aws

用户指南

AWS 安全代理

AWS 安全代理: 用户指南

Table of Contents

简介

了解 AWS 安全代理的功能、其工作原理以及其资源是如何组合在一起的。

什么是 AWS 安全代理？

AWS Security Agent 是一种边境代理，可在整个开发生命周期中主动保护您的应用程序。它可以根据您的组织要求进行量身定制的自动安全审查，构建威胁模型以确定您的应用程序可能如何受到攻击，并按需提供情境感知渗透测试。通过持续验证从设计到部署的安全性，AWS Security Agent 可帮助您尽早防范所有环境中的漏洞。

安全团队只需在 AWS 控制台中即可定义组织安全要求：经批准的授权库、日志标准和数据访问策略。AWS Security Agent 会在整个开发过程中自动执行这些安全要求，根据您的标准评估架构文档和代码，并在发现违规行为时提供具体指导。这为所有团队的设计和代码审查提供了一致的安全执行。

为了进行部署验证，AWS Security Agent 将渗透测试从周期性的瓶颈转变为按需功能。安全团队提供目标 URL、身份验证详情、源代码和应用程序文档。AWS Security Agent 可以深入了解应用程序，并执行复杂的攻击链来发现和验证漏洞，使团队能够在需要时进行测试。

关键功能

AWS 安全代理提供涵盖整个开发生命周期的全面安全功能。

设计安全审查

AWS Security Agent 通过提供有关设计文档的实时安全反馈并在编写任何代码之前评估是否符合组织安全要求来转移安全性。安全团队通过 Web 应用程序上传文档并获得补救指导，以确定发现的优先顺序，从而将耗时的人工审查加快为有针对性的分析。通过主动将您的安全标准嵌入到每一次设计审查中，您可以减少后期架构返工，并与多个开发团队保持同步。

威胁建模

AWS Security Agent 会为您的应用程序构建威胁模型，以识别它们可能受到攻击的方式。提供技术设计文档（范围文档）以定义代理关注的内容，提供源代码作为代理了解您现有系统的上下文，或者两者兼而有之。AWS Security Agent 会生成系统概述，描述您的应用程序的架构、组件、信任边界、数据流和安全状况，以及按 STRIDE 类别（欺骗、篡改、拒绝、信息泄露、拒绝服务、权限提升）分类的一组威胁，并附有严重性级别和可操作的建议。威胁模型是可重复使用的配置，您可以随着代码和设计的发展而重新运行这些配置，从而在整个开发生命周期中进行迭代安全评估。

代码安全审查

AWS 安全代理通过两种互补的方法主动保护应用程序。首先，您可以在 Web 应用程序中创建代码审查，对来自、、Bitbucket GitHub GitLab、E GitHub nterprise Server 存储库或 S3 存储桶的完整源代码进行全面扫描，识别安全漏洞并验证整个代码库是否符合组织要求。其次，您可以为连接的存储库启用自动拉取请求分析，AWS Security Agent 会审查代码更改并将安全发现直接作为拉取请求或合并请求注释发布。在这两种情况下，开发人员都会收到补救指导，AWS Security Agent 可以自动生成拉取请求或将请求与已识别漏洞的代码修复合并。这在所有存储库中嵌入了安全专业知识，从而减少了开发管道中与安全相关的延迟，并扩展了所有代码库的评估。

渗透测试

AWS Security Agent 通过部署专门的 AI 代理来提供按需渗透测试，通过量身定制的多步骤攻击场景来发现、验证、报告和修复安全漏洞。代理通过分析源代码和文档来识别和利用自动安全扫描工具无法发现的漏洞，从而了解应用程序的上下文。它通过影响分析、可重现的攻击路径记录调查结果，并使用随时可实施的代码修复创建拉取请求，将定期评估转化为持续验证，可以扩展到所有应用程序，而不是仅限于关键应用程序。

优势

On-demand 可访问性

AWS 安全代理提供对安全专业知识的即时访问权限。开发团队可以在需要时按需运行设计审查、威胁模型、代码分析和渗透测试，以适应现代开发周期的节奏，并在每个阶段实现主动安全。

自动验证组织需求

在 AWS 控制台中定义组织的安全要求一次。在每次设计和代码安全审查期间，AWS Security Agent 会自动验证所有应用程序的这些要求，从而确保团队满足您的特定要求，而不是通用清单。

扩展安全专业知识

AWS Security Agent 通过自动执行组织安全要求来扩展安全审查以匹配开发速度。通过 AWS 控制台集中配置需求，通过结果分析进行全面审查，并管理整个组织的渗透测试范围。

针对已确认漏洞的可操作修复

AWS Security Agent 通过基于证据的漏洞利用来验证渗透测试期间发现的安全发现，提供可重现的漏洞利用路径、全面的影响分析，并使用开发人员友好的语言创建带有随时可实施的修复程序的拉取请求。这可以帮助团队专注于合法的高影响力安全风险，而不会将时间浪费在误报上。

多云和混合云支持

AWS Security Agent 可在 AWS、本地、混合、多云和 SaaS 环境中运行，无论您选择何种基础设施或平台，都能确保一致的安全指导和测试。

AWS 安全代理功能

AWS 安全代理在您的整个开发生命周期中提供四项核心安全功能：

- 设计安全审查 — 审查设计和架构文档以识别安全风险。通过 Web 应用程序上传文档，该服务会根据您的组织安全要求对其进行分析。AWS Security Agent 会评估是否符合您定义的安全要求，例如经批准的授权库、日志标准和数据访问策略。这可以帮助您在开发过程的早期发现不安全的设计和违反策略的行为。
- 威胁建模-为您的应用程序构建威胁模型，以确定其可能受到攻击的方式。提供设计文档作为范围文档以定义分析重点，将源代码作为上下文供代理了解您的现有系统，或者两者兼而有之。AWS Security Agent 会生成系统概述，描述您的应用程序的架构、组件、信任边界、数据流和安全状况，以及按 STRIDE 类别（欺骗、篡改、拒绝、信息泄露、拒绝服务、权限提升）分类的一组威胁，以及严重性级别和可行的建议。每种威胁都链接到源代码中的证据。
- 代码安全审查 — 分析存储库和 S3 源代码中的代码，以识别跨语言、框架和架构的安全漏洞和违反组织安全要求的行为。在 Web 应用程序中创建代码审查以对您的完整源代码执行全面扫描，或者启用拉取请求注释以自动查看中的代码更改 GitHub。AWS Security Agent 提供具体的补救指导，并且可以针对已识别的漏洞自动生成包含代码修复的拉取请求。这可确保所有开发团队始终如一地执行您的安全策略。
- On-demand 渗透测试 — 通过量身定制的多步攻击场景发现、验证、报告和修复实时 Web 应用程序和 API 中的安全漏洞。通过指定测试范围、身份验证详细信息和资源，将服务配置为通过 Web 应用程序创建 pentest。AWS Security Agent 根据提供的源代码和文档开发应用程序环境，并执行复杂的攻击链，以识别静态分析和传统工具遗漏的可利用漏洞。它还提供随时可实施的代码修复，并直接在您的代码存储库中创建拉取请求，使您能够更快地解决漏洞。

AWS 安全代理的工作原理

AWS Security Agent 跨三个接口运行，可在整个开发生命周期中提供主动的应用程序安全。安全配置在 AWS 管理控制台中管理，安全审查在安全代理 Web 应用程序中进行，自动代码和安全发现补救直接在代码存储库平台中进行，例如 GitHub。

概述

AWS 安全代理由三个主要组件组成：

- AWS 管理控制台-配置代理空间、定义安全要求、配置渗透测试、连接代码存储库和管理用户访问权限
- Sec@@ urity Agent Web Application-在分配的代理空间内进行设计审查、创建和运行威胁模型、执行渗透测试、创建和运行代码审查以及查看安全发现
- 代码存储库集成-接收有关拉取请求和渗透测试补救拉取请求的自动代码审查。目前 AWS 安全代理支持连接 GitHub。

您使用以下三个角色之一与 AWS Security Agent 合作，您可以通过您使用的界面来识别这些角色：

角色	你在哪里工作，做什么
管理员	在 AWS 管理控制台中工作 — 设置代理空间、定义安全要求、配置功能和管理用户访问权限。
User	在 S ecurity Agent Web 应用程序中运行 — 运行设计审查、威胁模型、渗透测试和代码审查，并审查由此产生的结果。
开发者	在GitHub 或 IDE（例如 Kiro 或 Claude Code）中工作 — 接收有关拉取请求的自动安全调查结果，并在不离开开发环境的情况下运行代码扫描。

我们自始至终都使用这些角色名称来表示谁执行每项任务。一个人可以担任多个角色。

控制台配置

管理员通过 AWS 管理控制台配置 AWS 安全代理。

代理空间-当您在 AWS 管理控制台中创建第一个代理空间时，AWS 安全代理会为您的账户创建 Web 应用程序。您创建的每个 Agent Space 都代表您想要保护的不同应用程序或项目。在 Web 应用程序中，用户在进行安全评估时选择要在哪个代理空间中工作。

安全要求-定义 AWS Security Agent 在设计和代码审查期间评估的安全要求，这些要求按安全要求包整理。根据行业标准启用 AWS-managed 软件包，为组织的政策创建自定义包，或者通过上传安全文档来生成需求。要求适用于所有代理空间。

渗透测试配置-通过以下方式为每个代理空间配置渗透测试功能：

- 通过 DNS 或 HTTP 验证验证目标域以进行测试
- 配置 VPC 访问权限以测试私有应用程序
- 为渗透测试运行设置 CloudWatch 日志
- 为测试证书配置 AWS Secrets Manager 或 Lambda 函数
- 为其他应用程序上下文指定 S3 存储桶

代码审查配置-通过以下方式为每个代理空间配置代码审查功能：

- 连接 GitHub 存储库或包含源代码的 S3 存储桶
- 选择代码审查设置（安全漏洞、自定义要求或两者兼而有之）
- 启用拉取请求注释以自动查看中的代码更改 GitHub
- 为代码审查运行设置 CloudWatch 日志

威胁建模配置-通过以下方式为每个代理空间配置威胁建模功能：

- 连接源代码存储库或包含源代码的 S3 存储桶
- 添加范围文档（功能设计文档）以将分析重点放在特定功能上
- 为威胁模型运行配置 CloudWatch 日志记录
- 为 AWS 资源访问设置服务角色

集成-将 GitHub 存储库连接到每个代理空间以启用关键功能：

- 为更准确的渗透测试提供应用程序上下文
- 启用代码审查以进行完整存储库扫描和拉取请求分析
- 通过获取代码审查和渗透测试结果的拉取请求实现自动代码修复

用户访问权限-管理用户访问安全客户端 Web 应用程序的方式。如果您已启用 IAM 身份中心 (SSO)，请在 AWS 安全代理控制台中分配用户以提供对 Web 应用程序的直接 SSO 访问权限。如果您使用 IAM-only 访问权限，则拥有 AWS 控制台访问权限的用户可以通过控制台中的管理员访问链接启动任何 Agent Space 的 Web 应用程序。

Web 应用程序活动

用户访问安全客户端 Web 应用程序以在其分配的代理空间内进行安全评估。

选择代理空间-登录 Web 应用程序时，用户可以选择要在哪个代理空间中工作。用户只能查看和访问他们被分配到的代理空间。

设计审查-上传设计文档和架构规范，以便根据组织安全要求进行分析。使用补救指南查看调查结果。

威胁模型-通过提供源代码、技术设计文档（范围文档）或同时提供两者来创建和运行威胁模型。范围文档定义了代理重点分析的内容；源代码提供了有关您现有系统的上下文。每次运行都会生成系统概述，描述应用程序的架构、信任边界、数据流和安全状况，以及按STRIDE类别分类的一组威胁，包括严重级别和可行的建议。

代码审查-创建和运行代码审查，对您的完整源代码进行全面的静态分析。选择 GitHub 存储库或 S3 来源，配置扫描设置，并根据补救指南查看详细的安全发现。AWS Security Agent 可识别安全漏洞，并验证整个代码库是否符合贵组织的自定义安全要求。

渗透测试-通过提供目标网址、身份验证详细信息和文档来配置和执行渗透测试。AWS Security Agent 执行自主测试，通过多步攻击场景发现可利用的漏洞。

审查结果-检查设计审查、威胁模型、代码审查和渗透测试的详细安全发现，包括影响分析、可重现的攻击路径和补救指南。

验证修复-实施补救措施以验证漏洞已得到解决后的 Re-run 安全评估。

GitHub 整合

AWS Security Agent 直接 GitHub 与集成，在开发人员的工作流程中提供自动安全反馈。

拉取请求评论-管理员安装 AWS Security Agent GitHub 应用程序并通过代码审查评论为代理空间启用代码审查后，AWS 安全代理会自动分析连接存储库中的拉取请求。开发人员直接在拉取请求评论中收到安全发现和补救指导，根据组织安全要求和常见漏洞验证代码更改。

自动修复-当用户为代码审查启用自动代码修复时，AWS Security Agent 会针对已发现的漏洞生成修复程序，并将拉取请求提交到关联的 GitHub 存储库。

渗透测试补救-当管理员在控制台中启用查找补救功能时，用户可以从 Web 应用程序请求对渗透测试结果进行自动修复。AWS Security Agent 向关联 GitHub 存储库打开拉取请求，其中包含用于修复该漏洞的代码修复。

了解资源层次结构和生命周期

AWS Security Agent 以分层结构组织安全测试资源，该结构决定了组织中共享的内容以及每个应用程序的作用范围。了解此结构有助于您有效地配置 AWS 安全代理，并知道在哪里可以找到和管理不同的资源。

在整个组织中共享的内容

AWS Security Agent 中的某些资源只需在组织级别配置一次，即可应用于您的所有应用程序和代理空间。这些租户级资源可提供一致性并减少重复的配置工作。

资源	什么是	为什么要共享
安全要求	组织安全标准，用于定义 AWS 安全代理在设计和代码审查期间验证的内容	您的安全策略适用于所有应用程序。只需定义一次，AWS 安全代理即可在任何地方强制执行它们。
GitHub 集成	授权连接 AWS 安全代理的注册 GitHub 组织或用户账户	只需注册一次 GitHub 组织即可，然后根据需要将特定存储库连接到任何代理空间。
IAM 身份中心配置	控制用户如何访问 AWS 安全代理的 SSO 设置	集中式身份管理适用于组织中的所有座席空间。

> **Important**
>
> 安全要求的更改会影响所有 Agent Spaces 未来的所有设计审查和代码审查。现有评论不受影响。

每个特工空间的作用域有多大

每个 Agent Space 代表你想要保护的不同应用程序或项目。Agent Space 级别的资源仅限于该特定应用程序，允许不同的团队独立使用自己的配置和评估。

资源	什么是	为什么每个应用程序都有其作用域
渗透测试配置	测试应用程序中特定功能、API 端点或功能的配置	每个应用程序都有唯一的目标、身份验证方法和特定于该应用程序的范围界限。
设计评论	对设计文件进行单独的建筑安全评估	每个应用程序都有自己的架构和设计文档，这些文档需要经过独立评估。
威胁模型	威胁建模评估，用于构建系统概述并识别源代码、设计文档或两者兼而有之的威胁	每个应用程序都有自己的代码和设计，并且是独立建模的威胁。威胁模型是可重复使用的配置，您可以随着代码和设计的发展而重新运行这些配置。
集成	连接到该代理空间的来源和文档提供商（GitHub、 GitLab、Bitbucket、E GitHub nterprise Server 和 Confluence）	不同的应用程序依赖于不同的来源和文档。在 Agent Space 级别连接它们可以保持应用程序界限畅通。
代码审查设置	配置代码审查功能，包括连接的来源、扫描设置和公关评论启用	每个应用程序都有自己的存储库和独立配置的安全审查需求。
渗透测试修复设置	配置哪些连接的存储库可以接收渗透测试结果的自动修复拉取请求	团队可以根据其应用程序的工作流程控制 AWS 安全代理在何处提交代码更改。
用户分配	有权访问此特定代理空间的用户	团队只能看到他们负责的应用程序的安全评估，从而使工作井井有条，重点突出。

> **Tip**
>
> 我们建议为每个应用程序或项目创建一个代理空间，以保持团队之间的明确界限并有效地组织安全评估。

GitHub 存储库如何融入层次结构

GitHub 存储库通过将组织资源连接到特定应用程序的多步骤过程进行集成：

1. 在租户级别注册-为您的 GitHub 组织或用户账户授权 AWS 安全代理 GitHub 应用程序一次
2. 在代理空间级别连接-选择要连接到每个代理空间的特定存储库
3. 配置每个存储库的使用情况-为每个连接的存储库启用特定功能：
   - 代码审查-完整的源代码扫描和自动拉取请求分析
   - 渗透测试上下文-渗透测试期间从源代码中理解应用程序
   - 自动代码修复-自动拉取请求包含漏洞修复，用于代码审查和渗透测试结果

单个存储库可以连接到多个代理空间，每个代理空间都启用了不同的功能。

安全功能之间的主要区别

AWS Security Agent 中的每项安全功能都遵循不同的工作流程模型，具体取决于安全团队的使用方式。

渗透测试：可重复使用的配置，可独立执行

渗透测试使用支持迭代安全测试的配置和运行模型：

- 一次创建，多次执行-使用范围边界、身份验证和测试参数为特定目标（API 端点、功能区域）定义配置
- 独立执行-在提高安全性的同时，多次执行相同的配置。每次执行都是独立的，会产生新的发现

此模型支持在开发和部署改进措施时持续进行安全验证。

设计审查：使用克隆 One-off 进行评估

设计审查是独立评估，不遵循可重复使用的配置模型：

- 单一评估-每次设计评审都会根据贵组织的安全要求对上传的文档进行一次分析
- 无法重新运行-设计评审不可重复使用。您无法重新进行相同的审查
- 克隆以进行更新-克隆现有设计评审以创建预加载原始文档的新审阅，允许您更新文档并运行新的分析

此模型支持时间点架构安全评估。

代码审查：可重复使用的配置，包括按需扫描和自动 PR 分析

代码审查提供了两种操作模式来保护您的源代码：

- 完整代码审查（Web 应用程序）-创建用于选择 GitHub 存储库或 S3 源的代码审查配置，然后按需运行全面扫描。每次运行都会对您的完整源代码进行静态分析，并生成带有补救指南的结果。随着代码的发展，您可以重新运行相同的代码审查配置。
- 拉取请求评论 (GitHub)-启用对关联 GitHub 存储库的自动分析。当拉取请求被标记为可供审查时，AWS Security Agent 会自动对其进行审查，并将安全发现作为评论直接发布到中 GitHub。

这两种模式都使用您配置的代码审查设置（安全漏洞、自定义要求或两者兼而有之），并支持通过拉取请求进行自动代码修复。

威胁模型：可重复使用的配置，可按需运行

威胁模型使用支持对您的架构进行迭代评估的配置和运行模型：

- 一次创建，多次运行 — 通过选择源代码作为来源、将设计文档作为范围文档上传或两者兼而有之，来定义威胁模型。按需运行它，并随着代码和设计的演变而重新运行。
- 灵活的输入-仅对源代码运行威胁模型，仅对设计文档运行威胁模型，或两者兼而有之。范围文档定义了代理重点分析的内容；源代码提供了有关您现有系统的上下文。
- 系统概述和威胁 — 每次运行都会生成系统概述，描述应用程序的架构、信任边界、数据流和安全状况，以及按STRIDE类别分类的一组威胁，以及严重性、证据和可行的建议。

了解资源关系

层次结构决定了您在何处配置和访问不同资源：

在 AWS 管理控制台中：

- 配置租户级资源（安全要求、 GitHub 集成、IAM 身份中心）
- 创建和管理代理空间
- 配置 Agent Space 设置（连接的存储库、启用代码审查、渗透测试补救）

在安全客户端 Web 应用程序中：

- 创建和管理渗透测试配置和测试执行
- 创建和管理设计评审
- 针对连接的存储库和 S3 源创建、管理和运行代码审查
- 针对源代码、范围文档或两者创建、管理和运行威胁模型
- 查看渗透测试、代码审查、设计审查和威胁模型的发现

在 GitHub：

- 以拉取请求评论的形式查看拉取请求代码审查结果
- 接收自动修复拉取请求，以获取代码审查和渗透测试结果（在代理空间中启用后）

> **Note**
>
> 拉取请求代码审查结果显示在 GitHub。完整的代码审查、渗透测试和设计审查结果显示在安全客户端 Web 应用程序中。

开始使用

注册，找到实现目标的正确起点，然后使用快速入门进行首次评估。

注册获取 AWS 账户

要开始使用 AWS，您需要一个 AWS 帐户。有关创建 AWS 账户的信息，请参阅《AWS 账户管理参考指南》中的 AWS 账户入门。

选择您的路径

为您想要使用 AWS 安全代理做的事情找到正确的起点。使用下表将您的目标与任务相匹配，然后直接转到该页面。如果您不熟悉 AWS 安全代理，请the section called “什么是 AWS 安全代理？”先阅读以获得简要概述。

我想...	谁	从这里开始
在设置 AWS 安全代理之前先了解 AWS 安全代理的用途	Everyone	the section called “什么是 AWS 安全代理？”
设置服务并创建代理空间	Admin	the section called “设置 AWS 安全代理”
测试我的上线或已部署的应用程序是否存在可利用的漏洞	用户	the section called “快速入门：运行渗透测试”
扫描我的源代码中是否存在漏洞和违反政策的情况	用户	the section called “快速入门：运行代码审查”
模拟我的应用程序会如何受到攻击 (STRIDE)	用户	the section called “快速入门：运行威胁模型”
在我编写代码之前，请先获取有关设计的安全反馈	用户	the section called “创建设计评审”
无需离开我的 IDE 即可扫描代码（Kiro 或 Claude Code）	开发者版	the section called “从 IDE 运行代码安全扫描”

我想...	谁	从这里开始
合并前只扫描我更改过的行	开发者版	the section called “使用 S3 运行差分代码扫描”
获取有关拉取请求和合并请求的自动安全评论	Admin	the section called “启用代码审查”
查看调查结果并决定要先修复什么	用户	渗透测试、代码审查、威胁模型、设计审查

> **Note**
>
> AWS Security Agent 使用三个角色，您可以通过您使用的界面来识别这些角色：管理员在 AWS 管理控制台中工作（设置和配置），用户在 Web 应用程序中工作（运行评估和查看结果），开发人员在 IDE 中工作，例如 Kiro GitHub 或 Claude Code。一个人可以担任多个角色。有关完整定义，请参阅the section called “AWS 安全代理的工作原理”。

快速入门：运行渗透测试

本快速入门将引导您使用 AWS 安全代理进行首次渗透测试（pentest）。渗透测试会根据经过验证的目标域对已部署的应用程序进行练习，并返回安全调查结果，每个发现均附有严重性评级和支持证据。它涵盖了可公开访问的应用程序；要测试托管在私有 VPC 中的应用程序，请参阅the section called “启用渗透测试”。

> **Note**
>
> 您需要访问 AWS 管理控制台来设置 AWS 安全代理和定义测试范围，还需要访问 Web 应用程序才能创建和运行渗透测试。

先决条件

开始之前，请确保您已具备以下条件：

- 访问 AWS 管理控制台，并有权设置 AWS 安全代理。

- 托管您要测试的应用程序的实时目标域。
- 可以在同一 AWS 账户中为该域或一个 Route 53 托管区域添加 DNS 记录，这样您就可以验证所有权。
- （可选）应用程序的源代码存储库（GitHub GitLab、或 Bitbucket），用于为代理提供更多背景信息。

步骤 1：在 AWS 控制台中设置 AWS 安全代理

如果您尚未设置 AWS 安全代理，请完成初始设置：

1. 在 AWS 管理控制台中导航到 AWS 安全代理。
2. 选择设置 AWS 安全代理。
3. 创建代理空间。代理空间可供多个用户使用，并且应针对您要测试的每个应用程序而定。输入名称和描述。名称应标识要进行渗透测试的应用程序。
4. 在“用户IAM-only 访问配置”下选择访问权限。
   - 本快速入门不包括使用 IAM Identity Center 启用单点登录 (SSO)，它允许用户直接从 AWS 控制台访问 Web 应用程序。
   - 要让没有 AWS 管理控制台访问权限的用户开始渗透测试，请启用 IAM 身份中心集成。有关更多信息，请参阅 the section called “授予用户访问 AWS 安全代理 Web 应用程序的权限”。
5. 选择设置 AWS 安全代理。

> **Note**
>
> 当您选择“设置”时，AWS Security Agent 会创建您的代理空间并建立一个 Web 应用程序，用户可以在其中运行渗透测试、代码审查、威胁模型和设计审查。

第 2 步：启用渗透测试并验证您的域名

> **Note**
>
> 在 AWS 控制台中，您可以定义可以测试的范围。然后，用户通过 Web 应用程序在该范围内运行特定的渗透测试。

1. 在左侧边栏中，选择座席空间，然后选择您在步骤 1 中创建的座席空间。
2. 选择 “渗透测试” 选项卡，然后选择 “设置渗透测试” 以打开向导。
3. 配置域-输入要测试的目标域并选择验证方法，即 DNS TXT 记录或 HTTP 路由。该域必须处于活动状态，并且托管您要测试的应用程序。选择下一步。
4. 验证域-验证目标域表中每个域的所有权。AWS 安全代理仅针对经过验证的域运行测试。有关详细步骤和要复制的确切值，请参阅the section called “启用应用程序域进行渗透测试”。
   - 同一 AWS 账户中的 Route 53 个域-选择域名并选择One-click 验证。AWS 安全代理为您创建 DNS 记录并完成验证。
   - DNS TXT 记录（其他 DNS 提供商）-在 “目标域名” 表中，复制记录名称和记录值，将其作为 TXT 记录添加到您的 DNS 提供商，然后选择该域并选择验证。DNS 更改可能需要一段时间才能传播，因此验证可能无法立即完成。
   - HTTP 路由-在您的 Web 服务器`.well-known/aws/securityagent-domain-verification.json`上创建文件，添加格式为的验证令牌`{"tokens": ["<token>"]}`，然后选择域并选择 “验证”。
5. （可选）配置其他功能-添加可选资源，例如 CloudWatch 日志组和凭据。服务访问权限部分已预先配置：除非您选择现有 IAM 角色，否则 AWS 安全代理会创建具有所需权限的服务角色。

第 3 步：Connect 源代码（可选）

连接源代码提供商可为 AWS 安全代理提供有关您的应用程序的背景信息，并提高渗透测试覆盖范围。此为可选步骤。

1. 在 “渗透测试” 选项卡上，在页面顶部的 “Connect 用于渗透测试的源代码提供者” 横幅上选择 “添加”。
2. 注册并连接您的提供商，然后选择可供渗透测试使用的存储库。

有关完整的集成流程，请参阅 Connect AWS Security Agent 连接到 GitHub 存储库或您的提供商的连接主题。

第 4 步：创建并运行渗透测试

> **Note**
>
> 您可以在 AWS 安全代理 Web 应用程序中创建和运行渗透测试。我们建议在与生产环境非常接近的测试环境下运行测试。

1. 启动 Web 应用程序：选择 Web 应用程序选项卡，然后选择管理员访问权限。
2. 在左侧边栏中，选择渗透测试，然后选择创建渗透测试。
3. 渗透测试详情-设置测试范围和日志来源：
   a. 输入 Pentest 的名字。
   b. 在 “目标 URL” 下，选择或输入要测试的经过验证的域名（例如`https://example.com`）。只有经过验证的域名才能进行测试，子域名会被自动覆盖。
   c. （可选）在 “可访问的 URL” 下，添加代理必须访问才能登录和导航但不应攻击的域名，例如目标网域之外的身份提供商、CDN 或 API。AWS Security Agent 可以访问这些域，但不能对其进行测试；只有您的目标域会受到攻击。
   d. 查看网络配置规则，以确认哪些端点可以在测试期间接收流量，哪些不能接收流量。
   e. 在 “权限” 下，选择服务角色和CloudWatch 日志组（可选）。选择下一步。
4. （可选）VPC 资源-如果您的目标位于无法公开访问的私有网络上，请配置 VPC。请参阅the section called “启用渗透测试”。
5. （可选）身份验证资源-提供凭据，以便代理可以访问经过身份验证的非公开路径。我们建议添加这些漏洞，以扩大覆盖范围，揭露登录背后的漏洞。
6. （可选）其他配置-添加文件、 GitHub 存储库或 S3 源等应用程序上下文以提高查找质量。您还可以在此处启用自动代码修复并设置其他运行选项。
7. 选择 “创建并执行” 以立即开始测试，或者选择 “创建 pentest” 将其保存并稍后运行。

第 5 步：查看渗透测试结果

1. 渗透测试可能需要几个小时才能完成。大多数在 16 小时内完成，具体取决于您的应用程序的大小和复杂性。
2. 运行从 Prefli ght 阶段开始，该阶段在测试开始之前验证设置：它设置日志基础架构，检查目标端点是否可访问，并准备测试环境。如果印前检查失败（例如，无法解析的目标域），则运行将在测试开始之前停止。打开 “印前检查” 选项卡，查看哪个检查失败了，解决这个问题，然后开始新的运行。
3. 运行完成后，将其打开并查看 “运行概述”、“日志” 和 “查找结果” 选项卡。
4. 在 “调查结果” 选项卡上，选择一项发现以查看其描述、严重程度、风险类型和支持证据。

有关更多详细信息，请参阅 the section called “创建渗透测试” 和 the section called “查看渗透测试的结果”。

快速入门：运行代码审查

本快速入门将引导您使用 AWS 安全代理进行首次代码审查。AWS Security Agent 会扫描您的源代码存储库是否存在安全漏洞以及是否符合您组织的安全要求。

> **Note**
>
> 您需要访问 AWS 管理控制台才能设置 AWS 安全代理，并需要访问 Web 应用程序才能创建和运行代码审查。

先决条件

开始之前，请确保您已具备以下条件：

- 访问 AWS 管理控制台，并有权设置 AWS 安全代理。
- 包含您要查看的代码的源代码存储库（GitHub GitLab、或 Bitbucket）或 Amazon S3 源代码。

步骤 1：在 AWS 控制台中设置 AWS 安全代理

如果您尚未设置 AWS 安全代理，请完成初始设置：

1. 在 AWS 管理控制台中导航到 AWS 安全代理。
2. 选择设置 AWS 安全代理。
3. 创建代理空间。代理空间可供多个用户使用，并且应针对您要测试的每个应用程序而定。输入您的第一个代理空间的名称和描述。此名称会在 Web 应用程序中显示给用户。名称应标识要查看其代码的应用程序。
4. 在“用户IAM-only 访问配置”下选择访问权限。
   - 本快速入门不包括通过 IAM 身份中心启用单点登录 (SSO)。这允许用户从 AWS 控制台直接访问 AWS 安全代理 Web 应用程序。
   - 要让没有 AWS 管理控制台访问权限的用户运行代码审查，请启用 IAM 身份中心集成。有关更多信息，请参阅 the section called “授予用户访问 AWS 安全代理 Web 应用程序的权限”。
5. 选择设置 AWS 安全代理。

> **Note**
>
> 当您选择 “设置” 时，AWS Security Agent 会创建您的代理空间并建立一个 Web 应用程序，用户可以在其中运行渗透测试、代码审查、威胁模型和设计审查。

第 2 步：启用和配置代码审查

> **Note**
>
> 如果您已经将 GitHub 存储库或 S3 存储桶连接到 Agent Space（例如，通过渗透测试设置），则代码审查已启用。您可以跳过此步骤，直接转到 Web 应用程序。

打开代码审查设置向导

1. 在左侧边栏中，选择 “座席空间”，然后选择您的座席空间。
2. 从标题或代码审查选项卡中选择 “启用代码审查”。

Connect 集成、存储库和存储桶

1. （如果您还没有集 GitHub 成）创建 GitHub 注册。如果您已经有一个，请跳到下一步。
   a. 在 “已连接的集成” 部分中，选择添加，然后选择创建新注册。
   b. 选择GitHub并选择 “下一步”。
   c. 选择 “安装并授权”，然后在 GitHub以下位置完成安装：
      i. 选择拥有您要查看的存储库的 GitHub 用户或组织。
      ii. 选择 “所有存储库” 或 “仅选择存储库”。
      iii. 选择 “安装并授权”，然后完成 GitHub 身份验证。
   d. 返回 AWS 管理控制台，输入注册名称并确认账户类型与您安装 GitHub 应用程序的位置相匹配。
   e. 选择 Connect 保存注册。

   有关完整的 GitHub 集成流程，请参阅将 AWS 安全代理连接到 GitHub 存储库。
2. Connect GitHub 存储库。在 “已连接的集成” 部分中，选择添加，然后选择您的 GitHub 注册。两步式 Connect GitHub 向导打开：
   a. 在 Connect GitHub 存储库上，选择要包含的存储库，然后选择下一步。

b. 在 “管理权能” 上，切换每个存储库的以下选项：

- 代码审查评论 — 让 AWS 安全代理将安全发现作为对存储库中拉取请求的评论发布。
- 自动修复 — 允许 AWS 安全代理 Web 应用程序的用户请求修复发现的拉取请求。

c. 选择 “保存” 返回到安装向导。

3. （可选）连接 S3 源。在 S3 存储桶部分，选择添加 S3 资源并输入包含源代码的存储桶的 S3 URI，或者选择浏览以选择一个。
4. 选择您的代码审查设置。默认的 “安全要求和漏洞发现” 会分析代码是否符合您启用的安全要求和常见漏洞。
5. 选择下一步。

可选配置

1. 配置可选的 CloudWatch 日志组和服务访问权限。默认服务角色已预先配置了所需的权限。
2. 选择保存。

第 3 步：创建并运行代码审查

> **Note**
>
> 您只能在 AWS 安全代理 Web 应用程序中创建和运行代码审查。

1. 选择 Web 应用程序选项卡，然后选择管理员访问权限以启动 AWS 安全代理 Web 应用程序。
2. 在左侧边栏中，选择代码审查。
3. 选择 “创建代码审查”。
4. 配置代码审查：

a. 输入标识此审查范围的标题（例如，“计费服务安全审查”）。

b. 在 “来源” 下，选择您在步骤 2 中连接到代理空间的 GitHub 存储库，或输入要扫描的 S3 源。

c. 从您配置的角色中选择服务角色。

d. （可选）选择 “启用自动代码修复”，让 AWS Security Agent 自动提交包含所有发现的修复的拉取请求。

5. 选择 “创建代码审查”。

6. 在代码审查详细信息页面上，选择开始审查。

步骤 4：查看代码审查结果

1. 代码审查通常需要 30-60 分钟，具体取决于您的代码库的大小。
2. 运行从预检阶段开始，在扫描开始之前验证设置：它确认 AWS Security Agent 可以从您的存储库或 S3 源中提取您的源代码并设置扫描环境。如果印前检查失败（例如，它无法访问您的源），则运行将在静态分析之前停止。打开 “印前检查” 选项卡，查看哪个检查失败了，解决这个问题，然后开始新的运行。
3. 完成后，导航到已完成的运行并选择 Finding s 选项卡。
4. 查看列表详细信息视图中的调查结果：
   a. 从左侧面板中选择一个查找结果以查看其详细信息。
   b. 查看 “描述”、“代码位置”、“证据” 和 “建议的修复” 部分。
   c. 使用 R emediate 代码生成包含修复的拉取请求，或者查看自动修复 PR（如果您启用了该选项）。

有关更多详细信息，请参阅 the section called “创建代码审查” 和 the section called “查看代码审查的结果”。

快速入门：运行威胁模型

本快速入门将引导您使用 AWS 安全代理运行您的第一个威胁模型。威胁模型分析您的应用程序架构，并生成系统概述（AWS Security Agent 如何理解您的系统）和一组威胁（如何攻击威胁，每个威胁都有严重性级别、STRIDE 分类和建议）。您可以在设计文档（范围文档）上运行威胁模型以定义焦点，在源代码（来源）上运行威胁模型以提供有关现有系统的上下文，或者两者兼而有之。

> **Note**
>
> 您需要访问 AWS 管理控制台才能设置 AWS 安全代理，并需要访问 Web 应用程序才能创建和运行威胁模型。

步骤 1：在 AWS 控制台中设置 AWS 安全代理

如果您尚未设置 AWS 安全代理，请完成初始设置：

1. 在 AWS 管理控制台中导航到 AWS 安全代理。
2. 选择设置 AWS 安全代理。
3. 创建代理空间。代理空间可供多个用户使用，并且应针对您想要保护的每个应用程序而定。输入您的第一个代理空间的名称和描述。名称应标识您想要威胁模型的应用程序。
4. 在 “用户IAM-only 访问配置” 下选择访问权限。
   - 本快速入门不包括通过 IAM 身份中心启用单点登录 (SSO)。这允许用户从 AWS 控制台直接访问 AWS 安全代理 Web 应用程序。
   - 如果您想让没有 AWS 管理控制台访问权限的用户能够执行启动威胁模型等任务，则应启用 IAM Identity Center 集成。
5. 选择设置 AWS 安全代理。

> **Note**
>
> 当您选择 “设置” 时，AWS Security Agent 会创建您的代理空间并建立一个 Web 应用程序，用户可以在其中运行渗透测试、代码审查、威胁模型和设计审查。

第 2 步：Connect 源代码

> **Note**
>
> 如果您已经将存储库或 S3 存储桶连接到 Agent Space（例如，通过代码审查或渗透测试设置），则可以跳过此步骤直接转到 Web 应用程序。您始终可以在上传的范围文档上运行威胁模型，而无需连接任何源代码。

连接包含您希望代理用作了解现有系统的上下文的源代码的存储库或 S3 存储桶：

1. 在左侧边栏中，选择 “座席空间”，然后选择您的座席空间。
2. 导航到 “集成” 部分以连接您的源代码提供商。
3. 或者，连接包含您的源代码的 S3 存储桶。

有关完整的集成流程，请参阅将 AWS 安全代理连接到 GitHub 存储库。

步骤 3：创建并运行威胁模型

> **Note**
>
> 您可以在 AWS 安全代理 Web 应用程序中创建和运行威胁模型。

1. 从 AWS 管理控制台的 “Web 应用程序” 选项卡启动 Web 应用程序。或者，如果您配置了 IAM 身份中心，请直接登录。
2. 在左侧边栏中，选择威胁模型。
3. 选择创建威胁模型。
4. 配置威胁模型：
   a. 输入标识此威胁模型范围的标题（例如，“计费服务” 或 “checkout-api”）。
   b. 至少提供一个输入：
      - 在 S cope 文档下，上传设计文档（DOC、DOCX、JPEG、MD、PDF、PNG 或 TXT），输入 S3 URI 或选择 Confluence 页面。
      - 在 “来源” 下，从关联的集成中选择存储库或输入包含源代码的 S3 URI。

      > **Tip**
      >
      > 提供源代码和范围文档，将威胁模型的范围限定为特定的设计（例如功能设计文档），同时向代理提供源代码作为上下文，以了解您的现有系统。

   c. 从您配置的角色中选择服务角色。
   d. （可选）为日志选择一个 CloudWatch 日志组。
5. 选择创建威胁模型。
6. 在威胁模型详细信息页面上，选择开始运行。

步骤 4：查看威胁

1. 运行将通过其分析任务进行。您可以在 “印前检查” 选项卡上监控进度，在 “日志” 选项卡上查看任务详细信息。
2. 完成后，运行状态将更改为 “已完成”。
3. 选择概述选项卡以查看系统概述和严重性分布。

4. 选择 “威胁” 选项卡以查看已识别的威胁：

   a. 从列表中选择一个威胁，在侧面板中查看其详细信息。

   b. 查看 “陈述”、“严重程度”、“STRIDE” 类别、“建议”、“证据” 和其他字段。

   c. 处理或分类每个威胁时，将威胁状态更新为 “已解决” 或 “已解除”。

有关更多详细信息，请参阅 the section called “创建威胁模型” 和 the section called “查看来自威胁模型的威胁”。

适用于管理员（控制台）

作为管理员，您可以在 AWS 管理控制台中设置 AWS 安全代理，并配置代理空间，供用户通过 AWS 安全代理 Web 应用程序进行访问。每个代理空间代表一个具有特定权限和资源的不同环境。

我们建议您为要测试的每个应用程序创建一个唯一的代理空间。例如，如果您有两个内部项目（计费应用程序和任务跟踪应用程序），则应创建两个单独的座席空间。

示例配置

考虑由管理员设置代理空间来评估内部计费应用程序的安全性。管理员将：

- 验证域名（例如`beta.billing.example.com`）
- 连接 Connect GitHub 并启用代码审查
- 通过为渗透测试分配适当的 VPC、子网和安全组来配置网络访问权限

当用户启动渗透测试或设计审查时，他们可以从这些预先配置的资源中进行选择，在您定义的防护栏内工作，同时保持灵活性以满足其特定评估需求。

设置和创建代理空间

为您的组织设置 AWS 安全代理，并创建代理空间，以确定每个应用程序的资源和评估范围。

设置 AWS 安全代理

通过创建您的第一个代理空间并确定用户访问 Web 应用程序的方式，为您的组织配置 AWS 安全代理。

这种初始设置使管理员能够在 AWS 控制台中配置安全功能，并允许用户通过安全代理 Web 应用程序访问设计审查和渗透测试。完成一次性设置后，您可以通过简化的流程创建其他代理空间。

概述

了解代理空间

代理空间是用于保护特定应用程序或项目的专用工作区。它包含该应用程序的所有安全审查、可选连接的 GitHub 存储库、渗透测试配置、结果和发现。

Agent Spaces 通过将每个应用程序的安全评估分开来帮助您组织安全工作，从而使团队能够专注于其特定的应用程序。我们建议为每个应用程序或项目创建一个代理空间，以保持明确的界限。

安全要求是在组织级别定义的，适用于所有代理空间，而每个代理空间则维护自己的设计文档、代码存储库、渗透测试配置、渗透测试结果和安全发现。

特工空间中包含的内容

每个代理空间包含：

- （可选）与应用程序或项目关联的已连接 GitHub 存储库
- 之前对该应用程序的设计审查
- 应用程序特有的渗透测试配置和边界
- 渗透测试测试结果和安全发现

你将在安装过程中配置什么

在首次设置期间，您将做出适用于所有座席空间的组织决策：

- 访问方法-选择用户访问安全代理 Web 应用程序的方式（IAM 身份中心 SSO 或通过 AWS 控制台进行 IAM-only 访问）
- 权限-配置 Web 应用程序用于访问 AWS 服务的 IAM 角色
- First Agent Sp ace-使用名称和描述创建你的初始特工空间

> **Note**
>
> 完成此设置后，创建其他代理空间就更简单了，只需提供名称和描述即可。有关创建后续代理空间的详细信息，请参阅the section called “创建代理空间”。

先决条件

在开始之前，请确保您满足以下条件：

- 创建和管理 AWS 安全代理资源的权限
- 了解贵组织的身份管理要求
- （可选）有权创建 IAM 角色和配置 IAM 身份中心的 AWS 账户（如果使用 SSO 访问权限）
- （可选）如果您想使用自定义权限配置，请使用现有 IAM 角色

第 1 步：创建您的第一个特工空间

定义将在 Web 应用程序中向用户显示的第一个代理空间的基本属性。

1. 在 AWS 安全代理控制台页面上，选择设置安全代理。
2. 在座席空间名称字段中，输入您的座席空间的名称。

> **Note**
>
> Agent Space 名称在 Web 应用程序中向用户显示，它有助于确定该空间代表哪个应用程序或项目。

3. （可选）在描述字段中，提供有助于区分代理空间用途的描述。

> **Tip**
>
> 该描述有助于区分代理空间的用途。我们建议描述此 Agent Space 将保护的特定应用程序或项目，例如 “客户门户 Web 应用程序”、“支付处理微服务” 或 “内部分析平台”。

第 2 步：选择您的访问方式

选择用户访问安全客户端 Web 应用程序的方式。您可以选择通过 IAM 身份中心启用 SSO 访问或通过 AWS 控制台提供访问权限。

> **Note**
>
> 如果您选择 IAM-only 访问权限然后想使用 IAM Identity Center，则需要删除您的 AWS 安全代理设置并重新完成设置过程。

1. 在 “访问方法” 部分，选择以下选项之一：
   - IAM 身份中心 (SSO)-通过 IAM 身份中心为您的团队启用 SSO 访问权限。对于需要集中用户管理并希望用户无需通过 AWS 控制台即可直接访问 Web 应用程序的团队，建议使用此选项。
   - IAM-only 访问-通过 AWS 控制台中的管理员访问链接提供访问权限。此选项更易于设置，适合喜欢基于控制台的访问权限或不需要 SSO 功能的团队。
2. 如果您选择了 IAM 身份中心 (SSO)，请继续执行下面的步骤 2a。
3. 如果您选择了IAM-only 访问权限，请跳至步骤 3。

步骤 2a：配置 IAM 身份中心（仅限 SSO 访问权限）

仅当您选择 IAM 身份中心 (SSO) 作为访问方法时，才完成这些步骤。

1. 在 Connect to IAM 身份中心部分，查看 AWS 区域。

> **Important**
>
> 您的应用程序必须在启用 IAM 身份中心的同一区域进行配置。显示的区域是将创建您的代理空间的地方。必须在您创建代理空间的同一区域启用 IAM 身份中心。

2. 在 IAM 身份中心部分，选择以下选项之一：
   - 选择创建账户实例以创建新的 IAM Identity Center 账户实例
   - 如果同一区域中已存在组织实例，AWS 安全代理将自动连接到该实例

> **Note**
>
> AWS 安全代理连接到 IAM 身份中心后，您可以通过将 IAM 身份中心中的用户分配给应用程序来管理访问权限。

3. 如果您正在连接 Active Directory 或外部身份提供商，请查看信息警报。

> **Important**
>
> 如果您已经在管理 Active Directory 或其他身份源中的用户，并计划将该身份源连接到 IAM Identity Center，请取消设置并先前往 IAM Identity Center 控制台。在设置 AWS 安全代理之前，选择要连接的身份源。稍后更改身份源可能会移除现有的用户分配。

步骤 3：配置权限（可选）

配置您的安全代理 Web 应用程序用于访问 AWS 服务、API 和账户的 IAM 角色。

1. 找到 “权限配置-可选” 部分。
2. 如果该部分已折叠，请选择权限配置部分将其展开。
3. 选择下列选项之一：
   - 创建默认角色-AWS 安全代理会自动创建一个具有 Web 应用程序必要权限的新 IAM 角色
   - 使用其他角色-从可用选项中选择现有 IAM 角色

4. 查看角色描述：

> **Note**
>
> 将为您的 Web 应用程序创建一个默认 IAM 角色，用于访问其他 AWS 服务、API 和账户。该角色将被授予安全评估操作所需的权限。

步骤 4：完成设置

配置完所有必需的设置后，完成设置以创建您的第一个代理空间并建立安全客户端 Web 应用程序。

1. 查看所有配置部分以确保准确性。
2. 选择设置。
3. AWS 安全代理会创建您的代理空间、建立 Web 应用程序并配置必要的 AWS 资源。

> **Note**
>
> 初始设置后，您可以选择配置功能，包括渗透测试边界、安全要求和 GitHub 集成。

后续步骤

设置 AWS 安全代理后：

- 为您的组织定义安全要求
- 为您的代理空间配置渗透测试功能，包括域验证和 VPC 访问权限
- Connect GitHub 存储库以查看代码审查和渗透测试环境
- （如果使用 IAM 身份中心）在 AWS 安全代理控制台中将用户分配到代理空间
- （如果使用 IAM-only 访问权限）通过 AWS 控制台中的管理员访问链接启动 Web 应用程序
- 为其他应用程序创建额外的代理空间（请参阅the section called “创建代理空间”）

创建代理空间

创建代理空间以保护组织中的应用程序或项目。每个 Agent Space 都为特定于该应用程序或项目的安全评估、发现和配置提供了一个工作区，可供多个用户访问。

此过程假设您已经完成了 AWS 安全代理的初始设置。如果您尚未设置 AWS 安全代理，请参阅the section called “设置 AWS 安全代理”。

概述

了解代理空间

代理空间是用于保护特定应用程序或项目的专用工作区。它包含该应用程序的所有安全审查、可选连接的代码存储库、渗透测试配置、结果和发现。

Agent Spaces 通过将每个应用程序的安全评估分开来帮助您组织安全工作，从而使团队能够专注于其特定的应用程序。我们建议为每个应用程序或项目创建一个代理空间，以保持明确的界限。

有关代理空间及其如何融入组织安全结构的全面说明，请参阅the section called “了解资源层次结构和生命周期”。

特工空间中包含的内容

每个代理空间包含：

- （可选）与应用程序或项目关联的连接代码存储库
- 之前对应用程序或项目的设计审查
- 应用程序特有的渗透测试配置和边界
- 渗透测试测试结果和安全发现

创建代理空间

为要保护的应用程序或项目创建新的代理空间。

1. 在 AWS 安全代理控制台中，导航到代理空间页面。
2. 选择 “创建代理空间”。
3. 在座席空间名称字段中，输入您的座席空间的名称。

> **Note**
>
> Agent Space 名称在 Web 应用程序中向用户显示，它有助于确定该空间代表哪个应用程序或项目。

4. （可选）在描述字段中，提供有助于区分代理空间用途的描述。

> **Tip**
>
> 该描述有助于区分代理空间的用途。我们建议描述此 Agent Space 将保护的特定应用程序或项目，例如 “客户门户 Web 应用程序”、“支付处理微服务” 或 “内部分析平台”。

5. 选择创建。

> **Note**
>
> AWS 安全代理将创建您的代理空间。现在，您可以配置功能并连接此应用程序特有的资源。

后续步骤

创建代理空间后：

- 连接源代码存储库（GitHub、 GitLab、Bitbucket 或 En GitHub terprise Server）以进行代码审查和渗透测试环境
- Connect Confluence 获取设计审查和渗透测试中的文档背景
- 为连接的存储库启用代码审查功能（请参阅the section called “为 GitHub 仓库启用拉取请求代码审查”）
- 配置渗透测试功能，包括域名验证
- （如果使用 IAM Identity Center）在 “代理空间” 页面的 “Web 应用程序” 部分下将用户分配到此代理空间。（见the section called “授予用户访问 AWS 安全代理 Web 应用程序的权限”）
- （如果使用 IAM-only 访问权限）具有控制台访问权限的用户可以通过此代理空间的管理员访问链接启动 Web 应用程序

Connect 集成

连接并管理源代码提供程序（GitHub、 GitLab、Bitbucket 和 En GitHub terprise Server）和文档提供程序（Confluence），以及您的代理空间用于代码审查、渗透测试和威胁建模的专用网络连接。

如何与代理空间进行集成

AWS Security Agent 连接到第三方源代码和文档提供商，因此代理可以在安全评估期间分析您的代码和文档。在连接特定提供商之前，了解集成与代理空间和功能的关系会有所帮助。

注册一次，随时随地重复使用

将提供商连接到 AWS 安全代理涉及两个不同的步骤，分别发生在不同的级别：

1. 注册集成（账户级别）。您只需从 AWS 安全代理管理控制台的集成页面注册一次提供商。注册表示与提供商账户（例如 GitHub 组织、 GitLab 群组、Bitbucket 工作空间或 Confluence 站点）的授权连接。注册是账户级别的资源。
2. 将资源连接到代理空间（代理空间级别）。在代理空间中，您可以连接来自已注册集成的资源（源代码提供者的存储库或 Confluence 的页面），并配置代理如何使用这些资源。此步骤特定于每个代理空间。

您账户中的每个代理空间都可重复使用一次注册。如果您在设置一个代理空间时注册了提供商，则当您将资源连接到另一个代理空间时，可以进行相同的注册，而无需再次注册同一个提供商。

一个连接，跨功能共享

当您将资源连接到代理空间时，该资源将在代理的权能之间共享。您不必为每项功能单独连接存储库。

- 通过连接资源来授予读取权限。连接存储库可让 AWS Security Agent 读取该存储库以进行完整的代码扫描（代码审查）、渗透测试上下文、威胁建模和设计审查。连接 Confluence 页面可以让代理在设计审查、威胁建模和渗透测试中阅读文档背景信息。
- 每个存储库的写入操作都是可选的。将源代码存储库连接到 Agent Space 时，还可以为每个存储库启用写入操作：
  - 代码审查评论-AWS Security Agent 将审查结果作为对拉取请求（或合并请求 GitLab）的评论发布。
  - 代码修复-AWS 安全代理打开拉取请求（或合并请求），并修复已发现的漏洞。

    这些写入操作不适用于公共存储库。 Read-only 分析仍然适用。

> **Note**
>
> Confluence 是文档提供商，而不是源代码提供商。AWS 安全代理读取连接的 Confluence 页面，为设计审查、威胁建模和渗透测试提供背景信息。选择页面授予读取权限；没有按页面切换功能。

支持的提供商

AWS 安全代理支持以下提供商：

- 源代码- GitHub （云托管 GitHub 和云托管的 GitHub 企业）、 GitHub 企业服务器（自托管）、（云托管）、 GitLab （自托管）和 Bitbuck GitLab Self-Managed et Cloud。
- 文档-Confluence Cloud。

对于无法通过公共互联网访问的自托管提供商，您可以通过私有连接路由代理的流量。有关更多信息，请参阅 the section called “Connect 连接到私有托管的源代码管理”。

后续步骤

- 从 “集成” 页面注册提供商。请参阅您的提供商的连接主题，例如the section called “将 AWS 安全代理连接到 GitHub 存储库”。
- 将资源连接到代理空间并配置功能。请参阅the section called “启用代码审查”和the section called “启用渗透测试”。

将 AWS 安全代理连接到 GitHub 存储库

将您的 AWS 安全代理连接到 GitHub 存储库，以启用代码审查、威胁建模、渗透测试和自动修复功能。AWS 安全代理支持云托管 GitHub 和云托管的企业版 GitHub 。在开始之前，the section called “如何与代理空间进行集成”请查看以了解如何在座席空间中重复使用注册以及在各个功能之间共享注册。

GitHub 集成有多种用途：

> **Note**
>
> 本页涵盖云托管 GitHub (github.com) 和云托管的企业版。 GitHub 有关自托管 GitHub 企业服务器的信息，请参阅the section called “将 AWS 安全代理连接到 GitHub 企业服务器”。

- 代码审查-根据组织安全要求自动分析每个拉取请求中的代码更改，并按需运行完整存储库扫描
- 威胁建模-通过分析源代码、数据流和架构，提供对应用程序的理解
- 渗透测试上下文-通过分析源代码，为渗透测试提供对应用程序的理解
- 自动修复-提交拉取请求，修复安全评估期间发现的漏洞

连接 GitHub 到 AWS 安全代理需要为您的 GitHub 组织或用户账户授权 AWS 安全代理 GitHub 应用程序，然后在 AWS 控制台中注册连接。

GitHub 整合的工作原理

拉取请求分析发生在内部 GitHub。在您授权 GitHub 应用程序、连接存储库并在 AWS 管理控制台中启用代码审查注释后，AWS Security Agent 会自动扫描每个新拉取请求中的更改（仅对更改后的代码进行差异扫描），并将发现结果作为拉取请求注释发布。

您可以在 AWS Security Agent Web 应用程序中创建和运行完整的代码审查（扫描存储库的整个代码库），而不是在 AWS 安全代理 Web 应用程序中 GitHub。

渗透测试和威胁建模在 AWS 安全代理 Web 应用程序中启动。用户选择连接的存储库来提供应用程序上下文，并指定渗透测试的目标域。如果您启用自动修复，则用户可以通过向连接的存储库打开拉取请求来请求 AWS Security Agent 修复发现的问题。

先决条件

在开始之前，请确保您满足以下条件：

- GitHub 组织管理员访问权限或 GitHub 用户帐户所有者访问权限
- 在 AWS 管理控制台中为您的代理空间配置集成的权限
- 了解要连接哪些存储库以进行代码审查、威胁建模和渗透测试

Important

一个 GitHub 应用程序只能向一个 GitHub 账户或 GitHub 组织安装一次。如果您需要将同一个 GitHub 组织连接到 AWS 安全代理，则必须使用首次注册集成时所在的 AWS 账户。

Note

如果您的 GitHub 企业组织已启用 IP 许可名单，则必须接受 GitHub 应用程序上允许的 IP 地址。您也可以选择将这些 IP 地址自动添加到您的允许列表中。有关更多信息，请参阅 GitHub 文档中的允许 GitHub 应用程序访问和启用允许的 IP 地址。
以下 IP 地址用于访问您的 GitHub 资源：

- 美国东部（弗吉尼亚州北部）(us-east-1)
  - 34.228.181.128
  - 44.219.176.187
  - 54.226.244.221
- 美国西部（俄勒冈州）(us-west-2)
  - 34.212.16.133
  - 52.89.67.212
  - 54.187.135.61
- 亚太地区（孟买）(ap-south-1)
  - 13.126.209.199
  - 13.234.6.24
  - 35.154.102.216
- 亚太地区（新加坡）(ap-southeast-1)
  - 18.139.13.125
  - 47.130.240.215
  - 54.179.238.173
- 亚太地区（悉尼）(ap-southeast-2)
  - 13.237.95.197
  - 13.238.84.102
  - 52.64.174.242

- 亚太地区（东京）(ap-northeast-1)
  - 13.192.12.233
  - 35.74.181.230
  - 57.183.50.158
- 欧洲地区（法兰克福）(eu-central-1)
  - 18.158.110.140
  - 52.57.96.160
  - 52.59.55.56
- 欧洲地区（爱尔兰）(eu-west-1)
  - 34.251.85.24
  - 52.30.157.157
  - 52.51.192.222
- 南美洲（圣保罗）（sa-east-1）
  - 54.94.247.213
  - 54.207.222.14
  - 54.232.201.242

授权并注册 AWS 安全代理 GitHub 应用程序

授权 AWS 安全代理 GitHub 应用程序访问您的 GitHub 组织或用户账户，然后在 AWS 控制台中注册连接。

> **Important**
>
> 无需关闭浏览器或导航离开，即可完成此过程中的所有步骤。如果注册过程中断，则可能需要卸载 GitHub 应用程序并重新开始。

1. 在 AWS 安全代理管理控制台中，导航到 “集成”。
2. 选择添加集成。
3. 选择 GitHub。
4. 选择下一步。

5. 选择 “安装并授权”。

   您将被重定向 GitHub 到以完成授权。确保您 GitHub 使用对要连接的组织或用户帐户具有管理员访问权限的帐户登录。

6. 在中 GitHub，选择要安装 AWS 安全代理 GitHub 应用程序的账户或组织。
7. 选择 AWS 安全代理可以访问的存储库：
   - 所有存储库-授予对组织或用户帐户中所有当前和将来的存储库的访问权限
   - 仅选择存储库-从下拉列表中选择特定的存储库。您可以一次选择多个存储库。

   > **Note**
   >
   > 您可以随时通过访问 GitHub 组织中的 GitHub 应用程序设置或用户帐户设置来修改仓库访问权限。

8. 选择 “安装并授权”。
9. 您将被重定向回 AWS 管理控制台以完成注册。
10. 在注册详细信息部分，配置以下字段：
    a. 注册名称-输入此 GitHub 连接的描述性名称。使用标识 GitHub 组织或用户帐户的名称，例如 “Acme-Corp-Org” 或 “Production-Repos”。
    b. 账户类型-从下拉列表中选择以下选项之一：
       - 组织-如果您关联了 GitHub 组织帐户
       - 用户-如果您关联了个人 GitHub 用户帐户
    c. 组织名称（仅在您选择了组织时才会出现）-输入显示在中的 GitHub 组织确切名称 GitHub。
11. 选择连接。
12. 您会看到一条确认消息，然后返回到集成页面，您的新 GitHub 连接及其注册名称将出现在该页面中。要连接其他 GitHub 组织或用户帐户，请再次选择 “添加集成”，重复此过程。

排除 GitHub 集成故障

如果您在 GitHub 集成过程中遇到问题，请使用以下指导来解决常见问题。

无法完成注册

如果您无法完成注册流程（例如，您的浏览器已关闭、离开注册页面或遇到会话中断），AWS Security Agent GitHub 应用程序可能已安装在您的 GitHub 组织中，但未在 AWS 控制台中注册。

症状：

- 当您再次尝试授权 GitHub 应用程序时，会 GitHub 显示 “配置” 而不是 “安装”
- 您无法在 AWS 控制台中完成注册
- 该集成未出现在您的集成列表中

解决方法：

1. 从您的 GitHub 组织或用户账户中卸载 AWS 安全代理 GitHub 应用程序。
2. 返回 AWS 安全代理控制台，从头开始重新开始集成过程。

多个 AWS 账户正在尝试整合同一个 GitHub 组织

一个 GitHub 应用程序只能向一个 GitHub 账户或 GitHub 组织安装一次。如果您需要使用已与其他 AWS 安全代理账户集成的 GitHub 组织中的存储库，则必须使用首次注册集成的 AWS 账户。

解决方法：

- 确定哪个 AWS 安全代理账户注册了 GitHub 集成
- 使用该 AWS 账户创建代理空间并连接存储库
- 如果您需要将集成转移到其他 AWS 账户，请先从原始 AWS 账户中卸载该 GitHub 应用程序，然后将其与新账户集成

后续步骤

连接 GitHub 到 AWS 安全代理后：

- 导航到要使用这些存储库的代理空间
- 选择 “启用代码审查” 或 “设置渗透测试”，将特定存储库连接到您的代理空间并配置其使用情况
- 启用自动修复，允许 AWS 安全代理提交包含漏洞修复的拉取请求
- 在 GitHub 组织设置中查看 GitHub 应用程序权限和存储库访问权限

移除集 GitHub 成

当您不再需要 AWS Security Agent 从特定 GitHub 组织或用户账户访问存储库时，请移除 GitHub 集成。在移除 AWS 控制台中的集成 GitHub 之前，必须先从中卸载 AWS 安全代理 GitHub 应用程序。

移除的先决条件

在移除 GitHub 集成之前，请确保您已经：

- 已检查哪些代理空间通过此集成连接了存储库
- 已了解影响：移除此集成将使用此集成中的存储库在所有 Agent Spaces 中断代码审查、渗透测试上下文和渗透测试补救的存储库连接
- GitHub 组织管理员权限或用户帐户所有者访问权限以卸载 GitHub 应用程序

步骤 1：从中卸载 AWS 安全代理 GitHub 应用程序 GitHub

首先，从您的 GitHub 组织或用户账户中卸载 AWS 安全代理 GitHub 应用程序。

对于 GitHub 组织：

1. 前往 github.com 并打开你的组织页面。
2. 在左侧边栏中，选择 “设置”。
3. 在 “代码、计划和自动化” 下，选择 “已安装的 GitHub 应用程序”。
4. 在列表中找到 AWS 安全代理应用程序。
5. 选择 AWS 安全代理应用程序旁边的配置。
6. 滚动到配置页面底部，然后选择 “卸载”。
7. 出现提示时确认卸载。

对于 GitHub 用户帐户：

1. 前往 github.com。
2. 选择您的头像。
3. 选择设置。
4. 在左侧边栏中，选择 “应用程序”。
5. 打开 “已安装的 GitHub 应用程序” 选项卡。
6. 在列表中找到 AWS 安全代理应用程序。
7. 选择 AWS 安全代理应用程序旁边的配置。
8. 滚动到配置页面底部，然后选择 “卸载”。

9. 出现提示时确认卸载。

步骤 2：从 AWS 安全代理中移除集成

卸载 GitHub 应用程序后，从 AWS 控制台中删除集成注册。

1. 在 AWS 安全代理管理控制台中，导航到 “集成”。
2. 在 GitHub 集成列表中找到要删除的集成。
3. 通过单击来选择集成。
4. 选择移除 。
5. 查看确认对话框，该对话框会向您发出有关影响的警告：

> **Warning**
>
> 移除此集成将影响所有通过该 GitHub 组织或用户帐户连接存储库的代理空间。这会中断：
>
> - 连接存储库的代码审查功能
> - 来自互联存储库的渗透测试上下文
> - 互联存储库的渗透测试修复功能
>
> 在继续操作 GitHub 之前，请确保您已从中卸载了 AWS 安全代理 GitHub 应用程序。

6. 如果您尚未从中卸载该 GitHub 应用程序 GitHub，则会收到一条警告。返回步骤 1 先完成卸载。
7. 如果您已卸载 GitHub 应用程序并了解其影响，请选择确认删除。
8. 该集成已从您的集成列表中删除。任何包含此集成存储库的 Agent Spaces 都无法再访问这些存储库以进行代码审查、渗透测试上下文或自动修复。

将 AWS 安全代理连接到 GitLab 存储库

将您的 AWS 安全代理连接到 GitLab 云存储库，以启用代码审查、威胁建模、渗透测试和自动修复功能。在开始之前，the section called “如何与代理空间进行集成”请查看以了解如何在座席空间中重复使用注册以及在各个功能之间共享注册。

GitLab 集成有多种用途：

- 代码审查-根据组织安全要求自动分析每个合并请求中的代码更改，并按需运行完整存储库扫描
- 威胁建模-通过分析源代码、数据流和架构，提供对应用程序的理解

- 渗透测试上下文-通过分析源代码，为渗透测试提供对应用程序的理解
- 自动修复-提交合并请求，修复安全评估期间发现的漏洞

连接 GitLab 到 AWS 安全代理需要提供具有相应权限的 GitLab 访问令牌，然后在 AWS 控制台中注册连接。

GitLab 整合的工作原理

合并请求分析发生在内 GitLab。在您提供访问令牌、连接项目并在 AWS 管理控制台中启用代码审查注释后，AWS Security Agent 会自动扫描每个新合并请求中的更改（仅对更改后的代码进行差异扫描），并将发现结果作为合并请求注释发布。

您可以在 AWS Security Agent Web 应用程序中创建和运行完整的代码审查（扫描项目的整个代码库），而不是在 AWS 安全代理 Web 应用程序中 GitLab。

渗透测试和威胁建模在 AWS 安全代理 Web 应用程序中启动。用户指定目标域并选择连接的存储库来提供应用程序上下文。如果您启用自动修复，则用户可以通过向连接的存储库打开合并请求来请求 AWS Security Agent 修复发现的问题。

> **Note**
>
> 公共 GitLab 存储库不提供自动补救功能，以避免在修复漏洞之前将其泄露。

先决条件

在开始之前，请确保您满足以下条件：

- 一个拥有维护者或所有者访问你想要连接的项目的权限的 GitLab.com 账户
- 具有您的连接类型所需范围的 GitLab 访问令牌：
  - 个人-具有所有读取权限和`api`权限的个人访问令牌。
  - 群组-`read_repository` 范围为`read_api`和的群组访问令牌。
- 在 AWS 安全代理管理控制台中配置集成的权限

> **Important**
>
> 将令牌的到期时间设置为当前日期之后的最多 365 天。

> **Note**
>
> GitLab 个人访问令牌可在多个 AWS 账户中使用。与 Atlassian 集成不同 GitHub ，将同一个 GitLab 账户连接到多个 AWS 安全代理实例没有限制。

注册 GitLab 连接

1. 在 AWS 安全代理管理控制台中，导航到 “集成”。
2. 选择添加集成。
3. 选择 GitLab，然后选择 “下一步”。
4. 在 “选择账户类型” 下，选择以下选项之一：
   - 个@@ 人-Connect 您的个人 GitLab 用户帐户。
   - 群组-Connect 包含多个项目的 GitLab 群组。输入您的群组 ID。
5. 在访问令牌字段中，粘贴您的 GitLab 访问令牌。
6. 在 “注册名称” 字段中，输入此连接的描述性名称，例如`Engineering-Team-GitLab`。有效字符包括字母、数字、句点、下划线和连字符。
7. 选择连接。

   您将返回 “集成” 页面，新连接及其注册名称将出现在该页面中。

排除 GitLab 集成故障

如果您在 GitLab 集成过程中遇到问题，请使用以下指导来解决常见问题。

令牌无效或已过期

症状

- 集成无法连接
- 以前运行的集成停止运行
- 表示身份验证失败的错误消息

解决方案

1. 在中 GitLab，导航到您的用户设置，然后选择访问令牌。

2. 验证您的令牌尚未过期。
3. 验证令牌是否具有其类型所需的范围。
4. 如果令牌已过期，请创建新令牌并在 AWS 控制台中更新集成。

速率限制

症状

- 代码审查期间出现间歇性故障
- 延迟合并请求分析

解决方案

- GitLab 对 API 请求应用速率限制。如果您有大量仓库或频繁的合并请求，则某些请求可能会受到限制。
- 等待速率限制窗口重置，或联系 GitLab 支持人员提高速率限制。

后续步骤

连接 GitLab 到 AWS 安全代理后：

- 导航到要使用这些存储库的代理空间
- 选择 “启用代码审查” 或 “设置渗透测试”，将特定项目连接到 Agent Space 并配置其使用情况
- 启用代码修复以允许 AWS 安全代理提交包含漏洞修复的合并请求
- 有关私有托管的 GitLab 实例，请参阅 the section called “将 AWS 安全代理连接到 GitLab Self-Managed”

将 AWS 安全代理连接到 GitLab Self-Managed

将您的 AWS 安全代理连接到 GitLab Self-Managed 实例，以便对托管在您自己的基础设施上的存储库启用代码审查、威胁建模、渗透测试和自动修复功能。

GitLab Self-Managed 集成的工作原理与 GitLab Cloud 相同（参见the section called “将 AWS 安全代理连接到 GitLab 存储库”），但需要对私有实例的网络连接进行额外配置。在开始之前，the section called “如何与代理空间进行集成”请查看以了解如何在座席空间中重复使用注册以及在各个功能之间共享注册。

> **Note**
>
> GitLab Self-Managed 是通过GitLab集成注册的，而不是单独的集成类型。在注册流程中，选择使用 GitLab 自托管终端节点并提供您的实例 URL。云托管 GitLab 流程如中所述。the section called “将 AWS 安全代理连接到 GitLab 存储库”

先决条件

在开始之前，请确保您满足以下条件：

- 一个 GitLab Self-Managed 实例，可以是：
  - 可通过互联网公开访问，或者
  - 可通过私人连接进行访问（请参阅the section called “Connect 连接到私有托管的源代码管理”）
- 具有您的连接类型所需范围的 GitLab 访问令牌：
  - 个人-具有所有读取权限和`api`权限的个人访问令牌。
  - 群组-`read_repository` 范围为`read_api`和的群组访问令牌。
- 维护者或所有者对您要连接的项目的访问权限
- 您的 GitLab 实例必须使用最低为 1.2 的 TLS 版本为 HTTPS 流量提供服务

> **Note**
>
> 如果您的 GitLab Self-Managed 实例使用私有证书颁发机构颁发的 TLS 证书，则可以在创建私有连接时提供证书的 PEM-encoded 公钥。这允许 AWS 安全代理信任与您的实例的 TLS 连接。

注册 GitLab Self-Managed 连接

1. 在 AWS 安全代理管理控制台中，导航到集成。
2. 选择添加集成。
3. 选择 GitLab，然后选择 “下一步”。
4. 在 “选择账户类型” 下，选择 “个人” 或 “群组”。如果您选择群组，请输入您的群组 ID。
5. 选择使用 GitLab 自托管终端节点。

6. 例如，在GitLab 自托管终端节点 URL 字段中，输入您的实例的 URL。`https://gitlab.example.com`
7. 如果您的实例不可公开访问，请选择使用私有连接连接到终端节点，然后选择现有的私有连接或创建新的私有连接。请参阅the section called “Connect 连接到私有托管的源代码管理”。
8. 在访问令牌字段中，粘贴您的 GitLab 访问令牌。
9. 在注册名称字段中，输入此连接的描述性名称。有效字符包括字母、数字、句点、下划线和连字符。
10选择连接。

    您将返回 “集成” 页面，新连接及其注册名称将出现在该页面中。

私有连接

如果您的 GitLab Self-Managed 实例不可公开访问，则必须在注册集成之前创建私有连接。有关详细the section called “Connect 连接到私有托管的源代码管理”说明，请参阅。

> **Important**
>
> Service-managed 私有连接要求 GitLab Self-Managed 实例在创建代理空间的同一 AWS 账户中运行。要进行跨账户访问，请使用自我管理的私有连接，在那里您可以提供自己的 VPC Lattice 资源配置。

排除 GitLab Self-Managed 集成故障

除了中的故障排除步骤外the section called “将 AWS 安全代理连接到 GitLab 存储库”，自管理实例还存在以下特定问题：

无法访问实例

症状

- 连接因超时或网络错误而失败
- 集成以前可以正常运行但已停止运行

解决方案

- 验证您的 GitLab 实例是否正在运行且可访问

- 如果使用私有连接，请验证 VPC Lattice 资源网关是否运行正常，以及 ENI 是否与您的实例建立了网络连接
- 验证安全组允许配置的端口上的流量
- 验证 TLS 证书有效且未过期

TLS 证书错误

症状

- 连接失败并 SSL/TLS 出现错误

解决方案

- 验证您的实例是否支持 TLS 1.2 或更高版本的 HTTPS
- 如果使用私有证书颁发机构，请确保在私有连接设置期间提供了 PEM-encoded 公钥
- 验证证书未过期

后续步骤

连接 GitLab Self-Managed 到 AWS 安全代理后：

- 导航到要使用这些存储库的代理空间
- 选择 “启用代码审查” 或 “设置渗透测试” 以连接特定项目
- 为基于合并请求的修复启用代码修复

移除集 GitLab 成

当您不再需要 AWS 安全代理从特定 GitLab 账户或群组访问存储库时，请移除 GitLab 集成。

移除的先决条件

在移除 GitLab 集成之前，请确保您已经：

- 已检查哪些代理空间通过此集成连接了存储库
- 已了解其影响：移除此集成将使用此集成中的存储库在所有 Agent Spaces 中断代码审查、渗透测试上下文、威胁建模和自动修复的存储库连接

从 AWS 安全代理中移除集成

1. 在 AWS 安全代理管理控制台中，导航到集成。
2. 找到要删除的 GitLab 集成。
3. 通过单击来选择集成。
4. 选择移除 。
5. 查看确认对话框并选择确认删除。

> **Note**
>
> 移除集成后，您可能还想撤销中的个人访问令牌 GitLab 以防止进一步访问。导航到您的 GitLab 用户设置，然后删除或撤消令牌。

同样的过程也适用于 GitLab Self-Managed 集成。

将 AWS 安全代理连接到 GitHub 企业服务器

将您的 AWS 安全代理连接到 GitHub 企业服务器 (GHES) 实例，以便对托管在您自己的基础设施上的存储库启用代码审查、威胁建模、渗透测试和自动修复功能。

GitHub 企业服务器集成提供与云托管相同的功能 GitHub （参见 Connect AWS Security Agent 连接到 GitHub 存储库 ），并提供了与自托管实例的网络连接的额外配置。在开始之前，the section called “如何与代理空间进行集成”请查看以了解如何在座席空间中重复使用注册以及在各个功能之间共享注册。

> **Note**
>
> GitHub 企业服务器是通过GitHub集成注册的，而不是通过单独的集成类型注册的。在注册流程中，选择GitHub 企业服务器作为实例类型。云托管 GitHub.com 流程如中所述。the section called “将 AWS 安全代理连接到 GitHub 存储库”

GitHub 企业服务器集成的工作原理

拉取请求分析发生在 GitHub 企业服务器中。在 AWS 管理控制台中授权连接、连接存储库并启用代码审查注释后，AWS 安全代理会自动扫描每个新拉取请求中的更改（仅对更改后的代码进行差异扫描），并将发现结果作为拉取请求注释发布。

您可以在 AWS Security Agent Web 应用程序中创建和运行完整的代码审查（扫描存储库的整个代码库），而不是在 E GitHub nterprise Server 中。

渗透测试和威胁建模在 AWS 安全代理 Web 应用程序中启动。用户指定目标域并选择连接的存储库来提供应用程序上下文。如果您启用自动修复，则用户可以通过向连接的存储库打开拉取请求来请求 AWS Security Agent 修复发现的问题。

先决条件

在开始之前，请确保您满足以下条件：

- 一个 GitHub 企业服务器实例，它可以是：
  - 可通过互联网公开访问，或者
  - 可通过私人连接进行访问（请参阅the section called “Connect 连接到私有托管的源代码管理”）
- 您的 GHES 实例的站点管理员或组织管理员访问权限
- 您的 GHES 实例必须使用最低 TLS 版本为 1.2 的 HTTPS 流量提供服务
- 在 AWS 安全代理管理控制台中配置集成的权限

> **Note**
>
> GitHub 企业服务器集成可在多个 AWS 账户中使用。

注册 GitHub 企业服务器连接

注册 GitHub 企业服务器连接使用 OAuth-based 授权流程。

> **Important**
>
> 无需关闭浏览器或导航离开，即可完成此过程中的所有步骤。如果注册过程中断，则可能需要从头开始重新启动。

1. 在 AWS 安全代理管理控制台中，导航到 “集成”。
2. 选择添加集成。
3. 选择 GitHub，然后选择 “下一步”。

4. 在实例类型下，选择GitHub 企业服务器。
5. 例如，在GitHub 企业服务器 URL 字段中，输入您的实例的 HTTPS 网址`https://github.example.com`。
6. 如果您的实例不可公开访问，请选择使用私有连接连接到终端节点，然后选择现有的私有连接或创建新的私有连接。请参阅the section called “Connect 连接到私有托管的源代码管理”。
7. 在注册详细信息部分，配置以下字段：
   a. 注册名称-输入此连接的描述性名称。有效字符包括字母、数字、句点、下划线和连字符。
   b. GitHub 帐户类型-选择 “组织” 或 “用户”。
   c. 组织名称（仅在您选择了组织时才会出现）-输入您的 GitHub 企业服务器组织的确切名称。名称区分大小写。
8. 选择连接。

> **Note**
>
> AWS Security Agent 会将您从控制台重定向到您的 GitHub 企业服务器实例完成授权。授权完成后，您返回控制台，新连接将显示在 “集成” 页面上。

私有连接

如果您的 E GitHub nterprise Server 实例不可公开访问，则必须在注册集成之前创建私有连接。有关详细the section called “Connect 连接到私有托管的源代码管理”说明，请参阅。

> **Important**
>
> Service-managed 私有连接要求 GHES 实例在创建代理空间的同一 AWS 账户中运行。要进行跨账户访问，请使用自我管理的私有连接。

> **Note**
>
> 如果您的 GHES 实例使用私有证书颁发机构颁发的 TLS 证书，请在创建私有连接时提供该证书的 PEM-encoded 公钥。

GitHub 企业服务器集成疑难解答

如果您在将 AWS 安全代理连接到 GitHub 企业服务器时遇到问题，请使用以下指导来诊断和解决常见问题。

OAuth 重定向失败

症状

- 在授权流程中，浏览器重定向失败
- 在 GHES 上授权后显示错误页面

解决方案

- 验证您的浏览器是否可以访问您的 GHES 实例
- 确保 OAuth 回传网址配置正确
- 从头开始重新启动集成过程

无法访问实例

症状

- 连接因超时或网络错误而失败

解决方案

- 验证您的 GHES 实例是否正在运行且可访问
- 如果使用私有连接，请验证 VPC 莱迪思连接
- 验证安全组允许配置的端口上的流量
- 验证 TLS 证书是否有效（最低 TLS 1.2）

后续步骤

将 GitHub 企业服务器连接到 AWS 安全代理后：

- 导航到要使用这些存储库的代理空间
- 选择 “启用代码审查” 或 “设置渗透测试” 以连接特定的存储库

- 启用代码修复以允许 AWS 安全代理提交包含漏洞修复的拉取请求

移除 GitHub 企业服务器集成

当您不再需要 AWS 安全代理从特定 GHES 实例访问存储库时，请移除 GitHub 企业服务器集成。

移除的先决条件

在删除 GHES 集成之前：

- 查看哪些代理空间通过此集成连接了存储库
- 了解影响：移除会破坏所有关联存储库的代码审查、渗透测试环境、威胁建模和自动修复

移除集成

1. 在 AWS 安全代理管理控制台中，导航到集成。
2. 找到要删除的 GitHub 企业服务器集成。
3. 选择集成。
4. 选择移除 。
5. 查看确认对话框并选择确认删除。

> **Note**
>
> 移除后，您可能还想撤销 GHES 实例上的 OAuth 应用程序。导航到您的 GHES 组织设置并删除已授权的应用程序。

找到你的 Atlassian 安装 ID

在 AWS 控制台中注册 Bitbucket 或 Confluence 集成之前，您需要从 Atlassian 网站上安装的 AWS Security Agent Forge 应用程序中找到安装 ID。您将此 ID 粘贴到注册流程中。

适用于 Bitbuc

1. 在 Bitbucket 中，导航到您的工作区设置。
2. 从侧栏中选择 Forge 应用程序。

3. 在已安装的应用程序列表中找到 Connect with AWS 安全代理。
4. 选择 “复制到剪贴板” 以复制安装 ID。

为了融合

1. 在 Confluence 中，导航到 C onfl uence 管理。
2. 从侧栏中选择 “应用程序”。
3. 在已安装的应用程序列表中找到 Connect with AWS 安全代理。
4. 选择 “复制到剪贴板” 以复制安装 ID。

在 AWS 安全代理管理控制台中完成注册时，您将需要此安装 ID。

将 AWS 安全代理连接到 Bitbucket 存储库

将您的 AWS 安全代理连接到 Bitbucket 云存储库，以启用代码审查、威胁建模、渗透测试和自动修复功能。在开始之前，the section called “如何与代理空间进行集成”请查看以了解如何在座席空间中重复使用注册以及在各个功能之间共享注册。

Bitbucket 集成有多种用途：

- 代码审查-根据组织安全要求自动分析每个拉取请求中的代码更改，并按需运行完整存储库扫描
- 威胁建模-通过分析源代码、数据流和架构，提供对应用程序的理解
- 渗透测试上下文-为渗透测试提供对应用程序的理解
- 自动修复-提交拉取请求，修复安全评估期间发现的漏洞

将 Bitbucket 连接到 AWS 安全代理需要在您的 Atlassian 网站上安装 AWS 安全代理 Forge 应用程序并完成 OAuth 授权流程。

> **Note**
>
> AWS 安全代理仅支持 Bitbucket Cloud。不支持 Bitbucket 服务器和 Bitbucket 数据中心。

Bitbucket 集成的工作原

拉取请求分析在 Bitbucket 中进行。在您安装 Forge 应用程序、连接存储库并在 AWS 管理控制台中启用代码审查注释后，AWS 安全代理会自动扫描每个新拉取请求中的更改（仅对更改后的代码进行差异扫描），并将发现结果作为拉取请求注释发布。

您可以在 AWS 安全代理 Web 应用程序中创建和运行完整的代码审查（扫描存储库的整个代码库），而不是在 Bitbucket 中。

渗透测试和威胁建模在 AWS 安全代理 Web 应用程序中启动。用户指定目标域并选择连接的存储库来提供应用程序上下文。

> **Note**
>
> 公共 Bitbucket 存储库不提供自动修复功能，以避免在修复漏洞之前将其泄露。

先决条件

在开始之前，请确保您满足以下条件：

- 具有管理员访问权限的 Bitbucket 云工作空间
- 具有站点管理员权限的 Atlassian 帐户
- 在 AWS 安全代理管理控制台中配置集成的权限

> **Important**
>
> 一个 Atlassian 站点只能与每个区域的一个 AWS 账户关联。如果您需要将同一 Bitbucket 站点连接到同一区域内不同 AWS 账户中的 AWS 安全代理，则必须先移除现有集成。

> **Note**
>
> Atlassian Forge 应用程序的定价适用于此集成。有关更多信息，请参阅 Atlassian 文档中的 Forge 平台定价。

注册 Bitbucket 连接

1. 在 AWS 安全代理管理控制台中，导航到 “集成”。
2. 选择添加集成。
3. 选择 Bitbucket，然后选择 “下一步”。
4. 按照屏幕上的说明在您的 Bitbucket 工作空间中安装 AWS 安全代理 Forge 应用程序。安装完成后，复制安装 ID。请参阅the section called “找到你的 Atlassian 安装 ID”。
5. 在 “安装 ID” 字段中，粘贴您从 Bitbucket 复制的安装 ID。
6. 例如，在 Bitbucket 工作空间字段中，输入你的 Bitbucket 工作空间标志。`acme-corp`
7. 选择授权。

   您将被重定向到 Atlassian 以授权 AWS 安全代理访问您的 Bitbucket 工作空间。授权完成后，您将返回控制台。
8. 在注册详细信息部分，输入此连接的注册名称。有效字符包括字母、数字、句点、下划线和连字符。
9. 选择连接。

> **Note**
>
> 连接后的配置延迟
>
> 选择 Connect 后，在 Atlassian 端进行配置可能需要几分钟时间。在此期间，集成会报告待处理状态。如果您尝试在置备完成之前选择存储库或启用代码审查，则可能会看到一条消息，提示安装尚未可用。请等待几分钟，然后重试。

排除 Bitbucke

如果您在 Bitbucket 集成过程中遇到问题，请使用以下指南来解决常见问题。

无法完成注册

如果注册过程中断（浏览器关闭、会话超时），Forge 应用程序可能已安装在您的 Atlassian 网站上，但未在 AWS 控制台中注册。

解决方案

- 返回 AWS 安全代理控制台并重新启动集成过程。

- Forge 应用程序仍处于安装状态，无需重新安装。

网站已连接到其他 AWS 账户

症状

- 指示 Atlassian 网站已与其他账户关联时出错

解决方案

- 每个区域一个 Atlassian 站点只能连接到一个 AWS 账户。
- 确定哪个 AWS 账户已有集成并使用该账户，或者先移除现有集成。

安装不可用

症状

- 当您选择存储库或启用代码审查时，您会看到一条消息，表明 Bitbucket 安装处于状态`PENDING_INSTALLATION`，而不是`AVAILABLE`状态。

解决方案

- 安装仍在在 Atlassian 端进行配置。配置通常会在几分钟内完成。请稍等几分钟，然后再试一次。
- 如果几分钟后状态仍不可用，请移除集成并重新注册。在重新注册之前，请从 Bitbucket 工作区中卸载 Forge 应用程序。有关说明，请参阅移除 Bitbucket 集成。如果您在未卸载之前的 Forge 应用程序的情况下重新注册，则安装可能会陷入待处理状态。

后续步骤

将 Bitbucket 连接到 AWS 安全代理后：

- 导航到要使用这些存储库的代理空间
- 选择 “启用代码审查” 或 “设置渗透测试”，将特定存储库连接到您的代理空间
- 启用代码修复以允许 AWS 安全代理提交包含漏洞修复的拉取请求

删除 Bitbucket 集成

当您不再需要 AWS 安全代理从特定 Bitbucket 工作空间访问存储库时，请移除 Bitbucket 集成。

移除的先决条件

在移除 Bitbucket 集成之前：

- 检查哪些代理空间通过此集成连接了存储库
- 了解影响：移除会破坏所有关联存储库的代码审查、渗透测试环境、威胁建模和自动修复

步骤 1：从 AWS 安全代理中移除集成

1. 在 AWS 安全代理管理控制台中，导航到集成。
2. 找到您要移除的 Bitbucket 集成。
3. 选择集成。
4. 选择移除 。
5. 查看确认对话框并选择确认删除。

第 2 步：卸载 Forge 应用程序

从 AWS 安全代理中删除集成后，从您的 Atlassian 网站上卸载 Forge 应用程序：

1. 在 Bitbucket 中，导航到工作区设置。
2. 选择 Forge 应用程序。
3. 找到 Connect with AWS 安全代理。
4. 选择卸载。

> **Important**
>
> Forge 应用程序不会自动卸载
>
> 在 AWS 安全代理控制台中移除集成不会从您的 Atlassian 网站上卸载 Forge 应用程序。如果您打算再次注册相同的 Bitbucket 工作空间，则必须先卸载 Forge 应用程序。否则，新安装可能会陷入待处理状态。

将 AWS 安全代理连接到 Confluence

将您的 AWS 安全代理连接到 Confluence Cloud，为安全评估提供文档背景。与代码提供商不同，Confluence 充当文档来源，提供威胁模型、架构文档、API 规范和其他提高安全审查质量的材料。在开始之前，the section called “如何与代理空间进行集成”请查看以了解如何在代理空间中重复使用注册。

Confluence 集成有多种用途：

- 设计审查上下文-为安全设计审查提供架构文档和设计规范
- 威胁建模-为威胁分析提供现有威胁模型和系统文档
- 渗透测试上下文-提供应用程序文档，以便在渗透测试期间加深理解

将 Confluence 连接到 AWS 安全代理需要在您的 Atlassian 网站上安装 AWS 安全代理 Forge 应用程序并完成 OAuth 授权流程。

> **Note**
>
> AWS 安全代理仅支持 Confluence 云。不支持 Confluence 数据中心和 Confluence 服务器。

Confluence 集成的工作原理

Confluence 是文档提供商，而不是源代码提供商。在您安装 Forge 应用程序并连接 AWS 管理控制台中的空间或页面后，AWS 安全代理可以访问您的 Confluence 内容，以便在安全评估期间提供背景信息。

AWS Security Agent 会读取页面内容以了解您的应用程序架构、安全要求和设计决策。这种上下文提高了设计审查、代码审查和渗透测试期间安全发现的质量和相关性。

先决条件

在开始之前，请确保您满足以下条件：

- 具有管理员访问权限的 Confluence Cloud 站点
- 具有站点管理员权限的 Atlassian 帐户
- 在 AWS 安全代理管理控制台中配置集成的权限

> **Important**
>
> 一个 Atlassian 站点只能与每个区域的一个 AWS 账户关联。如果您需要将同一 Confluence 站点连接到同一区域内不同 AWS 账户中的 AWS 安全代理，则必须先移除现有集成。

> **Note**
>
> Atlassian Forge 应用程序的定价适用于此集成。有关更多信息，请参阅 Atlassian 文档中的 Forge 平台定价。

注册 Confluence 连接

1. 在 AWS 安全代理管理控制台中，导航到 “集成”。
2. 选择添加集成。
3. 选择 Confluence，然后选择 “下一步”。
4. 按照屏幕上的说明在您的 Atlassian 网站上安装 AWS Security Agent Forge 应用程序。安装完成后，复制安装 ID。请参阅the section called “找到你的 Atlassian 安装 ID”。
5. 例如，在 Confluence 网站网址字段中，输入您的网站网址。`https://acme.atlassian.net`
6. 在 “安装 ID” 字段中，粘贴您从 Confluence 中复制的安装 ID。
7. 选择授权。

   您将被重定向到 Atlassian，以授权 AWS 安全代理访问您的 Confluence 网站。授权完成后，您将返回控制台。
8. 在注册详细信息部分，输入此连接的注册名称。有效字符包括字母、数字、句点、下划线和连字符。
9. 选择连接。

为代理空间选择页面

注册 Confluence 集成后，将特定页面连接到代理空间。选择一个页面会授予 AWS 安全代理对该页面内容的读取（提取）权限。没有每页功能选项——代理会读取每个连接的页面。在连接向导的查看步骤中，您可以删除任何不希望代理访问的页面。

排除 Confluence 集成

如果您在 Confluence 集成过程中遇到问题，请使用以下指南来解决常见问题。

无法完成注册

如果注册过程中断，Forge 应用程序可能会安装在您的 Atlassian 网站上，但不会在 AWS 控制台中注册。

解决方案

- 返回 AWS 安全代理控制台并重新启动集成过程。
- Forge 应用程序仍处于安装状态，无需重新安装。

已从 Atlassian 中卸载 Forge 应用程序

如果已从您的 Atlassian 站点卸载 AWS 安全代理 Forge 应用程序，而 AWS 安全代理中仍存在该集成：

症状

- 集成显示在 AWS 控制台中，但无法访问 Confluence 内容
- 尝试列出或阅读页面时出错

解决方案

- 从 Atlassian Marketplace 重新安装 Forge 应用程序
- 如果问题仍然存在，请移除 AWS 控制台中的集成并重新注册

网站已连接到其他 AWS 账户

解决方案

- 每个区域一个 Atlassian 站点只能连接到一个 AWS 账户。
- 确定哪个 AWS 账户已有集成并使用该账户，或者先移除现有集成。

后续步骤

将 Confluence 连接到 AWS 安全代理后：

- 导航到要使用此文档的代理空间
- 选择要包含的特定页面作为设计审查和渗透测试的上下文
- 如果需要，可通过 S3 上传其他文档（请参阅the section called “从 S3 存储桶中提供代理资源”）

移除 Confluence 集成

当您不再需要 AWS 安全代理从特定 Confluence 站点访问文档时，请移除 Confluence 集成。

移除的先决条件

在移除 Confluence 集成之前：

- 检查哪些代理空间有通过此集成连接的页面
- 了解影响：移除将打破所有代理空间使用此集成中的内容进行设计审查、渗透测试和威胁建模的文档背景

步骤 1：从 AWS 安全代理中移除集成

1. 在 AWS 安全代理管理控制台中，导航到 “集成”。
2. 找到您要移除的 Confluence 集成。
3. 选择集成。
4. 选择移除 。
5. 查看确认对话框并选择确认删除。

第 2 步：卸载 Forge 应用程序（可选）

从 AWS 安全代理中移除集成后，您可以选择从 Atlassian 网站上卸载 Forge 应用程序：

1. 在 Confluence 中，导航到 C onfl uence 管理。
2. 选择 “应用程序”。
3. 找到 Connect with AWS 安全代理。
4. 选择卸载。

Connect 连接到私有托管的源代码管理

> **Important**
>
> AWS 安全代理的私有连接处于预览版，可能会发生变化。

AWS Security Agent 可以连接到在私有网络中运行的源代码控制系统，例如 GitLab Self-Managed 实例和 GitHub 企业服务器。私有连接使用 Amazon VPC Lattice 在 AWS 安全代理和您的私有基础设施之间建立安全的连接，而无需将您的系统暴露在公共互联网上。

私有连接的工作原理

私有连接在 AWS 安全代理和您的 VPC 中的目标资源之间创建了一条安全的网络路径。在幕后，AWS 安全代理使用 Amazon VPC Lattice 来建立这种连接。VPC Lattice 是一项 AWS 应用程序联网服务，用于连接、保护和监控 VPC 和账户间服务之间的流量，而无需您管理底层网络基础设施。

创建私有连接时：

1. 您提供与目标服务具有网络连接的 VPC、子网和（可选）安全组。
2. AWS 安全代理创建服务托管资源网关，并在您指定的子网中配置弹性网络接口 (ENI)。
3. 代理使用资源网关通过私有网络路径将流量路由到目标服务的 IP 地址或 DNS 名称。

Service-managed 与自我管理的私有连接对比

AWS 安全代理支持两种私有连接模式：

Service-managed（建议用于简单设置）：

- AWS 安全代理为您创建和管理 VPC 莱迪思资源网关。
- 目标服务必须在创建代理空间的同一 AWS 账户中运行。
- 不能跨越多个 AWS 账户。

Self-managed（适用于高级或跨账户设置）：

- 您可以创建和管理自己的VPC Lattice资源配置。
- 您提供现有资源配置的 Amazon 资源名称 (ARN)。

- 支持跨账户访问（客户管理跨账户联网）。

先决条件

在创建私有连接之前，请确认您已经：

- 活跃的代理空间
- 使用已知私有 IP 地址GitLab Self-Managed 或 DNS 名称可私密访问的目标服务（或 GitHub 企业服务器）
- 目标服务必须使用 TLS 最低版本为 1.2 的 HTTPS 流量提供服务
- 您的 VPC 中的子网（1-20 个子网）。我们建议在多个可用区中选择子网以实现高可用性。
- （可选）用于控制 ENI 流量的安全组（最多 5 个）

> **Note**
>
> VPC Lattice 不支持以下可用区：`use1-az3`、`usw1-az2`、`apne1-az3`、`apne2-az2`、、`euc1-az2`、`euw1-az4cac1-az3`、`ilc1-az2`。

> **Note**
>
> 私有连接是账户级别的资源。创建私有连接后，可以在需要访问同一主机的多个集成和代理空间中重复使用该连接。

> **Note**
>
> 当您为使用 OAuth 身份验证的提供商选择私有连接时，私有连接将同时应用于提供商端点和令牌交换端点。确保私有连接配置的主机地址可以将流量路由到两个端点。

创建私有连接

1. 在 AWS 安全代理管理控制台中，导航到 “集成”。
2. 选择 “专用连接” 选项卡。

3. 选择 “创建私有连接”。
4. 在 “连接详细信息” 下，输入名称。名称可以包含小写字母、数字和连字符，并且必须以字母或数字开头和结尾。
5. 在连接详情部分，选择 AWS 安全代理如何连接到您的目标服务：
   - 要让 AWS Security Agent 创建和管理连接，请清除 “使用现有资源配置”，然后填写 “资源位置”、“访问控制” 和 “服务目标详情” 部分。
   - 要通过您已经创建的 VPC Lattice 资源配置进行路由，请选择使用现有资源配置，然后完成现有资源配置部分。服务管理部分不适用。
6. (Service-managed) 在 “资源位置” 下：
   a. 选择您的资源所在的 VPC。
   b. 选择一个或多个子网。我们建议在至少两个可用区中选择子网。
   c. （可选）选择 IP 地址类型（IPv4、IPv6 或双堆栈）。
7. (Service-managed) 在 “访问控制” 下：
   a. 选择与您连接的资源关联的安全组。
   b. （可选）在 “高级配置” 下，输入端口范围，例如，最多 11 个以逗号分隔的 TCP 端口或范围。`443, 8080-8090`
8. (Service-managed) 在 “服务目标详细信息” 下：
   a. 在主机地址字段中，输入目标服务的 DNS 主机名或 IP 地址。
   b. 对于 DNS 解析，对于可公开解析的主机地址，选择在 VPC 中（私有 DNS），选择只能在 VPC 内解析的地址。
   c. （可选）如果您的目标使用自签名证书或私有证书颁发机构，请在 “证书 PEM-encoded 公钥” 字段中输入公钥。这允许 AWS 安全代理信任 TLS 连接。
9. (Self-managed) 在 “现有资源配置” 下，对于资源配置，从列表中选择 VPC Lattice 资源配置，或者输入资源配置 ID（以开头`rcfg-`）或其 ARN。该列表显示了当前区域的资源配置。
10选择创建连接。

    连接状态更改为 “正在创建”。这一过程耗时最多 10 分钟。完成后，状态将更改为 “可用”。

使用私有连接进行集成

注册 GitLab Self-Managed 或 En GitHub terprise Server 集成时，选择 “使用专用连接连接到端点”，然后从 “专用连接” 列表中选择您的连接。AWS 安全代理通过该连接将流量路由到提供商。只有状态为 “可用” 的连接才会出现在列表中。

您也可以在注册期间选择 “创建私有连接”。AWS Security Agent 会在您创建连接时保留您的注册草稿，然后将您返回到注册流程。

安全性

- 没有公开的互联网接触-所有流量都保留在 AWS 网络上。
- Customer-controlled 安全组-您可以管理入站和出站流量规则。
- Service-linked 权限最低的角色-AWS 安全代理使用服务相关角色，其范围仅限于标记为的资源。`AWSSecurityAgentManaged`

> **Note**
>
> 如果您的组织有限制VPC Lattice API操作的服务控制策略 (SCP)，则服务管理资源网关是通过服务相关角色创建的。确保您的 SCP 允许服务相关角色执行必要的操作。

验证私有连接

私有连接达到 “可用” 状态后，请验证连接：

- 确保您的目标服务正在运行并接受预期端口上的连接
- 验证连接到 ENI 的安全组是否允许目标端口上的出站流量
- 验证您的服务的安全组是否允许来自您的 VPC CIDR 范围内的 VPC Lattice 数据平面 IP 的入站流量
- 确认子网路由表允许 ENI 子网和您的服务之间的流量

删除私有连接

1. 在 AWS 安全代理控制台中，导航到 “集成”。
2. 选择 “专用连接” 选项卡。
3. 选择要删除的专用连接。
4. 选择删除。

> **Important**
>
> 您无法删除集成当前正在使用的私有连接。先移除集成，然后删除私有连接。

后续步骤

- the section called “将 AWS 安全代理连接到 GitLab Self-Managed”-Connect 私有 GitLab 实例
- the section called “将 AWS 安全代理连接到 GitHub 企业服务器”-Connect 私有 GitHub 企业服务器

将代理连接到私有 VPC 资源

如果您要运行渗透测试的应用程序在公共互联网上不可用，则需要向 AWS 安全代理提供 VPC 配置。AWS 安全代理将使用此 VPC 配置（包括 VPC、子网和安全组）来访问应用程序。

> **Note**
>
> 在私有 VPC 中测试终端节点时，仅允许解析为已知私有 IP 范围内的 IP 的终端节点（有关更多信息，请参阅 VPC CIDR 块）。允许使用以下 IPv4 和 IPv6 范围：
>
> ```
> 10.0.0.0/8
> 172.16.0.0/12
> 192.168.0.0/16
> fd00::/8
> ```

> **Note**
>
> 连接到子网时，AWS 安全代理将在为渗透测试配置的子网中创建 ENI（弹性网络接口）。此 ENI 没有关联的公有 IP 地址，这意味着它无法与公有子网中的 VPC 互联网网关通信。如果您的渗透测试需要开放的互联网接入，请改用带有关联的 VPC NAT 网关的私有子网

您可以从 AWS 管理控制台向 AWS 安全代理授予对 VPC 的常规访问权限。在安全客户端 Web 应用程序中，用户选择渗透测试的特定配置。

在代理空间中添加 VPC

1. 导航到代理空间概述页面
2. 选择操作，然后选择编辑渗透测试配置
3. 在 VPC 标题下，指定 VPC、子网和安全组

您最多可以添加 5 个 VPC。

所需的代理空间服务角色权限

要使用 VPC 运行渗透测试，您的代理空间服务角色必须包含以下权限：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeNetworkInterfaceAttribute",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "ec2:CreateNetworkInterfacePermission",
      "Resource": "arn:aws:ec2:{{region}}:{{accountId}}:network-interface/*",
      "Condition": {
        "ArnEquals": {
          "ec2:Subnet": [
            "arn:aws:ec2:{{region}}:{{accountId}}:subnet/[[subnetIds]]"
          ]
        }
      }
    }
  ]
}
```

在安全客户端 Web 应用程序中为渗透测试选择特定 VPC 配置

1. 导航到渗透测试概述页面
2. 选择您需要为其添加 VPC 配置的渗透测试，然后选择修改渗透测试详情

3. 选择 “下一步” 进入 “VPC 资源” 部分
4. 选择 VPC、子网和安全组
5. 选择 “下一步” 进入最后一部分并保存渗透测试

> **Note**
>
> Cross-account 目前支持对使用 AWS Resource Access Manager 共享的 VPC 资源（子网和安全组）进行渗透测试。用于身份验证凭证的 Secrets Manager 密钥和 Lambda 函数必须在与您的 AWS 安全代理设置相同的 AWS 账户中配置。

对另一个 AWS 账户中的 VPC 资源运行渗透测试

您可以使用 AWS Resource Access Manager 对与您的账户共享的 VPC 资源进行渗透测试。两个账户必须属于同一 AWS 组织。

1. （可选）为您的 AWS 组织启用自动资源共享

```
aws ram enable-sharing-with-aws-organization
```

2. 使用拥有 VPC 资源的 AWS 账户的证书，与渗透测试所有者账户共享子网和安全组资源

```
aws ram create-resource-share \
    --name SharePentestResources \
    --resource-arns <subnet ARN> <security group ARN> \
    --principals <penetration test owner account ID>
```

3. 导航到代理空间概述页面
4. 选择渗透测试并找到服务角色名称
5. 验证 IAM 角色是否授予对共享 VPC 资源的访问权限
6. 选择操作，然后选择编辑渗透测试配置
7. 在 VPC 标题下，指定共享 VPC、子网和安全组并保存更新的配置。
8. 导航到 AWS 安全代理 Web 应用程序上的渗透测试概述页面
9. 选择您需要为其添加 VPC 配置的渗透测试，然后选择修改渗透测试详情

10更新渗透测试以使用共享 VPC 资源

配置权能

为代理空间（包括补救、S3 源和目标域）启用和配置渗透测试、代码审查和威胁建模。

启用渗透测试

将 AWS 安全代理配置为对您的应用程序运行自主渗透测试。此设置使 AWS 安全代理能够访问您的 AWS 资源、验证域名所有权并执行全面的安全测试，以识别您的 Web 应用程序和 API 中可利用的漏洞。

启用渗透测试允许 AWS Security Agent 持续验证您的应用程序安全，而不会出现手动测试的延迟。通过配置必要的 AWS 集成，您可以确保 AWS Security Agent 可以测试公有和私有应用程序 CloudWatch、将发现记录到并访问证书以进行经过身份验证的测试。

在此过程中，您将配置目标域，可以选择设置 VPC、 CloudWatch 日志、证书存储和 Lambda 函数以进行测试，配置 S3 集成以提供更多上下文，并通过 IAM 角色设置服务访问权限。

先决条件

在开始之前，请确保您满足以下条件：

- 具有创建 IAM 角色的管理权限的 AWS 账户
- 域名所有权验证功能（修改 DNS 或 HTTP 记录）
- 将要测试的目标域或应用程序作为目标
- （可选）测试私有应用程序时的 VPC 配置详情
- （可选）如果向 AWS 安全代理提供了其他项目，则为 S3 存储桶

步骤 1：配置域名

在向导的第一步中，指定要测试的目标域以及 AWS Security Agent 应如何验证所有权。

1. 在目标域名部分，在域名字段中输入您的域名。
2. 选择验证方法：
   - DNS_TXT — 通过在域名的 DNS 配置中添加 TXT 记录来证明域名所有权。
   - HTTP_ROU TE — 通过在域名的特定网址上托管验证文件来证明域名所有权。

- PRIV@@ ATE_VPC-仅适用于私有 VPC 渗透测试。验证域是否解析为私有 CIDR 范围内的 IP
- 有关更多信息，请参阅 the section called “启用应用程序域进行渗透测试”。

3. 选择添加其他域名以添加其他域名（总共最多 5 个）。
4. 选择 “下一步” 继续进行域名验证。

第 2 步：验证域名

在向导的第二步中，验证您配置的每个域的所有权。AWS 安全代理需要经过验证的所有权才能执行渗透测试。

1. 查看目标域表中列出的域。
2. 对于每个域名，选择它并根据您选择的方法触发验证：
   - Route 53 域名（相同的 AWS 账户）：选择One-click 验证。AWS 安全代理会自动创建 DNS 记录并完成验证。
   - DNS TXT（其他 DNS 提供商）：复制验证令牌，在您的 DNS 注册商处添加 TXT 记录，然后选择域名并选择验证。
   - HTTP 路由：将验证令牌放在 Web 服务器上所需的路由路径上，然后选择域并选择 “验证”。有关更多信息，请参阅 the section called “启用应用程序域进行渗透测试”。
3. 选择 “下一步” 继续进行可选配置。

> **Note**
>
> Sub-domains 使用 DNS TXT 或 HTTP 路由验证时，经过验证的域名不需要单独验证。对于私有 VPC 验证，目标终端节点域名必须与为验证配置的完整域名匹配。

> **Note**
>
> 对于 VPC 内的私有域，即使域验证状态为 “无法访问”，您也可以继续。每次 pentest 运行开始时，AWS 安全代理都会尝试对私有终端节点进行域名验证。

步骤 3：（可选）配置其他功能

向导的第三步允许您配置可选的 AWS 资源以扩大渗透测试范围。此步骤中的所有部分都是可选的，默认情况下是折叠的。

（可选）配置 VPC 设置

如果您计划测试托管在 VPC 中的私有目标域，请为 AWS 安全代理配置 VPC 设置。此部分是可选的，默认情况下处于折叠状态。

1. 展开 VPC 部分。
2. 在 VPC 下拉列表中，选择托管您的私有目标域的 VPC。
3. 在子网下拉列表中，选择 AWS 安全代理应使用的 VPC 子网：

> **Tip**
>
> 要获得高可用性，请从多个可用区域中选择多个子网。确保您的子网包含用于出站连接的 NAT 网关。

4. 在安全组下拉列表中，选择 AWS 安全代理应使用的 VPC 安全组：

> **Important**
>
> 确保您的安全组允许 AWS 安全代理执行渗透测试的出站连接。

（可选）配置 CloudWatch 日志

配置 CloudWatch 为存储渗透测试运行的日志。此部分是可选的，默认情况下处于折叠状态。

1. 展开 “CloudWatch 日志” 部分。
2. 在 “日志组” 下拉列表中，选择现有的 CloudWatch 日志组
3. 如果未选择任何日志组，AWS 安全代理将创建一个名为`/aws/securityagent/<agent name>/<pentest id>`且具有相应权限的日志组。

> **Note**
>
> 确保您的 IAM 角色有权写入所选 CloudWatch 日志组。

（可选）为测试凭证配置密钥

如果您的应用程序需要身份验证凭证进行测试，AWS 安全代理可以安全地从 AWS Secrets Manager 中检索这些证书。此部分是可选的，默认情况下处于折叠状态。

1. 展开 “机密” 部分。
2. 选择包含您的应用程序证书的 AWS Secrets Manager 密钥。
3. 在 Web 应用程序中配置渗透测试时，可以引用这些密钥进行经过身份验证的测试。

> **Important**
>
> 证书经过加密并存储在 AWS Secrets Manager 中。确保您的 IAM 角色有权访问 Secrets Manager for AWS 安全代理，以便在测试期间使用这些证书。

（可选）为测试凭证配置 Lambda 函数

配置可在测试期间为您的应用程序提供凭证的 Lambda 函数。此部分是可选的，默认情况下处于折叠状态。

1. 展开 L ambda 函数部分。
2. 选择可以为您的应用程序提供身份验证凭证的 Lambda 函数。
3. AWS 安全代理将在渗透测试期间调用这些函数以动态获取证书。

> **Note**
>
> 确保您的 IAM 角色有权调用指定的 Lambda 函数。Lambda 函数应以预期格式返回证书，以便 AWS 安全代理在测试期间使用。

（可选）配置 S3 存储桶

如果您计划上传文档或项目以作为输入提供给 AWS 安全代理，请提供 S3 存储桶详细信息。此部分是可选的，默认情况下处于折叠状态。

1. 展开 S 3 存储桶部分。
2. 在存储桶字段中，输入或搜索您的 S3 存储桶名称。

> **Note**
>
> 您也可以稍后连接 GitHub 存储库或直接在 Web 应用程序中上传文件。所提供的信息可以确保全面覆盖，减少误报，并提供可行的结果。

（可选）配置服务访问权限

AWS 安全代理需要一个 IAM 角色才能访问您的 AWS 资源（VPC、 CloudWatch 日志组、Secrets Manager、Lambda 函数等）以进行渗透测试。此部分是可选的，默认情况下处于折叠状态。

1. 展开 “服务访问权限” 部分。
2. 默认情况下，AWS 安全代理使用具有渗透测试所需权限的默认 IAM 角色。
3. 要自定义 IAM 角色，请选择以下选项之一：
   a. 创建默认角色 — AWS 安全代理会自动创建具有必要权限的新 IAM 角色
   b. 使用现有服务角色-从下拉菜单中选择现有 IAM 角色
4. 如果使用现有角色：
   a. 选择 “选择现有角色” 下的下拉菜单
   b. 从列表中选择您的 IAM 角色
   c. 如果需要，选择刷新图标以更新列表

> **Note**
>
> 默认 IAM 角色包括访问渗透测试所需的 VPC 资源、 CloudWatch 日志组、Secrets Manager、Lambda 函数和其他服务的权限。除非您有特定的安全要求，否则建议使用默认 IAM 角色。

第 4 步：保存并启用渗透测试

配置完所有必需的设置后，为您的 AWS 安全代理启用渗透测试。

1. 查看所有配置部分以确保准确性。
2. 选择保存。
3. AWS 安全代理会验证您的配置并创建必要的 AWS 资源。

后续步骤

启用渗透测试后：

- 在 AWS 安全代理 Web 应用程序中配置渗透测试范围
- 为测试结果设置通知首选项
- 在发现渗透测试结果时对其进行审查并做出回应
- （可选）配置其他存储库以进行发现补救

有关运行和管理渗透测试的更多信息，请参阅 AWS 安全代理 Web 应用程序文档。

为 GitHub 仓库启用拉取请求代码审查

现在，拉取请求代码审查已配置为代码审查设置向导的一部分。当您将 GitHub 存储库连接到代理空间时，您可以为各个存储库启用拉取请求注释，让 AWS 安全代理自动分析拉取请求并发布安全发现。

有关完整的设置说明，请参阅the section called “启用代码审查”。

有关拉取请求结果在中的显示方式 GitHub 以及如何响应这些结果的信息，请参阅the section called “查看拉取请求中的代码安全发现”。

启用代码审查

通过连接 GitHub 存储库或 S3 存储桶中的源代码，将您的代理空间配置为启用代码审查。代码审查会分析您的源代码是否存在安全漏洞，以及是否符合组织的自定义安全要求。

设置代码审查配置是一项代理 Space-wide 操作。您连接的集成和 S3 存储桶将跨功能共享，包括代码审查和渗透测试。

完成设置后，用户可以在 AWS 安全代理 Web 应用程序中创建和运行代码审查，以扫描存储库和 S3 源是否存在安全问题。

> **Note**
>
> 如果您已经将 GitHub 存储库连接到 Agent Space（例如，通过渗透测试设置），则代码审查已启用。您可以跳过此设置，直接转到 Web 应用程序来创建代码审查。请参阅the section called “创建代码审查”。

先决条件

在开始之前，请确保您满足以下条件：

- 在 AWS 管理控制台中创建的代理空间（参见the section called “创建代理空间”）
- 以下源代码输入中至少有一个：
  - 安装了 AWS 安全代理 GitHub 应用程序的 GitHub 组织或用户账户（参见the section called “将 AWS 安全代理连接到 GitHub 存储库”）
  - 包含您要查看的源代码的 S3 存储桶
- 为您的代理空间配置集成的权限
- （可选）如果您计划使用安全要求验证，则在安全要求包中至少启用一项安全要求（请参阅the section called “管理安全要求”）

访问代码审查设置向导

导航到 Agent Space 的代码审查设置。

1. 在 AWS 安全代理控制台中，选择您的代理空间。
2. 从以下位置之一选择 “启用代码审查”：
   - 守则审查卡
   - “代码审查” 选项卡，然后选择 “启用代码审查”

> **Tip**
>
> 如果您希望 AWS Security Agent 自动处理代码审查反馈，请同时启用代码修复。有关详细信息，请参阅the section called “使用户能够开始修复渗透测试和代码审查结果”。

您将被定向到安装程序代码审查配置向导。

第 1 步：Connect 集成、存储库和存储桶

在向导的第一步中，连接源代码输入并配置代码审查设置。必须至少添加一个 GitHub 存储库或 S3 存储桶才能继续。

> **Important**
>
> 此处配置的集成和 S3 存储桶将在您的代理空间中共享。更改适用于代码审查和渗透测试功能。

Connect GitHub 存储库

从您的授权 GitHub 组织或用户帐户中添加 GitHub 存储库。选择 “添加” 将打开两步式 Connect GitHub 向导，在该向导中，您首先选择存储库，然后配置 AWS Security Agent 可以对每个存储库执行的操作。

1. 在 “已连接集成” 部分中，选择 “添加”。
2. 选择包含您要查看的存储库的 GitHub 注册。
3. 在 “Connect GitHub 存储库” 步骤中，选中要连接的每个存储库对应的复选框。
4. 选择 “下一步” 转至 “管理权能”。

> **Note**
>
> 在代码审查和渗透测试期间，以只读方式访问连接的存储库以进行代码分析。您可以配置写入操作，例如在下一步中发布评论和打开修正拉取请求。

> **Note**
>
> 如果您尚未注册 GitHub 集成，请选择 “设置” 以导航到 “集成” 页面，您可以在其中授权 AWS 安全代理 GitHub 应用程序。有关更多信息，请参阅 the section called “将 AWS 安全代理连接到 GitHub 存储库”。

配置 GitHub 存储库功能

在 Connect GitHub 向导的 “管理功能” 步骤中，选择 AWS 安全代理可以在每个存储库中执行的操作。代码审查注释和代码修正是按存储库独立设置的。

1. 对于每个存储库，切换代码审查评论，让 AWS Security Agent 将安全发现作为拉取请求的评论发布。

2. 对于每个存储库，开启代码修复，让 AWS 安全代理修复发现的问题。启用后，Web 应用程序用户可以请求修复发现结果的拉取请求，拉取请求作者可以发表评论`@AWS-Security-Agent fix all findings.`以让代理处理其审阅评论。
3. 选择 “保存” 以应用您的选择并返回代码审查设置向导。

为仓库启用代码审查注释后：

- 当拉取请求标记为 “准备审核” 时，AWS Security Agent 会自动分析这些请求。不分析拉取请求草稿。
- 安全发现以审查评论的形式直接发布在拉取请求上，并附有具体的补救指南。
- 分析使用您配置的代码审查设置（安全漏洞、自定义要求或两者兼而有之）。

> **Note**
>
> 拉取请求评论仅适用于私有 GitHub 仓库。

为存储库启用代码修复后，Web 应用程序用户可以开始修复该存储库上的代码审查和渗透测试结果，AWS Security Agent 会以拉取请求的形式提供每个修复。有关更多信息，请参阅 the section called “使用户能够开始修复渗透测试和代码审查结果”。

有关拉取请求结果如何显示在中 GitHub 以及如何响应这些结果的更多信息，请参阅the section called “查看拉取请求中的代码安全发现”。

添加 S3 存储桶

添加包含源代码或上下文资源的 S3 存储桶以供代码审查。

1. 在 S3 存储桶部分，选择添加 S3 资源。
2. 输入存储桶的 S3 URI 或包含源代码的前缀。
3. 选择添加。

> **Note**
>
> 您最多可以添加 10 个 S3 资源。S3 存储桶是跨功能共享的，包括代码审查和渗透测试。

> **Tip**
>
> 您可以添加包含源代码、配置文件、基础设施即代码模板或其他您希望 AWS Security Agent 分析安全问题的项目的 S3 存储桶。

配置代码审查设置

配置 AWS 安全代理在代码审查期间分析的安全问题的类型。此设置适用于在此代理空间中启用代码审查的所有存储库和源。

1. 在代码审查设置部分，选择以下选项之一：
   - 安全要求验证-验证代码是否符合您在安全要求包中启用的安全要求。
   - 安全漏洞发现-识别代码中的常见安全漏洞。
   - 安全要求和漏洞发现 — 分析代码是否符合您启用的安全要求和常见的安全漏洞。这是默认设置。

> **Note**
>
> 启用安全要求验证后，AWS Security Agent 会根据您在安全要求包中启用的安全要求检查代码。如果您选择安全要求验证，但未启用至少一项安全要求，AWS 安全代理将无法识别基于要求的发现。有关安全要求的更多信息，请参阅the section called “管理安全要求”。

1. 选择 “下一步” 继续进行可选配置。

步骤 2：可选配置

在向导的第二步中，为您的代码审查环境配置可选的 CloudWatch 日志记录和服务访问设置。

（可选）配置 CloudWatch 日志

配置 CloudWatch 日志组以捕获和分析代码审查期间的应用程序行为。

1. 展开 “CloudWatch 日志” 部分。
2. 在日志组下拉列表中，从您的 AWS 账户中选择一个或多个现有 CloudWatch 日志组。

> **Note**
>
> 如果您未选择日志组，AWS Security Agent 会自动创建一个默认日志组来存储代码审查执行日志。

（可选）配置服务访问权限

配置 AWS 安全代理用来访问您的 AWS 资源（例如 S3 存储桶和用于代码审查的 CloudWatch 日志）的 IAM 服务角色。

1. 展开 “服务访问权限” 部分。
2. 选择下列选项之一：
   - 创建默认角色 — AWS 安全代理会自动创建一个具有代码审查所需权限的新 IAM 角色。
   - 使用现有服务角色-从下拉菜单中选择现有 IAM 角色。
3. 如果使用默认角色，请输入服务角色名称。该名称在账户中的所有角色中必须是唯一的，必须使用字母数字字符和字`+=,.@-_`符，并且不能包含空格。

> **Note**
>
> 服务角色名称的最大长度为 64 个字符。如果您不选择现有角色，则会自动创建服务角色。

1. 选择 “保存” 以完成设置。

设置完成后

完成代码审查设置向导后：

- 您的 Agent Space 页面上的代码审查卡显示为 “就绪” 状态。
- 用户可以启动 Web 应用程序并创建代码审查以扫描连接的存储库和 S3 源。
- 您可以随时通过在 “代码审查” 选项卡上选择 “编辑配置” 来修改您的配置。

编辑代码审查配置

要在初始设置后修改代码审查配置，请执行以下操作：

1. 在 AWS 安全代理控制台中导航到您的代理空间。
2. 选择 “代码审查” 选项卡。
3. 选择 Edit configuration (编辑配置)。
4. 根据需要更新您的集成、S3 存储桶、代码审查设置、 CloudWatch 日志或服务访问权限。
5. 选择保存。

后续步骤

设置代码审查配置后：

- 启动 Web 应用程序以创建和运行代码审查（请参阅the section called “创建代码审查”）
- 随着 GitHub 代码库的增长，连接其他存储库或 S3 存储桶
- 为组织特定的验证配置安全要求包（请参阅）the section called “管理安全要求”
- 查看拉取请求结果在中的显示方式 GitHub （请参阅the section called “查看拉取请求中的代码安全发现”）

启用威胁建模

通过连接源代码存储库和配置 AWS 资源，配置您的代理空间以启用威胁建模。威胁建模会分析应用程序的架构，并从源代码、设计文档或两者中识别安全威胁。

设置威胁建模配置是一项代理 Space-wide 操作。您连接的集成和 S3 存储桶在威胁建模、代码审查和渗透测试等功能之间共享。

完成设置后，用户可以在 AWS 安全代理 Web 应用程序中创建和运行威胁模型。

> **Note**
>
> 如果您已经将存储库或 S3 存储桶连接到 Agent Space（例如，通过代码审查或渗透测试设置），则威胁建模可能已经启用。您可以直接访问 Web 应用程序来创建威胁模型。请参阅the section called “创建威胁模型”。

先决条件

在开始之前，请确保您满足以下条件：

- 在 AWS 管理控制台中创建的代理空间（参见the section called “创建代理空间”）
- 为您的代理空间配置集成的权限
- （可选）安装了 AWS 安全代理应用程序的 GitHub GitLab、或 Bitbucket 组织（请参阅将 AWS 安全代理连接到 GitHub 存储库、将 A WS 安全代理连接到存储库或将 AWS 安全代理连接到 Bitbucket 存储 GitLab 库）

访问威胁建模设置向导

导航到 Agent Space 的威胁建模配置。

1. 在 AWS 安全代理控制台中，选择您的代理空间。
2. 从 “威胁模型” 卡片或 “威胁模型” 选项卡中选择 “配置威胁模型”。

您将被定向到 “配置威胁模型” 向导，其中包含两个可选步骤。

步骤 1：Connect 源代码存储库（可选）

连接您要为其启用威胁建模的 GitHub GitLab、或 Bitbucket 存储库。威胁模型本身是在 Web 应用程序中创建和查看的。

> **Important**
>
> 此处配置的集成将在您的代理空间中共享。更改适用于威胁建模、代码审查和渗透测试功能。

Connect 存储库

1. 在 “已连接集成” 部分中，选择添加。
2. 选择包含您要使用的存储库的注册。
3. 选中要连接的每个存储库对应的复选框。
4. 选择 “保存” 以应用您的选择。

> **Note**
>
> 如果您尚未注册集成，请选择设置以导航到集成页面，您可以在其中授权 AWS Security Agent 应用程序。有关更多信息，请参阅将 AWS 安全代理连接到 GitHub 存储库、将 AWS 安全代理连接到存储 GitLab 库或将 AWS 安全代理连接到 Bitbucket 存储库。

1. 选择 “下一步” 继续进行可选配置。

步骤 2：可选配置

为您的威胁建模环境配置 S3 存储桶、 CloudWatch 日志记录和服务访问设置。这些设置与您的 Agent Space 上的其他功能共享。

S3 存储桶（可选）

添加包含您希望代理在威胁建模期间用作上下文的源代码的 S3 存储桶。

1. 在 S3 存储桶部分，选择添加 S3 资源。
2. 输入存储桶或前缀的 S3 URI。
3. 选择添加。

> **Note**
>
> 您最多可以添加 10 个 S3 资源。

CloudWatch 日志（可选）

配置 CloudWatch 日志组以捕获和分析威胁模型运行期间的应用程序行为。

1. 在CloudWatch 日志部分，从您的 AWS 账户中选择一个或多个现有 CloudWatch 日志组。

服务访问

配置 AWS 安全代理用来访问您的 AWS 资源（例如 S3 存储桶和威胁建模 CloudWatch 日志）的 IAM 服务角色。需要服务角色才能启用威胁建模。

1. 在服务访问权限部分，从下拉列表中选择现有的 IAM 服务角色，或者将其留空以自动创建服务角色。

> **Note**
>
> 如果您不选择现有角色，则会自动创建服务角色。

1. 选择 “保存” 以完成配置。

设置完成后

完成威胁建模配置后：

- 您的 Agent Space 页面上的威胁模型卡片显示就绪状态。
- 用户可以启动 Web 应用程序，并根据关联的源代码、上传的范围文档、Confluence 页面或这些输入的组合创建威胁模型。

后续步骤

启用威胁建模后：

- 启动 Web 应用程序以创建和运行威胁模型（请参阅创建威胁模型）
- 随着代码库的增长，连接其他存储库或 S3 存储桶
- 连接 Confluence 以允许选择页面作为范围文档（请参阅将 AWS 安全代理连接到 Confluence）

使用户能够开始修复渗透测试和代码审查结果

在 AWS 管理控制台中，您可以启用自动修复，这样 AWS Security Agent Web 应用程序的用户就可以针对特定发现请求修复。AWS Security Agent 以 GitHub 拉取请求的形式将每个修复程序发送到受影响的存储库。

您可以从代理空间中启用每个存储库的修复。“渗透测试” 和 “代码审查” 选项卡打开相同的存储库配置向导，因此您可以使用任一选项卡来管理 “启用自动修复” 设置。补救适用于这两种功能的发现，因此您只需要为每个存储库启用一次即可。

先决条件

在开始之前，请确保您满足以下条件：

1. 已启用渗透测试（参见the section called “启用渗透测试”）或代码审查（请参阅the section called “启用代码审查”）
2. 为您的 GitHub 组织安装并授权了 AWS 安全代理 GitHub 应用程序（参见the section called “将 AWS 安全代理连接到 GitHub 存储库”）

打开存储库配置向导

选择与仓库设置方式相匹配的入口点。

从 “渗透测试” 选项卡中

使用此路径添加用于渗透测试的 GitHub 存储库并配置补救措施。

1. 导航到 “代理空间” 概述页面。
2. 选择 “渗透测试” 选项卡。
3. 选择拥有您的存储库的 GitHub 注册账户：
   a. 如果您尚未将任何 GitHub 注册与代理空间相关联，请在 “Connect GitHub to AWS 安全代理信息” 框中选择 “添加” 以选择注册。
   b. 如果已关联一个或多个 GitHub 注册，请在 “已连接的集成” 部分中选择 “添加” 以选择其他注册。
4. 选择 “下一步” 以选择 GitHub 存储库。
5. 选择 “下一步” 配置存储库功能。

从 “代码审查” 选项卡中

当您的存储库已连接以进行代码审查时，请使用此路径。

1. 导航到 “代理空间” 概述页面。
2. 选择 “代码审查” 选项卡。
3. 在 GitHub 存储库表中，选择编辑以打开存储库配置向导。

启用自动修复并保存

进入存储库功能后，请通过上述任一路径：

1. 在 “已启用自动修复” 列中，将要修复的每个存储库标记为 “已启用”。
2. 选择 Connect 保存配置。

Web 应用程序的用户现在可以开始对这些存储库中的发现进行修复，AWS Security Agent 将使用建议的修复程序打开拉取请求。

从 S3 存储桶中提供代理资源

您可以授予 AWS 安全代理访问 S3 存储桶中资源的权限，例如 API 文档、威胁模型文档和支持渗透测试的源代码。

您可以在 AWS 管理控制台中授予 AWS 安全代理对 S3 存储桶的读取权限。在安全客户端 Web 应用程序中，用户从 S3 中选择相关的单个资源。

1. 导航到 “代理空间” 概述页面
2. 选择操作，然后选择编辑渗透测试配置
3. 在 S3 存储桶部分下，定义包含 S3 存储桶的 S3 URI。您可以添加多个 S3 存储桶。

启用应用程序域进行渗透测试

在对应用程序进行渗透测试之前，您需要添加目标域并验证所有权。AWS 安全代理将仅对经过验证的域执行渗透测试。

> **Note**
>
> 您无需验证您的应用程序可能使用的辅助域。您只需要验证要针对哪个域名进行渗透测试即可。

步骤 1：添加目标域名

要添加目标域，请导航到 Agent Space 概述页面上的 “渗透测试” 选项卡。根据您是否已经配置了渗透测试，请使用以下方法之一：

- First-time 设置：选择 “设置渗透测试” 以打开渗透测试向导。在第一步中，输入域名并选择验证方法。
- 向现有配置中添加域：在 “目标域” 表中，选择 “添加域”。输入域名并在模态中选择一种验证方法。

您可以添加基础域名或子域名，例如example.com或billing.example.com。AWS 建议使用您有权创建TXT记录的子域。

> **Note**
>
> 当您使用 DNS TXT 或 HTTP 路由验证验证域名时，该域的子域名会自动被覆盖。例如，验证example.com允许您进行测试，api.example.com也可以billing.example.com不进行额外验证。对于私有 VPC 验证，目标终端节点域名必须与为验证配置的完整域名匹配。

选择以下验证方法之一：

- DNS TXT 记录：通过与您的 DNS 提供商创建 DNS TXT 记录来证明域名所有权。
- HTTP 路由：通过在您的 Web 服务器上创建包含 AWS 安全代理提供的唯一令牌的路由，证明域名所有权。
- 私有 VPC：仅可用于私有 VPC 渗透测试。验证域是否解析为私有 CIDR 范围内的 IP。配置的域名必须与任何关联的目标终端节点的完整域名相匹配

第 2 步：验证域名所有权

添加域名后，使用您选择的方法验证所有权。您可以随时从 “目标域” 表中触发验证，方法是选择域并选择 “验证”。

使用 DNS TXT 记录进行验证

AWS 安全代理生成 TXT DNS 记录值。您必须向 DNS 提供商添加此记录才能证明所有权。

如果域名是在 Route 53（相同的 AWS 账户）中注册的：

AWS 安全代理可以自动创建 DNS 记录。从 “目标域” 表中选择域并选择One-click 验证。AWS 安全代理创建 DNS 验证记录并自动完成验证。

如果该域名是在其他 DNS 提供商处注册的：

1. 复制 AWS 安全代理提供的 TXT 记录值。
2. 向您的 DNS 注册商添加 TXT 记录。
3. 返回到 “目标域” 表，选择该域，然后选择 “验证”。

使用 HTTP 路由验证进行验证

此方法通过在您的 Web 服务器上放置一个唯一的令牌来证明域名所有权。只有域所有者或授权的 Web 管理员才能在 Web 服务器上创建路由，这证明了所有权。

1. 在 Web 服务器上使用以下路径创建文件：

```
.well-known/aws/securityagent-domain-verification.json
```

2. 使用以下格式将 AWS 安全代理提供的令牌放在文件中：

```
{
  "tokens": ["<insert-token>"]
}
```

3. 返回到 “目标域” 表，选择该域，然后选择 “验证”。AWS 安全代理向验证 URL 发送 HTTPS GET 请求并验证令牌。
4. 如果该域名可在公共互联网上访问，则在运行验证之前，请确保您的域名具有有效的 SSL 证书。

> **Note**
>
> 如果您的域名已在多个代理空间中注册，并且您正在使用 HTTP 路由验证，则可以将两个代理空间的令牌放在同一个tokens数组中。

绕过域名所有权验证

有权在终端上执行渗透测试但无法完成所有权验证的客户可以向我们申请手动验证。请提交客户支持案例以提出此请求。您的请求必须包括您的业务用例和手动所有权验证的理由（即您被授权对端点进行渗透测试的原因）。

用于渗透测试的托管目标域

在 AWS 管理控制台中，您可以添加和管理代理空间使用的目标域以进行渗透测试。这些目标域名需要经过验证，然后才能用于渗透测试。有关验证目标域的更多信息，请参阅 the section called “启用渗透测试”

先决条件

在开始之前，请确保您满足以下条件：

1. 已启用渗透测试（参见the section called “启用渗透测试”）

管理目标域资源

- 导航到目标域名概述页面。
- 您应该会看到与您的账户关联的所有目标域名资源
  - 如果您没有看到任何与您的账户关联的目标域名，请按照中的the section called “启用渗透测试”步骤创建和验证目标域名
- 目标域可以在代理空间之间重复使用并共享验证状态
  - 要将现有目标域添加到代理空间，请导航到代理空间的 “渗透测试” 选项卡。选择 “添加域名”，然后在 “域名” 字段的 “从以前注册的可用域名中选择” 下选择所需的域名
  - 目标域必须先与代理空间相关联，然后才能将其用于渗透测试
- 从代理空间中移除域并不会删除该域。关联的目标域名可以从目标域名概述页面中永久删除

验证目标域名

要在渗透测试中使用目标域，必须首先使用以下方法之一对其进行验证：

- Route 53 域名（相同的 AWS 账户）：选择One-click 验证。AWS 安全代理会自动创建 DNS 记录并完成验证。
- DNS TXT（其他 DNS 提供商）：复制验证令牌，在您的 DNS 注册商处添加 TXT 记录，然后选择域名并选择验证。
- HTTP 路由：将验证令牌放在 Web 服务器上所需的路由路径上，然后选择域并选择 “验证”。有关更多信息，请参阅 the section called “启用应用程序域进行渗透测试”。
- 私有 VPC：验证目标域的 IP 是否在私有 CIDR 范围内（the section called “将代理连接到私有 VPC 资源”有关私有 CIDR 范围的列表，请参阅）。渗透测试目标端点域名必须与为验证配置的完整域名相匹配。仅适用于私有 VPC 渗透测试

> **Note**
>
> 对于 DNS TXT、HTTP 路由和 Route 53 验证，已验证域的子域名会自动被覆盖，无需单独验证。要进行私有 VPC 验证，必须对每个域进行单独验证。

管理安全要求

使用 AWS-managed 软件包、自定义包或从您自己的文档中导入的要求，定义 AWS Security Agent 在设计和代码审查期间评估的安全要求。

管理安全要求

将安全客户端用于分析应用程序 AWS 的安全要求整理成安全需求包。包是安全要求的集合。您可以启用 AWS托管包，创建自己的自定义包，并通过上传安全文档来生成自定义包的要求。

概述

安全要求定义了安全标准或策略， AWS 安全代理在设计审查和代码审查期间根据这些标准或策略来评估您的应用程序。当设计或代码不符合要求时， AWS 安全客户端会报告发现结果。每项安全要求都属于一个安全要求包。安全要求在所有代理空间中共享，并在设计和代码审查期间进行评估。

AWS 安全客户端提供两种类型的软件包，分别显示在不同的选项卡上：

- 托管安全需求包 — 根据行业标准和最佳实践 AWS提供的安全要求包。您可以立即启用这些包，以根据常见的安全策略进行评估，而无需自定义配置。托管包中的要求是只读的。您可以使用 AWS-managed 需求作为自定义要求的模板。有关可用托管包的列表，请参阅AWS 托管安全要求包。
- 自定义安全要求包 — 您为强制执行组织的特定策略和标准而创建的包。您可以从头开始在自定义包中构建需求，将自定义 AWS托管需求作为起点，或者通过上传安全文档来生成这些需求。

> **Note**
>
> 合规性框架包旨在帮助确定可能与某些框架的合规性相关的安全要求。他们不评估您的合规性，也不保证您会通过审计。

您可以将包作为一个单元启用和禁用。启用软件包后， AWS 安全代理会在设计和代码审查期间根据该包中的要求评估您的应用程序。

> **Note**
>
> 启用或禁用包适用于新的设计审查和代码审查。现有评论不受影响。

启用或禁用托管包

启用 AWS托管包以根据一组 AWS维护的安全要求评估您的应用程序。当您不再需要它时，请将其禁用。

1. 在 AWS 控制台中，导航到 AWS 安全客户端。
2. 在导航窗格中，选择安全要求。
3. 选择托管安全要求包选项卡。
4. 选择要启用或禁用的包。
5. 请执行以下操作之一：
   a. 要启用该包，请选择 “启用包”。
   b. 要禁用该包，请选择 “禁用包”。

> **Tip**
>
> 选择一个包以查看其包含的安全要求。选择一项要求以查看其完整定义，包括其适用性、合规性标准和补救指南。

创建自定义包

创建自定义包，对特定于您的组织的安全要求进行分组。创建包后，手动向其添加需求，自定义 AWS托管需求，或者上传文档以生成需求。

1. 在 AWS 控制台中，导航到 AWS 安全客户端。
2. 在导航窗格中，选择安全要求。
3. 选择 “自定义安全要求包” 选项卡。
4. 选择 “创建安全需求包”。
5. 输入安全要求包名称和可选描述。
6. 选择 “创建安全需求包”。

> **Note**
>
> 创建包后，您可以添加或上传源文档，以生成包的安全要求。

手动添加安全要求

在自定义包中添加安全要求以定义安全客户端强制 AWS 执行的安全标准。

1. 在 AWS 控制台中，导航到 AWS 安全客户端。
2. 在导航窗格中，选择安全要求。
3. 选择 “自定义安全要求包” 选项卡，然后选择要向其添加要求的包。
4. 选择手动添加需求。
5. 配置以下字段：
   a. 安全要求名称-输入用于标识安全控制的描述性名称，例如`enforce-encryption-at-rest`（最多 80 个字符）。
   b. 描述-简要描述此安全要求检查的内容（最多 500 个字符）。
   c. 适用性-描述应评估此安全要求的方案、系统类型或条件。包括应将要求标记为 NOT_APPLIABLE 的场景（最多 10,000 个字符）。
   d. 合规性标准 — 定义什么构成此安全要求的合规性与不合规性。提供 AWS 安全客户端在评估合规性时要查找的指标、示例和技术细节（最多 10,000 个字符）。
   e. 补救指南（可选）-提供有关如何修复违规行为的指导，包括组织内部文档或标准的链接（最多 10,000 个字符）。
6. 请执行以下操作之一：
   a. 要在不启用需求的情况下创建需求，请选择创建安全需求。
   b. 要创建需求并启用该需求，请选择创建并启用安全要求。

> **Note**
>
> 只有当包含要求的包处于启用状态时，才会在安全审查期间对其进行评估。要控制是否对包进行评估，请启用或禁用该包。

自定义 AWS-托管安全要求

创建使用 AWS托管要求作为起点的自定义安全要求。 AWS Security Agent 会预先填充托管需求中的需求详细信息，然后您可以对其进行编辑以适合您的组织。

1. 在 AWS 控制台中，导航到 AWS 安全客户端。
2. 在导航窗格中，选择安全要求。

3. 选择 “自定义安全要求包” 选项卡，然后选择要向其添加要求的自定义包。
4. 选择 “创建自定义安全要求”。
5. 要预先填充表单，请在 “自定义 AWS托管安全要求” 部分，搜索并选择要用作 AWS模板的托管要求。
6. 编辑任何字段以满足贵组织的需求。
7. 请执行以下操作之一：
   a. 要在不启用需求的情况下创建需求，请选择创建安全需求。
   b. 要创建并立即启用该要求，请选择创建并启用安全要求。

> **Note**
>
> 自定义 AWS托管需求会创建独立的自定义安全要求。以后对 AWS-managed 要求的更改不会影响您的自定义版本。

通过上传文档生成需求

从现有的安全文档中生成自定义包的安全要求，而不是手动编写每个要求。 AWS Security Agent 会读取您上传的文档，识别与安全相关的内容，并生成您可以查看和编辑的结构化要求。有关完整程序，请参阅根据文档生成安全要求。有关在上传的文档中应包含哪些内容的指导，请参阅为需求生成准备文档。

> **Important**
>
> 上传新的源文档会重新生成该包的所有要求。所有现有需求，包括您手动添加的要求，都将被替换。

编辑自定义安全要求

修改自定义安全要求以更新其定义、标准或补救指南。您无法编辑 AWS托管包中的要求。

1. 在 AWS 控制台中，导航到 AWS 安全客户端。
2. 在导航窗格中，选择安全要求。
3. 选择 “自定义安全要求包” 选项卡，然后选择包含该要求的包。
4. 选择要编辑的要求，然后选择 “编辑自定义安全要求”。

5. 根据需要更新必填字段。安全要求名称在创建后无法更改。
6. 选择 “更新安全要求”。

> **Note**
>
> 对安全要求的更改适用于新的设计审查和代码审查。现有评论不受影响。

启用或禁用包

启用或禁用安全需求包以控制 AWS 安全客户端是否评估其包含的要求。您可以将包作为一个单元启用和禁用。

1. 在 AWS 控制台中，导航到 AWS 安全客户端。
2. 在导航窗格中，选择安全要求。
3. 选择包裹类型的选项卡，然后选择包装。
4. 请执行以下操作之一：
   a. 要启用该包，请选择 “启用包”。
   b. 要禁用该包，请选择 “禁用包”。

> **Note**
>
> 启用包后，将在安全审查期间对包中的要求进行评估。当包被禁用时，不会对其要求进行评估。

删除自定义安全要求

当您不再希望安全客户端评估自定义 AWS 安全要求时，请将其删除。

1. 在 AWS 控制台中，导航到 AWS 安全客户端。
2. 在导航窗格中，选择安全要求。
3. 选择 “自定义安全要求包” 选项卡，然后选择包含该要求的包。
4. 选择一个或多个安全要求，然后选择 “删除安全要求”。
5. 确认删除操作。

> **Note**
>
> 删除安全要求后，将不再在 future 的审查中对其进行评估。在过去的评论中，这些要求仍然以只读结果显示。

定义安全要求的最佳实践

创建安全要求时，请遵循以下指导方针，以帮助 AWS 安全代理准确评估您的应用程序并提供可行的调查结果。

安全要求名称-使用标识安全控制的清晰、具体的名称。避免使用无法传达要求目的的通用术语。

描述-解释控制检查的内容及其缓解的风险。这可以帮助用户理解该要求为何重要。

适用性-描述应评估需求的工作负载、系统类型或条件。说明何时应将要求标记为 NOT_APPLIABLE，并说明具体情况，以避免误报。诸如 “此控件适用于所有...” 和 “如果... 则标记为不适用” 之类的短语设置了明确的范围界限。

合规标准 — 将其分为两部分：什么表明合规，什么表示不合规。具体说明技术指标，并包括边缘情况。从 “如果设计演示出来就符合要求...” 开头，然后是技术细节，然后是 “如果... 设计不合规...”，再加上表示违规的模式。

补救指南-提供包含技术细节和配置示例的指导。包括指向组织内部文档的链接，以便开发人员拥有修复违规行为所需的资源。

后续步骤

配置安全需求包后：

- 创建设计审查或代码审查，根据您启用的软件包评估您的应用程序。
- 启用渗透测试以补充您的设计和代码审查。
- 授予用户访问 AWS 安全客户端 Web 应用程序的权限。

AWS 托管安全要求包

AWS 托管安全需求包是根据行业标准和最佳实践 AWS 创建和维护的安全要求的集合。您可以让托管包在设计审查和代码审查期间根据其要求评估您的应用程序，而无需自己编写要求。

托管包中的安全要求是只读的。您无法更改它们。您可以使用 AWS托管需求作为模板来创建自定义需求，然后编辑副本以适合您的组织。有关更多信息，请参阅管理安全要求。

AWS 随着时间的推移更新托管包。在托管包中 AWS 添加或修改需求时，更改将适用于启用该套餐的每个账户。

> **Note**
>
> 合规性框架包旨在帮助确定可能与某些框架的合规性相关的安全要求。他们不评估您的合规性，也不保证您会通过审计。

AWS 托管包：ASA 基础包

适用于大多数工作负载的一组基本安全要求。该包涵盖了常见的应用程序安全问题，例如身份验证和授权、数据保护、日志记录和输入验证。启用此包可以为您的设计和代码审查建立安全基准。

AWS 托管包：AWS Well-Architected Pack

安全要求源自《 AWS Well-Architected 框架》的安全支柱。此包根据有关保护数据、管理身份和权限以及检测和响应安全事件的 AWS 指导对您的应用程序进行评估。有关该框架的更多信息，请参阅安全支柱-AWS Well-Architected 框架。

AWS 托管包：NIST CSF Pack

源自 NIST 网络安全框架 (CSF) 的安全要求。此包根据框架中提取的控制区域（例如身份和访问管理、数据安全和保护技术）评估您的应用程序。启用此包可使您的设计和代码审查与 NIST CSF 保持一致。

AWS 托管包：PCI DSS Pack

安全要求源自支付卡行业数据安全标准 (PCI DSS)。该软件包根据与处理持卡人数据相关的控制区域（例如访问控制、传输和静态数据的加密以及记录）评估您的应用程序。启用此包可使您的设计和代码审查与 PCI DSS 保持一致。

根据文档生成安全要求

上传现有的安全文档，生成自定义包的安全要求。 AWS Security Agent 会读取您上传的文档，识别与安全相关的内容，并生成包含适用性、合规性标准和补救指南的结构化安全要求。您可以先查看和编辑生成的需求，然后再将其用于审核。

当您已经记录了安全策略、标准或指导方针时，请使用此方法来代替手写每项要求。有关上传文档中应包含哪些内容的指导，请参阅为需求生成准备文档。

支持的文件格式

您可以上传以下文件格式：

- DOC
- DOCX
- MD
- PDF
- TXT

生成需求

1. 在 AWS 控制台中，导航到 AWS 安全客户端。
2. 在导航窗格中，选择安全要求。
3. 选择 “自定义安全要求包” 选项卡。
4. 创建包，或选择现有的自定义包以生成需求。
5. 选择 “选择文件”，或者将您的文档拖放到上传区域。
6. 选择 “生成需求”。
7. 等待 AWS 安全客户端处理文档。生成在后台运行，大文件可能需要几分钟。在生成过程中，您无法重新开始上传该包。
8. 查看生成的安全要求。根据需要编辑、删除或添加需求。如果您希望 AWS 安全客户端在安全审查期间评估其要求，请启用该包。

> **Note**
>
> AWS Security Agent 在工作负载级别生成需求，重点关注文档中与安全相关的内容。它生成的需求数量取决于您提供的内容。查看生成的需求，确认它们反映了您的意图。

将要求替换为新的上传

上传新的源文档会重新生成对包的要求。

> **Important**
>
> 当您将新的源文档上传到包时，AWS 安全客户端会重新生成该包的所有要求。所有现有需求，包括您手动添加的要求，都将被替换。为了保持您当前的要求，请不要上传新文件。

1. 在 AWS 控制台中，导航到 AWS 安全客户端。
2. 在导航窗格中，选择安全要求。
3. 选择 “自定义安全要求包” 选项卡，然后选择该包。
4. 开始新的上传，并确认上传新的源文档取代了现有要求。
5. 选择您的文件，然后选择 “生成需求”。

后续步骤

- 查看并启用生成的需求。请参阅管理安全要求。
- 创建设计审查或代码审查，以对照包评估您的应用程序。

为需求生成准备文档

当您根据文档生成安全要求时，Sec AWS urity Agent 会读取您的文档，并根据它找到的安全相关内容生成结构化要求。您无需以任何特定的方式格式化文档，也不需要预先写好适用性、合规标准和补救指南。 AWS 安全客户端从您提供的内容中获取这些信息。上传你已经拥有的、用你自己的话写好的安全文档。

本页介绍要包含的内容，以便 AWS 安全客户端生成准确、有用的要求。有关上传程序和文件限制，请参阅根据文档生成安全要求。

要包括什么

AWS 安全客户端会查找描述您的安全期望的内容。涵盖以下主题的文档提出了最严格的要求：

- 访问控制和授权
- 身份验证
- 数据保护和加密
- 网络安全
- 日志和审计

- 事件响应
- 漏洞和补丁管理
- 适用于您的工作负载的合规义务

良好的来源文档包括安全策略、工程标准、控制目录、架构指南和安全编码标准。

在工作负载级别写入

AWS Security Agent 生成适用于单个工作负载或应用程序的需求，与其在设计和代码审查期间评估的范围相同。描述如何构建和操作安全工作负载的内容会生成需求。 Organization-wide 诸如多账户策略、root 用户管理和账户级日志记录设置之类的治理主题不在此范围内，不会产生工作负载要求。

描述结果，而不是单一的实现

陈述您期望的安全结果，而不是单一的规定产品或配置。 AWS Security Agent 生成的要求侧重于所需的安全功能，并允许多个有效实施。例如，与命名一项特定服务或设置的声明相比，“静态数据已加密”会产生更清晰、适用范围更广的要求。

具体说明好看的样子

您的文档对预期行为的描述越具体， AWS 安全客户端就越能更好地定义合规性标准。在可能的情况下，描述合规设计所展示的内容以及表明违规的内容。与具体的、与工作量相关的陈述相比，模糊或纯粹的抱负陈述产生的要求更少、更弱。

查看你得到的回报

生成的每项要求都包括 AWS 安全客户端从您的文档中得出的名称、适用性、合规性标准和补救指南。查看生成的需求，编辑任何需要完善的需求，然后启用您想让 Sec AWS urity Agent 评估的需求。有关更多信息，请参阅管理安全要求。

管理用户和访问权限

控制谁可以访问每个代理空间，并配置 AWS 安全代理使用的 IAM 角色和加密密钥。

授予用户访问 AWS 安全代理 Web 应用程序的权限

AWS Security Agent 为用户提供了两种访问 Web 应用程序的方法，具体取决于您在设置过程中配置代理空间的方式。

访问方法概述

IAM 身份中心 (SSO)-如果您在创建代理空间时启用了 IAM 身份中心，则用户可以直接通过 SSO 访问 Web 应用程序。您可以通过 AWS Security Agent 控制台将用户分配到代理空间，然后用户使用其身份中心证书登录。

管理员访问权限-拥有 AWS 控制台访问权限的用户可以通过 AWS 管理控制台中代理空间概述页面上的管理员访问链接启动 Web 应用程序。

通过 IAM 身份中心 (SSO) 授予访问权限

如果您使用 IAM Identity Center 配置了代理空间，则可以使用 AWS 安全代理控制台将用户分配到代理空间。

通过 AWS 安全代理控制台分配用户

1. 在 AWS 安全代理管理控制台中，导航到您的代理空间。
2. 选择 Web 应用程序选项卡。
3. 在用户表中，选择添加用户。
4. 从 IAM 身份中心选择现有用户或创建新用户。
5. 确认用户分配。

> **Tip**
>
> 分配到代理空间的用户可以通过使用 SSO 凭证通过 IAM 身份中心登录来访问 Web 应用程序。

使用 SSO 访问 Web 应用程序

将用户分配到代理空间后：

1. 用户导航到代理空间的 Web 应用程序 URL。

> **Tip**
>
> 选择 “复制 Web 应用程序 URL”，在 AWS 安全代理控制台的代理空间详细信息页面上查找网络应用程序 URL。用户应将此 URL 加入书签以便于访问。

2. 用户使用其 SSO 凭据登录。
3. 身份验证后，用户可以选择代理空间并开始进行安全评估。

授予访问 IAM-only 权限

如果您为代理空间配置了 IAM-only 访问权限，则拥有 AWS 控制台访问权限的用户可以通过管理员访问链接启动 Web 应用程序。

使用管理员访问链接

1. 登录 AWS 安全代理控制台。
2. 导航到您要访问的代理空间。
3. 在 Agent Space 登录页面的 Web 应用程序选项卡上，选择管理员访问按钮以启动 Web 应用程序。
4. Web 应用程序将在新选项卡中打开，用户将自动通过身份验证。

> **Note**
>
> 管理员访问链接仅适用于已通过 AWS 控制台进行身份验证并具有相应 AWS 安全代理权限的用户。此方法不需要配置 IAM 身份中心。

后续步骤

授予用户访问 Web 应用程序的权限后：

- 用户可以创建和管理渗透测试配置和运行
- 用户可以创建和管理设计评审
- 用户可以查看安全发现和补救指南
- 为安全发现配置通知首选项

为 AWS 安全代理创建 IAM 角色

AWS 安全代理以三种方式使用 IAM 角色：

1. 应用程序角色：创建 AWS 安全代理应用程序时使用。对于 IAM Identity Center 和管理员访问链接用例，该服务扮演此角色以授予 WebApp 用户与 AWS 安全代理 API 交互的权限。

2. 渗透测试服务角色：在创建代理空间时指定为可用角色列表。稍后，WebApp 用户在创建渗透测试时会选择其中一个角色。AWS 安全代理服务扮演此角色在测试期间访问您的 AWS 资源。
3. 操作者角色：用于对您的目标 Web 应用程序（例如 AWS API Gateway API API）的请求进行身份验证和授权。这些角色是在创建代理空间期间提供的。AWS 安全代理扮演角色与您的目标应用程序进行交互。

应用程序角色

应用程序角色用于在服务中创建 AWS 安全代理应用程序。对于 IAM Identity Center 和管理员访问链接身份验证方案，AWS 安全代理服务扮演此角色来授予 WebApp 用户与 AWS 安全代理 API 交互的必要权限。

所需权限

此角色需要以下权限：

- 调用 AWS 安全代理 API 操作
- 读取和写入应用程序配置数据
- 访问用户会话信息
- 管理 WebApp 用户的身份验证令牌

信任策略

信任策略必须允许 AWS 安全代理服务担任此角色：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

权限策略

该角色应包括以下权限：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "securityagent:GetApplication",
        "securityagent:UpdateApplication",
        "securityagent:ListAgentInstances",
        "securityagent:CreatePentestSession"
      ],
      "Resource": "arn:aws:securityagent:*:*:application/*"
    }
  ]
}
```

> **Note**
>
> 根据您的特定应用程序要求和最低权限原则自定义权限。

渗透测试服务角色

渗透测试服务角色是在创建代理空间时以可用角色列表的形式指定的。当 WebApp 用户创建渗透测试时，他们会选择其中一个角色。然后，AWS 安全代理服务将扮演此角色来访问和测试您的 AWS 资源。

所需权限

在渗透测试期间，此角色需要访问和分析您的 AWS 资源的权限：

- 阅读并描述 VPC 配置和网络拓扑
- 检查 EC2 实例、安全组和网络 ACL
- 分析 IAM 策略和资源权限
- 读取 CloudWatch 日志和指标
- 访问与安全测试相关的 AWS 服务配置

信任策略

信任策略必须允许 AWS 安全代理服务在渗透测试操作中扮演此角色：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "your-external-id"
        }
      }
    }
  ]
}
```

> **Note**
>
> 在允许跨账户或服务访问时，使用外部 ID 可提高安全性。

权限策略

该角色应包括对您的 AWS 资源的只读访问权限。考虑使用以下托管策略：

- `SecurityAudit`-用于安全审计的 AWS 托管策略
- `ViewOnlyAccess`- Read-only 访问大多数 AWS 服务

或者创建具有特定权限的自定义策略：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*",
        "vpc:Describe*",
        "iam:Get*",
        "iam:List*",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "cloudwatch:Describe*",
        "cloudwatch:Get*",
        "cloudwatch:List*",
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": "*"
    }
  ]
}
```

Important

仅授予渗透测试范围所需的最低权限。根据您要在安全测试中包含的 AWS 服务，查看和调整权限。

演员角色

Actor 角色用于在渗透测试期间对目标 Web 应用程序的请求进行身份验证和授权。这些角色是在代理空间创建期间提供的，AWS 安全代理假设它们与您的目标应用程序终端节点（例如 AWS API Gateway API、Lambda 函数 URL 或其他 AWS-hosted 应用程序）进行交互。

所需权限

此角色需要以下权限：

- 调用 API Gateway 端点
- 执行 Lambda 函数
- 访问特定于应用程序的 AWS 资源
- 使用目标应用程序的身份验证机制进行身份验证
- 对您的应用程序终端节点执行 HTTP 操作

信任策略

信任策略必须允许 AWS 安全代理服务担任此角色：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "your-external-id"
        }
      }
    }
  ]
}
```

权限策略

权限取决于您的目标应用程序架构。以下是常见场景的示例：

适用于 API Gateway 应用程序

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "execute-api:Invoke"
      ],
      "Resource": "arn:aws:execute-api:us-east-1:*:your-api-id/*"
    }
  ]
}
```

对于 Lambda 函数网址

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunctionUrl",
        "lambda:InvokeFunction"
      ],
      "Resource": "arn:aws:lambda:us-east-1:*:function:your-function-name"
    }
  ]
}
```

适用于使用 Cognito 身份验证的 Application Load Balancer

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cognito-idp:InitiateAuth",
        "cognito-idp:RespondToAuthChallenge"
      ],
      "Resource": "arn:aws:cognito-idp:us-east-1:*:userpool/your-user-pool-id"
    }
  ]
}
```

Important

配置 Actor Role 权限以匹配目标应用程序的身份验证和授权要求。该角色的访问权限应与安全客户端在渗透测试期间模拟的用户或服务相同。

AWS 安全代理的客户托管密钥

默认情况下，AWS 安全客户端使用 AWS托管的加密密钥对所有静态数据进行加密。您可以选择使用密 AWS 钥管理服务 (AWS KMS) 中的客户托管密钥来加密您的数据，从而完全控制保护资源的加密密钥。

AWS 安全代理支持资源级客户托管密钥。在创建顶级资源（例如代理空间或集成）时，可以指定 KMS 密钥来加密属于该资源及其子资源的所有数据。例如，在创建代理空间时指定客户托管密钥会加密与该代理空间关联的所有数据，包括代理空间配置、渗透测试配置、作业和执行详细信息、安全发现、发现的端点和屏幕截图。您还可以提供已使用您自己的客户托管密钥加密的 AWS 资源，例如 S3 存储桶、CloudWatch 日志日志组或 Secrets Manager 密钥。有关所需的 KMS 权限，请参阅the section called “所需的 KMS 权限”。

客户托管密钥的工作原理

AWS 安全代理使用 AWS KMS 分层密钥环使用客户管理的密钥对您的数据进行加密。当您在资源创建期间指定 KMS 密钥时，该服务会创建一个受您的 KMS 密钥保护的分支密钥并将其存储在 Amazon DynamoDB-based 密钥存储中。对于每个加密操作，分层密钥环都会从活动分支密钥中派生一个唯一的包装密钥，该密钥会加密每个请求的唯一数据加密密钥。

分层密钥环具有以下优点：

- Per-resource 加密范围 — 每个顶级资源都有自己的分支密钥，因此不同资源的数据使用不同的密钥进行加密。
- 自动密钥轮换- AWS 安全客户端定期轮换分支密钥。轮换后，新数据将使用新的分支密钥版本进行加密，而旧版本则保留以解密先前加密的数据。

加密上下文

AWS 安全代理使用您的 KMS 密钥在所有加密操作中使用加密上下文。加密上下文是一组非秘密密钥值对，为加密操作提供额外的经过身份验证的数据。

加密上下文密钥遵循以下格式`aws:securityagent:_<resource-type>_`，其中 *`<resource-type>`* 是要加密的资源类型（例如，`agent-space`或`integration`）。该值是资源的亚马逊资源名称 (ARN)。

> **Note**
>
> 对于集成，加密上下文密钥包含`aws-crypto-ec:`前缀，由此产生了格式`aws-crypto-ec:aws:securityagent:integration`。

您可以使用加密上下文将 KMS 密钥策略的范围缩小到特定的资源类型。有关更多信息，请参阅 the section called “所需的 KMS 权限”。

Default encryption (默认加密)

当您在未指定客户托管密钥的情况下创建资源时， AWS 安全代理会使用底层存储服务（Amazon DynamoDB 和 Amazon S3）提供的 AWS托管加密密钥对资源进行加密。默认加密不需要其他配置。

应用程序的默认 KMS 密钥

您可以在应用程序级别设置默认 KMS 密钥。配置默认 KMS 密钥后，如果您在资源创建期间未明确指定 KMS 密钥，则 AWS 安全客户端会将其用作新代理空间和集成的备用密钥。

要设置默认 KMS 密钥，请在使用 AWS CLI 或 SDK 创建或更新应用程序时指定`defaultKmsKeyId`参数。在应用程序安装过程中，控制台会提示您指定默认 KMS 密钥。

当您在资源创建期间明确指定 KMS 密钥时，该密钥优先于应用程序级别的默认密钥。

先决条件

在配置客户托管密钥之前，请完成以下先决条件：

- 在 KMS 中创建对称加密 KMS 密 AWS 钥。密钥必须满足以下要求：
  - 密钥类型：对称
  - 密钥用法：加密和解密
  - 关键规格：`SYMMETRIC_DEFAULT`
  - 密钥状态：已启用

  有关说明，请参阅《密 AWS 钥管理服务开发人员指南》中的创建密钥。
- 配置 KMS 密钥策略和 IAM 策略以授予 AWS 安全代理所需的权限。有关更多信息，请参阅 the section called “所需的 KMS 权限”。

所需的 KMS 权限

AWS 在两种情况下，安全代理需要 KMS 权限：

- 加密 AWS 安全代理中的数据-当您为代理空间或集成指定客户托管密钥时，该服务需要使用该密钥对您的数据进行加密操作的权限。
- 使用 CMK-encrypted AWS 资源-当您提供使用客户托管密钥加密的 AWS 资源（例如 S3 存储桶、CloudWatch 日志日志组或 Secrets Manager 密钥）时，服务需要权限才能使用这些密钥来访问加密的资源。

AWS 安全代理使用不同的身份访问您的 KMS 密钥，具体取决于操作：

- AWS 管理控制台操作-管理员在管理控制台中创建或 AWS 管理资源（例如，创建代理空间或更新应用程序）时，该服务使用管理员角色调用 AWS KMS。
- Web 应用程序操作 — 当 IAM Identity Center 用户通过 AWS 安全代理 Web 应用程序访问数据（例如，查看渗透测试结果或开始渗透测试）时，该服务使用在 AWS 安全代理设置期间创建的应用程序角色来调用 AWS KMS。
- 访问客户提供的资源-当服务在渗透测试期间访问客户提供的 AWS 资源（例如 S3 存储桶、CloudWatch 日志日志组和 Secrets Manager 密钥）时，它将使用渗透测试服务角色调用 KMS。AWS
- 异步工作流程-对于在任何用户会话之外运行的后台操作（例如，渗透测试执行、代码审查处理和分支密钥轮换），该服务使用自己的服务主体 (`securityagent.amazonaws.com`) 代表您调用 AWS KMS。

以下各节描述了每个身份所需的权限。

密钥策略

您的 KMS 密钥策略必须授予 AWS 安全代理使用该密钥进行加密操作的权限。所需的密钥策略声明取决于您计划加密的资源类型以及您向服务提供的 CMK-encrypted AWS 资源。

代理空间的关键政策

以下密钥策略向 AWS 安全客户端授予加密和解密 Agent Space 数据所需的权限，包括渗透测试结果和屏幕截图。

替换策略中的以下占位符值：

- *111122223333* — 您的 AWS 账户 ID

- *MyRole* — 用于在控制台中管理 AWS 安全代理的 IAM 角色
- *MyApplicationRole* — 安装 AWS 安全客户端期间创建的应用程序角色
- *us-east-1* — 您使用 AWS 安全代理的 AWS 区域

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowKeyMetadataValidationForApplicationAndAgentSpaces",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyRole"
      },
      "Action": [
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "securityagent.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowUseOfHierarchicalKeyringForAgentSpaces",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyRole"
      },
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:securityagent:agent-space":
 "arn:aws:securityagent:us-east-1:111122223333:agent-space/*"
        },
        "StringEquals": {
```

```
          "kms:ViaService": "securityagent.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowSynchronousDataAccessForAgentSpaces",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyApplicationRole"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:securityagent:agent-space":
"arn:aws:securityagent:us-east-1:111122223333:agent-space/*"
        },
        "StringEquals": {
          "kms:ViaService": "securityagent.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowAsynchronousDataAccessForAgentSpaces",
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:securityagent:agent-space":
"arn:aws:securityagent:us-east-1:111122223333:agent-space/*"
        },
        "ArnLike": {
```

```
          "aws:SourceArn": "arn:aws:securityagent:us-east-1:111122223333:agent-space/*"
        }
      }
    },
    {
      "Sid": "AllowAsynchronousDataAccessForCodeRemediation",
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "*"
    }
  ]
}
```

Important

要轮换分支密钥，您的 KMS 密钥策略必须向 AWS 安全代理服务委托人 (`securityagent.amazonaws.com`) 授予`kms:GenerateDataKeyWithoutPlaintext`、`kms:ReEncryptTo`和`kms:ReEncryptFrom`权限，否则分支密钥轮换将失败。有关所需权限的更多信息，请参阅轮换分支密钥。

集成的关键政策

以下密钥策略向 AWS 安全代理授予加密和解密集成数据（包括代码审查结果）所需的权限。

替换策略中的以下占位符值：

- *111122223333* — 您的 AWS 账户 ID
- *MyRole* — 用于在控制台中管理 AWS 安全代理的 IAM 角色
- *us-east-1* — 您使用 AWS 安全代理的 AWS 区域

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Sid": "AllowKeyAccessValidationForIntegrations",
      "Effect": "Allow",
      "Principal": {
          "AWS": "arn:aws:iam::111122223333:role/MyRole"
      },
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws-crypto-ec:aws:securityagent:integration":
"arn:aws:securityagent:us-east-1:111122223333:integration/*"
        },
        "StringEquals": {
          "kms:ViaService": "securityagent.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowKeyMetadataValidationForIntegrations",
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": "kms:DescribeKey",
      "Resource": "*"
    },
    {
      "Sid": "AllowAsynchronousDataAccessForIntegrations",
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt",
        "kms:Encrypt",
```

```
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom"
      ],
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:securityagent:us-east-1:111122223333:integration/*"
        },
        "StringLike": {
          "kms:EncryptionContext:aws-crypto-ec:aws:securityagent:integration":
 "arn:aws:securityagent:us-east-1:111122223333:integration/*"
        }
      }
    }
  ]
}
```

> **Note**
>
> 如果您想对多种资源类型使用相同的 KMS 密钥，则可以将代理空间、集成和安全需求包密钥策略声明合并到单个密钥策略中。

安全需求包的关键策略

以下密钥策略向 AWS 安全代理授予加密和解密安全需求包数据所需的权限。安全要求包不使用 Web 应用程序角色。该策略使用两种访问路径：

- 同步路径 — 当您调用 API 时，该服务使用您转发的凭据代表您调用 AWS KMS（通过`kms:ViaService`条件）。
- 异步路径-对于可以使用包的异步操作，该服务使用自己的服务主体直接调用 AWS KMS。

替换策略中的以下值：

- *111122223333* — 您的 AWS 账户 ID
- *MyRole* — 用于调用安全需求包操作的 IAM 角色
- *us-east-1* — 您使用 AWS 安全代理的 AWS 区域

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowKeyMetadataValidation",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyRole"
      },
      "Action": [
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "securityagent.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowUseOfHierarchicalKeyringForSecurityPacks",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyRole"
      },
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "securityagent.us-east-1.amazonaws.com"
        },
        "StringLike": {
          "kms:EncryptionContext:aws:securityagent:security-requirement-pack":
"arn:aws:securityagent:us-east-1:111122223333:security-requirement-pack/*"
        }
      }
    },
    {
      "Sid": "AllowAsynchronousDataAccessForSecurityPacks",
      "Effect": "Allow",
```

```
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:securityagent:security-requirement-pack":
 "arn:aws:securityagent:us-east-1:111122223333:security-requirement-pack/*"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:securityagent:us-east-1:111122223333:security-
requirement-pack/*"
        }
      }
    }
  ]
}
```

该政策包含三个声明：

- AllowKeyMetadataValidation— 授予`kms:DescribeKey`权限，以便 AWS 安全代理可以验证您的客户托管密钥是否存在、已启用，并在资源创建期间指定 CMK 时使用对称加密。使用`kms:ViaService`条件来确保呼叫通过服务发出。
- AllowUseOfHierarchicalKeyringForSecurityPacks— 授权`kms:GenerateDataKeyWithoutPlaintext`（创建新的分支密钥）、`kms:ReEncryptTo`、`kms:ReEncryptFrom`（轮换现有分支密钥）和`kms:Decrypt`（检索现有分支密钥），用于分层密钥环操作。这些操作使用您通过同步路径转发的凭证，并根据加密上下文条件限制在安全需求包资源范围内。
- AllowAsynchronousDataAccessForSecurityPacks— 在没有可用的呼叫者凭据时 `kms:GenerateDataKeyWithoutPlaintext`、`kms:Decrypt`、`kms:ReEncryptTo`、`kms:ReEncryptFrom`向 AWS 安全代理服务主体授予、、和用于异步操作的权限。

与 Agent Spaces 密钥策略不同，安全要求包策略不包括 Web 应用程序角色主体。使用您的管理员角色通过 AWS 管理控制台访问安全需求包，而不是通过 AWS 安全客户端 Web 应用程序进行访问。所有同步操作都使用单调用者角色（通过`kms:ViaService`）。

> **Note**
>
> 如果您对代理空间和安全要求包使用相同的 KMS 密钥，则可以将这两个部分中的密钥策略声明合并为一个密钥策略。

的关键政策 CMK-encrypted AWS 资源

如果您向 AWS 安全代理提供使用客户托管密钥加密的 AWS 资源，则必须更新每个资源的 KMS 密钥上的密钥策略以授予必要的权限。这适用于以下资源：

- S3 存储桶 — 如果您提供来自使用客户托管密钥 (SSE-KMS) 加密的 S3 存储桶的学习资源，则密钥策略必须允许渗透测试服务角色解密对象。
- Secr@@ ets Manager 密钥 — 如果您的渗透测试凭据存储在使用客户托管密钥加密的 Secrets Manager 密钥中，则密钥策略必须允许渗透测试服务角色解密这些机密。如果 Web 应用程序代表您创建密钥，则密钥策略还必须允许应用程序角色对这些密钥进行加密。
- CloudWatch 日志日志组-如果用于存储渗透测试执行 CloudWatch 日志的日志组使用客户托管密钥加密，则密钥策略必须允许 CloudWatch 日志通过服务角色验证 KMS 密钥，并允许 CloudWatch 日志服务主体执行加密操作。

> **Important**
>
> AWS 安全客户端还可以代表你创建 CloudWatch 日志日志组和 Secrets Manager 密钥。配置 KMS 密钥策略时，还要为这些服务创建的资源添加相应的语句。

以下关键政策声明授予 CMK-encrypted 资源所需的权限。仅包含适用于您的配置的语句。如果资源使用您为代理空间指定的 KMS 密钥，请将这些语句添加到该密钥的策略中。如果资源使用不同的 KMS 密钥，请改为将这些语句添加到该密钥的策略中。

替换策略中的以下占位符值：

- *111122223333* — 您的 AWS 账户 ID

- *MyApplicationRole* — 安装 AWS 安全客户端期间创建的应用程序角色
- *MyPenTestServiceRole* — 渗透测试服务角色
- *us-east-1* — 您使用 AWS 安全代理的 AWS 区域

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowKmsKeyAccessForCreatingSecrets",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyApplicationRole"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
        "StringLike": {
          "kms:ViaService": "secretsmanager.*.amazonaws.com",
          "kms:EncryptionContext:SecretARN": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:*"
        }
      }
    },
    {
      "Sid": "AllowKmsKeyDecryptionForS3Objects",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyPenTestServiceRole"
      },
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
```

```
        },
        "StringLike": {
          "kms:ViaService": "s3.*.amazonaws.com",
          "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::YOUR-BUCKET-NAME*"
        }
      }
    },
    {
      "Sid": "AllowKmsKeyDecryptionForSecretsManagerSecrets",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyPenTestServiceRole"
      },
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
        "StringLike": {
          "kms:ViaService": "secretsmanager.*.amazonaws.com",
          "kms:EncryptionContext:SecretARN": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:YOUR-SECRET-NAME*"
        }
      }
    },
    {
      "Sid": "AllowKmsKeyValidationForCloudWatchLogsLogGroups",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyPenTestServiceRole"
      },
      "Action": [
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
        "StringLike": {
          "kms:ViaService": "logs.*.amazonaws.com"
```

```
        }
      }
    },
    {
      "Sid": "AllowKmsKeyAccessForCloudWatchLogsLogGroups",
      "Effect": "Allow",
      "Principal": {
        "Service": "logs.us-east-1.amazonaws.com"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
        "StringLike": {
          "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:us-
east-1:111122223333:log-group:YOUR-LOG-GROUP-NAME*"
        }
      }
    }
  ]
}
```

Note

- 只有当您提供使用客户托管密钥加密的 S3 存储桶中的学习资源时，才需要提供该`AllowKmsKeyDecryptionForS3Objects`声明。有关为 S3 SSE-KMS 进行配置的更多信息，请参阅使用保护数据 SSE-KMS。
- 只有当您的渗透测试凭证存储在使用客户托管密钥加密的 Secrets Manager 密钥中时，才需要该`AllowKmsKeyDecryptionForSecretsManagerSecrets`声明。在渗透测试期间，服务角色需要访问权限才能解密密钥。有关机密加密的更多信息，请参阅 Secrets Manager 中的密钥加密和解密。
- 该`AllowKmsKeyValidationForCloudWatchLogsLogGroups`语句授予服务角色，`kms:DescribeKey`这样 CloudWatch Logs 就可以在将 KMS 密钥与日志组关联时通过服务角色对其进行验证。

- 如果 Web 应用程序使用客户托管密钥代表您创建 Secrets Manager 密钥，则必须提供该`AllowKmsKeyAccessForCreatingSecrets`声明。应用程序角色需要`kms:GenerateDataKey`并`kms:Decrypt`使用您的 KMS 密钥对密钥进行加密。
- 该`AllowKmsKeyAccessForCloudWatchLogsLogGroups`语句向 CloudWatch 日志服务主体 (`logs.us-east-1.amazonaws.com`) 授予加密和解密日志数据所需的权限。如果您未提供现有日志组，但 AWS 安全客户端代表您创建了日志组，则也需要此语句。有关更多信息，请参阅使用 AWS KMS 加密 CloudWatch 日志中的日志数据。
- 如果多个资源共享同一 KMS 密钥，则可以将这些语句合并为一个密钥策略。

管理员角色

管理员角色（用于在控制台中管理 AWS 安全代理的 IAM 角色）不需要其他 IAM 策略。

应用程序角色

除了 KMS 密钥策略外，还要将以下 IAM 策略附加到 AWS 安全代理设置期间指定的应用程序角色。此策略授予角色使用客户托管密钥加密和解密代理空间数据以及在 Secrets Manager AWS 中创建密钥的权限。

替换策略中的以下占位符值：

- *111122223333* — 您的 AWS 账户 ID
- *us-east-1* — 您使用 AWS 安全代理的 AWS 区域
- 中的 KMS 密钥 ARN `Resource` — 您的 KMS 密钥的 ARN

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSynchronousDataAccessForAgentSpaces",
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
        "arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890cd"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333",
          "kms:ViaService": "securityagent.us-east-1.amazonaws.com"
        },
        "StringLike": {
          "kms:EncryptionContext:aws:securityagent:agent-space":
 "arn:aws:securityagent:us-east-1:111122223333:agent-space/*"
        }
      }
    },
    {
      "Sid": "AllowKmsKeyAccessForCreatingSecrets",
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890cd"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
        "StringLike": {
          "kms:ViaService": "secretsmanager.*.amazonaws.com",
          "kms:EncryptionContext:SecretARN": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:*"
        }
      }
    }
  ]
}
```

> **Note**
>
> 应用程序角色是您在设置 AWS 安全代理期间指定或控制台创建的 IAM 角色。Web 应用程序扮演此角色来检索渗透测试执行日志并代表您创建 Secrets Manager 密钥。

- 如果您为渗透测试配置身份验证资源并选择直接输入凭据而不是指定现有密钥，则必须使用该`AllowKmsKeyAccessForCreatingSecrets`语句。Web 应用程序代表您创建密钥，并且该应用程序角色需要此语句中的权限才能使用为您的 Agent Space 指定的客户托管密钥对密钥进行加密。更新此语句中的 `Resource` ARN，使其与用于代理空间的 KMS 密钥相匹配。

服务角色

在渗透测试期间， AWS 安全客户端扮演渗透测试服务角色来访问您的 AWS 资源。如果其中任何资源使用客户托管密钥加密，则必须向服务角色授予使用相应的 KMS 密钥的权限。这适用于以下资源：

- S3 存储桶 — 如果您从使用客户托管密钥加密的 S3 存储桶中提供学习资源（例如 API 文档、威胁模型或源代码），则服务角色需要权限才能解密该存储桶中的对象。有关为 S3 SSE-KMS 进行配置的更多信息，请参阅使用保护数据 SSE-KMS。
- Secr@@ ets Manager 密钥 — 如果您的渗透测试凭据存储在使用客户托管密钥加密的 Secrets Manager 密钥中，则服务角色需要解密这些密钥的权限。有关机密加密的更多信息，请参阅 Secrets Manager 中的密钥加密和解密。
- CloudWatch 日志日志组-如果用于存储渗透测试执行日志的日志组使用客户托管密钥（包括 AWS 安全代理代表您创建的日志组）进行加密，则服务角色需要`kms:DescribeKey`权限才能验证密钥。CloudWatch CloudWatch 日志服务主体负责处理日志数据的实际加密和解密，这些权限是通过密钥策略授予的（请参阅the section called “密钥策略”）。有关加密日志数据的更多信息，请参阅使用 AWS KMS 加密日志中的 CloudWatch 日志数据。

> **Important**
>
> 如果您使用不同于为代理空间指定的 KMS 密钥来加密这些资源，则必须为每个 KMS 密钥授予服务角色权限，以保护该服务角色在渗透测试期间访问的资源。

将以下 IAM 策略附加到渗透测试服务角色。仅包含适用于您的配置的语句。

替换策略中的以下占位符值：

- *111122223333* — 您的 AWS 账户 ID
- *us-east-1* — 您使用 AWS 安全代理的 AWS 区域

- 中的 KMS 密钥 ARN `Resource` — 用于加密每个资源的客户托管密钥的 ARN

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowKmsAccessForEncryptedS3Buckets",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE-S3-KEY-ID"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
        "StringLike": {
          "kms:ViaService": "s3.*.amazonaws.com",
          "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::YOUR-BUCKET-NAME*"
        }
      }
    },
    {
      "Sid": "AllowKmsAccessForEncryptedSecrets",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE-SECRETS-KEY-ID"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
        "StringLike": {
          "kms:ViaService": "secretsmanager.*.amazonaws.com",
          "kms:EncryptionContext:SecretARN": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:YOUR-SECRET-NAME*"
        }
      }
```

```
    },
    {
      "Sid": "AllowKmsKeyValidationForCloudWatchLogsLogGroups",
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE-CLOUDWATCH-LOGS-KEY-ID"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
        "StringLike": {
          "kms:ViaService": "logs.*.amazonaws.com"
        }
      }
    }
  ]
}
```

Note

- 只有当您提供使用客户托管密钥加密的 S3 存储桶中的学习资源时，才需要提供该`AllowKmsAccessForEncryptedS3Buckets`声明。更新 `Resource` ARN 以匹配用于加密您的 S3 存储桶的 KMS 密钥，然后更新该`kms:EncryptionContext:aws:s3:arn`值以匹配您的存储桶名称。
- 只有当您的渗透测试凭证存储在使用客户托管密钥加密的 Secrets Manager 密钥中时，才需要该`AllowKmsAccessForEncryptedSecrets`声明。更新 `Resource` ARN 以匹配用于加密您的密钥的 KMS 密钥，并更新该`kms:EncryptionContext:SecretARN`值以匹配您的密钥名称。
- 仅当您的 CloudWatch 日志日志组使用客户托管密钥（包括 AWS 安全客户端代表您创建的日志组）加密时，才需要该`AllowKmsKeyValidationForCloudWatchLogsLogGroups`声明。更新 `Resource` ARN 以匹配用于加密您的日志组的 KMS 密钥。
- 如果多个资源共享同一 KMS 密钥，则可以合并语句并在字段中列出共享密钥 ARN 一次。`Resource`

使用客户托管密钥创建资源

在设置 AWS 安全代理或创建单个资源时，您可以指定客户管理的密钥。KMS 密钥对属于该资源及其子资源的所有数据进行加密。

在安装过程中设置默认 KMS 密钥（控制台）

您可以在初始 AWS 安全客户端设置期间配置默认 KMS 密钥。除非您指定其他密钥，否则此密钥将用作新 Agent Spaces 和集成的默认密钥。

1. 在设置 AWS 安全代理页面上，展开加密-可选部分。
2. 选择 “自定义加密设置（高级）”。
3. 对于选择 AWS KMS 密钥，请从您的账户中选择 KMS 密钥，或输入 KMS 密钥 ARN。
4. 完成其余的设置步骤，然后选择设置 AWS 安全代理。

AWS 安全代理会验证 KMS 密钥以确认其存在、是否已启用，并使用对称加密。如果验证失败，安装程序将无法继续进行，并且您会收到一条错误消息。

使用客户托管密钥创建座席空间（控制台）

1. 在 AWS 安全客户端控制台中，导航到 “代理空间” 页面。
2. 选择创建代理空间。
3. 输入代理空间的名称和可选描述。
4. 展开 Advanced（高级）部分。
5. 在 “数据加密” 下，选择以下选项之一：
   - 使用应用程序默认密钥-使用在安装 AWS 安全客户端期间配置的默认 KMS 密钥。如果配置了默认密钥，则默认情况下会选择此选项。
   - 使用其他密钥-为此代理空间指定不同的 KMS 密钥。选择 “自定义加密设置（高级）”，然后在 “选择 AWS KMS 密钥” 中，从您的账户中选择 KMS 密钥或输入 ARN。
6. 选择创建。

使用客户托管密钥（控制台）创建集成

1. 在 AWS 安全客户端控制台中，导航到 “集成” 页面。
2. 选择 “添加集成”，然后选择您的提供商（例如 GitHub）。
3. 完成步骤 1：安装并授权提供商应用程序。

4. 在步骤 2：注册详细信息中，输入注册名称并配置提供商设置。
5. 在 “数据加密” 部分中，选择 “自定义加密设置（高级）”。
6. 对于选择 AWS KMS 密钥，请从您的账户中选择 KMS 密钥，或输入 KMS 密钥 ARN。
7. 选择连接。

使用客户托管密钥创建资源 (AWS (CLI 或 SDK)

要使用 AWS CLI 或 SDK 指定客户托管密钥，请执行以下操作：

- 在调用`CreateApplication`为应用程序设置默认 KMS 密钥时，请包含`defaultKmsKeyId`参数。在没有显式 KMS 密钥的情况下创建代理空间或集成时，此密钥用作备用密钥。
- 在调用`CreateAgentSpace`或加密`CreateIntegration`特定资源时包括`kmsKeyId`参数。这优先于应用程序级别的默认值。

有关参数的详细信息，请参阅《AWS 安全客户端 API 参考》。

如果您未指定 KMS 密钥，则 AWS 安全客户端将使用应用程序级默认 KMS 密钥（如果已配置）。否则，将使用 AWS托管密钥对资源进行加密。

密钥轮换

AWS 安全客户端会自动定期轮换分支密钥。分支密钥轮换会创建分支密钥的新活动版本，同时保留所有以前的版本。旋转后：

- 新数据使用新的分支密钥版本进行加密。
- 以前加密的数据仍可使用较旧的分支密钥版本进行解密。
- 无需对数据进行重新加密。

分支密钥轮换与 KMS 密钥轮换是分开的。您可以单独为客户托管密钥启用自动 KMS 密钥轮换。有关更多信息，请参阅《密钥管理服务开发人员指南》中的轮换 KMS AWS 密钥。

分支密钥轮换后，在短时间内，先前分支密钥的缓存副本仍可用于新的加密操作。此窗口由内部缓存存存活时间 (TTL) 决定，不会影响数据的安全性。

注意事项

在 AWS 安全客户端中使用客户托管密钥时，请记住以下几点：

- 客户托管密钥仅适用于新资源。在配置客户托管密钥之前创建的现有资源将继续使用 AWS托管加密。现有数据没有迁移。
- 客户管理的密钥是在创建时指定的。您在创建资源时指定 KMS 密钥。您无法为现有资源添加或更改 KMS 密钥。
- 支持多个 KMS 密钥。您可以为不同的资源使用不同的 KMS 密钥。例如，您可以将一个密钥用于生产代理空间，将另一个密钥用于开发代理空间。
- 禁用或删除 KMS 密钥会使数据无法访问。如果您禁用或计划删除 KMS 密钥，则 AWS 安全客户端无法解密使用该密钥加密的数据。在重新启用密钥之前，受影响的资源将无法访问。删除 KMS 密钥会永久阻止访问使用该密钥加密的所有数据。
- 需要密钥策略权限。如果您从 KMS 密钥策略中删除所需权限，则 AWS 安全客户端将无法访问加密数据。确保密钥策略授予管理员角色（用于管理 AWS 控制台中的服务）、应用程序角色（由 Web 应用程序使用）、渗透测试服务角色（由代理担任运行渗透测试）和 AWS 安全客户端服务主体的权限，如中所述the section called “所需的 KMS 权限”。
- AWS KMS 配额适用。针对您的 KMS 密钥进行的加密操作计入您的 AWS KMS 请求配额。在正常使用下，分层密钥环架构通过分支密钥缓存最大限度地减少了 KMS 调用。有关更多信息，请参阅《AWS 密钥管理服务开发人员指南》中的申请配额。

问题排查

查找使用 AWS 安全代理时常见错误的解决方案。

访问被拒绝：选择的 GitHub 账户类型不正确或指定的组织名称不正确

- 您将 AWS Security Agent 应用程序安装到所需的 GitHub 组织中，`GitHub Account Type`但错误地`User`将其设置为 `Organization`
- 您将 AWS Security Agent 应用程序安装到所需的 GitHub 组织中，并正确设置为，`Organization`但将该`Organization Name`字段留空，或者输入的组织名称与您安装该应用程序的组织名称不符 `GitHub Account Type`

解决方案：

1. 前往组织 GitHub 并从组织中卸载该应用程序。有关更多信息，请参阅 the section called “步骤 1：从中卸载 AWS 安全代理 GitHub 应用程序 GitHub”。
2. 返回集成页面，通过单击，重新启动集成过程，然后再次将应用程序安装到所需的组织中`Add Integrations`，并将其授权到所需的 GitHub 组织中。

3. Organization从下拉列表中选择 GitHub account type
4. 确保Organization Name您输入的内容与您安装应用程序的输入完全相同。
5. 选择 Connect 以创建您的 GitHub 组织集成。

访问被拒绝：权限不足，无法将 GitHub 应用程序安装到组织中

当您尝试将 AWS Security Agent 应用程序安装到所需的 GitHub 组织中时，您将在安装页面的按钮上看到两条不同的消息。

- 组织Member将看到 Authorize & Request
- 组织Owner将看到 Install & Authorize

您可以按照以下步骤验证自己是Member GitHub 组织Owner成员还是该组织的成员。

1. 前往 g ithub.com
2. 在网站上选择您的个人资料
3. 导航到下拉菜单Organizations上并选择它
4. 从组织列表中找到您想要安装 AWS Security Agent 的组织，它会指定您是Member组织名称还是组织名称Owner旁边。

可能的解决方案：

- 在尝试创建集成之前，请所有者批准您的安装请求
- 让所有者将你在 GitHub 组织中的角色从 a 更新Member为 a，Owner然后重新启动集成过程

在渗透测试期间，代理无法连接到端点

如果渗透测试代理无法调用已配置的目标 URL 或未能成功导航目标端点：

- 如果您的终端节点调用配置的目标 URL 之外的域名，请确认在 pentest 配置中将其他域名添加为可访问网址
- 渗透测试目前仅适用于在 HTTP/HTTPS 端口 80 或 443 上提供流量的端点
- 如果您配置了 WAF，请检查您的 WAF 是否未阻止渗透测试流量。您可以按User-Agent标头将渗透测试流量列入许可名单，标题默认设置为 securityagent

获得更多帮助

如果您在尝试这些故障排除步骤后仍然遇到问题：

- 查看 AWS 安全代理服务状态页面
- 如需有关复杂配置问题的帮助，请联系 AWS Support

适用于用户和开发人员（Web 应用程序和 IDE）

在 Web 应用程序、IDE 或关联源提供商中运行设计审查、威胁模型、代码审查和渗透测试，然后查看和修复发现的结果。

登录 AWS 安全代理 Web 应用程序

您应登录 AWS 安全代理 Web 应用程序以完成以下任务：

- 进行设计审查
- 进行渗透测试

访问方法

AWS Security Agent 提供了不同的方法来访问 Web 应用程序，具体取决于您的组织在初始设置期间如何配置访问权限以及您是否具有 AWS 控制台访问权限。

IAM 身份中心 (SSO)-如果您的组织启用了 IAM 身份中心，则可以使用您的 SSO 凭证登录。在访问 Web 应用程序之前，必须将您分配到 IAM Identity Center 中的至少一个代理空间。

管理员访问权限 (IAM)-如果您拥有相应权限的 AWS 控制台访问权限，则可以通过任何 Agent Space 页面上的管理员访问链接启动 Web 应用程序。此方法通过您现有的控制台会话提供身份验证。

> **Note**
>
> 您的组织使用的主要访问方法是在最初的 AWS 安全代理设置过程中确定的。有关配置用户访问权限和分配的信息，请参阅the section called “授予用户访问 AWS 安全代理 Web 应用程序的权限”。

使用 IAM 身份中心 (SSO) 登录

如果您的组织配置了 IAM Identity Center 访问权限，则您可以使用 SSO 凭证通过多个入口点登录 Web 应用程序。

先决条件

- 您已被分配到 IAM 身份中心中的至少一个代理空间

- 您拥有 IAM 身份中心 SSO 证书

访问 Web 应用程序

根据需要选择以下方法之一：

要查看您分配到的所有代理空间，请执行以下操作：

1. 在 AWS 安全代理控制台中，导航到 “设置”。
2. 找到 Web 应用程序域并复制 URL。

> **Tip**
>
> 将此 URL 加入书签以便于访问。这是查看您分配到的所有代理空间的通用入口点。

3. 导航到 Web 应用程序 URL。
4. 出现提示时，输入您的 IAM 身份中心证书。
5. 身份验证后，您将看到分配给您的所有代理空间的列表。
6. 选择您要使用的代理空间。

要直接访问特定的代理空间（需要访问 AWS 控制台），请执行以下操作：

1. 登录 AWS 管理控制台。
2. 导航到 AWS 安全代理控制台。
3. 导航到您要访问的代理空间。
4. 选择下列选项之一：
   - “代理空间” 概述页面上的 “启动 Web 应用程序” 按钮
   - 在 “代码审查” 部分中@@ 启动 Web 应用程序按钮
   - 在 “渗透测试” 部分@@ 启动 Web 应用程序按钮
5. 出现提示时，输入您的 IAM 身份中心证书。
6. 身份验证后，您将直接进入 Web 应用程序中的代理空间。

> **Note**
>
> 如果您没有 AWS 控制台访问权限，请使用 “设置” 页面上的 Web 应用程序域访问所有分配的代理空间。

使用管理员访问权限 (IAM) 登录

如果您拥有 AWS 控制台访问权限并拥有相应的 AWS 安全代理权限，则可以通过管理员访问链接启动带有自动身份验证的 Web 应用程序。

先决条件

- 您已登录到 AWS 管理控制台
- 您拥有访问 AWS 安全代理资源的适当权限

访问 Web 应用程序

选择以下方法之一：

要访问特定的代理空间，请执行以下操作：

1. 登录 AWS 管理控制台。
2. 导航到 AWS 安全代理控制台。
3. 导航到您要访问的代理空间。
4. 在 “座席空间” 概述页面上选择 “管理员访问权限” 按钮。
5. Web 应用程序将在新选项卡中打开，其中包含该代理空间的自动身份验证。

> **Note**
>
> 管理员访问按钮始终可供拥有 AWS 控制台访问权限的用户使用，无论您的组织是否使用 IAM Identity Center 作为主要访问方式。

创建设计评审

上传文件供 AWS 安全代理审查，根据组织安全要求评估您的设计文档。设计审查有助于在开发生命周期的早期发现安全问题，使您能够在最具成本效益的时候解决架构问题。

AWS Security Agent 会根据贵组织的安全要求分析您的设计文档，在开始实施之前提供详细的安全调查结果以改善安全状况。

在此过程中，您将通过上传设计文件进行分析来创建设计评审。

先决条件

在开始之前，请确保您满足以下条件：

- 访问 AWS 安全代理 Web 应用程序
- 可供上传的设计文档（DOC、DOCX、JPEG、MD、PDF、PNG 和 TXT）
- 每个文件必须不超过 2MB，所有文件的总大小不得超过 6MB
- 了解您的组织启用了哪些安全要求

第 1 步：开始创建设计评审

在 Agent Web 应用程序中导航到设计评审创建页面。

1. 登录 AWS 安全代理 Web 应用程序。
2. 导航到设计评论部分。
3. 选择 “创建设计评审”。

> **Tip**
>
> 您可以在 AWS 安全代理控制台中导航至 “安全要求” 页面，查看贵组织已启用的安全要求。选择任何已启用的要求以查看其详细信息。这些要求用于分析您的设计文件。

第 2 步：命名您的设计评论

提供描述性名称，以帮助确定本次设计审查的目的和范围。

1. 在 “设计审查名称” 部分中，找到 “名称” 字段。
2. 为您的设计评审输入一个描述性名称。

> **Note**
>
> 名称应清楚地标识正在审核的项目、功能或组件。最多 80 个字符。

第 3 步：上传设计文件

上传您希望 AWS 安全代理分析的安全合规性设计文档。

1. 在 “要查看的文件” 部分，查看文件要求：

> **Important**
>
> 每次设计评审最多可以上传 5 个文件。每个文件必须不超过 2MB，所有文件的总大小不得超过 6MB。支持的格式：DOC、DOCX、JPEG、MD、PDF、PNG 和 TXT。

2. 使用以下方法之一上传您的文件：
   a. 拖放 — 将文件直接拖到文件放置区中
   b. 浏览-使用 “选择文件” 按钮浏览并选择计算机中的文件
3. 确认所有必需的文件都已上传。

> **Tip**
>
> 为了获得最佳结果，请附上架构图、设计规格和技术文档，这些文档描述了系统的安全相关组件和数据流。

第 4 步：启动设计审查

配置完所有必需的信息后，开始对设计文档进行安全分析。

1. 查看所有上传的文件和设置以确保准确性。
2. 选择 “开始设计审查”。
3. AWS 安全代理根据已启用的安全要求分析您的设计文档。

> **Note**
>
> 设计审查过程通常在几分钟内完成，具体取决于上传文件的数量和大小。您会收到基于组织安全要求的安全调查结果。

后续步骤

开始设计审查后：

- 在 Agent Web 应用程序中监控审阅进度
- 查看安全调查发现
- 与您的开发团队分享发现
- 解决设计中已确定的安全问题
- 更新设计文档并在需要时重新提交

查看设计审查的结果

审查结果可帮助您了解哪些安全要求得到满足，哪些需要注意，以及在开始实施之前应采取哪些措施来改善设计的安全状况。

在此过程中，您将学习如何访问、筛选和解释设计审查结果，以有效解决安全问题。

先决条件

在开始之前，请确保您满足以下条件：

- 已完成的设计审查
- 访问 AWS 安全代理 Web 应用程序
- 熟悉贵组织已启用的安全要求

第 1 步：访问设计评审

导航到您的设计评审以查看结果和摘要信息。

1. 登录 AWS 安全代理 Web 应用程序。

2. 导航到设计评论部分。
3. 从列表中选择要检查的设计评审。

> **Tip**
>
> 设计审查详细信息页面显示审阅状态、完成日期和已审阅文件数量的摘要。

步骤 2：查看调查结果摘要

查看高级摘要，了解您的设计的总体安全状况。

1. 找到 “摘要” 部分。
2. 查看每个合规状态类别的计数：“合规” Non-compliant、“数据不足” 和 “不适用”。

> **Note**
>
> 摘要提供了每种状态类型的计数，可帮助您快速评估需要关注的发现数量。有关每种状态的详细说明，请参阅步骤 4。

步骤 3：筛选和浏览调查结果

使用筛选和搜索功能将重点放在特定的发现结果或合规性状态上。

1. 在 “查看结果” 部分，找到筛选控件。
2. 要按状态筛选，请执行以下操作：
   a. 选择状态下拉菜单。
   b. 选择特定的合规性状态以仅查看具有该状态的调查结果。
3. 要搜索特定的安全要求，请执行以下操作：
   a. 在搜索栏中输入关键字。
   b. 当您键入时，结果会自动更新。
4. 使用分页控件浏览多页搜索结果。

> **Tip**
>
> 首先按Non-compliant状态筛选，确定设计中需要立即关注的发现的优先顺序。

步骤 4：了解合规状态

每项发现都显示合规性状态，表明您的设计如何满足特定的安全要求：

- 合规-根据分析，您的设计符合安全要求
- Non-compliant— 您的设计违反或未充分满足安全要求
- 数据不足-上传的文件缺少足够的信息来确定合规性
- 不适用-安全要求不适用于您的系统设计

> **Important**
>
> 重点关注Non-compliant和数据不足状态，因为这些状态需要采取措施。通过更新您的设计来解决不合规的发现问题，并通过上传其他设计文档来解决数据不足的问题。

第 5 步：查看查找详情

选择个别调查结果以查看详细的理由和补救指南。

1. 在调查结果表中，选择安全要求名称。
2. 查看调查结果的详细信息，其中包括：
   - 正在评估的具体安全要求
   - 解释调查结果为何获得合规状态的评论，包括有关缺失或不合规内容的具体细节
   - 为解决这一发现而推荐的补救指南
   - 指向贵组织内部文件或安全要求标准的链接

> **Note**
>
> 该评论通过具体细节解释了 AWS 安全代理的推理。对于数据发现不足，该评论会指出缺少哪些信息，例如 “设计文档未提及身份验证机制” 或 “未找到有关静态数据加密的信息”。

步骤 6：解决调查结果

对需要注意的发现采取行动，以改善设计的安全状况。

如需了解Non-compliant调查结果：

1. 查看建议的补救指南。
2. 查看所有链接的内部文档，了解更多背景信息。
3. 更新您的设计文档以满足安全要求。
4. 记录您所做的更改，以备将来参考。

如果发现的数据不足，请执行以下操作：

1. 仔细阅读评论以了解缺少哪些具体信息。
2. 使用缺失的细节创建或更新设计文档。
3. 准备更新后的文件以供重新提交。

后续步骤

查看您的设计结果后：

- 将调查结果报告下载为 CSV 文件，以便与您的团队共享
- 更新设计文档以解决不合规发现的问题
- 为发现的数据不足创建其他文档
- 与您的开发团队分享发现以供讨论
- 克隆此设计评审以创建预先加载原始文档的新审阅，允许您更新名称并再次运行分析以验证改进
- 继续实施符合合规要求的设计

有关管理安全要求的更多信息，请参阅the section called “管理安全要求”。

有关创建设计评论的更多信息，请参阅the section called “创建设计评审”。

创建威胁模型

在 AWS 安全代理 Web 应用程序中创建威胁模型，以分析您的应用程序架构并识别安全威胁。威胁模型产生两个主要输出：描述系统构建方式和行为方式的系统概述，以及一组描述系统如何受到攻击的威胁，每种威胁都有严重性级别、STRIDE 分类和可行的建议。

威胁模型接受两种输入，您可以提供其中一种或两者兼而有之：

- 范围文档 — 定义威胁模型重点内容的技术设计文档、API 规范或架构文档。如果提供，则代理将其分析范围限定在这些文档中描述的上下文中。您可以上传文件（DOC、DOCX、JPEG、MD、PDF、PNG、TXT）、提供 S3 URI 或通过集成连接 Confluence 页面。
- 来源-来自连接的存储库（GitHub、 GitHub 企业服务器、 GitLab GitLab Self-Managed、或 Bitbucket）或 S3 位置的源代码。AWS 安全代理读取源代码以了解您的现有系统。当与作用域文档结合使用时，源代码提供了有关当前实现的上下文。

在此过程中，您可以通过配置威胁模型的输入和权限来创建威胁模型，然后开始运行。

先决条件

在开始之前，请确保您满足以下条件：

- 访问 AWS 安全代理 Web 应用程序
- 至少以下配置之一：
  - 用作源的已连接存储库（GitHub GitHub 企业服务器、 GitLab GitLab Self-Managed、或 Bitbucket）（请参阅the section called “启用威胁建模”）
  - 包含源代码的 S3 存储桶
  - 设计要作为范围文档或 Confluence 集成上传的文档

> **Tip**
>
> 如果您已经将存储库或 S3 存储桶连接到 Agent Space（例如，通过代码审查或渗透测试设置），则可以立即创建威胁模型，无需进行其他设置。

AWS 安全代理如何运行威胁模型

您提供的输入决定了 AWS 安全代理如何运行威胁模型。有三种情况。

您提供的输入	AWS 安全代理如何运行威胁模型
仅限源代码（源代码）	AWS Security Agent 会分析源代码以了解您的应用程序结构，对组件进行分类，提取数据流，构建威胁模型，并根据系统的实际实施方式识别威胁。
仅限范围文档（设计文档）	AWS Security Agent 运行的威胁模型侧重于您的文档中描述的设计。它根据范围文档中描述的架构、数据流和组件来识别威胁，这对于在任何代码存在之前对设计进行威胁建模非常有用。
源代码和范围文档	AWS Security Agent 将威胁模型的范围限定为范围文档（例如，新功能的设计文档）中描述的上下文。它使用源代码来了解您的现有系统，然后对范围文档中描述的设计进行威胁建模，无论该设计是否已经实现。

访问威胁模型页面

导航到 Web 应用程序中的威胁建模部分。

1. 登录 AWS 安全代理 Web 应用程序。
2. 在左侧边栏中，选择威胁模型。
3. 您会看到现有威胁模型的列表，包括其标题、最新运行状态和按严重程度划分的威胁计数。

创建威胁模型

通过配置其输入和权限来设置新的威胁模型。

1. 在威胁模型页面上，选择创建威胁模型。

配置威胁模型详细信息

为您的威胁模型提供标题。

1. 在标题字段中，输入威胁模型的描述性名称。

> **Tip**
>
> 使用标识正在建模的应用程序或服务的名称。例如，“计费服务” 或 “结账API”。

添加范围文档

提供定义威胁模型重点内容的技术设计文档，例如架构文档、API 规范或设计提案。提供范围文档时，代理将其分析范围限定在这些文档中描述的上下文中。如果您提供来源，则范围文档是可选的。

1. 在范围文档部分，使用以下一种或多种方法添加文档：
   - 上传文件 — 直接上传设计文档（DOC、DOCX、JPEG、MD、PDF、PNG 或 TXT）。
   - S3 URI — 输入指向存储在连接的 S3 存储桶中的文档的 S3 URI。
   - Confluence 页面 — 如果您将 Confluence 集成连接到代理空间，请从 Confluence 工作区中选择页面。

> **Tip**
>
> 为了获得最佳结果，请提供架构图、API 规格和技术文档，以描述您的系统组件、数据流和信任边界。

添加来源

选择 AWS 安全代理用于了解您的现有系统的源代码。当与作用域文档结合使用时，源代码提供了有关当前实现的上下文——代理使用它来了解已经存在的内容。如果您提供范围文档，则来源是可选的。

1. 在源代码部分，使用以下一种或多种方法添加源代码：

集成存储库

从连接到代理空间（ GitHub 企业服务器GitHub、、或 Bitbucket）的存储库中进行选择。 GitLab GitLab Self-Managed

1. 选中要作为来源包含的每个存储库旁边的复选框。
2. 使用搜索字段按名称查找特定的存储库。

> **Note**
>
> 此处仅显示与您的代理空间相连的存储库。要添加更多存储库，请要求您的管理员更新代理空间配置。

S3 来源

添加指向存储在连接的 S3 存储桶中的源代码的 S3 URI。

1. 输入要包含的每个源代码位置的 S3 URI。

配置 AWS 资源

为此威胁模型选择 IAM 服务角色和可选 CloudWatch 日志组。

1. 在服务角色下拉列表中，从您配置的服务角色中选择 IAM 角色。

> **Note**
>
> 服务角色必须有权访问 S3 中的源代码并写入 CloudWatch 日志。服务角色是在 AWS 管理控制台中设置代理空间期间配置的。

2. （可选）在日志组下拉列表中， CloudWatch 选择用于存储威胁模型执行日志的日志组。

创建威胁模型

1. 审核 配置。
2. 选择创建威胁模型。

您将被重定向到威胁模型详细信息页面，您可以在其中开始运行。

运行威胁模型

创建威胁模型后，开始运行以开始分析。

1. 在威胁模型详细信息页面上，选择开始运行。
2. AWS 安全代理开始分析您的来源和范围文档。

您也可以从威胁模型列表页面开始运行，方法是选择要运行的威胁模型旁边的开始运行。

监控威胁模型的运行情况

跟踪威胁模型的执行进度。

运行状态

运行页标题显示一个状态指示器：

- 进行中-威胁模型代理正在运行。
- 正在停止 — 已请求取消；代理在停止之前正在完成其当前任务。
- 已完成-运行成功完成。
- 失败-运行遇到错误。将显示一条包含详细信息的错误消息。解决问题后重新开始运行。
- 已停止 — 用户取消了运行。停止之前产生的所有威胁都将被保留。

运行页面选项卡

在运行详情页面上，在选项卡之间导航：

- 概述-查看运行摘要（作业 ID、开始时间、状态、持续时间），其严重性级别饼图显示了按严重性（严重、高、中、低）细分的威胁。在摘要下方，查看STRIDE威胁类别条形图，该条形图显示了每个STRIDE类别中有多少威胁。运行完成后，查看代理生成的系统概述。
- 配置-查看用于此次运行的来源、文档和权限（服务角色、 CloudWatch 日志组）。
- Prefli ght — 查看威胁分析开始之前运行的预检检查的状态，包括日志基础架构设置、S3 源访问验证（如果适用）和测试环境设置。进度条显示已完成的检查次数。
- 日志-查看代理在运行期间执行的任务的可筛选列表。选择任意任务可在侧面板中查看其详细日志输出。
- 威胁-查看运行期间发现的威胁（请参阅the section called “查看来自威胁模型的威胁”）。

运行历史记录

每个威胁模型都保留所有运行的历史记录。在威胁模型详情页面上：

- 运行表列出了所有运行及其开始时间、状态、持续时间、威胁计数和 ID。
- 选择任何跑步的开始时间链接以查看其完整详细信息。

> **Tip**
>
> Re-run 在更新源代码或修改范围文档以查看系统概述和威胁如何随时间变化之后，使用相同的威胁模型。

编辑威胁模型

要修改威胁模型配置，请执行以下操作：

1. 在威胁模型详细信息页面上，选择编辑。
2. 根据需要更新标题、来源、范围文档或 AWS 资源。
3. 选择保存更改。

> **Note**
>
> 编辑威胁模型不会影响之前完成的运行。它们会保留运行时使用的配置快照。

删除威胁模型

要删除威胁模型及其所有相关的运行和威胁，请执行以下操作：

1. 在威胁模型详细信息页面上，打开操作菜单并选择删除。
2. 确认删除操作。

> **Important**
>
> 如果当前正在运行，则必须先将其停止，然后再删除威胁模型。

后续步骤

运行威胁模型后：

- 查看系统概述和威胁（请参阅the section called “查看来自威胁模型的威胁”）
- Re-run 为验证威胁是否得到解决而进行更改后的威胁模型

- 随着应用程序的发展，调整来源和范围文档

查看来自威胁模型的威胁

威胁模型运行完成后，请查看系统概述和威胁，以了解您的应用程序可能如何受到攻击以及如何应对。系统概述是一份全面的文档，描述了您的应用程序的架构、信任边界、数据流和安全状况。每项威胁都包括声明、严重级别、STRIDE 分类、受影响的资产以及应对建议。

先决条件

在开始之前，请确保您满足以下条件：

- 已完成的威胁模型运行
- 访问 AWS 安全代理 Web 应用程序

步骤 1：访问威胁模型运行

导航到已完成的威胁模型运行。

1. 登录 AWS 安全代理 Web 应用程序。
2. 在左侧边栏中，选择威胁模型。
3. 选择要检查的威胁模型。
4. 在运行表中，通过选择其开始时间链接来选择已完成的运行。

步骤 2：查看系统概述

系统概述全面描述了 AWS 安全代理如何理解您的系统。它是一个结构化文档，可以包括：

- 目的 — 系统做什么以及为谁服务。
- 功能-系统提供的关键功能。
- 设计意图 — 设计变更或正在进行威胁建模的功能（当提供范围文档时）。
- 架构-系统的构建方式，包括部署模式和通信协议。
- 组件-系统组件及其用途和关键交互作用表。
- 信任边界 — 安全环境发生变化的地方，包括每个过境点都有哪些保护措施。
- 数据流 — 详细描述数据如何在系统中移动，包括每个步骤的协议、凭据和保护。
- 安全状况-当前的身份验证、加密和访问控制机制。

- 敏感资产 — 需要保护的数据和凭证及其分类和暴露点。
- 关键假 Security-relevant 设 — 代理对系统的假设。

要查看系统概述，请执行以下操作：

1. 选择 “概览” 选项卡。
2. 查看运行摘要部分，其中显示了作业 ID、开始时间、状态和持续时间。
3. 查看严重级别图表，该图表显示了按严重性（严重、高、中、低）分列的威胁以及计数和百分比。
4. 查看威胁类别图表，该图表显示了每个 STRIDE 类别中有多少威胁（欺骗、篡改、拒绝、信息泄露、拒绝服务、权限提升）。
5. 在 “系统概述” 部分，阅读代理的完整分析。

> **Tip**
>
> 如果系统概述无法准确反映您的系统，请完善您的输入——添加相关存储库作为来源或上传更完整的范围文档——然后再次运行威胁模型。

步骤 3：查看威胁

导航到 “威胁” 选项卡，查看运行期间发现的所有威胁。

1. 选择 “威胁” 选项卡。
2. 威胁显示为列表，每张卡片都显示威胁标题、严重性标记、状态和上次更新的时间戳。您可以按严重性、状态或按标题搜索来筛选威胁。
3. 从列表中选择一个威胁，在右侧面板中查看其全部详细信息。

威胁严重性

每种威胁都被分配了一个严重级别：

- 严重-需要立即采取行动；利用漏洞可能导致整个系统受损。
- 高-需要立即注意；利用漏洞可能会对安全造成重大影响。
- 中度 — 应在合理的时间范围内解决；会导致整体安全风险。
- 低 — 可以作为定期维护的一部分来解决；将即时风险降至最低。

威胁详情

选择威胁以在右侧面板中查看其详细信息。详细信息分为以下几个部分：

元数据行：

- 严重性-代理分配的风险级别（严重、高、中或低）。
- STRIDE 类别 — 威胁分类：欺骗、篡改、拒绝、信息泄露、拒绝服务或权限提升。
- 创建于-发现威胁的时间。
- 上次更新时间-上次修改威胁的时间。

描述部分：

- 声明 — 对威胁的自然语言描述：威胁源可以做什么、影响是什么，以及哪些条件使之成为可能。
- 来源-威胁的行为者或来源（例如，“经过身份验证的用户” 或 “外部攻击者”）。
- 操作-威胁源可以执行的操作（例如，“将 SQL 查询注入搜索参数”）。
- 影响-威胁操作的直接后果（例如，“未经授权访问客户记录”）。

参考文献部分：

- 受影响的资产 — 受威胁影响的特定资产（例如，“客户付款记录” 或 “DynamoDB 表”）。
- 先决条件 — 必须符合条件才能利用威胁。
- 受影响的目标-受影响的安全目标：机密性、完整性、可用性、授权、身份验证或 Non-repudiation。
- 证据 — 支持威胁的源代码文件路径，链接回存储库中的特定文件。
- Anchor — 将威胁与源代码中的特定节点关联起来的代码参考。

推荐部分：

- 建议 — 应对威胁的可行指南。

步骤 4：手动创建威胁

您可以添加代理未识别的威胁，例如，在手动审查期间发现的威胁或来自外部来源的威胁。

1. 在已完成运行的 “威胁” 选项卡上，选择 “创建威胁”。

2. 填写威胁详情：
   - 声明 — 威胁的自然语言描述。
   - 严重性-选择 “严重”、“高”、“中” 或 “低”。
   - 来源-威胁的实施者或来源。
   - 先决条件-威胁可被利用所需的条件。
   - 操作-威胁源可以做什么。
   - 影响-威胁操作的直接后果。
   - 受影响的资产-受影响的特定资产（以逗号分隔）。
   - 受影响的安全目标-从机密性、完整性、可用性、授权、身份验证或中进行选择 Non-repudiation。
   - STRIDE 类别 — 选择适用的类别。
   - 建议-应对威胁的指南。
3. 选择创建。

手动创建的威胁与代理生成的威胁一起出现在威胁列表中。

步骤 5：编辑和分类威胁

查看威胁时，您可以编辑威胁的详细信息并更新其状态以跟踪进度。

1. 从列表中选择一个威胁。
2. 在威胁详细信息面板中选择编辑图标以打开编辑表单。
3. 您可以修改以下字段：
   - 状态-跟踪威胁生命周期：
     - 打开-威胁已确认并需要注意（默认）。
     - 已解决-您已解决问题。
     - 已@@ 驳回-您审查了威胁并确定其不适用。
   - 严重性-如果代理的评估与您的背景不符，请调整严重性级别。
   - 声明-完善威胁描述。
   - 来源、先决条件、操作、影响-根据您的领域知识更新威胁详细信息。
   - 受影响的资产-添加或移除受影响的资产（以逗号分隔）。
   - 受影响的安全目标-选择受影响的安全目标（机密性、完整性、可用性、授权、身份验证等 Non-repudiation）。
4. 选择保存以应用您的更改。

第 6 步：下载报告

运行完成后，您可以下载一份总结系统概述和所有已识别威胁的 PDF 报告。

1. 在 “已完成的运行” 页面上，选择 “生成报告”。
2. PDF 会下载到您的计算机上。

步骤 7：查看印前检查和日志

如果您需要调查代理如何得出结论或调试部分故障，请执行以下操作：

1. 选择 “印前检查” 选项卡，查看威胁分析开始之前运行的预检检查的状态。每项检查都显示其状态（完成、进行中或待处理），进度条显示已完成的检查数量。您可以展开已完成的支票以查看其详细日志输出。
2. 选择 “日志” 选项卡可查看代理在运行期间执行的任务的可筛选列表。从列表中选择任意任务，即可在侧面板中查看其详细日志输出。

后续步骤

查看您的威胁模型结果后：

- 首先根据代理的建议解决高严重性威胁
- 在实施修复时更新威胁状态
- 运行新的威胁模型以验证您的更改是否能解决已识别的威胁
- 随着应用程序的发展，调整来源和范围文档（请参阅the section called “创建威胁模型”）

查看拉取请求中的代码安全发现

为您的存储库启用拉取请求的代码审查后，AWS Security Agent 会自动分析拉取请求并将安全发现直接发布到您的源代码控制提供商中。这使开发人员无需离开拉取请求即可在正常工作流程中解决安全问题。

> **Note**
>
> 此页面适用于 GitHub 拉取请求、 GitLab 合并请求和 Bitbucket 拉取请求。所有提供商的体验都差不多。

代码审查在拉取请求中的工作原理

当您在启用了代码审查的存储库中提交拉取请求（或合并请求 GitLab）时，AWS 安全代理会自动开始分析。

1. 拉取请求分析触发器-在您启用了代码审查功能的存储库中，拉取请求被标记为 “准备审阅” 时，就会触发代码审查。不分析拉取请求草稿。
2. 分析确认-当 AWS Security Agent 开始分析您的拉取请求时，它会发布一条初步评论：“AWS 安全代理正在分析您的代码...” 这可以让您知道分析已经开始并正在进行中。
3. 审查完成-分析完成后，AWS Security Agent 会将审查结果发布到您的拉取请求中。所有安全发现都会在一次审查中批量整理，以使您的拉取请求井井有条，并最大限度地减少通知。

了解代码审查结果

AWS Security Agent 会根据其在分析期间发现的内容提供不同类型的结果。

当发现安全问题时

如果 AWS Security Agent 发现您的代码更改中存在安全问题，它会发布一份包含以下内容的评论：

- 摘要-对所有安全发现的高级概述，描述所发现问题的类型及其潜在影响
- 个人调查结果-详细的安全调查结果以主题评论的形式显示在主要审查下，每项发现包括：
  - 安全问题描述
  - 代码中发现问题的位置
  - 解释如何解决问题的补救指南
  - 基于您的代码审查设置的相关上下文（违反安全要求、常见漏洞或两者兼而有之）

> **Note**
>
> 分析的安全问题类型取决于您的代码审查设置。如果您配置了安全要求验证，则发现结果将参考您组织的自定义安全要求。如果您配置了安全漏洞发现，则发现结果将识别出常见的安全漏洞。有关代码审查设置的更多信息，请参阅the section called “为 GitHub 仓库启用拉取请求代码审查”。

未发现任何安全问题时

如果 AWS Security Agent 完成分析后发现您的代码更改中没有安全问题，则会发布一条评论：“未发现任何问题。”这确认审查已成功完成，并且根据您配置的代码审查设置，您的代码更改未触发任何安全发现。

回应安全调查结果

查看 AWS 安全代理发布的安全发现后，您可以直接在源代码控制提供商处采取行动。

- 解决发现问题-根据调查结果中提供的补救指南更新您的代码，然后将新的提交推送到拉取请求。AWS 安全代理将分析更新的代码。
- 解决对话-解决安全发现后，将对话标记为已解决以跟踪您的进度。

> **Tip**
>
> 每项发现都包括针对所发现的安全问题量身定制的具体补救指南。请仔细阅读本指南，了解安全风险以及如何有效解决该风险。

筛选代码审查结果

您可以通过向存储库添加`filtering.md`文件来自定义 AWS 安全代理分析代码的方式。此文件允许您通过提供有关代码库的上下文以及将文件或文件夹排除在分析范围之外，从而减少误报。

创建筛选文件

`filtering.md`在存储库根`.awssecurityagent`目录下创建一个名为的文件：

```
.awssecurityagent/filtering.md
```

在分析拉取请求时，AWS Security Agent 会从存储库的主分支（例如`main`或`mainline`）读取此文件。

文件结构

该`filtering.md`文件使用标准 Markdown 格式，具有 AWS 安全代理可以识别的特定部分。文件必须包含`Code Review`标题，后跟以下一个或两个部分：`IgnorePatterns`和`ContextHints`（无空格）。

以下示例显示了filtering.md文件的完整结构：

```
# filtering.md

## Code Review

### IgnorePatterns

**/*.md

/myapp/src/**/*.snap

/myapp/config/README

### ContextHints

- The backend is a trusted system and won't return non-standard protocols.
- URL is generated from server with presigned token, so no SSRF security
 vulnerabilities.
- AppSec has verified that we are allowed to use cache with an eviction policy.
```

忽略模式

该IgnorePatterns部分指定了 AWS 安全代理在代码审查期间应跳过的文件和文件夹。用于定义glob patterns要从分析中排除哪些路径。

格式要求：

- 每个图案都必须位于自己的线上。
- 用空线分隔每个图案。这样可以确保在 GitHub 或代码审查工具中查看文件时可以正确呈现。
- 模式遵循标准的 glob 格式。例如，**/*.md匹配所有 markdown 文件，/myapp/src/**/*.snap匹配根目录下/myapp/src/文件夹内的所有.snap文件。
- 在本节中，我们最多支持 1000 个忽略模式。

上下文提示

本ContextHints节提供了有关您的代码库的其他背景信息，可帮助 AWS 安全代理进行更准确的评估。使用上下文提示来解释架构决策、安全异常或其他可能影响结果解释方式的信息。

格式要求：

- 每个提示都必须以短划线 (-) 开头，后跟一个空格。
- 将每个提示写成一行不超过 500 个字符的自由格式文本。
- 每个提示都应描述有关您的代码库的一个特定上下文。
- 在本节中，我们最多支持 20 个上下文提示。

在 AWS Security Agent 完成其初始分析后应用上下文提示，这有助于筛选不适用于您的特定用例的发现。

后续步骤

查看代码安全发现后：

- 根据补救指南更新您的代码
- 推送新提交以触发对更改的重新分析
- 如果需要，调整代码审查设置（请参阅the section called “为 GitHub 仓库启用拉取请求代码审查”）
- 查看贵组织的安全要求以了解验证标准
- 考虑进行渗透测试，对已部署的应用程序进行全面的安全验证

创建代码审查

在 AWS 安全代理 Web 应用程序中创建代码审查，以扫描您的源代码存储库和 S3 源代码是否存在安全漏洞。代码审查会对整个代码库进行全面的静态分析，识别安全问题并提供补救指导。

与分析单个代码更改的基于拉取请求的代码审查不同（参见the section called “查看拉取请求中的代码安全发现”），按需代码审查会扫描您的完整源代码以识别安全漏洞并验证是否符合组织的安全要求。

在此过程中，您将通过选择源代码输入、配置权限并运行审查来创建代码审查。

先决条件

在开始之前，请确保您满足以下条件：

- 访问 AWS 安全代理 Web 应用程序
- 您的代理空间中至少有一个已连接的存储 GitHub 库或 S3 存储桶

> **Tip**
>
> 如果您已经将 GitHub 存储库连接到 Agent Space，则代码审查已准备就绪，无需进行其他设置。从 Agent Space 页面的代码审查卡片中选择 “在 We b 应用程序中启动”，或者直接启动 Web 应用程序。
> 如果您需要连接其他源或配置 S3 存储桶，请参阅the section called “启用代码审查”。

访问代码审查页面

导航到 Web 应用程序中的代码审查部分。

1. 登录 AWS 安全代理 Web 应用程序。
2. 在左侧边栏中，选择代码审查。
3. 您会看到现有代码审查的列表及其源信息、上次运行状态和结果摘要。

创建代码审查

通过配置其源代码输入和权限来设置新的代码审查。

1. 在代码审查页面上，选择创建代码审查。

配置代码审查详细信息

提供标题并选择要查看的源代码。

1. 在标题字段中，输入代码审查的描述性名称。

> **Tip**
>
> 使用标识应用程序、存储库或审查范围的名称。例如，“计费服务安全审查” 或 “基础架构代码审计”。

2. 在 “来源” 部分，选择此评论的源代码输入。从两个选项卡中选择：

GitHub 存储库

从连接到您的代理空间的存储库中进行选择。

1. 选择 “GitHub 存储库” 选项卡。
2. 在 “集成存储库” 表格中，选中要包含在审阅中的每个存储库旁边的复选框。
3. 使用搜索字段按名称查找特定的存储库。

> **Note**
>
> 此处仅显示通过代码审查配置连接到代理空间的存储库。要添加更多存储库，请在管理控制台中选择 “管理” 或要求管理员更新 Agent Space 配置。

S3 来源

从连接到您的代理空间的 S3 存储桶中选择 ZIP 文件。您的 Agent Space 管理员可以配置哪些 S3 存储桶可用。存储在其中一个存储桶中的任何 ZIP 文件都可用作代码审查的来源。

1. 选择 S 3 源选项卡。
2. 输入您要包含在审阅中的每个 ZIP 文件的 S3 URI。您最多可以添加 30 个 S3 源。

> **Note**
>
> S3 源必须是存储在连接到代理空间的 S3 存储桶中的 ZIP 文件。要提供更多存储桶，请参阅the section called “启用代码审查”。

配置 权限

为此代码审查选择 IAM 服务角色和可选 CloudWatch 日志组。

1. 在 “权限” 部分中，找到 “服务角色” 下拉列表。
2. 从您配置的服务角色中选择 IAM 角色。

> **Note**
>
> 服务角色必须有权访问 S3 中的源代码并写入 CloudWatch 日志，以及代码审查所需的任何其他 AWS 资源。服务角色是在 AWS 管理控制台中的代码审查设置期间配置的。

3. （可选）在CloudWatch 日志组下拉列表中，选择用于存储代码审查执行日志的日志组。

> **Note**
>
> 如果您未选择日志组，AWS 安全代理会创建一个用于存储代码审查日志的默认日志组。

配置自动代码修复

启用自动修复，让 AWS Security Agent 在审查完成后立即为所有发现生成代码修复。

1. 在 “自动代码修复” 部分，选中 “启用自动代码修复” 复选框。

AWS 安全代理如何提供修复取决于来源：

- 私有 GitHub 存储库 — AWS 安全代理向存储库提交包含修复的拉取请求。
- 公共 GitHub 存储库 — 为了避免在漏洞修复之前将其泄露，AWS 安全代理不会打开拉取请求。取而代之的是，它在发现结果中附上了建议的差异，你可以从网络应用程序下载并私下申请。
- S3 源-代码补救不可用。查看发现的详细信息并手动应用修复程序。

> **Important**
>
> 所有具有读取权限的人都可以看到提交到私有仓库的修复拉取请求。合并前请查看更改。仅当选择 GitHub 存储库作为源时，自动代码修复才可用。

> **Note**
>
> 禁用后，您仍然可以在审查完成后手动触发针对单个 GitHub-sourced 发现的代码修复。

配置模拟验证

启用模拟验证以动态确认发现的漏洞是否可在正在运行的应用程序中被利用。

1. 在 “模拟验证” 部分，选中 “启用模拟验证” 复选框。

启用后，AWS 安全代理会在静态分析完成后配置模拟环境。它加载您的源代码，启动环境内部的服务，并尝试利用扫描期间发现的漏洞。然后，使用验证状态更新调查结果，表明该漏洞是否已成功利用。

> **Note**
>
> 模拟验证目前仅适用于独立的 dockeriable 应用程序。当选择多个存储库作为来源时，模拟验证不可用。

> **Important**
>
> 模拟验证会增加代码审查运行的处理时间。验证步骤预置环境并运行漏洞利用尝试。这通常需要 1-3 个小时，具体取决于发现的数量和应用程序的复杂性。

允许的网络目的地

模拟环境的出站网络访问权限仅限于预定义的许可名单。在验证期间，您的应用程序可以访问以下域：

下表列出了允许的域名及其用途。

域：	用途
`.amazonaws.com` ，`.aws.amazon.com`	AWS 服务
`.public.ecr.aws`	Amazon ECR Public
`.docker.com` ，`.docker.io`	Docker Hub
`.github.com` ，`.githubusercontent.com`	GitHub
`.gitlab.com`	GitLab
`.npmjs.com` ，`.npmjs.org`	npm 注册表

域：	用途
.pypi.org ，.pypi.python.org ，.pythonhosted.org	Python 包索引
.crates.io ，.rustup.rs	Rust 软件包
.maven.org ，.gradle.org	Java/Gradle 包裹
.nuget.org	.NET 包
.rubygems.org ，.ruby-lang.org	Ruby 软件包
.golang.org ，.pkg.go.dev ，.goproxy.io	Go 套餐
.nodejs.org ，.yarnpkg.com	Node.js
.alpinelinux.org ，.debian.org ，.ubuntu.com ，.centos.org ，.fedoraproject.org	Linux 发行版本库
.cloudfront.net	CloudFront 分布
.google.com ，.googleapis.com	谷歌 API
.microsoft.com ，.visualstudio.com	微软服务
.sourceforge.net ，.bitbucket.org	源托管

> **Note**
>
> 如果您的应用程序需要网络访问不在此列表中的域，则网络防火墙将阻止该连接。依赖上面未列出的外部 API 或服务的应用程序可能无法在模拟环境中正确启动。

创建代码审查

1. 检查您的配置以确保准确性。
2. 选择 “创建代码审查”。

您将被重定向到代码审查详细信息页面，您可以在其中开始审核。

运行代码审查

创建代码审查后，开始运行以开始分析。

1. 在代码审查详细信息页面上，选择开始审查。
2. AWS 安全代理开始分析您的源代码。

您也可以从 “代码审查” 列表页面开始审阅，方法是选择要运行的代码审查旁边的 “开始审阅”。

监控代码审查的运行情况

跟踪代码审查的执行进度。

查看运行阶段

代码审查运行会经历以下阶段，显示为进度指示器：

1. 预检 — AWS 安全代理会验证您的源代码访问权限并设置测试环境。预检检查包括：
   - 服务基础设施设置
   - S3 源访问验证
   - 设置测试环境
2. 静态分析 — AWS 安全代理会扫描您的源代码中是否存在安全漏洞和违反要求的情况。

3. 模拟验证（可选）— 如果启用了模拟验证，AWS Security Agent 将配置模拟环境并部署您的应用程序。然后，它会尝试利用静态分析期间发现的漏洞。此步骤确认目标运行时发现的结果是否可被利用。只有当您在创建代码审查期间启用模拟验证时，才会出现此阶段。
4. 最终确定 — AWS 安全代理汇编调查结果并生成结果摘要。

查看运行详情

在运行详细信息页面上，在选项卡之间导航以监控进度：

- 代码审查运行-查看运行摘要，包括运行 ID、创建时间、状态、持续时间、任务时长、严重级别细分和风险类型图表。
- 印前检查-查看每个验证步骤的印前检查进度和状态。
- 代码审查日志 — 查看 AWS Security Agent 在审查期间确定和执行的任务，以及每个步骤的详细任务日志。
- 模拟验证-启用模拟验证后，查看配置状态、单个发现的验证任务及其结果。
- 调查结果-审查完成后查看安全调查结果（请参阅the section called “查看代码审查的结果”）。

运行历史记录

每次代码审查都会保留所有运行的历史记录。在代码审查详情页面上：

- “最新运行” 部分显示最近的运行及其开始时间、状态、持续时间和 ID。
- “所有运行” 表列出了之前的所有运行及其开始时间、状态、持续时间、结果摘要和 ID。
- 选择 “监控运行” 以查看最新活动运行的详细信息。
- 选择任何跑步的开始时间链接以查看其完整详细信息。

后续步骤

运行代码审查后：

- 查看安全调查结果及其补救指南（请参阅the section called “查看代码审查的结果”）
- 通过自动拉取请求或手动修复来修复发现（请参阅the section called “修复代码审查结果”）
- 实施修复后再进行审查以验证补救措施
- 随着代码库的发展，调整代码审查配置或源代码

查看代码审查的结果

代码审查运行完成后，查看运行摘要和安全发现，以了解源代码中的漏洞。每项调查结果都包含描述、严重性评级、代码位置、风险推理和建议的修复方法，以帮助您确定安全问题的优先级并解决这些问题。

先决条件

在开始之前，请确保您满足以下条件：

- 已完成或正在运行的代码审查
- 访问 AWS 安全代理 Web 应用程序

第 1 步：访问代码审查运行

导航到已完成的代码审查运行以查看摘要和发现。

1. 登录 AWS 安全代理 Web 应用程序。
2. 在左侧边栏中，选择代码审查。
3. 选择要检查的代码审查。
4. 在 “所有运行” 表中，通过单击其开始时间链接来选择已完成的运行。或者，选择 “监控运行” 以查看最新运行情况。

步骤 2：监控运行进度

使用步骤指示器跟踪代码审查运行的进度。

1. 在页眉下方找到水平步进指示器。
2. 查看每个阶段的状态：
   - 印前检查-验证对源代码的访问权限并设置扫描环境。检查包括服务基础设施设置、S3 源访问验证和测试环境设置（包括 GitHub 访问检查）。
   - 静态分析-扫描源代码中是否存在安全漏洞和违反要求的情况。
   - 模拟验证（可选）-启用后，将配置模拟环境并尝试利用已发现的漏洞来攻击正在运行的应用程序。
   - 定稿-汇编调查结果并生成结果摘要。

> **Note**
>
> 每个步骤都显示一个状态指示器（“已完成”或“进行中”）。当所有阶段都显示已完成时，运行即告完成。只有当您在创建代码审查期间启用模拟验证时，才会显示模拟验证步骤。

步骤 3：查看运行摘要

导航到“代码审查运行”选项卡以查看高级结果。

1. 选择“代码审查运行”选项卡。
2. 运行摘要部分提供了运行状态、持续时间和其他高级详细信息。它还提供了按严重级别和风险类型分类的安全发现的仪表板。
3. AWS Security Agent 提供的应用程序概述提供了运行完成后的代码审查扫描摘要

步骤 4：查看印前检查结果

导航至“印前检查”选项卡，确认所有来源访问检查均已通过。

1. 选择“印前检查”选项卡。
2. 查看显示已完成检查数量的印前检查进度指示器。
3. 确认每项检查都显示成功状态：
   - 服务基础架构设置-确认测试环境已准备就绪
   - S3 源访问验证-确认对 S3 源代码的访问权限（如果适用）
   - 设置测试环境-确认分析环境已配置

> **Note**
>
> 如果任何印前检查失败，则运行将无法继续进行静态分析。查看检查详细信息以识别和解决访问问题，然后开始新的运行。

第 5 步：查看代码审查日志

导航到代码审查日志选项卡，查看 AWS 安全代理在分析期间执行的任务。

1. 选择“代码审查日志”选项卡。

2. 浏览左侧面板中的任务列表。
3. 选择一项任务，在右侧面板中查看其详细日志。

> **Tip**
>
> 使用搜索字段按名称筛选任务。这些日志可以深入了解 AWS 安全代理检查了哪些内容以及它是如何得出结论的。

步骤 6：导航到调查结果

选择 Find in gs 选项卡可查看运行中的所有安全调查结果。

> **Note**
>
> 默认置信度过滤器
>
> 默认情况下，您只能看到代理置信度高的结果。要同时显示代理置信度为中或低且误报的结果，请关闭隐藏未经验证的发现开关。

1. 选择 “调查结果” 选项卡。
2. 调查结果以分割视图显示，左边是发现结果列表，右边是选定结果的详细信息。
3. 查看每张搜索卡上显示的信息：
   - 查找标题-概述安全问题的描述性名称
   - 严重性标记- Color-coded 严重性指示器：
     - 严重（红色）-需要立即采取行动；利用漏洞可能导致系统受损
     - 高（红色）-需要立即注意；利用漏洞可能会对安全造成重大影响
     - 中（橙色）— 应在合理的时间范围内解决；会增加整体安全风险
     - 低（黄色）— 可作为定期维护的一部分解决；将直接风险降至最低
   - 上次更新-上次更新调查结果的时间戳

第 7 步：查看查找详情

选择单个发现结果以查看有关每个漏洞的全面信息。

1. 在左侧面板中选择一个查找结果，在右侧面板中显示其详细信息。
2. 查看可用的操作：
   - 解决查找结果-解决问题后，将查找结果标记为已解决
   - 修复代码-生成包含修复的拉取请求（适用于 GitHub 源代码）
3. 使用此调查结果准确吗？提供反馈 — 选择 “是” 或 “否” 以帮助提高 future 分析的准确性。
4. 查看 “概述” 部分中的关键属性：
   - 代理信心 — AWS Security Agent 对这一发现的信心水平
   - 严重性-带有颜色编码徽章的风险等级
   - 风险类型-安全风险的类别
   - 验证状态-启用模拟验证后，表示发现结果是否已确认可在运行环境中被利用：
     - 已确认-该漏洞已在模拟环境中成功利用
     - 未复制-无法在目标运行时利用该漏洞
     - 验证失败-由于错误，验证尝试未得出结论
     - 未验证-未对此发现进行模拟验证。当验证未启用或验证步骤在得出此结果之前超时时，就会发生这种情况。
   - 已解决-调查结果是否已解决 (Yes/No)
   - 上次更新-最近更新的时间戳
5. 将 “描述” 部分展开，内容为：
   - 安全问题的详细解释
   - 如何利用该漏洞
   - 对您的应用程序的潜在影响
   - 代理为确认调查结果而执行的验证步骤
6. 展开代码位置部分以查看：
   - 发现问题的特定文件路径
   - 简要描述在每个地点发现的东西
   - 链接到相关代码的行号徽章
7. 展开 “证据” 部分以查看：
   - 支持该发现的特定代码、配置或观测结果
   - 简要解释为什么每件物品都显示了漏洞
   - 每份证据的受影响文件和行号
8. 展开 “建议的修复” 部分以查看：

- AWS Security Agent 为补救发现而建议的更改
- 每项建议更改的示例代码或配置

> **Tip**
>
> 使用代码位置快速导航到存储库中受影响的文件。每个位置都包含足够的上下文，无需阅读整个文件即可理解问题。

第 8 步：确定发现的优先顺序并解决问题

验证发现并对发现的结果采取行动，以修复漏洞并改善应用程序的安全状况。

对于具有高严重性且代理可信度高的发现：

1. 仔细查看 “描述” 和 “代码位置” 部分。
2. 使用 R emediate 代码生成包含修复的拉取请求，或者查看自动修复 PR（如果您启用了该选项）。
3. 计划进行后续代码审查，以验证修复是否有效。

对于中等严重程度且代理可信度高的发现：

1. 根据您的风险承受能力和业务环境确定优先级。
2. 在您的常规开发冲刺计划中包括补救任务。
3. 考虑多项中等严重程度的发现是否会带来更高的风险。

对于中等或低置信度调查结果：

1. 查看描述和代码位置部分，以验证漏洞发生的假设和条件。
2. 如果漏洞有效，请根据严重性和风险承受能力确定优先级，并考虑修复任务。

步骤 9：跟踪修复进度

使用 findings 界面并重新运行来跟踪哪些漏洞已得到解决。

1. 在实施修复时，使用 Resolve finding 将查找结果标记为已解决。
2. 运行新的代码审查，以验证修复是否解决了发现的问题，并且不会引入新问题。

3. 查看代码审查详细信息页面上的 “所有运行” 表，比较一段时间内不同运行的结果。

> **Tip**
>
> 合并修复拉取请求后，开始重新运行相同的代码审查，以确认漏洞已得到解决。比较两次跑步的发现次数，以跟踪您的进度。

后续步骤

查看您的代码审查结果后：

- 高度可信地确定高严重性发现的优先顺序，以便立即进行补救
- 使用 R emediate 代码为 GitHub-sourced 发现结果生成自动修复（请参阅the section called “修复代码审查结果”）
- 实施修复后运行其他代码审查以验证补救措施
- 随着代码库的发展，调整代码审查源或设置（请参阅）the section called “启用代码审查”

修复代码审查结果

查看代码审查的安全发现后，您可以使用 AWS 安全代理针对 GitHub 存储库源中的发现生成代码修复程序。对于私有存储库，AWS 安全代理会打开包含建议修复程序的拉取请求。对于公共存储库，AWS Security Agent 会将可下载的差异附加到您可在本地应用的发现中。来自 S3 来源的发现必须手动修复。

先决条件

在开始之前，请确保您满足以下条件：

- 已完成的代码审查并附有调查结果
- GitHub 作为代码审查来源连接的存储库
- 访问 AWS 安全代理 Web 应用程序
- 熟悉应用程序的架构和安全要求

代码修复的工作原理

当您触发发现的代码修复时，AWS 安全代理会分析该发现及其代码位置，然后生成代码修复。修复程序的交付方式取决于来源：

- 私有 GitHub 存储库 — AWS 安全代理向相关存储库提交拉取请求。拉取请求包括对安全问题和所做更改的描述。
- 公共 GitHub 存储库 — AWS Security Agent 在发现中附上了建议的差异，因此漏洞在修复之前不会被披露。您可以从 Web 应用程序下载差异，然后使用将其应用到本地。`git apply /path/to/code_remediation_changes.diff`
- S3 源-代码补救不可用。使用发现的描述、代码位置和风险推理来手动应用修复。

> **Important**
>
> AWS Security Agent 创建的拉取请求对所有拥有存储库读取权限的用户都可见。合并前请检查更改，确保它们符合应用程序的要求。

自动代码修复

如果您在创建代码审查时启用了自动代码修复，AWS Security Agent 会在审查完成后立即为所有符合条件的发现生成修复程序。您无需采取任何其他操作——当运行结束时，拉取请求会出现在您连接的私有 GitHub 存储库中，而差异会出现在公共 GitHub 存储库的结果上。

手动代码修复

如果禁用了自动代码修复，则可以针对来自 GitHub 源的单个发现触发补救措施。

1. 导航到已完成的代码审查运行的 “发现” 选项卡。
2. 选择要修复的调查结果。
3. 在查找结果详细信息面板中，选择修复代码。
4. AWS Security Agent 会生成代码修复，并根据存储库的可见性向存储库提交拉取请求或在发现结果中附加可下载的差异。

查看补救拉取请求

AWS 安全代理向私有 GitHub 存储库提交补救拉取请求后：

1. 导航到您的 GitHub 存储库。
2. 找到 AWS 安全代理创建的拉取请求。
3. 查看更改，包括：
   - 解释安全发现和修复的描述
   - 针对该漏洞的代码进行了更改
   - 有关补救方法的任何相关背景
4. 如果修复合适，请合并拉取请求，或者将其关闭并实施替代解决方案。

> **Tip**
>
> 合并修复拉取请求后，开始运行新的代码审查，以验证修复程序是否解决了发现的问题，并且不会引入新的安全问题。

应用补救差异

要获得来自公共 GitHub 存储库的结果，请在本地应用可下载的差异。

1. 在查找结果详细信息面板中，从 “代码修复” 部分下载差异。
2. 在存储库的本地克隆中，运行`git apply /path/to/code_remediation_changes.diff`。
3. 查看已应用的更改，根据您的应用程序对其进行测试，然后通过正常的开发工作流程进行提交。

限制

- 代码补救仅适用于来自 GitHub 存储库来源的发现。来自 S3 来源的发现必须手动修复。
- AWS 安全代理根据其对漏洞的分析生成修复程序。在合并或提交更改之前，请先查看所有更改，以确保它们适合您的应用程序。
- 除了自动修复之外，一些复杂的发现可能需要手动干预。

后续步骤

修复发现后：

- 查看和合并 GitHub 仓库中的拉取请求，或者通过正常工作流程提交已应用的差异

- 运行新的代码审查以验证修复并检查是否存在剩余问题
- 确认补救后解决 Web 应用程序中的发现问题
- 根据需要调整代码审查来源或设置（请参阅the section called “启用代码审查”）

使用 S3 运行差分代码扫描

运行差异（差异）代码扫描，仅分析源代码中更改的行，而不是执行完整的存储库扫描。差异扫描比完全扫描更快，可以生成针对特定代码更改的结果，因此非常适合开发工作流程中的合并前验证。

与由第三方源代码控制提供商事件自动触发的基于拉取请求的代码审查不同（参见the section called “查看拉取请求中的代码安全发现”），差异扫描是通过 AWS Security Agent API 或 SDK 以编程方式启动的。您可以将统一的差异文件上传到 S3，并在开始代码审查任务时引用该文件。

差异扫描的工作原理

差异扫描会在仓库的完整上下文中分析您的代码更改。当您使用源代码上传到 S3 创建代码审查资源时，差异扫描会使用完整的存储库作为上下文，同时将发现的重点放在差异中更改的行上。这使扫描能够识别您的更改与现有代码的交互方式所产生的安全问题，例如身份验证流程中断、跨模块的数据处理不安全或暴露现有漏洞的更改。

扫描过程:. 您将统一的差异文件（的输出`git diff`）上传到连接到代理空间的 S3 存储桶。您使用指向差异的 S3 位置的`diffSource`参数调用 `StartCodeReviewJob` API。AWS Security Agent 仅分析差异中已更改的代码，以确定是否存在安全漏洞以及是否符合贵组织的安全要求。调查结果仅限于更改后的行，包括严重性等级、代码位置和补救指南。

先决条件

在开始之前，请确保您满足以下条件：

- 启用了代码审查的代理空间（请参阅the section called “启用代码审查”）
- 已为目标存储库创建的代码审查资源，其完整源代码已上传到连接到代理空间的 S3 存储桶。完整的存储库提供了上下文，可以更深入地分析您的更改与现有代码的交互方式。有关创建代码审查和上传源代码的说明，请参阅the section called “创建代码审查”。
- 连接到您的代理空间的 S3 存储桶
- 上传到 S3 存储桶并调用的 IAM 权限 `securityagent:StartCodeReviewJob`
- 根据您的代码更改生成的统一差异文件（例如，的`git diff main..feature-branch`输出）

> **Tip**
>
> 为了获得最准确的结果，请确保您的代码审查资源已将完整源代码的最新版本上传到 S3。差异扫描使用这个完整的存储库作为上下文，在分析更改的行时了解您的应用程序架构、数据流和现有的安全控制措施。

步骤 1：生成差异文件

生成一个表示要扫描的代码更改的统一差异。

```
git diff main..feature-branch > changes.diff
```

或者，为分阶段的更改生成差异：

```
git diff --cached > changes.diff
```

> **Tip**
>
> 差异文件应采用标准的统一差异格式。AWS Security Agent 会解析文件路径并更改差异标头中的行以确定其分析范围。

第 2 步：将差异上传到 S3

将差异文件上传到连接到您的代理空间的 S3 存储桶。

```
aws s3 cp changes.diff s3://my-security-agent-bucket/diffs/changes.diff
```

> **Important**
>
> S3 存储桶必须是已连接到您的代理空间的存储桶。与您的代理空间关联的 IAM 服务角色必须对该位置具有读取权限。有关连接 S3 存储桶的说明，请参阅the section called “启用代码审查”。

第 3 步：开始差异代码审查作业

使用diffSource参数调用 StartCodeReviewJob API 以启动差异扫描。

使用 Amazon CLI

```
aws securityagent start-code-review-job \
  --agent-space-id "your-agent-space-id" \
  --code-review-id "your-code-review-id" \
  --diff-source '{"s3Uri": "s3://my-security-agent-bucket/diffs/changes.diff"}'
```

使用 AWS 开发工具包 (Python)

```
import boto3

client = boto3.client('securityagent')

response = client.start_code_review_job(
    agentSpaceId='your-agent-space-id',
    codeReviewId='your-code-review-id',
    diffSource={
        's3Uri': 's3://my-security-agent-bucket/diffs/changes.diff'
    }
)

print(f"Job started: {response['codeReviewJobId']}")
```

使用 AWS 开发工具包 (JavaScript/TypeScript)

```
import { SecurityAgentClient, StartCodeReviewJobCommand } from '@aws-sdk/client-
securityagent';

const client = new SecurityAgentClient({ region: 'us-east-1' });

const response = await client.send(new StartCodeReviewJobCommand({
  agentSpaceId: 'your-agent-space-id',
  codeReviewId: 'your-code-review-id',
  diffSource: {
    s3Uri: 's3://my-security-agent-bucket/diffs/changes.diff',
  },
}));
```

```
console.log(`Job started: ${response.codeReviewJobId}`);
```

API 会立即返回，codeReviewJobId并设置为 status “和” IN_PROGRESS。

步骤 4：监控扫描

轮询代码审查任务状态，直到其完成。

```
aws securityagent get-code-review-job \
  --agent-space-id "your-agent-space-id" \
  --code-review-id "your-code-review-id" \
  --code-review-job-id "the-job-id"
```

作业的进展与完整的代码审查相同：印前检查 → 静态分析 → 定稿。差异扫描通常比完整扫描更快地完成，因为差异扫描分析的代码行更少。

步骤 5：查看调查发现

作业完成后，列出代码审查作业的结果。

```
aws securityagent list-findings \
  --agent-space-id "your-agent-space-id" \
  --code-review-id "your-code-review-id" \
  --code-review-job-id "the-job-id"
```

每项调查发现包括：

- 对安全问题的描述
- 严重性级别（严重、高、中或低）
- 引用差异中特定行的代码位置
- 解释潜在影响的风险推理
- 补救指南

有关了解和根据发现采取行动的更多信息，请参阅the section called “查看代码审查的结果”。

限额和限制

- 差异中更改文件的最大数量：3,000 个文件或 5 MB 的差异总大小

- 每次扫描的结果上限为 30 个，按严重程度排序（严重 > 高 > 中 > 低）

后续步骤

运行差异扫描后：

- 查看调查结果并应用补救指南（请参阅the section called “查看代码审查的结果”）
- 启用拉取请求注释，以便对每个拉取请求进行自动代码审查（请参阅the section called “为 GitHub 仓库启用拉取请求代码审查”）
- 定期运行完整的代码审查，以发现个别更改之外的问题（参见the section called “创建代码审查”）
- 使用 IDE 集成直接从开发环境运行差异扫描（请参阅the section called “从 IDE 运行代码安全扫描”）

从 IDE 运行代码安全扫描

使用 Kiro 或 Claude 代码直接从您的 IDE 运行 AWS 安全代理代码安全扫描。IDE 集成允许您扫描本地源代码中是否存在安全漏洞，仅对更改后的代码运行差异扫描，以及对设计文档进行威胁模型审查，所有这些都无需离开开发环境。发现结果显示在代码旁边，并附有补救指南，您可以直接从 IDE 中应用自动修复。

IDE 集成的工作原理

IDE 集成使用 AWS 安全代理 MCP（模型上下文协议）服务器将您的 IDE 连接到 AWS 安全代理服务：

1. MCP 服务器将您的源代码打包到 ZIP 存档中。
2. 档案将上传到服务器在您的 AWS 账户中自动配置的 S3 存储桶。
3. 服务器调用 AWS 安全代理 API 开始扫描。
4. AWS Security Agent 会分析代码是否存在安全漏洞以及是否符合贵组织的安全要求。
5. 结果将返回到 IDE，包括代码位置、描述、严重性评级和补救建议。

MCP 服务器负责处理所有编排（代理空间配置、S3 存储桶创建、IAM 角色设置、代码打包和扫描轮询），因此您只需要在本地配置 AWS 凭证即可。

> **Note**
>
> 唯一的先决条件是在您的本地环境中配置 AWS 证书（例如`aws configure`，通过 AWS SSO 或环境变量）。MCP 服务器会在首次使用时自动配置代理空间、IAM 服务角色和 S3 存储桶。

先决条件

在开始之前，请确保您满足以下条件：

- 使用以下权限在本地配置的 AWS 证书：
  - `iam:CreateRole`，`iam:PutRolePolicy`（用于一次性设置服务角色）
  - `s3:CreateBucket`, `s3:PutObject`, `s3:PutPublicAccessBlock`, `s3:PutLifecycleConfiguration`
  - `securityagent:CreateAgentSpace`, `securityagent:UpdateAgentSpace`, `securityagent:ListAgentSpaces`, `securityagent:BatchGetAgentSpaces`
  - `securityagent:CreateCodeReview`, `securityagent:StartCodeReviewJob`, `securityagent:BatchGetCodeReviewJobs`
  - `securityagent:ListFindings`, `securityagent:BatchGetFindings`
  - `securityagent:StartCodeRemediation`（用于自动修复）
  - `sts:GetCallerIdentity`
- 已安装 uv（Python 包运行器）
- Python 3.10 或更高版本
- 以下 IDE 之一：
  - Kiro（安装 AWS 安全代理功能）
  - Claude Code（安装 AWS 安全代理插件）

安装 MCP 服务器

Kiro

从 Kiro 市场安装 AWS 安全代理 Power。Power 将 MCP 服务器配置、转向指令和生命周期挂钩捆绑在一起，以提供自动安全扫描建议。

或者，也可以在项目中手动配置 MCP 服务器：`.kiro/mcp.json`

```
{
  "mcpServers": {
    "security-agent": {
      "command": "uvx",
      "args": ["awslabs.security-agent-mcp-server@latest"],
      "env": {
        "AWS_PROFILE": "default",
        "AWS_REGION": "us-east-1",
        "FASTMCP_LOG_LEVEL": "ERROR"
      }
    }
  }
}
```

Claude Code

安装 AWS Security Agent 插件，该插件提供设置、完整扫描、差异扫描、威胁模型审查和补救的技能。

在项目中配置 MCP 服务器：`.mcp.json`

```
{
  "mcpServers": {
    "awslabs.security-agent-mcp-server": {
      "command": "uvx",
      "args": ["awslabs.security-agent-mcp-server@latest"],
      "env": {
        "AWS_PROFILE": "default",
        "AWS_REGION": "us-east-1",
        "FASTMCP_LOG_LEVEL": "ERROR"
      }
    }
  }
}
```

> **Tip**
>
> 如果你不想在本地安装 Python，也可以通过 Docker 运行 MCP 服务器。有关 Docker 配置的信息，请参阅 MCP 服务器文档。

First-time 设置

首次使用时，MCP 服务器会自动配置所需的 AWS 资源：

资源	命名惯例	用途
特工空间	User-chosen 或自动创建	用于存放扫描、评论和笔试的容器
IAM 服务角色	`SecurityAgentScanRole`	假设读`securityagent.amazonaws.com` 取上传的代码
S3 存储桶	`security-agent-scans-{account}-{region}`	存储压缩后的源代码（30 天自动过期生命周期）

要明确触发安装程序，请询问您的 IDE：

```
Set up the security agent
```

该设置会验证您的 AWS 证书，创建或重复使用代理空间，配置 IAM 角色和 S3 存储桶，并确认准备就绪。如果您已经有来自 AWS 管理控制台的代理空间，则设置会显示它们并询问要使用哪一个。

> **Note**
>
> 如果尚未配置，则安装程序会在首次扫描时自动运行。您无需单独运行它。

运行全面的安全扫描

扫描整个项目是否存在安全漏洞：

```
Scan this project for security issues
```

IDE：

1. 存档您的源代码（不包括`.git`、`node_modules`、、构建工件和其他非必要目录）

2. 将档案上传到 S3
3. 开始完整的代码审查作业
4. 投票完成时间（通常在 1 小时左右）
5. 显示按严重程度分组的调查结果

扫描进度

完全扫描通常需要大约 1 小时，具体取决于代码库的大小。IDE 每 5 分钟检查一次进度，并在结果准备就绪时通知您。您可以随时询问状态：

```
How's the scan going?
```

运行差异扫描

为了在开发过程中更快地获得反馈，请仅扫描自 git ref 以来更改过的代码：

```
Scan my changes against main for security issues
```

默认情况下，如果您未指定基本引用，则扫描会将未提交的更改与之进行比较。HEAD你可以明确指定不同的基数：

```
Diff scan my uncommitted changes
```

差异扫描会上传完整的存储库上下文和 git diff 补丁，然后仅针对更改的行运行分析。结果通常在 5-15 分钟内得出。

有关 S3 差异扫描 API 的更多信息，请参阅the section called “使用 S3 运行差分代码扫描”。

进行威胁模型审查

使用 STRIDE 方法分析设计文档以了解安全态势的变化：

```
Run a threat model review on my spec
```

IDE 会识别您的`requirements.md`和`design.md`文件（通常位于下方`.kiro/specs/`），将其与源代码一起上传，并运行威胁模型分析。结果确定：

- 按STRIDE分类的安全威胁（欺骗、篡改、拒绝、信息泄露、拒绝服务、权限提升）

- 每种威胁的严重性等级
- 对代码库中特定资产的影响
- 缓解建议

> **Tip**
>
> 在根据规范生成实施任务之前，先进行威胁模型审查。这可以在编写代码之前发现安全设计问题。

查看调查发现

扫描完成后，搜索结果将按严重性分组显示在您的 IDE 中：

```
# CRITICAL: SQL Injection in user-service.ts
   File: src/api/user-service.ts:45
   User input flows directly into SQL query without parameterization

# HIGH: Hardcoded credentials in config.ts
   File: src/config/database.ts:12
   Database password stored in source code
```

还会写一份详细的报告，其中`.security-agent/findings-{scan_id}.md`包含每项发现的完整信息，包括风险类型、置信度分数、代码位置和补救代码。

要查看先前扫描的结果，请执行以下操作：

```
Show my security findings
```

应用自动修复

查看调查结果后，应用自动补救措施：

```
Fix the security findings
```

IDE 按严重程度（严重 → 高 → 中 → 低）自上而下处理调查结果，并使用每个发现的补救指南应用代码修复。应用所有修复后，它会自动运行验证差异扫描以确认问题已解决。

您也可以修复个别发现：

```
Fix the SQL injection in user-service.ts
```

> **Important**
>
> 在提交之前，请务必查看自动修复程序。确认修复不会破坏现有功能或引入回归。

自动扫描建议 (Kiro)

使用 Kiro Power 时，AWS 安全代理可以在您更改安全敏感代码后自动建议差异扫描。Power 会安装一个挂钩，该挂钩会评估每个完成的编码回合，并在更改影响以下内容时建议进行扫描：

- 身份验证或授权逻辑
- 密码学或机密处理
- 输入验证或数据清理
- 网络请求或外部集成
- 安全配置（CORS、标头、会话处理）

您可以随时通过删除来选择不接收自动建议`.kiro/hooks/security-diff-scan-suggester.kiro.hook`。

可用的扫描类型

扫描类型	Duration	使用案例	命令
完全扫描	大约 1 小时	全面审查整个代码库	“扫描我的项目是否存在安全问题”
差异扫描	5—15 分钟	仅更改代码（提交前、PR 前）	“扫描我的更改” 或 “与主屏幕进行差异扫描”
威胁模型	5—30 分钟	设计文档（requirements.md、design.md）	“对我的规格进行威胁模型审查”

环境变量

变量	说明	默认
AWS_REGION	适用于 SecurityAgent API 调用的 AWS 区域	us-east-1
AWS_PROFILE	AWS 凭证资料名称	默认配置文件
FASTMCP_LOG_LEVEL	MCP 服务器日志级别（调试、信息、警告、错误）	WARNING

问题排查

“未配置。先运行安装程序。”

您的.security-agent/config.json失踪或代理空间已不存在。让 IDE 运行安装程序：

```
Set up the security agent
```

扫描失败 AccessDenied

确保您的 AWS 凭证具有 “先决条件” 部分中列出的所需 IAM 权限。最常见的缺失权限是iam:CreateRole（仅在首次设置时才需要）。

扫描超时或耗时太长

- 对大型代码库进行全面扫描可能需要 1 个多小时
- 使用差异扫描进行迭代开发 — 只需几分钟即可完成
- 检查您的源存档中是否包含不必要的目录（MCP 服务器会自动排除常见模式）

空差异-扫描不做任何更改

如果您在没有未提交的更改的情况下运行差异扫描，MCP 服务器会报告 “与 HEAD 无更改”，并且不会启动扫描。先提交或暂存您的更改，或者指定不同的基本引用。

限额和限制

- 最大源存档大小：2 GB
- S3 存储桶生命周期：上传的代码在 30 天后自动删除

后续步骤

运行第一次 IDE 安全扫描后：

- 为自动 GitHub 集成启用拉取请求代码审查注释（请参阅the section called “为 GitHub 仓库启用拉取请求代码审查”）
- 为组织特定的策略验证配置安全要求（请参阅）the section called “管理安全要求”
- 定期运行全面扫描，以发现整个代码库中的问题（参见the section called “创建代码审查”）
- 在实施之前，请对新功能规格进行威胁模型审查

创建渗透测试

通过配置测试范围、目标域和 AWS 资源访问权限，为您的 Web 应用程序设置自动渗透测试。渗透测试通过模拟针对经过验证的域名的真实攻击场景，帮助识别正在运行的应用程序中的安全漏洞。

AWS Security Agent 会根据配置的范围和权限对您的 Web 应用程序执行全面的安全测试，在攻击者发现可利用漏洞之前提供有关这些漏洞的详细调查结果。

在此过程中，您将通过配置测试详细信息、定义测试范围和设置所需权限来创建渗透测试。

先决条件

在开始之前，请确保您满足以下条件：

- 访问 AWS 安全代理 Web 应用程序
- 至少有一个经过验证的域名可供测试
- 具有 AWS 安全代理相应权限的 IAM 角色
- 了解应用程序的架构和关键路径

开始创建渗透测试

在 Agent Web 应用程序中导航到渗透测试创建页面。

1. 登录 AWS 安全代理 Web 应用程序。
2. 导航到 “渗透测试” 部分。
3. 选择 “创建渗透测试”。

> **Tip**
>
> 只有经过验证的域名才能包含在渗透测试中。请您的管理员在 AWS 管理控制台中验证域名。请参阅the section called “启用应用程序域进行渗透测试”。

为你的渗透测试命名

提供描述性名称，以帮助确定此渗透测试的目的和范围。

1. 在渗透测试名称字段中，输入渗透测试的描述性名称。

   Example

   名称应清楚地标识正在测试的应用程序、环境或组件。最多 100 个字符。

配置渗透测试范围

定义将测试哪些域和 URL 路径，并配置可选的排除项以控制测试边界。

添加目标域名

指定将主动进行安全漏洞测试的经过验证的域名。

1. 在渗透测试范围部分中，找到目标网址。
2. 展开 “已验证域名” 部分以查看可用域名。
3. 在目标 URL 字段中，输入目标域 URL。

> **Important**
>
> 只能测试经过验证的域名。该 URL 必须位于您之前在 AWS 安全代理中验证过的域名下。Sub-domains 已验证的域名不需要单独验证。

4. 要添加多个目标域，请执行以下操作：
   a. 选择添加域。
   b. 输入每个额外的域名 URL。
5. 要移除目标网域，请选择域名网址旁边的移除。

> **Tip**
>
> 为了获得最佳效果，请包括应用程序用户流中的所有域，包括 API、身份验证服务和内容交付的子域名。 Sub-domains 已验证的父域名不需要单独验证。

排除风险类型（可选）

如果特定风险类别不适用于您的应用程序，请选择将其排除在测试范围之外。

1. 找到 “排除风险类型” 字段。
2. 选择下拉列表以查看可用的风险类型。
3. 选择要从渗透测试中排除的一种或多种风险类型。

> **Note**
>
> 排除风险类型会限制测试范围。仅排除与您的应用程序无关或您想单独测试的风险类型。

添加超出范围的 URL 路径（可选）

指定在渗透测试期间不应测试的 URL 路径。AWS 安全代理排除指定路径及其下方嵌套的所有路径。例如，如果您添加`https://example.com/admin`为超出范围的 URL，`https://example.com/admin/tools`则也超出了范围。

1. 找到 “Out-of-scope 网址” 部分。
2. 在 Out-of-scope URL 输入字段中，输入要排除的 URL 路径（例如，`/admin/delete`或`/api/reset`）。

> **Warning**
>
> Out-of-scope 路径未经过漏洞测试。确保仅排除测试期间不应访问的路径，例如破坏性操作或敏感的管理功能。

3. 要添加多个超出范围的路径，请执行以下操作：
   a. 选择 “添加网址”。
   b. 输入每条其他路径。
4. 要移除路径，请选择路径旁边的 “删除”。

添加可访问的域名（可选）

指定测试所需但不是漏洞测试目标的域。

1. 找到 “可访问的网址” 部分。
2. 在 “可访问的 URL” 输入字段中，输入在测试期间应可访问的域。

> **Note**
>
> 为目标域之外的第三方服务（例如 Okta、Auth0、Stripe）添加可访问的域名。这是必需的，这样 AWS 安全代理才能在测试期间访问这些 URL 进行登录和导航。AWS 安全代理不会对这些域进行渗透测试，它们仅用于访问目的。即使可访问的域名与您的目标域名属于不同的域名，也无需进行所有权验证。只有目标域名需要经过验证的所有权。

3. 要添加多个可访问的域，请执行以下操作：
   a. 选择 “添加网址”。
   b. 输入每个额外的域名。
4. 要移除某个域名，请选择该域名旁边的移除。

添加自定义 HTTP 标头（可选）

指定自定义 HTTP 标头，这些标头将添加到 AWS 安全代理在渗透测试期间发出的任何请求中。

1. 找到 “自定义 HTTP 标头” 部分。
2. 在自定义 HTTP 标头输入字段中，输入将与出站请求关联的标头名称和值。

> **Note**
>
> 默认情况下，除非指定了不同的自定义标头值，否则 AWS 安全代理会将自定义User-Agent标头User-Agent设置为 securityagent。

3. 要添加多个自定义标题，请执行以下操作：
   a. 选择添加标头。
   b. 输入每个额外的自定义标题。
4. 要删除自定义标题，请选择标题旁边的移除。

配置 IAM 角色

为此渗透测试选择预先配置的服务角色。AWS 安全代理使用代理 Space-based 权限模型，即管理员在设置您的代理空间时配置 IAM 角色。您可以从已经配置并可以使用的角色中进行选择。

1. 在 “权限” 部分中，找到 “服务角色” 下拉列表。
2. 选择授予 AWS 安全代理访问所需 AWS 资源的权限的 IAM 角色。

> **Important**
>
> 所选的 IAM 角色必须有权访问渗透测试所需的 VPC 资源、 CloudWatch 日志和任何其他 AWS 服务。验证该角色与 AWS 安全代理的信任关系是否正确。

3. 找到CloudWatch 日志组下拉列表。
4. 选择将存储渗透测试日志的日志组。 （可选）

> **Note**
>
> 选定的 CloudWatch 日志组将存储渗透测试执行的详细日志，包括发出的请求、收到的响应和发现的漏洞。
> 如果您不选择日志组，则会自动创建一个带有`/aws/securityagent`前缀的新 CloudWatch 日志组来存储渗透测试日志。

自动代码修复

选中 “启用自动修复” 复选框。

> **Important**
>
> 为了修复源代码存储库中的安全发现，AWS 安全代理可能会向您的存储库提交拉取请求。拉取请求可能对所有对仓库具有读取权限的用户可见。

配置 VPC 资源（可选）

如果您的目标域为私有域且托管在 VPC 中，请配置 AWS 安全代理应运行渗透测试的 VPC 设置。只有不可公开访问的应用程序才需要执行此步骤。

> **Note**
>
> 如果您的目标域名可以公开访问，请跳过此步骤。只有测试托管在 Amazon Virtual Private Cloud 中的私有应用程序时才需要 VPC 配置。

为渗透测试环境选择 VPC、子网和安全组。

1. 在 VPC 部分，找到 VPC ID 下拉列表。
2. 选择托管目标域的 VPC。

> **Important**
>
> 选定的 VPC 必须包含您在步骤 3 中指定的目标域。确保 VPC 具有适当的路由和网络配置，以允许 AWS 安全代理访问您的应用程序。

3. 找到 “子网” 下拉列表。
4. 选择一个或多个应在其中运行渗透测试的子网。

> **Note**
>
> 选择能够通过网络访问目标应用程序的子网。渗透测试将使用部署在这些子网中的资源执行。

5. 找到 “安全组” 下拉列表。
6. 选择控制渗透测试网络访问权限的安全组。

> **Important**
>
> 所选安全组必须允许您的目标域和任何可访问域的出站流量。确保安全组规则允许进行全面测试所需的网络访问权限。

配置身份验证凭证（可选）

如果您的目标域需要身份验证，请提供证书，以允许 AWS Security Agent 在渗透测试期间访问您的应用程序的受保护区域。只有需要用户身份验证的应用程序才需要执行此步骤。

> **Note**
>
> 如果您的目标域名不需要身份验证，或者您要测试的所有区域均可公开访问，请跳过此步骤。仅在需要 AWS 安全代理来测试应用程序中经过身份验证的部分时才配置证书。

添加 凭证

提供 AWS 安全代理用于访问您的应用程序的身份验证证书。

1. 在 “凭据 #1” 部分中，选择一种凭据输入方法：
   - 输入证书-将您的证书直接输入到 AWS 安全代理中。
   - 高级设置-对于敏感的凭证信息，请使用高级选项，例如 AWS Secrets Manager 或 AWS Lambda 函数。有关详细信息，请参阅the section called “为渗透测试提供身份验证凭证”。

     > **Tip**
     >
     > 对于生产环境或敏感证书，我们建议使用高级设置选项来安全地引用存储在 AWS Secrets Manager 或 Systems Manager Parameter Store 中的证书。

输入凭证详情

提供经过身份验证的帐户的用户名和密码。

1. 在“用户名”字段中，输入用于身份验证的用户名。
2. 在“密码”字段中，输入用于身份验证的密码。

> **Important**
>
> 确保您提供的凭证具有适合您要测试的区域的相应访问级别。凭证应代表典型用户的访问级别，而不是管理权限。

选择访问域

指定哪个目标域将使用这些凭据进行身份验证。

1. 在访问域下拉列表中，选择将使用这些凭据的域。

> **Note**
>
> 如果您有多个目标域需要不同的凭据，则可以在完成此凭据配置后单击“添加其他凭据”来添加其他凭据集。

配置代理登录提示（可选）

提供指导 AWS 安全代理完成应用程序的身份验证过程的说明。

1. 如果您的身份验证流程需要特定说明，请展开代理登录提示部分。
2. 输入说明如何在应用程序的登录流程中使用所提供的凭据。

> **Note**
>
> 代理登录提示告诉代理如何将您的凭据应用于您的应用程序。这对于复杂的身份验证流程、多步登录过程或具有非标准登录过程的应用程序非常有用。包括分步说明，例如“导航到/login，在'电子邮件'字段中输入用户名，输入密码，然后选择'登录'”。

添加多个凭据（可选）

如果您的应用程序需要多组凭据或不同的域需要单独的身份验证，请添加其他凭据集。

1. 完成第一个凭据配置后，选择添加其他凭据。
2. 为每个其他凭据集重复凭证配置步骤。
3. 要删除凭据集，请选择凭证标题旁边的删除。

> **Tip**
>
> 在应用程序中测试不同的用户角色、访问多个经过身份验证的域名或验证基于角色的访问控制时，请配置多个凭据。

附加其他资源（可选）

提供补充资源，帮助 AWS 安全代理进行更彻底、更准确的渗透测试。其他资源可能包括架构图、API 文档、配置文件、 GitHub 存储库或提供应用程序背景信息的 S3-hosted 材料。

> **Note**
>
> 其他资源是可选的，但建议使用。提供有关您的应用程序的全面信息有助于确保全面的测试覆盖率，减少误报，并提供更具可操作性的结果。

为渗透测试添加资源

选择现有资源或上传有助于指导渗透测试的新文件。

1. 在 “已连接的资源” 部分，您可以：
   - 选择 “从可用资源中选择”，从已连接到 AWS 安全代理的资源（例如 GitHub 存储库或 S3 存储桶）中进行选择。
   - 选择 “上传” 可直接从本地系统添加新文件。

> **Tip**
>
> 有用的资源包括 API 文档、架构图、 OpenAPI/Swagger 规范、配置文件、身份验证流程图以及描述应用程序结构和行为的任何其他材料。

从可用资源中选择

从已经与 AWS 安全代理集成的资源中进行选择。

1. 选择 “从现有资源中选择”。
2. 浏览来自互联来源的可用资源列表，例如：
   - GitHub 存储库，在 “GitHub 存储库” 选项卡下
   - S3 桶
   - 之前上传的文件
   - 文档存储库
3. 选择要包含在渗透测试中的资源。
4. 选择 “添加到渗透测试” 以附加所选资源。

Example

我们建议您选择相关 GitHub 存储库并将其添加到您的 pentest 中，这样 AWS Security Agent 就可以了解您的应用程序环境，并通过拉取请求（启用后）生成随时可以实施的代码修复

> **Note**
>
> 从可用来源中选择的资源与其原始位置保持同步。如果您更新 GitHub 存储库或 S3 文件，则渗透测试将使用更新的版本。

> **Note**
>
> 如果您有与 pentest 关联的私有 VPC 并配置了 GitHub 存储库，请确保可以通过您的私有 VPC 访问 GitHub 该私有 VPC 以提取 GitHub 资源。在大多数情况下，您需要确保默认允许通过 VPC NAT 网关的出站流量，或者配置特定规则以允许 GitHub IP 的出站流量（参见 GitHub Meta API 终端节点）

上传新资源

直接从本地系统上传文件或向 AWS 安全代理提供纯文本内容。

1. 选择上传。

2. 选择以下输入法之一：
   - 上传本地文件-从本地系统中选择一个或多个文件。
   - 粘贴纯文本-直接在输入字段中键入或粘贴文本内容。选择上传。
3. 然后选择 “添加” 以完成上传。
4. 上传的资源显示在 “已连接的资源” 表中。

> **Tip**
>
> 如果您想在不创建单独文件的情况下快速提供 API 端点列表、网址模式、测试说明或其他基于文本的信息，请使用纯文本选项。

> **Important**
>
> 确保上传的文件和粘贴的内容不包含敏感信息，例如生产凭证、私钥或个人身份信息 (PII)。使用经过消毒的配置文件和文档版本。

Connect 现有资源

现有资源可以来自您之前上传到 AWS Security Agent 的资源、您的 S3 存储桶以及您的集成 GitHub 存储库。选择 “从现有资源中选择” 以将其选中。

管理连接的资源

查看、整理和移除渗透测试附带的资源。

连接的资源表显示渗透测试中包含的所有资源以及以下信息：

- 名称-文件名或资源标识符
- 类型-资源类别（上传的文件、S3 资源、 GitHub 存储库等）

要管理资源，请执行以下操作：

1. 使用复选框选择一个或多个资源。
2. 选择 “从渗透测试中移除” 以分离所选资源。

> **Note**
>
> 您可以通过单击列标题按名称或类型对表格进行排序。这有助于在处理许多文件时整理资源。

创建渗透测试

完成并启动您的渗透测试配置。

配置完所有设置后，就可以开始创建渗透测试了。

1. 查看所有配置部分以确保准确性。
2. 请选择以下选项之一：
   - 选择 “创建渗透” 以保存配置，而无需立即运行。
   - 选择 “创建并执行” 以保存配置并立即开始渗透测试。
   - 选择 “取消” 以放弃渗透测试配置。

> **Important**
>
> 在运行渗透测试之前，请验证：
>
> - 所有目标域名均经过正确验证并可访问
> - IAM 角色具有适当的权限
> - Out-of-scope 已正确配置路径以防止测试破坏性操作
> - 您有权在所有目标域上执行安全测试

> **Note**
>
> 渗透测试开始后，您可以从 “渗透测试运行” 部分监控其进度。测试可能需要几个小时才能完成。大多数在 16 小时内完成，具体取决于您的应用程序的范围和复杂性。

查看渗透测试的结果

AWS 安全代理开始渗透测试后，在渗透测试日志页面上实时监控渗透测试的执行情况。AWS 安全代理记录渗透测试期间的所有操作。完成后，查看 pentest 摘要，其中包括应用程序概述、已确定端点的覆盖范围以及对安全发现的风险评估。

评估安全发现以解决应用程序漏洞。每项发现都包含影响评估、严重性评级、支持证据和补救拉取请求详细信息（启用自动代码修复时）。

先决条件

在开始之前，请确保您满足以下条件：

- 已完成或正在进行的渗透测试
- 访问 AWS 安全代理 Web 应用程序

第 1 步：访问渗透测试运行

导航到您的渗透测试运行以查看概述、日志和发现页面。

1. 登录 AWS 安全代理 Web 应用程序。
2. 导航到 “渗透测试” 部分。
3. 从列表中选择要检查的渗透测试运行。

> **Tip**
>
> 渗透测试详细信息页面显示测试状态、完成日期和发现的结果数量的摘要。

步骤 2：监控测试进度

使用步数指示器跟踪渗透测试运行的进度。

1. 在页眉下方找到水平步进指示器。
2. 查看每个测试阶段的状态：
   - 印前检查-初始设置和连接检查
   - 静态分析-代码和配置分析

- 渗透测试 — 运行时测试和漏洞扫描
- 定稿-最终验证和报告生成

> **Note**
>
> 每个步骤都显示一个状态指示器（“完成”、“进行中” 或 “待定”）。在整个测试过程中都会发现和验证结果，每个阶段完成时都会出现新的漏洞。

步骤 3：导航到渗透测试运行概述选项卡

1. “运行摘要” 部分提供了测试状态、持续时间和其他高级详细信息。它还提供了按严重级别和风险类型分类的安全发现的仪表板
2. AWS 安全代理提供的应用程序概述提供了渗透测试运行的摘要
3. AWS 安全代理发现的终端节点提供了在 pentest 运行期间由 AWS 安全代理发现和测试的所有终端节点的列表

步骤 4：导航到 “渗透测试日志” 选项卡

访问 AWS 安全代理在渗透测试期间执行的所有操作的详细日志。

1. 这些操作按操作类型和风险类型进行分类。
2. 选择特定操作以查看详细日志：
   - 测试摘要-代理操作和结果 High-level 摘要
   - 渗透测试日志-所有测试活动的详细日志

> **Note**
>
> 验证器操作提供日志，用于验证每个类别的调查结果

> **Note**
>
> “查找结果” 选项卡显示分割视图，左侧为查找结果列表，右侧为选定的查找结果详细信息。

步骤 5：导航到调查结果

列表中的每项发现都会显示关键信息，以帮助您快速评估其重要性。

> **Note**
>
> 默认置信度过滤器
>
> 默认情况下，您只能看到代理置信度高的结果。要同时显示代理置信度为中或低且误报的结果，请关闭隐藏未经验证的发现开关。

查看每张搜索卡上显示的信息：

- 查找名称-漏洞的标题和标识符
- 置信度徽章 — 表示客服人员对调查结果的置信度（高、中或低）
- 严重性标记-使用颜色编码显示风险级别：
  - 严重（红色）-需要立即采取行动；利用漏洞可能导致系统受损
  - 高（红色）-需要立即注意；利用漏洞可能会对安全造成重大影响
  - 中（橙色）— 应在合理的时间范围内解决；会增加整体安全风险
  - 低（黄色）— 可作为定期维护的一部分解决；将直接风险降至最低
  - 信息（蓝色）— 仅供参考；风险微乎其微，甚至没有即时风险
- 上次更新时间戳-显示上次修改或验证查找结果的时间
- 描述预览-漏洞的简要摘要

> **Important**
>
> 使用 “严重” 或 “高” 严重性标记和 “高可信度” 来确定发现的优先级，因为这些都是经过验证的漏洞，需要立即修复。

第 6 步：查看查找详情

选择单个发现结果以查看有关每个漏洞的全面信息。

1. 在左侧面板中选择查找结果名称以在右侧面板中显示其详细信息。

2. 查看验证状态：

> **Note**
>
> 如果发现显示未知 “AWS 安全代理尚未验证此发现”，则表示漏洞检测仍在确认中。这些发现可能需要手动验证。

3. 查看顶部显示的关键属性：
   - 代理信心 — AWS Security Agent 对这一发现的信心水平
   - 严重性-带有颜色编码徽章的风险等级
   - 查找日志-选择 “跟踪操作和日志” 以查看详细的执行日志和证据
   - 风险类型-安全风险的类别或类型（例如，绕过身份验证、SQL 注入）
4. 将 “描述” 部分展开，内容为：
   - 该漏洞的详细说明
   - 漏洞的工作原理
   - 为什么它会带来安全风险
   - 对您的应用程序的潜在影响
5. 展开 “风险推理” 部分以了解严重程度的计算：
   - CVSS（通用漏洞评分系统）指标明细
   - 攻击向量 (AV)-如何利用漏洞
   - 攻击复杂度 (AC)-漏洞利用的难度有多大
   - 需要权限 (PR)-需要什么访问级别
   - 用户交互 (UI)-是否需要用户操作
   - 范围 (S)-漏洞是否影响其他组件
   - 机密性、完整性和可用性影响
6. 展开 “重现步骤” 部分以查看：
   - 重现漏洞的详细技术步骤
   - 请求和响应示例
   - 触发问题的特定参数或条件
7. “验证脚本” 部分提供了一种重现发现结果的可执行方法。展开此部分（如果有）以查看：
   - 说明-如何设置和运行验证脚本
   - 环境变量-列出运行脚本之前必须配置的必需变量。为了安全起见，敏感值已被删除。

- 下载脚本-选择下载可执行文件验证脚本

> **Note**
>
> 验证脚本仅适用于已确认的漏洞。代理必须成功生成并验证了可执行的复制脚本。

> **Note**
>
> 验证脚本是使用生成式 AI 生成的。在执行之前请查看脚本，并且仅在您有权测试的系统上运行该脚本。有关负责任地测试 AI-generated 脚本的指南，请参阅 AWS 负责任的 AI 政策。

> **Tip**
>
> 验证脚本提供了一种可执行的方法来独立重现调查结果。使用您自己的凭据设置所需的环境变量，然后对目标系统运行脚本以确认漏洞存在。

> **Tip**
>
> 使用 “跟踪操作和日志” 链接访问完整的证据包，包括证明该漏洞的 HTTP 请求、响应和利用企图。

编辑调查结果

您可以编辑任何调查结果以更正其详细信息或完善代理的评估。以下字段是可编辑的：

- 名称-发现的标题
- 描述-安全漏洞的详细描述
- 状态-调查结果的当前状态
- RiskType — 已确定的安全风险类型
- RiskLevel — 严重性级别（严重、高、中、低、信息）
- RiskScore — 数字风险评分

- 推理 — 分配风险评分的理由
- AttackScript — 演示Proof-of-concept 漏洞的代码
- CustomerNot e — 可选备注，解释您进行编辑的理由

要编辑调查结果，请执行以下操作：

1. 导航到查找结果详情页面。
2. 选择编辑图标可修改前面列表中的任何字段。
3. 保存更改。

> **Note**
>
> 您的编辑内容会立即保存并反映在结果中。如果启用了 Findings 个性化，则您更改的所有可编辑字段都将用于优化代理的首选项，以备将来运行时使用。

查看原始代理版本

编辑查找结果时，版本选择器会出现在查找结果详细信息标题中。这允许您查看原始代理评估：

1. 在查找结果详细信息标题中找到版本选择器。
2. 选择 “原始” 以查看最初由代理制作的调查结果。
3. 选择 “最新” 以返回到应用编辑内容的当前版本。

查看个性化调查结果

启用调查结果个性化后，AWS 安全代理会从您对调查结果的编辑中学习。在将来的渗透测试运行中，代理会将这些偏好应用于类似的发现。根据您之前的编辑对查找结果进行调整后，详细信息面板会显示个性化更改部分。有关启用此功能的信息，请参阅启用或禁用查找结果个性化。

1. 在查找结果详细信息面板中找到 “个性化更改” 部分。

> **Note**
>
> 只有在 AWS Security Agent 与您之前的编辑保持一致的发现时，才会显示个性化更改部分。与任何学习偏好不匹配的发现将完全按照代理产生的结果进行显示，但不包括本节。

2. 查看更改内容和原因的摘要。摘要列出了每个调整后的属性及其原始代理生成的值、个性化值和推理。例如：

根据你之前的编辑，对最初由系统得出的结果进行了以下更改：风险等级从中更改为危急；风险评分从5.3更改为9.5；理由：已升级严重性，以反映应用程序源代码通过可公开访问的源地图的全部暴露情况。

1. 在决定如何采取行动之前，请查看摘要以了解调查结果是如何调整的。
2. 要覆盖个性化调查结果，请像编辑任何其他调查结果一样再次对其进行编辑（例如，更改严重性或风险评分）。您的最新编辑始终优先：AWS Security Agent 会从中学习，并在下次运行时应用更新的首选项，取代之前的规则。

> **Note**
>
> 个性化更改会调整风险等级、riskScore、姓名、描述、推理和 AttackScript 等查找属性，以反映 AWS 安全代理从您的编辑中学到的标准。在将来的运行中，您之前编辑过的任何字段都可能会根据类似的结果进行调整。

个性化如何从你的编辑中学习

启用调查结果个性化后，AWS Security Agent 会从您对调查结果进行的所有编辑中学习，并对未来的运行进行类似的调整。您编辑的任何字段（如前面的 “编辑调查结果” 部分所列）都将用于完善代理的学习偏好。

> **Note**
>
> 对于状态字段，只有将发现标记为 FALSE_POSITIVE 才会被视为可学习的偏好。更改其他状态值不会触发学习。

在下一次测试中，代理会根据你学到的偏好评估新发现，并在适用的情况下进行调整。任何经过调整的结果上都会显示 “个性化更改” 部分，解释了更改的内容。

> **Note**
>
> 个性化只能从最近完成的 pentest 运行中进行的编辑中学习。必须在 pentest 开始时启用该功能才能开始学习。如果您在以后的运行中编辑相同的结果，则以最近的编辑为准，代理会更新其首选项以反映您的最新决定。

启用或禁用调查结果个性化

默认情况下，“发现个性化” 处于启用状态。要禁用或重新启用它，请执行以下操作：

1. 导航到你的 pentest 配置。
2. 找到 “调查结果个性化” 开关。
3. 要启用调查结果个性化，请打开开关。要禁用，请将其关闭。

禁用后，搜索结果将以代理生成的原始状态显示，不应用任何个性化设置。之前学过的首选项会被保留，如果该功能重新启用，则会再次应用。

步骤 7：解释 CVSS 指标

了解 CVSS 指标有助于您评估真正的严重性并确定补救工作的优先顺序。

查看 “推理” 部分时，请注意以下关键指标：

- 攻击向量 (Network/Adjacent/Local/Physical)-表示可以远程执行攻击
- 攻击复杂度 (Low/High)-显示是否需要特殊条件才能利用攻击
- 需要权限 (None/Low/High)-确定攻击者需要的访问级别
- 用户互动 (None/Required)-确定漏洞利用是否需要用户参与
- Confidentiality/Integrity/Availability 影响 (None/Low/High)-衡量对系统安全性的影响

> **Important**
>
> 网络攻击向量、低复杂性和高 confidentiality/integrity 影响力的发现代表了需要立即关注的最危险的漏洞。

第 8 步：确定发现的优先顺序并解决问题

对发现的结果采取行动，以修复漏洞并改善应用程序的安全状况。

对于高度可信的严重性和高严重性发现：

1. 查看 “描述” 和 “步骤”，彻底重现章节。
2. 通过 “跟踪操作和日志” 链接访问详细日志，以收集完整的证据。
3. 通过以下方法之一访问随时可以实现的代码修复：
   - 对于自动修复：使用修复部分中的拉取请求链接
   - 对于手动请求：在调查结果页面上选择 “修正代码” 以请求拉取请求先决条件：
   - 管理员必须在 AWS 安全代理控制台中为 GitHub 存储库启用代码修复
   - 存储库必须包含在您的 pentest 配置中
4. 计划进行后续渗透测试，以验证修复是否有效。

对于严重程度为 “中” 和 “低” 的发现：

1. 根据您的风险承受能力和业务环境确定优先级。
2. 在您的常规开发冲刺计划中包括补救任务。
3. 考虑多个低严重程度的发现是否会带来更高的风险。
4. 记录任何可接受的风险，并提供适当的理由。

> **Important**
>
> 不要忽视低严重性的发现。通常可以将多个低严重性漏洞链接在一起以造成更严重的漏洞，尤其是在与社交工程或物理访问结合使用时。

步骤 9：跟踪修复进度

使用调查结果界面跟踪哪些漏洞已得到解决以及哪些漏洞需要采取进一步措施。

1. 在进行修复时，请参阅 “重现步骤” 部分以验证您的修复。
2. 记录每项发现的补救方法，以备将来参考和合规审计。

> **Tip**
>
> 维护一份修复日志，将每项发现与其解决方案对应起来，包括代码更改、配置更新或为解决漏洞而做出的架构决策。

后续步骤

查看您的渗透测试结果后：

- 高度自信地确定关键和高严重性发现的优先顺序，以便立即进行补救
- 在测试环境中下载、查看和运行验证脚本，以测试脚本并识别漏洞
- 在问题管理系统中创建跟踪单，其中包含查找详细信息和证据的链接
- 实施修复和安全控制措施以解决已发现的漏洞
- 监控渗透测试运行进度指示器中是否有新发现的漏洞
- 安排后续渗透测试，以验证漏洞是否已得到适当修复
- 根据发现更新您的应用程序安全测试流程和威胁模型
- 查看 CVSS 指标以了解应用程序的整体安全状况

有关执行渗透测试的更多信息，请参阅the section called “创建渗透测试”。

有关了解安全客户端生命周期的更多信息，请参阅the section called “了解资源层次结构和生命周期”。

为渗透测试提供身份验证凭证

提供证书，使 AWS 安全代理能够测试您的 Web 应用程序中经过身份验证的区域。如果没有证书，代理只能测试可公开访问的页面和 API。

配置身份验证凭证

1. 在渗透测试创建工作流程中，找到 “身份验证凭证-可选” 部分。
2. 在 “凭证 #1” 部分，选择您的凭证输入方法：
   - 输入凭据-直接输入凭据。最适合开发和测试环境。
   - 高级设置-使用 AWS-native 凭据管理。建议用于生产环境和敏感凭证。

高级选项

如果您选择高级设置，则可以从三种凭据策略中进行选择：

- IAM 角色假设-适用于使用 AWS Cognito 或 IAM 身份验证的应用程序
- AWS Secrets Manager-通过加密和轮换实现安全的证书存储
- Lambda 函数-用于动态证书生成或复杂的身份验证流程

直接输入凭证

1. 选择 “输入凭据”。
2. 输入用户名和密码。
3. 在 “访问 URL” 下拉列表中，选择将使用这些凭据的 URL。必须从目标端点列表中选择。
4. （可选）在 2FA-可选字段中，为需要双重身份验证的应用程序提供 TOTP 密钥。您可以：
   - 直接输入 TOTP 密钥（例如`JBSWY3DPEHPK3PXP`），或输入完整的 `otpauth://totp/` URI（例如`otpauth://totp/Example:user@example.com?secret=JBSWY3DPEHPK3PXP&issuer=Example`）。
   - 选择上传图标，从您的身份验证器应用程序设置页面上传二维码图像。在本地扫描二维码，并自动提取 TOTP URI。

     当提供 TOTP 密钥时，代理会自动生成新的一次性验证码，并在登录期间检测到 2FA 提示时输入这些代码。
5. （可选）如果您的应用程序具有复杂的身份验证流程，请展开 Agent Space 登录提示以提供特定的登录说明。

> **Important**
>
> 使用具有代表访问权限的测试账户，而不是个人或管理账户。

使用高级设置

当您选择高级凭证策略（Secrets Manager、Lambda 或 IAM 角色）时，AWS 安全代理会使用服务角色直接从已配置的 AWS 资源中检索您的证书。使用 Agent Space 登录提示向代理提供有关如何将这些凭据应用于应用程序的说明，例如，要导航到哪个登录 URL 以及要填写哪些表单字段。

> **Note**
>
> Secrets Manager 密钥和 Lambda 函数必须与您的 AWS 安全代理设置位于同一 AWS 账户中。 Cross-account 目前不支持证书。

1. 选择高级设置。
2. 在用户访问策略下拉列表中，选择以下选项之一：

选择可供代理担任的可用的 IAM 角色

对于使用 AWS Cognito、带有 IAM 身份验证的 API Gateway 或其他 AWS-native 身份验证系统的应用程序，请使用此选项。IAM 角色必须具有允许 AWS Security Agent 担任该角色的信任关系以及访问应用程序身份验证系统的权限。

从连接的 AWS Secrets Manager 中选择静态证书

使用此选项通过加密、轮换和访问审计，安全地从 AWS Secrets Manager 检索证书。

IAM 角色必须具有`secretsmanager:GetSecretValue`和`secretsmanager:DescribeSecret`权限。

代理直接从 Secrets Manager 中检索密钥值。使用 Agent Space 登录提示告诉代理如何将这些凭据应用于应用程序的登录流程，例如，要导航到哪个 URL 以及要填写哪些表单字段。您可以使用任何格式来存储您的密钥，因为代理将使用您在登录提示中提供的说明来解释该格式。

例如，如果代理要在上提交 username/password 登录表单https://example.com/login，则可以将您的密钥格式化为 JSON，其中包含`username`和`password`字段。如果应用程序需要 TOTP-based 2FA，请添加一个直接包含 TOTP 密钥或完整 `otpauth://totp/` URI 的`totpSecret`字段：

```
{
  "username": "test-user",
  "password": "secure-password-here",
  "totpSecret": "JBSWY3DPEHPK3PXP"
}
```

然后，配置身份验证指令：。将访问 URL 设置为`https://example.com`（或从目标端点列表中选择的任何其他 URL）。在 Agent Space 登录提示中输入以下内容：“导航到https://example.com/login表单中提供的用户名和密码并将其输入表单。”

再举一个例子，如果您要在 HTTP 标头中提供 API 密钥，则可以将其存储为纯文本：

```
"api-key-here"
```

然后，配置身份验证指令：。在 Agent Space 登录提示中输入以下内容：“将 X-API-Key 标题设置为所有请求提供的 API 密钥。”

> **Important**
>
> 仅支持 TOTP-based 2FA。不支持短信、电子邮件、推送通知、硬件密钥和 OAuth 身份验证。

选择可用的 Lambda 函数以动态检索证书

此选项用于复杂的身份验证系统、动态证书生成或与外部身份提供商的集成。

IAM 角色必须具有`lambda:InvokeFunction`权限，并且该功能必须在 30 秒内完成。

渗透测试运行时，AWS 安全代理会同步调用您的 AWS Lambda 函数，并通过一个识别渗透测试和正在解析的特定凭证的事件：

```
{
  "pentest_arn": "arn:aws:securityagent:us-west-2:111122223333:pentest/pt-
abcd1234-5678-90ab-cdef-EXAMPLE11111",
  "actor_identifier": "Credential1"
}
```

- `pentest_arn`— 正在解析证书的渗透测试的 Amazon 资源名称 (ARN)。使用它来确定函数中请求的范围、授权或审计。
- `actor_identifier`— 您在控制台中为此凭据分配的凭据名称（例如）。`Credential1`当单个函数提供多个凭证时，使用它来返回正确的凭证。

代理直接使用函数的输出作为凭证。使用 Agent Space 登录提示告诉代理如何将这些证书应用于您的应用程序，就像使用 AWS Secrets Manager 一样。the section called “从连接的 AWS Secrets Manager 中选择静态证书”有关如何格式化 Lambda 函数输出和支持的身份验证类型的示例，请参阅。

配置多个凭证

要测试不同的用户角色或身份验证系统，请执行以下操作：

1. 选择添加其他凭据。
2. 使用任一输入法配置其他凭据。
3. 要删除凭据，请在凭据部分选择删除。

登录优化

登录优化允许 AWS Security Agent 从之前的 pentest 运行中学习导航模式，并将其应用于后续运行。这些模式包括登录流程和多步身份验证工作流程。这减少了代理重新发现如何进行身份验证所花费的时间，从而实现更快的扫描速度和更一致的测试覆盖范围。

登录优化的工作原理

在第一次运行 pentest 时，代理会发现如何使用您提供的凭据浏览应用程序的登录流程。它会记录成功的导航步骤，并生成捕获所学登录工作流程的技能文件。

在随后的运行中，代理会读取保存的技能文件并直接应用学到的导航模式，跳过发现阶段。这意味着：

- 更快地进行经过身份验证的测试 — 代理可以立即应用经过验证的导航模式，而不是从头开始探索
- 行为更加一致——代理遵循同样的行之有效的路径，而不是探索替代方案

> **Note**
>
> 登录优化在运行中的第一个登录会话中进行学习。同一次运行中的后续登录会话将受益于在第一次会话中学到的知识。在 future 运行时，所有登录会话都将受益于保存的技能。

启用或禁用登录优化

默认情况下，登录优化处于启用状态。要禁用或重新启用它，请执行以下操作：

1. 在 pentest 配置中，导航到 “身份验证凭据-可选” 部分。
2. 找到 “登录优化” 开关。
3. 要启用登录优化，请打开开关。要禁用，请将其关闭。

禁用后，代理会在每次运行时从头开始重新发现登录流程。之前学过的导航技能会被保留，如果该功能重新启用，则会再次应用。

> **Tip**
>
> 如果您的应用程序的登录流程发生了显著变化（例如重新设计的登录页面或新的身份验证步骤），则代理会通过在下次运行时更新其所学技能来自动进行调整。您无需手动重置优化。

修复渗透测试发现

在查看渗透测试的结果时，您可以请求 AWS 安全代理尝试修复发现。对于私有 GitHub 存储库，AWS 安全代理会打开包含建议修复程序的拉取请求。对于公共存储库，补救措施以可下载的差异文件形式提供，您可以在本地应用该文件。

您必须在 AWS 管理控制台中启用查找补救措施。（请参见 the section called “使用户能够开始修复渗透测试和代码审查结果”。）用户可以从 AWS 安全代理 Web 应用程序开始补救特定发现。

先决条件

在开始之前，请确保您满足以下条件：

- 已完成或正在进行的渗透测试
- 访问 AWS 安全代理 Web 应用程序
- 熟悉应用程序的架构和安全要求

步骤 1：启用或禁用自动修复

在创建或修改渗透测试时，可以配置代码修复选项。如果您启用自动修复，AWS Security Agent 将在渗透测试期间确认发现后自动尝试修复关联的 GitHub 存储库。您也可以手动启动代码修复。在编辑渗透测试详细信息的视图中，在 “自动代码修复” 部分，启用或禁用代码修复。

步骤 2：选择用于代码修复的存储库

1. 选择 “下一步” 一直到最后一步其他学习资源。
2. 选择 “从资源中选择”。
3. 选择GitHub 存储库。
4. 选择要进行代码修复的存储库。
5. 保存渗透测试。
6. 您可以在渗透测试学习资源选项卡下看到成功关联的存储库。

第 3 步：开始渗透测试并查看结果

运行渗透测试以检测发现。有关更多信息，请参阅 the section called “查看渗透测试的结果”。

步骤 4：开始并查看代码修复

1. 导航到调查结果。
2. 如果您启用了自动代码修复，则在 AWS 安全代理确认发现后，就会开始代码修复。
3. 如果要手动启动代码修复，请选择修复代码按钮。
4. 在调查结果的 “代码修复” 部分，您可以查看代码修复状态和拉取请求链接。如果 GitHub 存储库是公开的，则代码修复可以作为可下载文件而不是拉取请求提供。您可以运行将更改应用`git apply /path/to/code_remediation_changes.diff`到本地存储库。

安全和参考

AWS 安全代理的安全指南、服务配额和文档历史记录。

AWS 安全代理中的安全

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。责任共担模式将其描述为云的 安全性和云中 的安全性：

- 云安全 — 负责保护 AWS 在 AWS 云中运行 AWS 安全代理的基础设施。这包括服务基础架构、AI 模型和渗透测试代理。 Third-party 作为AWS 合规计划的一部分，审计师定期测试和验证我们安全的有效性。
- 云中的安全性 – 您的责任包括以下各个方面：
  - 通过 IAM 策略和权限管理对 AWS 安全代理的访问权限
  - 保护您提供给服务的内容，包括设计文档、代码存储库和用于渗透测试的应用程序 URL
  - 配置监控哪些存储库和应用程序
  - 审查服务提供的安全调查结果并采取行动
  - 根据提供的补救指南保护您的应用程序
  - 您的数据的敏感性、您公司的要求以及适用的法律法规

本文档可帮助您了解在使用 AWS 安全代理时如何应用分担责任模型。

AWS 安全代理和 AI 辅助渗透测试的安全注意事项

AWS Security Agent 是一款前沿代理，可在整个开发生命周期中主动保护您的所有环境中的应用程序。它根据您的要求进行定制的自动安全审查，安全团队集中定义标准，这些标准将在审查期间自动验证。Security Agent 执行根据您的应用程序定制的按需渗透测试，发现并报告经过验证的安全风险。这种方法可以扩展应用程序中的安全专业知识，以匹配开发速度，同时提供全面的安全保障。通过整合从设计到部署的安全性，它有助于尽早大规模预防漏洞。

安全团队只需在 AWS 控制台中即可定义组织安全要求：经批准的授权库、日志标准和数据访问策略。AWS Security Agent 会在整个开发过程中自动执行这些安全要求，根据您的标准评估架构文档和代码，并在发现违规行为时提供具体指导。这样可以跨团队实现一致的安全执行，并根据开发速度调整审查规模。

为了进行部署验证，AWS Security Agent 将渗透测试从周期性的瓶颈转变为按需功能。安全团队提供目标 URL、身份验证详情、源代码和文档。AWS Security Agent 可以深入了解应用程序，并执行复杂的攻击链来发现和验证漏洞，使团队能够在需要时进行测试。

关键功能

AWS 安全代理提供涵盖整个开发生命周期的全面安全功能。

设计安全审查

AWS Security Agent 按需提供有关设计文档的安全反馈，并在编写代码之前评估是否符合组织安全要求。安全团队通过 Web 应用程序上传设计文档，在该应用程序中，代理会根据您的安全要求对其进行分析，并根据补救指南显示发现的结果。这可以将长达数小时的人工审查加速为有针对性的分析，使团队能够在补救最有效的时候解决安全问题。

代码安全审查

AWS Security Agent 会分析拉取请求或上传的代码，以了解组织安全要求和常见安全问题，例如缺少输入验证和 SQL 注入风险。代理直接在您的代码存储库平台中提供补救指导。安全团队配置要监控哪些存储库，在所有代码库中扩展评估范围，同时保持对关键问题的监督。

On-demand 渗透测试

AWS Security Agent 提供按需渗透测试，通过量身定制的多步攻击场景发现并报告经过验证的安全漏洞。AWS Security Agent 部署专门的 AI 代理，这些代理根据提供的文档和证书开发应用程序环境，然后执行复杂的攻击链来识别传统工具遗漏的复杂漏洞。它通过影响分析、可重现的攻击路径和随时可实施的代码修复来记录调查结果，将渗透测试从数周缩短到几个小时，并将验证扩展到整个应用程序组合。

常见问题解答

安全&控制

AWS 安全代理如何进行身份验证并维护对系统的访问权限？

渗透测试是 AWS 安全代理中唯一可以在运行时向用户系统进行身份验证的功能。在开始笔试之前，AWS 安全代理接受静态用户名和密码证书（存储在 Secrets Manager 中）或证书供应商（作为 Lambda 函数）形式的证书作为配置。这些凭证用于在笔试的整个生命周期中行使用户的 system/application 正常功能。我们鼓励用户创建具有适当范围权限的新证书，以便进行渗透测试。

用户能否控制测试的范围和深度以防止意外对系统造成影响？

AWS 安全代理允许客户选择要在终端节点中探索的特定漏洞类别。用户可以指定超出范围的 URL，以防止 AWS 安全代理对这些目标执行渗透测试。 https://docs.aws.amazon.com/securityagent/latest/userguide/perform-penetration-test.html

AWS 安全代理本身会构成安全风险吗？

AWS Security Agent 被指示发现安全风险，但要故意使用影响最小的有效负载（例如提取 SQL 版本而不是在发现 SQL 注入攻击时删除表）来发现安全风险。AWS Security Agent 还仅限于确定性护栏，以防止风险行为，例如对目标应用程序造成过大负载。虽然有防护措施，但仍可能存在无意或非明显的业务逻辑交互，因此，我们始终建议在预生产环境中进行渗透测试。

AWS 安全代理收集哪些数据，这些数据存储在哪里？

AWS Security Agent 允许用户上传项目，以提供有关其正在测试的应用程序的背景信息。有关数据保护的更多信息，请参阅the section called “数据保护”。AWS Security Agent 将自动选择您所在地理区域内的最佳区域来处理您的推理请求。这样可以最大限度地提高可用计算资源和模型可用性，并提供最佳的客户体验。您的数据将仅存储在请求发起的区域，但是，输入提示和输出结果可能会在该区域之外进行处理。所有数据都将通过 Amazon 的安全网络进行加密传输。有关更多信息，请参阅跨区域推理。

有哪些控制措施可以阻止针对端点的未经授权的测试？

指定为渗透测试的目标 URL 的端点将需要 DNS 验证或 HTTP 验证作为所有权的衡量标准。AWS 安全代理将要求客户向终端节点的 DNS 中添加 TXT 记录或公开返回验证字符串的 HTTP 路由作为所有权证明。只有在出示所有权证明后，用户才能继续进行笔试。网络将阻止向目标以外的 URL 和可访问的 URL 发出的请求。

客户有责任确保他们获得适当的授权，可以测试所有可能受其渗透测试活动影响的系统。AWS 安全代理的所有使用都必须遵守 AWS 可接受使用政策 (https://aws.amazon.com/aup/)。

用户如何使用 AWS 安全代理阻止和举报任何滥用行为？

AWS 安全代理会持续监控访问目标网址之外的 URL 的请求和尝试次数。如果发现滥用行为，例如试图使用 AWS Security Agent 对第三方终端节点进行未经授权的测试，则该账户中正在进行的渗透测试都将被终止。客户可以联系 AWS Support 或他们的 AWS 账户团队寻求帮助。

AWS 安全代理能否取代笔试工作流程？

AWS 安全代理不是专业的渗透测试服务，我们鼓励用户将 AWS 安全代理集成到他们的安全审查工作流程中。AWS Security Agent 可以在软件生命周期的开发阶段提供按需渗透测试的可访问性，因为与

渗透测试专业人员合作可能为时过早、不切实际或需要过于频繁地重新评估。安全专业人员可以查看 AWS Security Agent 的发现，对其进行验证、解释或扩展以获取新的发现（如果存在）。

用户能否为不同的团队成员设置基于角色的访问控制 (RBAC)？

可以。AWS Security Agent 与 AWS IAM 身份中心集成，允许管理员管理可以访问 AWS 安全代理 Web 应用程序的团队成员，该应用程序允许用户创建、管理和查看设计审查和笔试。

测试能力

AWS 安全代理可以检测哪些类型的漏洞？

AWS 安全代理可检测 OWASP 十大网络应用程序中的漏洞。AWS Security Agent 提供了特定的风险类型，您可以在下面概述的测试中包括或排除这些风险。发现可能出现在这些风险类别中，也可能来自通过跟踪这些风险类别组合的线索而发现的新发现。

- 任意文件上传
  - Artenty File Upload 确认应用程序应该能够抵御虚假和恶意文件，从而保护应用程序和用户的安全
- 代码注入
  - 代码注入是攻击类型的总称，攻击类型包括注入代码，然后 interpreted/executed 由应用程序注入代码
- 命令注入
  - 命令注入是一种攻击，其目标是通过易受攻击的应用程序在主机操作系统上执行任意命令
- Cross-Site 脚本编写 (XSS)
  - Cross-Site 脚本 (XSS) 攻击是一种注入，在这种注入中，恶意脚本会被注入原本良性且值得信赖的网站
- 不安全的直接对象引用
  - 当应用程序根据用户提供的输入提供对对象的直接访问时，就会出现不安全的直接对象引用 (IDOR)
- JSON 网络令牌漏洞
  - JWT 是常见的漏洞来源，无论是在应用程序中的实现方式还是在底层库中
- 本地文件包含
  - 文件包含漏洞允许攻击者添加文件，通常利用目标应用程序中实现的 “动态文件包含” 机制
- 路径遍历
  - 路径遍历攻击（也称为目录遍历）旨在访问存储在 Web 根文件夹之外的文件和目录

- 权限升级
  - 当用户访问的资源或功能超出正常允许的范围时，就会发生权限升级，而应用程序本应阻止此类提升或更改
- Server-Side 请求伪造 (SSRF)
  - Server-Side 当攻击者可以滥用服务器上的功能来读取或更新内部资源时，就会发生请求伪造 (SSRF)
- Server-Side 模板注入
  - 当用户输入以不安全的方式嵌入到模板中并导致在服务器上远程执行代码时，就会出现服务器端模板注入漏洞 (SSI)
- SQL 注入
  - SQL 注入攻击包括通过从客户端到应用程序的输入数据插入或 “注入” SQL 查询
- XML 外部实体
  - XML 外部实体攻击是一种针对解析 XML 输入的应用程序的攻击。当包含对外部实体引用的 XML 输入由配置较弱的 XML 解析器处理时，就会发生此攻击

AWS 安全代理支持哪些身份验证方法？

AWS 安全代理支持常见的身份验证方法，包括 OAuth 和 JWT。有关详情，请参阅 文档。

AWS 安全代理如何处理速率限制和拒绝服务 (DOS) 防护？

AWS Security Agent 有防护栏，可防止其中断或关闭正在测试的终端节点，包括 DOS。它具有内部速度控制功能，可以检测和处理意想不到的交通模式。

AWS 安全代理能否同时测试 REST 和 GraphQL API？

是的，AWS 安全代理可以测试 API 终端节点。我们鼓励客户提供 API 文档作为额外学习资源，让 AWS Security Agent 能够更好地了解每个 API 的形状和功能。

用户如何验证 AWS 安全代理是否已覆盖所有关键应用程序逻辑和终端节点？

AWS Security Agent 将对目标应用程序进行广度优先的探索，并在尝试任何漏洞利用之前尝试正常使用它。这使它能够在运行时建立对应用程序的工作理解，并发现关键的应用程序逻辑和端点。鉴于其随机性，AWS Security Agent 不能保证能够发现和测试任何目标应用程序的所有关键应用程序和终端节点。AWS Security Agent Web 应用程序可以查看所有发现的终端节点以及在渗透测试日志中采取的操作。

精度&可靠性

AWS 安全代理如何在报告之前验证调查结果？

AWS 安全代理使用确定性验证器来帮助验证报告的发现。在无法使用确定性验证器的风险类型中，AWS Security Agent 将独立重播发现步骤，以获得对发现有效性的信心。默认情况下，AWS Security Agent 仅报告高或中等可信度调查结果，并隐藏未经验证的发现。

AWS 安全代理能否适应自定义应用程序逻辑？

AWS Security Agent 可以选择接受源代码、威胁模型、设计文档和 API 文档作为其他学习资源，以获取用户指导的关于渗透测试生命周期中使用的目标应用程序的背景信息。

用户能否在执行之前查看 AWS 安全代理测试方法？

目前无法预览 AWS 安全代理的操作流程。AWS Security Agent 计划本质上是动态的，它基于对目标应用程序的探索。客户可以通过观察渗透测试日志，实时监控 AWS Security Agent 的探索过程。如果日志显示无效或不受欢迎的轨迹，客户可以停止正在进行的 pentest 运行。

集成&部署

AWS 安全代理是否与安全工具（SIEM、漏洞管理）或 CI/CD 管道集成？

AWS 安全代理不与任何现有的安全工具或 CI/CD 管道集成。

AWS 安全代理如何处理特定于环境的配置？

AWS Security Agent 可以配置为使用特定的 IAM 角色、在 VPC 内运行，使用客户指定的应用程序相关证书，并使用 Github 源存储库作为目标应用程序的源代码引用。

AWS 安全代理能否在气隙或隔离的环境中运行？

可以将 AWS 安全代理配置为连接到 VPC，包括无法访问出站互联网的 VPC。

多个团队成员可以同时运行测试吗？

AWS Security Agent 支持每个账户同时运行 5 次 pentest，与谁开始测试无关。客户最多可以创建 100 个代理空间和 1,000 个 Pentest 项目。

运营影响

对测试系统的性能有什么影响？

AWS Security Agent 有防护栏，可防止其中断或关闭正在测试的终端节点。这包括对 AWS 安全代理可以向终端节点发出的调用次数进行速度控制。由于笔试活动，系统或被测端点的流量应该会有所增加，并且可能会触发监控警报。我们建议只在预生产环境中运行 AWS 安全代理或任何笔试活动。

用户能否安排或限制 AWS 安全代理？

AWS Security Agent 没有公共 API，也无法安排笔试运行。在开始笔试运行时，AWS Security Agent 也不对向目标终端节点发出的请求提供并发控制。如果 AWS Security Agent 导致目标终端节点出现问题，客户可以停止正在进行的渗透测试。

完整的安全评估通常需要多长时间？

每个 pentest 的运行时间取决于目标应用程序的广度以及配置为要评估的风险类型。大多数渗透测试会在 16 小时内完成。

适用于 AWS 安全代理的 AWS 托管策略

AWS 托管式策略是由 AWS 创建和管理的独立策略。AWS 托管策略旨在为许多常见用例提供权限，以便您可以开始向用户、群组和角色分配权限。

请记住，AWS 托管策略可能不会为您的特定用例授予最低权限权限，因为它们可供所有 AWS 客户使用。我们建议通过定义特定于使用案例的客户管理型策略来进一步减少权限。

您无法更改 AWS 托管策略中定义的权限。如果 AWS 更新了 AWS 托管策略中定义的权限，则更新会影响该策略所关联的所有委托人身份（用户、群组和角色）。当新的 AWS 服务启动或现有服务可以使用新的 API 操作时，AWS 最有可能更新 AWS 托管策略。

有关更多信息，请参阅 IA M 用户指南中的 A WS 托管策略。

AWS 托管策略： SecurityAgentWebAppAPIPolicy

授予与安全客户端 Web 应用程序 API 交互的权限。此策略使用户能够配置和执行自动安全渗透测试、管理测试执行、查看安全发现以及访问安全代理资源。此策略引用了旧版代理实例资源类型和特定的传统 IAM 操作。

权限详细信息

此策略授予与安全测试控制服务 (securityagent: *) 进行交互的权限，以便：

- Pentest 管理：创建、更新、删除和列出渗透测试及其执行任务
- 安全发现：查看、描述和更新已完成测试的安全发现，包括相关内容和元数据。
- 任务管理：列出并检索代码审查和文档审查任务
- 资源发现：列出并查看代理实例、构件、集成和发现的端点
- 测试执行：使用实时监控功能启动和停止 pentest 执行
- 代码修复：针对安全发现启动自动代码修复

要查看最新版本的 JSON 策略文档，请参阅 SecurityAgentWebAppAPIPolicyAWS 托管策略参考指南。

AWS 托管策略： AWSSecurityAgentWebAppPolicy

授予与安全客户端 Web 应用程序 API 交互的权限。此策略使用户能够配置和执行自动安全渗透测试、管理测试执行、查看安全发现以及访问安全代理资源。

权限详细信息

此策略授予与安全测试控制服务 (securityagent: *) 进行交互的权限，以便：

- Pentest 管理：创建、更新、删除和列出渗透测试及其作业
- 安全发现：查看、描述和更新已完成测试的安全发现，包括相关内容和元数据。
- 任务管理：列出并检索代码审查和设计审查任务
- 资源发现：列出并查看代理空间、工件、集成和发现的端点
- 测试执行：使用实时监控功能启动和停止 pentest 作业
- 代码修复：针对安全发现启动自动代码修复

要查看最新版本的 JSON 策略文档，请参阅 AWSSecurityAgentWebAppPolicyAWS 托管策略参考指南。

AWS 安全代理更新了 AWS 托管策略

查看自该服务开始跟踪这些更改以来，AWS 安全代理的 AWS 托管策略更新的详细信息。

要接收此特定文档页面的所有源文件更改通知，您可以通过 RSS 阅读器订阅以下 URL：

更改	描述	日期
已将权限添加至 AWSSecurityAgentWebAppPolicy。	为新DesignReviewFeedback 资源类型添加TargetDomain 了资源权限。	2026年3月31日
添加了新的托管策略 AWSSecurityAgentWebAppPolicy。	AWSSecurityAgentWebAppPolicy 为新 AgentSpace 资源类型和 IAM 操作名称更改添加了托管策略。	2026年2月9日
已将权限添加至 SecurityAgentWebAppAPIPolicy。	已添加securityagent:StartCodeRemediation 以允许用户针对安全发现启动自动代码修复。	2026 年 1 月 20 日
已将权限添加至 SecurityAgentWebAppAPIPolicy。	添加securityagent:BatchGetSecurityTestContentMetadata 允许用户在控制台中查看图像。	2025 年 12 月 5 日
AWS 安全代理开始跟踪更改。	AWS 安全代理开始跟踪其 AWS 托管策略的更改。	2025 年 12 月 2 日

AWS 安全代理中的数据保护

AWS 分担责任模式适用于 AWS 安全代理中的数据保护。如本模型所述，AWS 负责保护运行所有 AWS 云的全球基础设施。您负责维护对托管在此基础结构上的内容的控制。您还负责所使用的 AWS 服务的安全配置和管理任务。有关欧洲数据保护的信息，请参阅 AWS 安全博客上的 AWS 责任共担模型和 GDPR 博客文章。出于数据保护目的，我们建议您保护 AWS 账户证书，并使用 AWS IAM 身份中心或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证（MFA）。
- 用于 SSL/TLS 与 AWS 资源进行通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。

- 使用 AWS 设置 API 和用户活动日志 CloudTrail。有关使用 CloudTrail 跟踪捕获 AWS 活动的信息，请参阅 A WS CloudTrail 用户指南中的使用跟 CloudTrail 踪。
- 使用 AWS 加密解决方案以及 AWS 服务中的所有默认安全控制。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在通过命令行界面或 API 访问 AWS 时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 终端节点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准（FIPS）第 140-3 版》https://aws.amazon.com/compliance/fips/。

强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括您使用控制台、API、AWS CLI 或 AWS 软件开发工具包使用 AWS 安全代理或其他 AWS 服务时。在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供 URL，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

静态加密

默认情况下，AWS 安全代理使用加密密钥对所有静态数据进行 AWS-managed 加密。这包括：

- 设计文档和代码-您为安全审查提供的所有设计文档、代码存储库和应用程序工件均使用加密进行 AES-256 加密。
- 安全调查结果-所有安全发现、漏洞报告和补救建议均经过静态加密。
- 配置数据-对安全要求、自定义策略和服务配置进行加密。
- 审核日志-所有服务活动日志和审计跟踪都经过加密。

AWS 安全代理使用 AWS 密钥管理服务 (AWS KMS) 来管理加密密钥。您可以选择使用客户管理的密钥来加密数据，从而完全控制保护资源的加密密钥。有关更多信息，请参阅 the section called “客户自主管理型密钥”。

传输中加密

AWS 安全代理使用传输层安全 (TLS) 1.2 或更高版本对传输中的所有数据进行加密。这适用于：

- API 通信 — 您的应用程序与 AWS 安全代理之间的所有 API 调用都使用带有 TLS 加密的 HTTPS。
- 控制台访问 — 通过 HTTPS 访问 AWS 安全代理控制台。
- 存储库连接-与其他代码存储库的连接使用加密协议。 GitHub

- 代理通信 — AWS 安全代理服务和渗透测试代理之间的所有通信都使用加密通道。

密钥管理

AWS 安全代理使用 AWS 密钥管理服务 (AWS KMS) 来管理加密密钥。默认情况下，使用 AWS-managed 密钥对数据进行加密。在创建代理空间和集成等资源时，您可以选择指定客户托管密钥。有关更多信息，请参阅 the section called “客户自主管理型密钥”。

互联网络流量隐私

AWS 安全代理使用公共互联网与云托管的源代码控制提供商（GitHub、、 GitLab Bitbucket）和 Confluence Cloud 通信。

对于自托管提供商（ GitHub 企业服务器）GitLab Self-Managed，您可以使用 Amazon VPC Lattice 配置私有连接，将所有流量保持在 AWS 网络内。有关更多信息，请参阅 the section called “Connect 连接到私有托管的源代码管理”。

在默认配置中，AWS 安全代理使用公共互联网访问您的应用程序进行渗透测试。您可以选择将渗透测试配置为使用 VPC 来访问您的应用程序。有关更多信息，请参阅 the section called “将代理连接到私有 VPC 资源”。

Cross-Region 数据处理

AWS Security Agent 使用跨区域推理来优化可用的计算资源和模型可用性。根据请求来源的区域，我们可能会在不同的区域处理输入提示并输出结果。

- 在美国东部（弗吉尼亚北部）— us-east-1、美国西部（俄勒冈）— us-west-2、亚太地区（悉尼）— ap-southeast-2、亚太地区（东京）— ap-northeast-1、欧洲（法兰克福）和欧洲（爱尔兰）— eu-central-1eu-west-1，AWS Security Agent 使用地理跨区域推理。对于大多数要素，数据处理仍位于发出请求的地理边界（例如美国、欧盟、澳大利亚或日本）内。对于代码补救，来自澳大利亚和日本的请求将在欧盟处理。有关特定功能的路由详细信息，请参阅跨区域推理表。
- 在亚太地区（孟买）ap-south-1、亚太地区（新加坡ap-southeast-1）和南美洲（圣保罗）sa-east-1，AWS 安全代理使用全球跨区域推理。我们可能会在任何商业 AWS 区域处理输入提示和输出结果。

在所有情况下，您的数据都仅存储在请求发出的地区。跨区域操作期间传输的所有数据都保留在 AWS 网络上，不会通过公共互联网。我们对 AWS 区域之间传输的数据进行加密。

Cross-Region 推理始终处于启用状态，无法选择退出。有关哪些区域处理每个功能的请求的详细信息，请参阅中的 “跨区域推理” 部分。the section called “安全最佳实践”

数据删除

当您从 AWS 安全代理中删除数据时：

- 数据已标记为删除，无法再通过该服务进行访问。
- 数据将在 30 天内从所有 AWS 安全代理系统中删除。

删除您的数据

1. 在 AWS 控制台中，导航到 AWS 安全代理。
2. 选择要删除的数据（安全审查、调查结果或自定义要求）。
3. 选择删除，然后确认删除。

AWS 安全代理的身份和访问管理

AWS Identity and Access Management (IAM) AWS 服务 可以帮助管理员安全地控制对 AWS 资源的访问权限。 IAM 管理员控制谁可以通过身份验证（登录）和授权（拥有权限）使用 AWS 安全代理资源。 IAM 无需支付额外费用即可使用。 AWS 服务

受众

您使用 AWS Identity and Access Management (IAM) 的方式会有所不同，具体取决于您在 AWS 安全代理中所做的工作。

服务用户 — 如果您使用 AWS Security Agent 服务完成工作，则您的管理员会为您提供所需的证书和权限。当您使用更多 AWS 安全代理功能来完成工作时，您可能需要额外的权限。了解如何管理访问权限有助于您向管理员请求适合的权限。

如果您无法访问 AWS 安全代理中的某项功能，请参阅the section called “对 AWS 安全代理的身份和访问进行故障排除”。

服务管理员-如果您负责公司的 AWS 安全代理资源，则可能拥有对 AWS 安全代理的完全访问权限。您的工作是确定您的服务用户应访问哪些 AWS 安全代理功能和资源。然后，您必须向 IAM 管理员提交更改服务用户权限的请求。查看此页面上的信息以了解的基本概念 IAM。要详细了解贵公司如何使用 AW IAM S 安全代理，请参阅the section called “AWS 安全代理的工作原理 IAM”。

IAM 管理员-如果您是 IAM 管理员，则可能需要详细了解如何编写策略来管理 AWS 安全代理的访问权限。要查看可在中使用的 AWS 安全代理基于身份的策略示例 IAM，请参阅。the section called “AWS 安全代理基于身份的策略示例”

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 AWS 账户根用户身份进行身份验证（登录 AWS）IAM 用户，或者通过担任 IAM 角色进行身份验证。AWS 建议使用 IAM 角色，避免直接使用根用户。

您可以使用通过身份源提供的凭据以 AWS 联合身份登录。 AWS IAM Identity Center (IAM Identity Center) 用户、贵公司的单点登录身份验证以及您的 Google 或 Facebook 凭据就是联合身份的示例。当您以联合身份登录时，您的管理员之前使用 IAM 角色设置了联合身份。当你使用联合访问 AWS 时，你就是在间接扮演一个角色。

根据您的用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录的更多信息 AWS，请参阅 AWS Sign-In 用户指南中的如何登录您的 AWS 账户。

如果您 AWS 以编程方式访问，则会 AWS 提供软件开发套件 (SDK) 和命令行接口 (CLI)，以便使用您的凭据对请求进行加密签名。如果您不使用 AWS 工具，则必须自己签署请求。有关使用推荐的方法自行签署请求的更多信息，请参阅 AWS 一般参考中的签名版本 4 签名流程。

无论使用何种身份验证方法，您可能还需要提供其它安全信息。例如， AWS 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅 Multi-factor AWS IAM 身份中心（AWS 单一的继任者 Sign-On）用户指南中的身份验证和 IA M 用户指南中的 AWS 中使用多重身份验证 (MFA)。

亚马逊云科技账户根用户

首次创建时 AWS 账户，您首先需要一个单一登录身份，该身份可以完全访问账户中的所有资源 AWS 服务 和资源。此身份称为 AWS 账户根用户，使用您创建账户时使用的电子邮件地址和密码登录即可访问。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关需要您以 root 用户身份登录的任务的完整列表，请参阅《账户管理参考指南》中的 “需要根用户凭证的任务”。

联合身份

作为最佳实践，要求人类用户（包括需要管理员访问权限的用户）使用与身份提供商的联合身份验证 AWS 服务 通过临时证书进行访问。

联合身份是指您的企业用户目录、Web 身份提供商、Identity Center 目录中的用户，或者任何使用 AWS 服务 通过身份源提供的凭据进行访问的用户。 AWS Directory Service当联合身份访问时 AWS 账户，他们将扮演角色，角色提供临时证书。

要集中管理访问权限，建议您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中创建用户和群组，也可以连接并同步到您自己的身份源中的一组用户和群组，以便在您的所有 AWS 账户 和应用程序中使用。有关 IAM 身份中心的信息，请参阅什么是 IAM 身份中心？ 在 AWS IAM 身份中心（AWS 单一身份中心的继任者 Sign-On）用户指南中。

IAM 用户 和群组

IAM 用户是指您内部 AWS 账户 对个人或应用程序具有特定权限的身份。在可能的情况下，我们建议使用临时证书，而不是创建 IAM 用户 谁拥有长期证书，例如密码和访问密钥。但是，如果您有需要长期凭证的特定用例 IAM 用户，我们建议您轮换访问密钥。有关更多信息，请参阅《IAM 用户指南》中的对于需要长期凭证的使用案例，应在需要时更新访问密钥。

IAM 群组是一种标识，用于指定集合 IAM 用户。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可以拥有一个名为 iamAdmins 的群组，并授予该群组管理资源的权限。 IAM

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅 IAM 用户指南中的何时创建 IAM 用户 （而不是角色）。

IAM 角色

IAM 角色是您内部具有特定权限 AWS 账户 的身份。它类似于 IAM 用户，但与特定的人无关。您可以 AWS Management Console 通过切换 IAM 角色在中临时扮演角色。您可以通过调用 AWS CLI 或 AWS API 操作或使用自定义 URL 来代入角色。有关使用角色的方法的更多信息，请参阅 IAM 用户指南中的使用 IAM 角色。

IAM 具有临时证书的角色在以下情况下很有用：

- 联合用户访问：要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关联合身份验证的角色的信息，请参阅《IAM 用户指南》中的为第三方身份提供者创建角色。如果您使用 IAM Identity Center，则需要配置权限集。为控制您的身份在进行身份验证后可以访问的内容，IAM Identity Center 会将权限集与IAM中的角色相关联。有关权限集的信息，请参阅 AWS IAM 身份中心（AWS Single 的继任者 Sign-On）用户指南中的权限集。
- 临时 IAM 用户 权限- IAM 用户 可以代入一个 IAM 角色来临时获得特定任务的不同权限。

- Cross-account 访问权限 — 您可以使用 IAM 角色允许其他账户中的某人（受信任的委托人）访问您账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些资源 AWS 服务，您可以将策略直接附加到资源（而不是使用角色作为代理）。要了解角色和基于资源的跨账户访问策略之间的区别，请参阅 IAM 用户指南中的 IAM 角色与基于资源的策略的区别。
- Cross-service 访问权限 — 有些 AWS 服务 使用其他功能 AWS 服务。例如，当您在服务中进行调用时，该服务通常会在其中运行应用程序 Amazon EC2 或在其中存储对象 Amazon S3。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
  - 委托人权限-当您使用 IAM 用户 或角色在中执行操作时 AWS，您被视为委托人。策略向主体授予权限。使用某些服务时，您可能会执行一个操作，此操作然后在不同服务中触发另一个操作。在这种情况下，您必须具有执行这两个操作的权限。
  - 服务角色-服务 IAM 角色是服务代替您执行操作的角色。 IAM 管理员可以从内部创建、修改和删除服务角色 IAM。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。
  - Service-linked 角色 — 服务相关角色是一种链接到的服务角色。 AWS 服务该服务可以代替您执行操作。 Service-linked 角色出现在您的， AWS 账户 并且归服务所有。 IAM 管理员可以查看但不能编辑服务相关角色的权限。
- 上运行的应用程序 Amazon EC2 -您可以使用 IAM 角色管理在 Amazon EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时证书。这比在 Amazon EC2 实例中存储访问密钥更可取。要为 Amazon EC2 实例分配 AWS 角色并使其可供其所有应用程序使用，您需要创建一个附加到该实例的实例配置文件。实例配置文件包含角色并允许在 Amazon EC2 实例上运行的程序获得临时证书。有关更多信息，请参阅 IAM 用户指南中的使用 IAM 角色向在 Amazon EC2 实例上运行的应用程序授予权限。

要了解是否使用 IAM 角色，请参阅 IAM 用户指南中的何时创建 IAM 角色（而不是用户）。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略是其中的一个对象 AWS ，当与身份或资源关联时，它会定义其权限。 AWS 在委托人（用户、root 用户或角色会话）发出请求时评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的结构和内容的更多信息，请参阅 IAM 用户指南中的 JSON 策略概览。

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

每个 IAM 实体（用户或角色）一开始都没有权限。默认情况下，用户无法执行任何操作，甚至无法更改自己的密码。要为用户授予执行某些操作的权限，管理员必须将权限策略附加到用户。或者，管理员

可以将用户添加到具有预期权限的组中。当管理员向群组授予权限时，该群组中的所有用户都被授予这些权限。

IAM 无论您使用何种方法执行操作，策略都会定义该操作的权限。例如，假设您有一个允许 `iam:GetRole` 操作的策略。拥有该策略的用户可以从 AWS Management Console AWS CLI、或 AWS API 获取角色信息。

Identity-based 政策

Identity-based 策略是可以附加到身份（例如 IAM 用户、角色或群组）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅 IAM 用户 IAM 指南中的创建策略。

Identity-based 策略可以进一步归类为内联策略或托管策略。内联策略直接嵌入单个用户、组或角色中。托管策略是独立的策略，您可以将其附加到中的多个用户、群组和角色 AWS 账户。托管策略包括 AWS 托管策略和客户托管策略。要了解如何在托管式策略和内联策略之间进行选择，请参阅 IAM 用户指南中的在托管式策略与内联策略之间进行选择。

Resource-based 政策

Resource-based 策略是您附加到资源（例如 Amazon S3 存储桶）的 JSON 策略文档。服务管理员可以使用这些策略来定义指定的委托人（账户成员、用户或角色）可以在什么条件下对该资源执行哪些操作。 Resource-based 策略是内联策略。没有基于托管资源的策略。

访问控制列表（ACL）

访问控制列表（ACL）是一种策略类型，用于控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，但它们不使用 JSON 策略文档格式。 Amazon S3 AWS WAF、和 Amazon VPC 是支持 ACL 的服务示例。要了解有关 ACL 的更多信息，请参阅《Amazon Simple Storage Service 开发人员指南》中的访问控制列表（ACL）概述。

其他策略类型

AWS 支持其他不太常见的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- 权限边界-权限边界是一项高级功能，您可以在其中设置基于身份的策略可以向 IAM 实体（IAM 用户 或角色）授予的最大权限。您可为实体设置权限边界。由此产生的权限是实体的基于身份的策略及其权限边界的交集。 Resource-based 在`Principal`字段中指定用户或角色的策略不受权限边界的限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅 IAM 用户指南中的IAM 实体的权限边界。

- 服务控制策略 (SCP)-SCP 是 JSON 策略，用于指定组织或组织单位 (OU) 的最大权限。 AWS Organizations AWS Organizations 是一项用于对您的企业拥有的多 AWS 账户 项进行分组和集中管理的服务。如果在组织内启用了所有特征，则可对任意或全部账户应用服务控制策略（SCP）。SCP 限制成员账户中实体的权限，包括每个亚马逊云科技账户根用户。有关组织和 SCP 的更多信息，请参阅 AWS Organi zations 用户指南中的 SCP 的工作原理。
- 会话策略：会话策略是当您以编程方式为角色或联合用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅 IAM 用户指南中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的策略评估逻辑。

AWS 安全代理的工作原理 IAM

在使用 IAM 管理 AWS 安全代理访问权限之前，请先了解 AWS 安全代理有哪些 IAM 功能可供使用。

IAM 功能	AWS 安全代理支持
the section called “Identity-based AWS 安全代理的策略”	是
the section called “Resource-based AWS 安全代理中的策略”	否
the section called “AWS 安全代理的策略操作”	是
the section called “AWS 安全代理的策略资源”	部分
the section called “AWS 安全代理的策略条件密钥”	是
the section called “AWS 安全代理中的访问控制列表 (ACL)”	否
the section called “Attribute-based 使用 AWS 安全代理进行访问控制 (ABAC)”	否

IAM 功能	AWS 安全代理支持
the section called “在 AWS 安全代理中使用临时证书”	是
the section called “AWS 安全代理的转发访问会话”	是
the section called “AWS 安全代理的服务角色”	否
the section called “Service-linked AWS 安全代理的角色”	是

要全面了解 AWS Security Agent 和其他人是如何 AWS 服务 使用的 IAMAWS 服务 ，请参阅 IAM 用户指南 IAM中的使用方法。

Identity-based AWS 安全代理的策略

支持基于身份的策略：是

Identity-based 策略是您可以附加到身份（例如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源，以及允许或拒绝操作的条件。您无法在基于身份的策略中指定主体，因为它适用于其附加的用户或角色。要了解您在 JSON 策略中使用的所有元素，请参阅 IAM 用户指南中的 IAM JSON 策略元素参考。

Identity-based AWS 安全代理的策略示例

要查看 AWS 安全代理基于身份的策略示例，请参阅。the section called “AWS 安全代理基于身份的策略示例”

Resource-based AWS 安全代理中的策略

支持基于资源的策略：否

Resource-based 策略是您附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什

么条件下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户访问，您可以将整个账户或其它账户中的 IAM 实体指定为基于资源的策略中的主体。将跨账户主体添加到基于资源的策略只是建立信任关系工作的一半而已。当委托人和资源位于不同的 AWS 账户中时，可信账户中的 IAM 管理员还必须向委托人实体（用户或角色）授予访问资源的权限。他们通过将基于身份的策略附加到实体以授予权限。但是，如果基于资源的策略向同一个账户中的主体授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅 IAM 用户指南中的跨账户在 IAM 中访问资源。

AWS 安全代理的策略操作

支持动作是

管理员可以使用亚马逊云科技 JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

IAM 基于身份的策略的`Action`元素描述了该策略允许或拒绝的一个或多个具体操作。策略操作通常与关联的 AWS API 操作同名。此策略用于策略中以授予执行关联操作的权限。

AWS 安全代理中的策略操作在操作前使用以下前缀：`securityagent:`。例如，要授予某人使用 AWS Security Agent `CreateEnvironment` API 操作创建环境的权限，您需要将该`securityagent:CreateEnvironment`操作包含在他们的策略中。策略语句必须包含 `Action` 或 `NotAction` 元素。AWS Security Agent 定义了自己的一组操作，这些操作描述了您可以使用该服务执行的任务。

要在单个语句中指定多项操作，请使用逗号将它们隔开，如下所示：

```
"Action": [
      "securityagent:action1",
      "securityagent:action2"
```

您也可以使用通配符（*）指定多个操作。例如，要指定以单词 `List` 开头的所有操作，包括以下操作：

```
"Action": "securityagent:List*"
```

AWS 安全代理的策略资源

支持策略资源：部分

管理员可以使用亚马逊云科技 JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 Resource 或 NotResource 元素。作为最佳实践，请使用其 Amazon 资源名称（ARN）指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符 (*) 指示语句应用于所有资源。

```
"Resource": "*"
```

某些 AWS 安全代理 API 操作支持多种资源。例如，在调用 ListEnvironments API 操作时可以引用多个环境。要在单个语句中指定多个资源，请使用逗号分隔 ARN。

```
"Resource": [
      "EXAMPLE-RESOURCE-1",
      "EXAMPLE-RESOURCE-2"
```

例如，AWS 安全代理环境资源具有以下 ARN：

```
arn:${Partition}:securityagent:${Region}:${Account}:environment/${EnvironmentId}
```

要my-environment-2在您的语句中指定环境my-environment-1，请使用以下示例 ARN：

```
"Resource": [
        "arn:aws:securityagent:us-east-1:123456789012:environment/my-environment-1",
        "arn:aws:securityagent:us-east-1:123456789012:environment/my-environment-2"
```

要指定属于特定账户的所有环境，请使用通配符 (*)：

```
"Resource": "arn:aws:securityagent:us-east-1:123456789012:environment/*"
```

AWS 安全代理的策略条件密钥

支持特定于服务的策略条件键：是

管理员可以使用亚马逊云科技 JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Condition元素（或Condition块）允许您指定语句生效的条件。Condition 元素是可选的。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 AWS 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则使用逻辑OR运算来 AWS 评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有当资源标有资源 IAM 用户 名称时，您才能授予访问该资源的 IAM 用户 权限。有关更多信息，请参阅 IAM 用户指南中的IAM 策略元素：变量和标签。

AWS 安全代理定义了自己的条件密钥集，还支持使用一些全局条件键。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的AWS 全局条件上下文密钥。

AWS 安全代理中的访问控制列表 (ACL)

支持 ACL：否

访问控制列表（ACL）控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，但它们不使用 JSON 策略文档格式。

Attribute-based 使用 AWS 安全代理进行访问控制 (ABAC)

支持 ABAC（策略中的标签）：否

在 AWS 安全代理中使用临时证书

支持临时凭证：是

当您使用临时证书登录时，某些 AWS 服务不起作用。有关更多信息，包括哪些 AWS 服务使用临时证书，请参阅 IAM 用户指南中与 IAM 配合使用的 AWS 服务。

如果您使用除用户名和密码之外的任何方法登录 AWS 管理控制台，则使用的是临时证书。例如，当您使用公司的单点登录 (SSO) 链接访问 AWS 时，该过程会自动创建临时证书。当您以用户身份登录控制台，然后切换角色时，您还会自动创建临时凭证。有关切换角色的更多信息，请参阅《IAM 用户指南》中的从用户切换到 IAM 角色（控制台）。

您可以使用 AWS CLI 或 AWS API 手动创建临时证书。然后，您可以使用这些临时证书访问 AWS。AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅 IAM 中的临时安全凭证。

AWS 安全代理的转发访问会话

支持转发访问会话（FAS）：是

当您使用 IAM 用户或角色在 AWS 中执行操作时，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用 AWS 服务的委托人的权限以及

请求的 AWS 服务向下游服务发出请求。只有当服务收到需要与其他 AWS 服务或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两项操作的权限。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。

AWS 安全代理的服务角色

支持服务角色：否

服务角色是由一项服务担任、代表您执行操作的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅 I A M 用户指南中的创建角色以向 AWS 服务委派权限。

Service-linked AWS 安全代理的角色

支持服务关联角色：是

服务相关角色是一种与 AWS 服务关联的服务角色。该服务可以代替您执行操作。 Service-linked 角色出现在您的 AWS 账户中并归服务所有。IAM 管理员可以查看但不能编辑服务关联角色的权限。

AWS 安全代理基于身份的策略示例

默认情况下 IAM 用户 ，角色无权创建或修改 AWS 安全代理资源。他们也无法使用 AWS Management Console AWS CLI、或 AWS API 执行任务。 IAM 管理员必须创建 IAM 策略，授予用户和角色对其所需的指定资源执行特定 API 操作的权限。然后，管理员必须将这些策略附加到需要这些权限的 IAM 用户 或群组。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅 IAM 用户指南中的使用 JSON 编辑器创建策略。

策略最佳实践

Identity-based 策略决定是否有人可以在您的账户中创建、访问或删除 AWS 安全代理资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的 AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅 IA M 用户指南中的工作职能托AWS 管策略或 AWS 托管策略。
- 应用最低权限权限-使用 IAM 策略设置权限时，仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限权限。有关使用应用权限 IAM 的更多信息，请参阅 IAM 用户指南 IAM中的策略和权限。

- 使用 IAM 策略中的条件进一步限制访问权限-您可以在策略中添加条件以限制对操作和资源的访问权限。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 CloudFormation。有关更多信息，请参阅 IAM 用户指南中的 IAM JSON 策略元素：条件。
- 用于 IAM Access Analyzer 验证您的 IAM 策略以确保权限的安全性和功能性 — IAM Access Analyzer 验证新的和现有的策略，使策略符合 IAM 策略语言 (JSON) 和 IAM 最佳实践。 IAM Access Analyzer 提供 100 多项政策检查和切实可行的建议，以帮助您制定安全和实用的策略。有关更多信息，请参阅 IAM 用户指南中的IAM Access Analyzer 策略验证。
- 需要多重身份验证 (MFA)-如果您的账户中有 IAM 用户 需要root用户的情况，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅 I AM 用户指南中的配置 MFA-protected API 访问权限。

使用 AWS 安全代理控制台

要访问 AWS 安全代理控制台，IAM 委托人必须拥有一组最低权限。这些权限必须允许委托人列出和查看您的 AWS 安全代理资源的详细信息 AWS 账户。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的主体，控制台将无法按预期正常运行。

为确保您的 IAM 委托人仍然可以使用 AWS 安全代理控制台，请使用您自己的唯一名称创建策略。将策略附加到主体。有关更多信息，请参阅 IAM 用户指南中的为用户添加权限：

有关政策的详细信息，请参阅the section called “AWS 托管策略： SecurityAgentWebAppAPIPolicy”。

对于仅调用 AWS CLI 或 AWS API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与您尝试执行的 API 操作相匹配的操作。

对 AWS 安全代理的身份和访问进行故障排除

使用以下信息来帮助您诊断和修复在使用 AWS Security Agent 时可能遇到的常见问题，以及 IAM。

AccessDeniedException

如果您`AccessDeniedException`在调用 AWS API 操作时收到，则表示您正在使用的 IAM 委托人证书没有进行该调用所需的权限。

```
An error occurred (AccessDeniedException) when calling the CreateEnvironment operation:
User: arn:aws:iam::111122223333:user/user_name is not authorized to perform:
securityagent:CreateEnvironment on resource:
 arn:aws:securityagent:region:111122223333:environment/my-env
```

在前面的示例消息中，用户无权调用 AWS 安全代理 `CreateEnvironment` API 操作。要向 IAM 委托人提供 AWS 安全代理管理员权限，请参阅the section called “AWS 安全代理基于身份的策略示例”。

有关 IAM 的更多一般信息，请参阅 IA M 用户指南中的使用策略控制 AWS 资源的访问权限。

我想允许我以外的人进入 AWS 账户 访问我的 AWS 安全代理资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以代入角色。对于支持基于资源的策略或访问控制列表（ACL）的服务，您可以使用这些策略向人员授予对您的资源的访问权。

要了解更多信息，请参阅以下内容：

- 要了解 AWS 安全代理是否支持这些功能，请参阅the section called “AWS 安全代理的工作原理 IAM”。
- 要了解如何提供对您拥有的资源的访问权限，请参阅 IAM 用户指南 IAM 用户 中的提供 AWS 账户对您拥有的其他资源的访问权限。 AWS 账户
- 要了解如何向第三方提供对您的资源的访问权限 AWS 账户，请参阅 IAM 用户指南中的向第三方提供访问权限。 AWS 账户
- 要了解如何通过联合身份验证提供访问权限，请参阅 IAM 用户指南中的向经过外部身份验证的用户提供访问权限（联合身份验证）。
- 要了解使用角色和基于资源的策略进行跨账户访问的区别，请参阅 IAM 用户指南中的 IAM 角色与基于资源的策略的区别。

事件响应

AWS Security Agent 是一项主动式安全测试服务，旨在在漏洞被利用之前识别和防范漏洞。该服务侧重于通过设计审查、代码分析和渗透测试进行预防性安全验证，而不是被动的事件检测或响应。

事件检测

AWS 安全代理不提供事件检测功能。该服务作为一种主动安全测试工具运行，可在开发期间和部署之前验证应用程序是否存在漏洞。要进行运行时安全监控和事件检测，请使用亚马逊 GuardDuty、AWS Security Hub 或亚马逊等服务 CloudWatch。

事件警报

AWS 安全代理不会针对安全事件生成实时警报。该服务在完成设计审查、代码分析或渗透测试后，通过 AWS 控制台提供安全发现。这些发现代表的是测试期间发现的潜在漏洞，而不是活跃的安全事件。

事故补救

AWS 安全代理不提供自动事件补救。该服务可识别安全漏洞并提供补救指导，包括：

- 已识别漏洞的详细描述
- 经过验证的发现的可重现漏洞利用路径
- 具体的代码修复和实现指南
- 对发现的问题进行影响分析

开发和安全团队使用本指南在漏洞进入生产环境之前对其进行手动修复。

支持事件响应活动

虽然 AWS 安全代理不是为事件响应而设计的，但安全团队可以使用该服务来支持事后活动：

漏洞验证

安全事件发生后，使用 AWS 安全代理测试其他应用程序或环境中是否存在类似漏洞。

安全态势评估

进行渗透测试，以验证作为事故补救一部分实施的安全改进。

根本原因分析

使用代码安全审查功能来识别其他代码库中可能存在的类似漏洞。

AWS 安全代理的合规性验证

要了解某项 AWS 服务是否在特定合规计划的范围内，请参阅按合规计划划分的 AWS 服务，然后选择您感兴趣的合规计划。有关一般信息，请参阅 AWS 合规性计划。

您可以使用 AWS Artifact 下载第三方审计报告。有关更多信息，请参阅在 AWS Artifact 中下载报告。

您在使用 AWS 服务时的合规责任取决于您的数据敏感度、贵公司的合规目标以及适用的法律和法规。亚马逊云科技 提供以下资源来帮助实现合规性：

- Security Compliance & Governance：这些解决方案实施指南讨论了架构考虑因素，并提供了部署安全性和合规性功能的步骤。
- 符合 HIPAA 要求的服务参考：列出符合 HIPAA 要求的服务。并非所有 AWS 服务都符合 HIPAA 资格。

- AWS 合规资源 — 这套工作手册和指南可能适用于您的行业和所在地区。
- AWS 客户合规指南 — 从合规角度了解责任共担模式。这些指南总结了保护 AWS 服务的最佳实践，并将指南与跨多个框架（包括美国国家标准与技术研究院 (NIST)、支付卡行业安全标准委员会 (PCI) 和国际标准化组织 (ISO)）的安全控制对应起来。
- 使用 AWS Config 开发人员指南中的规则评估资源 — AWS Config 服务评估您的资源配置在多大程度上符合内部实践、行业指南和法规。
- AWS Security Hub — 这项 AWS 服务可全面了解您在 AWS 中的安全状态。Security Hub 使用安全控制来评估您的 AWS 资源，并检查您是否符合安全行业标准和最佳实践。有关受支持服务及控制措施的列表，请参阅 Security Hub 控制措施参考。
- A@@ mazon GuardDuty — 此 AWS 服务通过监控您的环境中是否存在可疑和恶意活动，来检测对您的 AWS 账户、工作负载、容器和数据的潜在威胁。 GuardDuty 通过满足某些合规性框架规定的入侵检测要求，可以帮助您满足各种合规性要求，例如 PCI DSS。
- AWS Audit Manager — 这项 AWS 服务可帮助您持续审计 AWS 的使用情况，从而简化风险管理以及遵守法规和行业标准的方式。

AWS 安全代理的弹性

> **Note**
>
> AWS Security Agent 可在以下区域使用：美国东部（弗吉尼亚北部）— us-east-1 us-west-2、美国西部（俄勒冈）— ap-south-1、亚太地区（孟买）—、亚太地区（新加坡）— ap-southeast-1、亚太地区（悉尼）— ap-southeast-2、亚太地区（东京）— ap-northeast-1、欧洲（法兰克福）— eu-central-1、欧洲（爱尔兰）— eu-west-1 和南美洲（圣保罗）— sa-east-1。它使用跨区域推理。在亚太地区（孟买）、亚太地区（新加坡）和南美洲（圣保罗），AWS 安全代理使用全球跨区域推理，将推理请求路由到任何商业 AWS 区域。在所有其他区域，它使用地理跨区域推理，将数据处理保持在特定的地理边界内。AWS Security Agent 是应用程序开发过程中使用的工具，不应作为关键基础设施或面向客户的基础设施进行部署。

亚马逊云科技 全球基础设施围绕亚马逊云科技区域和可用区构建。亚马逊云科技区域提供多个在物理上独立且隔离的可用区，这些可用区通过延迟低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错能力和可扩展性。

有关 AWS 区域和可用区的更多信息，请参阅 AWS 全球基础设施。

AWS 安全代理中的基础设施安全

作为一项托管服务，AWS 安全代理受 AWS 全球网络安全的保护。有关 AWS 安全服务以及 AWS 如何保护基础设施的信息，请参阅 AWS 云安全。要使用基础设施安全最佳实践设计您的 AWS 环境，请参阅 AWS Well-Architected 框架中的安全。

网络隔离

AWS 安全代理是一项完全托管的服务，可通过 AWS 控制台和 AWS 安全代理 Web 应用程序进行访问。该服务的访问权限通过 AWS Identity and Access Management (IAM) 或 AWS IAM 身份中心进行控制，后者可以与您的身份提供商集成。

AWS 安全代理支持由 AWS 提供支持的接口 VPC 终端节点 PrivateLink，使您无需通过公共互联网即可从 VPC 私密访问服务 API。有关更多信息，请参阅 the section called “接口 VPC 端点”。

AWS Security Agent 需要互联网访问才能对目标应用程序执行渗透测试和控制平面操作。该服务使用 AWS-managed 基础设施进行测试，不在您的账户中配置 EC2 实例或其他计算资源。如果您配置 VPC 访问以进行渗透测试，则安全代理会在您的子网中创建一个 ENI，但此 ENI 没有公有 IP 地址。有关更多信息，请参阅 the section called “将代理连接到私有 VPC 资源”。

Multi-tenancy 和资源隔离

AWS 安全代理是一项多租户服务。安全审查、调查结果和客户数据与各个 AWS 账户隔离，并进行静态加密。AWS 采用标准的基础设施隔离控制，确保一个客户的安全测试活动不会影响另一个客户的性能或机密性。

AWS 安全代理和接口 VPC 终端节点 (AWS PrivateLink)

您可以使用 AWS PrivateLink 在您的 VPC 和 AWS 安全代理之间创建私有连接。您可以像访问您的 VPC 一样访问安全代理，无需使用互联网网关、NAT 设备、VPN 连接或 Direct Connect 连接。您的 VPC 中的实例不需要公有 IP 地址即可访问安全代理。

您可以通过创建由 AWS PrivateLink 提供支持的接口端点来建立此私有连接。我们将在您为接口端点启用的每个子网中创建一个端点网络接口。这些是请求者管理的网络接口，用作发往安全客户端的流量的入口点。

有关更多信息，请参阅 AWS PrivateLink 指南 AWS PrivateLink中的通过访问 AWS 服务。

安全代理的注意事项

在为安全客户端设置接口端点之前，请查看 AWS PrivateLink 指南中的注意事项。

安全代理支持从您的 VPC 调用其所有 API 操作，包括管理代理空间、渗透测试和安全发现的操作。

在创建终端节点时，您可以选择 IPv4、IPv6 或双堆栈。

安全客户端支持 VPC 终端节点策略。默认情况下，允许通过接口端点对安全客户端进行完全访问。您可以通过将终端节点策略附加到接口终端节点或将安全组与端点网络接口关联来控制访问权限。

对安全代理的所有 API 调用均使用 TLS 1.2 或更高版本进行加密，包括通过 VPC 终端节点进行的调用。有关数据保护的更多信息，请参阅 the section called “数据保护”。安全客户端 API 调用已登录 AWS CloudTrail。有关更多信息，请参阅 the section called “示例： AWS 安全客户端日志文件条目”。

为安全客户端创建接口终端节点

您可以使用 Amazon VPC 控制台或 AWS 命令行界面 (AWS CLI) 为安全代理创建接口终端节点。有关更多信息，请参阅 AWS PrivateLink 指南中的创建接口终端节点。

使用以下服务名称为安全客户端创建接口终端节点：

```
com.amazonaws.<region>.securityagent
```

要使用 AWS CLI 创建接口终端节点，请运行以下命令。将区域、VPC ID、子网 ID 和安全组 ID 替换为您自己的值。

```
aws ec2 create-vpc-endpoint \
    --vpc-id vpc-1a2b3c4d \
    --vpc-endpoint-type Interface \
    --service-name com.amazonaws.<region>.securityagent \
    --subnet-ids subnet-1a2b3c4d subnet-5e6f7a8b \
    --security-group-ids sg-1a2b3c4d \
    --private-dns-enabled
```

> **Important**
>
> 必须为接口终端节点启用私有 DNS。安全客户端要求您使用默认的区域 DNS 名称（例如`securityagent.us-east-1.api.aws`）通过终端节点发出 API 请求。不支持直接调用端点特定的 VPCE URL。

要使用私有 DNS，您必须将您的 VPC `true` 的`enableDnsSupport`和`enableDnsHostnames`属性都设置为。有关更多信息，请参阅《Amazon VPC 用户指南》中的 VPC 的 DNS 属性。

为接口终端节点配置安全组

创建接口终端节点时，可以将安全组与端点网络接口关联以控制安全代理的流量。创建一个安全组，允许来自您的 VPC 中需要与安全代理通信的资源的入站 HTTPS 流量（端口 443）。

安全组应包含以下入站规则：

- 类型：HTTPS
- 协议：TCP
- 端口范围：443
- 来源：您的 VPC CIDR 块（例如`10.0.0.0/16`）或需要访问安全客户端的资源的安全组 ID

有关更多信息，请参阅 AWS PrivateLink 指南中的安全组。

为接口端点创建端点策略

端点策略是一种 IAM 资源，您可以将其附加到接口端点。默认端点策略允许通过接口端点对安全客户端进行完全访问权限。要控制允许从您的 VPC 访问安全代理的权限，请将自定义终端节点策略附加到接口终端节点。

端点策略指定以下信息：

- 可以执行操作的委托人（AWS 账户、IAM 用户和 IAM 角色）。
- 可执行的操作。
- 可对其执行操作的资源。

有关更多信息，请参阅 AWS PrivateLink 指南中的使用终端节点策略控制对服务的访问权限。

示例： AWS 安全代理操作的 VPC 终端节点策略

以下是 AWS 安全客户端的端点策略示例。当连接到终端节点时，此策略向所有资源的所有委托人授予访问列出的 AWS 安全客户端操作的权限。

```
{
   "Statement": [
      {
         "Effect": "Allow",
```

```
         "Principal": "*",
         "Action": [
            "securityagent:CreatePentest",
            "securityagent:ListPentests",
            "securityagent:BatchGetPentests",
            "securityagent:StartPentestExecution",
            "securityagent:StopPentestExecution",
            "securityagent:ListFindings",
            "securityagent:BatchGetFindings"
         ],
         "Resource": "*"
      }
   ]
}
```

示例：拒绝来自指定 AWS 账户的所有访问的 VPC 终端节点策略

以下 VPC 终端节点策略拒绝 AWS 账户使用该终端节点访问123456789012所有资源。此策略允许来自其他账户的所有操作。

```
{
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": "*",
         "Action": "securityagent:*",
         "Resource": "*"
      },
      {
         "Effect": "Deny",
         "Principal": {
            "AWS": "123456789012"
         },
         "Action": "securityagent:*",
         "Resource": "*"
      }
   ]
}
```

问题排查

调用安全客户端 API 时连接超时

- 验证 VPC 终端节点状态是否为`available`：

```
aws ec2 describe-vpc-endpoints --vpc-endpoint-ids vpce-1a2b3c4d \
    --query "VpcEndpoints[].State"
```

- 确认与终端节点关联的安全组允许端口 443 上来自您的 VPC CIDR 或源安全组的入站 TCP 流量。
- 检查您的 VPC 路由表是否未将安全客户端流量路由到 Internet 网关或 NAT 网关，而不是终端节点。

的 DNS 解析失败 **securityagent.<region>.api.aws**

- 确认终端节点上已启用私有 DNS：

```
aws ec2 describe-vpc-endpoints --vpc-endpoint-ids vpce-1a2b3c4d \
    --query "VpcEndpoints[].PrivateDnsEnabled"
```

- 确认您的 VPC `true` 上`enableDnsSupport`和`enableDnsHostnames`都设置为：

```
aws ec2 describe-vpc-attribute --vpc-id vpc-1a2b3c4d --attribute enableDnsSupport
aws ec2 describe-vpc-attribute --vpc-id vpc-1a2b3c4d --attribute enableDnsHostnames
```

Access denied调用安全客户端 API 时出错

- 检查 VPC 终端节点策略，确保它允许您尝试执行的操作。
- 验证您的 IAM 身份策略是否授予了所需的`securityagent:`权限。
- 如果使用自定义终端节点策略，请确认终端节点策略和 IAM 策略都允许请求。

AWS 安全代理中的配置和漏洞分析

配置和 IT 控制是 AWS 和您（我们的客户）之间的共同责任。有关更多信息，请参阅 AWS 责任共担模型。

Cross-service 混乱的副手预防

混淆代理问题是一个安全问题，即没有执行操作权限的实体可能会迫使更具权限的实体执行该操作。在 AWS 中，跨服务模拟可能会导致混淆副手问题。 Cross-service 当一个服务（调用服务）调用另一个服务（被调用的服务）时，可能会发生模仿行为。可以操纵调用服务以使用其权限对另一个客户的资

源进行操作，否则该服务不应有访问权限。为了防止这种情况，AWS 提供了一些工具，可帮助您保护所有服务委托人的数据，这些服务委托人已被授予访问您账户中资源的权限。有关更多信息，请参阅 AWS Identity and Access Management 用户指南中的Cross-service 混淆副手预防。

AWS 安全代理的安全最佳实践

AWS Security Agent 提供了许多安全功能，供您在制定和实施自己的安全策略时考虑。以下最佳实践是一般指导原则，并不代表完整安全解决方案。这些最佳实践可能不适合环境或不满足环境要求，请将其视为有用的考虑因素而不是惯例。

使用非生产环境进行渗透测试

AWS 安全代理使用来自 Kali Linux 发行版的一套全面的渗透测试工具。这些工具旨在识别安全漏洞，并可能执行修改应用程序状态、数据或系统配置的操作。

最佳实践：针对反映您的生产设置的非生产环境进行渗透测试。这些测试环境应该：

- 不包含实时客户数据或敏感生产信息
- 与生产系统隔离
- 具有与生产相似的配置和安全控制
- 不要使用凭据访问生产系统

在生产环境中进行测试可能会导致：

- 修改或删除数据
- 服务中断或性能降级
- 意想不到的状态变化
- 触发安全警报或事件响应程序

验证 AI-generated 安全调查结果

AWS 安全代理使用 AI 代理执行安全分析。由于 AI 系统的非确定性，渗透测试运行可能会在不同的执行中产生不同的结果。

最佳实践：在采取补救措施之前验证安全发现：

- 查看 AWS 安全代理针对每项发现生成的验证脚本

- 在测试环境中执行验证脚本以确认漏洞
- 考虑运行多次渗透测试以确保覆盖面全面
- 运用专业的安全判断来评估发现的严重性和可利用性

在您的特定部署环境中，并非所有已发现的问题都可能代表可利用的漏洞。

查看并测试生成的补救代码

AWS 安全代理可以针对已识别的漏洞生成代码修复和安全改进。这些 AI-generated 修复需要在部署前进行验证。

最佳实践：查看所有生成的代码更改：

- 检查建议的修复程序的完整性和正确性
- 在非生产环境中彻底修复了测试问题
- 确认修复程序不会引入新的漏洞或破坏功能
- 应用修复后，使用 AWS 安全代理重新测试，或运行提供的验证脚本
- 遵循贵组织的代码审查和批准流程

代码仓库访问权限

AWS Security Agent 可以通过拉取请求注释和代码审查集成提供有关代码变更的安全指导。为了保护敏感的安全信息，此功能在特定的限制下运行。

限制：代码安全指南仅限于私有仓库。这可确保：

- 安全调查结果对您的组织保密
- 修复之前不会公开披露潜在的漏洞
- 生成的修复建议不会泄露漏洞利用细节

AWS 安全代理不为公共存储库提供代码安全指南。它不会对公开安全发现的公共存储库或开源项目发表评论。

AWS Security Agent 渗透测试可以审查和修复您为渗透测试配置的私有和公有存储库。如果存储库是公开的，则补救代码将以可下载的差异文件而不是拉取请求的形式提供。

可访问的网址

可访问的 URL 指定了渗透测试环境在测试期间可以访问的其他端点。当您的应用程序依赖外部服务（例如第三方身份验证提供商或 CDN）时，这些是必需的。必须将测试所需的所有网络依赖项指定为目标 URL 或可访问 URL。网络会阻止访问任何未指定的端点。

安全隐患：AWS 安全代理未被指示对可访问的 URL 进行安全测试。通过指定可访问的 URL，即表示对这些依赖关系的信任。在测试期间，渗透测试数据（包括凭据）可能会传输到这些可访问的 URL 端点。

跨区域推理

AWS 安全代理会自动选择最佳区域来处理您的推理请求。这样可以最大限度地提高可用计算资源和模型可用性，并提供最佳的客户体验。您的数据仅存储在发出请求的区域；但是，输入提示和输出结果可能会在该区域之外进行处理。我们通过 AWS 网络传输加密的所有数据。

AWS Security Agent 根据区域使用两种类型的跨区域推理：

- 地理跨区域推断 — 将大多数要素的数据处理保持在特定的地理边界（例如美国、欧盟、澳大利亚或日本）内。对于代码补救，来自澳大利亚和日本的请求将在欧盟处理。用于美国东部（弗吉尼亚北部）— us-east-1、美国西部（俄勒冈）— us-west-2、亚太地区（悉尼）— ap-southeast-2、亚太地区（东京）— ap-northeast-1、欧洲（法兰克福）和欧洲（爱尔兰）— eu-west-1。eu-central-1
- 全球跨区域推理 — 将推理请求路由到任何商业 AWS 区域，从而优化可用资源并实现更高的模型吞吐量。用于亚太地区（孟买）— ap-south-1、亚太地区（新加坡）和南美洲（圣保罗）— sa-east-1。ap-southeast-1

对于使用全球跨区域推理的区域，输入提示和输出结果可能会在任何商用 AWS 区域进行处理。跨区域操作期间传输的所有数据都保留在 AWS 网络上，不会通过公共互联网。我们对 AWS 区域之间传输的数据进行加密。

下表根据请求的来源区域和所使用的功能，描述了您的推理请求的处理位置。

请求来源	除代码修复之外的所有功能	代码修复
美国-美国东部（弗吉尼亚北部）-us-east-1 ，美国西部（俄勒冈）-us-west-2	美国	美国

请求来源	除代码修复之外的所有功能	代码修复
欧盟-欧洲（爱尔兰）-eu-west-1 ，欧洲（法兰克福）-eu-central-1	欧盟	欧盟
澳大利亚-亚太地区（悉尼）-ap-southeast-2	澳大利亚	欧盟
日本-亚太地区（东京）-ap-northeast-1	日本	欧盟
南美洲-南美洲（圣保罗）-sa-east-1	任何商业 AWS 区域	任何商业 AWS 区域
印度 — 亚太地区（孟买）— ap-south-1	任何商业 AWS 区域	任何商业 AWS 区域
东南亚-亚太地区（新加坡）-ap-southeast-1	任何商业 AWS 区域	任何商业 AWS 区域

Cross-Region 推理始终处于启用状态，无法选择退出。 Cross-Region 推断不受服务控制政策 (SCP) 或 AWS Control Tower 中将客户内容限制在特定区域的客户政策的影响。有关 AWS 安全代理如何在跨区域处理期间保护您的数据的更多信息，请参阅Cross-Region 数据处理。

IAM 操作和资源迁移

AWS Security Agent 是一款前沿代理，可在整个开发生命周期中主动保护您的所有环境中的应用程序。如果您在 2026 年 2 月 9 日之前加入 AWS 安全代理，那么您将受到 2026 年 3 月 9 日对现有代理实例资源的更改以及 AWS 安全代理 IAM 操作的影响。为了准备发布公共 API/SDK 支持，代理实例资源已重命名为 Agent Space，并且正在重命名特定的 IAM 操作。这些更改将影响您在 2026 年 3 月 9 日之前创建的任何应用程序或代理实例 IAM 角色。为了避免在 2026 年 3 月 9 日之后出现身份验证问题，您需要按照准备迁移下的步骤进行操作。

> **Note**
>
> 如果您在 2026 年 2 月 9 日之后创建任何新的代理实例，则新的代理实例将创建为代理空间，无需执行任何迁移步骤。

计划中的变更

AWS 安全代理正在将代理实例资源重命名为代理空间：arn:aws:securityagent:us-east-1:{{accountId}}:agent-instance/*已重命名为。arn:aws:securityagent:us-east-1:{{accountId}}:agent-space/*此外，以下 IAM 操作正在重命名：

- securityagent:ListAgentInstances 已重命名为 securityagent:ListAgentSpaces
- securityagent:ListControls 已重命名为 securityagent:ListSecurityRequirements
- securityagent:BatchGetAgentInstances 已重命名为 securityagent:BatchGetAgentSpaces
- securityagent:BatchGetSecurityTestContentMetadata 已重命名为 securityagent:BatchGetPentestJobContentMetadata
- securityagent:BatchGetTasks 已重命名为 securityagent:BatchGetPentestJobTasks
- securityagent:CreateDocumentReview 已重命名为 securityagent:CreateDesignReview
- securityagent:GetDocumentReview 已重命名为 securityagent:GetDesignReview
- securityagent:GetDocumentReviewArtifact 已重命名为 securityagent:GetDesignReviewArtifact
- securityagent:ListDocumentReviews 已重命名为 securityagent:ListDesignReviews
- securityagent:ListDocumentReviewComments 已重命名为 securityagent:ListDesignReviewComments
- securityagent:ListTasks 已重命名为 securityagent:ListPentestJobTasks
- securityagent:StartPentestExecution 已重命名为 securityagent:StartPentestJob
- securityagent:StopPentestExecution 已重命名为 securityagent:StopPentestJob
- securityagent:DeleteDocumentReview 已重命名为 securityagent:DeleteDesignReview

为迁移做准备

为了避免在 2026 年 3 月 9 日之后出现问题，同时在 2026 年 3 月 9 日之前继续使用 AWS 安全代理，您需要在 2026 年 3 月 9 日 roles/policies之前信任您的 IAM 中的新资源和旧资源以及 IAM 操作。以下说明将提供迁移到新资源和操作格式的指南：

1. 登录您的 AWS 账户并导航到 AWS 安全代理控制台

2. 在左侧面板中，选择 “设置”，然后在 “服务角色” 下选择角色
3. 在关联角色的 IAM 控制台中，选择添加权限和附加策略
4. 选择AWSSecurityAgentWebAppPolicy并选择 “添加权限”
   a. 重要提示：请确认您已选择AWSSecurityAgentWebAppPolicy作为新政策，而不是已选择SecurityAgentWebAppAPIPolicy
5. 验证您的 IAM 角色现在是否同时具有权限策略AWSSecurityAgentWebAppPolicy和SecurityAgentWebAppAPIPolicy权限策略
6. 在同一 IAM 角色控制台中，选择信任关系，然后选择编辑信任策略
7. 将您的信任策略更新为以下格式，将 {{accountID}} 替换为您的 AWS 账户 ID

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{accountId}}"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:securityagent:us-east-1:{{accountId}}:application/*",
            "arn:aws:securityagent:us-east-1:{{accountId}}:agent-space/*",
            "arn:aws:securityagent:us-east-1:{{accountId}}:agent-instance/*"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": "sts:SetContext",
      "Condition": {
```

```
        "StringEquals": {
          "aws:SourceAccount": "{{accountId}}"
        },
        "ForAllValues:ArnEquals": {
          "sts:RequestContextProviders": "arn:aws:iam::aws:contextProvider/
IdentityCenter"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:securityagent:us-east-1:{{accountId}}:application/*",
            "arn:aws:securityagent:us-east-1:{{accountId}}:agent-space/*",
            "arn:aws:securityagent:us-east-1:{{accountId}}:agent-instance/*"
          ]
        }
      }
    }
  ]
}
```

8. 返回到 AWS 安全代理控制台。在左侧面板中，选择代理空间
9. 对于每个启用了渗透测试的 Agent Space，请执行以下步骤
   a. 导航到代理空间并选择渗透测试
   b. 在 “服务访问权限” 下，在 “服务角色名称” 下选择角色
   c. 在关联角色的 IAM 控制台中，选择信任关系，然后选择编辑信任策略
   d. 将您的信任策略更新为以下格式，将 {{accountID}} 替换为您的 AWS 账户 ID

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{accountId}}"
        },
        "ArnLike": {
```

```
          "aws:SourceArn": [
            "arn:aws:securityagent:us-east-1:{{accountId}}:agent-space/*",
            "arn:aws:securityagent:us-east-1:{{accountId}}:agent-instance/*"
          ]
        }
      }
    }
  ]
}
```

日志记录 AWS 安全代理 API 调用时使用 AWS CloudTrail

AWS 安全代理与 AWS CloudTrail一项服务集成，该服务提供 AWS 安全客户端中用户、角色或 AWS 服务所采取的操作的记录。 CloudTrail 将 AWS 安全客户端的所有 API 调用捕获为事件。捕获的调用包括来自 AWS 安全客户端控制台的调用和对 AWS 安全客户端 API 操作的代码调用。如果您创建跟踪，则可以允许将 CloudTrail 事件持续传输到 Amazon S3 存储桶，包括 AWS 安全客户端的事件。如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的事件历史记录中查看最新的事件。使用收集的信息 CloudTrail，您可以确定向 Sec AWS urity Agent 发出的请求、发出请求的 IP 地址、谁发出了请求、何时发出请求以及其他详细信息。

要了解更多信息 CloudTrail，请参阅《AWS CloudTrail 用户指南》。

AWS 中的安全代理信息 CloudTrail

CloudTrail 在您创建 AWS 账户时已在您的账户上启用。当 AWS 安全客户端中发生活动时，该活动会与其他 AWS 服务 CloudTrail 事件一起记录在事件历史记录中。您可以在自己的 AWS 账户中查看、搜索和下载最近发生的事件。有关更多信息，请参阅使用事件历史记录查看 CloudTrail 事件。

要持续记录您的 AWS 账户中的事件，包括 AWS 安全客户端的事件，请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。默认情况下，当您在控制台中创建跟踪时，该跟踪将应用于所有 AWS 区域。跟踪记录 AWS 分区中所有区域的事件，并将日志文件传送到您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，以进一步分析和处理 CloudTrail 日志中收集的事件数据。有关更多信息，请参阅下列内容：

- 创建跟踪记录概述
- CloudTrail 支持的服务和集成
- 配置 Amazon SNS 通知 CloudTrail
- 接收来自多个地区的 CloudTrail 日志文件和接收来自多个账户的 CloudTrail 日志文件

所有 AWS 安全客户端操作都由记录 CloudTrail。例如，对CreatePentestBatchGetPentestJobs、和ListFindings操作的调用会在 CloudTrail 日志文件中生成条目。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根用户证书还是 AWS 身份和访问管理 (IAM) 用户凭证发出。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务发出。

有关更多信息，请参阅 CloudTrail userIdentity 元素。

示例： AWS 安全客户端日志文件条目

了解 AWS 安全客户端日志文件条目

跟踪是一种配置，允许将事件作为日志文件传输到您指定的 Amazon S3 存储桶。 CloudTrail 日志文件包含一个或多个日志条目。事件代表来自任何来源的单个请求，包括有关请求的操作、操作的日期和时间、请求参数等的信息。 CloudTrail 日志文件不是公共 API 调用的有序堆栈跟踪，因此它们不会按任何特定的顺序出现。

以下示例显示了演示该CreatePentest操作的 CloudTrail 日志条目：

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2025-01-15T10:30:00Z",
    "eventSource": "securityagent.amazonaws.com",
    "eventName": "CreatePentest",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "203.0.113.12",
    "userAgent": "aws-cli/2.13.0",
    "requestParameters": {
        "pentestName": "WebApp-Security-Test",
```

```
        "targetUrl": "https://example.com",
        "testScope": "OWASP-Top-10"
    },
    "responseElements": {
        "pentestId": "pt-1234567890abcdef0",
        "pentestArn": "arn:aws:securityagent:us-east-1:123456789012:pentest/
pt-1234567890abcdef0"
    },
    "requestID": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
    "eventID": "12345678-1234-1234-1234-123456789012",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```


服务配额

AWS Security Agent 的配额限制了您可以创建的资源数量和执行操作的速率。通过通过 AWS Support 提交申请，可以增加标记为可调整的配额。有关更多信息，请参阅创建支持案例和工单管理。

操作配额

操作配额限制了安全测试和审查功能的每月使用量，以帮助管理服务容量。

资源	Scope	配额	可调整
设计评论	每个地区每个账户每月一次	200	是
PR 代码评论	每个地区每个账户每月一次	1000	是

配置配额

配置配额限制了您可以在 AWS 安全代理环境中配置的资源和设置的数量。

资源	Scope	配额	可调整
代理空间	每个区域的每个账户	100	是
集成	每个区域的每个账户	20	否
每次集成的整合资源	每次集成	50	不可以
自定义安全要求	每个区域的每个账户	20	否
Pentest 项目	每个区域的每个账户	1000	是
同时进行 pentest 运行	每个区域的每个账户	5	是
代码审查项目	每个区域的每个账户	1000	是
并发代码审查运行	每个区域的每个账户	5	是

文档历史记录

下表介绍了 AWS 安全代理用户指南的一些主要更新和新功能。

变更	说明	日期
私有连接（预览版）	添加了有关私有连接的文档。这项新功能允许 AWS 安全代理使用 Amazon VPC Lattice 连接到在私有网络中运行的源代码控制系统，而无需将您的系统暴露在公共互联网上。	2026 年 6 月 16 日
威胁建模（预览版）	添加了威胁建模文档。这项新功能可根据设计文档（范围文档）、源代码（源代码）或两者构建应用程序的威胁模型。范围文档是定义分析重点的功能设计文档，而源代码则提供了有关您现有系统的背景信	2026 年 6 月 8 日

	息。如果您不提供范围文档，则代理仅从源代码生成威胁模型。每次运行都会生成系统概述和一组按STRIDE类别分类的威胁，以及严重性等级和建议。	
完整存储库代码审查（预览版）	添加了完整存储库代码审查的文档。这项新功能可对整个代码库执行上下文感知安全分析，并针对发现结果生成代码补救措施。	2026 年 5 月 12 日
更新了用于查找补救措施的区域可用性	在 AWS 安全代理 Web 应用程序中进行渗透测试发现补救措施的可用区域已更新。	2026 年 4 月 16 日
更新了启用查找补救措施的区域可用性	在 AWS 管理控制台中启用渗透测试发现补救措施的区域已更新。	2026 年 4 月 16 日
客户自主管理型密钥支持	添加了客户托管密钥 (CMK) 支持的文档。现在，您可以在创建代理空间和集成时指定客户托管的 KMS 密钥，以便使用您控制的密钥对数据进行加密。	2026年3月31日
AWS 托管策略更新	为新类型 TargetDomain 和 DesignReviewFeedback 资源类型添加了 AWSSecurityAgentWebAppPolicy 托管策略。	2026年3月31日
AWS 托管策略更新	为新 AgentSpace 资源类型和 IAM 操作名称更改添加了 AWSSecurityAgentWebAppPolicy 托管策略。	2026年2月9日

AWS 托管策略更新	已更新 SecurityAgentWebAppAPIPolicy ，允许买家删除设计评论。	2026 年 1 月 28 日
AWS 托管策略更新	已更新 SecurityAgentWebAppAPIPolicy ，允许客户针对安全发现启动自动代码修复。	2026 年 1 月 20 日
AWS 托管策略更新	已更新 SecurityAgentWebAppAPIPolicy 为允许客户在控制台中查看图像。	2025 年 12 月 5 日
AWS 安全代理初始版本	服务的初始文档发布	2025 年 12 月 2 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。