Introduction to the Splunk Add-on for Amazon Security Lake

Version	0.0.4
Vendor products	Amazon Security Lake
Add-on has a web UI	Yes. This add-on contains views for configuration.

Use the Splunk Add-on for Amazon Security Lake to ingest security events from AWS services normalized to the Open Cybersecurity Schema Framework (OCSF) schema. These events will be made available as multi-event Apache Parquet objects in an S3 bucket. Each object will have a corresponding SQS notification, once ready for download. The ASDL - Splunk Add-on for AWS will periodically check for new SQS notifications, download new objects, convert the files from Parquet to JSON and index each event into your Splunk platform deployment. The indexed events will be searchable using the OCSF schema.

Splunk Add-On for Amazon Security Lake is only compatible with Splunk running upon Linux 64-bit. This is primarily because the add-on requires installation of Python for Scientific Computing (Linux 64-bit), please ensure it is installed before installing this add-on.

Download the Splunk Add-on for Amazon Security Lake from Splunkbase. See Configure the Splunk Add-on for Amazon Security Lake for information about configuring this add-on.

See Release notes for the Splunk Add-on for Amazon Security Lake for a summary of new features, fixed issues, and known issues.

Source types for the Splunk Add-on for Amazon Security Lake

The Splunk Add-on for Amazon Security Lake collects Amazon security events and transforms to JSON before indexing.

See Troubleshoot the Splunk Add-on for Amazon Security Lake to find source types for internal logs.

See the following table for source types and event types for AWS data mapping:

Pull-based API data collection sourcetypes

Data type	Source type	Description	input types	Data models
Amazon Security Lake events	aws:asl	`aws:asl` represents AWS API dataset data collection from AWS CloudTrail management events, VPC flow logs, Route 53 Resolver query logs, S3 data events, Lambda function execution activity. Individual services can be searched using category field.	SQS-based S3	• CIM: None • ES Custom: None • ITSI: None

Hardware and software requirements for the Splunk Add-on for Amazon Security Lake

To install and configure the Splunk Add-on for Amazon Security Lake, you must have the `admin` or `sc_admin` role permissions.

Splunk platform requirements

Splunk Add-on for Amazon Security Lake can only be used with a Splunk installation running on Linux 64-bit. To use this add-on, you must download and install the Linux - Python for Scientific Computing (for Linux 64-bit) Python interpreter on every Splunk platform instance where the Splunk Add-on for Amazon Security Lake will be installed.

For Splunk Enterprise system requirements, see System requirements for use of Splunk Enterprise on-premises in the Splunk Enterprise Installation Manual. For information about installation locations and environments, see the **Installation overview** topic in this manual.

AWS account prerequisites

To set up your AWS configuration to work with your Splunk platform instance, make sure you have met the following AWS prerequisites.

Account privileges

Verify that you have the following account privileges:

- A valid AWS account with permissions to configure the AWS services that provide your data.
- Permission to create Identity and Access Management (IAM) roles and users. This lets you set up AWS account IAM roles or Amazon Elastic Compute Cloud (EC2) IAM roles to collect data from your AWS services.

When configuring your AWS account to send data to your Splunk platform deployment, the best practice is that you should not allow `"*"` (all resource) statements as part of action elements. This level of access could potentially grant unwanted and unregulated access to anyone given this policy document setting. The best practice is to write a refined policy describing the specific action allowed by specific users or specific accounts, or required by the specific policy holder.

For more information, see the Basic examples of Amazon SQS policies topic in the Amazon Simple Queue Service Developer Guide.

AWS region limitations

This add-on is intended to run in the following AWS regions:

- US East (Ohio) (us-east-2)
- US East (N. Virginia) (us-east-1)
- US West (Oregon) (us-west-2)
- Asia Pacific (Sydney) (ap-southeast-2)
- Asia Pacific (Tokyo) (ap-northeast-1)
- Europe (Frankfurt) (eu-central-1)
- Europe (Ireland) (eu-west-1)

Network configuration requirements

The Splunk Add-on for Amazon Security Lake makes REST API calls using HTTPS on port 443. Data inputs for this add-on use large amounts of memory. See Sizing, performance, and cost considerations for the Splunk Add-on for Amazon Security Lake for more information.

AWS encryption requirements

Amazon Web Services supports the following server-side encryption types:

- Server-side encryption with Amazon S3-managed encryption keys (SSE-S3). For SSE-S3 configurations, the unique key is used for encrypting each object)
- Server-side encryption with AWS Key Management Service (SSE-KMS). SSE-KMS will manage encryption. AWS will manage the master key.
- Server-side encryption with customer-provided encryption keys (SSE-C). KMS service will manage encryption/ The client needs to provide a custom master key.

The Splunk Add-on for Amazon Security Lake uses all server-side encryptions. Client-side encryption is not used. Server side encryption is handled by AWS. AWS SDK for Python does not support client-side encryption.

Provision a subscriber for the Splunk Add-on for Amazon Security Lake

Perform the following command line interface steps to set up a subscriber for the Splunk Add-on for Amazon Security Lake. This creates the resources needed to make the Amazon Security Lake events available to be consumed into your Splunk platform deployment.

Create a subscriber

Perform the following CLI command to create a subscriber:

[cloudshell-user@ip-10-0-0-142 ~]$ aws securitylake create-subscriber --account-id <> --external-id <placeholder_text> --subscriber-name <subscriber-name> –access-types S3 --source-types awsSourceType=<"ROUTE53" or "VPC_FLOW" or "CLOUD_TRAIL" or "SH_FINDINGS">

CLI Output:

{

"roleArn": "<IAM role ARN>",

"s3BucketArn": "<S3 ARN>",

"subscriptionId": "<subscription ID>"

}

Create an SQS queue

Perform the following CLI command to create an SQS Queue

aws securitylake create-subscription-notification-configuration --create-sqs --subscription-id <subscription ID from above command>

CLI Output:

```
{

  "queueArn": "arn:aws:"

}
```

For more information, see the *create-subscriber* and *create-subscription-notification-configuration* topics in the AWS CLI Command Reference manual.

https://awscli.amazonaws.com/v2/documentation/api/latest/reference/securitylake/create-subscriber.html

https://awscli.amazonaws.com/v2/documentation/api/latest/reference/securitylake/create-subscription-notification-configuration.html

Verify information in SQS Queue

Perform the following steps in your Amazon deployment to verify the information in the SQS Queue that Security Lake creates.

1. In your AWS console, navigate to the Amazon SQS service.
2. In the **Queues** section, navigate to the SQS Queue that Security Lake created, and click on the name.
3. On the information page for the SQS Queue that Security Lake created, perform the following validation steps.
    1. Click on the **Monitoring** tab to verify that events are flowing into the SQS Queue.
    2. Click on the **Dead-letter queue** tab to verify that a dead-letter queue (DLQ) has been created. If a DLQ has not been created, see the **Configuring a dead-letter queue (console)** topic in the AWS documentation.

Verify events are flowing into S3 bucket

Perform the following steps in your Amazon deployment to verify that parquet events are flowing into your configured S3 buckets.

1. In your AWS console, navigate to the Amazon S3 service.
2. Navigate to the **Buckets** section, and click on the S3 bucket that Security Lake created for each applicable region.
3. In each applicable bucket, navigate to the **Objects** tab, and click through the directories to verify that Security Lake has available events flowing into the S3 bucket. If Security Lake is enabled on more than one AWS account, check to see if each applicable account number is listed, and that parquet files exist inside each account.
4. In each applicable S3 bucket, navigate to the **Properties** tab.
5. Navigate to **Event notifications**, and verify that the Security Lake SQS Queue that was created has event notifications turned on, and the data destination is the Security Lake SQS queue.

Configure IAM policies

After you set up and configured a subscriber in the Amazon Security Lake service, perform the following modifications to your IAM policies to make the Splunk Add-on for Amazon Security Lake work:

1. Update a user to assume a role. Then modify the assumed role so that it doesn't reference an **External ID**.
2. Update your boundary policy to work with the Splunk Add-on for Amazon Security Lake.

Update a user to assume a role

Modify your Security Lake subscriber role to associate an existing user with a role, and modify the assumed role so that it doesn't reference an External ID. You must get access to the subscription role notification that was created as part of the Amazon Security Lake subscriber provisioning.

1. In your AWS console, navigate to the Amazon IAM service.
2. In your Amazon IAM service, navigate to the **Roles** page.
3. On the **Roles** page, select the Role name of the subscription role notification that was created as part of the Security Lake subscriber provisioning process.
4. On the Summary page, navigate to the **Trust relationships** tab.
5. Modify the **Trusted entity** policy with the following updates:
    1. Remove any reference to the **External ID** that was created during the Security Lake subscriber provisioning process.

2. On the stanza containing the ARN, Attach the username from your desired user account to the end of the ARN. For example, `"arn:aws:iam:772039352793:user/jdoe"`, where `jdoe` is the user name.

For more information, see the following example Trust entity:

```
{

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

"Statement": [

{

"Sid": "1",

"Effect": "Allow",

"Principal": {

"AWS": "arn:aws:iam::772039352793:user/jdoe"

},

"Action": "sts:AssumeRole"

}

]

}
```

3. This step connects a user to the role that was created, and lets a user take their secret key access key to then configure the Security Lake service.

6. In your Amazon IAM service, navigate to the **Users** page.
7. On the **Users** page, select the User name of the user who has been connected to the role that was created.
8. On the **Summary** page, navigate to the **Access keys** section, and copy the user's **Access key ID**. If no access keys currently exist, first click the **Create access key** button.

Update your boundary policy to work with the Splunk Add-on for Amazon Security Lake

1. In your Amazon IAM service, navigate to the **Roles** page.
2. On the **Roles** page, select the Role name of the subscription role notification that was created as part of the Security Lake subscriber provisioning process.
3. On the Summary page, navigate to the **Permissions policies** tab, and click on the **Policy name** for your Amazon Security Lake subscription role, in order to modify the role policy.
4. On the **Edit policy** page, click on the **JSON** tab.
5. Navigate to the **Resource** column of the role policy.
6. Under the existing S3 resources stanzas, add a stanza containing the Amazon Resource Name (ARN) of the SQS Queue that was created during the Security Lake service subscriber provisioning process.
7. Navigate to the **Action** column of the role policy.
8. Review the contents of the **Action** column, and add the following stanzas, if they do not already exist:

```
"sqs:GetQueueUrl",

"sqs:ReceiveMessage",

"sqs:SendMessage",

"sqs:DeleteMessage",

"sqs:GetQueueAttributes",

"sqs:ListQueues",
```

For more information, see the following example:

```
{

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

"Statement": [

{

"Sid": "1",

"Effect": "Allow",

"Action": [

"s3:GetObject",
```

```
"s3:GetObjectVersion",
"sqs:GetQueueUrl",
"sqs:ReceiveMessage",
"sqs:SendMessage",
"sqs:DeleteMessage",
"sqs:GetQueueAttributes",
"sqs:ListQueues",
"s3:ListBucket",
"s3:ListBucketVersions",
"kms:Decrypt"
],
"Resource": [
"arn:aws:s3:::aws-security-data-lake-us-east-2-o-w5jts1954e/aws/CLOUD_TRAIL/*",
"arn:aws:s3:::aws-security-data-lake-us-east-2-o-w5jts1954e/aws/VPC_FLOW/*",
"arn:aws:s3:::aws-security-data-lake-us-east-2-o-w5jts1954e/aws/ROUTE53/*",
"arn:aws:s3:::aws-security-data-lake-us-east-2-o-w5jts1954e",
"arn:aws:sqs:us-east-2:772039352793:splunk-public-sqs"
]
}
]
}
```

9. Save your changes.

Sizing, performance, and cost considerations for the Splunk Add-on for Amazon Security Lake

Before you configure the Splunk Add-on for Amazon Security Lake, review these sizing, performance, and cost considerations. Inputs running on search heads and search head clusters in the Splunk Cloud Platform Victoria Experience architecture scale up and down as required to handle inputs and search load.

General

The following table provides general guidance on sizing, performance, and cost considerations for the Amazon Security Lake data input:

Splunk Platform Environment	Number of indexers	Inputs (configured to batch size 1)	Instance Sizing	Max Throughput Indexed	Expected Average Throughput Indexed
Splunk Cloud Platform Classic Experience	1	1	4G 1 vCPU (t4g.medium)	15 GB/h	12GB/h
Splunk Cloud Platform Classic Experience	3 or more	1	8G 2 vCPU (t4g.large)	60GB/h	30GB/h
Splunk Cloud Platform Classic Experience	6 or more	2	8G 2 vCPU (t4g.large)	146GB/h	70GB/h

Splunk Cloud Platform Classic Experience	11 or more	5	16G 4 vCPU (t4g.xlarge)	243GB/h	150GB/h
Splunk Cloud Platform Victoria Experience		1	N/A	72GB/h	30GB/h
Splunk Cloud Platform Victoria Experience		2	N/A	144GB/h	70GB/h
Splunk Cloud Platform Victoria Experience		5	N/A	360GB/h	150GB/h
Splunk Enterprise	3 or more	1	8G 4 Core	72GB/h	30GB/h
Splunk Enterprise	6 or more	2	8G 4 Core Forwarder x 1	150GB/h	70GB/h
Splunk Enterprise	11 or more	5	16G 8 Core Forwarder x 1	260GB/h	150GB/h

Splunk Enterprise	24 or more	10 inputs across 2 machines	16G 8 Core Forwarder x 2	520GB/h	300GB/h

Splunk Enterprise indexing will be limited to available indexer throughput. For more information, see the **Summary of performance recommendations** topic in the Capacity Planning Manual.

Adding a new instance can be calculated by adding each instances' throughput together, as long as indexers are sized accordingly. Splunk Enterprise deployments will be similar in sizing to Splunk Cloud Platform Classic Experience. Input configuration screens require data transfer from AWS to populate the services, queues, and buckets available to your accounts. If your network to AWS is slow, data transfer might be slow to load. If you encounter timeout issues, you can manually type in resource names.

Deploy the Splunk Add-on for Amazon Security Lake

Complete the following steps to deploy the Splunk Add-on for Amazon Security Lake:

1. Install the Splunk Add-on for Amazon Security Lake.
2. Manage accounts for the Splunk Add-on for Amazon Security Lake.
3. Configure inputs for the Splunk Add-on for Amazon Security Lake.

Installation overview for the Splunk Add-on for Amazon Security Lake

1. Download the Splunk Add-on for Amazon Security Lake from Splunkbase.
2. Use the tables in this topic to determine where to install this add-on.
3. Perform any prerequisite steps specified in the tables before installing.
4. Use the link in the Installation walkthrough section to perform the installation.

Distributed deployments

Use the following tables to install the Splunk Add-on for Security Lake in a deployment that uses forwarders to get data in, such as a distributed deployment. You might need to install the add-on in multiple places.

Where to install this add-on

Unless otherwise noted, you can safely install all add-ons to all tiers of a distributed Splunk platform deployment. See Where to install Splunk add-ons in *Splunk Add-ons* for more information.

This table provides a reference for installing this specific add-on to a distributed deployment of the Splunk platform:

	Recommended	Required	
Search heads	Yes	No	Data inputs for this add-on require large amounts of memory. See Hardware and software requirements for the Splunk Add-on for Amazon Security Lake.
Indexers	Yes	No	Not required as the parsing operations occur on the heavy forwarders.
Heavy forwarders	Yes	Yes	This add-on requires heavy forwarders to perform data collection through modular inputs and to perform the setup and authentication with AWS in Splunk Web.
Universal forwarders	No	No	This add-on requires heavy forwarders.

Distributed deployment compatibility

This table provides a quick reference for the compatibility of this add-on with Splunk distributed deployment features:

	Recommended	
Search head clusters	Yes	You can install this add-on on a search head cluster for all search-time functionality, but configure inputs on forwarders to avoid duplicate data collection. Before installing this add-on to a cluster, make the following changes to the add-on package: #Remove the eventgen.conf files and all files in the samples folder. 1. Remove the inputs.conf file.
Indexer clusters	Yes	Before installing this add-on to a cluster, make the following changes to the add-on package: 1. Remove the inputs.conf file.
Deployment server	No	Deployment servers support deploying unconfigured add-ons only. ● Using a deployment server to deploy the configured add-on to multiple forwarders acting as data collectors causes duplication of data. ● The add-on uses the credential vault to secure your credentials, and this

	credential management solution is incompatible with the deployment server.

Installation walkthroughs

See the following links, or About installing Splunk add-ons in the *Splunk Add-Ons* manual, for an installation walkthrough specific to your deployment scenario:

- Install the Splunk Add-on for Amazon Security Lake

= Install the Splunk Add-on for Amazon Security Lake=

Perform the following applicable steps to install the Splunk Add-on for Amazon Security Lake on your Splunk platform deployment.

<note>The Splunk Add-on for Amazon Security Lake package size is greater than 115MB. File upload and installation might take a few minutes.</note>

<note>Verify that you have downloaded and installed the Python for Scientific Computing Python interpreter on every Splunk platform instance where the Splunk Add-on for Amazon Security Lake will be installed.</note>

==Install the Splunk Add-on for Amazon Security Lake in a Splunk Cloud Platform Deployment==

Install the Splunk Add-on for Amazon Security Lake to your Splunk Cloud Platform instance using the app browser in Splunk Cloud Platform:

#From the Splunk Web home screen, click on the gear icon next to '"Apps"' in the navigation bar.
#Click '"Browse more apps"'.
#Find the Splunk Add-on for Amazon Security Lake, then click '"Install"'.
#Follow the on-screen prompts to complete your installation.
#Install the Splunk Add-on for Amazon Security Lake to an Inputs Data Manager. Request that [https://www.splunk.com/en_us/about-splunk/contact-us.html#tabs/tab_parsys_tabs_CustomerSupport_4 Splunk Cloud Platform Support] installs the Splunk Add-on for Amazon Security Lake on your Splunk Cloud Platform instance.

==Install the Splunk Add-on for Amazon Security Lake in a single-instance Splunk Enterprise deployment==

Follow these steps to install the Splunk Add-on for Amazon Security Lake in a single-instance deployment:
#From the Splunk Web home screen, click the gear icon next to '"Apps"' in the navigation bar.
#Click '"Install app from file"'.
#Locate the downloaded file and click '"Upload"'.
#If Splunk Enterprise prompts you to restart, do so.
#Verify that the add-on appears in the list of apps and add-ons. You can also find it on the server at <code>$SPLUNK_HOME/etc/apps/Splunk_TA_amazon_security_lake</code>.

==Install the Splunk Add-on for Amazon Security Lake in a distributed Splunk Enterprise deployment==

If you are using a distributed Splunk Enterprise deployment, follow the instructions in each of the following sections to deploy the Splunk Add-on for Amazon Security Lake to your search heads, indexers, and forwarders. You must install Splunk Add-on for Amazon Security Lake on a heavy forwarder. You cannot use this add-on with a universal forwarder. You can install this add-on onto search heads and indexers.

===Heavy forwarders===

To install the Splunk Add-on for Amazon Security Lake to a heavy forwarder, follow these steps:

#Download the Splunk Add-on for Amazon Security Lake, if you have not already done so.
#From the Splunk Web home screen on your heavy forwarder, click the gear icon next to '"Apps"'.
#Click '"Install app from file"'.
#Locate the downloaded file and click '"Upload"'.
#If the forwarder prompts you to restart, do so.
#Verify that the add-on appears in the list of apps and add-ons. You can also find it on the server at <code>$SPLUNK_HOME/etc/apps/Splunk_TA_amazon_security_lake</code>.

===Search heads===

To install the Splunk Add-on for Amazon Security Lake to a search head, follow these steps:

#Download the Splunk Add-on for Amazon Security Lake, if you have not already done so.
#From the Splunk Web home screen, click the gear icon next to '"Apps"'.
#Click '"Install app from file"'.
#Locate the downloaded file and click '"Upload"'.
#If Splunk Enterprise prompts you to restart, do so.
#Verify that the add-on appears in the list of apps and add-ons.

<note>Make sure the add-on is not visible. If the Visible column for the add-on is set to '"Yes"', edit the properties and change the visibility to '"No."' Disable visibility of add-ons on search heads to avoid inputs from being created on search heads. Data collection for search heads might conflict with users' search activity.</note>

You can also find the add-on on the server at <code>$SPLUNK_HOME/etc/apps/Splunk_TA_amazon_security_lake</code>.

===Search head clusters===

Before deploying the Splunk Add-on for Amazon Security Lake to a search head cluster, make the following changes to the add-on package:

#Remove the eventgen.conf files and all files in the samples folder, if they exist.
#Remove the inputs.conf and inputs.conf.spec files. If you are collecting data locally from the machines running your search head nodes, keep these files.
#Use the deployer to deploy an add-on to the search head cluster members.

See [https://docs.splunk.com/Documentation/Splunk/8.0.5/DistSearch/PropagateSHCconfigurationchanges Use the deployer to distribute apps and configuration updates] in the Splunk Enterprise "Distributed Search" manual.

===Indexers===

To install the Splunk Add-on for Amazon Security Lake to an indexer, follow these steps:

#Download the Splunk Add-on for Amazon Security Lake, if you have not already done so.
#Unpack the .tgz package.
#Place the resulting <code>Splunk_TA_amazon_security_lake</code> folder in the <code>$SPLUNK_HOME/etc/apps</code> directory on your indexer.
#Restart the indexer.

===Indexer clusters===

#Remove the eventgen.conf files and all files in the samples folder, if they exist.
#Remove the inputs.conf and inputs.conf.spec files. If you are collecting data locally from the machines running your indexer nodes, keep these files.
#Deploy add-ons to peer nodes on indexer clusters using a master node.

For more information about using a master node to deploy to peer nodes of an indexer cluster, see [https://docs.splunk.com/Documentation/Splunk/8.0.5/Indexer/Manageappdeployment Manage app deployment across all peers] in "Managing Indexers and Clusters of Indexers".

Configure Amazon Security Lake inputs for the Splunk Add-on for Amazon Security Lake

Complete the steps to configure Amazon Security Lake inputs for the Splunk Add-on for Amazon Security Lake:

1. Configure AWS accounts for the Amazon Security Lake input.
2. Configure Amazon Security Lake inputs either through Splunk Web or configuration files.

Configuration prerequisites

This data input supports the following compression types:

- Apache Parquet file format.

Configure AWS accounts for the Amazon Security Lake input

Add your AWS account to the Splunk Add-on for Amazon Security Lake

1. On the Splunk Web home page, click on **Splunk Add-on for Amazon Security Lake** in the navigation bar.
2. Navigate to the **Configuration** page,
3. On the **Configuration** page, navigate to the **Account** tab.
4. Click the **Add** button.
5. On the **Add Account** page, add a **Name**, the **Key ID** of the user who was given Security Lake configuration privileges, **Secret Key**, and **Region Category**.
6. Click the **Add** button.
7. Navigate to the **IAM Role** tab.
8. Click the **Add** button.
9. Add the ARN role that was created during the Security Lake service provisioning process.
10. Click the **Add** button.

Configure an Amazon Security Lake input using Splunk Web

To configure inputs in Splunk Web, click **Splunk Add-on for Amazon Security Lake** in the navigation bar on Splunk Web home, then choose one of the following menu paths depending on which data type you want to collect:

- **Create New Input > Amazon Security Lake > SQS-Based S3**

You must have the admin_all_objects role enabled in order to add new inputs.

Choose the menu path that corresponds to the data type you want to collect. The system automatically sets the source type and display relevant field settings in the subsequent configuration page.

Use the following table to complete the fields for the new input in the .conf file or in Splunk Web:

`aws_account`	AWS Account	The AWS account or EC2 IAM role the Splunk platform uses to access the keys in your S3 buckets. In Splunk Web, select an account from the drop-down list. In inputs.conf, enter the friendly name of one of the AWS accounts that you configured on the Configuration page or the name of the automatically discovered EC2 IAM role. If the region of the AWS account you select is **GovCloud**, you may encounter errors such as"Failed to load options for S3 Bucket". You need to manually add **AWS GovCloud**

		Endpoint in the **S3 Host Name** field. See http://docs.aws.amazon.com/govcloud-us/latest/UserGuide/using-govcloud-endpoints.html for more information.
`aws_iam_role`	Assume Role	The IAM role to assume.
`using_dlq`	Force using DLQ (Recommended)	Check the checkbox to remove the checking of DLQ (Dead Letter Queue) for ingestion of specific data. In inputs.conf, enter `0` or `1` to respectively disable or enable the checking. (Default value is `1`)
`sqs_queue_region`	AWS Region	AWS region that the SQS queue is in.
`private_endpoint_enabled`	Use Private Endpoints	Check the checkbox to use private endpoints of AWS Security Token Service (STS) and AWS Simple Cloud Storage (S3) services for authentication and data collection. In inputs.conf, enter `0` or `1` to respectively disable or enable use of private endpoints.

`sqs_private_endpoint_url`	Private Endpoint URL (SQS)	Private Endpoint (Interface VPC Endpoint) of your SQS service, which can be configured from your AWS console.
		
		Formats :
		
		`<http/https>://vpce-<endpoint_id>-<unique_id>.sqs.<region_id>.vpce.amazonaws.com`
		`<http/https>://vpce-<endpoint_id>-<unique_id>-<availability_zone>.sqs.<region_id>.vpce.amazonaws.com`

`s3_private_endpoint_url`	Private Endpoint URL (S3)	Private Endpoint (Interface VPC Endpoint) of your S3 service, which can be configured from your AWS console.
		
		Formats :
		
		`<http/https>://vpce-<endpoint_id>-<unique_id>.s3.<region_id>.vpce.amazonaws.com`
		`<http/https>://vpce-<endpoint_id>-<unique_id>-<availability_zone>.s3.<region_id>.vpce.amazonaws.com`

`sts_private_endpoint_url`	Private Endpoint URL (STS)	Private Endpoint (Interface VPC Endpoint) of your STS service, which can be configured from your AWS console.
		
		Formats :
		
		`<http/https>://vpce-<endpoint_id>-<unique_id>.sts.<region_id>.vpce.amazonaws.com`
		`<http/https>://vpce-<endpoint_id>-<unique_id>-<availability_zone>.sts.<region_id>.vpce.amazonaws.com`

`sqs_queue_url`	SQS Queue Name	The SQS queue URL.
`sqs_batch_size`	SQS Batch Size	The maximum number of messages to pull from the SQS queue in one batch. Enter an integer between 1 and 10 inclusive. Set a larger value for small files, and a smaller value for large files. The default SQS batch size is 10. If you are dealing with large files and your system memory is limited, set this to a smaller value.

`s3_file_decoder`	S3 File Decoder	The decoder to use to parse the corresponding log files. The decoder is set according to the **Data Type** you select. If you select a **Custom Data Type**, choose one from `Cloudtrail`, `Config`, `ELB Access Logs`, `S3 Access Logs`, or `CloudFront Access Logs`.
`sourcetype`	Source Type	The source type for the events to collect, automatically filled in based on the decoder chosen for the input.
`interval`	Interval	The length of time in seconds between two data collection runs. The default is 300 seconds.
`index`	Index	The index name where the Splunk platform puts the Amazon Security Lake data. The default is main.

Configure an Amazon Security Lake input using configuration files

When you configure inputs manually in inputs.conf, create a stanza using the following template and add it to `$SPLUNK_HOME/etc/apps/ASDL_Splunk_TA_aws/local/inputs.conf`. If the file or path does not exist, create it.

```
[aws_sqs_based_s3://test_input]
aws_account = account_name
interval = 300
private_endpoint_enabled = 0
s3_file_decoder = ASDL
sourcetype = aws:asdl
sqs_batch_size = 10
sqs_queue_region = us-west-1
sqs_queue_url =
https://sqs.us-west-1.amazonaws.com/772089552793/parquet-test-queue
using_dlq = 1
```

Some of these settings have default values that can be found in `$SPLUNK_HOME/etc/apps/ASDL_Splunk_TA_aws/default/inputs.conf`:

```
[aws_sqs_based_s3]
using_dlq = 1
```

The previous values correspond to the default values in Splunk Web, as well as some internal values that are not exposed in Splunk Web for configuration. If you copy this stanza to your `$SPLUNK_HOME/etc/apps/ASDL_Splunk_TA_aws//local` and use it as a starting point to configure your inputs.conf manually, change the `[aws_sqs_based_s3]` stanza title from `aws_sqs_based_s3` to `aws_sqs_based_s3://<name>` and add the additional parameters that you need for your deployment.

Valid values for `s3_file_decoder` are `CustomLogs`, `CloudTrail`, `ELBAccessLogs`, `CloudFrontAccessLogs`, `S3AccessLogs`, `Config`.

If you want to ingest custom logs other than the natively available AWS log types, you must set `s3_file_decoder = CustomLogs`. This setting lets you ingest custom logs into the Splunk platform instance, but it does not parse the data. To process custom logs into meaningful events, you need to perform additional configurations in props.conf and transforms.conf to parse the collected data to meet your specific requirements.

For more information on these settings, see `/README/inputs.conf.spec` under your add-on directory.

Automatically scale data collection with Amazon Security Lake inputs

With the Amazon Security Lake input type, you can take full advantage of the auto-scaling capability of the AWS infrastructure to scale out data collection by configuring multiple inputs to ingest logs from the same S3 bucket without creating duplicate events. This is particularly useful if you are ingesting logs from a very large S3 bucket and hit a bottleneck in your data collection inputs.

1. Create an AWS auto scaling group for your heavy forwarder instances where the SQS-based S3 inputs is running.
   To create an auto-scaling group, you can either specify a launch configuration or create an AMI to provision new EC2 instances that host heavy forwarders, and use bootstrap script to install the Splunk Add-on for Amazon Security Lake and configure SQS-based S3 Amazon Security Lake inputs. For detailed information about the auto-scaling group and how to create it, see http://docs.aws.amazon.com/autoscaling/latest/userguide/AutoScalingGroup.html.
2. Set CloudWatch alarms for one of the following Amazon SQS metrics:
   - **ApproximateNumberOfMessagesVisible**: The number of messages available for retrieval from the queue.
   - **ApproximateAgeOfOldestMessage**: The approximate age (in seconds) of the oldest non-deleted message in the queue.
3. For instructions on setting CloudWatch alarms for Amazon SQS metrics, see http://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/SQS_AlarmMetrics.html.
4. Use the CloudWatch alarm as a trigger to provision new heavy forwarder instances with SQS-based S3 inputs configured to consume messages from the same SQS queue to improve ingestion performance.

Troubleshoot the Splunk Add-on for Amazon Security Lake

Use the following information to troubleshoot the Splunk Add-on for Amazon Security Lake. For helpful troubleshooting tips that you can apply to all add-ons see Troubleshoot add-ons, and Support and resource links for add-ons in the *Splunk Add-ons* manual.

Data collection errors and performance issues

You can choose dashboards from the Health Check menu to troubleshoot data collection errors and performance issues.

The Health Overview dashboard gives you an at-a-glance view of the following data collection errors and performance metrics for all input types:

- Errors count by error category
- Error count over time by input type, host, data input, and error category
- Throughput over time by host, input type, and data input

The S3 Health Details dashboard focuses on the generic, incremental, and SQS-based S3 input types and provides indexing time lag and detailed error information of these multi-purpose inputs.

You can customize the health dashboard. See the About the Dashboard Editor topic in the *Dashboards and Visualizations* manual.

Configure log levels

1. Click **Splunk Add-on for Amazon Security Lake** in the navigation bar on Splunk Web.
2. Click **Configuration** in the app navigation bar.
3. Click the **Logging** tab.
4. Adjust the log levels for each of the AWS services as needed by changing the default level of `INFO` to `DEBUG` or `ERROR`.

These log level configurations apply only to runtime logs. Some REST endpoint logs from configuration activity log at DEBUG, and some validation logs log at ERROR. These levels cannot be configured.

Low throughput for the Splunk Add-on for Amazon Security Lake

If you do not achieve the expected AWS data ingestion throughput, follow these steps to troubleshoot the throughput performance:

1. Identify the problem in your system.
2. Adjust the factors affecting performance.
3. Verify whether performance meets your requirements.

1. Identify the problem in your system that prevents it from achieving a higher level of throughput performance. The problem in AWS data ingestion might be caused one of the following components:
    - The amount of data the Splunk Add-on for Amazon Security Lake can pull in through API calls
    - The heavy forwarder's capacity to parse and forward data to the indexer tier, which involves the throughput of the parsing, merging, and typing pipelines
    - The index pipeline throughput
2. To troubleshoot the indexing performance on the heavy forwarder and indexer, refer to Troubleshooting indexing performance in the Capacity Planning Manual.
3. Troubleshoot the performance of the problem component.
   If heavy forwarders or indexers are affecting performance, refer to the Summary of performance recommendations in the Splunk Enterprise *Capacity Planning* Manual.
   If the Splunk Add-on for AWS is affecting performance, adjust the following factors:
    - Parallelization settings
      To achieve optimal throughput performance, set the value of `parallelIngestionPipelines` to 2 in the server.conf file if your resource capacity permits. For information about `parallelIngestionPipelines`, see Parallelization settings in the Splunk Enterprise *Capacity Planning* Manual.
    - AWS data inputs
      If you have sufficient resources, you can increase the number of inputs to improve throughput, but be aware that this also consumes more memory and CPU. Increase the number of inputs to improve throughput until memory or CPU is running short.
      If you are using SQS-based S3 inputs, you can horizontally scale data collection by configuring more inputs on multiple heavy forwarders to consume messages from the same SQS queue.
    - File format
      Compressed files consume much more memory than plain text files.
4. When you resolve the performance issue, see if the improved performance meets your requirements. If not, repeat the previous steps to identify the next bottleneck in the system and address it until you're satisfied with the overall throughput performance.

Problem saving during account or input configuration

If you experience errors or trouble saving while configuring your AWS accounts on the setup page, go to `$SPLUNK_HOME/etc/system/local/web.conf` and and change the following timeout setting:

```
[settings]

splunkdConnectionTimeout = 300
```

Certificate verify failed (_ssl.c:741) error message

If you create a new input, you might receive the following error message:

`certificate verify failed (_ssl.c:741)`

Perform the following steps to resolve the error:

1. Navigate to `$SPLUNK_HOME/etc/auth/cacert.pem` and open the **cacert.pem** file with a text editor.
2. Copy the text from your deployment's proxy server certificate, and paste it into the **cacert.pem** file.
3. Save your changes.

Internet restrictions prevent add-on from collecting AWS data

If your deployment has a security policy that doesn't allow connection to the public internet from AWS virtual private clouds (VPCs), this might prevent the Splunk Add-on for Amazon Security Lake from collecting data from S3 inputs, and other inputs which depend on access to AWS services.

To identify this issue in your deployment:

1. Check if you have a policy that restricts outbound access to the public Internet from your AWS VPC.

Identify if you have error messages that show that your attempts to connect to sts.amazonaws.com result in a timeout. For example:

```
ConnectTimeout: HTTPSConnectionPool(host='sts.amazonaws.com',
port=443): Max retries exceeded


with url: / (Caused by
ConnectTimeoutError(<botocore.awsrequest.AWSHTTPSConnection object


at 0x7fdfd97bc350>,


'Connection to sts.amazonaws.com timed out. (connect timeout=60)'))
host = si3-splunk1


index = 0014000000kbznqaa1


source =
/opt/splunkcoreengine/ce_customers/0014000000KBzNQAA1/1425528/si3-spl
unk1-sh_ds_


ls_-20190708-190819/log/splunk_ta_aws_aws_cloudwatch.log sourcetype =
splunk_ta_


aws_aws_cloudwatch
```

To fix this issue in your deployment:

1. Your VPC endpoint interface needs to be set up in your AWS environment. See the AWS documentation for details regarding VPC endpoints.

2. Update the Splunk instance that is being used for data collection to use your VPC endpoint as a gateway to allow connections to be established to your AWS services:
    a. In your Splunk instance, navigate to `./etc/apps/Splunk_TA_aws/bin/3rdparty/botocore/data/endpoints.json`, and open using a text editor.
    b. Update the `hostname` to use the hostname of your VPC endpoint interface. For example:

Before

```
"sts": {

"defaults": {

"credentialScope": {

"region": "us-east-1"

},

"hostname": "sts.amazonaws.com"
```

After

```
},

"sts" : {

"defaults" : {

"credentialScope" : {

"region" : "us-east-1"

},

"hostname" : "<Enter VPC endpoint Interface DNS name here>"
```

c. Save your changes.
d. In your Splunk instance, navigate to `./etc/apps/Splunk_TA_aws/bin/3rdparty/botocore/data/endpoints.json`, and open using a text editor.
e. Update the `hostname` to use the hostname of your VPC endpoint interface. For example:

Before

```
"sts": {

"defaults": {

"credentialScope": {

"region": "us-east-1"

},

"hostname": "sts.amazonaws.com"
```

After

```
},

"sts" : {

"defaults" : {

"credentialScope" : {

"region" : "us-east-1"

},

"hostname" : "<Enter VPC endpoint Interface DNS name here>"
```

f. Save your changes.

3. Restart your Splunk instance.

4. Validate that the connection to your VPC has been established.

Failed to load input and configuration page when running the Splunk software on a custom management port

If the Splunk software fails to load input and configuration page while running on the custom management port (for example, `<IP>:<CUSTOM_PORT>`), perform the following troubleshooting steps.

1. Navigate to `$SPLUNK_HOME/etc/`
2. Open `splunk-launch.conf` using a text editor.
3. Add the environment variable `SPLUNK_MGMT_HOST_PORT=<IP>:<CUSTOM_PORT>`
4. Save your changes.
5. Restart your Splunk instance.

API reference for the Splunk Add-on for Amazon Security Lake

See the following sections for API reference information for the Splunk Add-on for Amazon Security Lake.

Account

`https://<host>:<mPort>splunk_ta_aws_aws_account`

API for AWS Account settings.

GET, POST, or DELETE

Collapse

API for AWS Account settings

Request parameters

name	Boolean `true`	Name
key_id	Boolean `true`	Key ID
secret_key	Boolean `true`	Secret Key
category	Boolean `true`	Region Category
iam	Boolean `false`	Identifies EC2 Instance Role

Amazon Security Lake input

`https://<host>:<mPort>ASDL_Splunk_TA_aws_aws_sqs_based_s3`

API for the Amazon Security Lake input.

GET, POST, or DELETE

Collapse

API for the Amazon Security Lake input

Request parameters

name	Boolean `true`	Name
aws_account	Boolean `true`	AWS Account
aws_iam_role	Boolean `false`	Assume role
sqs_queue_region	Boolean `true`	Name of the AWS SQS region
sqs_queue_url	Boolean `true`	URL of the AWS SQS queue
sqs_batch_size	Boolean `true`	Maximum number of messages
s3_file_decoder	Boolean `true`	Name of an S3 file decoder
interval	Boolean `true`	Interval
sourcetype	Boolean `true`	Sourcetype API for `aws:description`
index	Boolean `true`	Index

Searching Splunk Add-on for Amazon Security Lake Events

Once AWS log events are being pulled from the data lake and indexed into your Splunk environment you are ready to start searching them using the Open Cyber Security Framework (OCSF) schema.

- Schema Explorer - https://schema.ocsf.io/
- Open Cyber Security Framework (OCSF) Whitepaper - https://github.com/ocsf/ocsf-docs/blob/main/Understanding%20OCSF.pdf
- Open Cyber Security Framework (OCSF) GitHub Project - https://github.com/ocsf

Example Threat Detection Searches

Review AssumeRoleWithSAML

AssumeRoleWithSAML is an event that needs to be monitored on its own. It is associated with users escalating privileges or accessing cross organization unit, cross enterprise, federation resources. AssumeRoleWithSAML is the target function in hybrid environments, especially Microsoft ADFS where an attacker with high privileges can forge SAML assertions and use AssumeRoleWithSAML to cross access cloud resources.

```
index="asdl-events" | spath "api.operation" | search
"api.operation"=AssumeRoleWithSAML | spath "cloud.region" | search
"cloud.region"="*" | spath "http_request.user_agent" | search
"http_request.user_agent"="*" | spath "identity.idp.name" | search
"identity.idp.name"="*" | spath "identity.user.type" | search
"identity.user.type"=* | spath "resources{}.uid" | search
"resources{}.uid"="*" | spath "src_endpoint.ip" | search
"src_endpoint.ip"="*"| spath "identity.idp.name" | search
"identity.idp.name"="*" |table src_endpoint.ip http_request.user_agent
cloud.region identity.idp.name identity.user.type identity.idp.name
resources{}.uid
```

API access denied operation by resource and user type

An AccessDenied error may indicate several things, including misconfigurations, expired credentials, scanning attempts, or exploitation attempts. This search provides information about failed access attempts against S3 buckets which are constantly being scanned or profiled by opportunistic or malicious actors on the internet. This also applies to internal environments where sensitive data is stored and classified under restricted access.

```
index=* | spath "api.response.error" | search
"api.response.error"=AccessDenied "resources{}.type"="AWS::S3::Bucket" |
stats count by api.operation api.response.message resources{}.uid
identity.user.type
```

Find users performing ListBuckets api operation

The API operation "ListBuckets" is an uncommon operation for scripts or set procedures to execute. This operation indicates the intention to obtain information about storage buckets, their names and owners. This is not necessarily malicious but is a type of activity associated with the discovery stage of an attack wherein an attacker gathers information to determine if your AWS environment may be susceptible to a broader attack. This may indicate suspicious activity and it is usually associated with post exploitation activities.

```
index=* "api.operation"=ListBuckets |stats count by http_request.user_agent
identity.user.credential_uid identity.session.mfa identity.user.type
src_endpoint.uid identity.user.uuid
```

Find PutObject command api operation by endpoint, region and user

Users must have WRITE permission in order to place objects in a bucket. This search may be noisy but can aid hunting for misconfigured bucket permissions, misuse or unauthorized use of storage resources within AWS infrastructure.

```
index=* "api.operation"=PutObject | stats count by cloud.region
identity.session.issuer src_endpoint.uid http_request.user_agent
identity.user.credential_uid resources{}.type resources{}.uid
```

AssumeRole by targeted resource

The AssumeRole API operation is extensively used in AWS environments as a mechanism to avoid using fully privileged accounts in order to perform specific tasks that require elevation. The AssumeRole operation is not unusual or malicious in and of itself, it is used to access AWS resources that users might not normally have access to. However, it must be monitored to

assure the validity of users or processes using it. This search, by itself, does not prove malicious use but it does provide indicators for hunting malicious use of the AssumeRole operation. Hunting should especially consider the context of Trust Policy permissions, regions, type of access keys, user agent (exploitation tools or Desktop CLI display different UA than cloud requests) and cross domain access (especially source IP if organization has IP access policy).

```
index=* "api.operation"=AssumeRole | stats count by cloud.region
http_request.user_agent identity.invoked_by identity.user.uid
identity.user.type src_endpoint.domain resources{}.uid src_endpoint.ip
```

Monitor ListSecrets by users and IPs

The ListSecrets operation is sensitive by nature and should be monitored. Misuse or misconfiguration of this privilege may result in unauthorized access, misuse of resources or compromise of organization resources. Source endpoint IP, user identity UID, user agent and frequency should be considered when monitoring use of this operation.

```
index=* "api.operation"=ListSecrets | stats count by
http_request.user_agent identity.user.credential_uid identity.user.uid
src_endpoint.ip
```

Monitor ListSecrets by Non Root accounts

This search focuses on ListsSecrets operations performed by NON ROOT accounts. It focuses on temporary keys that may belong to users or services.

```
index=* "api.operation"=ListSecrets identity.user.uid!=AKIA*| stats count
by http_request.user_agent identity.user.credential_uid identity.user.uid
src_endpoint.ip
```

Monitor ListSecrets from Root Account

This search identifies ListSecrets operations by users with keys beginning with AKIA* which are usually assigned to long term users or ROOT accounts. Root account use is discouraged unless strictly necessary and its use in abnormal manner may indicate compromise.

```
index=* "api.operation"=ListSecrets identity.user.uid=AKIA* | stats count by http_request.user_agent identity.user.credential_uid identity.user.uid src_endpoint.ip
```

Release notes for the Splunk Add-on for Amazon Security Lake

This version of the Splunk Add-on for Amazon Security Lake was released on November 29th, 2022.

Compatibility

This version of the Splunk Add-on for Amazon Security Lake is compatible with the following software, CIM versions, and platforms:

Splunk platform versions	8.0 and later
CIM	N/A
Compatible OS for data collection	Linux 64-bit
Vendor products	Amazon Security Lake

New features

This is the first release of this add-on.

Fixed issues

This version of this add-on contains no fixed issues.

Known issues

This version of this add-on contains no fixed issues.

Third-party software attributions

This add-on incorporates the following third-party libraries.

Third-party software attributions for the Splunk Add-on for Amazon Security Lake

Quickstart guide for the Splunk Add-on for Amazon Security Lake

The following quickstart guide outlines how to install and configure this add-on to ingest data from your Amazon Web Services (AWS) platform deployment. For more information on any of the following steps, see the applicable reference topic in this manual.

1. Provision your Amazon Security Lake data pipeline.
    a. Gather required fields
        i. IAM role to assume
        ii. SQS queue name to ingest
        iii. AWS region
        iv. IAM secret key and Key ID
2. Download and install the Python for Scientific Computing on every Splunk platform instance where the ASDL - Splunk Add-on for AWS will be installed.
3. Download and install the ASDL - Splunk Add-on for AWS

    Note: The add-on package size is greater than 115MB. File upload and installation might take a few minutes

4. Restart your Splunk platform instance.
5. Configure the ASDL - Splunk Add-on for AWS
    a. Add AWS account information
        i. Navigate to the **ASDL - Splunk Add-on for AWS**, and click on **Configuration > Account > Add**
        ii. Add Key ID

        iii. Add IAM secret key
        iv. Save your changes
    b. Add IAM Role
        i. Navigate to the **ASDL - Splunk Add-on for AWS**, and click on **Configuration > IAM Role > Add**
        ii. Add name of IAM role
        iii. Add role ARN
        iv. Save your changes
6. Add input
    a. Navigate to the **ASDL - Splunk Add-on for AWS**, and click on **Inputs > Create New Input > ASDL > SQS-Base S3**
    b. Add input name
    c. Select AWS Account
    d. Select Assume Role
    e. Select AWS Region
    f. Enter SQS queue name (name, not full ARN)
        i. For example: arn:aws:sqs:us-west-2:934522865364:MYQueue is MYQueue
    g. Select index
        i. Select the destination for your data
    h. Save your changes