Building a scalable vulnerability management program on AWS

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Building a scalable vulnerability management program on AWS

Table of Contents

Building a scalable vulnerability management program on AWS

Anna McAbee and Megan O'Neil, Amazon Web Services (AWS)

October 2023 (document history)

Depending on the underlying technology you're using, a variety of tools and scans can generate security findings in a cloud environment. Without processes in place to handle these findings, they can begin to accumulate, often leading to thousands to tens of thousands of findings in a short amount of time. However, with a structured vulnerability management program and proper operationalization of your tooling, your organization can handle and triage a large number of findings from diverse sources.

Vulnerability management focuses on discovering, prioritizing, assessing, remediating, and reporting on vulnerabilities. *Patch management*, on the other hand, focuses on patching or updating software to remove or remediate security vulnerabilities. Patch management is just one aspect of vulnerability management. Generally, we recommend establishing both a *patch-in-place process* (also known as a *mitigate-in-place* process) to address critical, patch-now scenarios, and a standard process that you run on a regular cadence in order to release patched Amazon Machine Images (AMIs), containers, or software packages. These processes help prepare your organization to respond quickly to a zero-day vulnerability. For critical systems in a production environment, using a patch-in-place process can be faster and more reliable than rolling out a new AMI across the fleet. For regularly scheduled patches, such as operating system (OS) and software patches, we recommend that you build and test using standard development processes, as you would any software-level change. This provides better stability for standard operating modes. You can use Patch Manager, a capability of AWS Systems Manager, or other third-party products as patch-in-place solutions. For more information about using Patch Manager, see Patch management in *AWS Cloud Adoption Framework: Operations Perspective*. Also, you can use EC2 Image Builder to automate the creation, management, and deployment of customized and up-to-date server images.

Building a scalable vulnerability management program on AWS involves managing traditional software and network vulnerabilities in addition to cloud configuration risks. A cloud configuration risk, such as an unencrypted Amazon Simple Storage Service (Amazon S3) bucket, should follow a similar triage and remediation process as a software vulnerability. In both of these cases, the application team must own and be accountable for the security of their application, including

the underlying infrastructure. This distribution of ownership is key for an effective and scalable vulnerability management program.

This guide discusses how to streamline the identification and remediation of vulnerabilities in order to reduce overall risk. Use the following sections to build and iterate on your vulnerability management program:

1. Prepare – Prepare your people, processes, and technology to identify, assess, and remediate vulnerabilities in your environment.
2. Triage and remediate – Route security findings to the relevant stakeholders, identify the appropriate remediation action, and then take the remediation action.
3. Report and improve – Use reporting mechanisms to identify opportunities for improvement, and then iterate on your vulnerability management program.

Building a cloud vulnerability management program often involves iteration. Prioritize recommendations in this guide and regularly revisit your backlog to stay current with technology changes and your business requirements.

Intended audience

This guide is intended for large enterprises that have three primary teams who are responsible for security related findings: a security team, a Cloud Center of Excellence (CCoE) or cloud team, and application (or *developer*) teams. This guide uses the most common enterprise operating models and builds upon those operating models to enable a more efficient response to security findings and improve security outcomes. Organizations using AWS might have different structures and different operating models; however, you can modify many of the concepts in this guide to fit different operating models and smaller organizations.

Objectives

This guide can help you and your organization:

- Develop policies to streamline vulnerability management and ensure accountability
- Establish mechanisms to distribute responsibility for security to the application teams
- Configure relevant AWS services according to best practices for scalable vulnerability management

- Distribute ownership of security findings
- Establish mechanisms to report on and iterate on your vulnerability management program
- Improve security finding visibility and improve overall security posture

Prepare your scalable vulnerability management program

Preparing to build a scalable vulnerability management program involves educating people, developing processes, and implementing the proper technology according to best practices. People, processes, and technology are equally important for an effective vulnerability management program, and you must tightly integrate them to manage vulnerabilities at scale.

This section of the guide reviews the foundational actions you can take to prepare your scalable vulnerability management program on AWS.

Topics

Define a vulnerability management plan

The first step when preparing your cloud vulnerability management program is defining your *vulnerability management plan*. This plan includes the policies and processes your organization follows. This plan should be documented and accessible by all stakeholders. A vulnerability management plan is a high-level document that typically includes the following sections:

- **Goals and scope** – Outline the goals, functions, and scope of vulnerability management.
- **Roles and responsibilities** – List the vulnerability management stakeholders and detail their responsibilities.
- **Vulnerability severity and prioritization definitions** – Determine how to classify the severity of a vulnerability and how to prioritize it.
- **Service level agreements (SLAs) for remediation** – For each severity level, define the maximum amount of time a remediation owner has to resolve a security finding. Because SLA compliance is

an integral part of having an effective and scalable vulnerability management program, consider how to track whether you're meeting these SLAs.

- **Exception process** – Detail the process of submitting, approving, and updating exceptions. This process should make sure that exceptions are legitimate, time-bound, and tracked.
- **Sources of vulnerability information** – List the sources or tools that generate security findings. For more information about AWS services that could be sources for security findings, see Configure AWS security services in this guide.

While these sections are common throughout companies of different sizes and industries, each organization's vulnerability management plan is unique. You need to build a vulnerability management plan that works best for your organization. Expect to iterate your plan over time to incorporate lessons learned and evolving technologies.

Distribute security ownership

The AWS shared responsibility model defines how AWS and its customers share responsibility for cloud security and compliance. In this model, AWS secures the infrastructure that runs all of the services offered in the AWS Cloud, and AWS customers are responsible for securing their data and applications.

You can mirror this model inside your organization and distribute the responsibilities between your cloud and application teams. This helps you scale your cloud security programs more effectively because the application teams take ownership of certain security aspects of their applications. The simplest interpretation of the shared responsibility model is that if you have access to configure the resource, then you are responsible for the security of that resource.

A key part of distributing security responsibilities to application teams is building self-service security tools that help your application teams automate. Initially, this can be a joint effort. The security team can translate security requirements into code-scanning tools, and then application teams can use those tools to build and share solutions with their internal developer community. This contributes to greater efficiencies across other teams that need to meet similar security requirements.

The following table outlines the steps for distributing ownership to application teams and provides examples.

Step	Action	Example
1	**Define your security requirements** – What are you trying to achieve? This might come from a security standard or compliance requirement.	An example security requirement is least-pri vilege access for application identities.
2	**Enumerate controls for a security requirement** – What does this requirement actually mean from a control perspective? What do I need to do to achieve this?	To achieve least-privilege for application identities, the following are two sample controls: • Use AWS Identity and Access Management (IAM) roles • Do not use wildcards in IAM policies
3	**Document guidance for the controls** – With these controls, what guidance can you provide to a developer to help them comply with the control?	Initially, you might start by documenting simple example policies, including secure and unsecure IAM policies and Amazon Simple Storage Service (Amazon S3) bucket policies. Next, you can embed policy-scanning solutions within continuous integrati on and continuous delivery (CI/CD) pipelines, such as using AWS Config rules for proactive evaluation.
4	**Develop reusable artifacts** – With the guidance, can you make it even easier and	You might create infrastru cture as code (IaC) to deploy IAM policies that follow the

Step	Action	Example
	develop reusable artifacts for developers?	principle of least privilege. You can store these reusable artifacts in a code repository.

Self-service might not work for all security requirements, but it can work for standard scenarios. By following these steps, organizations can empower their application teams to handle more of their own security responsibilities in a scalable way. Overall, the distributed responsibility model leads to more collaborative security practices within many organizations.

Develop a vulnerability disclosure program

For a defense-in-depth approach to vulnerability management, create a vulnerability disclosure program so that people inside or outside your organization can report security vulnerabilities or risks.

For people inside your organization, establish a process to submit risks or vulnerabilities. This can be done through a ticketing system or email. Regardless of the process you choose, it's essential that your employees are aware of the process and can easily submit any vulnerabilities or risks that they encounter.

For people outside your organization, establish an external webpage for submitting potential security vulnerabilities. As an example, see the AWS Vulnerability Reporting webpage. This webpage should also contain disclosure guidelines to help protect your organization's data and assets. A vulnerability disclosure program should not encourage potentially harmful activity, so it's essential that you have a clear policy with guidelines. Building a mature, responsible disclosure program is a goal to strive for as you mature your program. Most don't start with an external disclosure program, and it takes time to get it right.

Prepare your AWS environment

Before implementing any vulnerability management tooling, make sure that your AWS environment is architected to support a scalable vulnerability management program. The structure of your AWS accounts and your organization's tagging policies can simplify the process of building a scalable vulnerability management program.

Develop an AWS account structure

AWS Organizations helps centrally manage and govern an AWS environment as your business grows and scales its AWS resources. An *organization* in AWS Organizations consolidates your AWS accounts into logical groups, or *organizational units*, so that you can administer them as a single unit. You manage AWS Organizations from a dedicated account, called the *management account*. For more information, see AWS Organizations terminology and concepts.

We recommend that you manage your AWS multi-account environment in AWS Organizations. This helps create a full inventory of your company's accounts and resources. This complete asset inventory is a critical aspect of vulnerability management. Application teams should not use accounts that are outside of the organization.

AWS Control Tower helps you set up and govern an AWS multi-account environment, following prescriptive best practices. If you haven't already established a multi-account environment, AWS Control Tower is a good starting point.

We recommend using the dedicated account structure and best practices described in the AWS Security Reference Architecture (AWS SRA). The Security Tooling account should serve as your delegated administrator for your security services. More information about configuring your vulnerability management tooling in this account is provided later in this guide. Host applications in dedicated accounts in the Workloads organization unit (OU). This establishes strong workload-level isolation and explicit security boundaries for each application. For information about the design principles and benefits of using a multi-account approach, see Organizing Your AWS Environment Using Multiple Accounts (AWS whitepaper).

Having an intentional account structure and centrally managing security services from a dedicated account are critical aspects of a scalable vulnerability management program.

Define, implement, and enforce tags

Tags are key-value pairs that act as metadata for organizing your AWS resources. For more information, see Tagging your AWS resources. You can use tags to provide business context, such as business unit, application owner, environment, and cost center. The following table shows a set of sample tags.

Key	Value
BusinessUnit	HumanResources

Key	Value
CostCenter	CC101
ApplicationTeam	HumanResourcesTechnology
Environment	Production

Tags can help you prioritize findings. For example, it can help you:

- Identify the owner of a resource who is responsible for patching a vulnerability
- Track which applications or business units have a large number of findings
- Escalate the severity of findings for certain data classifications, such as personally identifiable information (PII) or payment card industry (PCI) data
- Identify the type of data in the environment, such as test data in a lower-level development environment or production data

To help you achieve effective tagging at scale, follow the instructions in Building your tagging strategy in *Best Practices for Tagging AWS Resources* (AWS whitepaper).

Monitor AWS security bulletins

We highly recommend monitoring AWS security bulletins on a regular and frequent basis. Security bulletins can notify you of any new security-related vulnerabilities, affected services, and applicable updates. You can also subscribe to an RSS feed for the security bulletins and build a process to ingest and address these bulletins as part of your vulnerability management program.

Configure AWS security services

AWS offers a variety of security services that are designed to help protect your AWS environment. For your vulnerability management program, we recommend that you enable the following AWS services in each account:

- Amazon GuardDuty helps detect active threats in your environment. A GuardDuty finding could help you identify an unknown vulnerability that was exploited in your environment. It could also help you understand the effects of an unpatched vulnerability.

- AWS Health provides ongoing visibility into your resource performance and the availability of your AWS services and accounts.
- AWS Identity and Access Management Access Analyzer analyzes the resource-based policies in your AWS environment to identify resources that are shared with an external entity. This can help you identify vulnerabilities associated with unintended access to your resources and data. For each instance of a resource shared outside of your account, IAM Access Analyzer generates a finding.
- Amazon Inspector is a vulnerability management service that continuously scans your AWS workloads for software vulnerabilities and unintended network exposure.
- AWS Security Hub CSPM helps you check your AWS environment against security industry standards and can identify cloud configuration risks. It also provides a comprehensive view of your AWS security state by aggregating findings from other AWS security services and third-party security tools.

This section discusses how to enable and configure Amazon Inspector and Security Hub CSPM to help you establish a scalable vulnerability management program.

Using Amazon Inspector in your vulnerability management program

Amazon Inspector is a vulnerability management service that continually scans your Amazon Elastic Compute Cloud (Amazon EC2) instances, Amazon Elastic Container Registry (Amazon ECR) container images, and AWS Lambda functions for software vulnerabilities and unintended network exposure. You can use Amazon Inspector to gain visibility and prioritize resolution of software vulnerabilities across your AWS environments.

Amazon Inspector continuously assesses your environment throughout the lifecycle of your resources. It automatically rescans resources in response to changes that could introduce a new vulnerability. For example, it rescans when you install a new package on an EC2 instance, when you install a patch, or when a new common vulnerabilities and exposures (CVE) that affects the resource is published. When Amazon Inspector identifies a vulnerability or an open network path, it produces a finding that you can investigate. The finding provides comprehensive information about the vulnerability, including the following:

- Amazon Inspector risk score
- Common Vulnerability Scoring System (CVSS) score
- Affected resource

- Vulnerability intelligence data about the CVE from Amazon, Recorded Future, and Cybersecurity and Infrastructure Security Agency (CISA)
- Remediation recommendations

For instructions on setting up Amazon Inspector, see Getting started with Amazon Inspector. The *Activate Amazon Inspector* step in this tutorial provides two configuration options: a standalone account environment and a multi-account environment. We recommend using the multi-account environment option if you want to monitor multiple AWS accounts that are members of an organization in AWS Organizations.

When you set up Amazon Inspector for a multi-account environment, you designate an account in the organization to be the Amazon Inspector delegated administrator. The delegated administrator can manage findings and some settings for organization members. For example, the delegated administrator can view the details of aggregated findings for all member accounts, enable or disable scans for member accounts, and review scanned resources. The AWS SRA recommends that you create a Security Tooling account and use it as the Amazon Inspector delegated administrator.

Using AWS Security Hub CSPM in your vulnerability managment program

Building a scalable vulnerability management program on AWS involves managing traditional software and network vulnerabilities in addition to cloud configuration risks. AWS Security Hub CSPM helps you check your AWS environment against security industry standards and can identify cloud configuration risks. Security Hub CSPM also provides a comprehensive view of your security state in AWS by aggregating security findings from other AWS security services and third-party security tools.

In the following sections, we provide best practices and recommendations for setting up Security Hub CSPM to support your vulnerability management program:

- Setting up Security Hub CSPM
- Enabling Security Hub CSPM standards
- Managing Security Hub CSPM findings
- Aggregating findings from other security services and tools

Setting up Security Hub CSPM

For setup instructions, see Setting up AWS Security Hub CSPM. To use Security Hub CSPM, you must enable AWS Config. For more information, see Enabling and configuring AWS Config in the Security Hub CSPM documentation.

If you are integrated with AWS Organizations, from the organization management account, you designate an account to be the Security Hub CSPM delegated administrator. For instructions, see Designating the Security Hub CSPM delegated administrator. The AWS SRA recommends that you create a Security Tooling account and use it as the Security Hub CSPM delegated administrator.

The delegated administrator automatically has access to configure Security Hub CSPM for all member accounts in the organization and to view findings associated with those accounts. We recommend that you enable AWS ConfigSecurity Hub CSPM in all AWS Regions and all of your AWS accounts. You can configure Security Hub CSPM to automatically treat new organization accounts as Security Hub CSPM member accounts. For instructions, see Managing member accounts that belong to an organization.

Enabling Security Hub CSPM standards

Security Hub CSPM generates findings by running automated and continuous security checks against *security controls*. The controls are associated with one or more *security standards*. The controls help you determine whether the requirements in a standard are being met.

When you enable a standard in Security Hub CSPM, Security Hub CSPM automatically enables the controls that apply to the standard. Security Hub CSPM uses AWS Config rules to perform most of its security checks for controls. You can enable or disable Security Hub CSPM standards at any time. For more information, see Security controls and standards in AWS Security Hub CSPM. For a complete list of standards, see Security Hub CSPM standards reference.

If your organization does not already have a preferred security standard, we recommend using the AWS Foundational Security Best Practices (FSBP) standard. This standard is designed to detect when AWS accounts and resource deviate from security best practices. AWS curates this standard and updates it regularly to cover new features and services. After triaging the FSBP findings, consider enabling other standards.

Managing Security Hub CSPM findings

Security Hub CSPM provides several features that help you address large volumes of findings from across your organization and understand the security state of your AWS environment. To help you manage findings, we recommend enabling the following two Security Hub CSPM features:

- Use cross-Region aggregation to aggregate findings, finding updates, insights, control compliance statuses, and security scores from multiple AWS Regions to a single aggregation Region.
- Use consolidated control findings to reduce finding noise by removing duplicate findings. When consolidated control findings is turned on in your account, Security Hub CSPM generates a single new finding or finding update for each security check of a control, even if a control applies to multiple enabled standards.

Aggregating findings from other security services and tools

In addition to generating security findings, you can use Security Hub CSPM to aggregate finding data from several AWS services and supported third-party security solutions. This section focuses on sending security findings to Security Hub CSPM. The next section, Prepare to assign security findings, discusses how you can integrate Security Hub CSPM with products that can receive findings from Security Hub CSPM.

There are many AWS services, third-party products, and open-source solutions available that you can integrate with Security Hub CSPM. If you are just getting started, we recommend doing the following:

1. **Enable integrated AWS services** – Most AWS service integrations that send findings to Security Hub CSPM are automatically activated after you enable both Security Hub CSPM and the integrated service. For your vulnerability management program, we recommend enabling Amazon Inspector, Amazon GuardDuty, AWS Health, and IAM Access Analyzer in each account. These services automatically send their findings to Security Hub CSPM. For a complete list of supported AWS service integrations, see AWS services that send findings to Security Hub CSPM.

   > **Note**
   >
   > AWS Health sends findings to Security Hub CSPM if one of the following conditions are met:
   >
   > - The finding is associated with an AWS security service

- The finding **typecode** contains the words `security`, `abuse`, or `certificate`
- The finding AWS Health service is `risk` or `abuse`

2. **Set up third-party integrations** – For a list of the currently supported integrations, see Available third-party partner product integrations. Select any additional tools that can send findings to or receive findings from Security Hub CSPM. You might already have some of these third-party tools. Follow the product instructions to configure integration with Security Hub CSPM.

Prepare to assign security findings

In this section, you set up the tools that your teams use to manage and assign security findings. This section includes the following options:

- Manage findings in existing tools and workflows – This option integrates AWS Security Hub CSPM with existing systems that your teams use to manage their daily tasks, such as a product backlog. This option is recommended for teams that have established tools to manage their workflows.
- Manage findings in Security Hub CSPM – This option configures notifications for Security Hub CSPM events so that the appropriate team receives an alert and can address the finding in Security Hub CSPM.

Decide which workflow would work best for your teams, and make sure that security findings can make it promptly to their respective owners.

Manage findings in existing tools and workflows

We recommend additional Security Hub CSPM integrations for enterprise organizations that have established tools that teams use to manage or perform their daily tasks. You can import Security Hub CSPM finding data into several technology platforms. Examples include:

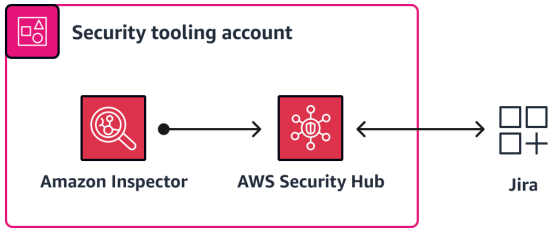
- Security information and event management (SIEM) systems help security teams triage operational security events. SIEM systems provide real-time analysis of security alerts that are generated by applications and network hardware.
- Governance, risk, and compliance (GRC) systems help compliance and governance teams monitor and report on risk management data. GRC tools are software applications that businesses can

use to manage policies, assess risk, control user access, and streamline compliance. You might use GRC tools to integrate business processes, reduce costs, and improve efficiency.

- Product backlog and ticketing systems help application and cloud teams manage features and prioritize development tasks. Atlassian Jira and Microsoft Azure DevOps are examples of these systems.

Integrating Security Hub CSPM findings directly with these existing enterprise systems can improve mean time to recovery (MTTR) and security outcomes because the daily operational workflow doesn't have to change. Teams can respond and learn from security findings much faster because they don't have to use separate workflows and tools. Integration makes addressing security findings part of the normal, standard workflow.

Security Hub CSPM integrates with multiple third-party partner products. For a complete list and instructions, see Available third-party partner product integrations in the Security Hub CSPM documentation. Common integrations include Atlassian - Jira Service Management, Bidirectionally integrate AWS Security Hub CSPM with Jira software, and ServiceNow – ITSM. The following diagram shows how you can configure Amazon Inspector to send findings to Security Hub CSPM and then configure Security Hub CSPM to send all findings to Jira.



Manage findings in Security Hub CSPM

You can build a cloud-based notification system for Security Hub CSPM findings by using Amazon EventBridge rules and Amazon Simple Notification Service (Amazon SNS) topics. This system notifies the appropriate team about a finding when it is created. For this approach, the multi-account strategy described in Develop an AWS account structure is critical because applications are separated into dedicated accounts. This helps you notify the correct teams for each finding.

Security or cloud teams might choose to receive events from all AWS accounts. In this case, build an EventBridge rule within the Security Hub CSPM delegated administrator account and subscribe an Amazon SNS topic that notifies these teams. For application teams, configure an EventBridge

rule and SNS topic within their respective application accounts. When a Security Hub CSPM finding occurs within an application account, the responsible team is notified about the finding.

Security Hub CSPM already automatically sends all new findings and all updates to existing findings to EventBridge as **Security Hub CSPM Findings - Imported** events. Each **Security Hub CSPM Findings - Imported** event contains a single finding. You can apply filters on EventBridge rules so that a finding initiates the rule only if the finding matches the filters. For instructions, see Configuring an EventBridge rule for automatically sent findings. For more information about creating and subscribing Amazon SNS topics, see Configuring Amazon SNS.

Consider the following when using this approach:

- For application teams, create EventBridge rules within each AWS account and AWS Region where the application is hosted.
- For security and cloud teams, create EventBridge rules in the Security Hub CSPM delegated administrator account. This notifies teams about all findings in the member accounts.
- Amazon SNS sends a notification each day if the status of the security finding is `NEW`. If you want to turn off the daily notifications, you can create a custom AWS Lambda function that changes the status of the finding from `NEW` to `NOTIFIED` after the Amazon SNS subscriber receives the notification.

Triage and remediate security findings in your AWS environment

Triaging a security finding involves routing the finding to the appropriate stakeholder, assessing and prioritizing the finding, then remediating it. This section reviews each of these steps in detail and provides recommendations for scalability and efficiency. It also includes examples to help illustrate the triage and remediation process.

Topics

Define ownership of security findings

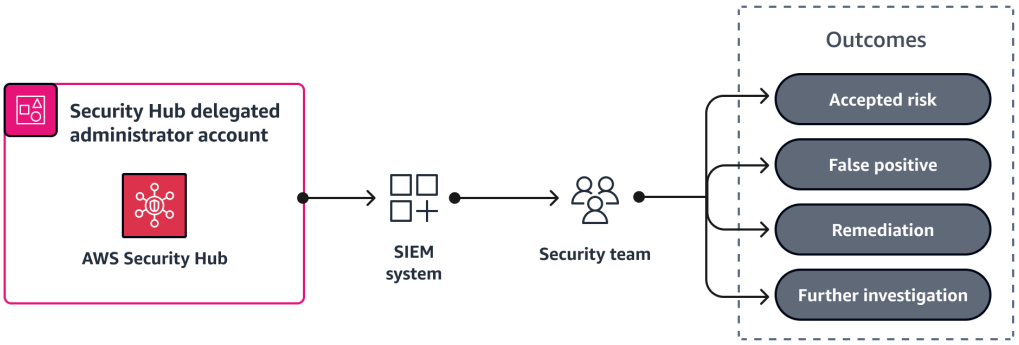
Defining an ownership model to triage security findings can be challenging, but it doesn't have to be. The security landscape changes constantly, and practitioners must be flexible to adapt to these changes. Adopt a flexible approach to developing your ownership model for security findings. Your initial model should enable your teams to act right away. We recommend starting with basic ownership logic and refining that logic over time. If you delay to define the perfect ownership criteria, the number of security findings will continue to grow.

To facilitate assignment of findings to the appropriate teams and resources, we recommend integrating AWS Security Hub CSPM with any existing systems that your teams use to manage their daily tasks. For example, you can integrate Security Hub CSPM with security information and event management (SIEM) systems or product backlog and ticketing systems. For more information, see Prepare to assign security findings in this guide.

The following is an example of an ownership model that you can use as a starting point:

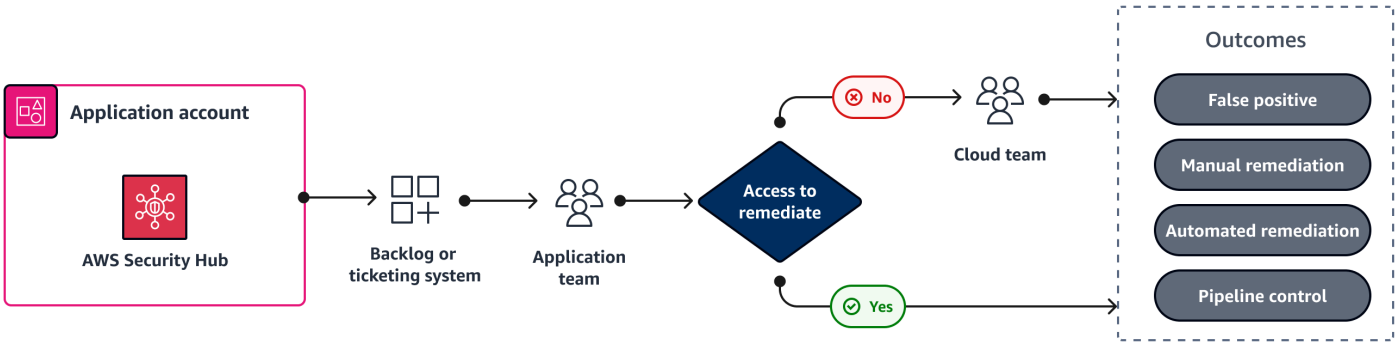
- **The security team reviews potentially active threats and helps assess and prioritize security findings**. The security team has the expertise and the tools to properly evaluate context. They understand the additional security-related data that helps them assess and prioritize vulnerabilities and investigate threat-detection events. If finding severity or additional tuning are

needed, see the Assess and prioritize security findings section in this guide. For an example, see Security team example in this guide.



- **Distribute security findings between the cloud and application teams** – As discussed in the Distribute security ownership section, the team that has access to configure the resource is responsible for its secure configuration. Application teams are responsible for the security findings related to the resources that they build and configure, and the cloud team is responsible for security findings related to the wide-reaching configurations. In most cases, application teams do not have access to change wide-reaching configurations and AWS services, such as AWS Control Tower, service control policies (SCPs) in AWS Organizations, networking-related VPC configurations, and AWS IAM Identity Center.

  For multi-account environments that separate applications into dedicated accounts, you can usually integrate security-related findings for the account into the application's backlog or ticketing system. From that system, the cloud team or application team can address the finding. For examples, see Cloud team example or Application team example in this guide.



- **Assign remaining, unresolved findings to the cloud team** – Residual findings might be related to default settings or wide-reaching configurations that the cloud team can address. This team likely has the most historical knowledge and access to resolve the finding. Overall, this is typically a significantly smaller subset of the total findings.

Assess and prioritize security findings

A critical component of an effective vulnerability management program is the ability to assess and prioritize security findings. This is where pulling in context, organizational history, and tuning detection systems comes into place. Prioritization of security findings helps establish the appropriate speed for response level.

For Amazon Inspector, AWS Security Hub CSPM, and Amazon GuardDuty, findings contain a severity label or score. We recommend prioritizing the investigation of all critical and high severity findings in Security Hub CSPM, including findings related to the Foundational Security Best Practices (FSBP) standard, Amazon Inspector, and GuardDuty. Finding severity labels are scores are determined as follows:

- The Amazon Inspector score is a highly contextualized score for each finding. It's calculated by correlating Common Vulnerability Scoring System (CVSS) base score information with network reachability results and exploitability data. Using this score, you can prioritize findings to focus on the most critical findings and vulnerable resources. In addition to the score, Amazon Inspector also provides enhanced vulnerability intelligence about Common Vulnerabilities and Exposures (CVE). This is a summary of the available intelligence about the CVE from Amazon as well as industry-standard security intelligence sources, such as Recorded Future and Cybersecurity and Infrastructure Security Agency (CISA). For example, Amazon Inspector can provide the names of known malware kits used to exploit a vulnerability. For more information, see Vulnerability Intelligence.
- Each GuardDuty finding has an assigned severity level and value that reflects the finding's potential risk to your environment. This level and value are determined by AWS security engineers. For example, a `High` severity level indicates that a resource is compromised and is actively being used for unauthorized purposes. We recommend that you treat a `High` severity GuardDuty finding as a priority and immediately remediate to prevent further unauthorized use.
- The severity of an Security Hub CSPM control finding is determined by the difficulty to exploit and the likelihood of compromise. The difficulty is determined by the amount of sophistication or complexity that is required to use the weakness to carry out a threat scenario. The likelihood of compromise indicates how likely it is that the threat scenario will result in a disruption or breach of your AWS services or resources.

In order to tune findings, you can suppress or archive specific findings directly in the respective service console or by using the service's API. In addition, you can make changes to findings in

Security Hub CSPM by using automation rules. GuardDuty and Amazon Inspector findings are automatically sent to Security Hub CSPM. You can use automation rules to automatically update (such as changing the severity) or suppress findings in near real-time, based on criteria that you define. As you create automation rules, we recommend adding context to the rule description, such as the date of creation or modification, who created it, and why the rule is needed. This information is often helpful for future reference.

Remediate security findings

After assessing and prioritizing a finding, the next action is remediating the finding. There are many different actions you could take to remediate a finding. For software vulnerabilities, you might update the operating system or apply a patch. For cloud configuration findings, you might update the resource configuration. In general, the actions you take to remediate can be grouped into one of the following outcomes:

- **Manual remediation** – You manually provide a fix to the vulnerability, such as modifying the properties of an AWS resource to enable encryption. If the finding is from one a managed check in Security Hub CSPM, then the finding includes a link to instructions for manually remediating the finding.
- **Reusable artifact** – You update the infrastructure as code (IaC) to fix the vulnerability and know that others could benefit from a similar solution. Consider uploading the updated IaC and a brief summary of the resolution to an internal shared code repository.
- **Automated remediation** – The vulnerability is automatically remediated through mechanisms you created.
- **Pipeline control** – You apply a control within your continuous integration and continuous delivery (CI/CD) pipeline that prevents deployment if the vulnerability is present.
- **Accepted risk** – You take no action or implement a compensating control, and you accept the risk that the vulnerability presents. Track the accepted risk in a dedicated location, such as a risk registry.
- **False positive** – You take no action because you have determined the finding didn't correctly identify a vulnerability.

A complete list of the various actions you can take and tools you can use to remediate a vulnerability is out of scope for this guide. However, there are some services and tools that you can help you remediate vulnerabilities at scale that are worth noting, including:

- Patch Manager, a capability of AWS Systems Manager, automates the process of patching managed nodes with both security-related updates and other types of updates. You can use Patch Manager to apply patches for both operating systems and applications.
- AWS Firewall Manager helps you centrally configure and manage firewall rules across your accounts and applications in AWS Organizations. As new applications are created, Firewall Manager makes it easier to bring new applications and resources into compliance by enforcing a common set of security rules.
- Automated Security Response on AWS is an AWS Solution that works with Security Hub CSPM and provides predefined response and remediation actions based on industry compliance standards and best practices for security threats.

Examples of triaging and remediating security findings

This section provides examples of the triage process for the security, cloud, and application teams. It discusses the types of findings each team commonly addresses and provides an example of how to respond. High-level remediation guidance is also included.

The following examples are included in this section:

- Security team example: Creating a Security Hub CSPM automation rule
- Cloud team example: Changing VPC configurations
- Application team example: Creating an AWS Config rule

Security team example: Creating a Security Hub CSPM automation rule

The security team receives findings related to threat detection, including Amazon GuardDuty findings. For a complete list of GuardDuty finding types that are categorized by AWS resource type, see Finding types in the GuardDuty documentation. Security teams must be familiar with all of these finding types.

For this example, the security team is accepting the level of associated risk for security findings in an AWS account that is used strictly for learning purposes and does not include important or sensitive data. The name of this account is `sandbox`, and the account ID is `123456789012`. The security team can create an AWS Security Hub CSPM automation rule that suppresses all GuardDuty findings from this account. They can either create a rule from a template, which covers many common use cases, or they can create a custom rule. In Security Hub CSPM, we recommend previewing the results of the criteria to confirm that the rule returns the intended findings.

> **Note**
>
> This example highlights the functionality of automation rules. We don't recommend suppressing all GuardDuty findings for an account. Context matters, and each organization must choose which findings to suppress based on data type, classification, and mitigation controls.

The following are the parameters used to create this automation rule:

- **Rule:**
  - **Rule name** is `Suppress findings from Sandbox account`
  - **Rule description** is `Date: 06/25/23 Authored by: John Doe Reason: Suppress GuardDuty findings from the sandbox account`
- **Criteria:**
  - `AwsAccountId` = `123456789012`
  - `ProductName` = `GuardDuty`
  - `WorkflowStatus` = `NEW`
  - `RecordState` = `ACTIVE`
- **Automated action:**
  - `Workflow.status` is `SUPPRESSED`

For more information, see Automation rules in the Security Hub CSPM documentation. Security teams have many options for investigating and remediating findings for detected threats. For extensive guidance, see the AWS Security Incident Response Guide. We recommend reviewing this guide to confirm that you have established strong incident response processes.

Cloud team example: Changing VPC configurations

The cloud team is responsible for triaging and remediating security findings that have common trends, such as changes to AWS default settings that might not suit your use case. These findings tend to affect many AWS accounts or resources, such as VPC configurations, or they include a restriction that should be placed across the entire environment. For the most part, the cloud team makes manual, one-time changes, such as adding or updating a policy.

After your organization has used an AWS environment for some time, you might find a set of anti-patterns developing. An *anti-pattern* is a frequently used solution for a recurring issue where the solution is counter-productive, ineffective, or less effective than an alternative. As an alternative to these anti-patterns, your organization can use environment-wide restrictions that are more effective, such as AWS Organizations service control policies (SCPs) or IAM Identity Center permissions sets. SCPs and permissions sets can provide additional restrictions for resource types, such as preventing users from configuring a public Amazon Simple Storage Service (Amazon S3) bucket. Although it can be tempting to restrict every possible security configuration, there are policy size limits for SCPs and permissions sets. We recommend a balanced approach to preventative and detective controls.

The following are some controls from the AWS Security Hub CSPM Foundational Security Best Practices (FSBP) standard that the cloud team might be responsible for:

- [EC2.2] The VPC default security group should not allow inbound and outbound traffic
- [EC2.6] VPC flow logging should be enabled in all VPCs
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [CloudTrail.1] CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events
- [Config.1] AWS Config should be enabled

For this example, the cloud team is addressing a finding for FSBP control EC2.2. The documentation for this control recommends not using the default security group because it allows broad access through the default inbound and outbound rules. Because the default security group cannot be deleted, the recommendation is to change the rule settings to restrict inbound and outbound traffic. To efficiently address this issue, the cloud team should use established mechanisms to modify the security group rules for all VPCs because each VPC has this default security group. In most cases, cloud teams manage VPC configurations by using AWS Control Tower customizations or an infrastructure as code (IaC) tool, such as HashiCorp Terraform or AWS CloudFormation.

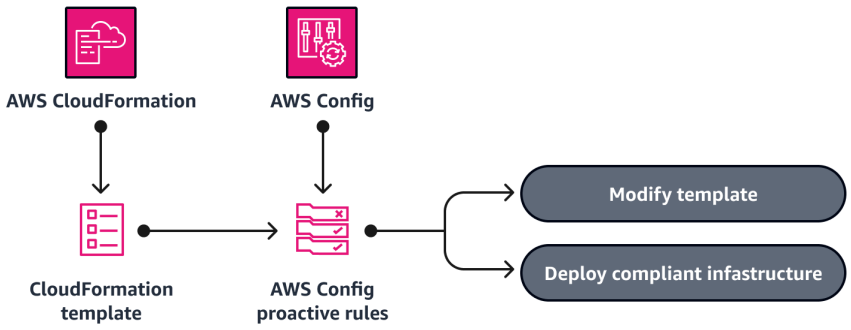
Application team example: Creating an AWS Config rule

The following are some controls from the Security Hub CSPM Foundational Security Best Practices (FSBP) security standard that the application or development team might be responsible for:

- [CloudFront.1] CloudFront distributions should have a default root object configured
- [EC2.19] Security groups should not allow unrestricted access to ports with high risk
- [CodeBuild.1] CodeBuild GitHub or Bitbucket source repository URLs should use OAuth
- [ECS.4] ECS containers should run as non-privileged
- [ELB.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS

For this example, the application team is addressing a finding for FSBP control EC2.19. This control checks whether unrestricted incoming traffic for the security groups is accessible to the specified ports that have the highest risk. This control fails if any of the rules in a security group allow ingress traffic from `0.0.0.0/0` or `::/0` for those ports. The documentation for this control recommends deleting the rules that allow this traffic.

In addition to addressing the individual security group rule, this is a great example of a finding that should result in a new AWS Config rule. By using the proactive evaluation mode, you can help prevent the deployment of risky security group rules in the future. Proactive mode evaluates resources before they have been deployed so that you can prevent misconfigured resources and their associated security findings. When implementing a new service or a new functionality, application teams can run rules in proactive mode as part of their continuous integration and continuous delivery (CI/CD) pipeline to identify noncompliant resources. The following image shows how you can use a proactive AWS Config rule to confirm that the infrastructure defined in an AWS CloudFormation template is compliant.



Another important efficiency can be gained in this example. When an application team creates a proactive AWS Config rule, they can share it in a common code repository so that other application teams can use it.

Each finding associated with a Security Hub CSPM control contains details about the finding and a link to the instructions for remediating the issue. Although cloud teams might encounter findings

that require manual, one-time remediation, when appropriate, we recommend building proactive checks that identify issues as early as possible in the development process.

Report and improve your vulnerability management program

Effective reporting for vulnerability management involves reviewing data, monitoring trends, and sharing knowledge. This provides visibility and helps teams improve their organizations security posture in the AWS Cloud.

Conduct monthly security operations meetings

Monthly security operations meetings are an effective mechanism to promote continued ownership, accountability, and alignment across teams. In the meeting, the stakeholders from the security, cloud, and application teams review data for outstanding security findings, findings outside of service level agreements (SLAs), and the teams that have the most findings.

These meetings help your teams identify anti-patterns, such as opportunities to add more restrictions. Preventative controls and automation opportunities can also be discovered and shared. The meetings also help identify what is working and not working well within the vulnerability management program so that you can make improvements.

By reviewing data, identifying anti-patterns and issues, and sharing information about controls and automations, teams can gain valuable insights and make ongoing refinements that can strengthen their security posture and reduce their security-related SLAs.

Use Security Hub CSPM insights to identify anti-patterns

AWS Security Hub CSPM insights can also help you identify anti-patterns and track your progress in remediating findings. A Security Hub CSPM *insight* is a collection of related findings. It identifies a security area that requires attention and intervention. Security Hub CSPM insights can help you identify specific requirements and develop reports. Security Hub CSPM offers several built-in, managed insights. To track security issues that are unique to your AWS environment and usage, you can create custom insights.

Conclusion and next steps

In summary, an effective vulnerability management program requires thorough preparation and requires that you enable the right tools and integrations, fine-tune those tools, efficiently triage issues, and continuously report and improve. By following the best practices in this guide, organizations can build a scalable vulnerability management program on AWS to help secure their cloud environments.

You can expand on this program to include additional security-related vulnerabilities and findings, such as application security vulnerabilities. AWS Security Hub CSPM supports custom product integrations. Consider using Security Hub CSPM as the integration point for additional security tools and products. This integration allows you to take advantage of the processes and workflows you've already established in your vulnerability management program, such as the direct integration with product backlogs and the monthly security review meetings.

The following table summarizes the phases and action items described in this guide.

Phase	Action items
Prepare	• Define a vulnerability management plan. • Distribute ownership of findings. • Develop vulnerability disclosure program. • Develop an AWS account structure. • Define, implement, and enforce tags. • Monitor AWS security bulletins. • Enable Amazon Inspector with a delegated administrator. • Enable Security Hub CSPM with a delegated administrator. • Enable Security Hub CSPM standards. • Set up Security Hub CSPM cross-Region aggregation. • Enable consolidated control findings in Security Hub CSPM.

Phase	Action items
	• Set up and manage Security Hub CSPM integrations, including applicable downstream integrations with SIEM, GRC, or product backlog or ticketing systems
Triage and remediate	• Route findings based on multi-account strategy. • Route findings to security, cloud, and application or developer teams. • Tune security findings to make sure that they are actionable for your specific environment. • Develop automated remediation mechanisms, when possible. • Implement CI/CD pipeline controls or other guardrails that help prevent security findings, when possible. • Use Security Hub CSPM automation rules to escalate or suppress findings.
Report and improve	• Hold monthly security operations meetings. • Use Security Hub CSPM insights to identify anti-patterns.

Resources

AWS service documentation

- Product integrations (AWS Security Hub CSPM)
- Integrating AWS Security Hub CSPM in Jira Service Management Cloud (AWS Security Hub CSPM)
- Automation rules (AWS Security Hub CSPM)
- Proactive evaluation rules (AWS Config)
- Patch Manager (AWS Systems Manager)

Other AWS resources

- Best practices for tagging AWS resources (AWS whitepaper)
- Automated Security Response on AWS (AWS Solutions Library)
- AWS Security Incident Response Guide (AWS Technical Guide)
- AWS security bulletins

Document history

The following table describes significant changes to this guide. If you want to be notified about future updates, you can subscribe to an RSS feed.

Change	Description	Date
Initial publication	—	October 12, 2023