布更新内容，而您也可以安排与他们定期开展审查会议。

实施步骤

1. 使用您最喜欢的 RSS 阅读器订阅各种博客和公告，或订阅每日功能更新 SNS 主题。
2. 评估要参加哪些 AWS 活动，获得有关新功能和服务的第一手信息。
3. 如果有任何有关更新安全服务和功能的问题，请安排与您的 AWS 账户团队进行讨论。

4. 请考虑订阅 Enterprise Support，这样就可以定期咨询技术客户经理（TAM）。

资源

相关最佳实践：

- PERF01-BP01 了解并掌握可用的云服务和功能
- COST01-BP07 及时了解新发布的服务

身份和访问管理

要使用 AWS 服务，您必须授予用户和应用程序访问 AWS 账户中资源的权限。在 AWS 上运行更多的工作负载时，您需要实施强大的身份管理和权限，确保适当的人员在适当的条件下有权访问适当的资源。AWS 提供了大量功能，有助于您管理人员和机器身份及其权限。这些功能的最佳实践分为两个主要领域。

主题

身份管理

在操作安全 AWS 工作负载时，您需要管理两类身份。

- 人员身份：需要访问 AWS 环境和应用程序的人员身份可以分类为三个组：员工、第三方和用户。

  员工 组包括作为组织成员的管理员、开发人员和操作员。他们需要访问权限才能管理、构建和运营您的 AWS 资源。

  第三方 是外部协作者，如承包商、供应商或合作伙伴。他们与您的 AWS 资源交互，这是他们与您的组织互动的一部分。

  用户 是应用程序的使用者。他们通过 Web 浏览器、客户端应用程序、移动应用程序或交互式命令行工具访问您的 AWS 资源。
- 机器身份：工作负载应用程序、操作工具和组件需要拥有身份，才能向 AWS 服务发出请求，例如读取数据。这些身份还包括在您的 AWS 环境（例如 Amazon EC2 实例或 AWS Lambda 函数）中运行的机器。您还可以管理需要访问 AWS 环境的外部方或 AWS 外部的机器的机器身份。

最佳实践

- SEC02-BP06 使用用户组和属性

SEC02-BP01 使用强大的登录机制

当不使用多重身份验证（MFA）等机制时，登录（使用登录凭证的身份验证）可能会带来风险，特别是在登录凭证被无意泄露或很容易猜到的情况下。使用强大的登录机制，通过要求使用 MFA 和强密码策略来降低这些风险。

期望结果：通过为 AWS Identity and Access Management（IAM）用户、AWS 账户根用户、AWS IAM Identity Center 和第三方身份提供者使用强大的登录机制，降低意外访问 AWS 中凭证的风险。这意味着需要 MFA，强制执行强密码策略，并检测异常登录行为。

常见反模式：

- 没有为身份执行强密码策略，包括复杂密码和 MFA。
- 在不同的用户之间共享相同的凭证。
- 不对可疑的登录使用检测性控制。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

有几种方法可以让人员身份登录到 AWS。在向 AWS 进行身份验证时，AWS 最佳做法是依赖于使用联合身份验证的集中式身份提供者（AWS IAM 和集中式 IdP 之间的直接 SAML 2.0 联合身份验证，或使用 AWS IAM Identity Center）。在这种情况下，请与身份提供者或 Microsoft Active Directory 建立安全登录过程。

第一次开设 AWS 账户时，您会从 AWS 账户根用户开始。您应仅使用账户根用户为用户（以及为需要根用户的任务）设置访问权限。在开设 AWS 账户后立即为账户根用户开启多重身份验证（MFA），并使用 AWS 最佳实践指南来保护根用户的安全，这一点至关重要。

AWS IAM Identity Center 专为员工用户设计，您可以在服务中创建和管理用户身份，并使用 MFA 保护登录流程。另一方面，AWSCognito 专为客户身份和访问管理（CIAM）而设计，它为应用程序中的外部用户身份提供用户池和身份提供者。

如果您在 AWS IAM Identity Center 中创建用户，请确保该服务中的登录过程安全，并开启 MFA。对于应用程序中的外部用户身份，可以使用 Amazon Cognito 用户池并确保该服务中的登录过程安全，也可以通过 Amazon Cognito 用户池中支持的身份提供者之一进行登录。

此外，对于 AWS IAM Identity Center 中的用户，可以使用 AWS Verified Access，通过在向他们授予访问 AWS 资源的权限之前验证用户的身份和设备状态，来提供一层额外的安全性。

如果您使用的是 AWS Identity and Access Management（IAM）用户，请使用 IAM 来保护登录过程。

可以同时使用 AWS IAM Identity Center 和直接 IAM 联合身份验证来管理对 AWS 的访问权限。可以使用 IAM 联合身份验证来管理对 AWS 管理控制台和服务的访问权限，并使用 IAM Identity Center 来管理对诸如 Quick 或 Amazon Q Business 等业务应用程序的访问权限。

无论采用何种登录方法，执行强登录策略都非常关键。

实施步骤

以下是一般的强登录建议。应根据公司策略或使用 NIST 800-63 等标准，对配置的实际设置进行设定。

- 需要 MFA。对于人员身份和工作负载，要求使用 MFA 是 IAM 最佳实践。启用 MFA 提供了一层额外的安全保障，这会要求用户提供登录凭证和一次性密码（OTP），或从硬件设备加密验证和生成的字符串。
- 强制执行最小密码长度，这是密码强度的主要因素。
- 强制执行密码复杂性，使密码更难以猜到。
- 允许用户更改其密码。
- 创建个人身份而不是共享凭证。通过创建个人身份，您可以为每个用户提供一组唯一的安全凭证。个人用户可以审计每个用户的活动。

IAM Identity Center 建议：

- IAM Identity Center 在使用默认目录时提供了预定义的密码策略，该策略确定了密码长度、复杂性和重用要求。
- 当身份源为默认目录、AWS Managed Microsoft AD 或 AD Connector 时，启用 MFA 并为 MFA 配置“上下文感知”或“始终开启”设置。
- 允许用户注册自己的 MFA 设备。

Amazon Cognito 用户池目录建议：

- 配置密码长度设置。
- 对于用户，要求使用 MFA。
- 使用 Amazon Cognito 用户池高级安全设置可实现自适应身份验证（可阻止可疑登录）等功能。

IAM 用户建议：

- 最好是使用 IAM Identity Center 或直接联合。不过，您可能需要 IAM 用户。在这种情况下，为 IAM 用户设置密码策略。您可以使用密码策略来定义诸如最小长度、密码是否需要非字母字符之类的要求。
- 创建 IAM 策略来强制执行 MFA 登录，允许用户管理自己的密码和 MFA 设备。

资源

相关最佳实践：

- SEC02-BP03 安全地存储和使用密钥
- SEC02-BP04 依赖集中式身份提供程序
- SEC03-BP08 在组织内安全地共享资源

相关文档：

- AWS IAM Identity Center 密钥策略
- IAM 用户密码策略
- 设置 AWS 账户根用户密码
- Amazon Cognito password policy
- AWS 凭证
- IAM 安全最佳实践

相关视频：

- Managing user permissions at scale with AWS IAM Identity Center
- Mastering identity at every layer of the cake

SEC02-BP02 使用临时凭证

进行任何类型的身份验证时，最好使用临时凭证而不是长期凭证，来降低或消除诸如凭证被无意泄露、共享或被盗之类的风险。

期望结果：为了降低长期凭证的风险，尽量对人类身份和机器身份使用临时凭证。长期凭证会带来诸多风险，例如通过上传到公共存储库而泄露。使用临时凭证可以大幅降低凭证被泄露的几率。

常见反模式：

- 开发人员使用 IAM 用户的长期访问密钥，而不是使用联合身份验证从 CLI 获得临时凭证。
- 开发人员在他们的代码中嵌入长期访问密钥，并将该代码上传到公有 Git 存储库。
- 开发人员在移动应用程序中嵌入长期访问密钥，然后在应用商店中提供这些密钥。
- 用户与其他用户共享长期访问密钥，或员工离开公司时仍持有长期访问密钥。
- 当可以使用临时凭证时，对机器身份使用长期访问密钥。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

对所有 AWS API 和 CLI 请求使用临时安全凭证，而不是长期凭证。在几乎所有情况下，对 AWS 服务的 API 和 CLI 请求都必须使用 AWS 访问密钥进行签名。这些请求可以使用临时凭证或长期凭证进行签名。只有在使用 IAM 用户或 AWS 账户根用户时，才应该使用长期凭证（也称为长期访问密钥）。在联合到 AWS 或通过其他方法代入 IAM 角色时，系统会生成临时凭证。即使使用登录凭证访问 AWS 管理控制台，系统也会生成临时凭证供您调用 AWS 服务。需要用到长期凭证的情况很少，所以可以使用临时凭证完成几乎所有任务。

尽量不要使用长期凭证，要多用临时凭证，并且还应少用 IAM 用户进行能访问，而应多用联合身份验证和 IAM 角色进行访问。虽然过去常使用 IAM 用户来访问人类身份和机器身份，但现在不建议使用 IAM 用户，避免使用长期访问密钥所带来的风险。

实施步骤

人员身份

对于员工、管理员、开发人员和操作员等员工身份：

- 您应该依赖集中式身份提供程序，并要求人类用户配合使用联合身份验证与身份提供程序两种方法，以便使用临时凭证访问 AWS。可以通过对每个 AWS 账户进行直接联合身份验证或使用 AWS IAM Identity Center 和您选择的身份提供者，来完成用户的联合身份验证。与使用 IAM 用户相比，联合身份验证除了消除使用长期凭证的情况之外，还具有许多优势。用户也可以从直接联合的命令行或通过使用 IAM Identity Center 来请求获得临时凭证。这意味着能够大幅减少需要使用 IAM 用户或用户长期凭证的情况。

对于第三方身份：

- 在授予软件即服务（SaaS）提供商等第三方访问 AWS 账户中资源的权限时，您可以使用跨账户角色和基于资源的策略。此外，可以为 B2B SaaS 客户或合作伙伴使用 Amazon Cognito OAuth 2.0 grant 客户端凭证流。

通过 Web 浏览器、客户端应用程序、移动应用程序或交互式命令行工具访问 AWS 资源的用户身份：

- 如果需要批准消费者或客户申请访问 AWS 资源，您可以使用 Amazon Cognito 身份池或 Amazon Cognito 用户池提供临时凭证。凭证的权限是通过 IAM 角色配置的。您也可以为未经身份验证的来宾用户定义一个具有有限权限的单独的 IAM 角色。

机器身份

对于机器身份，您就可能需要使用长期凭证了。在这些情况下，您应该要求工作负载使用具有 IAM 角色的临时凭证来访问 AWS。

- 对于Amazon Elastic Compute Cloud（Amazon EC2），您可以使用适用于 Amazon EC2 的角色。
- AWS Lambda 让您能够配置 Lambda 执行角色授予权限，以便利用临时凭证执行 AWS 操作。AWS 服务还有许多其他类似的模型，可以使用 IAM 角色授予临时凭证。
- 对于 IoT 设备，您可以使用 AWS IoT Core 凭证提供程序来请求临时凭证。
- 对于需要访问 AWS 资源的本地系统或在 AWS 之外运行的系统，您可以使用 IAM Roles Anywhere。

某些情况下不支持临时凭证，此时需要使用长期凭证。在这些情况下，可以定期审计和轮换这些凭证和定期轮换访问密钥。对于高度受限的 IAM 用户访问密钥，请考虑以下其它安全措施：

- 授予高度受限的权限：
  - 遵守最低权限原则（特定于操作、资源和条件）。
  - 考虑仅向 IAM 用户授予针对一个特定角色的 AssumeRole 操作。根据本地架构，此方法有助于隔离和保护长期 IAM 凭证。
- 在 IAM 角色信任策略中限制支持的网络来源和 IP 地址。
- 监控使用情况并针对未使用的权限或滥用行为设置警报（使用 AWS CloudWatch Logs 指标筛选条件和警报）。
- 强制执行权限边界 [服务控制策略（SCP）和权限边界相辅相成 - SCP 是粗粒度的，而权限边界是细粒度的]。
- 实施用于预置和（在本地保管库中）安全存储凭证的过程。

对于需要长期凭证的场景，其它一些选项包括：

- 构建自己的令牌出售 API（使用 Amazon API Gateway）。
- 对于必须使用长期凭证或 AWS 访问密钥以外凭证的场景（如数据库登录），可以使用旨在处理密钥管理的服务，如 AWS Secrets Manager。Secrets Manager 可以简化加密密钥的管理、轮换和安全存储。许多 AWS 服务都支持与 Secrets Manager direct integration。
- 对于多云集成，可以使用基于源凭证服务提供商（CSP）凭证的身份联合验证（请参阅 AWS STS AssumeRoleWithWebIdentity）。

有关轮换长期凭证的更多信息，请参阅轮换访问密钥。

资源

相关最佳实践：

- SEC02-BP03 安全地存储和使用密钥
- SEC02-BP04 依赖集中式身份提供程序
- SEC03-BP08 在组织内安全地共享资源

相关文档：

- 临时安全凭证
- AWS 凭证
- IAM 安全最佳实践
- IAM 角色
- IAM Identity Center ()
- 身份提供程序和联合身份验证
- 轮换访问密钥
- 安全合作伙伴解决方案：访问和访问控制
- AWS 账户根用户
- Access AWS using a Google Cloud Platform native workload identity
- How to access AWS resources from Microsoft Entra ID tenants using AWS Security Token Service

相关视频：

- Managing user permissions at scale with AWS IAM Identity Center
- Mastering identity at every layer of the cake

SEC02-BP03 安全地存储和使用密钥

工作负载需要能够自动向数据库、资源和第三方服务证明其身份。这是使用秘密访问凭证（如 API 访问密钥、密码和 OAuth 令牌）完成的。使用专门构建的服务来存储、管理和轮换这些凭证，有助于降低这些凭证泄露的可能性。

期望结果：实施安全管理应用程序凭证的机制来实现以下目标：

- 确定工作负载需要哪些密钥。
- 尽量使用短期凭证代替长期凭证，从而减少所需长期凭证的数量。
- 建立安全存储并自动轮换剩余的长期凭证。
- 审核对工作负载中存在的密钥的访问。
- 持续监控，验证开发期间没有在源代码中嵌入任何密钥。
- 降低凭证被无意中泄露的可能性。

常见反模式：

- 不轮换凭证。
- 将长期凭证存储在源代码或配置文件中。
- 在未加密状态下静态存储凭证。

建立此最佳实践的好处：

- 对存储的凭证进行静态和传输中加密。
- 通过 API 来把关对凭证的访问（可将 API 看作凭证自动售货机）。
- 审核和记录对凭证的访问（包括读和写）。
- 关注点分离：凭证轮换由一个单独的组件执行，该组件可与架构的其余部分隔离开来。
- 密钥自动按需分发给软件组件，并在中心位置进行轮换。
- 可以精细地控制对凭证的访问。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

过去，用于对数据库、第三方 API、令牌和其他密钥进行身份验证的凭证，可能已嵌入到源代码或环境文件中。AWS 提供了几种机制来安全地存储这些凭证，自动轮换凭证，并审核凭证的使用情况。

妥善管理密钥的方法是遵循相关指导，正确地删除、替换和轮换密钥。最安全的凭证是不必存储、管理或处理的凭证。某些凭证可能不再是正常运行工作负载所必需的，可以安全地删除。

对于仍然是正常运行工作负载所必需的凭证，可能有机会用临时或短期凭证替换长期凭证。例如，相较于对 AWS 秘密访问密钥进行硬编码，不妨考虑使用 IAM 角色将长期凭证替换为临时凭证。

可能无法删除或替换某些长期密钥。这些密钥可以存储在 AWS Secrets Manager 等服务中，在其中得到集中存储、管理和定期轮换。

对工作负载的源代码和配置文件进行审计，可以发现许多类型的凭证。下表总结了处理常见凭证类型的策略：

凭证类型	描述	建议采取的策略
IAM 访问密钥	用于在工作负载内代入 IAM 角色的 AWS IAM 访问密钥和私有密钥	替换：改用分配给计算实例（例如 Amazon EC2 或 AWS Lambda）的 IAM 角色。为了与需要访问 AWS 账户中资源的第三方实现互操作性，请询问他们是否支持 AWS 跨账户访问。对于移动应用程序，请考虑通过 Amazon Cognito 身份池（联合身份）使用临时凭证。对于在 AWS 之外运行的工作负载，请考虑使用 IAM Roles Anywhere 或 AWS Systems Manager 混合激活。对于容器，请参阅 Amazon ECS 任务 IAM 角色或 Amazon EKS 节点 IAM 角色。
SSH 密钥	用于登录 Linux EC2 实例的 Secure Shell 私有密钥，可手	替换：使用 AWS Systems Manager 或 EC2 Instance

凭证类型	描述	建议采取的策略
	动登录或作为自动流程的一部分登录	Connect，通过 IAM 角色提供对 EC2 实例的编程访问权限和人类访问。
应用程序和数据库凭证	密码 – 纯文本字符串	轮换：将凭证存储在 AWS Secrets Manager 中，并尽量建立自动轮换机制。
Amazon RDS 和 Aurora 管理数据库凭证	密码 – 纯文本字符串	替换：使用 Secrets Manager 与 Amazon RDS 集成或 Amazon Aurora。此外，在某些应用场景中，一些 RDS 数据库类型可以使用 IAM 角色代替密码（有关更多详细信息，请参阅 IAM 数据库身份验证）。
OAuth 令牌	密钥令牌 – 纯文本字符串	轮换：将令牌存储在 AWS Secrets Manager 中并配置自动轮换。
API 令牌和密钥	密钥令牌 – 纯文本字符串	轮换：存储在 AWS Secrets Manager 中，并尽量建立自动轮换机制。

一种常见的反模式是在源代码、配置文件或移动应用程序中嵌入 IAM 访问密钥。当需要 IAM 访问密钥与 AWS 服务通信时，请使用临时（短期）安全凭证。可以通过 EC2 实例的 IAM 角色、Lambda 函数的执行角色、移动用户访问的 Cognito IAM 角色和 IoT 设备的 IoT Core 策略，提供这些短期凭证。与第三方进行交互时，最好将访问权限委托给 IAM 角色，授予对账户资源的必要访问权限，而不是配置 IAM 用户并向第三方发送该用户的秘密访问密钥。

在许多情况下，工作负载需要存储与其他服务和资源进行互操作所必需的密钥。AWS Secrets Manager 旨在安全地管理这些凭证，以及 API 令牌、密码和其他凭证的存储、使用和轮换。

AWS Secrets Manager 提供五个关键功能，确保敏感凭证的安全存储和处理：静态加密、传输中加密、全面审核、精细访问控制和可扩展凭证轮换。AWS 合作伙伴提供的其他密钥管理服务或提供类似功能和保证的本地开发的解决方案，也可以接受。

在检索密钥时，可以使用 Secrets Manager 客户端缓存组件来缓存密钥，以备将来使用。检索已缓存密钥比从 Secrets Manager 中检索密钥的速度要快。此外，由于调用 Secrets Manager API 会产生费用，因此使用缓存可以降低成本。有关检索密钥的所有方法，请参阅 Get secrets。

> **Note**
>
> 某些语言可能要求您实施自己的内存加密来进行客户端缓存。

实施步骤

1. 使用自动化工具（如 Amazon CodeGuru）识别包含硬编码凭证的代码路径。
   a. 使用 Amazon CodeGuru 扫描代码存储库。审核完成后，在 CodeGuru 中按 Type=Secrets 进行筛选来查找有问题的代码行。
2. 识别可以删除或替换的凭证。
   a. 识别不再需要的凭证并标明要删除。
   b. 对于嵌入到源代码的 AWS 私有密钥，将其替换为与必要资源相关的 IAM 角色。如果部分工作负载在 AWS 之外，但需要 IAM 凭证才能访问 AWS 资源，请考虑采用 IAM Roles Anywhere 或 AWS Systems Manager 混合激活。
3. 对于其他需要使用轮换策略的第三方、长期密钥，请将 Secrets Manager 集成到代码中，以便在运行时检索第三方密钥。
   a. CodeGuru 控制台可以使用发现的凭证在 Secrets Manager 中自动创建密钥。
   b. 将 Secrets Manager 的密钥检索集成到应用程序代码中。
      i. 无服务器 Lambda 函数可以使用与语言无关的 Lambda 扩展。
      ii. 对于 EC2 实例或容器，AWS 用几种流行的编程语言提供了示例客户端代码，用于从 Secrets Manager 检索密钥。
4. 定期检查代码库并重新扫描，验证代码中没有添加新的密钥。
   a. 考虑使用诸如 git-secrets 之类的工具，防止向源代码存储库提交新的密钥。
5. 监控 Secrets Manager 活动，发现意外使用、不适当的密钥访问或试图删除密钥的迹象。
6. 减少人类接触凭证的机会。将读取、写入和修改凭证的权限仅授予专用于此目的的 IAM 角色，并仅向一小部分操作用户提供代入该角色的权限。

资源

相关最佳实践：

- SEC02-BP02 使用临时凭证
- SEC02-BP05 定期审计和轮换凭证

相关文档：

- AWS Secrets Manager 入门
- 身份提供程序和联合身份验证
- Amazon CodeGuru 推出 Secrets Detector
- AWS Secrets Manager 如何使用 AWS Key Management Service
- Secret encryption and decryption in Secrets Manager
- Secrets Manager 博客系列文章
- Amazon RDS 宣布与 AWS Secrets Manager 集成

相关视频：

- Best Practices for Managing, Retrieving, and Rotating Secrets at Scale
- Find Hard-Coded Secrets Using Amazon CodeGuru Secrets Detector
- Securing Secrets for Hybrid Workloads Using AWS Secrets Manager

相关讲习会：

- Store, retrieve, and manage sensitive credentials in AWS Secrets Manager
- AWS Systems Manager Hybrid Activations

SEC02-BP04 依赖集中式身份提供程序

对于工作人员身份（员工和合同工），请依赖允许您在集中位置管理身份的身份提供程序。这样就可以更轻松地跨多个应用程序和系统管理访问权限，因为您可以从单一位置创建、分配、管理、撤销和审核访问权限。

期望结果：您有一个集中式身份提供程序，可以在其中集中管理员工用户、身份验证策略 [例如要求多重身份验证（MFA）]，以及对系统和应用程序的授权（例如根据用户的群组成员资格或属性分配访问权限）。您的员工用户登录到中央身份提供程序并联合身份验证（单点登录）到内部和外部应用程序，这样用户就无需记住多个凭证。您的身份提供程序已与您的人力资源（HR）系统集成，因此人事变动

会自动与身份提供程序同步。例如，如果有人离开组织，您可以自动撤消对联合应用程序和系统（包括 AWS）的访问权限。您已在身份提供程序中启用了详细的审核日志记录，并且正在监控这些日志，以便发现异常用户行为。

常见反模式：

- 不使用联合身份验证和单点登录。员工用户在多个应用程序和系统中创建单独的用户账户和凭证。
- 尚未实现员工用户身份生命周期的自动化，例如将身份提供程序与 HR 系统集成。当用户离职或变换角色时，您使用手动流程来删除或更新他们在多个应用程序和系统中的记录。

建立此最佳实践的好处：通过使用集中式身份提供程序，您可以在一个位置管理员工用户身份和策略，可以向用户和群组分配应用程序的访问权限，还可以监控用户登录活动。通过与您的人力资源（HR）系统集成，当用户的角色发生更改时，这些更改会同步到身份提供程序，并自动更新为他们分配的应用程序和权限。当用户离职时，其身份将在身份提供程序中自动被禁用，从而撤消他们对联合应用程序和系统的访问权限。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

员工用户访问 AWS 的指南：员工用户（如组织中的员工和合同工）可能需要使用 AWS 管理控制台或 AWS Command Line Interface（AWS CLI）来访问 AWS，以便履行工作职能。您可以通过从集中式身份提供程序联合到 AWS，在两个层面上向员工用户授予对 AWS 的访问权限：直接联合到每个 AWS 账户，或联合到 AWS 组织中的多个账户。

要将员工用户与每个 AWS 账户直接联合，可以使用集中式身份提供程序来联合到该账户中的 AWS Identity and Access Management。IAM 的灵活性允许您启用单独的 SAML 2.0 或 Open ID Connect（OIDC）身份提供程序（针对每个 AWS 账户），并使用联合用户属性进行访问控制。员工用户会使用网络浏览器，通过提供凭证（如密码和 MFA 令牌码）来登录身份提供程序。身份提供程序向浏览器发出 SAML 断言，该断言将提交到 AWS 管理控制台登录 URL，以便允许用户通过代入 IAM 角色单点登录 AWS 管理控制台。用户还可以获取临时 AWS API 凭证用于 AWS CLI 或 AWS SDK（从 AWS STS），方法是使用身份提供程序的 SAML 断言代入 IAM 角色。

要对员工用户和 AWS 组织中的多个账户进行联合身份验证，可以使用 AWS IAM Identity Center 来集中管理员工用户对 AWS 账户和应用程序的访问权限。组织启用 Identity Center 并配置身份源。IAM Identity Center 提供一个默认身份源目录，可用来管理用户和组。您也可以选择外部身份源，方法是使用 SAML 2.0 连接外部身份提供程序并使用 SCIM 自动预置用户和组，或使用 Directory Service 连接到 Microsoft AD 目录。配置身份源后，即可通过以下方法为用户和组分配对 AWS 账户的访问权限：在权限集中定义最低权限策略。员工用户可以通过中央身份提供程序进行身份验证，登录 AWS 访问门

户并单点登录到 AWS 账户以及分配给他们的云应用程序。用户可以配置 AWS CLI v2 来使用 Identity Center 进行身份验证，并获取用于运行 AWS CLI 命令的凭证。Identity Center 还支持通过单点登录访问 AWS 应用程序，例如 Amazon SageMaker AI Studio 和 AWS IoT Sitewise Monitor portals。

遵循上述指导后，员工用户在 AWS 上管理工作负载时，将不再需要使用 IAM 用户和组来进行通用的操作。相反，用户和组是在 AWS 外部进行管理，并且能够以联合身份访问 AWS 资源。联合身份使用集中式身份提供程序定义的组。您应该识别并删除 AWS 账户中不再需要的 IAM 组、IAM 用户和长期用户凭证（密码和访问密钥）。您可以查找未使用的凭证（使用 IAM 凭证报告）、删除相应的 IAM 用户和删除 IAM 组。您可以将服务控制策略（SCP）应用于组织，帮助防止创建新的 IAM 用户和组，并强制通过联合身份访问 AWS。

> **Note**
>
> 您负责处理 SCIM 访问令牌的轮换，如自动预置文档中所述。此外，您还负责轮换支持身份联合验证的证书。

应用程序用户的指南：通过将 Amazon Cognito 用作集中式身份提供者，您可以管理应用程序（例如移动应用程序）用户的身份。Amazon Cognito 支持对 Web 和移动应用程序进行身份验证、授权和用户管理。Amazon Cognito 提供可扩展到数百万用户的身份存储，支持社交网络和企业身份联合验证，并提供高级安全功能来协助保护用户和业务。您可以将自定义 Web 或移动应用程序与 Amazon Cognito 集成，以便在几分钟内为应用程序添加用户身份验证和访问控制。Amazon Cognito 以 SAML 和 Open ID Connect（OIDC）等开放式身份标准为基础构建，支持各种合规性法规，并与前端和后端开发资源集成。

实施步骤

员工用户访问 AWS 的步骤

- 通过以下某种方法，使用集中式身份提供程序向 AWS 联合验证员工身份：
  - 通过与身份提供程序联合，使用 IAM Identity Center 来允许单点登录到 AWS 组织中的多个 AWS 账户。
  - 使用 IAM 将身份提供程序直接连接到每个 AWS 账户，从而实现精细的联合访问。
- 识别并移除被联合身份取代的 IAM 用户和群组。

适用于应用程序用户的步骤

- 将 Amazon Cognito 用作应用程序的集中式身份提供程序。

- 使用 OpenID Connect 和 OAuth 将自定义应用程序与 Amazon Cognito 集成。您可以使用 Amplify 库开发自定义应用程序，这些库提供了与各种 AWS 服务（例如用于身份验证的 Amazon Cognito）集成的简单接口。

资源

相关最佳实践：

- SEC02-BP06 使用用户组和属性
- SEC03-BP02 授予最低访问权限
- SEC03-BP06 基于生命周期管理访问权限

相关文档：

- AWS 中的身份联合验证
- IAM 安全最佳实操
- AWS Identity and Access Management 最佳实践
- IAM Identity Center 委派管理入门
- How to use customer managed policies in IAM Identity Center for advanced use cases
- AWS CLI v2: IAM Identity Center credential provider

相关视频：

- AWS re:Inforce 2022 - AWS Identity and Access Management (IAM) deep dive
- AWS re:Invent 2022 - Simplify your existing workforce access with IAM Identity Center
- AWS re:Invent 2018: Mastering Identity at Every Layer of the Cake

相关示例：

- Workshop: Using AWS IAM Identity Center to achieve strong identity management

相关工具：

- AWS 安全能力合作伙伴：身份和访问管理
- saml2aws

SEC02-BP05 定期审计和轮换凭证

定期审计和轮换凭证，以限制凭证可用于访问资源的时间。长期凭证会产生许多风险，可通过定期轮换长期凭证来降低这些风险。

期望结果：实施凭证轮换，以帮助降低长期凭证相关风险。定期审计并纠正不符合凭证轮换策略的情况。

常见反模式：

- 不审计凭证的使用情况。
- 不必要地使用长期凭证。
- 使用长期凭证，不定期轮换。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

当您无法依赖于临时凭证而需要长期凭证时，请审计凭证，以验证定义的控制措施 [例如多重身份验证（MFA）] 得以实施、定期轮换并具有适当的访问级别。

（最好通过自动化工具）定期验证，以确保实施正确的控制措施。对于人员身份，您应要求用户定期更改他们的密码并停用访问密钥，以支持临时凭证。从 AWS Identity and Access Management（IAM）用户转向集中式身份时，您可以生成凭证报告来审计您的用户。

我们还建议您在身份提供者中实施并监控 MFA。您可以设置 AWS Config 规则或使用 AWS Security Hub CSPM 安全标准来监控用户是否配置了 MFA。考虑使用 IAM Roles Anywhere 为机器身份提供临时凭证。在无法使用 IAM 角色和临时凭证的情况下，需要经常审计和轮换访问密钥。

实施步骤

- 定期审计凭证：对您的身份提供者和 IAM 中配置的身份进行审计，这有助于验证只有经过授权的身份才能访问您的工作负载。此类身份可能包括但不限于 IAM 用户、AWS IAM Identity Center 用户、Active Directory 用户或不同上游身份提供者中的用户。例如，删除离开组织的人员，并删除不再需要的跨账户角色。制定流程，以定期审计 IAM 实体所访问服务的权限。这有助于您确定需要修改的策略，以删除任何未使用的权限。使用凭证报告和 AWS Identity and Access Management Access Analyzer 来审计 IAM 凭证和权限。您可以使用 Amazon CloudWatch 为 AWS 环境中调用的特定 API 调用设置警报。Amazon GuardDuty 还可以提醒您注意意外活动，出现这种提醒，可表明对 IAM 凭证的访问过于宽松，或出现了意外访问情况。

- 定期轮换凭证：当您无法使用临时凭证时，请定期轮换长期 IAM 访问密钥（最多每 90 天一次）。如果在您不知情的情况下无意中泄露了访问密钥，这将限制凭证用于访问资源的时间。有关轮换 IAM 用户的访问密钥的信息，请参阅《轮换访问密钥》。
- 查看 IAM 权限：要增强您的 AWS 账户的安全性，请定期查看和监控每个 IAM 策略。验证这些策略是否遵循最低权限原则。
- 考虑自动创建和更新 IAM 资源：IAM Identity Center 自动执行许多 IAM 任务，比如角色和策略管理。或者，AWS CloudFormation 可用于自动部署 IAM 资源（包括角色和策略），以减少人为错误的机会，因为可以验证模板和控制版本。
- 对于机器身份，使用 IAM Roles Anywhere 替换 IAM 用户：IAM Roles Anywhere 将使您能够在传统上无法使用角色的领域（例如本地服务器）使用角色。IAM Roles Anywhere 使用可信的 X.509 certificate 向 AWS 进行身份验证并接收临时凭证。使用 IAM Roles Anywhere 便无需轮换这些凭证，因为长期凭证不再存储在本地环境中。请注意，您需要监控 X.509 证书，并在该证书即将到期时轮换它。

资源

相关最佳实践：

- SEC02-BP02 使用临时凭证
- SEC02-BP03 安全地存储和使用密钥

相关文档：

- AWS Secrets Manager 入门
- IAM 最佳实践
- 身份提供程序和联合身份验证
- 安全合作伙伴解决方案：访问和访问控制
- 临时安全凭证
- 获取您的 AWS 账户的凭证报告

相关视频：

- Best Practices for Managing, Retrieving, and Rotating Secrets at Scale
- Managing user permissions at scale with AWS IAM Identity Center
- Mastering identity at every layer of the cake

SEC02-BP06 使用用户组和属性

根据用户组和属性来定义权限，可以减少策略的数量和复杂性，这样就更容易实现最低权限原则。您可以通过用户组，根据用户在企业中履行的职能，从一个位置管理多个用户的权限。当用户履行类似的职能但面向的是不同资源子集时，可以利用部门、项目或位置等属性来提供额外的一层权限范围。

期望结果：您可以将基于职能的权限更改应用到履行该职能的所有用户。利用组成员资格和属性管控用户的权限，减少在单个用户级别管理权限的需求。您在身份提供者（IdP）中定义的组和属性会自动传播到您的 AWS 环境。

常见反模式：

- 分别管理每个用户的权限，然后在多个用户之间复制。
- 定义过于宽泛的组，授予过于宽泛的权限。
- 定义过于精细的组，造成成员重复和混淆。
- 对不同资源子集使用具有重复权限的组，但其实可以改为使用属性来进行控制。
- 在管理组、属性和成员资格时，没有使用与您的 AWS 环境集成的标准化身份提供者。
- 使用 AWS IAM Identity Center 会话时使用角色链

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

在称为策略 的文档中定义 AWS 权限，这些策略与某个主体关联，例如用户、组、角色或资源。可以根据工作职能、工作负载和 SDLC 环境来组织权限分配（组、权限、账户），从而扩展权限管理。对于员工团队，通过这种方法，您可以根据用户在企业中履行的职能来定义组，而不是根据所访问的资源来定义。例如，WebAppDeveloper 组可能附有策略，用于在开发账户中配置 Amazon CloudFront 等服务。`AutomationDeveloper` 组可能与 `WebAppDeveloper` 组具有一些重叠的权限。可以将这些共有的权限放置在单独的策略中并与这两个组关联，而不是将来自这两个职能部门的用户都放在同一个 `CloudFrontAccess` 组中。

除了使用组之外，您还可以使用属性来进一步限定访问范围。例如，您可以为 `WebAppDeveloper` 组中的用户设置“项目”属性，用来限定对其项目特定资源的访问权限。使用这种技巧，您就无需为处理不同项目的应用程序开发人员设置不同的组，因为他们除了项目不同外，所需的权限其实并无不同。在权限策略中引用属性的方式取决于权限的来源，这些权限可能包含在联合身份验证协议（例如 SAML、OIDC 或 SCIM）中，可能是自定义的 SAML 断言，也可能是在 IAM Identity Center 中设置。

实施步骤

1. 确定要在何处定义组和属性：
   a. 按照 SEC02-BP04 依赖集中式身份提供者中的指南，您可以确定是要在身份提供者中或 IAM Identity Center 中定义组和属性，还是在特定账户中使用 IAM 用户组。
2. 定义组：
   a. 根据职能和所需访问权限范围来确定组。考虑使用分层结构或命名约定来有效地整理组。
   b. 如果在 IAM Identity Center 中定义，则创建组并使用权限集关联所需的访问权限级别。
   c. 如果在外部身份提供者中定义，请确定该提供者是否支持 SCIM 协议，并考虑在 IAM Identity Center 中启用自动预置。此功能可在您的提供者和 IAM Identity Center 之间同步组的创建、成员指派和删除。
3. 定义属性：
   a. 如果使用外部身份提供者，则默认情况下，SCIM 和 SAML 2.0 协议都提供一些属性。使用 `https://aws.amazon.com/SAML/Attributes/PrincipalTag` 属性名称，可以通过 SAML 断言来定义并传递其它属性。有关定义和配置自定义属性的指南，请咨询身份提供者的文档。
   b. 如果在 IAM Identity Center 中定义角色，请启用基于属性的访问权限控制（ABAC）功能，然后根据需要定义属性。考虑符合组织的结构或资源标签策略的属性。

如果您需要从通过 IAM Identity Center 代入的 IAM 角色中获得的 IAM 角色链，则诸如 `source-identity` 和 `principal-tags` 的值将不会传播。有关更多详细信息，请参阅启用并配置访问控制属性。

1. 根据组和属性限定权限范围：
   a. 考虑在权限策略中加入条件，将主体的属性与所访问资源的属性进行比较。例如，您可以定义一个条件，仅当 `PrincipalTag` 条件键的值与相同名称的 `ResourceTag` 键的值匹配时，才支持访问资源。
   b. 定义 ABAC 策略时，请遵循 ABAC 授权最佳实践和示例中的指导。
   c. 随着组织需求的发展，请定期审核和更新组和属性结构，以确保最佳的权限管理。

资源

相关最佳实践：

- SEC02-BP04 依赖集中式身份提供程序

- SEC03-BP02 授予最低访问权限
- COST02-BP04 实施组和角色

相关文档：

- IAM 最佳实践
- 在 IAM Identity Center 中管理身份
- 什么是适用于 AWS 的 ABAC？
- 在 IAM Identity Center 中使用 ABAC
- ABAC 策略示例

相关视频：

- Managing user permissions at scale with AWS IAM Identity Center
- Mastering identity at every layer of the cake

权限管理

管理权限，控制需要访问 AWS 和您工作负载的人员和机器身份的访问权限。权限可让您控制哪些人可以在什么条件下访问哪些内容。通过为特定的人员身份和机器身份设置权限，可以向这些身份授予对特定资源执行特定服务操作的访问权限。此外，您可以为要授予的访问权限指定必须满足的条件。

可通过多种方式向不同类型的资源授予访问权限。一种方式是使用不同的策略类型。

IAM 中基于身份的策略属于托管或内联策略，会附加到 IAM 身份（用户、组或角色）。这些策略可让您指定该身份可执行哪些操作（其权限）。基于身份的策略可以进一步分类。

托管策略 - 基于身份的独立策略，可附加到您的 AWS 账户中的多个用户、组和角色。有两种托管策略：

- AWS 托管策略 – 由 AWS 创建和管理的托管策略。
- 客户管理型策略 – 您在 AWS 账户中创建和管理的托管策略。与 AWS 托管策略相比，客户管理型策略对策略的控制更精确。

托管策略是应用权限的首选方法。不过，您也可以使用直接添加到单个用户、组或角色的内联策略。内联策略维持策略与身份之间严格的一对一关系。删除身份即会删除内联策略。

大多数情况下，您应按照最低权限原则创建自己的客户管理型策略。

基于资源的策略附加到某个资源。例如，S3 存储桶策略是一个基于资源的策略。这些策略向一个主体授予权限，该主体既可以位于资源所在的账户中，也可以位于另一个账户中。有关支持基于资源的策略的服务列表，请参阅使用 IAM 的 AWS 服务。

权限边界使用托管策略来设置管理员能够设定的最高权限。这样，您就可以为开发人员赋予创建和管理权限的能力，例如创建一个 IAM 角色，但限制他们可以授予的权限，让他们无法利用他们创建的角色提升自己的权限。

AWS 中的基于属性的访问权限控制（ABAC）让您能够根据属性（成为标签）来授予权限。这些标签可以附加到 IAM 主体（用户或角色）和 AWS 资源。管理员可以创建可重复使用的 IAM 策略，这些策略根据 IAM 主体的属性来应用权限。例如，作为管理员，您可以使用单个 IAM 策略，向您所在组织中的开发人员授予对与其项目标签匹配的 AWS 资源的访问权限。当开发人员团队为项目添加资源时，会自动根据属性应用权限，从而消除了对每个新资源进行策略更新的需要。

Organizations 服务控制策略（SCP）为组织或组织部门（OU）的账户成员定义最大权限。SCP 限制基于身份的策略或基于资源的策略授予账户中实体（用户或角色）的权限，但不授予权限。

会话策略会担任角色或联合用户。在使用 AWS CLI 或 AWS API 会话策略限制基于角色或用户身份的策略授予会话的权限时，传递会话策略。这些策略限制所创建会话的权限，但不授予权限。有关更多信息，请参阅会话策略。

最佳实践

- SEC03-BP01 定义访问要求
- SEC03-BP02 授予最低访问权限
- SEC03-BP03 建立紧急访问流程
- SEC03-BP04 持续减少权限
- SEC03-BP05 为您的组织定义权限防护机制
- SEC03-BP06 基于生命周期管理访问权限
- SEC03-BP07 分析公共和跨账户访问
- SEC03-BP08 在组织内安全地共享资源
- SEC03-BP09 与第三方安全地共享资源

SEC03-BP01 定义访问要求

管理员、最终用户或其他组件都需要访问工作负载的每个组件或资源。明确定义什么人或什么内容应该有权访问每个组件，选择适当的身份类型以及身份验证和授权方法。

常见反模式：

- 在应用程序中进行硬编码或存储密码。
- 向每个用户授予自定义权限。
- 使用长期有效的凭证。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

管理员、最终用户或其他组件都需要访问工作负载的每个组件或资源。明确定义什么人或什么内容应该有权访问每个组件，选择适当的身份类型以及身份验证和授权方法。

应提供对组织内 AWS 账户的常规访问，方法是使用联合身份访问或集中式身份提供者。您还应将身份管理集中处理，确保对于 AWS 将访问集成到员工访问生命周期中已建立了既定做法。例如，当员工转岗到具有不同访问级别的职位时，该员工的小组成员资格也应进行更改以反映新的访问要求。

在定义非人类身份的访问要求时，请确定哪些应用程序和组件需要访问权限以及如何向其授予权限。建议使用通过最低权限访问模型构建的 IAM 角色。AWS托管式策略提供了预定义的 IAM 策略，这些策略涵盖了大多数常见使用案例。

AWS 服务（例如 AWS Secrets Manager 和 AWS Systems Manager Parameter Store）有助于安全地将密钥与应用程序或工作负载分离。在 Secrets Manager 中，您可以为凭证建立自动轮换。您可以通过使用您在创建参数时指定的唯一名称，使用 Systems Manager 来引用脚本、命令、SSM 文档、配置和自动化工作流中的参数。

您可以使用 AWS IAM Roles Anywhere 来获取 IAM 中的临时安全凭证，这种凭证适用于在 AWS 外部运行的工作负载。您的工作负载可以使用与 AWS 应用程序所用的相同 IAM 策略和 IAM 角色来访问 AWS 资源。

如果可能，请优先选择短期临时凭证而不是长期静态凭证。在一些场景中，需要具有编程访问权限和长期凭证的用户，此时请使用访问密钥上次使用的信息来轮换和删除访问密钥。

如果用户需要在 AWS 管理控制台之外与 AWS 交互，则需要编程式访问权限。授予编程式访问权限的方法取决于访问 AWS 的用户类型。

要向用户授予编程式访问权限，请选择以下选项之一。

哪个用户需要编程式访问权限？	目的	方式
IAM	（建议）使用控制台凭证作为临时凭证来签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。	按照您希望使用的界面的说明进行操作。 • 对于 AWS CLI，请参阅《AWS Command Line Interface用户指南》中的登录以进行 AWS 本地开发。 • 对于 AWS SDK，请参阅《AWS SDK 和工具参考指南》中的登录以进行 AWS 本地开发。
人力身份 （在 IAM Identity Center 中管理的用户）	使用临时凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。	按照您希望使用的界面的说明进行操作。 • 有关 AWS CLI 的更多信息，请参阅《AWS Command Line Interface 用户指南》中的配置 AWS CLI 以使用 AWS IAM Identity Center。 • 有关 AWS SDK、工具和 AWS API 的更多信息，请参阅《AWS SDK 和工具参考指南》中的 IAM Identity Center 身份验证。
IAM	使用临时凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。	按照《IAM 用户指南》中将临时凭证用于 AWS 资源中的说明进行操作。
IAM	（不推荐使用）	按照您希望使用的界面的说明进行操作。

哪个用户需要编程式访问权限？	目的	方式
	使用长期凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。	• 有关 AWS CLI 的更多信息，请参阅《AWS Command Line Interface 用户指南》中的使用 IAM 用户凭证进行身份验证。 • 有关 AWS SDK 和工具的更多信息，请参阅《AWS SDK 和工具参考指南》中的使用长期凭证进行身份验证。 • 有关 AWS API 的更多信息，请参阅《IAM 用户指南》中的管理 IAM 用户的访问密钥。

资源

相关文档：

- 基于属性的访问控制（ABAC）
- AWS IAM Identity Center
- IAM Roles Anywhere
- IAM Identity Center 的 AWS 托管式策略
- AWS IAM 策略条件
- IAM 使用案例
- 移除不必要的凭证
- 使用 策略
- 如何根据 AWS 账户、OU 或组织来控制对 AWS 资源的访问
- 使用 AWS Secrets Manager 中的增强搜索来轻松标识、安排和管理密钥

相关视频：

- 在最多 60 分钟的时间内成为 IAM 策略高手

- 职责分离、最低权限、委托和 CI/CD
- 简化身份和访问管理以实施创新

SEC03-BP02 授予最低访问权限

仅授予用户在特定条件下对特定资源执行特定操作所需的访问权限。使用组和身份属性来大规模动态设置权限，而不是为单个用户定义权限。例如，您可以允许一组开发人员访问，以便仅管理其项目的资源。使用这种方法，如果某个开发人员离开项目，则可以自动撤销该开发人员的访问权限，而无需更改底层访问策略。

期望结果：用户仅拥有其特定工作职能所需的最低权限。可以使用单独的 AWS 账户来将开发人员与生产环境隔离开来。当开发人员需要访问生产环境以执行特定任务时，他们仅在这些任务期间被授予有限和受控的访问权限。在他们完成必要的工作后，他们的生产访问权限会被立即撤销。您可以定期审核权限，并在不再需要时立即撤销权限，例如当用户变更角色或离开组织时。您可以将管理员权限限制在一个小型、受信任的组中，以降低暴露的风险。您仅向计算机或系统账户授予执行其预期任务所需的最低权限。

常见反模式：

- 默认情况下，您向用户授予管理员权限。
- 您使用根用户账户进行日常活动。
- 您创建过于宽松的策略，而没有限定适当的范围。
- 您的权限审核不频繁，这会导致权限蔓延。
- 您完全依赖基于属性的访问权限控制来实现环境隔离或权限管理。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

最低权限原则规定，只应允许身份执行完成特定任务所需的最小一组操作。这样可使可用性、效率和安全性达到平衡。根据此原则运营，有助于限制意外访问，还有助于跟踪谁有权访问哪些资源。默认情况下，IAM 用户和角色没有任何权限。默认情况下，根用户具有完全访问权限，应受到严格控制和监控，并且仅用于需要根访问权限的任务。

IAM 策略用于显式地为 IAM 角色或特定资源授予权限。例如，基于身份的策略可以附加到 IAM 组，而 S3 存储桶可由基于资源的策略控制。

创建 IAM 策略时，可以指定服务操作、资源以及为使 AWS 允许或拒绝访问而必须满足的条件。AWS 支持多种条件以协助您缩小访问权限范围。例如，通过使用 PrincipalOrgID 条件键，如果请求者不属于您的 AWS 组织，则您可以拒绝操作。

您还可以控制 AWS 服务代表您发出的请求，例如要求 AWS CloudFormation 创建一个 AWS Lambda 函数，方法是使用 CalledVia 条件键。您可以对不同的策略类型进行分层，以建立纵深防御并限制用户的总体权限。您还可以限制可以授予哪些权限以及在什么条件下授予权限。例如，可以让工作负载团队为他们所构建的系统创建自己的 IAM 策略，但前提是他们应用权限边界来限制他们可以授予的最大权限。

实施步骤

- 实施最低权限策略：向 IAM 组和角色分配具有最低权限的访问策略，以反映您所定义的用户的角色或职能。
- 通过单独的 AWS 账户隔离开发和生产环境：为开发和生产环境使用单独的 AWS 账户，并使用服务控制策略、资源策略和身份策略来控制它们之间的访问权限。
- 基于 API 使用情况制定策略：确定所需权限的一种方法是查看 AWS CloudTrail 日志。您可以使用此审核，根据用户在 AWS 内实际执行的操作来创建权限。IAM Access Analyzer 可以基于访问活动自动生成 IAM 策略。您可以在组织或账户级别，使用 IAM Access Advisor 来跟踪上次访问的关于某个特定策略的信息。
- 考虑使用工作职能的 AWS 托管式策略：当您开始创建细粒度的权限策略时，将 AWS 托管式策略用于常见的工作角色（如计费、数据库管理员和数据科学家）可能会有所帮助。这些策略有助于缩减用户具备的访问权限，同时您可以确定如何实施最低权限策略。
- 移除不必要的权限：检测并移除未使用的 IAM 实体、凭证和权限，以实现最低权限原则。可以使用 IAM Access Analyzer 来识别外部和未使用的访问权限，而 IAM Access Analyzer 策略生成有助于微调权限策略。
- 确保用户对生产环境的访问权限有限：用户应只能访问具有有效使用案例的生产环境。在用户执行需要生产访问权限的特定任务后，应撤销访问权限。限制对生产环境的访问可帮助防止发生影响生产的意外事件，并缩小意外访问的影响范围。
- 考虑权限边界：权限边界功能使用托管式策略，来设置基于身份的策略可向 IAM 实体授予的最大权限。实体的权限边界仅允许实体执行其基于身份的策略和权限边界同时允许的操作。
- 使用基于属性的访问权限控制和资源标签优化访问权限：使用资源标签的基于属性的访问权限控制（ABAC）可用于优化权限（如果支持）。可以使用将主体标签与资源标签进行比较的 ABAC 模型，根据您定义的自定义维度来优化访问权限。这种方法可以简化组织中的权限策略并减少其数量。

  - 建议仅当主体和资源均归您的 AWS 组织拥有时，才使用 ABAC 进行访问权限控制。外部各方可以为自己的主体和资源使用与您的组织相同的标签名称和值。如果您仅依靠这些名称/值对来授予对外部主体或资源的访问权限，则可能会提供意想不到的权限。
- 对 AWS Organizations 使用服务控制策略：Service control policies 集中控制您组织中成员账户的最大可用权限。重要的是，您可以使用服务控制策略来限制成员账户中的根用户权限。还要考虑使用 AWS Control Tower，它提供可充实 AWS Organizations 的规范性托管控制。您还可以在 Control Tower 内定义自己的控制。
- 为您的组织制定用户生命周期策略：用户生命周期策略定义了当用户加入 AWS、更改作业角色或范围或不再需要访问 AWS 时要执行的任务。在用户生命周期的每个步骤中执行权限审核，来验证权限受到适当限制并避免权限蔓延。
- 制定定期计划来审核权限并移除任何不需要的权限：您应该定期审核用户访问权限，以验证用户没有过于宽松的访问权限。AWS Config 和 IAM Access Analyzer 可以在审计用户权限时提供帮助。
- 建立工作角色矩阵：工作角色矩阵可直观显示您的 AWS 业务覆盖区域中所需的各种角色和访问级别。使用工作角色矩阵，您可以根据用户在组织内的职责来定义和分离权限。使用组，而不是将权限直接应用于各个用户或角色。

资源

相关文档：

- 授予最低权限
- IAM 实体的权限边界
- 用于编写最低权限 IAM 策略的方法
- IAM Access Analyzer makes it easier to implement least privilege permissions by generating IAM policies based on access activity
- 使用 IAM 权限边界将权限管理委派给开发人员
- 使用上次访问的信息优化权限
- IAM 策略类型及其使用时间
- 使用 IAM policy simulator 测试 IAM policy
- Guardrails in AWS Control Tower
- 零信任架构：AWS 视角
- 如何使用 CloudFormation StackSets 实施最低权限原则
- 基于属性的访问控制（ABAC）

- 查看用户活动以缩小策略范围
- 查看角色访问
- 使用标记来组织环境并推动问责制
- AWS 标记策略
- 为AWS资源添加标签

相关视频：

- 新一代权限管理
- 零信任：AWS 视角

SEC03-BP03 建立紧急访问流程

创建一个流程，便于在集中式身份提供程序偶尔出现问题时紧急访问您的工作负载。

必须针对可能导致紧急事件的不同故障模式设计流程。例如，在正常情况下，您的员工用户使用集中式身份提供程序联合到云端（SEC02-BP04）来管理其工作负载。但是，如果您的集中式身份提供程序出现故障，或者云中联合身份验证的配置被修改，则您的员工用户可能无法联合到云中。紧急访问流程允许授权管理员通过其他方式（例如其他联合形式或直接用户访问）访问云资源，以解决联合配置或工作负载的问题。在恢复正常的联合机制之前，将使用紧急访问流程。

期望结果：

- 您已经定义并记录了算是紧急情况的故障模式：考虑您的正常情况以及用户管理其工作负载所依赖的系统。考虑这些依赖项中的每一个在哪些情形下会发生故障并导致紧急情况。您可能会发现可靠性支柱中的问题和最佳实践有助于识别故障模式和构建更具韧性的系统，从而最大限度地降低发生故障的可能性。
- 您已记录了将故障确认为紧急情况所必须遵循的步骤。例如，您可以要求身份管理员检查主身份提供程序和备用身份提供程序的状态，如果两者均不可用，则宣布身份提供程序故障为紧急事件。
- 您已针对每种紧急情况或故障模式定义了紧急访问流程。应尽可能明确具体，这样可减少用户针对所有类型的紧急情况过度使用通用流程的倾向。紧急访问流程描述了每个流程的使用情形，以及哪些情况下不应使用该流程，并指出了可能适用的替代流程。
- 您的流程有详细的说明和行动手册，便于快速有效地遵循。请记住，对用户来说，紧急事件可能让人很煎熬，他们可能面临极大的时间压力，因此流程设计应尽可能简单。

常见反模式：

- 您没有详细记录并经过充分测试的紧急访问流程。您的用户没有为紧急情况做好准备，在出现紧急事件时遵循临时流程。
- 您的紧急访问流程依赖于与普通访问机制相同的系统（例如集中式身份提供程序）。这意味着，此类系统的故障可能会同时影响您的正常访问和紧急访问机制，并削弱您从故障中恢复的能力。
- 您的紧急访问流程被用于非紧急情况。例如，您的用户经常滥用紧急访问流程，因为他们发现直接进行更改比通过管道提交更改更容易。
- 您的紧急访问流程未生成足够的日志用于审核这些流程，或者没有监控日志以提醒可能存在的流程滥用。

建立此最佳实践的好处：

- 通过拥有记录详实且经过充分测试的紧急访问流程，您可以减少用户响应和解决紧急事件所花费的时间。这样可以缩短停机时间，提高您向客户提供的服务的可用性。
- 您可以跟踪每个紧急访问请求，检测未经授权企图对非紧急事件滥用该过程的行为，并发出警报。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

本节针对与部署在 AWS 上的工作负载相关的几种故障模式，提供创建紧急访问流程的指导，首先是适用于所有故障模式的通用指导，然后是不同故障模式类型的特定指导。

适用于所有故障模式的通用指南

在针对故障模式设计紧急访问流程时，请考虑以下几点：

- 记录流程的先决条件和假设：何时应使用该流程、何时不应使用该流程。它有助于详细说明故障模式并记录假设，例如其他相关系统的状态。例如，故障模式 2 的流程假设身份提供程序可用，但 AWS 上的配置已修改或已过期。
- 预先创建紧急访问流程所需的资源（SEC10-BP05）。例如，预先创建带有 IAM 用户和角色的紧急访问 AWS 账户，并在所有工作负载账户中创建跨账户 IAM 角色。这样可以验证在发生紧急事件时这些资源是否已准备就绪并且可用。通过预先创建资源，您就不必依赖 AWS 控制面板 API（用于创建和修改 AWS 资源），在紧急情况下，这些 API 可能不可用。此外，通过预先创建 IAM 资源，您也无需考虑由于最终的一致性问题而可能出现的延迟。
- 将紧急访问流程纳入事件管理计划（SEC10-BP02）。记录如何跟踪紧急事件并将其传达给组织中的其他人，例如同级团队、您的领导层，如果适用，还包括外部的客户和业务合作伙伴。

- 在现有的服务请求工作流程系统（如果有）中定义紧急访问请求流程。通常，此类工作流程系统允许您创建受理表来收集有关请求的信息，在工作流程的每个阶段跟踪请求，并添加自动和手动审批步骤。将每个请求与事件管理系统中所跟踪的相应紧急事件关联起来。通过拥有统一的紧急访问系统，您可以在单个系统中跟踪这些请求，分析使用趋势并改进流程。
- 确保您的紧急访问流程只能由授权用户启动，并且需要用户的同事或管理层（视情况而定）的批准。审批流程在工作时间内和工作时间之外应该能够有效运作。明确在主审批人抽不开身的情况下，如何允许辅助审批人审批请求，并沿管理链条上报，直至获得批准。
- 为紧急访问流程和机制实施稳健的日志记录、监控和警报机制。为所有成功和失败的紧急访问尝试生成详细的审计日志。将活动与事件管理系统中正在发生的紧急事件关联起来，并在预期时间段之外发生操作时发出警报，或者在正常运行期间使用紧急访问账户时发出警报。紧急访问账户仅限在紧急情况下访问，因为“打碎玻璃”程序可视为后门。与安全信息和事件管理（SIEM）工具或 AWS Security Hub CSPM 集成，来报告和审计紧急访问期间的所有活动。恢复正常操作后，自动轮换紧急访问凭证，并通知相关团队。
- 定期测试紧急访问流程，以确保步骤清楚明了，并快速高效地授予正确的访问级别。您的紧急访问流程应作为事件响应模拟（SEC10-BP07）和灾难恢复测试（REL13-BP03）的一部分进行测试。

故障模式 1：用于联合到 AWS 的身份提供程序不可用

如 SEC02-BP04 依赖集中式身份提供程序中所述，我们建议依靠集中式身份提供程序，来联合您的员工用户以授予对 AWS 账户的访问权限。您可以使用 IAM Identity Center 联合到 AWS 组织中的多个 AWS 账户，也可以使用 IAM 联合到单个 AWS 账户。在这两种情况下，员工用户都要先通过集中式身份提供程序进行身份验证，然后才会被重定向到 AWS 登录端点进行单点登录。

万一集中式身份提供程序不可用，员工用户就无法联合到 AWS 账户或管理其工作负载。在这种紧急情况下，您可以为一小部分管理员提供紧急访问流程，让他们访问 AWS 账户，来执行等不及集中式身份提供程序恢复正常的关键任务。例如，您的身份提供程序在 4 小时内不可用，在此期间，您需要修改生产账户中 Amazon EC2 Auto Scaling 组的上限，以应对客户流量意外激增的情况。您的紧急状况管理员应遵循紧急访问流程，以获得对特定生产 AWS 账户的访问权限并进行必要的更改。

紧急访问流程依赖于预先创建的紧急访问 AWS 账户，该账户仅用于紧急访问，并拥有 AWS 资源（如 IAM 角色和 IAM 用户）以支持紧急访问流程。在正常运营期间，任何人都不得访问紧急访问账户，而且您必须对滥用该账户的行为进行监控并发出警报（有关更多详情，请参阅前面的“通用指南”部分）。

紧急访问账户具有紧急访问 IAM 角色，有权在需要紧急访问的 AWS 账户中代入跨账户角色。这些 IAM 角色是预先创建的，并配置有信任策略，可信任应急账户的 IAM 角色。

紧急访问过程可以使用以下方法之一：

- 您可以在紧急访问账户中为紧急状况管理员预先创建一组 IAM 用户，并使用相关的强密码和 MFA 令牌。这些 IAM 用户有权代入 IAM 角色，然后在需要紧急访问时，允许跨账户访问 AWS 账户。我们建议尽可能少地创建此类用户，并将每个用户分配给一个紧急状况管理员。在紧急情况下，紧急状况管理员用户使用其密码和 MFA 令牌码登录紧急访问账户，切换到紧急账户中的紧急访问 IAM 角色，最后切换到工作负载账户中的紧急访问 IAM 角色，以执行紧急更改操作。这种方法的优点是，每个 IAM 用户都分配给一个紧急状况管理员，您可以通过查看 CloudTrail 事件来了解哪个用户已登录。缺点是，您必须维护多个 IAM 用户及其关联的长寿命密码和 MFA 令牌。
- 您可以使用紧急访问 AWS 账户根用户来登录紧急访问账户，代入用于紧急访问的 IAM 角色，并代入工作负载账户中的跨账户角色。建议为根用户设置一个强密码和多个 MFA 令牌。我们还建议将密码和 MFA 令牌存储在安全的企业凭证保管库中，该保管库可执行强身份验证和授权。您应确保密码和 MFA 令牌重置因素的安全：将账户的电子邮件地址设置为由云安全管理员监控的电子邮件分发列表，将账户的电话号码设置为同样由安全管理员监控的共享电话号码。这种方法的优点是只需管理一组根用户凭证。缺点是，由于这是共享用户，多个管理员都能以根用户身份登录。您必须审计企业保管库日志事件，以确定是哪位管理员查看了根用户密码。

故障模式 2：AWS 上的身份提供程序配置已修改或已过期

要允许您的员工用户联合到 AWS 账户，您可以使用外部身份提供程序配置 IAM Identity Center，或创建 IAM 身份提供程序（SEC02-BP04）。通常，您需要通过导入身份提供程序提供的 SAML 元数据 XML 文档，来配置这些服务。元数据 XML 文档包含一个 X.509 证书，该证书对应于身份提供程序用来签署其 SAML 断言的私钥。

管理员可能会错误地修改或删除 AWS 端的这些配置。在另一种情形下，导入到 AWS 的 X.509 证书可能会过期，而带有新证书的新元数据 XML 尚未导入到 AWS。这两种情形都可能使您的员工用户无法联合到 AWS，从而出现紧急情况。

在这种紧急情况下，您可以向您的身份管理员提供对 AWS 的访问权限以修复联合问题。例如，身份管理员使用紧急访问流程登录紧急访问 AWS 账户，切换到 Identity Center 管理员账户中的角色，并通过从身份提供程序导入最新的 SAML 元数据 XML 文档来更新外部身份提供程序配置，从而重新启用联合。修复联合后，您的员工用户将继续使用正常操作流程联合到其工作负载账户。

您可以按照前面的故障模式 1 中详述的方法来创建紧急访问流程。您可以向您的身份管理员授予最低访问权限，使其只能访问 Identity Center 管理员账户，并使用该账户对 Identity Center 执行操作。

故障模式 3：Identity Center 中断

如果发生 IAM Identity Center 或 AWS 区域中断这样的小概率事件，我们建议您设置一个可用于临时访问 AWS 管理控制台的配置。

紧急访问流程使用从身份提供程序到您的紧急账户中的 IAM 的直接联合。有关流程和设计注意事项的详细信息，请参阅《设置对 AWS 管理控制台的紧急访问》。

实施步骤

针对所有故障模式的通用步骤

- 创建专门用于紧急访问流程的 AWS 账户。预先创建账户中所需的 IAM 资源，例如 IAM 角色或 IAM 用户，以及可选的 IAM 身份提供程序。此外，在工作负载 AWS 账户中预先创建跨账户 IAM 角色，并与紧急访问账户中的相应 IAM 角色建立信任关系。您可以将 CloudFormation StackSets 与 AWS Organizations 结合使用，在您组织的成员账户中创建此类资源。

- 创建 AWS Organizations 服务控制策略（SCP），以拒绝删除和修改成员 AWS 账户中的跨账户 IAM 角色。
- 对紧急访问 AWS 账户启用 CloudTrail，并将跟踪事件发送到日志收集 AWS 账户中的中央 S3 存储桶。如果您使用 AWS Control Tower 来设置和管理您的 AWS 多账户环境，则您使用 AWS Control Tower 创建或在 AWS Control Tower 中注册的每个账户默认情况下都已启用 CloudTrail，并发送到专用日志存档 AWS 账户中的 S3 存储桶。
- 通过创建 EventBridge 规则来匹配紧急 IAM 角色所执行的控制台登录和 API 活动，来监控紧急访问账户的活动。当事件管理系统中所跟踪的正在发生的紧急事件之外出现活动时，向安全运营中心发送通知。

针对“故障模式 1：用于联合到 AWS 的身份提供程序不可用”和“故障模式 2：AWS 上的身份提供程序配置已修改或已过期”的其他步骤

- 根据您选择的紧急访问机制，预先创建资源：
  - 使用 IAM 用户：使用强密码和关联的 MFA 设备预先创建 IAM 用户。
  - 使用紧急账户的根用户：为根用户配置一个强密码，并将该密码存储在您的企业凭证库中。将多个物理 MFA 设备与根用户关联，并将设备存放在紧急状况管理员团队成员可以快速访问的位置。

针对“故障模式 3：Identity Center 中断”的其他步骤

- 如设置对 AWS 管理控制台的紧急访问中所详述的那样，在紧急访问 AWS 账户中，创建 IAM 身份提供程序，以启用从身份提供程序的直接 SAML 联合。
- 在 IdP 中创建没有成员的紧急行动组。
- 在紧急访问账户中创建与紧急行动组相对应的 IAM 角色。

资源

相关的 Well-Architected 最佳实践：

- SEC02-BP04 依赖集中式身份提供程序
- SEC03-BP02 授予最低访问权限
- SEC10-BP02 制定事件管理计划
- SEC10-BP07 执行实际演练

相关文档：

- 设置对 AWS 管理控制台的紧急访问
- 使 SAML 2.0 联合用户能够访问 AWS 管理控制台
- Break glass access

相关视频：

- AWS re:Invent 2022 - Simplify your existing workforce access with IAM Identity Center
- AWS re:Inforce 2022 - AWS Identity and Access Management (IAM) deep dive

相关示例：

- AWS Break Glass 角色
- AWS 客户行动手册框架
- AWS 事件响应行动手册样本

SEC03-BP04 持续减少权限

当您的团队确定所需的访问权限时，删除不需要的权限，并建立审核流程以实现最低权限。持续监控并删除供人类和机器访问的未使用的身份和权限。

期望结果：权限策略应遵循最低权限原则。随着工作职责和角色变得更加明确，需要审查您的权限策略以删除不必要的权限。如果无意中泄露或未经授权访问凭证，这种方法会缩小影响范围。

常见反模式：

- 默认为向用户授予管理员权限。
- 创建过于宽松但没有完全管理员权限的策略。
- 保留不再需要的权限策略。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

当团队和项目刚刚起步时，可以使用宽松的权限策略来激发创新并提高敏捷性。例如，在开发或测试环境中，开发人员可以获得广泛的访问权限以使用各种 AWS 服务。我们建议您持续评估访问权限，并仅限于访问完成当前作业所必需的服务和服务操作。对于人类和机器身份，均建议进行此项评估。机器身份有时称为系统或服务账户，是让 AWS 访问应用程序或服务器的身份。这种访问权限在生产环境中尤其重要，因为在该环境中，过于宽松的权限会产生广泛的影响，并可能暴露客户数据。

AWS 提供多种方法来帮助识别未使用的用户、角色、权限和凭证。AWS 还可帮助分析 IAM 用户和角色（包括关联的访问密钥）的访问活动，以及对 AWS 资源（如 Amazon S3 存储桶中的对象）的访问。AWS Identity and Access Management Access Analyzer 策略生成可帮助您根据主体与之交互的实际服务和操作来创建限制性权限策略。基于属性的访问权限控制（ABAC）可以帮助简化权限管理，因为您可以使用其属性向用户提供权限，而不必将权限策略直接附加到每个用户。

实施步骤

- 使用 AWS Identity and Access Management Access Analyzer：IAM Access Analyzer 帮助您标识企业和账户中与外部实体共享的资源，例如 Amazon Simple Storage Service（Amazon S3）存储桶或 IAM 角色。
- 使用 IAM Access Analyzer 策略生成：IAM Access Analyzer 策略生成可帮助您基于 IAM 用户或角色的访问活动创建精细的权限策略。
- 在生产之前跨较低环境测试权限：首先，使用 less critical sandbox and development environments，通过 IAM Access Analyzer 测试各种工作职能所需的权限。然后，在将这些权限应用于生产之前，在测试、质量保证和暂存环境中逐步收紧和验证这些权限。较低的环境最初可以拥有更宽松的权限，因为服务控制策略（SCP）通过限制所授予的最大权限来强制实施护栏。
- 确定 IAM 用户和角色的可接受时间范围和使用策略：使用上次访问时间戳可识别未使用的用户和角色并将它们移除。查看关于服务和操作的上次访问情况的信息，以确定和设置特定用户和角色的权限范围。例如，您可以使用关于上次访问情况的信息，以确定您的应用程序角色需要执行的特定 Amazon S3 操作，并只允许该角色访问这些操作。AWS 管理控制台中提供了上次获取的信息功能，您也可以对这些功能进行编程，以便将它们整合到您的基础架构工作流程和自动化工具中。

- 考虑在 AWS CloudTrail 中记录数据事件：默认情况下，CloudTrail 不会记录 Amazon S3 对象级活动（例如 `GetObject` 和 `DeleteObject`）或 Amazon DynamoDB 表活动（例如 `PutItem` 和 `DeleteItem`）等数据事件。考虑对这些事件使用日志记录，以确定哪些用户和角色需要访问特定的 Amazon S3 对象或 DynamoDB 表项目。

资源

相关文档：

- 授予最低权限
- 移除不必要的凭证
- 什么是 AWS CloudTrail？
- 使用 策略
- 记录和监控 DynamoDB
- 对 Amazon S3 存储桶和对象使用 CloudTrail 事件日志记录
- 获取您的 AWS 账户 的凭证报告

相关视频：

- 在最多 60 分钟的时间内成为 IAM 策略高手
- 职责分离、最低权限、委托和 CI/CD
- AWS re:Inforce 2022 - AWS Identity and Access Management (IAM) deep dive

SEC03-BP05 为您的组织定义权限防护机制

使用权限防护机制来减少可向主体授予的可用权限的范围。权限策略评估链包括您的防护机制，用于在做出授权决策时确定主体的有效权限。 您可以使用基于层的方法定义防护机制。在整个企业内广泛地应用一些防护机制，而对临时访问会话则以细粒度方式应用另一些防护机制。

期望结果：您可以使用单独的 AWS 账户对环境进行明确隔离。 使用服务控制策略（SCP）定义整个企业内的权限防护机制。在更靠近组织根级别的层次结构上设置较为广泛的防护机制，而在更靠近单独账户的级别上设置更严格的防护机制。

在支持资源策略的情况下，使用资源策略定义主体获得资源访问权限必须满足的条件。在适用的情况下，还应使用资源策略缩小允许操作的范围。在管理工作负载权限的主体上设定了权限边界，将权限管理委托给单独的工作负载负责人。

常见反模式：

- 在 AWS 组织内创建成员 AWS 账户，但不使用 SCP 来限制其根凭证的使用和可用权限。
- 根据最低权限原则分配了权限，但没有对可以授予的最大权限集施加防护机制。
- 依靠 AWS IAM 的隐式拒绝基础来限制权限，相信策略不会授予非预期的显式允许权限。
- 在同一个 AWS 账户中运行多个工作负载环境，然后依靠 VPC、标签或资源策略等机制来强制实施权限边界。

建立此最佳实践的好处：权限防护机制有助于建立人们对无法授予不需要的权限的信心，即使权限策略尝试这样做也是如此。 这样便可以减少需要考虑的最大权限范围，从而简化权限的定义和管理。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

我们建议您使用基于层的方法，为企业定义权限防护机制。在应用了额外的层时，这种方法可以系统性地减少可能的最大权限集。利用这种方法，您可以根据最低权限原则授予访问权限，从而减少由于策略配置错误而导致意外访问的风险。

建立权限防护机制的第一步是将您的工作负载和环境隔离到单独的 AWS 账户中。如果没有显式权限，一个账户的主体无法访问另一个账户中的资源，即使两个账户位于同一 AWS 组织或在同一组织单位（OU）下也是如此。您可以使用 OU 将要管理的账户分组为一个单位。

接下来，您需要减少可向企业成员账户中的主体授予的最大权限集。为此，您可以使用服务控制策略（SCP），您可以将其应用于 OU 或账户。SCP 可以强制执行常见的访问控制措施，例如限制对特定 AWS 区域的访问，防止资源被删除，或者禁用存在潜在风险的服务操作。您应用到组织根的 SCP 仅影响其成员账户，不影响管理账户。 SCP 仅管理企业中的主体。您的 SCP 不管理企业外部访问您资源的主体。

如果您使用的是 AWS Control Tower，则可以利用其 controls 和 landing zones 作为权限护栏和多账户环境的基础。登录区提供了一个预先配置的安全基线环境，为不同的工作负载和应用程序提供了单独的账户。这些护栏通过结合使用服务控制策略（SCP）、AWS Config 规则和其它配置，围绕安全性、运营和合规性实施强制性控制措施。但是，在使用 Control Tower 护栏和登录区以及自定义组织 SCP 时，请遵循 AWS 文档中概述的最佳实践来避免冲突并确保适当的治理，这一点至关重要。有关在 Control Tower 环境中管理 SCP、账户和组织单位（OU）的详细建议，请参阅 AWS Control Tower guidance for AWS Organizations。

通过遵守这些准则，可以有效地利用 Control Tower 的护栏、登录区和自定义 SCP，同时缓解潜在冲突并确保对多账户 AWS 环境进行适当的治理和控制。

下一步是使用 IAM 资源策略来限定您可以对其所管理的资源采取的可用操作的范围，以及代理主体必须满足的任何条件。这一点可以很宽泛，只要主体是您的组织的一部分，就可以允许执行所有操作（使用 PrincipalOrgId 条件键），也可以精细到仅允许特定 IAM 角色执行特定操作。对于 IAM 角色信任策略中的条件，您可以采用类似的方法。 如果资源或角色信任策略在所管理的角色或资源的同一个账户中，明确指定了主体，则该主体不需要附加授予相同权限的 IAM 策略。 如果主体与资源位于不同的账户中，则主体就需要附加 IAM 策略来授予这些权限。

通常，工作负载团队需要管理其工作负载所需的权限。 这可能要求团队创建新的 IAM 角色和权限策略。 您可以在 IAM 权限边界内获取允许团队授予的最大权限范围，并将此文档关联到一个 IAM 角色，然后团队可以使用该角色来管理其 IAM 角色和权限。 通过这种方法，团队能够灵活地完成其工作，同时降低拥有 IAM 管理访问权限的风险。

更精细的步骤是实施特权访问管理（PAM）和临时提升的访问权限管理（TEAM）技术。 PAM 的一个例子是要求主体在采取特权操作之前进行多重身份验证。 有关更多信息，请参阅《配置受 MFA 保护的 API 访问》。TEAM 需要一种解决方案，来管理主体在获得提升访问权限时需经过的审批，以及允许获得提升访问权限的时间范围。 一种方法是将主体临时添加到具有更高访问权限的 IAM 角色的角色信任策略中。 另一种方法是，在正常运行情况下，使用会话策略缩小 IAM 角色向主体授予的权限范围，然后在批准的时段内暂时取消此限制。要了解有关 AWS 和精选合作伙伴验证的解决方案的更多信息，请参阅《临时提升访问权限》。

实施步骤

1. 将您的工作负载和环境存放在单独的 AWS 账户中。
2. 使用 SCP 来减少可向企业成员账户中的主体授予的最大权限集。
    a. 在定义 SCP 来减少可向组织成员账户中的主体授予的最大权限集时，您可以选择允许列表 或拒绝列表 方法。允许列表策略显式指定允许的访问权限，并隐式阻止所有其它访问权限。拒绝列表策略显式指定不允许的访问权限，并且默认情况下允许所有其它访问权限。这两种策略都有其优势和权衡，适当的选择取决于您组织的具体要求和风险模型。有关更多详细信息，请参阅 Strategy for using SCPs。
    b. 此外，请查看 service control policy examples，来了解如何有效地构造 SCP。
3. 使用 IAM 资源策略缩小范围，并指定允许对资源执行操作的条件。 使用 IAM 角色信任策略中的条件来创建对代入角色的限制。
4. 将 IAM 权限边界分配给 IAM 角色，然后工作负载团队可以使用该角色来管理自己的工作负载的 IAM 角色和权限。
5. 根据您的需求评估 PAM 和 TEAM 解决方案。

资源

相关文档：

- 上的数据边界AWS
- 使用数据边界建立权限防护机制
- 策略评估逻辑

相关示例：

- 服务控制策略示例

相关工具：

- AWS 解决方案：临时提升的访问权限管理
- 经过验证的 TEAM 安全合作伙伴解决方案

SEC03-BP06 基于生命周期管理访问权限

在企业内主体（用户、角色和群组）的整个生命周期中，监控和调整授予主体的权限。在用户更改角色时调整组成员资格，并在用户离开企业时移除访问权限。

期望结果：您可以在企业内主体的整个生命周期中监控和调整权限，从而降低不必要权限的风险。在创建用户时授予合适的访问权限。随着用户职责的变化，您会修改访问权限，当用户不再活跃或已离开企业时，您会删除访问权限。您集中管理用户、角色和组的更改。您使用自动化方法将更改传播到 AWS 环境中。

常见反模式：

- 预先向身份授予过多或宽泛的访问权限，这些权限超出了最初的需求。
- 随着身份的角色和职责随着时间推移而发生变化，您未审核和调整访问权限。
- 让无效或离职的身份保留有效的访问权限。这样会增加未经授权访问的风险。
- 没有利用自动化功能来管理身份的生命周期。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

在身份（例如用户、角色、组）的整个生命周期中，谨慎管理和调整授予身份的访问权限。这一生命周期包括初始入职阶段、角色和职责的持续变化以及最终的离职或终止。根据生命周期的阶段主动管理访问权限，维护正确的访问权限级别。遵守最低权限原则，减少授予过多或不必要访问权限的风险。

您可以直接在 AWS 账户中管理 IAM 用户的生命周期，也可以通过从员工身份提供者到 AWS IAM Identity Center 的联合身份验证来进行管理。对于 IAM 用户，您可以在 AWS 账户中创建、修改和删除用户及其关联权限。对于联合用户，您可以使用 IAM Identity Center，通过 System for Cross-domain Identity Management（SCIM）协议，从组织的身份提供者同步用户和组信息，从而管理这些用户的生命周期。

SCIM 是一种开放标准协议，用于跨不同系统自动预置和取消预置用户身份。通过使用 SCIM 将身份提供者与 IAM Identity Center 集成，您可以自动同步用户和组信息，这有助于验证访问权限的授予、修改或撤销，而这些操作是基于企业中权威身份来源中的更改而进行的。

随着企业内员工角色和职责的变化，相应地调整员工的访问权限。您可以使用 IAM Identity Center 的权限集来定义不同的工作角色或职责，并将其关联到相应的 IAM 策略和权限。当员工的角色发生变化时，您可以更新向他们分配的权限集以反映他们的新职责。验证他们是否具有必要的访问权限，同时还遵守了最低权限原则。

实施步骤

1. 定义并记录访问权限管理生命周期流程，包括授予初始访问权限、定期审查和离职的程序。
2. 实施 IAM 角色、组和权限边界，以便从整体上管理访问权限并实施允许的最高访问权限级别。
3. 使用 IAM Identity Center，与联合身份提供者（例如 Microsoft Active Directory、Okta、Ping Identity）集成，作为用户和组信息的权威来源。
4. 使用 SCIM 协议，来将用户和组信息从身份提供者同步到 IAM Identity Center 的身份存储中。
5. 在 IAM Identity Center 中创建权限集，用于表示组织中的不同工作角色或职责。为每个权限集定义相应的 IAM 策略和权限。
6. 实施定期的访问权限审查，及时撤销访问权限，并持续改进访问权限管理生命周期流程。
7. 向员工提供访问权限管理最佳实践的培训，增强员工的意识。

资源

相关最佳实践：

- SEC02-BP04 依赖集中式身份提供程序

相关文档：

- 管理您的身份源
- 在 IAM Identity Center 中管理身份
- 使用 AWS Identity and Access Management Access Analyzer
- IAM Access Analyzer 策略生成

相关视频：

- AWS re:Inforce 2023 - Manage temporary elevated access with AWS IAM Identity Center
- AWS re:Invent 2022 - Simplify your existing workforce access with IAM Identity Center
- AWS re:Invent 2022 - Harness power of IAM policies & rein in permissions w/Access Analyzer

SEC03-BP07 分析公共和跨账户访问

持续监控重点关注公共访问和跨账户访问的调查发现。将公共访问和跨账户访问限制为仅限需要此访问的特定资源。

期望结果：了解您的哪些 AWS 资源是共享的，以及与谁共享。持续监控和审计您的共享资源，以验证它们仅与授权的主体共享。

常见反模式：

- 不保留共享资源的清单。
- 跨账户访问或公开访问资源时，没有遵循流程。

在未建立这种最佳实践的情况下暴露的风险等级：低

实施指导

如果您的账户在 AWS Organizations 中，您可以向整个组织、特定组织单位或个人账户授予资源访问权限。如果您的账户不是某个组织的成员，您可以与个人账户共享资源。您可以使用基于资源的策略[例如 Amazon Simple Storage Service（Amazon S3）存储桶策略] 或通过允许其他账户中的主体代入您账户中的 IAM 角色来授予直接跨账户访问权限。使用资源策略时，请验证访问权限是否仅授予给经过授权的主体。建立一个流程来审批所有需要可公开访问的资源。

AWS Identity and Access Management Access Analyzer 使用可证明安全性来标识从账户的外部访问某个资源时的所有访问路径。它持续审核资源策略，并报告公开访问和跨账户访问的调查发现，以使您能够轻松分析可能非常宽泛的访问权限。请考虑将 IAM Access Analyzer 与 AWS Organizations 一起配置，以验证您是否能够查看所有账户。IAM Access Analyzer 还允许您在部署资源权限之前预览调查发现。这样，您便可以验证策略更改仅按照意图，授权对您资源的公共和跨账户访问。在设计多账户访问权限时，您可以使用信任策略来控制在何种情况下可代入某个角色。例如，您可以使用 `PrincipalOrgId` 条件键拒绝尝试从您的 AWS Organizations 之外代入角色。

AWS Config 可以报告配置错误的资源，并可以通过 AWS Config 策略检查来检测配置了公有访问权限的资源。诸如 AWS Control Tower 和 AWS Security Hub CSPM 等服务简化了跨 AWS Organizations 的侦测性控制和护栏的部署，可以识别并修复公开暴露的资源。例如，AWS Control Tower 具有托管防护机制，可以检测是否有任何可由 AWS 账户恢复的 Amazon EBS 快照。

实施步骤

- 考虑将 AWS Config 用于 AWS Organizations：AWS Config 允许您将 AWS Organizations 中多个账户的结果聚合到委派管理员账户中。这样可提供全面的视图，并允许您跨账户部署 AWS Config 规则以检测可公开访问的资源。
- 配置 AWS Identity and Access Management Access Analyzer：IAM Access Analyzer 有助于您识别组织和账户中与外部实体共享的资源，例如 Amazon S3 存储桶或 IAM 角色。
- 在 AWS Config 中使用自动修复来响应 Amazon S3 存储桶的公共访问配置中的更改：您可以自动启用 Amazon S3 存储桶的阻止公共访问设置。
- 实施监控和警报，以确定 Amazon S3 存储桶是否已变为公共：您必须具备监控或警报机制，以便确定 Amazon S3 屏蔽公共访问权限何时关闭，以及 Amazon S3 存储桶是否变为公共。此外，如果您正在使用 AWS Organizations，您可以创建服务控制策略，以阻止对 Amazon S3 公共访问策略进行更改。AWS Trusted Advisor 可对具有开放访问权限的 Amazon S3 存储桶进行检查。如果向每个人授予“上传/删除”权限，那么任何人都可以向存储桶添加项目或者修改或删除存储桶中的项目，这样会产生潜在的安全问题。Trusted Advisor 检查可检查显式存储桶权限，以及可能覆盖存储桶权限的关联存储桶策略。您也可以使用 AWS Config 来监控 Amazon S3 存储桶是否具有公共访问权限。有关更多信息，请参阅《如何使用 AWS Config 来监控和响应允许公共访问的 Amazon S3 存储桶》。

在审核 Amazon S3 存储桶的访问控制时，务必考虑存储在存储桶中的数据的性质。Amazon Macie 是一项旨在协助您发现和保护敏感数据的服务，如个人身份信息（PII）、受保护健康信息（PHI）以及诸如私有密钥或 AWS 访问密钥等凭证。

资源

相关文档：

- 使用 AWS Identity and Access Management Access Analyzer
- AWS Control Tower 控件库
- AWS 基础安全最佳实践标准
- AWS Config 托管规则
- AWS Trusted Advisor 检查引用
- 通过 Amazon EventBridge 监控 AWS Trusted Advisor 的检查结果
- 管理组织内所有账户的 AWS Config 规则
- AWS Config 和 AWS Organizations
- 将您的 AMI 设为可在 Amazon EC2 中公开使用

相关视频：

- 保护多账户环境的最佳实践
- 深入了解 IAM Access Analyzer

SEC03-BP08 在组织内安全地共享资源

随着工作负载数量的增长，您可能需要共享对这些工作负载中资源的访问权限，或者跨多个账户多次预置资源。您可能需要进行构造来划分环境，例如划分成开发、测试和生产环境。但是，采取相互分离的构造并不会限制您安全共享权限。通过共享重叠的组件，您可以降低运维开销，并提供一致的体验，而不必猜测在多次创建同一资源时可能遗漏了什么。

期望结果：通过使用安全的方法在组织内共享资源，最大限度地减少意外访问，并帮助实施数据丢失防护计划。与管理单个组件相比，降低了运维开销，减少了多次手动创建同一组件时引起的错误，并提高了工作负载的可扩展性。您可以在多点故障场景中缩短问题解决时间，并在确定何时不再需要某个组件时更有信心。有关分析外部共享资源的规范性指南，请参阅《SEC03-BP07 分析公共和跨账户访问》。

常见反模式：

- 缺少对意外的外部共享进行持续监控和自动发出警报的流程。
- 缺乏关于应分享什么和不应分享什么的基准。
- 默认采用广泛的开放政策，而不是在需要时明确地分享。
- 手动创建在需要时重叠的基础资源。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

设计您的访问控制和模式，以安全地管理共享资源的使用，并且仅与可信实体共享。监控共享资源，持续检查共享资源访问权限，并在不适当或意外共享时发出警报。查看分析公共和跨账户访问，以帮助您建立治理，减少仅对需要访问的资源的外部访问权限，并建立持续监控和自动警报的流程。

AWS Organizations 内的跨账户共享受多个 AWS 服务支持，比如 AWS Security Hub CSPM、Amazon GuardDuty 和 AWS Backup。这些服务允许将数据共享到中心账户，可从中心账户访问，或从中心账户管理资源和数据。例如，AWS Security Hub CSPM 可将调查发现从个人账户转移到中心账户，在那里您可以查看所有调查发现。AWS Backup 可以对资源进行备份并在多个账户之间共享。可以使用 AWS Resource Access Manager（AWS RAM）共享其它公用资源，如 VPC subnets and Transit Gateway attachments、AWS Network Firewall 或 Amazon SageMaker AI pipelines。

要限制您的账户只能在组织内共享资源，请使用服务控制策略（SCP）来防止向外部主体授予访问权限。共享资源时，将基于身份的控制与网络控制结合起来，为组织创建数据边界，以帮助防止意外的访问。数据边界是一组预防性防护机制，用于帮助验证是否只有您的可信身份才能访问预期网络中的可信资源。这些控制措施会施加适当的限制，确定哪些资源可以共享，并防止共享或暴露不应被外泄的资源。例如，作为数据边界的一部分，您可以使用 VPC 端点策略和 `AWS:PrincipalOrgId` 条件，确保访问您的 Amazon S3 存储桶的身份属于您的组织。请务必注意，SCP 不适用于服务相关角色或 AWS 服务主体。

使用 Amazon S3 时，请关闭您的 Amazon S3 存储桶的 ACL，并使用 IAM 策略来定义访问控制。为了从 Amazon CloudFront 限制对 Amazon S3 来源的访问，请从来源访问身份（OAI）迁移到来源访问控制（OAC），后者支持使用其他功能，包括使用 AWS Key Management Service 进行服务器端加密。

在某些情况下，您可能希望允许在组织外部共享资源，或授予第三方访问您资源的权限。有关管理外部共享资源的权限的规范性指南，请参阅《权限管理》。

实施步骤

1. 使用 AWS Organizations：AWS Organizations 是一项账户管理服务，使您可以将多个 AWS 账户整合到您所创建的组织中并集中进行管理。您可以将账户分组为组织单位（OU），并将不同的策略附加到每个 OU，以帮助您满足预算、安全性和合规性需求。您还可以控制 AWS 人工智能（AI）和机器学习（ML）服务收集和存储数据的方式，并使用与 Organizations 集成的 AWS 服务的多账户管理。
2. 将 AWS Organizations 与 AWS 服务集成：当您使用 AWS 服务在组织的成员账户中代表您执行任务时，AWS Organizations 会在每个成员账户中为该服务创建一个 IAM 服务相关角色（SLR）。您

应使用 AWS 管理控制台、AWS API 或 AWS CLI 管理可信访问。有关开启可信访问权限的规范性指南，请参阅《将 AWS Organizations 与其它 AWS 服务结合使用》和《可与 Organizations 一起使用的 AWS 服务》。

3. 建立数据边界：数据边界提供了明确的信任和所有权边界。在 AWS 上，它通常表示为由 AWS Organizations 管理的 AWS 组织，以及访问您的 AWS 资源的任何本地网络或系统。建立数据边界的目标，是验证如果身份可信、资源可信并且网络符合预期，则支持进行访问。然而，建立数据边界并不是一个放之四海皆准的方法。根据您的特定安全风险模型和要求，评估并采纳 Building a Perimeter on AWS whitepaper 中概述的控制目标。您应仔细考虑自己独特的风险状况，并实施符合您安全需求的边界控制措施。
4. 在 AWS 服务中使用资源共享并相应进行限制：许多 AWS 服务支持您与另一账户共享资源，或以另一账户中的资源为目标，比如亚马逊机器映像（AMI）和 AWS Resource Access Manager（AWS RAM）。限制 `ModifyImageAttribute` API 以指定可信账户，从而与之共享 AMI。当需要使用 AWS RAM 来将共享限制于您的组织内部时，请指定 `ram:RequestedAllowsExternalPrincipals` 条件，以帮助防止来自不可信身份的访问。有关规范性指南和注意事项，请参阅《资源共享和外部目标》。
5. 使用 AWS RAM 在一个账户中或与其它 AWS 账户安全共享：AWS RAM 有助于您与账户中的角色和用户以及与其它 AWS 账户安全地共享已创建的资源。在多账户环境中，AWS RAM 使您能够一次性创建资源并与其他账户共享。这种方法有助于降低运维开销，同时通过与 Amazon CloudWatch 和 AWS CloudTrail 的集成提供一致性、可见性和可审计性，使用跨账户访问时无法获得这些好处。

   如果您拥有以前使用基于资源的策略共享的资源，则可以使用 `PromoteResourceShareCreatedFromPolicy API` 或等效 API 将资源共享升级为完全 AWS RAM 资源共享。

   在某些情况下，您可能需要采取其他步骤来共享资源。例如，要共享加密快照，您需要共享 AWS KMS 密钥。

资源

相关最佳实践：

- SEC03-BP07 分析公共和跨账户访问
- SEC03-BP09 与第三方安全地共享资源
- SEC05-BP01 创建网络层

相关文档：

- 存储桶拥有者向并非其拥有的对象授予跨账户权限
- 如何将信任策略与 IAM 结合使用
- 在 AWS 上构建数据边界
- 如何在向第三方授予对 AWS 资源的访问权限时使用外部 ID
- 可与 AWS Organizations 一起使用的 AWS 服务
- 在 AWS 上建立数据边界：仅允许可信身份获取公司数据

相关视频：

- 使用 AWS Resource Access Manager 实现精细访问
- 使用 VPC 端点保护您的数据边界
- 在 AWS 上建立数据边界

相关工具：

- 数据边界策略示例

SEC03-BP09 与第三方安全地共享资源

云环境的安全性不仅仅限于您的组织。您的组织有一部分数据可能要依赖第三方来管理。管理第三方托管系统的权限，应遵循及时访问的做法，使用最低权限原则和临时凭证。通过与第三方密切合作，您既可以缩小影响范围，又可以降低意外访问的风险。

期望结果：您避免使用长期 AWS Identity and Access Management（IAM）凭证，例如访问密钥和私密密钥，因为如果滥用，它们会构成安全风险。相反，可以使用 IAM 角色和临时凭证来改善您的安全状况，并最大限度地减少管理长期凭证的运营开销。在向第三方授予访问权限时，请在 IAM 信任策略中将通用唯一标识符（UUID）用作外部 ID，并将附加到角色的 IAM 策略置于您的控制之下，来确保最低权限访问。有关分析外部共享资源的规范性指南，请参阅 SEC03-BP07 分析公共和跨账户访问。

常见反模式：

- 采用默认的 IAM 信任策略，不附加任何条件。
- 使用长期 IAM 凭证和访问密钥。
- 重用外部 ID。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

您可能会希望允许在 AWS Organizations 外部共享资源，或授予第三方访问您账户的权限。例如，第三方提供的监控解决方案可能会需要访问您账户内部的资源。在这些情况下，请创建 IAM 跨账户角色，并仅向该角色提供第三方所需的权限。此外，使用外部 ID 条件定义信任策略。使用外部 ID 时，您或第三方可以为每个客户、第三方或租赁生成唯一 ID。创建唯一 ID 后，不应由除您之外的任何人控制它。第三方必须实施具体流程，以一种安全、可审计且可复制的方式将外部 ID 与客户关联起来。

您也可以使用 IAM Roles Anywhere 来管理 AWS 之外使用 AWS API 的应用程序的 IAM 角色。

如果第三方不再需要访问您的环境，则删除该角色。应避免向第三方提供长期凭证。了解其它支持共享的 AWS 服务，例如 AWS Well-Architected Tool 支持与其它 AWS 账户共享工作负载，而 AWS Resource Access Manager 有助于您安全地与其它账户共享您拥有的 AWS 资源。

实施步骤

1. 使用跨账户角色提供对外部账户的访问。跨账户角色可减少外部账户和第三方为服务客户而存储的敏感信息量。跨账户角色可让您将账户中 AWS 资源的访问权限安全地授予第三方（如 AWS 合作伙伴或组织内的其它账户），同时保持管理和审计该访问权限的能力。第三方可能从混合基础设施向您提供服务，或者将数据提取到一个异地位置。IAM Roles Anywhere 有助于您使第三方工作负载能够安全地与 AWS 工作负载交互，并进一步减少对长期凭证的需求。

   不应使用长期凭证或与用户关联的访问密钥来提供外部账户访问权限。而应使用跨账户角色来提供跨账户访问。

2. 进行尽职调查并确保第三方 SaaS 提供商的安全访问。当与第三方 SaaS 提供商共享资源时，请进行彻底的尽职调查，来确保他们采用安全和负责任的方法访问您的 AWS 资源。评估他们的责任共担模式，来了解他们提供了哪些安全措施以及哪些方面属于您的责任。确保 SaaS 提供商采用安全且可审计的流程来访问您的资源，包括使用外部 ID 和最低权限访问原则。使用外部 ID 有助于解决 confused deputy problem。

   实施安全控制以确保安全访问，并在向第三方 SaaS 提供商授予访问权限时遵守最低权限原则。这可能包括使用外部 ID、通用唯一标识符（UUID）和 IAM 信任策略，从而将访问权限限制在严格必要的范围内。与 SaaS 提供商密切合作，以便建立安全访问机制，定期审查他们对您的 AWS 资源的访问权限，并进行审计以确保符合安全要求。

3. 弃用客户提供的长期凭证。弃用长期凭证，使用跨账户角色或 IAM Roles Anywhere。如果必须使用长期凭证，请制定相应计划，逐渐转变成基于角色进行访问。有关管理密钥的详细信息，请参阅身

份管理。此外，与 AWS 账户团队和第三方合作来建立风险缓解运行手册。有关应对和缓解安全事件潜在影响的规范性指南，请参阅事件响应。

4. 验证设置是否具有规范性指南，或是否实现了自动化。外部 ID 不视为密钥，但外部 ID 不能是容易猜测的值，例如电话号码、姓名或账户 ID。将外部 ID 设置为只读字段，这样就无法为了冒充设置而更改外部 ID。

   您或第三方可以生成外部 ID。定义一个流程，确定谁负责生成 ID。无论创建外部 ID 的实体是什么，第三方都必须确保客户之间的唯一性和格式一致。

   为您账户中的跨账户访问创建的策略必须遵循最低权限原则。第三方必须为您提供使用 AWS CloudFormation 模板或等效模板的角色策略文档或自动化设置机制。这减少了手动创建策略时出错的机会，并提供了可审计的跟踪。有关使用 AWS CloudFormation 模板来创建跨账户角色的更多信息，请参阅 Cross-Account Roles。

   第三方应提供一个自动化的、可审计的设置机制。但是，通过使用角色策略文档（此文档大致列出了所需的访问权限），角色设置的自动化应该由您来完成。使用 AWS CloudFormation 模板或等效模板，您应将偏差检测纳入审计实践来监控变更。

5. 对变更做出解释。您的账户结构、您对第三方的需求或他们提供的服务可能会发生变更。您应预料到可能会发生变动和失败，并进行相应的规划：请安排合适的人员，建立适当的流程并采用正确的技术进行应对。应定期审计您提供的访问级别，并实施检测方法，以便在发生意外变更时向您发出警报。监控并审计角色的使用情况，以及外部 ID 的数据存储状态。若发生意外变更或存在不当访问模式，您应准备暂时或永久撤销第三方访问权限。此外，还要衡量撤销操作造成的影响，包括执行该操作所需的时间、涉及的人员、成本以及对其他资源的影响。

   有关检测方法的规范性指南，请参阅《检测最佳实践》。

资源

相关最佳实践：

- SEC02-BP02 使用临时凭证
- SEC03-BP05 为您的组织定义权限护栏
- SEC03-BP06 基于生命周期管理访问权限
- SEC03-BP07 分析公共和跨账户访问
- SEC04 检测

相关文档：

- 存储桶拥有者向并非其拥有的对象授予跨账户权限
- How to use trust policies with IAM roles
- 使用 IAM 角色委托跨 AWS 账户的访问权限
- 如何使用 IAM 访问其它 AWS 账户中的资源？
- IAM 安全最佳实操
- 跨账户策略评估逻辑
- 如何在向第三方授予对 AWS 资源的访问权限时使用外部 ID
- Collecting Information from AWS CloudFormation Resources Created in External Accounts with Custom Resources
- Securely Using External ID for Accessing AWS Accounts Owned by Others
- Extend IAM roles to workloads outside of IAM with IAM Roles Anywhere

相关视频：

- How do I allow users or roles in a separate AWS 账户 access to my AWS 账户?
- AWS re:Invent 2018: Become an IAM Policy Master in 60 Minutes or Less
- AWS Knowledge Center Live: IAM Best Practices and Design Decisions

相关示例：

- 配置对 Amazon DynamoDB 的跨账户访问
- AWS STS Network Query Tool

检测

检测由两个部分组成：意外或多余的配置更改检测，以及意外行为检测。第一部分检测可能出现在应用程序交付生命周期中的多个地方。利用基础设施即代码（例如 CloudFormation 模板），即可通过在 CI/CD 管道或源代码控制中实施检查，在部署工作负载之前检查是否存在多余配置。然后，在将工作负载部署到非生产和生产环境时，便可使用本机 AWS、开源或 AWS 合作伙伴工具来检查配置。这些检查可以针对不符合安全原则或最佳实践的配置，也可以针对在已测试和部署配置之间所做的更改。对于正在运行的应用程序，可以检查配置是否发生意外更改，包括在已知部署或自动扩展事件之外发生更改。

若要进行第二部分检测，即意外行为检测，您可以使用工具或在特定类型的 API 调用增加时发出警报。使用 Amazon GuardDuty 后，只要 AWS 账户内发生意外且可能未经授权的活动或恶意活动时，您就会收到提醒。您还应明确监控不希望在工作负载中使用的可变 API 调用以及会改变安全状况的 API 调用。

使用检测功能，您可以识别潜在安全配置错误、威胁或意外行为。检测是安全生命周期的重要组成部分，可用于支持质量流程、法律或合规义务，还可以用于威胁识别和响应工作。检测机制分为多种不同的类型。例如，可以分析来自工作负载的日志，找出正被利用的漏洞。您应定期审核与工作负载相关的检测机制，确保符合内外部的策略和要求。自动化警报和通知应基于所定义的条件，让团队或工具能够执行调查。这些机制都是重要的响应手段，可以帮助您的组织识别和了解异常活动的范围。

在 AWS 中，可以使用很多方法来解决检测性机制问题。以下各节旨在介绍如何使用这些方法：

最佳实践

- SEC04-BP01 配置服务和应用程序日志记录
- SEC04-BP02 在标准化位置收集日志、调查发现和指标
- SEC04-BP03 关联和扩充安全警报
- SEC04-BP04 启动对不合规资源的修复

SEC04-BP01 配置服务和应用程序日志记录

保留服务和应用程序的安全事件日志。这是审计、调查和运营使用案例的基本安全原则，也是由监管、风险与合规性（GRC，Governance, Risk, and Compliance）标准、政策和程序驱动的共同安全要求。

期望结果：当需要履行内部流程或义务（如安全事件响应）时，组织应能够及时、可靠且一致地从 AWS 服务和应用程序中检索安全事件日志。考虑将日志集中起来，以取得更好的运营成果。

常见反模式：

- 日志被永久存储或过早删除。
- 每个人都可以访问日志。
- 完全依赖手动流程进行日志治理和使用。
- 存储每一种类型的日志，以备不时之需。
- 仅在必要时检查日志完整性。

建立此最佳实践的好处：为安全事件实施根本原因分析（RCA）机制，并为您的监管、风险与合规性义务提供证据来源。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

在安全调查或基于要求的其他使用案例期间，您需要能够查看相关日志，以记录并了解事件的来龙去脉和时间线。警报生成也需要日志，以指示发生了某些感兴趣的操作。选择、启用、存储和设置查询、检索机制以及警报至关重要。

实施步骤

- 选择并启用日志源。进行安全调查之前，您需要捕获相关日志，以便以回溯方式重建 AWS 账户中的活动。选择与工作负载相关的日志源。

  日志源的选择标准应基于业务所需的使用案例。使用 AWS CloudTrail 或 AWS Organizations 跟踪为每个 AWS 账户 建立跟踪，并为其配置 Amazon S3 存储桶。

  AWS CloudTrail 是一项日志记录服务，可跟踪针对 AWS 账户捕获 AWS 服务活动所进行的 API 调用。它默认情况下启用，管理事件保留 90 天，可以使用 AWS 管理控制台、AWS CLI 或 AWS SDK 通过 CloudTrail 事件历史记录检索这些事件。为了更长久地保留和了解数据事件，请创建 CloudTrail 跟踪并将其与 Amazon S3 存储桶关联，也可以选择与 Amazon CloudWatch 日志组关联。或者，您可以创建 CloudTrail Lake，这可保留 CloudTrail 日志长达七年之久，并提供基于 SQL 的查询工具

  AWS 建议使用 VPC 的客户分别使用 VPC 流日志和 Amazon Route 53 Resolver 查询日志启用网络流量和 DNS 日志，并将其流式传输到 Amazon S3 存储桶或 CloudWatch 日志组。您可以为 VPC、子网或网络接口创建 VPC 流日志。对于 VPC 流日志，您可以选择使用流日志的方式和位置，以降低成本。

AWS CloudTrail 日志、VPC 流日志和 Route 53 解析器查询日志是支持 AWS 中安全调查的基本日志记录源。您还可以使用 Amazon Security Lake 以 Apache Parquet 格式和开放网络安全架构框架（OCSF）收集、标准化和存储这些日志数据，以便于查询。安全数据湖还支持其他 AWS 日志和来自第三方的日志。

AWS 服务可以生成基本日志源未捕获到的日志，如弹性负载均衡日志、AWS WAF 日志、AWS Config 记录器日志、Amazon GuardDuty 调查发现、Amazon Elastic Kubernetes Service（Amazon EKS）审计日志，以及 Amazon EC2 实例操作系统和应用程序日志。有关日志记录和监控选项的完整列表，请参阅《AWS Security Incident Response Guide》的 Appendix A: Cloud capability definitions – Logging and Events。

- 研究每项 AWS 服务和应用程序的日志记录功能：每项 AWS 服务和应用程序都为您提供了日志存储选项，每个选项都有自己的保留和生命周期功能。两种很常见的日志存储服务是 Amazon Simple Storage Service（Amazon S3）和 Amazon CloudWatch。如果保留期较长，建议使用 Amazon S3，因为它具有成本效益和灵活的生命周期功能。如果主要日志记录选项是 Amazon CloudWatch Logs，作为一种选择，您应该考虑将不太经常访问的日志存档到 Amazon S3。
- 选择日志存储：日志存储的选择通常与您使用的查询工具、保留能力、熟悉程度和成本有关。日志存储的主要选项是 Amazon S3 存储桶或 CloudWatch 日志组。

  Amazon S3 存储桶提供持久且经济高效的存储，并具有可选的生命周期策略。可以使用 Amazon Athena 等服务查询存储在 Amazon S3 存储桶中的日志。

  CloudWatch 日志组通过 CloudWatch Logs Insights 提供持久存储和内置查询工具。
- 确定适当的日志保留时长：使用 Amazon S3 存储桶或 CloudWatch 日志组存储日志时，必须为每个日志源建立足够的生命周期，以优化存储和检索成本。客户通常可以查询三个月到一年的日志，日志保留期长达七年。可用性和保留时长的选择应与您的安全要求以及法律法规和业务授权的综合因素相一致。
- 使用适当的保留时长和生命周期策略为每个 AWS 服务和应用程序启用日志记录：对于组织内的每个 AWS 服务或应用程序，请查找特定的日志记录配置指南：
  - 配置 AWS CloudTrail 跟踪
  - 配置 VPC 流日志
  - 配置 Amazon GuardDuty 调查发现导出
  - 配置 AWS Config 记录
  - 配置 AWS WAF Web ACL 流量
  - 配置 AWS Network Firewall 网络流量日志
  - 配置弹性负载均衡访问日志

  - 配置 Amazon Route 53 Resolver 查询日志
  - 配置 Amazon RDS 日志
  - 配置 Amazon EKS 控制面板日志
  - 为 Amazon EC2 实例和本地服务器配置 Amazon CloudWatch 代理
- 选择和实施日志查询机制：对于日志查询，可以使用 CloudWatch Logs Insights 对存储在 CloudWatch 日志组中的数据进行查询，使用 Amazon Athena 和 Amazon OpenSearch Service 对存储在 Amazon S3 中的数据进行查询。您还可以使用第三方查询工具，如安全信息和事件管理（SIEM）服务。

  选择日志查询工具的过程中，应考虑安全运营的人员、流程和技术方面。选择一款能够满足运营、业务和安全要求并可长期使用和维护的工具。请记住，当要扫描的日志数量保持在工具的限制范围内时，日志查询工具的工作状态最佳。由于成本或技术限制，拥有多款查询工具的情况并不罕见。

  例如，您可能使用第三方安全信息和事件管理（SIEM）工具对过去 90 天的数据执行查询，但由于 SIEM 的日志提取成本较高，使用 Athena 来执行 90 天以上的查询。无论采用何种实施方式，都要验证您的方法能够尽可能地减少充分提高运营效率所需的工具数量，尤其在安全事件调查期间。
- 使用日志发出警报：AWS 通过多项安全服务提供警报功能：
  - AWS Config 监控和记录您的 AWS 资源配置，并允许您对照所需的配置自动执行评估和修复。
  - Amazon GuardDuty 是一项威胁检测服务，可持续监控恶意活动和未经授权的行为，以保护您的 AWS 账户和工作负载。GuardDuty 可从 AWS CloudTrail 管理和数据事件、DNS 日志、VPC 流日志和 Amazon EKS 审计日志等来源提取、聚合和分析信息。GuardDuty 可直接从 CloudTrail、VPC 流日志、DNS 查询日志和 Amazon EKS 提取独立的数据流。您无需管理 Amazon S3 存储桶策略，也无需修改日志的收集和存储方式。仍建议保留这些日志，以便您自己进行调查和遵守法规。
  - AWS Security Hub CSPM 集中聚合、组织和优先处理来自多个 AWS 服务和可选第三方产品的安全警报或调查发现，以使您全面了解安全警报和合规性状态。

  您也可以使用自定义警报生成引擎来处理这些服务未涵盖的安全警报或与您的环境相关的特定警报。有关构建这些警报和检测的信息，请参阅《AWS Security Incident Response Guide》中的 Detection。

资源

相关最佳实践：

- SEC04-BP02 在标准化位置收集日志、调查发现和指标

- SEC07-BP04 定义可扩展的数据生命周期管理
- SEC10-BP06 预部署工具

相关文档：

- 《AWS Security Incident Response Guide》
- Amazon Security Lake 入门
- 入门：Amazon CloudWatch Logs

相关视频：

- AWS re:Invent 2022 - Introducing Amazon Security Lake

相关示例：

- 专为 AWS 提供的 Assisted Log Enabler
- AWS Security Hub CSPM调查发现历史导出

SEC04-BP02 在标准化位置收集日志、调查发现和指标

安全团队依靠日志和调查发现来分析事件，找出可能表明出现了未经授权活动或意外更改的事件。为了简化这种分析过程，请将安全日志和调查发现收集到标准化位置。 这样就能将需要分析的数据点用于关联，并可以简化工具集成。

期望结果：您采用标准化的方法来收集、分析和可视化日志数据、调查发现和指标。安全团队可以高效地关联、分析和可视化不同系统的安全数据，以便发现潜在的安全事件并识别异常情况。集成了安全信息和事件管理（SIEM，Security Information and Event Management）系统或其它机制，用于查询和分析日志数据，以便及时响应、跟踪和上报安全事件。

常见反模式：

- 团队独立负责和管理日志记录及指标的收集，但是采取了与企业的日志记录策略不一致的方法。
- 团队没有采取足够的访问控制措施来限制所收集数据的可见性以及对数据的更改。
- 团队没有将其安全日志、调查发现和指标包括在数据分类策略中进行管理。
- 团队在配置数据收集时，忽略了数据主权和本地驻留要求。

建立此最佳实践的好处：采用标准化日志记录解决方案来收集和查询日志数据及事件，可以改善从所包含的信息中获得的见解。为收集的日志数据配置自动处理生命周期，可以降低日志存储产生的成本。您可以根据数据的敏感性和团队需要的访问模式，对收集的日志信息建立精细的访问控制。您可以集成工具，用于关联和可视化数据，以及从数据中发掘洞察。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

企业内 AWS 使用量的增长会导致分布式工作负载和环境数量的增加。由于这些工作负载和环境都会生成有关其内部活动的数据，因此在本地收集和存储这些数据对安全运营带来了挑战。安全团队使用安全信息和事件管理（SIEM，Security Information and Event Management）系统等工具，从分布式来源收集数据，并完成关联、分析和响应等工作流。这需要管理一组复杂权限以便访问各种数据来源，而且提取、转换、加载（ETL）流程的操作会带来额外开销。

要克服这些挑战，可以考虑将所有相关的安全日志数据来源汇总到一个日志归档账户中，如 Organizing Your AWS Environment Using Multiple Accounts 中所述。这包括您的工作负载的所有与安全相关的数据，以及各种 AWS 服务生成的日志，例如 AWS CloudTrail、AWS WAF、弹性负载均衡和 Amazon Route 53。使用合适的跨账户权限，在单独的 AWS 账户中的标准化位置收集这些数据可以带来多种好处。这种做法有助于防止受感染的工作负载和环境中发生日志篡改，提供可供其它工具使用的单一集成点，并为配置数据留存和生命周期提供更简化的模型。 评估数据主权、合规范围和其它法规的影响，确定是否需要多个安全数据存储位置和保留期。

为了简化日志和调查发现的收集和标准化工作，请评估在您的日志存档账户中是否适合使用 Amazon Security Lake。您可以将 Security Lake 配置为自动从各种常见事件源中摄取数据，例如 CloudTrail、Route 53、Amazon EKS 和 VPC 流日志。您也可以将 AWS Security Hub CSPM 配置作为传输到 Security Lake 的数据来源，这样您就可以将来自其它 AWS 服务（例如 Amazon GuardDuty 和 Amazon Inspector）的调查发现与您的日志数据相关联。 您还可以使用第三方数据来源集成，或配置自定义数据来源。所有集成都将您的数据标准化为开放网络安全架构框架（OCSF，Open Cybersecurity Schema Framework）格式，并作为 Parquet 文件存储在 Amazon S3 存储桶中，从而消除了 ETL 处理需求。

将安全数据存储在标准化位置可提供高级分析功能。AWS 建议您将 AWS 环境中运行的安全分析工具部署到安全工具账户中，而且该账户独立于您的日志存档账户。通过这种方法，您可以实施深入的控制措施，用来保护日志和日志管理流程的完整性和可用性，与访问这些日志的工具区分开。 考虑使用服务（例如 Amazon Athena）来运行关联多个数据来源的按需查询。还可以集成可视化工具，例如 Quick。人工智能驱动的解决方案的应用日益广泛，可以执行很多功能，例如将发现结果转化为人类可读的摘要并以自然语言进行交互。为查询设置标准化数据存储位置后，这些解决方案通常会更容易集成。

实施步骤

1. 创建日志存档账户和安全工具账户
   a. 使用 AWS Organizations，在安全组织单位下创建日志存档账户和安全工具账户。如果您使用 AWS Control Tower 来管理企业，则会自动为您创建日志存档账户和安全工具账户。您可以根据需要，配置用于访问和管理这些账户的角色和权限。
2. 配置标准化安全数据位置
   a. 确定创建标准化安全数据位置的策略。 为此，您可以使用通用数据湖架构方法、第三方数据产品或 Amazon Security Lake 等方案。AWS 建议您从您的账户选择加入的 AWS 区域中收集安全数据，即使这些区域并未处于活跃使用状态。
3. 将数据来源配置为发布到您的标准化位置
   a. 确定您的安全数据的来源，并将这些来源配置为发布到您的标准化位置。评估能够以所需格式自动导出数据的方案，而不是那些需要开发 ETL 流程的方案。使用 Amazon Security Lake，您可以从支持的 AWS 来源和集成的第三方系统收集数据。
4. 配置工具来访问您的标准化位置
   a. 配置 Amazon Athena、Quick 或第三方解决方案等工具，来获取您的标准化位置所要求的访问权限。 将这些工具配置为在安全工具账户之外运行，并在适用时，允许对日志存档账户的跨账户读取访问。在 Amazon Security Lake 上创建订阅用户，以便向这些工具提供对您数据的访问权限。

资源

相关最佳实践：

- SEC01-BP01 使用账户分隔工作负载
- SEC07-BP04 定义数据生命周期管理
- SEC08-BP04 强制实施访问控制
- OPS08-BP02 分析工作负载日志

相关文档：

- AWS Whitepapers: Organizing Your AWS Environment Using Multiple Accounts
- AWS Prescriptive Guidance: AWS Security Reference Architecture (AWS SRA)
- AWS Prescriptive Guidance: Logging and monitoring guide for application owners

相关示例：

- Aggregating, searching, and visualizing log data from distributed sources with Amazon Athena and Quick
- How to visualize Amazon Security Lake findings with Quick
- Generate AI powered insights for Amazon Security Lake using Amazon SageMaker AI Studio and Amazon Bedrock
- Identify cybersecurity anomalies in your Amazon Security Lake data using Amazon SageMaker AI
- Ingest, transform, and deliver events published by Amazon Security Lake to Amazon OpenSearch Service
- Simplify AWS CloudTrail log analysis with natural language query generation in CloudTrail Lake

相关工具：

- Amazon Security Lake
- Amazon Security Lake 合作伙伴集成
- 开放式网络安全架构框架（OCSF）
- Amazon Athena
- Quick
- Amazon Bedrock：

SEC04-BP03 关联和扩充安全警报

出现意外的活动时，不同的来源可能会生成多个安全警报，此时需要进一步关联和扩充警报以便理解整个背景信息。实施自动化的安全警报关联和扩充，有助于更准确地识别和响应事件。

期望结果：当您的工作负载和环境中的活动生成了不同警报时，自动化机制会关联数据，并用其它信息扩充这些数据。通过这种预处理方法，您可以更详细地了解事件，这可以帮助调查人员确定事件的严重程度，以及是否构成了需要正式响应的事件。这个流程可减轻监控和调查团队的负担。

常见反模式：

- 除非职责分离要求另有规定，否则由不同团队的人员调查不同系统生成的调查发现和警报。
- 企业将所有安全调查发现和警报数据汇集到标准位置，但需要调查人员手动进行关联和扩充。
- 您完全依靠威胁检测系统的情报来报告调查发现并确定严重程度。

建立此最佳实践的好处：自动关联和扩充警报有助于减少调查人员的总体认知负荷和数据准备人工工作。这种做法可以缩短确定事件是否表示出现事故并启动正式响应所需的时间。其它背景信息还可以帮助您准确评测事件的真实严重性，因为其严重性可能会高于或低于任何警报所暗示的严重性。

在未建立这种最佳实践的情况下暴露的风险等级：低

实施指导

安全警报可能来自 AWS 中的许多不同来源，包括：

- Amazon GuardDuty、AWS Security Hub CSPM、Amazon Macie、Amazon Inspector、AWS Config、AWS Identity and Access Management Access Analyzer 和网络访问分析器等服务
- 来自对 AWS 服务、基础设施和应用程序日志的自动分析的警报，例如来自 Amazon OpenSearch Service 的安全分析。
- 响应 Amazon CloudWatch、Amazon EventBridge 或 AWS Budgets 等来源中账单活动变化的警报。
- 第三方来源，例如来自 AWS Partner Network 的威胁情报源和安全合作伙伴解决方案
- AWS 信任与安全团队或其它来源（例如客户或内部员工）发起的联系事宜。
- 使用 Threat Technique Catalog by AWS (TTC)，通过入侵指标（IoC）识别来协助识别和关联威胁行为者的行为。TTC 是 MITRE ATT&CK 框架的扩展，对所有已知和观测到的针对 AWS 资源的威胁行为者行为和技术进行了分类。

警报的最基本形式中包含有关谁（主体或身份）对什么（受影响的资源）做了哪些事（操作）的信息。对于每个这些来源，确定是否有办法可以为这些身份、操作和资源跨标识符创建映射，以此作为执行关联的基础。要想做到这一点，可以将警报来源与安全信息和事件管理（SIEM，Security Information and Event Management）工具集成在一起来为您执行自动关联，或者构建自己的数据管道和数据处理，也可将两种方法结合使用。

Amazon Detective 就是一个可以为您执行关联的服务示例。Detective 从各种 AWS 来源和第三方来源持续摄取警报，并使用不同形式的情报来描绘出这些关系的可视化图表，以协助进行调查。

虽然警报的初始严重程度可以用于协助排列优先顺序，但发生警报的具体环境决定了真正的严重程度。例如，Amazon GuardDuty 可能会发出警报，指出您工作负载中的某个 Amazon EC2 实例正在查询意外的域名。GuardDuty 可能会自行为此警报指定低严重程度。但是，通过自动关联发生警报时间前后的其它活动，可能会发现通过同一个身份部署了数百个 EC2 实例，这会增加总体运营成本。在这种情况下，这种相关的事件上下文将需要一个新的安全警报，且严重程度可能会被调整为高，这将加快进一步的行动。

实施步骤

1. 确定安全警报信息的来源。了解来自这些系统的警报所代表的身份、操作和资源，以便确定可能的关联性。
2. 建立一种机制来收集源自不同来源的警报。对于此目的，可以考虑使用 Security Hub CSPM、EventBridge 和 CloudWatch 等服务。
3. 确定用于数据关联和扩充的来源。示例来源包括 AWS CloudTrail、VPC 流日志、Route 53 Resolver 日志以及基础设施和应用程序日志。这些日志中的任何一个或所有日志都可以通过与 Amazon Security Lake 的单一集成来使用。
4. 将警报与您的数据关联和扩充来源集成，创建更详细的安全事件背景信息并确定严重程度。
    a. Amazon Detective、SIEM 工具或其它第三方解决方案可以自动执行一定级别的摄取、关联和扩展。
    b. 您也可以使用 AWS 服务来构建自己的服务。例如，您可以调用一个 AWS Lambda 函数来对 AWS CloudTrail 或 Amazon Security Lake 运行 Amazon Athena 查询，并将结果发布到 EventBridge。

资源

相关最佳实践：

- SEC10-BP03 准备取证能力
- OPS08-BP04 创建可操作的警报
- REL06-BP03 发送通知（实时处理和报警）

相关文档：

- AWS《Security Incident Response Guide》

相关示例：

- How to enrich AWS Security Hub CSPM findings with account metadata

相关工具：

- Amazon Detective

- Amazon EventBridge
- AWS Lambda
- Amazon Athena

SEC04-BP04 启动对不合规资源的修复

您的检测性控制措施可能会对不符合配置要求的资源发出警报。您可以手动或自动启动以编程方式定义的修复措施，用于修复这些资源并尽可能减少潜在影响。通过以编程方式定义的修复措施，您可以迅速采取一致的行动。

虽然利用自动化功能可以增强安全修复操作，但您应谨慎地实施和管理自动化功能。 您需要建立适当的监督和控制机制，确保自动化响应有效、准确且符合组织的策略和风险偏好。

期望结果：您定义了资源配置标准，以及在检测到资源不合规情况时应采取的修复措施。您尽可能以编程方式定义修复措施，以便手动或者自动启动这些措施。实施了检测系统，用于识别不合规的资源，并将警报发布到由您的安全人员监控的集中式工具。这些工具支持手动或自动运行您的程序化修复措施。采取了相应的监督和控制机制来管控自动修复措施的使用。

常见反模式：

- 您实施了自动化功能，但没有全面测试和验证修复措施。这可能会导致意外的后果，例如中断合法的业务运营或导致系统不稳定。
- 您利用自动化功能来改善响应时间和流程，但没有采取适当的监控和机制，以便在需要时进行人工干预和判断。
- 您完全依赖修复措施，而不是将修复措施作为一部分，融入到更全面的事件响应和恢复计划中。

建立此最佳实践的好处：面对错误配置，自动修复的响应速度比手动流程更快，这让您可以尽量减少潜在的业务影响，并减少出现意外使用情况的机会。以编程方式定义修复措施后，可以始终如一地应用这些措施，从而降低人为错误的风险。自动化功能还可以同时处理更大量的警报，这在大规模运行的环境中尤其重要。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

如 SEC01-BP03 识别并验证控制目标中所述，AWS Config 和 AWS Security Hub CSPM 等服务有助于您监控账户中资源的配置，以便符合您的要求。当检测到不合规的资源时，诸如 AWS Security Hub

CSPM 之类的服务有助于相应地发送警报并进行补救。这些解决方案为安全调查人员提供了一个中心位置，可用来监控问题和采取纠正措施。

除 AWS Security Hub CSPM 之外，AWS 还引入了 Security Hub Advanced。这项服务在 re:Invent 2025 上发布，它转变了组织优先考虑其最关键的安全问题并大规模响应以保护其云环境的方式。增强的 Security Hub 现在使用高级分析来自动关联、丰富云环境中的安全信号并确定这些信号的优先级。Security Hub 与 Amazon GuardDuty、Amazon Inspector、Amazon Macie 和 AWS Security Hub CSPM 无缝集成。Security Hub 中的相关调查发现可能导致全新的调查发现，称为暴露调查发现，其中包括基于每种资源中发现的漏洞的假设攻击路径。

一些不合规资源的情况会非常独特，需要人工判断才能修复，不过其它情况可以采取以编程方式定义的标准响应。例如，对于 VPC 安全组的配置错误，标准响应可以是删除不允许的规则并通知负责人。可以在 AWS Lambda 函数、AWS Systems Manager Automation 文档中或者通过其它您常用的代码编写环境来定义响应措施。确保环境能够使用 IAM 角色在 AWS 上进行身份验证，并具有执行纠正措施所需的最低权限。

在定义了所需的修复措施之后，您就可以确定启动修复措施的首选方式。AWS Config 可以为您启动修复。如果您使用 Security Hub CSPM，则可以通过自定义操作来执行这个过程，自定义操作会将调查发现信息发布到 Amazon EventBridge。然后，EventBridge 规则启动修复措施。您可以通过 Security Hub CSPM 配置补救措施，使其自动或手动运行。

对于程序化的修复措施，我们建议您全面记录所采取的行动及其结果，并进行审计。查看和分析这些日志，以评测自动化流程的有效性，并确定需要改进的地方。将 Amazon CloudWatch Logs 中的日志和修复结果作为调查发现备注收集到 Security Hub CSPM 中。

首先，您可以考虑 AWS 上的自动化安全响应，其中预先构建了修复措施，用于解决常见的安全配置错误。

实施步骤

1. 分析警报并确定其优先级。
    a. 将来自各种 AWS 服务的安全警报整合到 Security Hub CSPM 中，以便在集中位置进行查看、划分优先级和执行修复措施。
2. 制定修复措施。
    a. 使用 Systems Manager 和 AWS Lambda 等服务来运行程序化的修复措施。
3. 配置启动修复的方式。
    a. 使用 Systems Manager，定义将调查发现发布到 EventBridge 的自定义操作。将这些操作配置为手动或自动启动。

   b. 如果需要，您还可以使用 Amazon Simple Notification Service（SNS）向相应的利益相关者（例如安全团队或事件响应团队）发送通知和警报，以便进行手动干预或上报。

4. 查看和分析修复日志，了解有效性并发现改进的机会。
   a. 将日志输出发送到 CloudWatch Logs。在 Security Hub CSPM 中将结果捕获为调查发现备注。

资源

相关最佳实践：

- SEC06-BP03 减少人工管理工作和交互式访问

相关文档：

- AWS《Security Incident Response Guide》- Detection

相关示例：

- 上的自动化安全响应AWS
- Monitor EC2 instance key pairs using AWS Config
- Create AWS Config custom rules by using AWS CloudFormation Guard policies
- Automatically remediate unencrypted Amazon RDS DB instances and clusters

相关工具：

- AWS Systems Manager Automation
- 上的自动化安全响应AWS

基础设施保护

基础设施保护包括满足最佳实践和组织、法律及监管义务所必需的控制方法（例如深度防御）。使用这些方法对于在云中持续成功运营至关重要。

基础设施保护是信息安全计划的一个关键组成部分。其可确保工作负载中的系统和服务受到保护，防止意外和未经授权的访问以及潜在漏洞造成的危害。例如，您可以定义信任边界（例如网络边界和账户边界）、系统安全配置和维护（例如强化、最小化和修补）、操作系统身份验证和授权（例如用户、密钥和访问级别）以及其他适当的策略执行点（例如 Web 应用程序防火墙和/或 API 网关）。

区域、可用区、AWS Local Zones 和 AWS Outposts

务必熟知区域、可用区、AWS Local Zones 和 AWS Outposts 的知识，因为这些都是 AWS 安全全球基础设施的组件。

AWS 存在区域概念，区域是世界各地数据中心聚集的物理位置。每组逻辑数据中心称为可用区（AZ）。每个 AWS 区域由一个地理区域内的多个可用区组成，这些可用区彼此隔离并在物理上分开。如果对数据驻留有要求，则可以选择靠近所需位置的 AWS 区域。您保留对数据实际所在区域的完全控制权和所有权，因为这有助于满足您的区域合规性和数据驻留要求。每个可用区都有独立的电源、冷却和物理安防设施。如果跨可用区对应用程序进行分区，则可以实现更好的隔离和保护，防止受到停电、雷击、龙卷风、地震等事故和灾害的影响。一个可用区与其余可用区在物理上相距很远，尽管所有可用区之间的距离都在 100 公里（60 英里）以内。与此同时，AWS 中的所有可用区都通过完全冗余的专用城域光纤，以高带宽、低延迟的网络进行互联，从而在可用区之间实现高吞吐量、低延迟的联网。可用区之间的所有流量都经过加密。注重高可用性的 AWS 客户可以将应用程序设计为在多个可用区中运行，从而实现更高的容错能力。AWS区域满足最高级别的安全性、合规性和数据保护要求。

AWS Local Zones 将计算、存储、数据库和其他精选 AWS 服务安排在更靠近最终用户的地方。借助 AWS Local Zones，您可以轻松运行要求极高的应用程序（对最终用户的延迟仅为几毫秒），例如媒体和娱乐内容创作、实时游戏、油藏模拟、电子设计自动化和机器学习。每个 AWS Local Zone 位置都是一个 AWS 区域的扩展，可供您在其中运行对延迟敏感的应用程序，使用地理上靠近最终用户的 AWS 服务，例如 Amazon EC2、Amazon VPC、Amazon EBS、Amazon File Storage 和弹性负载均衡（ELB）。AWSLocal Zones 在本地工作负载和 AWS 区域中运行的工作负载之间提供高带宽的安全连接，允许通过相同的 API 和工具集无缝连接到所有区域内服务。

AWS Outposts 可将本机 AWS 服务、基础设施和运营模式引入几乎任何数据中心、主机托管空间或本地设施。您可以在各本地设施和 AWS 云中使用相同的 AWS API、工具和基础设施来提供真正一致的

混合体验。AWSOutposts 专为互联环境设计，可用于支持因低延迟或本地数据处理需求而必须保留在本地的工作负载。

在 AWS 中，有许多基础设施保护方法。以下各节旨在介绍如何使用这些方法。

主题

保护网络

无论是己方用户还是客户方用户，都可能位于任何地方。您必须摆脱传统模式，即不再对任何有权访问您网络的人员和设备保持信任。遵循在所有层应用安全原则时，也就采用了零信任（Zero Trust）方法。零信任安全是一种模型，在这种模型中，应用程序组件或微服务被视作彼此分离，且任何组件或微服务都不信任其他组件或微服务。

妥善规划和管理网络设计，这是为工作负载中的资源提供隔离和边界的基础。由于工作负载中的很多资源都运行在 VPC 中并继承安全属性，因此必须使用由自动化作为后盾的检查和保护机制来支持设计。同理，对于在 VPC 之外运行的工作负载，当使用纯粹边缘服务和/或无服务器环境时，这些最佳实践适用于更加简化的方法。请参阅 AWS Well-Architected Serverless Applications Lens，获得有关无服务器安全性的具体指导。

最佳实践

SEC05-BP01 创建网络层

根据工作负载组件的逻辑分组，按照数据敏感性和访问权限要求，将网络拓扑划分成不同的层。区分需要接受来自互联网的入站访问的组件（例如公有 Web 端点）与只需要进行内部访问的组件（例如数据库）。

期望结果：网络中的各个层是完整的深度防御安全方法的一部分，是工作负载身份验证和授权策略的有力补充。根据数据敏感性和访问权限要求进行了分层，并采用合适的流量流动和控制机制。

常见反模式：

- 您在单个 VPC 或子网中创建所有资源。
- 在构造网络层时，您没有考虑数据敏感性要求、组件行为或功能。
- 您使用 VPC 和子网的默认值，未考虑所有的网络层注意事项，而且也未考虑 AWS 托管服务对拓扑有何影响。

建立此最佳实践的好处：建立网络层是限制网络中不必要路径的第一步，尤其是在需要限制去往关键系统和数据的路径的情况下。通过这种方法，未经授权的操作者更难于访问您的网络，也更难导航到网络中的其它资源。彼此分隔的网络层带来的益处包括减少了检查系统的分析范围，例如进行入侵检测或恶意软件防御时。这可以减少误报的可能性和不必要的处理开销。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

在设计工作负载架构时，一种常见的做法是根据组件的职责将组件分隔到不同的层中。例如，Web 应用程序可以具有表示层、应用层和数据层。在设计网络拓扑时，您可以采用类似的方法。底层网络控制措施可用于强制执行工作负载的数据访问权限要求。例如，在三层 Web 应用程序架构中，您可以将静态的表示层文件存储在 Amazon S3 中，并通过 Amazon CloudFront 等内容分发网络（CDN）来提供这些文件。应用层可以有公有端点，该端点由应用程序负载均衡器（ALB）在 Amazon VPC 公有子网（类似于非军事区，简称 DMZ）中提供服务，并将后端服务部署到私有子网中。数据层中托管数据库和共享文件系统等资源，可以与应用层的资源位于不同的私有子网中。在每个这些层的边界（CDN、公有子网、私有子网），您都可以部署控制措施，仅允许授权流量穿过这些边界。

类似于根据工作负载组件的功能用途对网络层进行建模，此时同样需要考虑所处理数据的敏感性。以 Web 应用程序为例，虽然您的所有工作负载服务可能都位于应用层内，但不同的服务会处理具有不同敏感性级别的数据。在这种情况下，根据您的控制要求，您可能需要针对不同的数据敏感性来划分应用层，例如使用多个私有子网、同一个 AWS 账户 中的不同 VPC，甚至是不同 AWS 账户 中的不同 VPC。

网络层的另一个注意事项是工作负载组件的行为一致性。仍旧以上例来说明，在应用层中，您的服务可能接受来自最终用户或外部系统集成的输入，这些输入本质上比其它服务的输入风险更大。例如文件上传、要运行的代码脚本、电子邮件扫描等。将这些服务放在各自的网络层中，可以在这些服务周围建立更可靠的隔离边界，并可以防止它们的独特行为在检查系统中产生误报提醒。

在设计过程中，请考虑使用 AWS 托管服务会对您的网络拓扑造成什么影响。探索 Amazon VPC Lattice 等服务如何有助于简化跨网络层的工作负载组件互操作性。使用 AWS Lambda 时，除非有特殊

的原因，否则应将该服务部署在您的 VPC 子网中。对于限制访问互联网网关的安全策略，确定如何利用 VPC 端点和 AWS PrivateLink 来简化遵守这些策略所需的工作。

实施步骤

1. 查看您的工作负载架构。根据组件和服务提供的功能、所处理数据的敏感性及其行为，对组件和服务进行逻辑分组。
2. 如果组件需要响应来自互联网的请求，请考虑使用负载均衡器或其它代理来提供公有端点。探索通过使用 CloudFront、Amazon API Gateway、弹性负载均衡和 AWS Amplify 等托管服务来托管公有端点，从而转变安全控制模式。
3. 对于在计算环境中运行的组件，例如 Amazon EC2 实例、AWS Fargate 容器或 Lambda 函数，请根据您在第一步中的分组，将它们部署到私有子网中。
4. 对于完全托管式 AWS 服务，例如 Amazon DynamoDB、Amazon Kinesis 或 Amazon SQS，请考虑默认使用 VPC 端点来通过私有 IP 地址进行访问。

资源

相关最佳实践：

- REL02 计划网络拓扑
- PERF04-BP01 了解联网对性能的影响

相关视频：

- AWS re:Invent 2023 – AWS networking foundations

相关示例：

- VPC 示例
- 使用 AWS Fargate、AWS PrivateLink，和网络负载均衡器在 Amazon ECS 上私密访问容器应用程序
- 使用 Amazon CloudFront 通过 VPC 在 Amazon S3 存储桶中提供静态内容

SEC05-BP02 控制网络层中的流量流动

在网络的各层中，进一步对网络进行分段，从而将流量限制在对各个工作负载所必要的流动路径中。首先，将重点放在控制互联网或其它外部系统与工作负载和您环境之间的流量（南北向流量）上。然后，审查不同组件与系统之间的流量（东西向流量）。

期望结果：您只允许必要的网络流量流动，以便让工作负载的组件彼此通信，以及与其客户端和所依赖的任何其它服务进行通信。您在设计时充分考虑了各种因素，例如公有入口和出口与私有入口和出口的对比、数据分类、区域法规，以及协议要求。作为最低权限原则设计的一部分，您尽可能地使用点对点流动，而不是网络对等连接。

常见反模式：

- 您采用基于边界的方法来保护网络，并且仅在网络层的边界控制流量流动。
- 您假设一个网络层中的所有流量都经过了身份验证和授权。
- 您对入口流量或出口流量进行了控制，但没有对两者均进行控制。
- 您完全依靠工作负载组件和网络控制措施来对流量进行身份验证和授权。

建立此最佳实践的好处：这种做法有助于减少网络中未经授权移动的风险，并为您的工作负载增加了额外的授权层。通过执行流量流动控制，您可以限制安全事件的影响范围，同时加快检测和响应的速度。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

虽然通过网络层，您可以围绕工作负载中具有相似功能、数据敏感性级别和行为的组件来划定边界，不过您可以利用各种技术，在这些层中遵循最低权限原则来进一步划分组件，从而创建更精细的流量控制级别。在 AWS 中，网络层主要根据 Amazon VPC 中的 IP 地址范围使用子网进行定义。您也可以使用不同的 VPC 来定义层，例如按业务域来对微服务环境进行分组。使用多个 VPC 时，使用 AWS Transit Gateway 来仲裁路由。虽然这种方法使用安全组和路由表在第 4 层上（IP 地址和端口范围）提供流量控制，不过您可以使用其它服务（例如 AWS PrivateLink、Amazon Route 53 Resolver DNS 防火墙、AWS Network Firewall 和 AWS WAF）实现进一步的控制。

了解工作负载的数据流动和通信要求并列出清单，这涉及到连接发起方、端口、协议和网络层等。评估可用于建立连接和传输数据的协议，从而选择符合您的保护要求的协议（例如，HTTPS 而不是 HTTP）。在网络边界和每个层中收集这些要求。确定这些要求后，探索仅允许所需流量流经每个连接点的选项。作为开始，一种很好的方法是在 VPC 中使用安全组，因为安全组可以连接到使用弹性网络接口（ENI）的资源，例如 Amazon EC2 实例、Amazon ECS 任务、Amazon EKS 容器组（pod）或

Amazon RDS 数据库。与第 4 层防火墙不同，安全组可以设定一条规则，按照标识符来允许其它安全组的流量，这样当组中的资源随着时间发生变化时，能尽可能减少更新工作。您还可以使用安全组，通过入站和出站规则来筛选流量。

当流量在 VPC 之间移动时，对于简单路由通常会使用 VPC 对等连接，对于复杂路由则使用 AWS Transit Gateway。使用这些方法，您可以在源网络和目标网络的 IP 地址范围之间，协调流量流动。但是，如果您的工作负载只需要在不同 VPC 中的特定组件之间的流量流动，请考虑使用 AWS PrivateLink 进行点对点连接。为此，请确定哪些服务应充当产生器，哪些服务应充当使用器。为产生器部署兼容的负载均衡器，相应地启用 PrivateLink，然后接受使用器的连接请求。接下来，向产生器服务分配来自使用器 VPC 的私有 IP 地址，使用器可以使用该地址发出后续请求。这种方法减少了对网络对等连接的需求。评估 PrivateLink 时，请包括数据处理和负载均衡的成本。

虽然安全组和 PrivateLink 均可用于控制工作负载的组件之间的流量流动，不过还有另一个重要考虑因素，即如何控制允许您的资源访问哪些 DNS 域（如果有）。根据 VPC 的 DHCP 配置，您可以考虑使用两种不同的 AWS 服务来实现此目的。大多数客户使用默认的 Route 53 Resolver DNS 服务（也称为 Amazon DNS 服务器或 AmazonProvidedDNS），该服务可用于 VPC，地址为其 CIDR 范围 +2。通过这种方法，您可以创建 DNS 防火墙规则，然后将规则关联到 VPC，用来确定对您提供的域列表采取哪些操作。

如果您不使用 Route 53 Resolver，或者想在域筛选之外，利用更深入的检查和流量控制功能来补充 Resolver，请考虑部署 AWS Network Firewall。此服务使用无状态或有状态规则检查单独的数据包，以确定是拒绝还是允许流量。您可以使用类似的方法，通过 AWS WAF 来筛选流向公有端点的入站 Web 流量。有关这些服务的更多指导信息，请参阅《SEC05-BP03 实施基于检查的保护》。

实施步骤

1. 确定在工作负载的组件之间必需的数据流。
2. 对于入站和出站流量，采用深度防御方法应用多种控制措施，包括使用安全组和路由表。
3. 针对出入 VPC 和 VPC 之间的网络流量，使用防火墙定义精细的控制措施，例如 Route 53 Resolver DNS 防火墙、AWS Network Firewall 和 AWS WAF。考虑使用 AWS Firewall Manager 来集中配置和管理整个企业的防火墙规则。

资源

相关最佳实践：

- REL03-BP01 选择如何划分工作负载
- SEC09-BP02 在传输中执行加密

相关文档：

- VPC 的安全最佳实践
- AWS Network Optimization Tips
- Guidance for Network Security on AWS
- Secure your VPC's outbound network traffic in the AWS 云

相关工具：

- AWS Firewall Manager

相关视频：

- AWS Transit Gateway reference architectures for many VPCs
- Application Acceleration and Protection with Amazon CloudFront, AWS WAF, and AWS Shield
- AWS re:Inforce 2023: Firewalls and where to put them

SEC05-BP03 实施基于检查的保护

在各网络层之间设置流量检测点，确保传输中数据符合预期的类别和模式。 分析流量、元数据和模式，以便于更有效地识别、检测和响应事件。

期望结果：在各网络层之间穿行的流量均经过检查和授权。 允许和拒绝的决定基于明确的规则、威胁情报和偏离基线的行为。 流量越接近敏感数据，保护措施就越严格。

常见反模式：

- 仅依赖基于端口和协议的防火墙规则，而不利用智能系统。
- 根据当前可能发生变化的特定威胁模式制定防火墙规则。
- 只检查从私有子网传输到公有子网或从公有子网传输到互联网的流量。
- 没有网络流量基线视图，无法与异常行为对照。

建立此最佳实践的好处：检测系统允许您制定智能规则，例如仅当流量数据中存在特定条件时才允许或拒绝流量。根据最新的威胁情报，从 AWS 和合作伙伴提供的托管规则集获益，因为威胁状况会随着时间的推移而发生变化。 这减少了维护规则和研究折衷指标的开销，降低了误报的可能性。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

使用 AWS Network Firewall 或 AWS Marketplace 上其它可部署在网关负载均衡器（GWLB）后面的防火墙和入侵防御系统（IPS，Intrusion Prevention Systems），对有状态和无状态网络流量进行细粒度控制。AWS Network Firewall 支持与 Suricata 兼容的开源 IPS 规范，有助于保护您的工作负载。

AWS Network Firewall 和使用 GWLB 的供应商解决方案都支持不同的内联检查部署模型。 例如，您可以逐个 VPC 执行检查，在所检查 VPC 内集中进行检查，或者以混合模式进行部署，即东西向流量流经检查 VPC，而互联网入口则逐个 VPC 进行检查。 另一个考虑因素是，解决方案是否支持解包传输层安全性协议（TLS），从而能够对任一方向启动的流量进行深度数据包检查。有关这些配置的更多信息和深入细节，请参阅《AWS Network Firewall 最佳实践指南》。

如果您使用的是执行带外检查的解决方案，例如对来自以混杂模式运行的网络接口的数据包数据进行 pcap 分析，则可以配置 VPC 流量镜像。镜像流量计入接口的可用带宽，与非镜像流量收取相同的数据传输费用。您可以查看 AWS Marketplace 上是否有这些设备的虚拟版本，它们可能支持在 GWLB 后面进行内联部署。

对于通过基于 HTTP 的协议进行事务处理的组件，应使用 Web 应用程序防火墙（WAF，Web Application Firewall）保护应用程序免受常见威胁。AWS WAF 是一种 Web 应用程序防火墙，可让您在将符合可配置规则的 HTTP(S) 请求发送到 Amazon API Gateway、Amazon CloudFront、AWS AppSync 或应用程序负载均衡器之前，监控并阻止这些请求。在评估 Web 应用程序防火墙的部署时，可以考虑深度数据包检查，因为有些防火墙要求在流量检查前终止 TLS。要开始使用 AWS WAF，您可以将 AWS 托管式规则 与自己的规则结合使用，也可以使用现有的合作伙伴集成。

您可以使用 AWS Firewall Manager，在整个 AWS 组织内集中管理 AWS WAF、AWS Shield Advanced、AWS Network Firewall 和 Amazon VPC 安全组。

实施步骤

1. 确定是可以通过检查 VPC 等方式宽泛地确定检查规则的范围，还是需要更细粒度的针对每个 VPC 的方法。
2. 对于内联检查解决方案：
   a. 如果使用 AWS Network Firewall，则创建规则、防火墙策略和防火墙本身。配置完这些后，就可以将流量路由到防火墙端点以便启用检查。
   b. 如果使用带有网关负载均衡器（GWLB）的第三方设备，请在一个或多个可用区内部署和配置设备。然后，创建 GWLB、端点服务、端点，并为流量配置路由。

3. 对于带外检查解决方案：
    1. 在应该对入站和出站流量进行镜像的接口上，启用 VPC 流量镜像功能。您可以使用 Amazon EventBridge 规则调用 AWS Lambda 函数，以便在创建新资源时在接口上启用流量镜像功能。将流量镜像会话指向在设备前面处理流量的网络负载均衡器。
4. 对于入站 Web 流量解决方案：
    a. 要配置 AWS WAF，首先要配置 Web 访问控制列表（Web ACL）。Web ACL 是众多规则的集合，具有连续处理的默认操作（ALLOW 或 DENY），可定义 WAF 如何处理流量。您可以创建自己的规则和组，也可以在 Web ACL 中使用 AWS 托管规则组。
    b. 配置好 Web ACL 后，将 Web ACL 与 AWS 资源（如应用程序负载均衡器、API Gateway REST API 或 CloudFront 分配）关联，即可开始保护 Web 流量。

资源

相关文档：

- What is Traffic Mirroring?
- Implementing inline traffic inspection using third-party security appliances
- AWS Network Firewall example architectures with routing
- Centralized inspection architecture with AWS Gateway Load Balancer and AWS Transit Gateway

相关示例：

- 部署网关负载均衡器的最佳实践
- TLS inspection configuration for encrypted egress traffic and AWS Network Firewall

相关工具：

- AWS Marketplace IDS/IPS

SEC05-BP04 自动执行网络保护

使用 DevOps 实践 [例如基础设施即代码（IaC）和 CI/CD 管道] 自动部署网络保护。 这些实践有助于您通过版本控制系统跟踪网络保护措施的变更，缩短部署变更所需的时间，并有助于检测网络保护措施是否偏离了您所需的配置。

期望结果：您可以使用模板来定义网络保护，并将模板提交到版本控制系统中。 当有新的变更时，自动管道就会启动，协调这些变更的测试和部署。 进行策略检查和其它静态测试，以便在部署之前验证变更。 您可以将变更部署到暂存环境中，以便验证控制措施是否按预期运行。 一旦控制措施获得批准，还可自动部署到生产环境中。

常见反模式：

- 依靠各个工作负载团队各自定义完整的网络堆栈、保护措施和自动化。 不集中发布网络堆栈和保护措施的标准内容，供工作负载团队使用。
- 依靠中央网络团队来定义网络、保护措施和自动化的所有方面。 不将网络堆栈和保护措施的特定工作负载方面委托给该工作负载的团队。
- 在网络团队和工作负载团队之间的集中化和委托之间取得适当平衡，但不在 IaC 模板和 CI/CD 管道中应用一致的测试和部署标准。 没有在检查模板是否符合要求的工具中捕获所需的配置。

建立此最佳实践的好处：使用模板来定义网络保护，可以通过版本控制系统跟踪和比较随时间发生的变更。 使用自动化功能来测试和部署变更，可实现标准化和可预测性，增加成功部署的机会，减少重复的手动配置。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

SEC05-BP02 控制网络层中的流量流动和 SEC05-BP03 实施基于检查的保护中描述的许多网络保护控制措施都带有可根据最新威胁情报自动更新的托管规则系统。 保护 Web 端点的示例包括 AWS WAF 托管规则和 AWS Shield Advanced 自动应用层 DDoS 缓解。使用 AWS Network Firewall 托管规则组，也可随时更新低声誉域列表和威胁特征。

除了托管规则外，我们还建议您使用 DevOps 实践来自动部署网络资源、保护措施和您指定的规则。 您可以在 AWS CloudFormation 或您选择的其它基础设施即代码（IaC）工具中捕获这些定义，将其提交到版本控制系统，并使用 CI/CD 管道进行部署。 使用这种方法可获得 DevOps 在管理网络控制方面的传统优势，如更具可预测性的发布、使用 AWS CloudFormation Guard 等工具进行自动测试，以及检测已部署环境与所需配置之间的偏差等。

根据您在 SEC05-BP01 创建网络层中做出的决定，您可以采用集中管理方法创建 VPC，专用于入口、出口和检查流。 如 AWS Security Reference Architecture（AWS SRA）所述，您可以在专用网络基础设施账户中定义这些 VPC。 您可以使用类似的技术，来集中定义其它账户中工作负载使用的 VPC、其安全组、AWS Network Firewall 部署、Route 53 Resolver 规则和 DNS Firewall 配置以及其它网络资源。 您可以通过 AWS Resource Access Manager 与其它账户共享这些资源。 通过这种方法，您可以

将网络控制的自动测试和部署简化到网络账户中，只需管理一个目标即可。 您可以采用混合模式来实现这一点，即集中部署和共享某些控制措施，并将其它控制措施委托给各个工作负载团队及其各自的账户。

实施步骤

1. 确定网络和保护措施的哪些方面是集中定义的，哪些是工作负载团队可以维护的。
2. 创建环境来测试和部署对网络及其保护措施的变更。 例如，使用“网络测试”账户和“网络生产”账户。
3. 确定如何在版本控制系统中存储和维护模板。 将中央模板存储在有别于工作负载存储库的存储库中，而工作负载模板可存储在特定于该工作负载的存储库中。
4. 创建 CI/CD 管道来测试和部署模板。 定义测试方法，用于检查配置是否有误，以及模板是否符合公司标准。

资源

相关最佳实践：

- SEC01-BP06 自动部署标准安全控制措施

相关文档：

- AWS Security Reference Architecture - Network account

相关示例：

- AWS Deployment Pipeline Reference Architecture
- NetDevSecOps to modernize AWS networking deployments
- Integrating AWS CloudFormation security tests with AWS Security Hub CSPM and AWS CodeBuild reports

相关工具：

- AWS CloudFormation
- AWS CloudFormation Guard
- cfn_nag

保护计算

计算资源包括 EC2 实例、容器、AWS Lambda 函数、数据库服务、物联网设备等。每种计算资源类型都需要不同的保护方法。不过，确实存在需要考虑的共同策略：深度防御、管理漏洞、缩小攻击面、自动执行配置和操作以及远距离执行操作。在本节提供一般指导，可用于保护关键服务的计算资源。对于所使用的每项 AWS 服务，务必查看相应服务文档中的具体安全建议。

最佳实践

- SEC06-BP01 执行漏洞管理
- SEC06-BP02 从强化映像预置计算
- SEC06-BP03 减少人工管理工作和交互式访问
- SEC06-BP04 验证软件完整性
- SEC06-BP05 自动保护计算

SEC06-BP01 执行漏洞管理

频繁扫描和修补您的代码、依赖项和基础设施中的漏洞，以帮助防御新的威胁。

期望结果：您的解决方案可以持续扫描工作负载，来发现软件漏洞、潜在缺陷和意外的网络泄露。您已经制定了流程和过程，可以根据风险评测标准来识别这些漏洞、确定其优先级并对其进行修复。此外，您还为计算实例实施了自动补丁管理。您的漏洞管理程序已集成到软件开发生命周期中，并提供了在 CI/CD 管道期间扫描源代码的解决方案。

常见反模式：

- 未制定漏洞管理计划。
- 在不考虑严重性或风险规避的情况下执行系统修补。
- 使用已超过供应商提供的生命周期结束（EOL）日期的软件。
- 在分析安全问题之前，将代码部署到生产环境中。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

漏洞管理是维护安全且稳健的云环境的一个关键环节。它涉及一个全面的流程，包括安全扫描、问题的识别和优先级排序，以及用于修复已识别漏洞的修补操作。自动化在此流程中起着举足轻重的作用，因为它有助于对工作负载进行持续扫描来发现潜在问题和意外的网络泄露，以及实施修复工作。

AWS 责任共担模式是支撑漏洞管理的基本概念。根据该模式，AWS 负责保护底层基础设施的安全，包括运行 AWS 服务的硬件、软件、网络和设施。相反，您负责保护与 Amazon EC2 实例和 Amazon S3 对象等服务关联的数据、安全配置和管理任务。

AWS 提供一系列服务来支持漏洞管理计划。Amazon Inspector 持续扫描 AWS 工作负载中是否存在软件漏洞和意外网络访问，而 AWS Systems Manager 补丁管理器则有助于管理跨 Amazon EC2 实例的修补工作。这些服务可以与 AWS Security Hub CSPM 这一云安全态势管理服务集成，该服务可自动执行 AWS 安全检查，集中安全警报，并提供组织安全态势的全面视图。此外，Amazon CodeGuru 安全防御工具使用静态代码分析，来识别 Java 和 Python 应用程序在开发阶段期间的潜在问题。

通过将漏洞管理实践纳入软件开发生命周期，您可以在漏洞引入生产环境之前主动解决漏洞，从而降低安全事件的风险，并最大限度地减少漏洞的潜在影响。

实施步骤

1. 了解责任共担模式：查看 AWS 责任共担模式，来了解您在云端保护工作负载和数据的责任。AWS 负责保护底层云基础设施，而您负责保护您的应用程序、数据和所使用的服务。
2. 实施漏洞扫描：配置漏洞扫描服务（例如 Amazon Inspector），以便自动扫描计算实例（例如虚拟机、容器或无服务器函数），来查找软件漏洞、潜在缺陷和意外的网络泄露。
3. 建立漏洞管理流程：定义用于识别漏洞、确定漏洞优先级和修复漏洞的流程和过程。这可能包括制定定期漏洞扫描计划、建立风险评估标准以及根据漏洞严重程度定义修补时间表。
4. 设置补丁管理：使用补丁管理服务，来自动执行为操作系统和应用程序修补计算实例的过程。您可以将服务配置为扫描实例中缺少的补丁，并按计划自动安装这些补丁。可以考虑使用 AWS Systems Manager 补丁管理器来提供此功能。
5. 配置恶意软件防护：实施相应的机制来检测环境中的恶意软件。例如，可以使用诸如 Amazon GuardDuty 之类的工具来分析、检测 EC2 和 EBS 卷中的恶意软件并发出警报。GuardDuty 还可以扫描新上传到 Amazon S3 的对象中是否存在潜在的恶意软件或病毒，并在它们被摄取到下游进程之前采取措施将其隔离。
6. 在 CI/CD 管道中集成漏洞扫描：如果您使用 CI/CD 管道进行应用程序部署，请将漏洞扫描工具集成到您的管道中。诸如 Amazon CodeGuru 安全防御工具和开源选项之类的工具可以扫描源代码、依赖项和构件，来发现潜在的安全问题。
7. 配置安全监控服务：设置安全监控服务（例如 AWS Security Hub CSPM），来全面了解您在多个云服务中的安全状况。该服务应从各种来源收集安全调查发现，并以标准化格式呈现它们，以便于确定优先级和进行补救。
8. 实施 Web 应用程序渗透测试：如果您的应用程序是 Web 应用程序，并且您的组织具有必要的技能或可以聘请外部协助，请考虑实施 Web 应用程序渗透测试，以识别应用程序中的潜在漏洞。

9. 利用基础设施即代码实现自动化：使用基础设施即代码（IaC）工具（例如 AWS CloudFormation）来自动部署和配置资源，包括前面提到的安全服务。这种做法有助于您在多个账户和环境中创建更加一致和标准化的资源架构。
10. 监控并持续改进：持续监控漏洞管理计划的有效性，并根据需要进行改进。审核安全调查发现，评测补救工作的有效性，并相应地调整您的流程和工具。

资源

相关文档：

- AWS Systems Manager
- AWS Lambda 安全性概述
- Amazon CodeGuru
- 使用新的 Amazon Inspector 改进了云工作负载的自动化漏洞管理
- 使用 Amazon Inspector 和 AWS Systems Manager 自动执行 AWS 中的漏洞管理和修复 – 第 1 部分

相关视频：

- 保护无服务器和容器服务
- Security best practices for the Amazon EC2 instance metadata service

SEC06-BP02 从强化映像预置计算

通过从强化映像部署运行时环境，减少意外访问运行时环境的机会。只从可信注册表获取运行时依赖项（例如容器映像和应用程序库），并验证其签名。创建自己的专用注册表来存储可信映像和库，供构建和部署流程使用。

期望结果：您的计算资源是从强化的基准映像预置的。您只从可信注册表检索外部依赖项（例如容器映像和应用程序库），并验证其签名。这些依赖项存储在专用注册表中，供构建和部署流程参考。您会定期扫描和更新映像和依赖项，以便于应对任何新发现的漏洞。

常见反模式：

- 从可信注册表获取映像和库，但在投入使用前不验证其签名或进行漏洞扫描。
- 强化映像，但没有定期测试映像是否存在新漏洞或更新到最新版本。
- 安装或不删除在映像预期生命周期内不需要的软件包。

- 仅依靠打补丁来保持生产计算资源的最新状态。随着时间的推移，仅靠打补丁仍会导致计算资源偏离强化标准。打补丁也可能无法清除威胁行为者在安全事件中安装的恶意软件。

建立此最佳实践的好处：强化映像有助于减少在运行时环境中，可能允许未经授权的用户或服务进行意外访问的路径数量。如果发生任何意外访问，强化映像还可以缩小影响范围。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

要强化系统，请从最新版本的操作系统、容器映像和应用程序库开始。应用补丁以解决已知问题。删除所有不需要的应用程序、服务、设备驱动程序、默认用户和其它凭证，尽量减小系统。采取其它任何必要操作，例如禁用端口，以便创建一个只拥有工作负载所需资源和功能的环境。在此基础上，您可以安装必要的软件、代理或其它进程，以满足监控工作负载或管理漏洞等目的的需要。

您可以遵循可信来源提供的指导，如互联网安全中心（CIS，Center for Internet Security）以及美国国防信息系统局（DISA，Defense Information Systems Agency）的安全技术实施指南（STIG），从而减轻强化系统的负担。我们建议您从 AWS 或 APN 合作伙伴发布的亚马逊机器映像（AMI）开始，并使用 AWS EC2 Image Builder，以期综合使用 CIS 和 STIG 控制措施来自动配置。

虽然有可用的强化映像和 EC2 Image Builder 配方可应用 CIS 或 DISA STIG 建议，但您可能会发现，它们的配置会阻止您的软件成功运行。在这种情况下，您可以从未经强化的基础映像开始，安装软件，然后逐步应用 CIS 控制措施来测试其影响。对于任何阻止软件运行的 CIS 控制措施，请测试是否可以改为在 DISA 中实施更精细的强化建议。跟踪您能够成功应用的不同 CIS 控制措施和 DISA STIG 配置。在 EC2 Image Builder 中使用这些控制措施和配置，相应地定义映像强化配方。

对于容器化工作负载，Amazon Elastic Container Registry（ECR）公共存储库提供 Docker 的强化映像。您可以结合使用 EC2 Image Builder 与 AMI 来强化容器映像。

与操作系统和容器映像类似，您可以通过 pip、npm、Maven 和 NuGet 等工具，从公共存储库中获取代码包（或库）。我们建议您将私有存储库（例如在 AWS CodeArtifact 中）与可信的公共存储库进行集成，来管理代码包。这种集成可为您处理代码包的检索、存储和保持最新状态。然后，您的应用程序构建流程就可以使用一些技术 [例如软件组成分析（SCA，Software Composition Analysis）、静态应用程序安全测试（SAST，Static Application Security Testing）和动态应用程序安全测试（DAST，Dynamic Application Security Testing）等]，与您的应用程序一起获取和测试这些代码包的最新版本。

对于使用 AWS Lambda 的无服务器工作负载，可使用 Lambda 层简化对代码包依赖项的管理。使用 Lambda 层将不同函数之间共享的一组标准依赖项配置到独立的存档中。您可以通过自己的构建流程来创建和维护层，从而能够以集中方式使您的函数保持最新状态。

实施步骤

- 强化操作系统。使用来自可信来源的基础映像为基础，来构建强化的 AMI。使用 EC2 Image Builder 来帮助自定义安装在映像上的软件。
- 强化容器化资源。配置容器化资源以符合安全最佳实践。当使用容器时，在您的构建管道中对您的映像存储库定期实施 ECR 映像扫描，以便在您的容器中查找 CVE。
- 在使用 AWS Lambda 实现无服务器时，请使用 Lambda 层来隔离应用程序函数代码和共享的依赖项库。为 Lambda 配置代码签名，以便确保只有可信代码才能在您的 Lambda 函数中运行。

资源

相关最佳实践：

- OPS05-BP05 执行补丁管理

相关视频：

- Deep dive into AWS Lambda security

相关示例：

- Quickly build STIG-compliant AMI using EC2 Image Builder
- Building better container images
- Using Lambda layers to simplify your development process
- Develop & Deploy AWS Lambda Layers using Serverless Framework
- Building end-to-end AWS DevSecOps CI/CD pipeline with open source SCA, SAST and DAST tools

SEC06-BP03 减少人工管理工作和交互式访问

尽可能使用自动化方式来执行部署、配置、维护和调查任务。在紧急程序或安全（沙盒）环境中，如果自动化不可用，可以考虑手动访问计算资源。

期望结果：程序化脚本和自动化文档（运行手册）可捕获计算资源上的授权操作。这些运行手册可以通过变更检测系统自动启动，也可以在需要人工判断时手动启动。只有在无法实现自动化的紧急情况下，才允许直接访问计算资源。所有手动活动都会被记录下来并纳入审查流程，以便不断提高自动化能力。

常见反模式：

- 使用 SSH 或 RDP 等协议对 Amazon EC2 实例进行交互式访问。
- 维护个人用户登录信息，例如 `/etc/passwd` 或 Windows 本地用户。
- 多个用户共用一个密码或私钥来访问实例。
- 手动安装软件，手动创建或更新配置文件。
- 手动更新或修补软件。
- 登录实例来解决问题。

建立此最佳实践的好处：自动执行操作有助于降低意外更改和错误配置的操作风险。避免使用 Secure Shell（SSH）和远程桌面协议（RDP，Remote Desktop Protocol）进行交互式访问，可缩小计算资源的访问范围。这样可以消除一种执行未经授权操作的常见方式。可以在自动化文档和程序化脚本中捕获计算资源管理任务，这种机制以细粒度的方式定义和审计授权活动的全部范围。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

登录到实例上是一种传统的系统管理方法。安装服务器操作系统后，用户通常会手动登录，以便配置系统并安装所需的软件。在服务器的生命周期内，用户可能会登录服务器来更新软件、应用补丁、更改配置和解决问题。

然而，手动访问会带来一些风险。这需要一个能监听请求（如 SSH 或 RDP 服务）的服务器，这就可能为未经授权的访问提供潜在的路径。这还增加了与执行手动措施相关的人为出错风险。这些操作可能导致工作负载事件、数据损坏或毁坏或者其它安全问题。人工访问还需要防止共享凭证，从而增加了管理开销。

为了降低这些风险，您可以实施基于代理的远程访问解决方案，例如 AWS Systems Manager。AWS Systems ManagerAgent（SSM Agent）会启动一个加密通道，因此它不依赖于监听外部发起的请求。考虑配置 SSM Agent 以便通过 VPC 端点建立此通道。

利用 Systems Manager 可以精细控制您与托管实例进行交互的方式。您可以定义要运行的自动化操作、谁可以运行以及何时运行。Systems Manager 可以打补丁、安装软件和更改配置，而无需与实例进行交互式访问。Systems Manager 还可提供对远程 Shell 的访问，并将会话期间调用的每条命令及其输出记录到日志和 Amazon S3 中。AWS CloudTrail 会记录对 Systems Manager API 的调用，以供检查之用。

实施步骤

1. 在 Amazon EC2 实例上安装 AWS Systems Manager Agent（SSM Agent）。检查 SSM Agent 是否包含在基本 AMI 配置中并能够自动启动。
2. 验证与 EC2 实例配置文件相关联的 IAM 角色是否包含 `AmazonSSMManagedInstanceCore` 托管 IAM 策略。
3. 禁止在实例上运行 SSH、RDP 和其它远程访问服务。为此，您可以运行在启动模板的用户数据部分内配置的脚本，或者使用 EC2 Image Builder 等工具构建自定义 AMI。
4. 确保适用于 EC2 实例的安全组入口规则不允许访问端口 22/tcp（SSH）或端口 3389/tcp（RDP）。使用 AWS Config 等服务对配置错误的安全组实施检测和提醒。
5. 在 Systems Manager 中定义适当的自动化操作、运行手册和运行命令。使用 IAM 策略来定义谁可以执行这些操作以及允许执行这些操作的条件。请在非生产环境中彻底测试这些自动化操作。请尽可能调用这些自动化操作，而不是以交互方式访问实例。
6. 必要时，使用 AWS Systems Manager Session Manager 提供对实例的交互式访问。启用会话活动日志记录，以便在 Amazon CloudWatch Logs 或 Amazon S3 中保留审计跟踪记录。

资源

相关最佳实践：

- REL08-BP04 使用不可变基础设施进行部署

相关示例：

- Replacing SSH access to reduce management and security overhead with AWS Systems Manager

相关工具：

- AWS Systems Manager

相关视频：

- Controlling User Session Access to Instances in AWS Systems Manager Session Manager

SEC06-BP04 验证软件完整性

使用加密验证来验证工作负载使用的软件构件（包括映像）的完整性。 对软件进行加密签名，以防在计算环境中出现未经授权的更改。

期望结果：所有构件均从可信来源获得。供应商网站证书已通过验证。 下载的构件通过其签名进行加密验证。您自己的软件经过加密签名，并由您的计算环境进行验证。

常见反模式：

- 信任信誉良好的供应商网站，从中获取软件构件，但忽视证书过期通知。 在未确认证书有效的情况下就继续下载。
- 验证供应商网站证书，但是从这些网站下载的构件没有进行加密验证。
- 仅依靠摘要或哈希值来验证软件的完整性。 哈希值可用于确定构件未在原始版本的基础上进行修改，但不能证实其来源正确。
- 不签署您自己的软件、代码或库，即使它们仅用于自己的部署。

建立此最佳实践的好处：验证工作负载所依赖的构件是否完整，这有助于防止恶意软件进入计算环境。 对软件进行签名有助于防止未经授权的软件在计算环境中运行。 通过签署和验证代码，保护软件供应链。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

操作系统映像、容器映像和代码构件通常在分发时提供完整性检查，例如通过摘要或哈希值进行检查。 这样，客户端就可以通过计算自己的有效负载哈希值，并验证哈希值与发布的哈希值是否相同，来验证完整性。 虽然这些检查有助于验证有效负载是否未被篡改，但并不能证实有效负载来自原始来源（数据出处）。 验证数据出处时，需要有可信机构签发的证书对构件进行了数字签名。

如果在工作负载中使用下载的软件或构件，请检查提供商是否提供了用于验证数字签名的公钥。 以下这些示例说明 AWS 如何为我们发布的软件提供公钥和验证说明：

- EC2 Image Builder: Verify the signature of the AWSTOE installation download
- AWS Systems Manager：验证 SSM Agent 签名
- Amazon CloudWatch：验证 CloudWatch 代理软件包的签名

将数字签名验证过程纳入您用于获取和强化映像的流程中，如 SEC06-BP02 从强化映像预置计算中所述。

您可以使用 AWS Signer 来协助管理签名验证过程，以及您自己的软件和构件的代码签名生命周期。 AWS Lambda 和 Amazon Elastic Container Registry 均实现了与 Signer 的集成，能够验证代码和映像的签名。 您可以参考“资源”部分中的示例，将 Signer 纳入持续集成和持续交付（CI/CD，Continuous Integration and Delivery）管道，以便自动验证签名并签署自己的代码和映像。

资源

相关文档：

- Cryptographic Signing for Containers
- Best Practices to help secure your container image build pipeline by using AWS Signer
- Announcing Container Image Signing with AWS Signer and Amazon EKS
- 为 AWS Lambda 配置代码签名
- Best practices and advanced patterns for Lambda code signing
- Code signing using AWS Certificate Manager Private CA and AWS Key Management Service asymmetric keys

相关示例：

- Automate Lambda code signing with Amazon CodeCatalyst and AWS Signer
- Signing and Validating OCI Artifacts with AWS Signer

相关工具：

- AWS Lambda
- AWS Signer
- AWS Certificate Manager
- AWS Key Management Service
- AWS CodeArtifact

SEC06-BP05 自动保护计算

自动执行保护计算的操作，减少人工干预的需要。使用自动扫描检测计算资源中是否可能存在问题，并通过自动程序化响应或实例集管理操作进行修复。 在您的 CI/CD 流程中融入自动化功能，以最新的依赖项来部署值得信赖的工作负载。

期望结果：由自动化系统对计算资源进行所有扫描和修补。您可以使用自动验证来检查软件映像和依赖项是否来自可信来源，以及是否被篡改。自动检查工作负载是否有最新的依赖项，并对工作负载进行签名，以便在 AWS 计算环境中建立信任。 一旦检测到不合规的资源，就会启动自动修复措施。

常见反模式：

- 遵循不可变基础设施的做法，但没有制定紧急修补或更换生产系统的解决方案。
- 使用自动化技术来修复配置错误的资源，但没有手动覆盖机制。 在某些情况下，您可能需要调整要求，并且在进行这些更改之前需要暂停自动化操作。

建立此最佳实践的好处：自动化操作可以降低未经授权访问和使用计算资源的风险。 这有助于防止错误配置对生产环境产生影响，检测错误配置，并在发生错误配置时对其进行修复。 自动化操作还有助于检测未经授权的访问和使用计算资源的情况，从而缩短响应时间。 这反过来又能够缩小问题的总体影响范围。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

您可以应用安全性支柱实践中描述的自动化操作来保护计算资源。SEC06-BP01 执行漏洞管理描述了如何在 CI/CD 管道中使用 Amazon Inspector，以及如何持续扫描运行时环境，查看是否存在已知的通用漏洞披露（CVE，Common Vulnerabilities and Exposures）。 您可以遵循自动化运行手册，使用 AWS Systems Manager 来应用补丁或者利用新映像重新部署，从而始终使用最新的软件和库来更新计算实例集。 使用这些技术可减少对人工流程和交互式访问计算资源的需求。 请参阅《SEC06-BP03 减少人工管理工作和交互式访问》，了解更多信息。

自动化操作在部署值得信赖的工作负载方面也发挥着作用，如《SEC06-BP02 从强化映像预置计算》和《SEC06-BP04 验证软件完整性》中所述。 您可以使用 EC2 Image Builder、AWS Signer、AWS CodeArtifact 和 Amazon Elastic Container Registry（ECR）等服务，来下载、验证、构造和存储经过强化和批准的映像和代码依赖项。 通过与 Inspector 配合使用，这些服务都可以在您的 CI/CD 流程中发挥作用，这样您的工作负载只有在确认其依赖项是最新的且来自可信来源时，才会进入生产阶段。 工作负载还经过签名，这样 AWS Lambda 和 Amazon Elastic Kubernetes Service（EKS）等 AWS 计算环境就能在允许工作负载运行之前，验证工作负载是否未被篡改。

除了这些预防性控制措施之外，您还可以在计算资源的检测性控制中运用自动化。 例如，AWS Security Hub CSPM 提供 NIST 800-53 Rev. 5 标准，其中包括 [EC2.8] EC2 实例应使用实例元数据服务版本 2（IMDSv2）等检查内容。 IMDSv2 使用会话验证、阻止包含 X-Forwarded-For HTTP 标头的请求，以及设置网络 TTL 为 1 等技术，来阻止来自外部来源的流量检索有关 EC2 实例的信息。Security Hub CSPM 中的这项检查可检测 EC2 实例何时使用 IMDSv1，并启动自动修复措施。参阅《SEC04-BP04 启动对不合规资源的修复》，了解有关自动化检测和修复的更多信息。

实施步骤

1. 利用 EC2 Image Builder 自动创建安全、合规且经过强化的 AMI。 您可以在基础 AWS 和 APN 合作伙伴映像中融入符合互联网安全中心（CIS，Center for Internet Security）基准或安全技术实施指南（STIG，Security Technical Implementation Guide）标准的控制措施，从而生成自己的映像。
2. 自动执行配置管理。使用配置管理服务或工具，在计算资源中自动实施和验证安全配置。
   a. 使用 AWS Config 自动执行配置管理
   b. 使用 AWS Security Hub CSPM 自动管理安全性和合规性态势
3. 自动修补或替换 Amazon Elastic Compute Cloud（Amazon EC2）实例。AWSSystems Manager Patch Manager 使用安全相关的更新和其它类型的更新自动执行修补托管实例的流程。您可以使用 Patch Manager 来应用操作系统和应用程序的补丁。
   a. AWS Systems Manager Patch Manager
4. 自动扫描计算资源以便查找通用漏洞披露（CVE，Common Vulnerabilities and Exposures），并在构建管道中嵌入安全扫描解决方案。
   a. Amazon Inspector –
   b. ECR 映像扫描
5. 考虑使用 Amazon GuardDuty 自动检测恶意软件和威胁，以便保护计算资源。在 AWS 环境中调用 AWS Lambda 函数时，GuardDuty 还可以识别出潜在问题。
   a. Amazon GuardDuty
6. 考虑采用 AWS 合作伙伴解决方案。AWS合作伙伴提供业界领先的产品，这些产品与您的本地环境中的现有控制措施等效、相同或与之集成。这些产品对现有 AWS 服务起到补充作用，使您能够在云端和本地环境中部署全面的安全架构，进而实现无缝效果更好的体验。
   a. 基础结构安全性

资源

相关最佳实践：

- SEC01-BP06 自动部署标准安全控制措施

相关文档：

- Get the full benefits of IMDSv2 and disable IMDSv1 across your AWS infrastructure

相关视频：

- Security best practices for the Amazon EC2 instance metadata service

数据保护

在为任何工作负载设计架构之前，您应确定可能影响安全性的基本实践。例如，数据分级提供了一种基于敏感程度对数据进行分类的方法，加密通过让未经授权的用户无法获知数据的真正内容来保护数据。这些方法有助于避免误操作或履行监管义务等，意义重大。

在 AWS 中，实施数据保护时可以使用很多不同的方法。以下小节旨在介绍如何使用这些方法。

主题

数据分类

数据分类提供了一种基于关键性和敏感度对组织数据进行分类的方法，有助于确定适当的保护和保留控制措施。

最佳实践

SEC07-BP01 了解数据分类方案

了解工作负载正在处理的数据的分类、数据处理要求、相关业务流程、数据存储位置以及数据所有者是谁。 您的数据分类和处理方案应考虑工作负载的适用法律和合规性要求，以及需要采取哪些数据控制措施。了解数据是数据分类之旅的第一步。

期望结果：您的工作负载中的数据类型已得到充分了解并记录在案。 根据敏感数据的分类，采取适当的控制措施来保护敏感数据。 这些控制措施要考虑的因素包括：谁被允许访问数据以及访问的目的、数据的存储位置、数据的加密策略以及加密密钥的管理方式、数据的生命周期及其留存要求、适当的销毁流程、备份和恢复流程以及访问审计。

常见反模式：

- 没有正式的数据分类策略来定义数据敏感性级别及其处理要求
- 没有充分了解工作负载中数据的敏感性级别，也没有在架构和操作文档中记录这些信息
- 未能按照数据分类和处理策略的规定，根据数据的敏感性和要求，对数据采取适当的控制措施
- 未能向策略所有者提供有关数据分类和处理要求的反馈。

建立此最佳实践的好处：这种实践消除了工作负载中数据适当处理方面的模糊性。 运用正式策略来定义组织中数据的敏感性级别及其所需的保护措施，这有助于您遵守法律法规以及其它网络安全证明和认证。 工作负载所有者可以放心地了解敏感数据的存储位置和保护控制措施。 将这些内容记录在文档中，有助于团队新成员更好地理解这些内容，并在任职初期保持控制。这些实践还有助于通过合理调整各类数据的控制措施来降低成本。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

在设计工作负载时，您可能会直观地考虑保护敏感数据的方法。 例如，在多租户应用程序中，直观的做法是将每个租户的数据视为敏感数据，并采取保护措施，使一个租户无法访问另一个租户的数据。 同样，您也可以直观地设计访问控制措施，只有管理员可以修改数据，而其他用户只有读取级访问权限，或者根本没有访问权限。

通过在策略中定义和记录这些数据敏感性级别及其数据保护要求，您可以正式确定工作负载中存在哪些数据。然后，您可以确定是否制定了正确的控制措施，是否可以对控制措施进行审计，以及在发现数据处理不当时应采取哪些适当的应对措施。

为了有助于确定敏感数据在工作负载中的位置，请考虑使用数据目录。数据目录是一个数据库，用于映射组织中的数据、其位置、敏感度级别以及为保护这些数据而设置的控制措施。此外，如果可用，请考虑使用 resource tags。 例如，对于受保护健康信息（PHI），您可以应用标签键为 `Classification`、标签值为 `PHI` 的标签，以及标签键为 `Sensitivity`、标签值为 `High` 的标签。 然后，AWS Config 等服务可用于监控这些资源的更改，并在这些资源被修改导致其不符合保护要求（例如更改加密设置）时发出警报。 您可以使用标签策略（AWS Organizations 的一项功能），来获取标签键和可接受值的标准定义。建议标签键或标签值不要包含私有或敏感数据。

实施步骤

1. 了解组织的数据分类方案和保护要求。
2. 确定工作负载处理的敏感数据的类型。
3. 在数据目录中捕获数据，数据目录可提供数据在组织中的位置以及该数据的敏感度级别的单一视图。

4. 考虑在可用的情况下使用资源和数据级标记，来标记数据的敏感性级别以及其它有助于监控和响应事件的操作元数据。
   a. AWS Organizations 标签策略可用于执行标记标准。

资源

相关最佳实践：

- SUS04-BP01 实施数据分类策略

相关文档：

- Data Classification whitepaper
- 标记 AWS 资源的最佳实践

相关示例：

- AWS Organizations Tag Policy Syntax and Examples

相关工具

- AWS 标签编辑器

SEC07-BP02 根据数据敏感性应用数据保护控制措施

应用数据保护控制措施，为分类策略中定义的每一类数据提供适当水平的控制。 这种做法可以保护敏感数据，防止在未经授权的情况下访问和使用敏感数据，同时保持数据可用。

期望结果：您有一项分类策略，它定义了组织内数据的不同敏感性级别。 对于每个敏感性级别，您都有明确的指导原则，规定了经批准的存储和处理服务、位置及其所需的配置。 您可以根据所需的保护级别和相关成本，实施每个级别的控制措施。 如果数据出现在未经授权的位置、在未经授权的环境中处理、被未经授权的行为者访问，或者相关服务的配置变得不合规，您都能够进行监控并发出警报。

常见反模式：

- 对所有数据应用相同级别的保护控制措施。这可能导致为低敏感性数据预配过多的安全控制措施，或者对高敏感性数据保护不足。

- 在定义数据保护控制措施时，没有让安全、合规和业务团队的利益相关方参与进来。
- 忽视与实施和维护数据保护控制措施相关的运营开销和成本。
- 不定期进行数据保护控制措施审查，无法保持一直符合分类策略。
- 没有有关静态数据和传输中数据所在位置的完整清单。

建立此最佳实践的好处：贵组织通过根据数据分类级别调整控制措施，能够在需要时投资更高级别的控制措施。可能包括增加用于保护、监控、测量、修复和报告的资源。 在适合减少控制措施的情况下，您可以为员工、客户或成员提高数据的可访问性和完整性。 这种方法既能让贵组织在数据使用方面获得极大的灵活性，又能遵守数据保护要求。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

根据数据敏感性级别实施数据保护控制措施时，涉及几个关键步骤。首先，确定工作负载架构中不同的数据敏感性级别（如公开、内部、保密和受限），并评估在哪里存储和处理这些数据。接下来，根据数据的敏感性级别定义数据的隔离边界。我们建议您将数据分别放置到不同的 AWS 账户 中，使用服务控制策略（SCP）来限制每个数据敏感性级别所允许的服务和操作。这样，您就可以创建强大的隔离边界，并执行最低权限原则。

定义隔离边界后，根据数据敏感性级别实施适当的保护控制措施。参考保护静态数据和保护传输中数据中的最佳实践，实施加密、访问控制和审计等相关控制措施。考虑采用令牌化或匿名化等技术，来降低数据的敏感性级别。利用集中式令牌化和去令牌化系统，简化在整个企业中应用一致数据策略的过程。

持续监控和测试所实施控制措施的有效性。随着贵组织数据状况和威胁的变化，定期审查和更新数据分类方案、风险评估和保护控制措施。使所实施的数据保护控制措施符合相关行业法规、标准和法律要求。此外，培养安全意识和提供培训，让员工了解数据分类方案及其在处理和保护敏感数据方面的责任。

实施步骤

1. 确定工作负载中数据的分类和敏感性级别。
2. 规定每个级别的隔离边界，并确定执行策略。
3. 评估您定义的控制措施，这些控制措施管理访问、加密、审计、留存以及数据分类策略所要求的其它事项。
4. 评估在适当情况下降低数据敏感性级别的方案，例如采用令牌化或匿名化。
5. 自动测试和监控已配置资源来验证控制措施。

资源

相关最佳实践：

- PERF03-BP01 使用最能满足数据访问和存储要求的专用数据存储
- COST04-BP05 执行数据留存策略

相关文档：

- Data Classification whitepaper
- Best Practices for Security, Identify, & Compliance
- AWS KMS 最佳实践
- Encryption best practices and features for AWS services

相关示例：

- Building a serverless tokenization solution to mask sensitive data
- How to use tokenization to improve data security and reduce audit scope

相关工具：

- AWS Key Management Service (AWS KMS)
- AWS CloudHSM
- AWS Organizations

SEC07-BP03 自动识别和分类

自动识别和分类数据可帮助您实施正确的控制措施。使用自动化技术来增强人工判断，可降低人为出错和暴露的风险。

期望结果：您能够根据自己的分类和处理策略，来验证是否有适当的控制措施。自动化工具和服务有助于您识别数据的敏感性级别并加以分类。 自动化技术还有助于您持续监控环境，以便检测数据是否以未经授权的方式存储或处理，并发出警报，从而能够迅速采取纠正措施。

常见反模式：

- 完全依赖人工流程进行数据识别和分类，既容易出错又耗费时间。 这可能导致数据分类效率低下且不稳定，尤其是在数据量不断增长的情况下。
- 缺乏机制，无法跟踪和管理整个组织内的数据资产。
- 忽视了数据在组织内部移动和演变时，对数据进行持续监控和分类的需求。

建立此最佳实践的好处：数据识别和分类自动化可使数据保护控制措施的应用更加稳定和准确，从而降低人为出错的风险。 自动化技术还可以提供敏感数据访问和移动操作的可见性，有助于您检测到未经授权的处理并采取纠正措施。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

在工作负载的初始设计阶段，通常运用人工判断对数据进行分类，但作为一种预防性控制机制，应考虑建立若干系统，以期对测试数据进行自动识别和分类。例如，可以向开发人员提供工具或服务对代表性数据进行扫描，以便确定数据敏感性。 在 AWS 中，您可以将数据集上传至 Amazon S3，并使用 Amazon Macie、Amazon Comprehend 或 Amazon Comprehend Medical 对数据进行扫描。 同样，考虑在单元测试和集成测试中纳入数据扫描，以便检测哪里的敏感数据不在预期之内。如果在这一阶段对敏感数据发出警报，就能够在部署到生产环境之前突出保护方面的漏洞。AWS Glue、Amazon SNS 和 Amazon CloudWatch 中的敏感数据检测等其它功能也可用于检测 PII 并采取缓解措施。对于任何自动化工具或服务，都要了解其如何定义敏感数据，并根据需要使用其它人工或自动化解决方案来解决任何漏洞。

作为一种检测性控制措施，利用对环境的持续监控来检测敏感数据是否以不合规的方式存储。 这有助于检测一些情况，例如，敏感数据是否被发送到日志文件或者复制到数据分析环境，而没有进行适当的去标识化或修订。 可使用 Amazon Macie 对存储在 Amazon S3 中的数据进行持续监控，以便发现敏感数据。

实施步骤

1. 查看 SEC07-BP01 中介绍的组织内数据分类方案。
   a. 通过了解贵组织的数据分类方案，您可以建立与公司策略相一致的准确的自动识别和分类流程。
2. 对环境进行初步扫描，以便自动识别和分类。
   a. 对数据初步进行全面扫描有助于全面了解敏感数据在环境中的位置。如果最初不需要进行全面扫描，或者由于成本原因无法提前完成扫描，则应评估数据采样技术是否适合实现您的结果。例如，可以对 Amazon Macie 进行配置，以便在 S3 存储桶中执行广泛的自动敏感数据发现操作。 该功能利用采样技术，对敏感数据的位置进行初步分析，成本效益高。 然后，可以使用敏感数据发现作业对 S3 存储桶进行更深入的分析。其它数据存储也可以导出到 S3，由 Macie 扫描。

b. 为扫描中识别的数据存储资源建立在 SEC07-BP02 中定义的访问控制。

3. 配置对环境的持续扫描。

a. Macie 的自动敏感数据发现功能可用于对环境进行持续扫描。 可使用 Macie 中的允许列表排除已授权存储敏感数据的已知 S3 存储桶。

4. 将识别和分类纳入构建和测试流程。

a. 确定开发人员可用于在开发工作负载时扫描数据敏感性的工具。 在集成测试过程中使用这些工具，在敏感数据意外出现时发出警报，并阻止继续部署。

5. 实施系统或运行手册，以便在未经授权的位置发现敏感数据时采取行动。

a. 使用自动修复功能来限制对数据的访问。例如，如果您使用基于属性的访问权限控制（ABAC），则可以将此数据移到访问受限的 S3 存储桶，或者为对象添加标签。此外，可以考虑在检测到数据时对其进行屏蔽。

b. 提醒您的数据保护和事件响应团队调查事件的根本原因。他们汲取的任何经验教训都有助于预防未来的事件。

资源

相关文档：

- AWS Glue：检测和处理敏感数据
- 在 Amazon SNS 中使用托管数据标识符
- Amazon CloudWatch Logs：通过屏蔽帮助保护敏感的日志数据

相关示例：

- Enabling data classification for Amazon RDS database with Macie
- Detecting sensitive data in DynamoDB with Macie

相关工具：

- Amazon Macie
- Amazon Comprehend
- Amazon Comprehend Medical
- AWS Glue

SEC07-BP04 定义可扩展的数据生命周期管理

了解您的数据生命周期要求，因为这些要求与不同的数据分类等级和处理方式密切相关。 这可能包括数据首次进入您的环境时的处理方式、数据转换方式以及数据销毁规则。必须考虑数据的留存期限、访问、审计和跟踪溯源等因素。

期望结果：您可以在尽可能接近摄取点和摄取时间的情况下对数据进行分类。当数据分类需要执行屏蔽、令牌化或其它降低敏感性级别的处理时，您可以在尽可能接近摄取点和摄取时间的情况下执行这些操作。

根据数据分类情况，当数据不再适合保留时，您可以按照策略删除数据。

常见反模式：

- 采用“一刀切”的数据生命周期管理方法，而不考虑不同的敏感性级别和访问权限要求。
- 仅从可用数据或备份数据的角度考虑生命周期管理，而不是两者兼顾。
- 在未确定数据价值或出处的情况下，就假定进入您工作负载的数据是有效的。
- 依赖数据持久性来替代数据备份和保护。
- 在数据超过其时效性和必要的留存期限之后，仍然保留数据。

建立此最佳实践的好处：明确定义且可扩展的数据生命周期管理策略有助于保持监管合规性，提高数据安全性，优化存储成本，并在保持适当控制的同时实现高效的数据访问和共享。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

工作负载中的数据通常是动态变化的。 数据在进入工作负载环境时所采用的形式，可能不同于数据在业务逻辑、报告、分析或机器学习中存储或使用时的形式。 此外，数据的价值可能会随着时间的推移而变化。有些数据本质上具有时效性，会随着时间的推移而失去价值。 考虑数据的这些变化对于数据分类方案和相关控制措施下的评估有何影响。 尽可能使用自动生命周期机制（例如 Amazon S3 生命周期策略和 Amazon Data Lifecycle Manager），来配置数据留存、归档和过期流程。对于存储在 DynamoDB 中的数据，可以使用生存时间（TTL）功能来定义每个项目的过期时间戳。

区分可供使用的数据和作为备份存储的数据。 考虑使用 AWS Backup 来自动备份跨 AWS 服务的数据。 Amazon EBS 快照提供了一种使用 S3 功能（包括生命周期、数据保护和访问保护机制）复制 EBS 卷并存储 EBS 卷的方法。S3 对象锁定和 AWS Backup 保管库锁就是其中的两种保护机制，可以为您的备份提供额外的安全防御和控制。管理明确的职责分工和备份访问权限。在账户级别隔离所有备份，以便在事件发生期间与受影响环境保持隔离。

生命周期管理的另一个方面是记录数据在工作负载中进展的历史，即数据溯源跟踪。该功能可以让您确信，您知道数据来自何处、执行过哪些转换、更改是由哪位所有者或流程执行的以及在何时做的更改。 掌握这些历史记录后，有助于在潜在安全事件中解决问题和进行调查。 例如，您可以在 Amazon DynamoDB 表中记录有关转换的元数据。 在数据湖中，您可以针对每个数据管道阶段，在不同的 S3 存储桶中保存转换后数据的副本。在 AWS Glue Data Catalog 中存储架构和时间戳信息。 无论采用哪种解决方案，都需要考虑最终用户的需求，以便确定报告数据溯源情况所需的适当工具。 这样有助于您确定如何以合适的方式跟踪数据溯源情况。

实施步骤

1. 分析工作负载的数据类型、敏感性级别和访问权限要求，对数据进行分类，并制定适当的生命周期管理策略。
2. 设计并实施符合法律、监管和组织要求的数据留存策略及自动销毁流程。
3. 建立流程和自动化机制，以便能够根据工作负载要求和监管的变化，持续监控、审计和调整数据生命周期管理策略、控制措施及政策。
   a. 使用 AWS Config 检测未开启自动生命周期管理的资源

资源

相关最佳实践：

- COST04-BP05 执行数据留存策略
- SUS04-BP03 使用策略管理数据集的生命周期

相关文档：

- Data Classification Whitepaper
- AWS Blueprint for Ransomware Defense
- DevOps Guidance: Improve traceability with data provenance tracking

相关示例：

- How to protect sensitive data for its entire lifecycle in AWS
- Build data lineage for data lakes using AWS Glue, Amazon Neptune, and Spline

相关工具：

- AWS Backup
- Amazon Data Lifecycle Manager
- AWS Identity and Access Management Access Analyzer

保护静态数据

静态数据代表您在工作负载期间的任意时间段内保留在非易失性存储器中的任何数据。其中包括数据块存储、对象存储、数据库、存档、IoT 设备和用来保留数据的任何其他存储介质。在实施了加密和适当的访问控制时，保护静态数据可以降低未经授权访问的风险。

加密和令牌化是两个重要但不同的数据保护方案。

令牌化是一个支持您定义令牌以表示其他敏感信息的过程（例如代表客户信用卡号的令牌）。令牌自身必须没有任何意义，而且不能是从它令牌化的数据衍生而来 – 因此，无法将加密摘要用作令牌。通过认真规划令牌化方法，您可以为内容提供额外保护，并确保满足合规性要求。例如，如果您使用令牌而不是信用卡号，就可以缩小信用卡处理系统的合规性范围。

加密可以将内容转换为这样一种形式：如果用户没有将这些内容解密为纯文本所需的密钥，就无法读取。令牌化和加密都可用于酌情保护信息。此外，可以使用掩码这种技术编辑数据的某个部分，以使剩余的数据不被视为敏感数据。例如，PCI-DSS 允许在合规性范围边界之外保留卡号的最后四位数字，以供索引使用。

审计加密密钥的使用：务必了解并审计加密密钥的使用，确保对密钥正确实施访问控制措施。例如，使用 AWS KMS 密钥的任何 AWS 服务都会在 AWS CloudTrail 中记录每次密钥使用。随后，您可以使用 Amazon CloudWatch Logs Insights 等工具查询 AWS CloudTrail，确保密钥的所有使用都有效。

最佳实践

- SEC08-BP01 实施安全密钥管理
- SEC08-BP02 强制实施静态加密
- SEC08-BP03 自动执行静态数据保护
- SEC08-BP04 强制实施访问控制

SEC08-BP01 实施安全密钥管理

安全密钥管理包括密钥材料的存储、轮换、访问控制和监控，这些都是保护工作负载的静态数据安全所必需的。

期望结果：您拥有一种可扩展、可重复且自动化的密钥管理机制。该机制对密钥材料强制实施最低权限访问，并在密钥可用性、机密性和完整性之间提供适当的平衡。您可以监控对密钥的访问权限，如果需要轮换密钥材料，则使用自动流程轮换密钥材料。您不让人工操作员访问密钥材料。

常见反模式：

- 由人类访问未加密的密钥材料。
- 创建自定义加密算法。
- 访问密钥材料的权限过于宽泛。

建立此最佳实践的好处：通过为您的工作负载建立安全的密钥管理机制，您可以帮助保护您的内容免遭未经授权的访问。此外，您可能需要遵守对数据进行加密的监管要求。有效的密钥管理解决方案可以提供符合这些法规的技术机制，进而保护密钥材料。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

静态数据加密是一项基本的安全控制措施。为实施这种控制措施，工作负载需要一种机制，来安全地存储和管理用于加密静态数据的密钥材料。

AWS 提供的 AWS Key Management Service（AWS KMS）可为 AWS KMS 密钥提供持久、安全和冗余的存储。许多 AWS 服务都与 AWS KMS 集成来支持对您的数据进行加密。AWS KMS 使用经 FIPS 140-3 Level 3 验证的硬件安全模块来保护您的密钥。不存在以纯文本格式导出 AWS KMS 密钥的机制。

使用多账户策略部署工作负载时，应将 AWS KMS 密钥与使用这些密钥的工作负载保存在同一个账户中。This distributed model 将管理 AWS KMS 密钥的责任交给您的团队。在其它用例中，贵组织可以选择将 AWS KMS 密钥存储到集中式账户中。这种集中式结构需要额外的策略，以实现工作负载账户访问集中式账户中存储的密钥所需的跨账户访问，但可能更适用于多个 AWS 账户 共享单个密钥的用例。

无论密钥材料存储在哪里，都应通过使用 key policies 和 IAM 策略来严格控制对密钥的访问。密钥策略是控制对 AWS KMS 密钥的访问权限的主要方式。此外，AWS KMS 密钥授权可以提供对 AWS 服务的访问权限，从而代表您加密和解密数据。请查看 guidance for access control to your AWS KMS keys。

您应监控加密密钥的使用情况，以检测异常的访问规律。使用 AWS 托管密钥和 AWS KMS 中存储的客户自主管理型密钥执行的操作可以记录在 AWS CloudTrail 中，并应定期进行审查。特别注意监控密钥销毁事件。为了减少意外或恶意破坏密钥材料的情况，密钥销毁事件不会立即删除密钥材料。尝试删

除 AWS KMS 中的密钥时会经历一个 waiting period，默认为 30 天且最少为 7 天，这让管理员有时间审核这些操作并在必要时回滚请求。

大多数 AWS 服务使用 AWS KMS 的方式对您来说都是透明的，您只需要决定是使用 AWS 托管密钥还是客户自主管理型密钥。如果您的工作负载需要直接使用 AWS KMS 来加密或解密数据，则应使用 envelope encryption 来保护您的数据。此 AWS 加密开发工具包可为您的应用程序提供客户端加密原语，来实施信封加密并与 AWS KMS 集成。

实施步骤

1. 为密钥确定合适的密钥管理选项（AWS 托管或客户自主管理）。
   a. 为便于使用，AWS 为大多数服务提供 AWS 自有和 AWS 托管密钥，这样，无需管理密钥材料或密钥策略，即可提供静态加密功能。
   b. 使用客户自主管理型密钥时，请考虑使用默认密钥存储，以便在敏捷性、安全性、数据主权和可用性之间取得最佳平衡。其他用例可能需要使用附带 AWS CloudHSM 的自定义密钥存储或使用外部密钥存储。
2. 查看您用于工作负载的服务列表，以了解 AWS KMS 如何与该服务集成。例如，EC2 实例可以使用加密的 EBS 卷，验证从这些卷创建的 Amazon EBS 快照是否也使用客户自主管理型密钥进行加密，并减少未加密快照数据的意外泄露。
   a. AWS 服务如何使用 AWS KMS
   b. 有关 AWS 服务提供的加密选项的详细信息，请参阅该服务的用户指南或开发人员指南中的“静态加密”主题。
3. 实施 AWS KMS：AWS KMS 使您可以轻松创建和管理密钥，并控制各种 AWS 服务和应用程序中的加密使用情况。
   a. 入门：AWS Key Management Service（AWS KMS）
   b. 请查看 AWS KMS 密钥访问控制的最佳实践。
4. 考虑使用 AWS Encryption SDK：当应用程序需要在客户端加密数据时，使用包含 AWS KMS 集成的 AWS Encryption SDK。
   a. AWS Encryption SDK
5. 启用 IAM Access Analyzer 以自动审查是否存在过于宽泛的 AWS KMS 密钥策略并相应地发出通知。
   a. 考虑使用 custom policy checks，来验证资源策略更新是否不会授予对 KMS 密钥的公有访问权限。
6. 启用 Security Hub CSPM 以便在密钥策略配置错误、计划删除密钥或存在未启用自动轮换的密钥时，接收通知。

7. 确定适合您的 AWS KMS 密钥的日志记录级别。由于对 AWS KMS 的调用（包括只读事件）会被记录下来，因此与 AWS KMS 关联的 CloudTrail 日志可能会变得非常庞大。
   a. 一些组织倾向于将 AWS KMS 日志活动分成单独的跟踪。有关更多详细信息，请参阅《AWS KMS 开发者指南》的使用 CloudTrail 记录 AWS KMS API 调用部分。

资源

相关文档：

- AWS Key Management Service
- AWS 加密服务和工具
- 利用加密保护 Amazon S3 数据
- 信封加密
- 数字主权承诺
- 揭开 AWS KMS 密钥操作的神秘面纱、自带密钥、自定义密钥库和加密文字可移植性
- AWS Key Management Service 加密详情

相关视频：

- 中的加密原理AWS
- 在 AWS 上保护您的数据块存储
- AWS data protection: Using locks, keys, signatures, and certificates

相关示例：

- 使用 AWS KMS 实施高级访问控制机制

SEC08-BP02 强制实施静态加密

加密私有静态数据，来保持机密性并提供额外的一层防护，以防止无意的数据泄露或外流。加密可保护数据，因此，如果不首先对数据进行解密，则无法读取或访问加密的数据。清点和控制未加密的数据，以降低与数据泄露关联的风险。

期望结果：您拥有在私有数据处于静态时默认加密这些数据的机制。这些机制有助于保持数据的机密性，并提供额外的一层防护，以防止无意的数据泄露或外流。您可以维护未加密数据的清单，并了解为保护这些数据而采取的控制措施。

常见反模式：

- 不使用默认加密配置。
- 提供对解密密钥过于宽松的访问权限。
- 不监控加密和解密密钥的使用。
- 未加密便存储数据。
- 对所有数据使用相同的加密密钥，而不考虑数据用途、类型和分类。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

将加密密钥映射到工作负载中的数据分类。当对数据使用单个或非常少量的加密密钥时，这种方法有助于防止过于宽松的访问（请参阅 SEC07-BP01 了解数据分类方案）。

AWS Key Management Service（AWS KMS）与许多 AWS 服务集成，使加密静态数据更加轻松。例如，在 Amazon Elastic Compute Cloud（Amazon EC2）中，您可以对账户设置默认加密，以便自动加密新的 EBS 卷。使用 AWS KMS 时，请考虑需要对数据进行多严格的限制。默认和服务控制的 AWS KMS 密钥由 AWS 代表您进行管理和使用。对于需要对底层加密密钥进行精细访问的敏感数据，请考虑使用客户自主管理型密钥（CMK）。您可以完全控制 CMK，包括通过使用密钥策略进行轮换和访问管理。

此外，诸如 Amazon Simple Storage Service（Amazon S3）之类的服务现在默认加密所有新对象。这种实施提供了增强的安全性，而不会对性能产生任何影响。

其它服务，例如 Amazon Elastic Compute Cloud（Amazon EC2）或 Amazon Elastic File System（Amazon EFS），则支持默认加密设置。还可以使用 AWS Config 规则 来自动检查您是否正在为贵组织内的 Amazon Elastic Block Store (Amazon EBS) volumes、Amazon Relational Database Service (Amazon RDS) instances、Amazon S3 buckets 和其它服务使用加密。

AWS 还提供客户端加密选项，使您能够在将数据上传到云之前对数据进行加密。AWS Encryption SDK 提供了一种使用信封加密对数据进行加密的方法。您提供包装密钥，AWS Encryption SDK 为它加密的每个数据对象生成一个唯一数据密钥。如果需要托管的单租户硬件安全模块（HSM），请考虑 AWS CloudHSM。AWS CloudHSM 使您可在通过 FIPS 140-2 Level 3 验证的 HSM 上生成、导

入和管理加密密钥。AWS CloudHSM 的一些使用案例包括保护用于签发证书颁发机构（CA）的私有密钥，以及为 Oracle 数据库启用透明数据加密（TDE）。AWS CloudHSM 客户端开发工具包提供的软件使您可在将数据上传到 AWS 之前，使用存储在 AWS CloudHSM 中的密钥对客户端数据进行加密。Amazon DynamoDB Encryption Client 还使您可在将项目上传到 DynamoDB 表之前，对项目进行加密和签名。

实施步骤

- 为新的 Amazon EBS 卷配置默认加密：指定所有新创建的 Amazon EBS 卷要以加密形式创建，并选择使用 AWS 提供的默认密钥或您创建的密钥。
- 配置加密亚马逊机器映像（AMI）：通过复制已配置加密功能的现有 AMI，可自动加密根卷和快照。
- 配置 Amazon RDS 加密：通过使用加密选项，配置对您的 Amazon RDS 数据库集群和静态快照的加密。
- 使用策略创建和配置 AWS KMS 密钥，以限制对每个数据分类的相应主体的访问：例如，创建一个 AWS KMS 密钥用于加密生产数据，创建一个不同的密钥用于加密开发或测试数据。您还可以提供对其他 AWS 账户 的密钥访问权限。不妨考虑分开设立开发环境和生产环境的账户。如果您的生产环境需要解密开发账户中的构件，您可以编辑用于加密开发构件的 CMK 策略，使生产账户有能力解密这些构件。然后，生产环境可以摄取解密后的数据以用于生产。
- 在其它 AWS 服务中配置加密：对于您使用的其它 AWS 服务，请查看该服务的安全性文档，以确定该服务的加密选项。

资源

相关文档：

- AWS 加密工具
- AWS Encryption SDK
- AWS KMS 加密详情白皮书
- AWS Key Management Service
- AWS 加密服务和工具
- Amazon EBS Encryption
- Amazon EBS 卷的默认加密
- 加密 Amazon RDS 资源
- 如何为 Amazon S3 存储桶启用默认加密？
- 利用加密保护 Amazon S3 数据

相关视频：

- AWS 中的加密原理
- 在 AWS 上保护您的数据块存储

SEC08-BP03 自动执行静态数据保护

使用自动化技术来验证和执行静态数据控制。 使用自动扫描功能来检测数据存储解决方案的错误配置，并在可能的情况下通过自动程序化响应进行修复。 在 CI/CD 流程中融入自动化功能，以便在数据存储部署到生产环境之前检测出错误配置。

期望结果：自动化系统对数据存储位置进行扫描和监控，防止出现控制措施配置错误、未经授权的访问和意外使用。 检测到配置错误的存储位置后，自动修复措施就会启动。 自动化流程可创建数据备份，并在原始环境之外存储不可更改的副本。

常见反模式：

- 在支持的情况下，不考虑通过默认设置启用加密的选项。
- 在制定自动备份和恢复策略时，除操作事件外，不考虑安全事件。
- 不对存储服务执行公共访问设置。
- 不监控和审计保护静态数据的控制措施。

建立此最佳实践的好处：自动化有助于防止错误配置数据存储位置的风险。这有助于防止错误配置进入生产环境。这种最佳实践还有助于在发生错误配置时进行检测和修复。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

自动化是整个静态数据保护实践的主题。SEC01-BP06 自动部署标准安全控制措施介绍如何使用基础设施即代码（IaC）模板（例如 AWS CloudFormation）捕获资源配置。 这些模板已提交到版本控制系统，并用于通过 CI/CD 管道在 AWS 上部署资源。 这些技术同样适用于自动配置数据存储解决方案，如 Amazon S3 存储桶的加密设置。

您可以在 CI/CD 管道中使用 AWS CloudFormation Guard 中的规则，来检查您在 IaC 模板中定义的设置是否存在配置错误。 您可以使用 AWS Config 监控 CloudFormation 或其它 IaC 工具中尚未提供的设置，以防配置错误。 如 SEC04-BP04 启动对不合规资源的修复中所述，Config 针对错误配置生成的警报可自动修复。

将自动化功能融入权限管理策略，也是自动化数据保护不可或缺的组成部分。SEC03-BP02 授予最低访问权限和 SEC03-BP04 持续减少权限描述了如何配置最低权限访问策略，这些策略会受到 AWS Identity and Access Management Access Analyzer 的持续监控，以便得出何时可以减少权限的调查发现。 除了监控权限的自动化功能之外，您还可以配置 Amazon GuardDuty 以监控 EBS 卷（通过 EC2 实例）、S3 存储桶和受支持的 Amazon Relational Database Service 数据库的异常数据访问行为。

在检测敏感数据何时存储在未经授权的位置时，自动化功能也能发挥作用。SEC07-BP03 自动识别和分类描述了 Amazon Macie 如何监控您的 S3 存储桶中是否存在意外敏感数据，并生成可启动自动响应的警报。

按照 REL09 备份数据中的做法，制定自动数据备份和恢复策略。对于从安全事件中恢复和从操作事件中恢复，备份数据和恢复数据同样重要。

实施步骤

1. 在 IaC 模板中捕获数据存储配置。 在 CI/CD 管道中使用自动检查来检测错误配置。
   a. 您可以使用 CloudFormation 来处理 IaC 模板，并使用 CloudFormation Guard 来检查模板是否存在配置错误。
   b. 使用 AWS Config 以主动评估模式运行规则。使用此设置可在创建资源前，检查资源作为 CI/CD 管道中的一个步骤的合规性。
2. 监控资源中是否存在数据存储配置错误。
   a. 设置 AWS Config 以监控数据存储资源中控制措施配置的变化，并在检测到错误配置时生成警报，以便调用修复措施。
   b. 有关自动修复的更多指导，请参见 SEC04-BP04 启动对不合规资源的修复。
3. 通过自动化功能持续监控和减少数据访问权限。
   a. IAM Access Analyzer 可持续运行，在权限可能减少时发出警报。
4. 监控异常数据访问行为并发出警报。
   a. GuardDuty 可监控已知威胁特征，还监控 EBS 卷、S3 存储桶和 RDS 数据库等数据存储资源的基线访问行为偏差。
5. 对存储在意外位置的敏感数据进行监控并发出警报。
   a. 使用 Amazon Macie 持续扫描 S3 存储桶，查找敏感数据。
6. 自动对数据进行安全加密备份。
   a. AWS Backup 是一项托管服务，可为 AWS 上的各种数据来源创建加密和安全的备份。 弹性灾难恢复允许您复制完整的服务器工作负载，并保持持续的数据保护，恢复点目标（RPO）以秒为单位。 您可以配置这两项服务，使之协同工作，从而自动创建数据备份并复制到失效转移位置。 这有助于在受到操作或安全事件影响时保持数据可用。

资源

相关最佳实践：

- SEC01-BP06 自动部署标准安全控制措施
- SEC03-BP02 授予最低访问权限
- SEC03-BP04 持续减少权限
- SEC04-BP04 启动对不合规资源的修复
- SEC07-BP03 自动识别和分类
- REL09-BP02 保护并加密备份
- REL09-BP03 自动执行数据备份

相关文档：

- AWS Prescriptive Guidance: Automatically encrypt existing and new Amazon EBS volumes
- Ransomware Risk Management on AWS Using the NIST Cyber Security Framework (CSF)

相关示例：

- How to use AWS Config proactive rules and AWS CloudFormation Hooks to prevent creation of noncompliant cloud resources
- Automate and centrally manage data protection for Amazon S3 with AWS Backup
- AWS re:Invent 2023 - Implement proactive data protection using Amazon EBS snapshots
- AWS re:Invent 2022 - Build and automate for resilience with modern data protection

相关工具：

- AWS CloudFormation Guard
- AWS CloudFormation Guard 规则注册表
- IAM Access Analyzer
- Amazon Macie
- AWS Backup
- 弹性灾难恢复

SEC08-BP04 强制实施访问控制

为有助于保护静态数据，请使用隔离和版本控制等机制来强制实施访问控制。应用最低权限和条件访问控制。防止向公众授予访问您数据的权限。

期望结果：您验证只有获得授权的用户才能按照“需要知晓”的原则访问数据。您通过定期备份和版本控制来保护您的数据，以防止数据被有意或无意地修改或删除。您将关键数据与其它数据隔离，以保护其机密性和数据完整性。

常见反模式：

- 将具有不同敏感度要求或分类的数据存储在一起。
- 解密密钥的权限过于宽松。
- 数据分类不当。
- 不保留重要数据的详细备份。
- 提供对生产数据的持久访问。
- 未审计数据访问，也未定期检查权限。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

保护静态数据对于维护数据的完整性、机密性以及符合监管要求非常重要。您可以实施多种控制措施来协助实现这一目标，包括访问控制、隔离、条件访问和版本控制。

您可以按照最低权限原则强制实施访问控制，该原则仅向用户和服务提供执行其任务所需的权限。这包括对于加密密钥的访问权限。查看您的 AWS Key Management Service (AWS KMS) policies，来验证您授予的访问权限级别是否适当，以及相关条件是否适用。

可以通过为每个分类级别使用不同的 AWS 账户，根据不同的分类级别分离数据，并使用 AWS Organizations 管理这些账户。这种隔离有助于防止未经授权的访问，并最大限度地降低数据泄露的风险。

定期审核在 Amazon S3 存储桶策略中授予的访问权限级别。除非绝对必要，否则请避免使用可公开读取或写入的存储桶。考虑使用 AWS Config 来检测可公开可用的存储桶，并使用 Amazon CloudFront 来提供 Amazon S3 中的内容。验证正确配置了应不支持公开访问的存储桶，以防止公开访问。

对存储在 Amazon S3 中的关键数据实施版本控制和对象锁定机制。Amazon S3 版本控制保留对象的先前版本，以便在意外删除或覆盖时恢复数据。Amazon S3 对象锁定为对象提供强制访问控制，从

而防止对象在锁定到期之前被删除或覆盖，即使是根用户也是如此。此外，Amazon Glacier 文件库锁定为存储在 Amazon Glacier 中的归档提供了类似的功能。

实施步骤

1. 采用最低权限原则，强制实施访问控制：
   - 审核向用户和服务授予的访问权限，并验证他们是否只拥有执行其任务所需的权限。
   - 通过检查 AWS Key Management Service (AWS KMS) policies 来查看对加密密钥的访问权限。
2. 根据不同的分类级别将数据分开：
   - 对每个数据分类级别使用不同的 AWS 账户。
   - 使用 AWS Organizations 管理这些账户。
3. 审核 Amazon S3 存储桶和对象权限：
   - 定期审核在 Amazon S3 存储桶策略中授予的访问权限级别。
   - 除非绝对必要，否则请避免使用可公开读取或写入的存储桶。
   - 考虑使用 AWS Config 来检测公开可用的存储桶。
   - 使用 Amazon CloudFront 来提供 Amazon S3 中的内容。
   - 验证正确配置了应不支持公开访问的存储桶，以防止公开访问。
   - 可以对使用 IAM 身份验证的数据库和任何其它数据来源（例如 SQS 或第三方数据存储）应用相同的审核流程。
4. 使用 AWS IAM Access Analyzer：
   - 您可以配置 AWS IAM Access Analyzer 来分析 Amazon S3 存储桶，并在 S3 策略向外部实体授予访问权限时生成调查发现。
5. 实施版本控制和对象锁定机制：
   - 使用 Amazon S3 版本控制来保留对象的先前版本，这样可以从意外删除或覆盖中恢复。
   - 使用 Amazon S3 对象锁定来为对象提供强制访问控制，从而防止对象在锁定到期之前被删除或覆盖，即使是根用户也是如此。
   - 将 Amazon Glacier 文件库锁定用于存储在 Amazon Glacier 中的归档。
6. 使用 Amazon S3 清单：
   - 可以使用 Amazon S3 清单来审计和报告 S3 对象的复制和加密状态。
7. 审核 Amazon EBS 和 AMI 共享权限：
   - 审核 Amazon EBS 和 AMI 共享的共享权限，来验证映像和卷是否未与工作负载外部的 AWS 账户共享。
8. 定期审核 AWS Resource Access Manager 共享：

- 可以使用 AWS Resource Access Manager 来在 Amazon VPC 内共享资源，如 AWS Network Firewall 策略、Amazon Route 53 Resolver 规则和子网。
- 定期审计共享的资源，并停止共享不再需要共享的资源。

资源

相关最佳实践：

- SEC03-BP01 定义访问要求
- SEC03-BP02 授予最低访问权限

相关文档：

- AWS KMS 加密详情白皮书
- 管理对 Amazon S3 资源的访问权限简介
- 管理对 AWS KMS 资源的访问权限概览
- AWS Config 规则
- Amazon S3 + Amazon CloudFront: A Match Made in the Cloud
- 使用版本控制
- 使用 Amazon S3 对象锁定以锁定对象
- 共享 Amazon EBS 快照
- 共享 AMI
- Hosting a single-page application on Amazon S3
- AWS 全局条件键
- Building a Data Perimeter on AWS

相关视频：

- 在 AWS 上保护您的数据块存储

保护传输中数据

传输中数据是指从一个系统发送到另一个系统的任何数据。这包括您工作负载中资源之间的通信以及其他服务与最终用户之间的通信。通过为传输中数据提供适当级别的保护，您就可以保护工作负载数据的机密性和完整性。

保护 VPC 之间或本地位置之间的数据：您可以使用AWS PrivateLink 在 Amazon Virtual Private Cloud（Amazon VPC）之间或与 AWS 中托管的服务的本地连接之间创建安全的私有网络连接。您可以像访问自己私有网络上的服务那样访问 AWS 服务、第三方服务和其他 AWS 账户 账户中的服务。借助 AWS PrivateLink，无需使用互联网网关或 NAT，即可跨具有重叠 IP CIDR 的账户访问服务。您也不需要配置防火墙规则、路径定义或路由表。流量保留在 Amazon 主干网络上，不会流经互联网，所以数据会受到保护。您可以遵循行业特定的合规法规，例如《健康保险流通与责任法案》（HIPAA）和《欧美隐私盾协议》（EU/US Privacy Shield）。AWS PrivateLink 与第三方解决方案无缝协作，可创建精简的全球网络，从而让您加快向云端的迁移并利用可用的 AWS 服务。

最佳实践

- SEC09-BP01 实施安全密钥和证书管理
- SEC09-BP02 在传输中执行加密
- SEC09-BP03 对网络通信进行身份验证

SEC09-BP01 实施安全密钥和证书管理

传输层安全性协议（TLS）证书用于保障网络通信的安全，确立网站、资源和工作负载在互联网上以及专用网络上的身份。

期望结果：一个安全的证书管理系统，可以在公钥基础设施（PKI，Public Key Infrastructure）中预置、部署、存储和续订证书。安全密钥和证书管理机制可防止证书私钥材料泄露，并定期自动续订证书。它还与其他服务集成，为工作负载内的计算机资源提供安全的网络通信和标识。密钥材料永远不应能够通过人员的身份来访问。

常见反模式：

- 在证书部署或续订流程中执行人工步骤。
- 在设计私有证书颁发机构（CA，Certificate Authority）时，对 CA 层次结构的关注不够。
- 对公共资源使用自签名证书。

建立此最佳实践的好处：

- 通过自动化的部署和续订流程简化证书管理
- 鼓励使用 TLS 证书对传输中数据进行加密
- 提高了证书颁发机构执行的证书操作的安全性和可审计性
- 在 CA 层次结构的不同层级上组织管理职责

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

现代化工作负载广泛使用通过 PKI 协议（如 TLS）进行加密的网络通信。PKI 证书管理可能很复杂，但是，通过自动化的证书预置、部署和续订机制，可以减少与证书管理相关的麻烦。

AWS 提供了两种服务用于管理通用 PKI 证书：AWS Certificate Manager 和 AWS 私有证书颁发机构（AWS 私有 CA）。ACM 是客户用于预置、管理和部署证书的主要服务，适用于面向公众的工作负载以及私有 AWS 工作负载。ACM 使用 AWS 私有 CA 来颁发私有证书，并 integrates 许多其它 AWS 托管式服务，以便为工作负载提供安全的 TLS 证书。ACM 还可以从 Amazon Trust Services 颁发公开可信的证书。ACM 的公有证书可用于面向公众的工作负载，因为默认情况下，现代浏览器和操作系统信任这些证书。

利用 AWS 私有 CA，您可以建立自己的根证书颁发机构或从属证书颁发机构，并通过 API 颁发 TLS 证书。在 TLS 连接的客户端一侧控制和管理信任链的场景中，您可以使用这些类型的证书。除了 TLS 使用场景外，还可以使用 AWS 私有 CA 通过自定义模板向 Kubernetes 容器组（pod）、Matter 设备产品认证、代码签名和其他使用场景颁发证书。您还可以使用 IAM Roles Anywhere，向已经为其颁发了 X.509 证书（使用您的私有 CA 签名）的本地工作负载提供临时 IAM 凭证。

除了 ACM 和 AWS 私有 CA 之外，AWS IoT Core 还针对为物联网设备预置、管理和部署 PKI 证书提供专业化支持。AWS IoT Core 提供专门的机制，用于大规模将物联网设备载入到您的公有密钥基础设施中。

某些 AWS 服务，例如 Amazon API Gateway 和弹性负载均衡，提供自己的使用证书保护应用程序连接的能力。例如，API Gateway 和应用程序负载均衡器（ALB）都支持使用客户端证书的双向 TLS（mTLS），而这些证书是使用 AWS 管理控制台、CLI 或 API 创建和导出的。

建立私有 CA 层次结构的注意事项

当您需要建立私有 CA 时，请务必重视预先正确设计 CA 层次结构。在创建私有 CA 层次结构时，最佳实践是将 CA 层次结构的每个级别部署到单独的 AWS 账户 中。这个有意而为的步骤可减少 CA 层次结构中每个级别的暴露范围，使得发现 CloudTrail 日志数据中的异常变得更加简单，并可在某个账户遭

到未经授权的访问时，缩小访问或影响的范围。根 CA 应位于自己的独立账户中，并且只能用于发布一个或多个中间 CA 证书。

然后，在不同于根 CA 账户的账户中创建一个或多个中间 CA，为最终用户、设备或其他工作负载发布证书。最后，从您的根 CA 向中间 CA 颁发证书，后者随之向您的最终用户或设备颁发证书。有关规划 CA 部署和设计 CA 层次结构（包括弹性规划、跨区域复制、在组织中共享 CA 等）的更多信息，请参阅《Planning your AWS 私有 CA deployment》。

实施步骤

1. 确定您的使用场景所需的相关 AWS 服务：
   - 许多使用场景都可以利用现有的 AWS 公钥基础设施并使用 AWS Certificate Manager。ACM 可用于为 Web 服务器、负载均衡器或公共可信证书的其他用途部署 TLS 证书。
   - 在您需要建立自己的私有证书颁发机构层次结构或需要使用可导出证书时，请考虑 AWS 私有 CA。然后，可以使用 ACM 颁发多种类型的终端实体证书（使用 AWS 私有 CA）。
   - 对于必须为嵌入式物联网（IoT）设备大规模预置证书的使用场景，请考虑使用 AWS IoT Core。
   - 考虑在 Amazon API Gateway 或应用程序负载均衡器等服务中使用原生 mTLS 功能。
2. 尽可能实施自动证书续订：
   - 将 ACM 托管续订用于 ACM 颁发的证书以及集成的 AWS 托管服务。
3. 建立日志记录和审计跟踪：
   - 启用 CloudTrail 日志，以便跟踪对具有证书颁发机构的账户的访问。请考虑在 CloudTrail 中配置日志文件完整性验证，用于验证日志数据的真实性。
   - 定期生成和审查审计报告，列出您的私有 CA 已颁发或撤销的证书。这些报告可以导出到 S3 存储桶。
   - 部署私有 CA 时，您还需要创建一个 S3 存储桶，用于存储证书撤销列表（CRL，Certificate Revocation List）。有关根据工作负载要求配置此 S3 存储桶的指南，请参阅《Planning a certificate revocation list (CRL)》。

资源

相关最佳实践：

- SEC02-BP02 使用临时凭证
- SEC08-BP01 实施安全密钥管理
- SEC09-BP03 对网络通信进行身份验证

相关文档：

- How to host and manage an entire private certificate infrastructure in AWS
- How to secure an enterprise scale ACM Private CA hierarchy for automotive and manufacturing
- Private CA best practices
- How to use AWS RAM to share your ACM Private CA cross-account

相关视频：

- Activating AWS Certificate Manager Private CA（讲习会）

相关示例：

- Private CA workshop
- 物联网设备管理讲习会（包括设备预置）

相关工具：

- 使用 AWS 私有 CA 的 Kubernetes 证书管理器插件

SEC09-BP02 在传输中执行加密

实施您根据贵组织的政策、监管义务和标准定义的加密要求，以帮助满足组织、法律和合规性要求。如要在虚拟私有云（VPC）外部传输敏感数据，务必仅使用具有加密功能的协议。即使在不可信的网络中传输数据，加密也有助于保持数据的机密性。

期望结果：对资源与互联网之间的网络流量进行加密，以减少对数据的未经授权访问。根据您的安全需求，加密内部 AWS 环境中的网络流量。可以使用安全的 TLS 协议和密码套件对传输中数据进行加密。

常见反模式：

- 使用已弃用的 SSL、TLS 和密码套件组件版本（例如，SSL v3.0、1024 位 RSA 密钥和 RC4 密码）。
- 允许未加密的（HTTP）流量进出面向公众的资源。
- 未在 X.509 证书到期前监控和替换证书。

- 对 TLS 使用自签名 X.509 证书。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

AWS 服务提供使用 TLS 的 HTTPS 端点进行通信，从而可以在与 AWS API 通信时提供传输中加密。通过使用安全组，可以在虚拟私有云（VPC）中审计和阻止不安全的 HTTP 协议。也可以在 Amazon CloudFront 中或应用程序负载均衡器上，将 HTTP 请求自动重定向到 HTTPS。可以使用 Amazon Simple Storage Service (Amazon S3) bucket policy 来限制通过 HTTP 上传对象的功能，从而有效地强制使用 HTTPS 将对象上传到存储桶。您可以完全控制计算资源，以便在整个服务中实施加密。您也可以利用 VPN 连接从外部网络或 AWS Direct Connect 连接到您的 VPC 中，以便于对流量进行加密。验证客户端是否使用至少 TLS 1.2 来调用 AWS API，因为 AWS has deprecated the use of earlier versions of TLS as of February 2024。我们建议您使用 TLS 1.3。如果您对传输中加密有特殊要求，可以在 AWS Marketplace 中找到可用的第三方解决方案。

实施步骤

- 实施传输中加密：您定义的加密要求应基于最新的标准和最佳实践，且仅允许使用安全协议。例如，配置一个安全组，仅允许通过 HTTPS 协议访问应用程序负载均衡器或 Amazon EC2 实例。
- 在边缘服务中配置安全协议：使用 Amazon CloudFront 配置 HTTPS，并使用适合您的安全状况和使用案例的安全配置文件。
- 将 VPN 用于外部连接：考虑使用 IPsec VPN 来保护点对点或网络对网络连接，以帮助实现数据隐私和完整性。
- 在负载均衡器中配置安全协议：选择一个安全策略，该策略提供受客户端支持且将要连接到侦听器的强大密码套件。为应用程序负载均衡器创建 HTTPS 侦听器。
- 在 Amazon Redshift 中配置安全协议：将集群配置为要求安全套接字层（SSL）或传输层安全性协议（TLS）连接。
- 配置安全协议：查看 AWS 服务文档，以确定传输中加密功能。
- 上传到 Amazon S3 存储桶时配置安全访问：使用 Amazon S3 存储桶策略控制措施执行对数据的安全访问。
- 不妨考虑使用 AWS Certificate Manager：ACM 允许您预置、管理和部署用于 AWS 服务的公有 TLS 证书。
- 不妨考虑使用 AWS 私有证书颁发机构 满足私有 PKI 需求：AWS 私有 CA 允许您创建私有证书颁发机构（CA）层次结构，以签发可用于创建加密 TLS 通道的终端实体 X.509 证书。

资源

相关文档：

- 将 HTTPS 与 CloudFront 搭配使用
- 使用 AWS Virtual Private Network 将 VPC 连接到远程网络
- 为应用程序负载均衡器创建 HTTPS 侦听器
- 教程：在 Amazon Linux 2 上配置 SSL/TLS
- 使用 SSL/TLS 加密与数据库实例的连接
- 配置连接的安全选项

SEC09-BP03 对网络通信进行身份验证

使用传输层安全性协议（TLS）或 IPsec 等支持身份验证的协议来验证通信的身份。

将您的工作负载设计为在服务和应用程序之间通信或与用户通信时，使用经过身份验证的安全网络协议。使用支持身份验证和授权的网络协议，可以加强对网络流量的控制，并减少未授权访问的影响。

期望结果：工作负载具有明确定义的数据面板和控制面板，它们控制流量在服务之间的流动。在技术上可行的情况下，流量将使用经过身份验证和加密的网络协议。

常见反模式：

- 工作负载中存在未加密或未经身份验证的流量。
- 在多个用户或实体之间重用身份验证凭证。
- 仅依赖网络控制作为访问控制机制。
- 创建自定义身份验证机制，而不是依赖行业通用身份验证机制。
- VPC 中的服务组件或其它资源之间有过于宽松的流量流动。

建立此最佳实践的好处：

- 限制未经授权访问工作负载某一部分的影响范围。
- 提供更高级别的保障，即操作只能由经过身份验证的实体执行。
- 通过明确定义和强制执行预期的数据传输接口，改善服务的解耦。
- 通过请求归因和明确定义的通信界面，增强监控、日志记录和事件响应。
- 通过将网络控制与身份验证和授权控制相结合，为您的工作负载提供深度防御。

在未建立这种最佳实践的情况下暴露的风险等级：低

实施指导

您的工作负载的网络流量模式可分为两类：

- 东西向流量代表构成工作负载的服务之间的流量。
- 南北向流量代表您的工作负载和使用器之间的流量。

对南北向流量进行加密是常见做法，而使用经过身份验证的协议保护东西向流量则不太常见。现代安全实践建议，仅靠网络设计并不足以在两个实体之间建立可信关系。当两项服务可能位于公共网络边界内时，极佳做法仍然是对这些服务之间的通信进行加密、身份验证和授权。

例如，无论请求来自哪个网络，AWS 服务 API 都使用 AWS 签名版本 4（SigV4）签名协议对调用方进行身份验证。这种身份验证可确保 AWS API 可以验证请求操作的身份，然后可以将该身份与策略结合起来，作出授权决策，以确定是否应该允许该操作。

Amazon VPC Lattice 和 Amazon API Gateway 等服务允许您使用相同的 SigV4 签名协议，为自己的工作负载中的东西向流量添加身份验证和授权。如果您的 AWS 环境之外的资源要与服务进行通信，而服务需要基于 SigV4 的身份验证和授权，则您可以对非 AWS 资源使用 AWS Identity and Access Management（IAM）Roles Anywhere 来获取临时 AWS 凭证。这些凭证可用于对使用 SigV4 的服务请求进行签名，以授权访问权限。

验证东西向流量的另一种常见机制是 TLS 双向身份验证（mTLS）。许多物联网（IoT）、企业对企业应用程序和微服务都使用 mTLS，通过使用客户端和服务器端 X.509 证书来验证 TLS 通信两端的身份。这些证书可以由 AWS 私有证书颁发机构（AWS 私有 CA）颁发。可以使用 Amazon API Gateway 等服务，针对工作负载间或工作负载内的通信提供 mTLS 身份验证。应用程序负载均衡器还支持将 mTLS 用于内部或外部工作负载。虽然 mTLS 为 TLS 通信的两端提供身份验证信息，但它不提供授权机制。

最后，OAuth 2.0 和 OpenID Connect（OIDC）这两种协议通常用于控制用户对服务的访问，但如今在服务间流量中也变得越来越流行。API Gateway 提供了 JSON Web 令牌（JWT）授权方，允许工作负载使用 OIDC 或 OAuth 2.0 身份提供商颁发的 JWT 来限制对 API 路由的访问。可依据 OAuth2 范围来作出基本授权决策，但授权检查仍需要在应用层实现，仅靠 OAuth2 范围无法支持更复杂的授权需求。

实施步骤

- 定义并记录您的工作负载网络流：实施深度防御策略的第一步是定义工作负载的流量。
  - 创建数据流示意图，明确定义构成工作负载的不同服务之间如何传输数据。此示意图是强制这些流量通过经身份验证的网络渠道传输的第一步。

  - 在开发和测试阶段对您的工作负载进行检测，以验证数据流示意图是否准确反映了工作负载在运行时的行为。
  - 在执行威胁建模练习时，数据流示意图可能也很有用，如《SEC01-BP07 使用威胁模型识别威胁并确定缓解措施的优先级》中所述。
- 建立网络控制：考虑使用 AWS 功能建立起与数据流相符的网络控制。虽然网络边界不应该是唯一的安全控制措施，但它们在深度防御策略中提供了一个安全层来保护您的工作负载。
  - 使用安全组来建立、定义和限制资源之间的数据流。
  - 考虑使用 AWS PrivateLink 与 AWS 以及支持 AWS PrivateLink 的第三方服务通信。通过 AWS PrivateLink 接口端点发送的数据保留在 AWS 网络主干内，而不通过公共互联网传输。
- 在工作负载中实施跨服务的身份验证和授权：选择极其适合在工作负载中提供经过身份验证的加密流量的一组 AWS 服务。
  - 考虑使用 Amazon VPC Lattice 来保护服务间的通信。VPC Lattice 可以结合使用 Sigv4 身份验证与身份验证策略来控制服务间的访问。
  - 对于使用 mTLS 进行的服务间通信，请考虑使用 API Gateway 和应用程序负载均衡器。AWS 私有 CA 可用于建立私有 CA 层次结构，该层次结构能够颁发与 mTLS 结合使用的证书。
  - 与使用 OAuth 2.0 或 OIDC 的服务集成时，可以考虑使用 JWT 授权方的 API Gateway。
  - 对于工作负载和物联网设备之间的通信，可以考虑使用 AWS IoT Core，它提供多种网络流量加密和身份验证选项。
- 监控未经授权的访问：持续监控非预期的通信渠道、试图访问受保护资源的未授权主体以及其它不当访问模式。
  - 如果使用 VPC Lattice 来管理对服务的访问，请考虑启用和监控 VPC Lattice 访问日志。这些访问日志包括有关请求实体的信息、源和目的地 VPC 等网络信息以及请求元数据。
  - 考虑启用 VPC 流日志，以捕获网络流量的元数据并定期检查是否存在异常。
  - 有关规划、模拟和响应安全事件的更多指导，请参阅《AWS Security Incident Response Guide》和 AWS Well-Architected Framework 安全性支柱的“事件响应”部分。

资源

相关最佳实践：

- SEC03-BP07 分析公共和跨账户访问
- SEC02-BP02 使用临时凭证
- SEC01-BP07 使用威胁模型识别威胁并确定缓解措施的优先级

相关文档：

- Evaluating access control methods to secure Amazon API Gateway APIs
- Configuring mutual TLS authentication for a REST API
- How to secure API Gateway HTTP endpoints with JWT authorizer
- Authorizing direct calls to AWS services using AWS IoT Core credential provider
- 《AWS Security Incident Response Guide》

相关视频：

- AWS re:invent 2022: Introducing VPC Lattice
- AWS re:invent 2020: Serverless API authentication for HTTP APIs on AWS

相关示例：

- Amazon VPC Lattice Workshop
- Zero-Trust Episode 1 – The Phantom Service Perimeter workshop

事件响应

即使采用成熟的预防和检测性控制措施，您的组织也应实施机制来响应安全事件并缓解安全事件可能带来的影响。您的准备工作会极大地影响团队在意外事件发生期间采取有效行动、对问题进行隔离、遏制和取证并将运行状态恢复到已知良好状态的能力。在安全事件发生之前确保相关工具和访问权限部署到位，然后通过 GameDay 活动定期进行事件响应演练，这样有助于确保您有能力恢复并最大限度避免业务中断。

主题

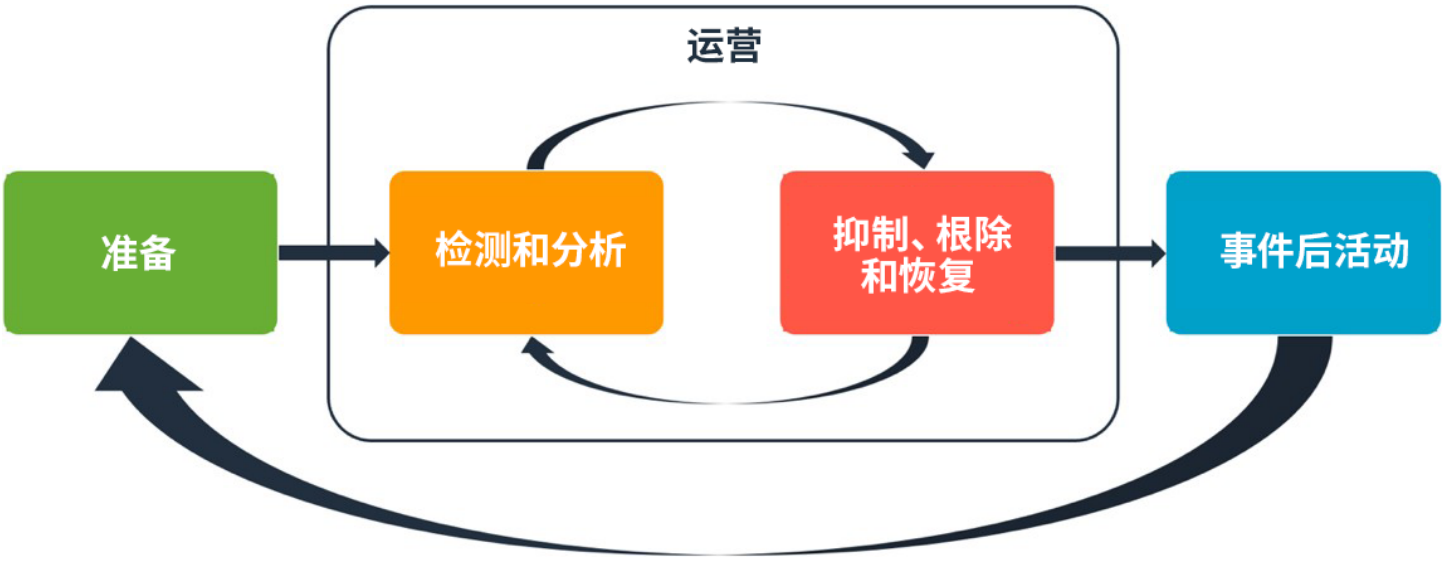
AWS 事件响应的各个方面

组织内的所有 AWS 用户都应对安全事件响应流程有基本的了解，并且安全人员应了解如何响应安全问题。教育、培训和经验对于成功的云事件响应计划至关重要，最好在处理可能发生的安全事件之前提前实施。云中成功的事件响应计划的基础在于准备工作、操作和事后活动。

要了解其中的各个方面，请考虑以下描述：

- 准备工作：让事件响应团队做好准备，以便在 AWS 中检测和响应事件，方法是启用检测控件，并确保对必要的工具和云服务拥有适当的访问权限。此外，还应通过人工和自动化的方式准备必要的行动手册，以确保可靠且一致的响应。
- 操作：按照 NIST 的事件响应阶段（检测、分析、遏制、根除和恢复）对安全事件和潜在事件采取相应操作。
- 事后活动：对安全事件和模拟的结果进行迭代，以提高响应的有效性，从响应和调查中获得更多价值，并进一步降低风险。您必须从事件中吸取经验教训，并对改进活动占有很大的所有权。

下图显示了这些方面的流程，与前面提到的 NIST 事件响应生命周期一致，但操作包括检测、分析、遏制、根除和恢复。



AWS 事件响应的各个方面

云响应的设计目标

尽管事件响应的一般流程和机制（例如《NIST SP 800-61 计算机安全事件处理指南》中定义的那些流程和机制）依然有效，但我们鼓励您评估这些与云环境中的安全事件响应相关的特定设计目标：

- 建立响应目标：与利益相关方、法律顾问和组织领导合作，以确定事件响应目标。一些共同的目标包括控制和缓解问题、恢复受影响的资源、保留数据为便取证、恢复到已知安全的操作，以及最终从事件中吸取教训。
- 利用云进行响应：在云端（即事件和数据的发生地）实施响应模式。
- 了解所拥有和需要的证据：通过复制日志、资源、快照和其他证据并将其存储在一个集中的响应专用云账户中来保存这些内容。使用标签、元数据和保留策略实施机制。您需要了解自己使用了哪些服务，然后确定调查这些服务的要求。为了便于您了解自己的环境，您还可以使用标记。
- 使用重新部署机制：如果安全异常可归因于一个配置错误，那么可能只需使用适当的配置重新部署资源来删除差异，即可完成修复。如果发现可能存在漏洞，请核实您重新部署时是否包括成功且经过验证的根本原因缓解措施。
- 尽可能自动化：当问题出现或事件重复发生时，建立机制，以程序化方式对常见事件进行分类和响应。对于自动化程度不足的独特、复杂或敏感事件，使用人工响应。
- 选择可扩展的解决方案：尽量让组织采用方法的可扩展性与云计算能力相匹配。实施可在您环境中扩展的检测和响应机制，有效地缩短检测与响应之间的时间差。
- 了解并改进流程：主动找出流程、工具或人员的不足，并实施计划来弥补这些不足。模拟是找出不足并改进流程的妥善方法。

这些设计目标会提醒您审查架构实施情况，确定是否同时具备事件响应能力和威胁检测能力。在规划云端实施时，应考虑如何应对事件，最好使用具备司法有效性的响应方法。在某些情况下，这意味着您可能需要专门为这些响应任务设置多个组织、账户和工具。这些工具和功能应通过部署管道提供给事件响应者。它们不应该是静态的，因为这会导致更大的风险。

准备

要想及时、有效地应对事件，为事件做好准备至关重要。“准备工作”涉及三个领域：

- 人员：要让员工做好应对安全事件的准备，需要确定事件响应的利益相关方，并对他们进行事件响应和云技术方面的培训。
- 流程：为安全事件做好流程准备，包括记录架构、制定全面的事件响应计划，以及创建行动手册，以便对安全事件做出一致响应。
- 技术：为安全事件做好技术准备，包括设置访问权限、汇总和监控必要的日志、实施有效的警报机制，以及培养响应和调查能力。

对有效的事件响应而言，每个领域都同等重要。没有这三项，任何事件响应计划都不完整或无效。您需要在人员、流程和技术方面做好准备，并将其紧密集成，以便为应对事件做好准备。

最佳实践

- SEC10-BP01 确定关键人员和外部资源
- SEC10-BP02 制定事件管理计划
- SEC10-BP03 准备取证能力
- SEC10-BP04 制定和测试安全事件响应行动手册
- SEC10-BP05 预置访问权限
- SEC10-BP06 预部署工具
- SEC10-BP07 运行模拟

SEC10-BP01 确定关键人员和外部资源

确定内部和外部人员、资源和法律义务，来协助组织应对事件。

期望结果：您有一份关键人员名单、他们的联系信息以及他们在应对安全事件时扮演的角色。您可以定期审查这些信息并进行更新，以便分别从内部工具和外部工具的角度反映人事变动。在记录这些信息时，您需要考虑所有第三方服务提供商和供应商，包括安全合作伙伴、云提供商和软件即服务

（SaaS，Software-as-a-Service）应用程序。在安全事件发生期间，有适当级别的责任、背景和访问权限的人员能够做出响应和恢复动作。

常见反模式：

- 在应对安全事件时，没有维护一份包含关键人员联系信息、角色和职责的最新关键人员名单。
- 在应对事件和从事件中恢复时，假设每个人都了解人员、依赖项、基础设施和解决方案。
- 没有代表关键基础设施或应用程序设计的文档或知识库。
- 没有为新员工制定适当的入职流程，无法有效参与安全事件响应，例如进行事件模拟。
- 没有制定当关键人员暂时无法到岗或者在安全事件中无法做出反应时，所需要的上报途径。

建立此最佳实践的好处：这种实践减少了事件发生期间用于确定合适人员及其角色的分流和响应时间。通过维护一份关键人员及其角色的最新名单，极大限度地减少事件发生期间的时间浪费，这样您就能够让合适的人员进行分流并从事件中恢复过来。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

确定组织中的关键人员：维护一份贵组织内需要参与事件的人员的联系名单。在发生组织变革、晋升和团队变动等人事变动时，应定期审查和更新这些信息。这对于事件经理、事件响应者和沟通负责人等关键角色尤其重要。

- 事件经理：事件经理在事件响应期间拥有全面权力。
- 事件响应者：事件响应者负责调查和修复活动。这些人员可能因事件类型而异，但通常是负责受影响的应用程序的开发人员和运营团队。
- 沟通负责人：沟通负责人负责内外部沟通，特别是与公共机构、监管机构和客户的沟通。
- 入职流程：定期对新员工进行培训和入职，使他们具备必要的技能和知识，以便有效地为事件响应工作做出贡献。将模拟和动手练习作为入职流程的一部分，以协助他们做好准备。
- 主题专家（SME）：对于分布式自主团队，我们建议您为任务关键型工作负载确定一名 SME。主题专家有助于我们深入了解事件中涉及的关键工作负载的运行和数据分类。

示例表格式：

```
  | Role | Name | Contact Information | Responsibilities |
1 | —- | —- | —- | —- |
```

```
2 | Incident Manager | Jane Doe| jane.doe@example.com | Overall authority during
 response |
3 | Incident Responder | John Smith | john.smith@example.com | Investigation and
 remediation |
4 | Communications Lead | Emily Johnson | emily.johnson@example.com | Internal and
 external communications |
5 | Communications Lead | Michael Brown | michael.brown@example.com | Insights on
 critical workloads |
```

考虑使用 AWS Systems Manager Incident Manager 功能，来捕获关键联系人、制定响应计划、自动执行随时待命方案并制定上报计划。通过随时待命方案自动安排和轮换所有员工，使工作负载的责任由其所有者分担。这促进了良好的实践，例如发布相关指标和日志，以及定义与工作负载相关的警报阈值。

确定外部合作伙伴：企业运用独立软件供应商（ISV）、合作伙伴和分包商开发的工具，为客户构建差异化解决方案。让各方的这些关键人员参与进来，他们有助于应对事件并从事件中恢复。我们建议您注册相应级别的 支持，以便通过支持案例及时联系 AWS 主题专家。考虑与所有关键解决方案提供商就工作负载达成类似安排。有些安全事件要求上市企业向相关公共机构和监管机构通报事件及影响。请维护和更新相关部门和负责人的联系信息。

实施步骤

1. 设置事件管理解决方案。
   a. 考虑在您的安全工具账户中部署 Incident Manager。
2. 在事件管理解决方案中定义联系人。
   a. 为每位联系人至少定义两种联系渠道（例如短信、电话或电子邮件），以便确保事件发生期间能够联系上。
3. 制定响应计划。
   a. 确定事件发生时应接洽的最合适的联系人。根据参与人员的角色，而不是单个联系人，制定上报计划。考虑纳入可能负责通知外部实体的联系人，即使他们没有直接参与解决事件也是如此。

资源

相关最佳实践：

- OPS02-BP03 确定对运营活动绩效负责的责任人

相关文档：

- AWS《Security Incident Response Guide》

相关示例：

- AWS 客户行动手册框架
- 准备和响应 AWS 环境中的安全事件

相关工具：

- AWS Systems Manager Incident Manager

相关视频：

- Amazon's approach to security during development

SEC10-BP02 制定事件管理计划

为事件响应制定的第一个文档是事件响应计划。事件响应计划旨在为您的事件响应计划和战略奠定基础。

建立此最佳实践的好处：要想成功实现可扩展的事件响应计划，制定全面且明确定义的事件响应流程是关键。在发生安全事件时，明确的步骤和工作流有助于您及时做出响应。您可能已经有事故响应流程。无论您当前的状态如何，定期更新、迭代和测试事件响应流程都很重要。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

对于响应、缓解安全事件的潜在影响并从中恢复来说，事件管理计划是至关重要的。事件管理计划是一个结构化的过程，用于及时地确定、补救和响应安全事件。

云的许多操作角色和要求都与本地环境中的相同。在创建事件管理计划时，应考虑最符合业务成果和合规性要求的响应和恢复策略，这一点非常重要。例如，如果您在 AWS 中运行符合美国 FedRAMP 标准的工作负载，请遵守 NIST SP 800-61 Computer Security Handling Guide 中的建议。同样，在运行存储个人身份信息（PII）的工作负载时，请考虑如何防止和应对与数据驻留和使用相关的问题。

在为 AWS 中的工作负载制定事件管理计划时，请首先使用 AWS 责任共担模式，以便构建针对事件响应的深度防御方法。在此模式中，AWS 负责管理云本身的安全，云内部的安全则由您负责。这意味着

您将保留控制权，并对选择实施的安全控制机制负责。《AWS Security Incident Response Guide》详细介绍了构建以云为中心的事件管理计划的关键概念和基本指南。

必须不断地迭代有效的事件管理计划，使其与您的云运营目标保持一致。在创建和改进事件管理计划时，请考虑使用下面详述的实施计划。

实施步骤

1. 定义组织内部用于处理安全事件的角色和职责。这应涉及不同部门的代表，包括：
   - 人力资源（HR）
   - 执行团队
   - 法务部门
   - 应用程序所有者和开发人员（主题专家或 SME）
2. 明确概述在事件发生期间谁负责、对谁问责、咨询谁以及通知谁（RACI）。创建 RACI 图表来促进快速和直接的沟通，并清楚地概述事件不同阶段的领导关系。
3. 在事件发生期间，让应用程序所有者和开发人员（SME）参与进来，因为他们可以提供有价值的信息和背景信息来帮助衡量影响。与这些 SME 建立关系，并在实际事件发生之前与他们练习事件响应场景。
4. 让值得信赖的合作伙伴或外部专家参与调查或响应过程，因为他们可以提供额外的专业知识和视角。
5. 使您的事件管理计划和角色与管理您组织的任何当地法规或合规要求保持一致。
6. 定期练习和测试您的事件响应计划，并涉及所有已定义的角色和职责。这有助于简化流程，并验证您对安全事件的响应井井有条且高效。
7. 定期审核和更新角色、职责和 RACI 图表，或者在组织结构或要求发生变化时进行审核和更新。

了解 AWS 响应团队和支持

- AWS 支持
  - 支持 包含一系列计划，这些计划旨在让您能够运用各种工具和专业知识来为成功部署和正常实施 AWS 解决方案提供支持。如果您需要技术支持及更多资源来规划、部署和优化 AWS 环境，则可以选择最符合 AWS 使用案例的支持计划。
  - 考虑将支持中心（在 AWS 管理控制台 中，需要登录）作为中心联系点，为影响您 AWS 资源的问题获取支持。对 支持 的访问由 AWS Identity and Access Management 控制。有关获取对 支持 功能的访问权限的更多信息，请参阅《Getting started with 支持》。
- AWS 客户事件响应团队（CIRT）

  - AWS 客户事件响应团队（CIRT）是一支专业的 AWS 全球团队，全天候向客户提供支持，协助客户解决根据 AWS 责任共担模式应由客户一方负责的安全事件。
  - 当 AWS CIRT 为您提供支持时，他们会为 AWS 上出现的安全事件提供分类和恢复方面的协助。他们可以使用 AWS 服务日志来协助分析根本原因，并为您提供恢复建议。他们还可以提供安全建议和最佳实践，从而让您以后能够避免出现安全事件。
  - AWS 客户要与 AWS CIRT 交流，可以开立 支持 案例。
- AWS 安全事件响应
  - 在 re:Invent 2024 上发布的 AWS 安全事件响应是一项托管式安全事件响应服务，它既使用现代分诊技术，又使用人工干预。该服务接收所有 GuardDuty 调查发现和发送给 AWS Security Hub CSPM 进行分诊的任何第三方调查发现，以仅针对需要调查的调查发现时向客户发出提醒。该服务还提供了一个门户，用于针对客户注意到的安全事件提交被动案例，并获得 AWS 的高级事件响应团队的支持。
- DDoS 响应支持
  - AWS 提供 AWS Shield，它提供了托管的分布式拒绝服务（DDoS）攻击保护服务，可保护在 AWS 上运行的 Web 应用程序。Shield 提供不间断检测和自动化内嵌缓解措施，可以最大限度地减少应用程序停机时间和延迟，因此无需与 支持 交流即可从 DDoS 保护中受益。Shield 分为两个级别：AWS Shield Standard 和 AWS Shield Advanced。要了解这两个级别之间的区别，请参阅《Shield 功能文档》。
- AWS Managed Services（AMS）
  - AWS Managed Services（AMS）可持续管理您的 AWS 基础设施，让您可以专注于应用程序。AMS 实施最佳实践来维护您的基础设施，让您能够降低运营开销和风险。AMS 可以自动执行常见活动 (例如更改请求、监控、补丁管理、安全性和备份服务)，并可以提供全生命周期服务来预置、运行和支持您的基础设施。
  - AMS 负责部署一套安全检测控制措施，并全天候提供对警报的第一线响应。启动警报后，AMS 遵循一组标准的自动和手动行动手册，验证是否有一致的响应。这些行动手册在功能部署期间与 AMS 客户共享，这样客户就能够开发并与 AMS 协调响应措施。

制定事件响应计划

事件响应计划旨在为您的事件响应计划和战略奠定基础。事件响应计划应包含在正式文档中。事件响应计划通常包括以下部分：

- 事件响应团队概述：概述事件响应团队的目标和职能。
- 角色和职责：列出事件响应利益相关者，并详细说明他们在发生事件时的角色。
- 沟通计划：详细介绍联系信息，以及在事件发生期间如何进行沟通。

- 后备沟通方法：此时的最佳实践是采用带外通信，作为事件沟通的后备。AWS Wickr 就是一个提供安全的带外通信渠道的应用程序示例。
- 事件响应阶段和应采取的行动：列举事件响应的各个阶段（例如，检测、分析、消除、遏制和恢复），包括在这些阶段中要采取的高级别操作。
- 事件严重性和优先级定义：详细说明如何对事件的严重性进行分类，如何确定事件的优先级，然后详细说明严重性定义对上报程序有何影响。

尽管这些内容部分在各种规模和行业的公司中很常见，但每个组织的事件响应计划都是独一无二的。您需要制定最适合贵组织的事件响应计划。

资源

相关最佳实践：

- SEC04 检测

相关文档：

- 《AWS Security Incident Response Guide》
- NIST：计算机安全事件处理指南

SEC10-BP03 准备取证能力

在发生安全事件之前，可以考虑构建取证能力来支持安全事件调查工作。

在未建立这种最佳实践的情况下暴露的风险等级：中

传统本地取证的概念适用于 AWS。有关开始在 AWS 云 中构建取证功能的关键信息，请参阅《Forensic investigation environment strategies in the AWS 云》。

设置好取证的环境和 AWS 账户 结构后，确定在以下四个阶段有效执行可靠取证方法所需的技术：

- 收集：收集相关的 AWS 日志，例如 AWS CloudTrail、AWS Config、VPC 流日志和主机级日志。收集受影响的 AWS 资源的快照、备份和内存转储（如果有）。
- 检查：通过提取和评测相关信息来检查收集到的数据。
- 分析：分析收集到的数据，以便了解事件并从中得出结论。
- 报告：提供分析阶段得出的信息。

实施步骤

准备取证环境

AWS Organizations 有助于您随着 AWS 资源的增长和扩展，集中管理和监管 AWS 环境。AWS 组织会整合您的 AWS 账户，这样您就能够将这些账户作为一个单元进行管理。您可以使用组织单元 (OU)，将账户分组到一起，作为一个单元管理。

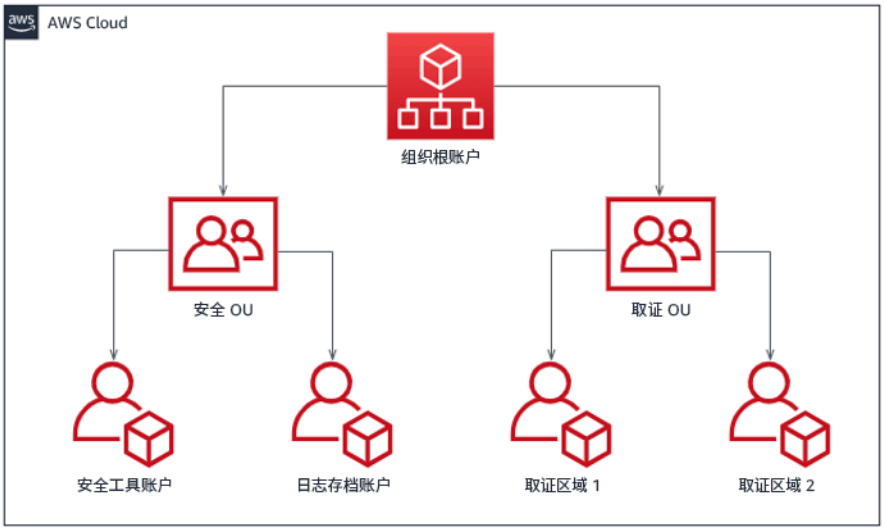
对于事件响应，拥有一个支持事件响应功能的 AWS 账户 结构（包括安全 OU 和取证 OU）会很有帮助。在安全 OU 中，您应该拥有以下账户：

- 日志存档：将日志聚合到权限有限的日志存档 AWS 账户 中。
- 安全工具：将安全服务集中在安全工具 AWS 账户 中。此账户以安全服务的委托管理员身份运行。

在取证 OU 中，您可以选择实施单一取证账户，也可以为您运营的每个区域实施账户，具体取决于哪种账户最适合您的业务和运营模式。如果为每个区域创建一个取证账户，就可以阻止在该区域之外创建 AWS 资源，降低资源被复制到非预期区域的风险。例如，如果您只在美国东部（弗吉尼亚州北部）区域（us-east-1）和 美国西部（俄勒冈州）（us-west-2）运营，那么您将在取证 OU 中拥有两个账户：一个用于 us-east-1，另一个用于 us-west-2。

可以为多个区域创建取证 AWS 账户。在将 AWS 资源复制到该账户时应小心谨慎，确保符合数据主权要求。由于预置新账户需要时间，因此必须在事件发生前创建和分析取证账户，以便响应人员能够做好准备，有效地使用这些账户进行响应。

下图显示了一个账户结构示例，其中包括一个取证 OU，涵盖了根据每个区域创建的取证账户：



用于响应事件而根据区域创建的账户结构

捕获备份和快照

为关键系统和数据库建立备份对于从安全事件中恢复和取证至关重要。有了备份，您就能够将系统恢复到以前的安全状态。在 AWS 上，您可以创建各种资源的快照。快照为您提供这些资源的时间点备份。有许多 AWS 服务能够在备份和恢复方面为您提供支持。有关这些服务以及备份和恢复方法的详细信息，请参阅《Backup and Recovery Prescriptive Guidance》以及《Use backups to recover from security incidents》。

特别是遇到勒索软件等情况时，妥善保护备份至关重要。有关保护备份的指导，请参阅《Top 10 security best practices for securing backups in AWS》。除了确保备份安全外，您还应当定期测试备份和还原流程，从而确保现有的技术和流程按预期运行。

自动取证

在安全事件期间，您的事件响应团队必须能够快速收集和分析证据，同时保持事件相关时间段的准确性（例如捕获与特定事件或资源相关的日志，或收集 Amazon EC2 实例的内存转储）。对于事件响应团队来说，手动收集相关证据既具有挑战性又很耗时，尤其是在存在大量实例和账户的情况下。此外，手动收集容易出现人为错误。出于这些原因，您应该尽可能开发和实现取证自动化功能。

AWS 提供了大量用于取证的自动化资源，这些资源在下面的“资源”部分中列出。这些资源是我们开发并由客户实施的取证模式示例。虽然这些资源可能是有用的参考架构，但可以考虑根据您的环境、要求、工具和取证流程对资源进行修改，或者创建新的取证自动化模式。

资源

相关文档：

- 《AWS Security Incident Response Guide》– Develop Forensics Capabilities
- 《AWS Security Incident Response Guide》– Forensics Resources
- AWS 云 中的取证调查环境策略
- 如何在 AWS 中自动实施取证磁盘收集
- AWS Prescriptive Guidance – 自动化事件响应和取证

相关视频：

- 自动化事件响应和取证

相关示例：

- 自动事件响应和取证框架
- Amazon EC2 的自动取证编排工具

SEC10-BP04 制定和测试安全事件响应行动手册

准备事件响应流程的关键环节是制定行动手册。事件响应行动手册提供了规范性指南和步骤，供发生安全事件时遵循。清晰的结构和步骤可简化响应，减少发生人为错误的可能性。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

应针对以下事件场景创建行动手册：

- 预期事件：应针对预期的事件创建行动手册。这包括拒绝服务（DoS）、勒索软件和凭证泄露等威胁。
- 已知的安全调查发现或警报：应该创建行动手册来应对已知的安全调查发现和警报，例如来自 Amazon GuardDuty 的调查发现和警报。当您收到 GuardDuty 调查发现时，行动手册应提供明确的步骤，以防止错误处理或忽略警报。有关修复措施的更多详细信息和指南，请参阅 Remediating security issues discovered by GuardDuty。

行动手册应包含安全分析师需要完成的技术步骤，以便充分调查和应对潜在的安全事件。

AWS 的客户事件响应团队（CIRT）发布了一个包含事件响应行动手册的 GitHub 存储库，而这些事件响应行动手册按威胁情景、类型和资源进行整理。这些行动手册可以进行调整，使其与现有的事件响应过程保持一致，也可以作为开发新的事件响应过程的基础。

实施步骤

行动手册中应包括的项目有：

- 行动手册概述：本行动手册针对哪些风险或事件场景？ 本行动手册的目标是什么？
- 先决条件：此事件场景需要哪些日志、检测机制和自动化工具？ 预期的通知是什么？
- 沟通和上报信息：谁参与其中，他们的联系信息是什么？ 每个利益相关方的责任是什么？
- 响应步骤：在事件响应的各个阶段，应采取哪些战术性措施？ 分析师应该进行哪些查询？ 应该运行什么代码才能达到预期的结果？
  - 检测：如何检测事件？
  - 分析：如何确定影响范围？

  - 控制：如何隔离事件来限制其影响范围？
  - 消除：如何从环境中消除威胁？
  - 恢复：受影响的系统或资源将如何恢复生产？
- 期望结果：运行查询和代码后，行动手册的期望结果是什么？

资源

相关的 Well-Architected 最佳实践：

- SEC10-BP02 – 制定事件管理计划

相关文档：

- 事件响应行动手册框架
- 制定自己的事件响应行动手册
- 事件响应行动手册样本
- 使用 Jupyter 行动手册和 CloudTrail Lake 构建 AWS 事件响应运行手册

SEC10-BP05 预置访问权限

确保事件响应者将正确的访问权限预置到 AWS 中，以缩短调查到恢复所需的时间。

常见反模式：

- 使用根账户进行事件响应。
- 变更现有账户。
- 在提供实时权限提升时直接操作 IAM 权限。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

AWS 建议尽可能减少或消除对长期有效凭证的依赖，转而使用临时凭证和实时权限提升机制。长期有效的凭证容易带来安全风险，并且会增加运营开销。对于大多数管理任务以及事件响应任务，建议您对管理访问实施身份联合验证以及临时上报。在此模型中，用户请求提升到更高级别的权限（例如事件响

应角色），如果用户符合提升条件，则会向审批者发送请求。如果请求获得批准，用户将收到一组临时的 AWS 凭证，可用于完成用户任务。在这些凭证过期后，用户必须提交新的提升请求。

在大多数事件响应场景中，建议使用临时权限提升。执行此操作的正确方法是使用 AWS Security Token Service 和会话策略来限定访问范围。

在一些场景中，联合身份不可用，例如：

- 与被盗用的身份提供者（IdP）相关的中断。
- 导致联合访问管理系统损坏的错误配置或人为错误。
- 恶意活动，例如分布式拒绝服务（DDoS，Distributed Denial of Service）事件或导致系统不可用的活动。

在上述情况下，应配置紧急 Break Glass 访问，以允许调查事件并及时给予补救。我们建议您使用具有适当权限的用户、组或角色，来执行任务和访问 AWS 资源。请仅将根用户用于需要根用户凭证的任务。要确认事件响应者对 AWS 和其他相关系统是否具有正确的访问权限级别，建议预置专用的账户。账户需要特许的访问权限，并且必须受到严格的控制和监视。在构建账户时，必须使用执行必要任务所需的最少权限，并且访问级别应基于作为事件管理计划的一部分创建的行动手册。

最好使用专门构建的专用用户和角色。通过添加 IAM 策略来临时提升用户或角色的访问权限，既会导致无法清楚地了解用户在事件期间拥有哪些访问权限，又会带来无法撤销提升的权限的风险。

请务必删除尽可能多的依赖项，以确保能在尽可能多的故障场景中获得访问权限。为了支持此操作，可创建一个行动手册，验证是否在专用的安全账户中创建事件响应用户作为用户，而不是通过任何现有的联合身份验证或单点登录（SSO）解决方案管理他们。每个响应者都必须拥有自己的指定账户。账户配置必须实施强密码策略和多重身份验证（MFA）。如果事件响应行动手册仅需要对 AWS 管理控制台 的访问权限，则用户不应配置访问密钥，并且应明确禁止用户创建访问密钥。可以使用 IAM 策略或服务控制策略（SCP，Service Control Policy）进行此配置，如 AWS 安全最佳实践（适用于 AWS Organizations SCP）中所述。用户仅能够在其他账户中代入事件响应角色，而不应具有其他任何权限。

在事件处理期间，可能需要向其他内部或外部人员授予访问权限，以支持调查、补救或恢复活动。在这种情况下，可以使用前面提到的行动手册机制，并且必须创建一个流程，确保在事件结束后立即撤消其他任何访问权限。

要确保能正确地监控和审计对事件响应角色的使用，至关重要的一点是，为此目的创建的 IAM 账户不会在人员之间共享，并且不会使用 AWS 账户根用户，除非特定任务要求这样做。如果需要根用户（例如，对特定账户的 IAM 访问权限不可用），请使用单独的流程和可用的行动手册来验证根用户登录凭证和 MFA 令牌的可用性。

要为事件响应角色配置 IAM 策略，请考虑使用 IAM Access Analyzer 来生成基于 AWS CloudTrail 日志的策略。为此，请在非生产账户中向事件响应角色授予管理员访问权限，并运行行动手册。完成后，会创建一个策略，仅允许已执行的操作。之后，可以跨所有账户将此策略应用于所有事件响应角色。您可能希望为每个行动手册创建一个单独的 IAM 策略，以便更轻松地进行管理和审计。示例行动手册可能包括针对勒索软件、数据泄露、丢失生产访问权限和其他场景的响应计划。

使用事件响应用户账户可在其他 AWS 账户 中代入专用的事件响应 IAM 角色。必须将这些角色配置为仅可由安全账户中的用户代入，并且信任关系必须要求调用主体已使用 MFA 进行身份验证。角色必须使用严格界定的 IAM 策略来控制访问。确保这些角色的所有 `AssumeRole` 请求都记录在 CloudTrail 中并发出提醒，并确保记录使用这些角色执行的任何操作。

强烈建议清楚地命名 IAM 账户和 IAM 角色，以便在 CloudTrail 日志中轻松找到他们。例如，将 IAM 账户命名为 *`<USER_ID>`*`-BREAK-GLASS`，并将 IAM 角色命名为 `BREAK-GLASS-ROLE`。

CloudTrail 用于记录 AWS 账户中的 API 活动，并且应该用于配置关于使用事件响应角色的提醒。请参阅博文，了解有关配置使用根密钥时的提醒。可以修改说明以配置 Amazon CloudWatch 指标筛选条件，从而筛选 `AssumeRole` 事件（与事件响应 IAM 角色相关）：

```
{ $.eventName = "AssumeRole" && $.requestParameters.roleArn =
 "<INCIDENT_RESPONSE_ROLE_ARN>" && $.userIdentity.invokedBy NOT EXISTS && $.eventType !
= "AwsServiceEvent" }
```

由于事件响应角色可能具有高级别的访问权限，因此，请务必将这些提醒转至广泛的群体，并及时采取适当的行动。

在事件处理期间，响应者可能需要访问不受 IAM 直接保护的系统。它们可能包括 Amazon Elastic Compute Cloud 实例、Amazon Relational Database Service 数据库或软件即服务（SaaS）平台。强烈建议不要使用 SSH 或 RDP 等本机协议，而是使用 AWS Systems Manager Session Manager 对 Amazon EC2 实例进行所有管理访问。可以使用安全且经过审计的 IAM 控制此访问。此外，还可以使用 AWS Systems Manager Run Command 文档自动实施行动手册的部分内容，这样可以减少用户出错的机会并缩短恢复时间。对于访问数据库和第三方工具，我们建议将访问凭证存储在 AWS Secrets Manager 中，并向事件响应者角色授予访问权限。

最后，事件响应 IAM 账户的管理应该添加到您的合并人员、移动人员和离开人员流程中，并定期进行检查和测试，以确认只允许预期访问。

资源

相关文档：

- 管理对 AWS 环境的临时提升的访问权限
- 《AWS Security Incident Response Guide》
- AWS Elastic Disaster Recovery
- AWS Systems Manager Incident Manager
- 为 IAM 用户设置账户密码策略
- 在 AWS 中使用多重身份验证（MFA）
- 使用 MFA 配置跨账户访问
- 使用 IAM Access Analyzer 生成 IAM 策略
- Best Practices for AWS Organizations Service Control Policies in a Multi-Account Environment
- 如何在使用 AWS 账户的根访问密钥时接收通知
- 使用 IAM 托管策略创建精细会话权限
- Break glass access

相关视频：

- 在 AWS 中自动化事件响应和取证
- 运行手册、事件报告和事件响应 DIY 指南
- 准备和响应 AWS 环境中的安全事件

SEC10-BP06 预部署工具

确保安全人员预部署了适当的工具，来缩短从调查到恢复的时间。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

要自动执行安全响应和操作功能，您可以使用 AWS 提供的一整套 API 和工具。您可以完全自动执行身份管理、网络安全、数据保护和监控功能，并使用您已采用的常见软件开发方法交付这些功能。当构建安全自动化时，您的系统可以监控、审核和启动响应，您不必安排人员监控您的安全位置并对事件做出人为响应。

如果您的事件响应团队继续以同样的方式响应警报，警报可能会让他们应接不暇。久而久之，团队对警报的敏感性可能会下降，并可能在处理正常情况时犯错或者错过异常警报。利用一些功能自动处理

重复和正常的警报，并将敏感、特殊的事件交由人员来处理，这样有助于避免疲于应对警报。集成异常检测系统（例如 Amazon GuardDuty、AWS CloudTrail Insights 和 Amazon CloudWatch Anomaly Detection）可以减轻常见阈值警报的负担。

您可以通过编程方式自动执行此流程中的步骤，从而改进手动流程。为事件定义修复模式之后，您可以将此模式分解为可执行的逻辑，并编写代码以执行此逻辑。然后，响应人员可以运行该代码来修复问题。久而久之，您就可以自动化越来越多的步骤，并最终自动处理各类常见事件。

在安全调查期间，您需要能够查看相关日志，以便记录并了解事件的来龙去脉和时间线。生成警报时也需要日志，因为日志可以指示某些相关操作已经发生。选择、启用、存储、设置查询和检索机制以及设置警报至关重要。此外，提供工具来搜索日志数据的有效方法是 Amazon Detective。

AWS 提供 200 多种云服务和数千种功能。我们建议您检查可支持和简化事件响应策略的服务。

除日志记录外，还应当制定并实施标记策略。标记有助于提供有关 AWS 资源用途的背景信息。标记也可用于实现自动化。

实施步骤

选择并设置用于分析和报警的日志

请参阅以下关于配置事件响应日志记录的文档：

- 安全事件响应的日志记录策略
- SEC04-BP01 配置服务和应用程序日志记录

启用安全服务来支持检测和响应

AWS 提供检测、预防和响应功能，而其它服务可用于构建自定义安全解决方案。有关与安全事件响应最相关的服务的列表，请参阅云功能定义和安全事件响应主页。

制定和实施标记策略

要获取围绕 AWS 资源的业务场景和相关内部利益相关方的背景信息，可能很困难。要做到这一点，可以采用标签的形式，标签为 AWS 资源分配元数据，并由用户定义的键和值组成。您可以创建标签，按照用途、所有者、环境、处理的数据类型以及您选择的其他标准对资源进行分类。

采用一致的标记策略可以加快响应速度，并通过快速识别和辨别 AWS 资源的背景信息，最大限度地减少在组织背景方面所花费的时间。标签还可以充当启动自动响应的机制。有关要标记的内容的详细信息，请参阅《标记 AWS 资源》。首先，您需要定义要在组织内实施的标签。之后，实施并强制执行这

一标记策略。有关实施和强制执行的详细信息，请参阅《使用 AWS 标签策略和服务控制策略（SCP）实施 AWS 资源标记策略》。

资源

相关的 Well-Architected 最佳实践：

- SEC04-BP01 配置服务和应用程序日志记录
- SEC04-BP02 在标准化位置收集日志、调查发现和指标

相关文档：

- 安全事件响应的日志记录策略
- 事件响应云功能定义

相关示例：

- 使用 Amazon GuardDuty 和 Amazon Detective 进行威胁检测和响应
- Security Hub 讲习会
- 使用 Amazon Inspector 进行漏洞管理

SEC10-BP07 运行模拟

随着组织不断发展壮大，威胁形势也会不断变化，因此务必要持续评估组织的事件响应能力。运行模拟（也称为实际演练）是可用于执行这种评估的一种方法。模拟过程使用现实世界中的安全事件场景，旨在模仿威胁主体采取的战术、技术和程序（TTP），让组织通过响应现实中可能发生的模拟网络事件，来练习和评估自己的事件响应能力。

建立此最佳实践的好处：模拟有多种好处：

- 检验网络准备情况，有助于事件响应人员树立信心。
- 测试工具和工作流程的准确性和有效性。
- 完善沟通和上报环节，使之与您的事件响应计划相吻合。
- 提供机会来应对不太常见的攻击载体。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

模拟主要分为三种类型：

- 桌面演练：桌面演练模拟方法是一种基于讨论的研讨会，让各个事件响应利益相关方参与进来，练习角色和职责，以及练习使用既定的沟通工具和行动手册。通常是用一整天的时间在虚拟场地和/或实地中协调完成演练。由于桌面演练以讨论为基础，因此侧重于流程、人员和协作。在讨论中，技术是必不可少的一部分，但事件响应工具或脚本的实际使用通常不包括在桌面演练中。
- 紫队演练：紫队演练可提高事件响应人员（蓝队）和模拟威胁主体（红队）之间的协作能力。蓝队由安全运营中心（SOC）的成员组成，但也可以包括在实际网络事件中会参与进来的其他利益相关方。红队由渗透测试团队或接受过攻击安全培训的关键利益相关方组成。在设计场景时，红队会与演练协调员相互协作，以确保场景的准确性与可行性。在紫队演练中，主要的关注点是支持事件响应工作的检测机制、工具和标准操作程序（SOP）。
- 红队演练：在红队演练中，进攻方（红队）模拟进行攻击，以在预定范围内实现特定目标或一系列目标。防御方（蓝队）不一定知道演练的范围和持续时间，如此，可以更真实地评估他们应对真实事件的能力。由于红队的演练可能是侵入性测试，因此务必谨慎行事，并实施控制措施，以确保该演练不会对环境造成实际破坏。

请考虑定期协调开展网络模拟。对于参与者和整个组织而言，每种演练类型都可以带来独特的好处，因此您可以选择从不太复杂的模拟类型（例如桌面演练）入手，然后再慢慢过渡到较为复杂的模拟类型（红队演练）。您应根据自身的安全成熟度、资源和期望结果选择模拟类型。由于红队演练的复杂性和成本，一些客户可能不会选择进行红队演练。

实施步骤

无论您选择哪种模拟类型，模拟通常都遵循以下实施步骤：

1. 定义核心演练要素：定义模拟场景和模拟要达成的目标。这两者都应该得到领导层的认同。
2. 确定关键利益相关方：演练至少需要演练协调员和参与者。根据具体的场景，可能会涉及其他利益相关方，例如法务、通信或行政等领域的领导层。
3. 构建和测试场景：如果有特定要素不可行，则可能需要在构建时重新定义该场景。本阶段的期望结果是最终确定的场景。
4. 协调开展模拟：采用的模拟类型决定了所需的协调工作（书面讨论场景对比高技术含量的模拟场景）。协调员应根据演练目标调整其协调战术，并应尽可能让所有演练参与者都参与进来，以实现最大利益。

5. 撰写事后报告（AAR）：确定哪些方面进展较为顺利、哪些方面需要改进以及可能存在的差距。AAR 应衡量模拟的有效性，并记录团队对模拟事件的响应情况，以便在将来的模拟中可以不断跟踪进度。

资源

相关文档：

- AWS 事件响应指南

相关视频：

- AWS 实际演练 – 安全版
- Running effective security incident response simulations

运营

“操作”是执行事件响应的核心。这是响应和修复安全事件的操作发生的地方。“操作”包括以下五个阶段：检测、分析、遏制、根除和恢复。下表中提供了这些阶段和目标的描述。

阶段	目标
检测	识别潜在的安全事件。
分析	确定安全事件是否为意外事件，并评估事件的影响范围。
遏制	尽量减小和限制安全事件的影响范围。
根除	移除与安全事件相关的未经授权的资源或构件。实施可消除安全事件的缓解措施。
恢复	将系统恢复到已知的安全状态并监控这些系统，确认威胁不会再次出现。

在应对和处理安全事件时，应将这些阶段作为指导，以便有效且可靠地进行响应。采取的实际操作会因事件而异。例如，涉及勒索软件的事件要遵循的响应步骤与涉及公共 Amazon S3 存储桶的事件不同。此外，这些阶段不一定按顺序发生。在遏制和根除之后，您可能需要重新分析，了解操作是否有效。

在人员、流程和技术方面做好充分的准备是有效运营的关键。因此，请遵循准备工作小节中的最佳实践，以便有效地应对活动的安全事件。

要了解更多信息，请参阅《AWS Security Incident Response Guide》的 Operations 小节。

事件后活动

威胁形势在不断变化，您的组织必须具备同样的动态性，才能有效保护自己的环境。持续改进的关键在于对事件和模拟的结果进行迭代，以提高有效检测、响应和调查可能的安全事件的能力，从而减少潜在漏洞，缩短响应时间，最终恢复安全运营。以下机制有助于验证您的组织是否已经准备就绪，可以利用最新的功能和知识有效应对任何情形。

最佳实践

- SEC10-BP08 建立从事件中吸取经验教训的框架

SEC10-BP08 建立从事件中吸取经验教训的框架

实现经验教训总结框架和根本原因分析能力不仅能够有助于提高事件响应能力，还有助于防止事件再次发生。通过从每次事件中吸取教训，您可以避免重复同样的错误、泄露或错误配置，这不仅可以改善您的安全态势，还可以最大限度地减少因可预防的情况而损失的时间。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

重要的是要实现一个经验教训总结框架，大体上确立并实现以下几点：

- 何时总结经验教训？
- 总结经验教训的过程涉及什么？
- 如何总结经验教训？
- 谁参与了这个过程，具体情况如何？
- 如何确定需要改进的领域？
- 如何确保有效跟踪和实施改进措施？

该框架不应关注或指责个人，而应侧重于改进工具和流程。

实施步骤

除了前面列出的大体上的成果外，重要的是要确保提出正确的问题，以便从流程中获得最大价值（可以带来切实可行的改进的信息）。请考虑以下问题，以便于您启动经验教训讨论：

- 发生了什么事件？
- 何时首次发现该事件？
- 是如何发现的？
- 哪些系统针对该活动发出了警报？
- 涉及哪些系统、服务和数据？
- 具体发生了什么？
- 哪些地方做得好？
- 哪些地方做得不好？
- 哪些流程或程序出现问题或未能扩展以应对事件？
- 以下方面有哪些地方有待改进：
  - People
    - 需要联系的人是否真的可以联系上，联系名单是否是最新名单？
    - 相应人员是否缺少有效应对和调查事件所需的培训或能力？
    - 相应的资源是否已就绪并随时可用？
  - 流程
    - 是否遵循了流程和程序？
    - 是否针对这种事件记录并提供了流程和程序？
    - 是否缺少必要的流程和程序？
    - 响应人员是否能够及时获得所需的信息来处理问题？
  - 技术
    - 现有警报系统是否能有效识别活动并发出警报？
    - 我们如何将检测时间缩短 50%？
    - 现有警报是否需要改进，或者是否需要针对这种事件设置新的警报？
    - 现有工具是否允许对事件进行有效调查（搜索/分析）？
    - 怎样才能更快地识别这种事件？
    - 如何防止这种事件再次发生？

- 谁是改进计划的负责人，如何检验改进计划的执行情况？
- 实施和测试额外监控或预防性控制机制和流程的时间表是怎样的？

此列表并非详尽无遗，但旨在作为一个起点，确定组织和业务需求是什么，以及如何分析这些需求，以便最有效地从事件中吸取经验教训，并不断改进您的安全态势。最重要的是，该列表开始将经验教训作为事件响应流程、文档和利益相关方期望的标准组成部分。

资源

相关文档：

- 《AWS Security Incident Response Guide》– Establish a framework for learning from incidents
- NCSC CAF 指南：总结经验教训

应用程序安全性

应用程序安全（AppSec）介绍了如何设计、构建和测试所开发工作负载的安全属性的整个过程。组织中应该有经过适当培训的人员，了解构建和发布基础设施的安全属性，并使用自动化来发现安全问题。

在软件开发生命周期（SDLC）和发布后流程的常规部分采用应用程序安全性测试，有助于确保您拥有一种结构化的机制来识别、修复和防止应用程序安全性问题进入生产环境。

在设计、构建、部署和操作工作负载时，应用程序开发方法应该包括安全控制机制。在此过程中，协调流程以持续减少缺陷并尽可能减少技术债务。例如，在设计阶段使用威胁建模有助于及早发现设计缺陷，这使得缺陷更易于修复，修复的成本更低，而不是等到以后再来缓解这些缺陷。

在 SDLC 中，越早的阶段，解决缺陷的成本和复杂性通常就会越低。解决问题最简单的方法就是从一开始就不要有问题，所以从威胁模型开始有助于您在设计阶段专注于实现正确的结果。随着 AppSec 计划日渐成熟，您可以增加使用自动化执行的测试数量，提高向构建者提出的反馈的准确性，并减少安全审查所需的时间。所有这些操作都可以提高所构建软件的质量，并加快将新功能推向生产环境的速度。

这些实施指南侧重于四个方面：组织和文化、管道的安全性、管道中的安全性以及依赖关系管理。每个领域提供了一组可以实施的原则，并提供了有关如何设计、开发、构建、部署和运营工作负载的端到端视图。

在 AWS 中，可以使用很多方法来处理应用程序安全计划。其中有些方法依赖于技术，而有些方法侧重于应用程序安全计划的人员和组织方面。

最佳实践

- SEC11-BP01 应用程序安全性培训
- SEC11-BP02 在整个开发和发布生命周期中执行自动化测试
- SEC11-BP03 定期执行渗透测试
- SEC11-BP04 实施代码审查
- SEC11-BP05 集中管理服务，方便获取软件包和依赖项
- SEC11-BP06 以编程方式部署软件
- SEC11-BP07 定期评测管道的安全属性
- SEC11-BP08 建立规程，让工作负载团队负责安全领域

SEC11-BP01 应用程序安全性培训

为您的团队提供有关安全开发和操作实践的培训，这有助于他们构建安全、高质量的软件。这种做法有助于您的团队在开发生命周期的早期预防、检测和修复安全问题。考虑开展涵盖威胁建模、安全编码实践以及使用服务进行安全配置和操作等方面的培训。通过自助服务资源为您的团队提供培训机会，并定期收集他们的反馈以实现持续改进。

期望结果：您为团队配备必要的知识和技能，让他们从一开始设计和构建软件时就考虑安全性。通过开展威胁建模和安全开发实践方面的培训，您的团队对潜在的安全风险以及如何在软件开发生命周期（SDLC）中缓解此类风险有了深刻的了解。这种积极主动的实现安全性的方法是团队文化的一部分，您可以尽早发现和修复潜在的安全问题。因此，您的团队可以更高效地交付高质量、安全的软件和功能，从而加快整体交付时间。您的组织内部有一种协作和包容的安全文化，在这种文化中，安全的所有权由所有构建者共享。

常见反模式：

- 您等到安全审查阶段，才考虑系统的安全属性。
- 您将所有安全决策都交给中心安全团队。
- 您没有传达在 SDLC 中做出的决策如何与组织的总体安全期望或策略相关联。
- 您过迟参与安全审查过程。

建立此最佳实践的好处：

- 在开发周期的早期更好地了解组织对安全性的要求。
- 能够更快地识别和修复潜在的安全问题，从而更快地交付功能。
- 提高软件和系统的质量。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

要构建安全且高质量的软件，请向您的团队提供培训，使其了解有关安全开发和运行应用程序的常见实践。这种实践有助于您的团队在开发生命周期的早期预防、检测和修复安全问题，从而加快交付时间。

要实现这一实践，可以考虑使用 Threat Modeling Workshop 等 AWS 资源，对您的团队进行威胁建模培训。威胁建模有助于您的团队了解潜在的安全风险，并从一开始设计系统时就考虑安全性。此外，您

还可以提供 AWS 培训和认证、行业或 AWS 合作伙伴就安全开发实践开展的培训。有关大规模设计、开发、保护和高效运营的全面方法的更多详细信息，请参阅 AWS DevOps Guidance。

明确定义并传达组织的安全审查流程，并概述您的团队、安全团队和其它利益相关方的职责。发布自助服务指南、代码示例和模板，演示如何满足您的安全要求。可以使用诸如 AWS CloudFormation、AWS Cloud Development Kit (AWS CDK) (AWS CDK) Constructs 和 Service Catalog 之类的 AWS 服务来提供预先批准的安全配置，并减少对自定义设置的需求。

定期收集团队对安全审查流程和培训体验的反馈，并利用这些反馈不断改进。开展 GameDay 或漏洞狂欢活动，以识别和解决安全问题，同时提高团队的技能。

实施步骤

1. 确定培训需求：通过调查、代码审查或与团队成员展开讨论，评测团队中有关安全开发实践的当前技能水平和知识差距。
2. 规划培训：根据确定的需求，制定涵盖诸如威胁建模、安全编码实践、安全测试和安全部署实践等相关主题的培训计划。利用 Threat Modeling Workshop、AWS 培训和认证以及行业或 AWS 合作伙伴培训计划等资源。
3. 安排和提供培训：为您的团队安排定期的培训课程或讲习会。这些课程可以由讲师指导或自定进度，具体取决于团队的偏好和空闲时间。鼓励动手练习和实际示例来增强学习效果。
4. 定义安全审查流程：与您的安全团队和其他利益相关者合作，来为您的应用程序明确定义安全审查流程。记录参与该流程的每个团队或个人的责任，包括您的开发团队、安全团队和其他相关的利益相关者。
5. 创建自助服务资源：开发自助服务指南、代码示例和模板，演示如何满足组织的安全要求。考虑使用诸如 CloudFormation、AWS CDK Constructs 和 Service Catalog 之类的 AWS 服务来提供预先批准的安全配置，并减少对自定义设置的需求。
6. 沟通和社交化：有效地向您的团队传达安全审查流程和可用的自助服务资源。举办培训课程或讲习会，让他们熟悉这些资源，并验证他们是否了解如何使用这些资源。
7. 收集反馈并改进：定期收集团队对安全审查流程和培训体验的反馈。利用这些反馈来确定需要改进的领域，并不断完善培训材料、自助服务资源和安全审查流程。
8. 开展安全练习：组织 GameDay 或漏洞狂欢活动，以识别和解决应用程序中的安全问题。这些练习不仅有助于发现潜在的漏洞，还可以为您的团队提供实践学习机会，来增强他们在安全开发和运营方面的技能。
9. 持续学习和改进：鼓励您的团队及时了解最新的安全开发实践、工具和技术。定期审核和更新您的培训材料和资源，以反映不断变化的安全形势和最佳实践。

资源

相关最佳实践：

- SEC11-BP08 建立规程，让工作负载团队负责安全领域

相关文档：

- AWS 培训 和认证
- 如何看待云安全治理
- 如何处理威胁建模
- 加速培训 – AWS Skills Guild
- AWS DevOps Sagas

相关视频：

- 主动式安全性：注意事项和方法

相关示例：

- 有关威胁建模的讲习会
- 开发人员的行业意识

相关服务：

- AWS CloudFormation
- AWS Cloud Development Kit (AWS CDK)（AWS CDK）Constructs
- 服务目录

SEC11-BP02 在整个开发和发布生命周期中执行自动化测试

在整个开发和发布生命周期中自动测试安全属性。自动化使得在发布软件前，更加容易始终如一反复识别软件中可能存在的问题，进而减少所提供的软件中存在安全问题的风险。

期望结果：自动测试的目标是提供一种程序化方式，在整个开发生命周期中常常尽早检测潜在问题。自动执行回归测试时，您可以重新运行功能测试和非功能测试，确认以前测试过的软件在更改后仍按预期

执行。定义安全性单元测试以检查常见的错误配置（如身份验证中断或缺失）时，可以在开发过程的早期识别并修复这些问题。

测试自动化根据应用程序的要求和期望的功能，使用专门构建的测试用例进行应用程序验证。自动测试的结果基于将生成的测试输出与其各自的预期输出进行比较，从而加快整个测试生命周期。回归测试和单元测试套件等测试方法最适合自动化。自动执行安全属性的测试使构建者无需等待安全审查即可自动接收反馈。静态或动态代码分析形式的自动化测试可以提高代码质量，并帮助在开发生命周期的早期检测潜在的软件问题。

常见反模式：

- 不传达自动化测试的测试用例和测试结果。
- 就在发布之前执行自动化测试。
- 使用自动化测试用例来应对经常变化的需求。
- 未能就如何处理安全测试的结果提供指导。

建立此最佳实践的好处：

- 减少对评估系统安全属性的人员的依赖。
- 在多个工作流程中得到一致的结果可提高一致性。
- 降低在生产软件中引入安全问题的可能性。
- 由于及早发现软件问题，可以缩短检测和修复之间的时间。
- 增加多个工作流中的系统或重复行为的可见性，可用于促进组织范围内的改进。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

在构建软件时，采用各种软件测试机制，以确保根据应用程序的业务逻辑测试应用程序的功能要求和非功能要求（重点关注应用程序可靠性、性能和安全性）。

静态应用程序安全性测试（SAST）分析源代码是否存在异常安全模式，并指出容易出现缺陷的代码。SAST 依赖于文档（需求规范、设计文档和设计规范）和应用程序源代码等静态输入来测试一系列已知的安全问题。静态代码分析器可以帮助加快大量代码的分析。NIST Quality Group 对源代码安全性分析器进行了比较，包括针对字节码扫描器和二进制码扫描器的开源工具。

动态分析安全测试（DAST）方法针对正在运行的应用程序执行测试，以识别潜在的意外行为，能够对静态测试作出补充。动态测试可用于检测通过静态分析无法检测到的潜在问题。通过在代码存储库、构

建和管道阶段进行测试，您可以检查出不同类型的潜在问题，防止这些问题进入到代码中。Amazon Q 开发者版在构建器的 IDE 中提供代码建议，包括安全扫描。Amazon CodeGuru 安全防御工具可以识别应用程序开发过程中的严重问题、安全问题以及难以发现的错误，并提供可提高代码质量的建议。提取软件物料清单（SBOM）还可让您提取一份正式记录，其中包含构建软件时使用的各个组件的详细信息和关系。这使您可以为漏洞管理提供信息，并快速识别软件或组件依赖关系以及供应链风险。

通过 Security for Developers 讲习会，学会使用 AWS 开发人员工具（例如，AWS CodeBuild、AWS CodeCommit 和 AWS CodePipeline）来自动实施发布管道，包括 SAST 和 DAST 测试方法。

在 SDLC 中，建立一个迭代过程，其中包括与安全团队一起定期审查应用程序。在发布准备情况审查过程中，应该处理和验证从这些安全审查中收集到的反馈。这些审查建立了健壮的应用程序安全态势，并为构建者提供切实可行的反馈，以解决潜在问题。

实施步骤

- 实施一致的 IDE、代码审查和 CI/CD 工具，其中包括安全测试。
- 考虑在 SDLC 中的哪个阶段适合阻塞管道，而不仅仅是通知构建者需要修复问题。
- Automated Security Helper (ASH) 是开源代码安全扫描工具的一个示例。
- 使用自动化工具（例如，与开发人员 IDE 集成的 Amazon Q 开发者版，以及用于在提交时扫描代码的 Amazon CodeGuru 安全防御工具）执行测试或代码分析，有助于构建者适时获得反馈。
- 使用 AWS Lambda 构建应用程序时，您可以使用 Amazon Inspector 来扫描函数中的应用程序代码。
- 当 CI/CD 管道中包括自动化测试时，您应该使用工单系统来跟踪软件问题的通知和修正。
- 对于可能生成结果的安全测试，链接到补救指南可帮助构建者提高代码质量。
- 定期分析使用自动化工具获得的结果，以确定下一个自动化、构建者培训或认知宣传活动的优先级。
- 要提取 SBOM 作为 CI/CD 管道的一部分，请使用 Amazon Inspector SBOM Generator，来为 CycloneDX SBOM 格式的归档、容器映像、目录、本地系统以及编译后的 Go 和 Rust 二进制文件生成 SBOM。

资源

相关最佳实践：

- DevOps Guidance: DL.CR.3 Establish clear completion criteria for code tasks

相关文档：

- 持续交付和持续部署

- AWS DevOps 能力合作伙伴
- 适用于应用程序安全性的 AWS 安全能力合作伙伴
- 选择 Well-Architected CI/CD 方法
- Secrets detection in Amazon CodeGuru Security
- Amazon CodeGuru Security Detection Library
- 通过有效的治理加快 AWS 上的部署
- AWS 方法如何自动实现无需干预的安全部署
- How Amazon CodeGuru Security helps you effectively balance security and velocity

相关视频：

- 无需干预：在亚马逊自动实现持续交付管道
- 自动执行跨账户 CI/CD 管道
- The Software Development Prcess at Amazon
- Testing software and systems at Amazon

相关示例：

- 开发人员的行业意识
- Automated Security Helper (ASH)
- AWS CodePipeline Governance - Github

SEC11-BP03 定期执行渗透测试

定期对软件执行渗透测试。此机制有助于识别无法通过自动化测试或人工代码审查加以检测的潜在软件问题。它还有助于您了解检测控制的有效性。渗透测试应设法确定软件是否会以意外方式执行，例如公开应受保护的数据，或者授予比预期更广泛的权限。

期望结果：使用渗透测试来检测、修复和验证应用程序的安全属性。在软件开发生命周期（SDLC）中应定期执行计划的渗透测试。在发布软件之前应处理渗透测试的结果。您应该分析渗透测试的结果，以

确定是否存在使用自动化可以发现的问题。拥有包括主动反馈机制的定期且可重复渗透测试流程，有助于为构建者提供指导并提高软件质量。

常见反模式：

- 仅对已知或普遍存在的安全问题进行渗透测试。
- 未使用相关的第三方工具和库对应用程序执行渗透测试。
- 仅对软件包安全问题进行渗透测试，而不评估已实施的业务逻辑。

建立此最佳实践的好处：

- 在发布之前增强对软件安全属性的信心。
- 有机会确定首选的应用程序模式，从而提高软件质量。
- 获得一个反馈环路，在开发周期早期确定自动化或额外培训可以在哪些方面改进软件的安全属性。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

渗透测试是一项结构化安全测试练习，让您可以运行计划的安全漏洞方案，以便检测、修复和验证安全控制机制。渗透测试从侦察开始，在这个过程中，根据应用程序的当前设计及其依赖项收集数据。生成并运行特定于安全方面的测试方案的精选列表。这些测试的主要目的是发现应用程序中的安全问题，有人会利用这些安全问题来获得对环境的非预期访问，或未经授权访问数据。当推出新功能时，或者应用程序的功能或技术实施方面发生重大变更时，您应该执行渗透测试。

您应该确定在开发生命周期的哪个阶段执行渗透测试最为合适。应当尽量晚些时候执行此测试，以便系统功能接近预期的发布状态，但也要留有足够的时间来修复任何问题。

实施步骤

- 采用结构化流程来确定渗透测试的范围，让这个流程基于威胁模型是保留场景相关性的好方法。
- 确定在开发周期的什么阶段执行渗透测试较为合适。这个阶段应该是在应用程序预期改动很细微，但仍留有足够时间进行修复的时候。
- 为构建者提供以下方面的培训：从渗透测试结果中可以期待获得什么，以及如何获得有关修复的信息。
- 使用工具自动执行常见或可重复的测试，从而加快渗透测试的速度。

- 分析渗透测试结果，以便确定系统性安全问题，并使用此数据为额外的自动化测试和正在进行的构建者培训提供信息。

资源

相关最佳实践：

- SEC11-BP01 应用程序安全性培训
- SEC11-BP02 在整个开发和发布生命周期中执行自动化测试

相关文档：

- AWS 渗透测试提供有关 AWS 上的渗透测试的详细指导
- 通过有效的治理加快 AWS 上的部署
- AWS 安全能力合作伙伴
- 使 AWS Fargate 上的渗透测试架构实现现代化改造
- AWS Fault injection Simulator

相关示例：

- 使用 AWS CodePipeline 自动执行 API 测试（GitHub）
- 自动安全助手（GitHub）

SEC11-BP04 实施代码审查

实施代码审查来协助验证正在开发的软件的质量和安全性。代码审查包括让除原始代码作者之外的团队成员审查代码是否存在潜在问题、漏洞，以及是否遵守编码标准和最佳实践。此过程有助于发现原始开发人员可能已忽略的错误、不一致和安全漏洞。运用自动化工具来协助进行代码审查。

期望结果：在开发过程中纳入代码审查，以提高正在编写的软件的质量。您可以通过在代码审查期间确定的经验教训，来提高团队中经验不足的成员的技能。您可以使用自动化工具和测试来发现自动化的机会并支持代码审查流程。

常见反模式：

- 您在部署前不执行代码审查。

- 您让同一个人编写和审查代码。
- 您不使用自动化和工具来协助或编排代码审查。
- 在构建者审查代码之前，您未对他们进行应用程序安全方面的培训。

建立此最佳实践的好处：

- 提高代码质量。
- 通过重复利用通用方法提高代码开发的一致性。
- 减少在渗透测试和后续阶段发现的问题。
- 改进团队内部的知识传授。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

代码审查有助于在开发过程中验证软件的质量和安全性。手动审查包括让除原始代码作者之外的团队成员审查代码是否存在潜在问题、漏洞，以及是否遵守编码标准和最佳实践。此过程有助于发现原始开发人员可能已忽略的错误、不一致和安全漏洞。

考虑使用 Amazon CodeGuru Security 来协助实施自动代码审查。CodeGuru 安全防御工具使用机器学习和自动推理来分析您的代码，并识别潜在的安全漏洞和编码问题。将自动化代码审查与现有代码存储库以及持续集成/持续部署（CI/CD）管道相集成。

实施步骤

1. 建立代码审查流程：
   - 定义何时应进行代码审查，例如，在将代码合并到主分支之前或部署到生产环境之前。
   - 确定谁应该参与代码审查流程，例如团队成员、高级开发人员和安全专家。
   - 决定代码审查方法，包括要使用的流程和工具。
2. 设置代码审查工具：
   - 评估并选择符合团队需求的代码审查工具，例如 GitHub 拉取请求或 CodeGuru 安全防御工具。
   - 将所选工具与您现有的代码存储库和 CI/CD 管道相集成。
   - 配置工具来强制实施代码审查要求，例如最少审查人员数量和审批规则。
3. 定义代码审查清单和指南：

- 创建代码审查清单或指南，其中概述了应审查的内容。考虑诸如代码质量、安全漏洞、对编码标准的遵守情况以及性能等因素。
- 与开发团队共享清单或指南，并确认每个人都了解预期结果。

4. 对开发人员进行代码审查最佳实践培训：
   - 为您的团队提供有关如何进行有效代码审查的培训。
   - 让您的团队了解应用程序安全原则和在审查期间需要注意的常见漏洞。
   - 鼓励开展知识共享和配对编程课程，以提高经验不足的团队成员的技能。
5. 实施代码审查流程：
   - 将代码审查步骤集成到开发工作流程中，例如创建拉取请求和分配审查人员。
   - 要求在合并或部署之前对代码更改进行代码审查。
   - 鼓励在审查过程中进行开放式沟通和提供建设性反馈。
6. 监控和改进：
   - 定期检查代码审查流程的有效性，并收集团队的反馈。
   - 确定实施自动化或工具改进的机会，以简化代码审查流程。
   - 根据经验教训和行业最佳实践，不断更新和完善代码审查清单或指南。
7. 培养代码审查文化：
   - 强调代码审查对于维护代码质量和安全性的重要性。
   - 庆祝代码审查流程取得成功和获得经验教训。
   - 鼓励营造一个协作和支持的环境，让开发人员能够舒适地提出和接受反馈。

资源

相关最佳实践：

- SEC11-BP02 在整个开发和发布生命周期中执行自动化测试

相关文档：

- DevOps Guidance: DL.CR.2 Perform peer review for code changes
- 关于 GitHub 中的拉取请求

相关示例：

- Automate code reviews with Amazon CodeGuru Security
- Automating detection of security vulnerabilities and bugs in CI/CD pipelines using Amazon CodeGuru Security CLI

相关视频：

- Continuous improvement of code quality with Amazon CodeGuru Security

SEC11-BP05 集中管理服务，方便获取软件包和依赖项

提供集中式服务，方便您的团队获取软件包和其它依赖项。通过采取这种做法，可以在将软件包纳入所编写的软件之前，对软件包进行验证；另外，还可以为分析贵组织所使用的软件提供数据来源。

期望结果：除了您编写的代码之外，还可以使用外部软件包来构建工作负载。这使您更容易实现重复使用的功能，例如 JSON 解析器或加密库。您将这些软件包和依赖项的来源集中在一起，以便安全团队可以在使用软件包和依赖项之前对其来源进行验证。您将此方法与手动和自动测试流程结合使用，来增强对所开发软件的质量的信心。

常见反模式：

- 您从互联网上的任意存储库中提取软件包。
- 您在将新软件包提供给构建者之前，未对其进行测试。

建立此最佳实践的好处：

- 更好地了解正在构建的软件中使用了哪些软件包。
- 了解谁使用了哪些软件包后，在需要更新软件包时，能够向工作负载团队发出通知。
- 降低软件中存在问题软件包的风险。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

以构建者易于使用的方式为软件包和依赖项提供集中管理服务。集中管理服务可以在逻辑上集中，而不用作为一个整体系统来实施。利用此方法，您可以通过满足构建者需求的方法来提供服务。您应该实施一种有效的方法：在发生更新或出现新需求时将软件包添加到存储库。AWS CodeArtifact 等 AWS 服务或类似的 AWS 合作伙伴解决方案提供了一种实现此功能的方法。

实施步骤

- 实施可在用于开发软件的所有环境中使用的逻辑集中式存储库服务。
- 在 AWS 账户 分配过程中包括对存储库的访问权限。
- 构建自动化以在存储库中发布软件包之前对其进行测试。
- 维护最常用软件包、语言和更改量最大的团队的指标。
- 为构建者团队提供一种自动化机制来请求新软件包和提供反馈。
- 定期扫描存储库中的软件包，以确定新发现的问题的潜在影响。

资源

相关最佳实践：

- SEC11-BP02 在整个开发和发布生命周期中执行自动化测试

相关文档：

- DevOps Guidance: DL.CS.2 Sign code artifacts after each build
- Supply chain Levels for Software Artifacts (SLSA)

相关示例：

- 通过有效的治理加快 AWS 上的部署
- 使用 CodeArtifact Package Origin Control 工具包加强软件包的安全性
- 多区域软件包发布管道（GitHub）
- 使用 AWS CodePipeline 在 AWS CodeArtifact 上发布 Node.js 模块（GitHub）
- AWS CDK Java CodeArtifact 管道示例（GitHub）
- 使用 AWS CodeArtifact 分发专用 .NET NuGet 包（GitHub）

相关视频：

- 主动式安全性：注意事项和方法
- AWS 安全理念（re:Invent 2017）
- 当安全、保障和紧迫性都很重要时：处理 Log4Shell

SEC11-BP06 以编程方式部署软件

尽可能以编程方式部署软件。通过采取这种做法，可以降低由于人为错误导致部署失败或引入意外问题的可能性。

期望结果：您测试的工作负载版本就是您部署的版本，每次都一致地执行部署。您可以将工作负载的配置外部化，这有助于您无需更改即可部署到不同的环境。您使用软件包的加密签名来验证环境之间没有任何变化。

常见反模式：

- 手动将软件部署到生产环境中。
- 手动对软件进行更改，以适应不同的环境。

建立此最佳实践的好处：

- 增强对软件发布过程的信心。
- 降低了失败的更改对业务功能造成影响的风险。
- 由于更改风险降低，从而加快了发布节奏。
- 针对部署过程中的意外事件的自动回滚功能。
- 能够以加密方式证明所测试的软件是部署的软件。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

要维护稳健且可靠的应用程序基础设施，应实施安全和自动化的部署实践。这种做法涉及从生产环境中移除持久的人员访问权限，使用 CI/CD 工具进行部署，以及将特定于环境的配置数据外部化。通过采用这种方法，您可以增强安全性，降低人为错误的风险，并简化部署流程。

您可以构建自己的 AWS 账户结构，以便从生产环境中移除持久的人员访问权限。这种做法可以最大限度地降低未经授权的更改或意外修改的风险，从而提高生产系统的完整性。您可以使用 AWS CodeBuild 和 AWS CodePipeline 之类的 CI/CD 工具来执行部署，而不是使用直接人员访问权限。您可以使用这些服务来自动执行构建、测试和部署流程，从而减少手动干预并提高一致性。

为了进一步增强安全性和可追溯性，您可以在测试应用程序包后对其进行签名，并在部署期间验证这些签名。为此，请使用诸如 AWS Signer 或 AWS Key Management Service（AWS KMS）之类的加密工具。通过对软件包进行签名和验证，您可以确保仅将经过授权和验证的代码部署到您的环境中。

此外，您的团队可以适当地设计工作负载，以便从外部源（例如 AWS Systems Manager Parameter Store）获得特定于环境的配置数据。这种做法可将应用程序代码与配置数据分开，这有助于您独立管理和更新配置，而无需修改应用程序代码本身。

为简化基础设施的预置和管理，可以考虑使用基础设施即代码（IaC）工具，例如 AWS CloudFormation 或 AWS CDK。可以使用这些工具来定义基础设施即代码，从而提高不同环境中部署的一致性和可重复性。

考虑使用金丝雀部署来验证软件是否成功部署。金丝雀部署涉及在部署到整个生产环境之前，先对一部分实例或用户推出更改。然后，您可以监控更改的影响并在必要时进行回滚，从而最大限度地降低出现广泛问题的风险。

按照 Organizing Your AWS Environment Using Multiple Accounts 白皮书中列出的建议进行操作。本白皮书提供了有关将环境（例如开发、暂存和生产）分离到不同 AWS 账户的指导，这种划分可以进一步增强安全性和隔离性。

实施步骤

1. 设置 AWS 账户结构：
   - 按照 Organizing Your AWS Environment Using Multiple Accounts 白皮书中的指导，为不同的环境（例如开发、暂存和生产）创建单独的 AWS 账户。
   - 为每个账户配置适当的访问控制和权限，以限制人员对生产环境的直接访问权限。
2. 实施 CI/CD 管道：
   - 使用 AWS CodeBuild 和 AWS CodePipeline 之类的服务设置 CI/CD 管道。
   - 将管道配置为自动构建、测试应用程序代码，并将其部署到相应的环境。
   - 将代码存储库与 CI/CD 管道相集成，以实现版本控制和代码管理。
3. 签署并验证应用程序包：
   - 在应用程序包经过测试和验证后，使用 AWS Signer 或 AWS Key Management Service（AWS KMS）对其进行签名。
   - 在将应用程序包部署到目标环境之前，配置部署过程来验证应用程序包的签名。
4. 使配置数据外部化：
   - 将特定于环境的配置数据存储在 AWS Systems Manager Parameter Store 中。
   - 修改应用程序代码，以便在部署或运行时期间从 Parameter Store 检索配置数据。
5. 实施基础设施即代码（IaC）：
   - 使用 AWS CloudFormation 或 AWS CDK 之类的 IaC 工具来定义和管理基础设施即代码。

- 创建 CloudFormation 模板或 CDK 脚本，来为应用程序预置和配置必要的 AWS 资源。
- 将 IaC 与 CI/CD 管道集成，以便在应用程序代码更改的同时自动部署基础设施变更。

6. 实施金丝雀部署：
   - 将您的部署流程配置为支持金丝雀部署，也即，在将更改部署到整个生产环境之前，先向一部分实例或用户推出这些更改。
   - 使用诸如 AWS CodeDeploy 或 AWS ECS 之类的服务来管理金丝雀部署并监控更改的影响。
   - 实施回滚机制，如果在金丝雀部署期间检测到问题，则还原到之前的稳定版本。
7. 监控和审计：
   - 设置监控和日志记录机制，来跟踪部署、应用程序性能和基础设施更改。
   - 使用诸如 Amazon CloudWatch 和 AWS CloudTrail 之类的服务来收集和分析日志和指标。
   - 实施审计和合规性检查，以验证对安全最佳实践和监管要求的遵守情况。
8. 持续改进：
   - 定期审查和更新您的部署实践，并纳入从以前的部署中吸取的反馈和经验教训。
   - 尽可能地实现部署过程自动化，以减少手动干预和潜在的人为错误。
   - 与跨职能团队（例如运营或安全）合作，来协调并持续改进部署实践。

通过执行这些步骤，可以在 AWS 环境中实施安全的自动化部署实践，从而增强安全性，降低人为错误的风险，并简化部署过程。

资源

相关最佳实践：

- SEC11-BP02 在整个开发和发布生命周期中执行自动化测试
- DL.CI.2 Trigger builds automatically upon source code modifications

相关文档：

- 通过有效的治理加快 AWS 上的部署
- 自动实现无需干预的安全部署
- Code signing using AWS Certificate Manager Private CA and AWS Key Management Service asymmetric keys
- Code Signing, a Trust and Integrity Control for AWS Lambda

相关视频：

- 无需干预：在亚马逊自动实现持续交付管道

相关示例：

- Blue/Green deployments with AWS Fargate

SEC11-BP07 定期评测管道的安全属性

对您的管道运用 Well-Architected 安全性支柱原则，尤其注意权限分离。定期评测管道基础设施的安全属性。通过有效管理管道的安全性，可以确保通过管道的软件的安全性。

期望结果：用于构建和部署软件的管道遵循与环境中任何其它工作负载相同的建议做法。您在管道中实施的测试不可由使用这些测试的团队编辑。您只向管道授予它们使用临时凭证执行的部署所需的权限。您可以实施安全措施来防止将管道部署到错误的环境中。您可以将管道配置为发出状态，以便可以验证构建环境的完整性。

常见反模式：

- 构建者可以绕过安全测试。
- 用于部署管道的权限过于宽松。
- 未将管道配置为验证输入。
- 不定期审查与 CI/CD 基础设施关联的权限。
- 使用长期或硬编码凭证。

建立此最佳实践的好处：

- 对通过管道构建和部署的软件的完整性有了更大的信心。
- 在出现可疑活动时可以停止部署。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

部署管道是软件开发生命周期的关键组成部分，应遵循与环境中任何其它工作负载相同的安全原则和做法。这包括实施适当的访问控制、验证输入，以及定期审查和审计与 CI/CD 基础设施关联的权限。

确认负责构建和部署应用程序的团队无法编辑或绕过在管道中实施的安全测试和检查。这种关注点分离有助于维护构建和部署过程的完整性。

首先，可以考虑使用 AWS Deployment Pipelines Reference Architecture。此参考架构为在 AWS 上构建 CI/CD 管道提供了安全且可扩展的基础。

此外，可以使用 AWS Identity and Access Management Access Analyzer 等服务为管道权限生成最低权限 IAM 策略，并作为管道中验证工作负载权限的一个步骤。这有助于验证管道和工作负载是否仅具有其特定功能所需的必要权限，从而降低未经授权的访问或操作的风险。

实施步骤

- 从 AWS 部署管道参考架构开始。
- 考虑使用 AWS IAM Access Analyzer 以编程方式生成管道的最低权限 IAM 策略。
- 将管道与监控和警报集成在一起，以便在发生意外或异常活动时您会得到通知，对于 AWS 托管式服务，Amazon EventBridge 支持您将数据路由到目标，例如 AWS Lambda 或 Amazon Simple Notification Service（Amazon SNS）。

资源

相关文档：

- AWS 部署管道参考架构
- 监控 AWS CodePipeline
- 的安全最佳实践AWS CodePipeline

相关示例：

- DevOps 监控控制面板（GitHub）

SEC11-BP08 建立规程，让工作负载团队负责安全领域

建立规程或机制，使构建者团队能够针对创建的软件作出安全决策。这些决策仍然需要由安全团队通过审查加以验证，但让构建者团队负责安全领域可以构建速度更快、安全性更高的工作负载。此机制还可促进负责任文化，进而对所构建系统的运营产生积极影响。

期望结果：您的团队中嵌入了安全所有权和决策权。您要么已经对团队进行了如何考虑安全的培训，要么已通过内嵌或关联的安全人员增强了团队。因此，您的团队在开发周期的早期就做出了更高质量的安全决策。

常见反模式：

- 将所有安全设计决策交给安全团队。
- 在开发过程中没有及早满足安全要求。
- 没有从构建者和安全人员那里获得关于计划运营的反馈。

建立此最佳实践的好处：

- 缩短完成安全审查的时间。
- 减少等到安全审查阶段才检测到安全问题的情况。
- 提高所编写软件的整体质量。
- 有机会识别和了解系统性问题或高价值改进领域。
- 进行安全审查后，发现的问题可以在早期进行修复，从而减少所需的返工量。
- 提升对安全功能的认知。

在未建立这种最佳实践的情况下暴露的风险等级：低

实施指导

从 SEC11-BP01 应用程序安全性培训 中的指导开始。然后确定您认为可能最适合您组织的计划的运营模式。两个主要模式是对构建者进行培训，或在构建者团队中加入安全人员。确定初始方法后，应使用单个或一小组工作负载团队进行试点，以证明该模式适用于您的组织。来自组织的构建者和安全团队的领导层支持有助于计划的成功交付。在构建此计划时，重要的是选择可以用来显示项目价值的指标。了解 AWS 如何解决这个问题是一个很好的学习经验。这个最佳实践非常注重组织变革和文化。您使用的工具应支持构建者和安全社区之间的协作。

实施步骤

- 首先对构建者进行应用程序安全性培训。
- 创建一个社区和入门培训计划来对构建者进行培训。
- 为计划选择一个名称。通常使用守护者、拥护者或倡导者。
- 确定要使用的模式：培训构建者、加入安全工程师或具有相关性安全角色。

- 从安全性、构建者和可能的其他相关团体中确定项目发起人。
- 跟踪参与计划的人数、审查所花时间以及来自构建者和安全人员的反馈等指标。使用这些指标来作出改进。

资源

相关最佳实践：

- SEC11-BP01 应用程序安全性培训
- SEC11-BP02 在整个开发和发布生命周期中执行自动化测试

相关文档：

- 如何处理威胁建模
- 如何看待云安全治理
- How AWS built the Security Guardians program, a mechanism to distribute security ownership
- How to build a Security Guardians program to distribute security ownership

相关视频：

- 主动式安全性：注意事项和方法
- AppSec tooling and culture tips from AWS and Toyota Motor North America

结论

保障安全性是一项持续性的工作。只要发生了事件，就应将其转化为提高架构安全性的机会。拥有强大的身份控制、自动响应安全事件、在多个级别保护基础设施以及通过加密管理合理分类的数据，可以提供每个组织都应实施的深度防御。借助本白皮书中讨论的编程函数以及 AWS 功能和服务，您可以更加轻松地执行这项工作。

AWS 致力于帮助您构建和运营既能保护信息、系统和资产，又能提供业务价值的架构。

贡献者

以下个人和组织参与了本文档的编撰：

- Jay Michael，Amazon Web Services 首席安全主管解决方案架构师
- Kiaan Sumeeet，Amazon Web Services 首席安全顾问
- Michael Fischer，Amazon Web Services 首席解决方案架构师
- Conor Colgan，Amazon Web Services 首席解决方案架构师
- Dave Walker，Amazon Web Services 安全性和合规性业务首席解决方案架构师
- Patrick Palmer，Amazon Web Services 安全性和合规性业务首席解决方案架构师
- Monka Vu Minh，Amazon Web Services 安全顾问
- Kurt Kumazon，Amazon Web Services 安全顾问
- Fahima Khan，Amazon Web Services 安全解决方案架构师
- Mutaz Hajeer，Amazon Web Services 高级安全解决方案架构师
- Luis Pastor，Amazon Web Services 高级安全解决方案架构师
- Colin Igbokwe,，Amazon Web Services 高级安全解决方案架构师
- Geoff Sweet，Amazon Web Services 高级安全解决方案架构师
- Anthony Harvey，Amazon Web Services 高级安全解决方案架构师
- Sowjanya Rajavaram，Amazon Web Services 高级安全解决方案架构师
- Krishna Prasad，Amazon Web Services 高级解决方案架构师
- Faisal Farooq，Amazon Web Services 高级解决方案架构师
- Arun Krishnaswamy，Amazon Web Services 高级解决方案架构师
- Dan Girard，Amazon Web Services 高级解决方案架构师
- Marc Luescher，Amazon Web Services 高级解决方案架构师
- Kyle Nicodemus，Amazon Web Services 高级技术客户经理
- Irina Szabo，Amazon Web Services 高级技术客户经理
- Arun Sivaraman，Amazon Web Services 解决方案架构师
- Stephen Novak，Amazon Web Services 技术客户经理
- Jonathan Risbrook，Amazon Web Services 技术客户经理
- Freddy Kasprzykowski，Amazon Web Services 全球金融服务业务经理
- Pat Gaw，Amazon Web Services 首席安全顾问

- Jason Garman，Amazon Web Services 首席安全解决方案架构师
- Mark Keating，Amazon Web Services 首席安全解决方案架构师
- Zach Miller，Amazon Web Services 首席安全解决方案架构师
- Maitreya Ranganath，Amazon Web Services 首席安全解决方案架构师
- Reef Dsouza，Amazon Web Services 首席解决方案架构师
- Brad Burnett，Amazon Web Services 安全解决方案架构师
- Matt Saner，Amazon Web Services 安全解决方案架构高级经理
- Priyank Ghedia，Amazon Web Services 高级安全解决方案架构师
- Arthur Mnev，Amazon Web Services 高级安全解决方案架构师
- Kyle Dickinson，Amazon Web Services 高级安全解决方案架构师
- Kevin Boland，Amazon Web Services 高级安全解决方案架构师
- Anna McAbee，Amazon Web Services 高级安全解决方案架构师
- Recep Meric Degirmenci，Amazon Web Services 高级安全解决方案架构师
- Daniel Salzedo，Amazon Web Services 高级安全技术产品经理
- Jake Izumi，Amazon Web Services 高级解决方案架构师
- Bert Bullough，Amazon Web Services 高级解决方案架构师
- Robert McCall，Amazon Web Services 解决方案架构师
- Angela Chao，Amazon Web Services AWS Enterprise Support 业务 ESL TAM
- Pratima Singh，Amazon Web Services ANZ 安全规范高级解决方案架构师
- Darran Boyd，Amazon Web Services AWS 安全部门 CISO 办公室首席负责人
- Byron Pogson，Amazon Web Services 高级安全解决方案架构师

延伸阅读

如需更多帮助，请查阅以下资源：

- AWS Well-Architected Framework 白皮书
- AWS 架构中心

文档修订

如需获取有关该白皮书更新的通知，请订阅 RSS 信息源。

变更	说明	日期
更新了最佳实践指南	根据以下领域的新指导更新了最佳实践：SEC 2、SEC 3、SEC 4、SEC 6、SEC 7、SEC 8、SEC 9、SEC 10 和 SEC 11 整个支柱的指导均已更新和完善。	2024 年 11 月 6 日
更新了最佳实践指南	对各支柱进行了大规模的最佳实践更新。对多个最佳实践进行了整合和重新排序。对 SEC 1、4、5、6、7、8、9 进行了重大变更。	2024 年 6 月 27 日
更新了最佳实践指南	根据以下领域的新指导更新了最佳实践：安全地运营工作负载和保护传输中数据。	2023 年 12 月 6 日
更新了最佳实践指南	对事件响应的指导和最佳实践进行了重大更新。 对准备工作的多个最佳实践进行了更新。“事件响应”增加了两个新领域：运营和事后活动。增加了新的最佳实践SEC10-BP08 建立从事件中吸取经验教训的框架。	2023 年 10 月 3 日
更新了最佳实践指南	根据以下领域的新指导更新了最佳实践：准备和模拟。	2023 年 7 月 13 日
针对新框架进行了更新。	为最佳实践更新了规范性指南并增加了新的最佳实践。添加	2023 年 4 月 10 日

	了应用程序安全性（AppSec）的新最佳实践领域。	
已更新白皮书	为最佳实践更新了新的实施指导。	2022 年 12 月 15 日
已更新白皮书	扩展了最佳实践并增加了改进计划。	2022 年 10 月 20 日
次要更新	更新了 IAM 信息来反映最新的最佳实践。	2022 年 6 月 28 日
次要更新	添加了其他 AWS PrivateLink 信息，并修复了失效链接。	2022 年 5 月 19 日
次要更新	增加了 AWS PrivateLink。	2022 年 5 月 6 日
次要更新	删除了非包容性用语。	2022 年 4 月 22 日
次要更新	添加了有关 VPC 网络访问分析器的信息。	2022 年 2 月 2 日
次要更新	修复了失效链接。	2021 年 5 月 27 日
次要更新	对全文进行了编辑性修改。	2021 年 5 月 17 日
主要更新	添加了治理小节，为各个小节添加了详细信息，并添加了新功能和服务。	2021 年 5 月 7 日
次要更新	更新了链接。	2021 年 3 月 10 日
次要更新	修复了失效链接。	2020 年 7 月 15 日
针对新框架进行了更新	已更新有关账户、身份和权限管理的指导。	2020 年 7 月 8 日
针对新框架进行了更新	针对扩展每个方面的建议、新的最佳实践、服务和功能进行了更新。	2020 年 4 月 30 日

已更新白皮书	反映新的 AWS 服务和功能以及最新参考的更新。	2018 年 7 月 1 日
已更新白皮书	更新了“系统安全配置和维护”小节来反映新的 AWS 服务和功能。	2017 年 5 月 1 日
初次发布	已发布安全性支柱 – AWS Well-Architected Framework。	2016 年 11 月 1 日

版权声明

客户有责任对本文档中的信息进行单独评测。本文档：(a) 仅供参考，(b) 代表当前的 AWS 产品和实践，如有更改，恕不另行通知，以及 (c) 不构成 AWS 及其附属公司、供应商或许可方的任何承诺或保证。AWS 产品或服务“按原样”提供，不附带任何明示或暗示的保证、陈述或条件。AWS 对其客户承担的责任和义务受 AWS 协议制约，本文档不是 AWS 与客户直接协议的一部分，也不构成对该协议的修改。

AWS 术语表

有关最新的 AWS 术语，请参阅 AWS 词汇表 参考中的 AWS 词汇表。