AWS Security Hub Partner Integration Use Cases & Guidance

Overview

Security Hub (SH) is an AWS Security External Security Service (ESS) product that was launched in public preview at re:Invent 2018. It is AWS's security and compliance center and a key feature is aggregation and prioritization of *findings* from services such as Inspector, Macie, and GuardDuty as well as a much larger ecosystem of partner-provided *products*. These partner products may be run within the customers' own accounts or the product owner's account. Products import customer findings into SH on behalf of the product's customers. This document covers key concepts and use cases around enabling partner products to have access to interact with SH on behalf of a customer.

Concepts

- **Findings** are point-in-time observations of security issues or compliance evaluations. Findings contain the details of these observations and are represented in the AWS Security Finding Format (ASFF). *Products* import customers' findings into SH on behalf of the product's customers. Findings are aggregated to provide customers with *insights* (answers to higher-level questions through finding aggregation) into their security posture. Findings are the primary resource stored in SH. SH will store millions of findings for our customers per region with a 90-day retention period.

- **Product** is a *Company*-owned resource that represents a system that generates and provides findings for customers. Products are systems that either run in the product owner's account(s) or have been deployed within the customer's account(s). Products may import findings in SH in near real-time or in batch.

- **ProductSubscription** is a customer-owned resource that represents the result of a customer subscribing to a product. This resource is created by SH on the customer's account at the time of subscription to the product. This resource is deleted when a customer unsubscribes from a product. The ProductSubscription effectively models a container of all findings generated by a specific product within customers accounts.

- **Company** is a resource that owns product(s). This acts as a product aggregator for the purposes of managing multiple products under a single entity. AWS partners are examples of a Company.

- **Managed Resource Policy** is a similar concept to IAM Managed Policies. SH fully controls the content of the ProductSubscription resource policies with the customer's consent. The purpose is two-fold, 1) to ensure access is correctly limited whenever possible to mitigate finding spoofing risks and 2) to simplify the goal of allowing/denying a product to import findings on the customer's behalf. While SH controls the policy, the customer can read the policy and consent to apply or remove it by the create/delete ProductSubscription processes.

- **Partner Product Account** is an AWS account where an AWS partner runs the infrastructure, services, and applications that power a partner's product. The application in the partner account is often made available to customers in a Software as a Service (SaaS) model.
- **Customer Account** is the AWS account where a customer is running their infrastructure and applications.
- **Cross Account Role** is an AWS Identity and Access Management (IAM) Feature that allows a principal in one AWS account to access AWS APIs in another account. These roles allow for the use of temporary credentials removing the need to provide long term credentials tied to named IAM users. See AWS documentation on Providing Access to Accounts Owned by Third Parties for more details.

Security Hub Integration Use Cases

There are several different use cases that would enable an AWS customer to receive SH findings from a partner via the BatchImportFindings API. The GetFindings API use case is not discussed in detail in this document, but a customer would also provide a partner with access to the GetFindings API via an IAM cross-account role that is created and maintained by the customer outside of SecurityHub.

SH allows customers to receive findings from partners who offer products that run inside the customer's account and outside of the customer's account. The permission configuration in the customer's account will differ depending on the model that the partner product is using.

With SH the customer is in control at all times for which partners are allowed to send findings into their account and will be able to revoke permissions to a partner at any time.

For all customers wishing to enable a partner to send security findings into their account, the first step is to subscribe to the partner product in Security Hub. This step is necessary for all the use cases that are outlined below. Subscribing to the partner product can be done in the SH console or via an API call. Once a customer subscribes to a partner product in SH, SH automatically creates a Managed Resource Policy grants the partner product permissions to send findings, via the BatchImportFindings API, into SH for the customer's account.

Below are three common use cases for partner products integrating with SH and the additional permissions needed for that use case to work.

Partner Hosted (SaaS) – Findings sent from partner account

This use case covers partners who host a product in their own AWS account and will make the call to the BatchImportFindings API, to send security findings for an AWS customer, from the partner product account.

For this use case no additional permissions are needed in the customer account beyond what is put in place when the customer subscribes to the partner product in SH. In the partner account the IAM principal that is calling the BatchImportFindings API will need to have sufficient permissions, via an IAM policy, which allow the principal to call the BatchImportFindings API for SH.

Below are the steps that a customer would go through to enable the partner product to send findings to the customer in SH.

Configuration:

1. The customer creates a subscription to a partner product SH.
2. SH generates the correct managed resource policy with the customer's confirmation.

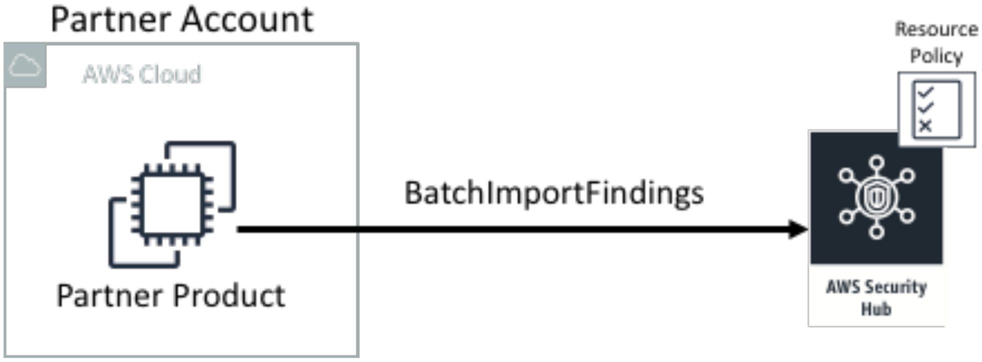
Import findings:

1. The partner product calls BatchImportFindings on SH with their own credentials to send security findings related to the customer’s account.

This is a sample of what an IAM policy which grants the principal in the partner account the necessary Security Hub permissions might look like:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "securityhub:BatchImportFindings",
            "Resource": "arn:aws:securityhub:us-west-2:*:product-subscription/threatslayer/threatslayer-pro"
        }
    ]
}
```

Here is a visual interpretation of how this use case would look:



Partner Hosted (SaaS) – Findings sent from customer account

This use case covers partners who host a product in their own AWS account but use a cross-account role to access the customer's account and need to call the BatchImportFindings API from the customer's account.

For this use case, the partner account assumes a customer-managed IAM role in the customer's account to call the BatchImportFindings API. As the call is made from the customer's account, the Managed Resource Policy allows the partner product account's ProductArn to be used in the call. The reason this additional permission definition is needed is that the SH managed resource permission grants permission for the partner product account and the partner product ARN, which is their unique identifier as a provider. Since the call is not happening from the partner product account the customer must explicitly grant permission to allow the partner product, who is accessing their account, to send findings to Security Hub.

Best practice for cross-account roles between partner and customer accounts is to utilize an external ID which is provided by the partner. This external ID will be part of the cross-account policy definition in the customer's account and is required to be provided by the partner as part of assuming the role. External IDs provide an additional layer of security when granting AWS account access to a partner and provides a unique identifier for a customer to ensure that the partner is accessing the correct customer account.

Below are the steps that a customer would go through to enable the partner product to send findings to the customer in SH, with a cross-account role.

Configuration:

1. The customer (or partner using cross-account roles working on behalf of the customer) starts the subscription to a product in SH.
2. SH generates the correct managed resource policy with the customer's confirmation.
3. The customer must configure the cross-account role either via CloudFormation or manually.
4. The product must securely store the customer role and external ID.

Import findings:

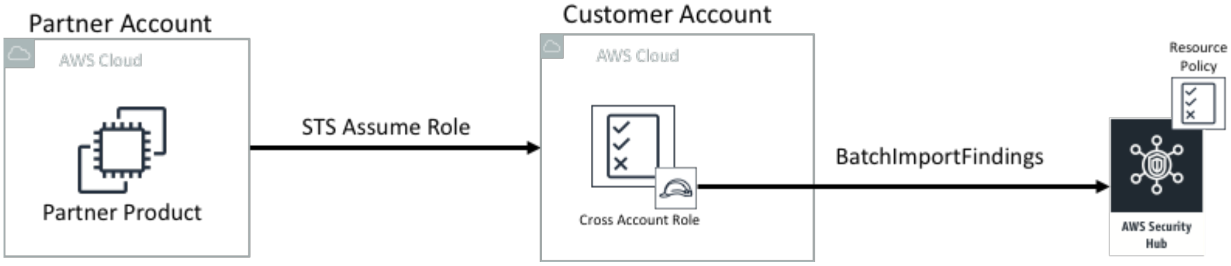
1. The product calls the AWS Security Token Service (STS) to assume the customer role.
2. The product calls BatchImportFindings on SH with the assume role's temporary credentials.

This is a sample of what an IAM policy granting the necessary Security Hub permissions to the partner’s cross account role might look like:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "securityhub:BatchImportFindings",
            "Resource": " arn:aws:securityhub:us-west-
2:012345678910:product-subscription/threatslayer/threatslayer-pro"
        }
    ]
}
```

The use of the specific product subscription in the resource section of the policy ensures that the partner can only send findings for the partner product that the customer is subscribed to.

Visually this use case is represented below:



Customer hosted partner solution – Findings sent from customer account

This use case covers partners who have a product that is deployed in the customer's AWS account and the BatchImportFindings API will be called from the solution running in the customer's account.

For this use case the partner product needs to be granted additional permissions to call the BatchImportFindings API. The approach to how this permission is granted will differ based on the partner solution and how it is architected in the customer's account.

An example of this approach would be a partner product that is running on an EC2 instance in the customer's account. This EC2 instance will need to have an EC2 instance role attached to it which grants that instance the ability to call the BatchImportFindings API to send security findings into the customer's account.

This use case is functionally equivalent to a customer loading findings into their own account for one of their own customer-owned products.

Below are the steps that a customer would go through to enable the partner product to send findings to the customer in SH, from the customer's account.

Configuration:

1. The customer deploys the partner product into their AWS account manually, via CloudFormation, or another deployment tool.
2. The customer defines the necessary IAM policy for the partner product to use when sending findings to SH.
3. The customer attaches the policy to the necessary components of the partner product (EC2 instance, container, lambda function, etc.).

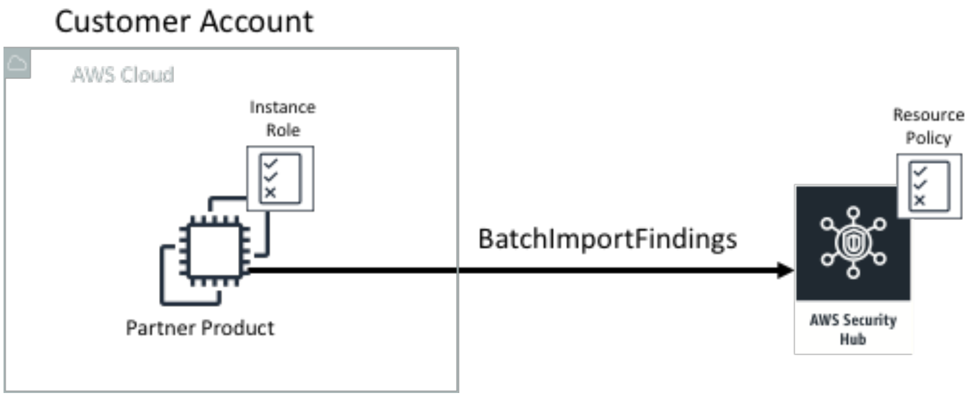
Import findings:

4. The partner product, using the AWS SDK or CLI makes a call to BatchImportFindings in SH from the component in the customer's account where the policy was attached.
5. During the API call the necessary temporary credentials are vended to allow the BatchImportFindings call to succeed.

This is a sample of what an IAM policy granting the necessary Security Hub permissions to the partner product in the customer account might look like:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "securityhub:BatchImportFindings",
            "Resource": "arn:aws:securityhub:us-west-2:012345678910:product-subscription/threatslayer/threatslayer-pro"
        }
    ]
}
```

Visually this use case is represented below:


Customer Account
AWS Cloud
Instance Role
Partner Product
BatchImportFindings
Resource Policy
AWS Security Hub