Data Forwarder & Splunk Configuration

VMware Empowering the Modern SOC

Table of contents

Data Forwarder & Splunk Configuration

Introduction

This article is supplementary material for the video “Data Forwarder & Splunk Configurations”, an end-to-end demo of getting alerts, watchlist hits, and endpoint events from VMware Carbon Black Cloud to Splunk via AWS S3 & SQS.

You’ll find an outline of each step, as well as artifacts such as sample AWS policies and links to references.

Some organizations may have two or more teams across Carbon Black Cloud, AWS, and Splunk that will be involved in the configuration; this article was designed to help each team identify what needs to be handed off to ensure success.

This is one of many possible ways to configure AWS; defer to your organization’s AWS or security team’s best practices.

Video

Benefits of the Data Forwarder

Video Timestamp: [00:33]

The Data Forwarder was built for low-latency data streaming, reliably, at scale. If your organization has high-volume alerts, or you're looking to bring the visibility that Watchlist Hits and Endpoint Events provide into Splunk, the Data Forwarder is your solution.

Our Carbon Black Cloud Splunk App offers native inputs for data sets Alerts, Audit Logs, Live Query Results, and Vulnerabilities. You configure the app with a Carbon Black Cloud API key, and it does the rest. The native input works well for lower-volume data sets; but if you're an enterprise SOC where scale and reliability is critical, the data forwarder is our recommended solution.

Data Types

Carbon Black Cloud currently offers three data types in the Data Forwarder. Each type should get its own forwarder, its own prefix (directory) in the S3 bucket, its own SQS queue, its own Splunk input, and its own Splunk Source Type. Here are examples for each:

Data Forwarder name	Data Forwarder Type	S3 Prefix	SQS queue name	Splunk AWS SQS-Based S3 Input Name	Splunk Source Type
Splunk Forwarder - Alerts	Alert	alerts/	cbc-demo-queue-alerts	CBC-Demo-Alerts	vmware:cbc:s3:alerts
Splunk Forwarder - Events	Endpoint event	events/	cbc-demo-queue-alerts	CBC-Demo-Events	vmware:cbc:s3:events
Splunk Forwarder – Watchlist Hits	Watchlist hit	watchlist-hits/	cbc-demo-queue-watchlist-hits	CBC-Demo-Watchlist-Hits	vmware:cbc:s3:watchlist:hits

AWS Configuration

S3 Bucket

Video Timestamp: [05:04]

The S3 bucket must be created in the correct region based on your Carbon Black Cloud Org URL, as documented in the VMware Documentation: Create an S3 Bucket in the AWS Console.

The bucket policy must allow Carbon Black Cloud's principal write-only access; the list of region-specific AWS principals and required permissions can be in the VMware Documentation: Configure the Bucket Policy to Allow Access. A sample policy is available in the Appendix: Sample Bucket Policy.

In the demo video, the bucket name was `cbc-demo-bucket`

Handoff: The S3 Bucket Name will be handed off to your Carbon Black Cloud team.

KMS Encryption: The Carbon Black Cloud Data Forwarder now supports KMS Encryption (Symmetric keys only). This requires granting additional permissions to allow Carbon Black Cloud's principal to access the key. See the Appendix: Sample Policy for KMS Encryption for additional details and examples.

SQS Queues

Video Timestamp: [06:38]

Deadletter

Most SQS consumers require a deadletter queue, essentially a place the consumer can dump bad or malformed messages from the primary queues if something goes wrong to avoid data loss or reprocessing bad data.

In the demo video, this queue was named `cbc-demo-queue-deadletter`.

Primary Queues

Create one queue per data type. In the demo video, these queues were:

- `cbc-demo-queue-alerts`
- `cbc-demo-queue-events`
- `cbc-demo-queue-watchlist-hits`

In each queue:

- Attach a queue policy that enables the S3 bucket `sqs:sendmessage` permissions. A sample policy can be found in the Appendix: Sample Queue Policy
- Specify the deadletter queue created in the step above

Handoff: Copy the ARN of each primary queue; these will be handed off to your SIEM team.

S3 Bucket Notification

Video Timestamp: [08:15]

The S3 bucket event notification is what pushes a message onto the queues when Carbon Black Cloud writes a new object into the bucket. There should be one notification per queue/data type.

In the demo video, the notifications were:

- CBC Demo Notification Alerts
  - Prefix: `alerts/`
  - Suffix: `.jsonl.gz`
  - Event Types: All Object Create Events
  - Destination Queue: `cbc-demo-queue-alerts`
- CBC Demo Notification Events
  - Prefix: `events/`
  - Suffix: `.jsonl.gz`

  - Event Types: All Object Create Events
  - Destination Queue: `cbc-demo-queue-events`

- CBC Demo Notification Watchlist Hits
  - Prefix: `watchlist-hits/`
  - Suffix: `.jsonl.gz`
  - Event Types: All Object Create Events
  - Destination Queue: `cbc-demo-queue-watchlist-hits`

Handoff: Note the prefix you defined for each notification; these will be handed off to your Carbon Black Cloud team.

AWS Access

Most Splunk instances will require AWS access keys. Splunk will use those keys to assume a role which has the permissions necessary; see the Policy & Permissions Diagram in the appendix for additional details.

This follows the guidance provided by Splunk in the AWS Add-on documentation, Create and configure roles to delegate permissions to IAM users.

AWS User

Video Timestamp: [09:39]

In the demo video the user `cbc-demo-user` was created with programmatic access only and with no permissions.

Copy the User ARN; you will need this when configuring the AWS Role's trust relationships below.

Handoff: Copy the Access Key ID and Secret Key; these will be handed off to the SIEM team.

AWS Role’s Policy

Video Timestamp: [10:31]

This policy defines what access Splunk requires for the SQS-based S3 input. These permissions are documented by Splunk in the AWS Add-on documentation, Configure AWS permissions for the SQS-based S3 input.

Required permissions for SQS:

- `GetQueueUrl`
- `ReceiveMessage`
- `SendMessage`
- `DeleteMessage`
- `GetQueueAttributes`
- `ListQueues`

Required permissions for S3 buckets and objects:

- `GetObject`

Required permissions for KMS (if you are using KMS Encryption on your S3 bucket)

- `Decrypt`

In the demo video, the policy name is `cbc-demo-policy`. That sample policy is available in the Appendix: Sample Role Policy (or Appendix: Sample Policy for KMS Encryption)

AWS Role

Video Timestamp: [11:19]

The AWS role’s “trusted entity” should be “another AWS Account”; however the account ID should be your own, which can be found in the upper-right of the AWS Console. Attach the role’s policy that was created in the previous step.

In the demo video, the role name is `cbc-demo-role`.

Once the role is created, open the role in the AWS console, go to the Trust relationships tab and click "edit trust relationship". Then replace the Principal -> AWS field with ARN of the user created above. A sample policy can be found in the Appendix: Sample Role Trusted Entity.

Handoff: Copy the role ARN; this will be handed off to the SIEM team.

Handoff

If another team in your organization is handling the Carbon Black Cloud or Splunk configuration, here’s what they’ll need.

The team with Carbon Black Cloud Access who will create the Data Forwarder will need:

- The S3 bucket name
- The S3 prefixes you defined for each data type in the event notifications

The SIEM team will need:

- The AWS Access Key ID and Secret Key associated with the AWS user
- The AWS Role ARN
- The AWS Region
- The ARNs of the queues you created and which data types they correspond to

Here’s the sample hand-off from the demo video:

Handoff to Carbon Black Cloud Team	
Artifact	**Sample Value**
	`cbc-demo-bucket`
	Alerts: `alerts/` Events: `event/` Watchlist Hits: `watchlist-hits/`

Handoff to Splunk Team	
Artifact	**Sample Value**
	Access Key: `AKIAXZN…` Secret Key: `6VSWaYp…`
	`arn:aws:iam::535601802221:role/cbc-demo-role`
	Alerts: `arn:aws:sqs:us-east-1:535601802221:cbc-demo-queue-alerts` Events: `arn:aws:sqs:us-east-1:535601802221:cbc-demo-queue-events` Watchlist Hits: `arn:aws:sqs:us-east-1:535601802221:cbc-demo-queue-watchlist-hits`

Carbon Black Cloud Configuration

Pre-requisites

Video Timestamp: [12:45]

- AWS team has created an S3 bucket in the specified region with the correct access policy
- If you're creating data forwarders from the Carbon Black Cloud console:
  - Carbon Black Cloud user in a role with the "View/Manage Data Forwarders" permissions
- If you're creating data forwarders from the Carbon Black Cloud Data Forwarder API
  - Carbon Black Cloud API key with `event-forwarder.settings CREATE, UPDATE`
- What data are you forwarding?
  - This should be determined in collaboration with your SIEM team based on their data budget and use cases.
  - See Also: Getting Started: Custom Filters for the Data Forwarder

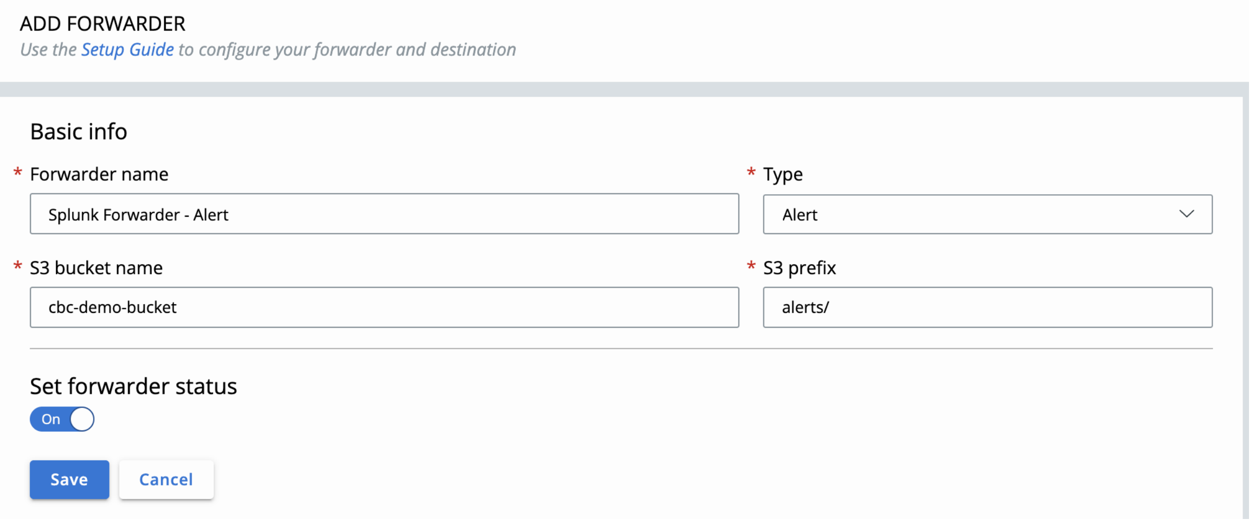
Data Forwarders

Video Timestamp: [13:44]

From Settings -> Data Forwarders, create one data forwarder per data type. It's helpful to include the destination (e.g. "Splunk") in the forwarder name to help future-you identify what that forwarder used for.

Be sure to specify the s3 prefix for each data type as specified by your AWS team.

More information can be found in Carbon Black Cloud documentation under Add a Data Forwarder.



ADD FORWARDER

Use the Setup Guide to configure your forwarder and destination

Basic info

* Forwarder name: Splunk Forwarder - Alert	* Type: Alert
* S3 bucket name: cbc-demo-bucket	* S3 prefix: alerts/

Set forwarder status: On

Troubleshooting

Video Timestamp: [14:26]

Common errors include:

- Forwarder is not enabled (Status = "Off")
- "Bucket does not exist"
  - Check the bucket name
  - Check your AWS team has created the bucket
- "Provide a valid bucket with appropriate permissions"
  - Check that the bucket was created in the correct region
  - Check the bucket policy allows the specified permissions to Carbon Black Cloud's principal
  - Check the bucket is not using unsupported encryption types - Carbon Black Cloud Data Forwarder currently

supports:

  - Amazon S3 key (aka “SSE-S3”) encryption
  - AWS Key Management Service key (aka “SSE-KMS”) encryption with Symmetric keys
    - Asymmetric KMS keys are not currently supported

- If the bucket is using KMS encryption, ensure the required permissions have been granted to Carbon Black Cloud's principal to access the KMS key
  - See Appendix: Sample Policy for KMS Encryption for examples

Splunk Configuration

Apps & Add-ons

Video Timestamp: [15:05]

- The Splunk Add-on for AWS (#1876)
  - Typically on the heavy forwarder (Splunk on-prem) or IDM (Splunk Cloud)
- Carbon Black Cloud App / IA / TA
  - See the Splunk Deployment Guide on Developer Network for details
  - Watch a demo video of configuring the app

As the Carbon Black Cloud App/IA/TA defines source types and event types that drive dashboards, CIM, and other behavior, configuring these before onboarding Data Forwarder data is strongly recommended. This includes creating an index for the Carbon Black Cloud data and specifying the index in the “Base Configuration” tab of the app administration.

Configure the AWS Account & Role

Video Timestamp: [15:51]

From the Configuration section -> Account tab, add the account with the Key ID and Secret Key provided by your AWS team.

From the IAM Role tab, add the Role ARN provided by your AWS team.

Configure the AWS Input

Video Timestamp: [16:11]

From the Inputs section, add a new input of type “Custom Data Type” -> “SQS-based S3”.

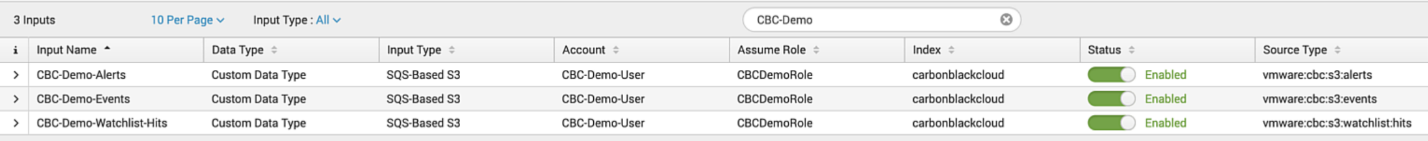
- Select the AWS Account and Role that were configured in the previous step
- Populate the region provided by your AWS team
- Populate the SQS queue name associated with the Alerts data type
- Change the Source Type to `vmware:cbc:s3:alerts` (see table below)
- Change the Index to your primary Carbon Black Cloud data index
- **IMPORTANT:** Ensure "Signature Validate All Events" is unchecked; otherwise you will encounter errors on data ingest
  - This option was added in Splunk Add-on for AWS v5.2+

Repeat that process for Event & Watchlist Hit data, using the correct queue and source type for each input.

This table maps the Data Forwarder type to the required Splunk Source Type. There are likely intermediary mappings between the Data Forwarder Type -> S3 Prefix -> SQS Queue that should be provided by your AWS team.

Caution: if the AWS and Data Forwarder were set up more than a few days before the Splunk input, Splunk will need to process that backlog of data. If you're concerned about the processing or license implications of that, your AWS team can purge the SQS queues before you onboard data to clear the backlog.

Data Forwarder Type	Splunk Source Type
Alert	`vmware:cbc:s3:alerts`
Endpoint Event	`vmware:cbc:s3:events`
Watchlist Hit	`vmware:cbc:s3:watchlist:hits`



3 Inputs 10 Per Page Input Type : All CBC-Demo

i	Input Name	Data Type	Input Type	Account	Assume Role	Index	Status	Source Type
>	CBC-Demo-Alerts	Custom Data Type	SQS-Based S3	CBC-Demo-User	CBCDemoRole	carbonblackcloud	Enabled	vmware:cbc:s3:alerts
>	CBC-Demo-Events	Custom Data Type	SQS-Based S3	CBC-Demo-User	CBCDemoRole	carbonblackcloud	Enabled	vmware:cbc:s3:events
>	CBC-Demo-Watchlist-Hits	Custom Data Type	SQS-Based S3	CBC-Demo-User	CBCDemoRole	carbonblackcloud	Enabled	vmware:cbc:s3:watchlist:hits

Health Check

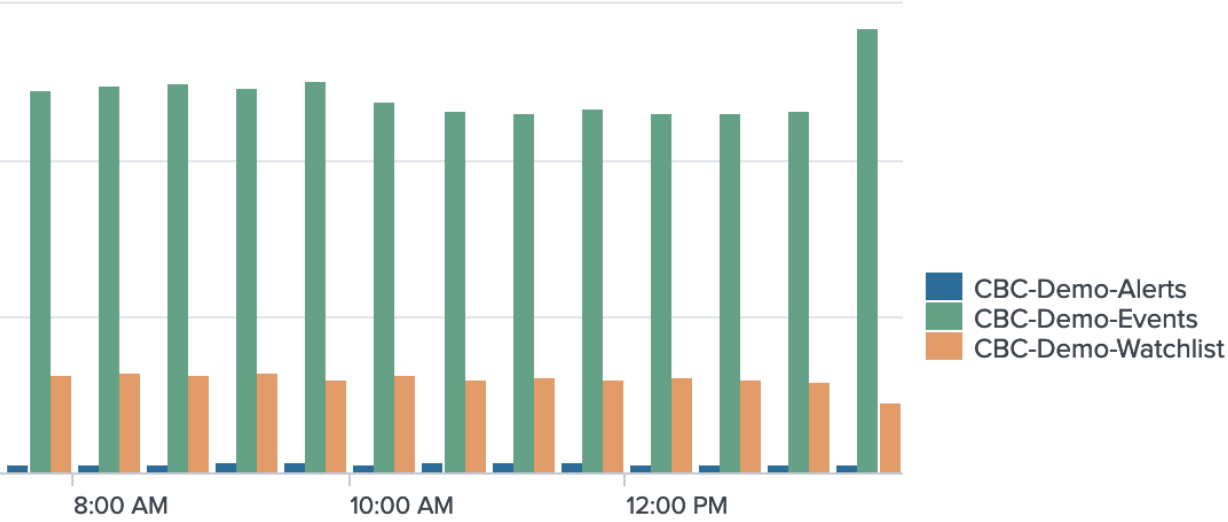
Video Timestamp: [17:17]

To confirm the configuration is working end-to-end, use the “Health Check” -> “Health Overview” dashboard in the Splunk Add-on for AWS.

You can also run the following query; consider changing the y-axis to a log axis via Format -> Y-Axis -> Scale -> Log. This helps show both high-volume events and low-volume alerts on the same chart.

Splunk Query: Data Volume by AWS Input

```
index="_internal" (host="*") (sourcetype=aws:s3:log OR sourcetype=aws:logs:log OR sourcetype=aws:sqsbaseds3:log
OR sourcetype=aws:description:log OR sourcetype=aws:cloudwatch:log) (datainput="*") (action=index OR
message="Sent data for indexing.")
| eval size = size / 1024 / 1024
| stats count by _time, host, sourcetype, datainput, size
| eval size=size*count
| eval size=round(size,2)
| timechart sum(size) by datainput
| fillnull value=0
```



Troubleshooting

Data Missing in Splunk

Video Timestamp: [23:15]

If there's no data flowing into Splunk, or some of the data types (Data Forwarder types) are missing but some data is flowing in, try these tips.

First check the "Splunk Add-on for AWS" for errors - go to Health Check > S3 Inputs Health Details.

If that doesn't make it clear, try these queries.

In the following queries, ensure you replace `carbonblackcloud` with your Splunk index name.

If using a KMS encrypted S3 bucket, ensure your AWS Role's Policy grants decrypt permission to the KMS key (see Appendix: Sample Policy for KMS Encryption). A "Failed to download file" error from the second query below suggests the KMS permissions are incorrect.

A common error (from "Detailed error messages..." query below) is: `Warning: This message does not have a valid SNS Signature None None doesn't match required format '^https://sns\\.[-a-z0-9]+\\.amazonaws\\.com(?:\\.cn)?/'`

- The Splunk Add-on for AWS v5.2+ added an option to the input to "Signature Validate All Events"
- This option must be **unchecked** in your SQS-Based S3 inputs.

Identify which data types are coming into Splunk (video timestamp [24:00])

```
index="carbonblackcloud" sourcetype="vmware:cbc:s3:*"
| timechart count by sourcetype
```

Detailed error messages from the SQS + S3 inputs (click to expand, video timestamp [24:30])

```
index="_internal" sourcetype="aws:sqsbaseds3:log" (level = ERROR OR level = WARNING)
| fillnull value="" ErrorCode, ErrorDetail
| eval ErrorDetail = if((ErrorDetail == "" or ErrorDetail == "''") and !isnull(message), message, ErrorDetail)
| lookup aws_health_error_type_lookup ErrorCode, ErrorDetail, sourcetype OUTPUT ErrorType
| lookup aws_log_sourcetype_modinput_lookup sourcetype OUTPUT modinput
| eval start_time=strftime(start_time, "%m/%d/%y %H:%M:%S")
| table host, modinput, datainput, start_time, ErrorType, ErrorDetail, uri
| sort host, modinput, datainput, start_time, ErrorCode
| rename host as Host, modinput as "Input Type", datainput as "Data Input", ErrorType as "Error Type"
start_time as "Start Time", ErrorDetail as "Error Detail"
```

Data not in Splunk Dashboard

Video Timestamp: [23:55]

Is there data flowing into Splunk, but you don't see it in the Splunk dashboard?

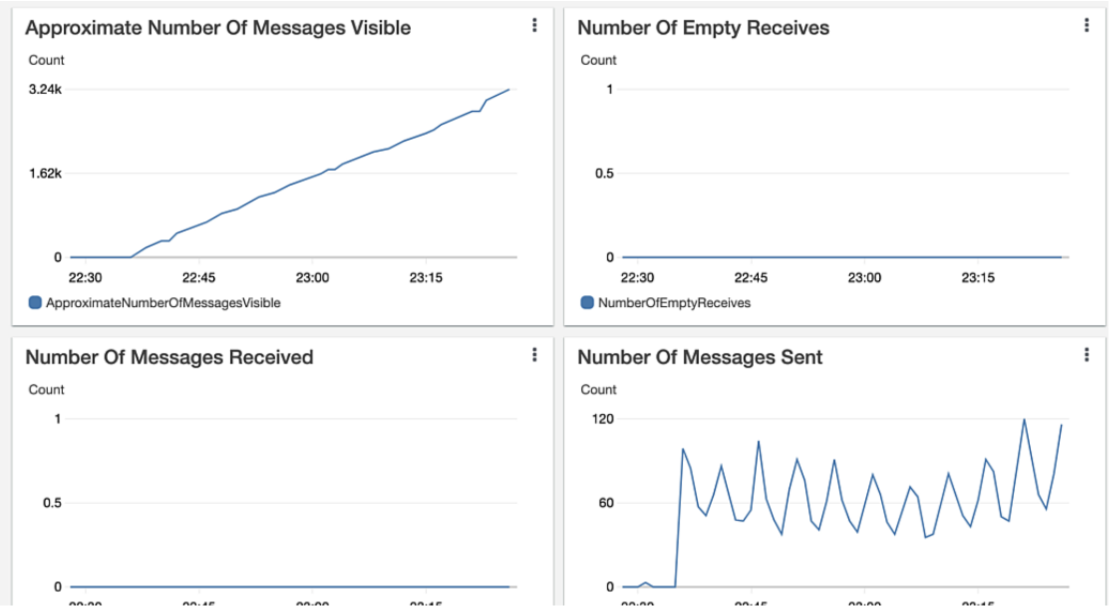
Verify the index name is set as you expect:

- In the "VMware Carbon Black Cloud" Splunk app, go to Administration > Application Configuration
- Under "VMware CBC Base Configuration", set the name of the Base Index (aka VMware CBC Base Index) from `index=default` to your chosen index name

SQS and S3 Issues

Video Timestamp: [24:50]

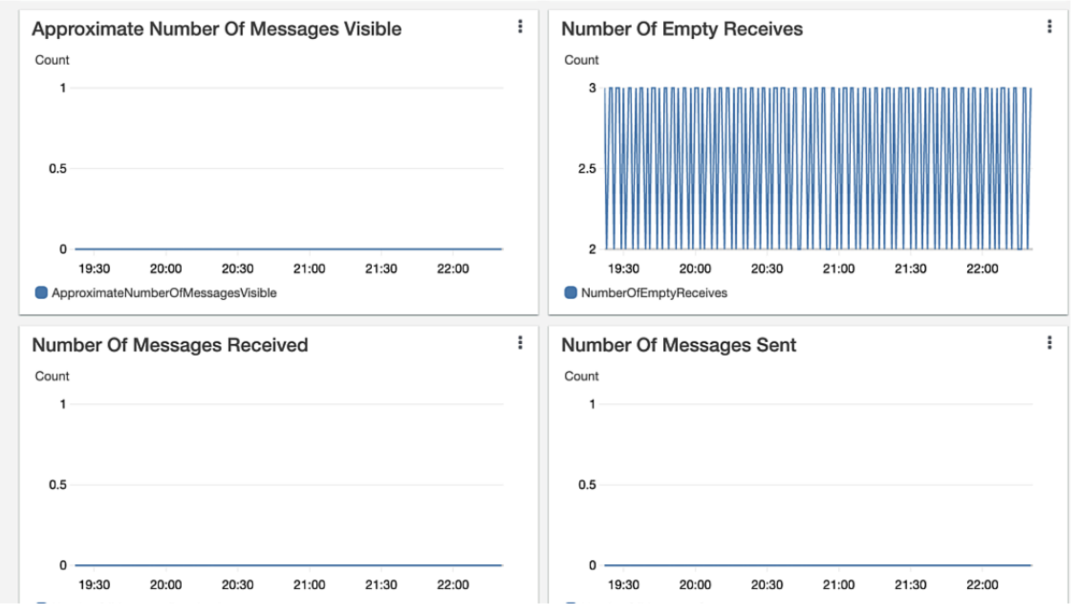
Issue 1: SQS Queue Messages increasing, but no Messages Received



This likely means the Data Forwarder is writing to S3 and S3 is writing new object notifications to SQS. But since Messages Received is zero, Splunk isn't successfully reading from the queue. Check the "Splunk Add-on for AWS" Inputs:

1. Are enabled
2. Are reading from the correct queues
3. For correct Principal and Role
4. For correct permissions for Principal/Role to read from SQS

Issue 2: No message sent to Queue from S3



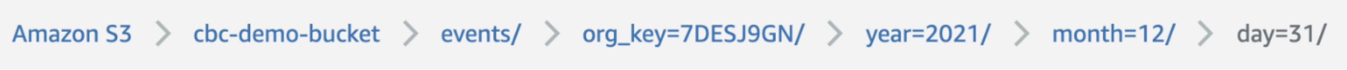
When number of Messages Visible and number of Messages Sent are both zero, no messages are getting to the queue. When Number of Empty Receives is non-zero and/or consistently populated, this means Splunk or the Add-on are checking the queues, but no messages available.

First, check your S3 bucket configuration (Properties):

1. Are the Event notifications created for all data types (aka types of Data Forwarders)?
2. Are the Event notifications configured to send to correct queues?
3. Are the Event notifications filtering for the expected prefixes and suffixes?

Then check your S3 bucket contents:

1. Are you seeing all expected prefixes as configured in CBC Data Forwarders?
2. In each prefix, do you see an org_key and data folder structure, with .jsonl.gz files at the bottom? Are files with recent date/time values available?

   Amazon S3 > cbc-demo-bucket > events/ > org_key=7DESJ9GN/ > year=2021/ > month=12/ > day=31/

3. Do you see healthcheck/healthcheck.json files under each Data Forwarder prefix?

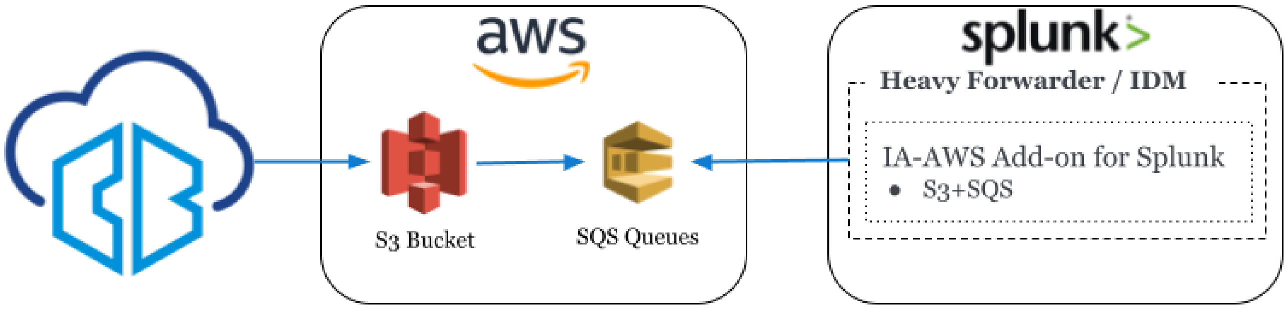
If you're not seeing this data in your bucket or the most recent data is out of date, check the bucket permissions still allow Carbon Black Cloud's principal write access and the Data Forwarder is still setup and enabled.

You can also test the Data Forwarder's connection from Carbon Black Cloud.

Appendix

Simple Data Flow

Video Timestamp: [01:20]



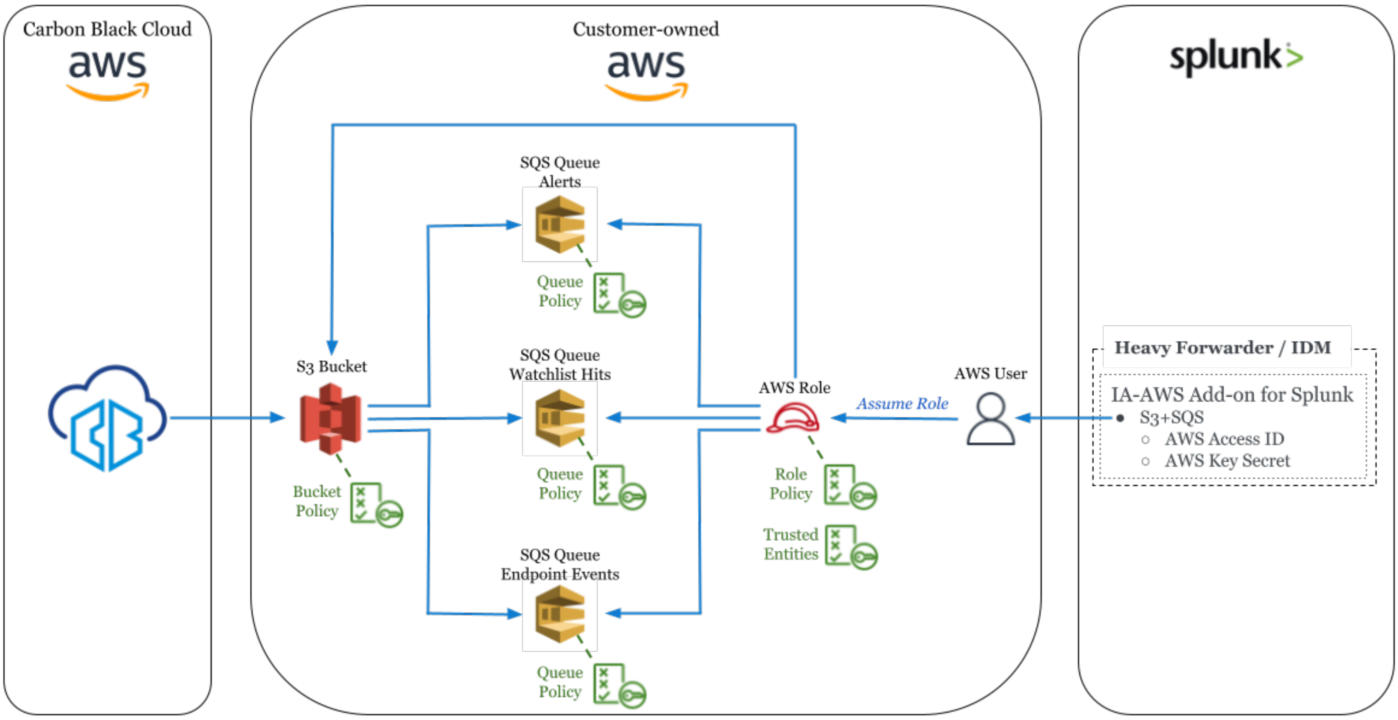
Reference Architecture

Video Timestamp: [02:10]

Policy & Permissions Diagram

Video Timestamp: [03:46]

The policy diagram describes the permissions and trust relationships between each artifact in the reference architecture.



Sample Bucket Policy

Video Timestamp: [06:13]

Replace the values:

- Resource: your bucket name.

- Principal -> AWS: This policy allows access from each region's Carbon Black Cloud AWS principals; remove all but the principal that corresponds to your region.

Sample Bucket Policy

```
{
    "Version": "2012-10-17",
    "Id": "cbc-demo-bucket-2021-12-01",
    "Statement": [
        {
            "Sid": "0480ecd1228e41ec98342970a90bec4c",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::132308400445:role/mcs-psc-prod-event-forwarder-us-east-1-event-forwarder",
                    "arn:aws:iam::132308400445:role/mcs-psc-prod-event-forwarder-eu-central-1-event-forwarder",
                    "arn:aws:iam::132308400445:role/mcs-psc-prod-event-forwarder-ap-northeast-1-event-
forwarder",
                    "arn:aws:iam::132308400445:role/mcs-psc-prod-event-forwarder-ap-southeast-2-event-
forwarder"
                ]
            },
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:GetObjectAcl",
                "s3:ListMultipartUploadParts",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::cbc-demo-bucket/*"
        }
    ]
}
```

Sample Queue Policy

Video Timestamp: [07:39]

Replace the values:

- Resource field: your queue ARN
- Condition -> ArnLike -> aws:SourceArn: Your bucket name

Sample Queue Policy

```
{
  "Version": "2008-10-17",
  "Id": "cbc-demo-queue-alerts-2021-12-01",
  "Statement": [
    {
      "Sid": "54856085abd9429385fbeeb73bfcba69",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "SQS:SendMessage",
      "Resource": "arn:aws:sqs:us-east-1:535601802221:cbc-demo-queue-alerts",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:::cbc-demo-bucket"
        }
      }
    }
   ]
}
```

Sample Role Policy

Video Timestamp: [10:39]

Replace the values:

- Resource:
  - Your queue ARNs
  - Your bucket name

Sample Role Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "0480ecd1228e41ec98342970a90bec4c",
            "Effect": "Allow",
            "Action": [
                "sqs:DeleteMessage",
                "s3:GetObject",
                "sqs:GetQueueUrl",
                "sqs:ReceiveMessage",
                "sqs:SendMessage",
                "sqs:GetQueueAttributes"
            ],
            "Resource": [
                "arn:aws:sqs:us-east-1:535601802221:cbc-demo-queue-alerts",
                "arn:aws:sqs:us-east-1:535601802221:cbc-demo-queue-events",
                "arn:aws:sqs:us-east-1:535601802221:cbc-demo-queue-watchlist-hits",
                "arn:aws:s3:::cbc-demo-bucket/*"
            ]
        },
        {
            "Sid": "17102cece21c40bb8f5c16f8a9ffb8bc",
            "Effect": "Allow",
            "Action": "sqs:ListQueues",
            "Resource": "*"
        }
    ]
}
```

Sample Role Trusted Entities

Video Timestamp: [11:50]

Replace the values:

- Principal -> AWS: Your user’s ARN

Sample Role Trusted Entities

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::535601802221:user/cbc-demo-user"
      },
      "Action": "sts:AssumeRole",
      "Condition": {}
    }
  ]
}
```

Sample Policies for KMS Encryption

Video Timestamp: [20:50]

Carbon Black Cloud's Principal will need the following permissions on your KMS key:

- `kms:GenerateDataKey`
- `kms:Decrypt`

If you're wondering why "Decrypt" is required, it's an AWS requirement for successful Multipart Uploads. If you do not grant Decrypt permissions, Carbon Black Cloud can successfully write smaller objects to your encrypted bucket. However, larger objects will silently fail, resulting in data loss.

When using KMS Encryption, the following policies will need to be added modified:

- Add a policy attached to the KMS key, which enables Carbon Black Cloud to access the key for standard and multi-part uploads
  - Resource field: Replace with your KMS key's ARN.
  - Principal -> AWS fields: This policy allows access from each region’s Carbon Black Cloud AWS principals; remove all but the principal that corresponds to your region.
- Modify the Role Policy to enable the SIEM to decrypt objects in the bucket using the KMS key
  - Resource field: Replace the following values
    - Your queue ARNs
    - Your bucket name
    - Your KMS Key ARN

Sample KMS Policy

```
{
    "Version": "2012-10-17",
    "Id": "cbc-demo-key-2022-03-18",
    "Statement": [
        {
            "Sid": "Allow access for Carbon Black Cloud",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::132308400445:role/mcs-psc-prod-event-forwarder-eu-central-1-event-forwarder",
                    "arn:aws:iam::132308400445:role/mcs-psc-prod-event-forwarder-ap-southeast-2-event-
forwarder",
                    "arn:aws:iam::132308400445:role/mcs-psc-prod-event-forwarder-ap-northeast-1-event-
forwarder",
                    "arn:aws:iam::132308400445:role/mcs-psc-prod-event-forwarder-us-east-1-event-forwarder"
                ]
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:535601802221:key/fb1aaaaa-8c3a-428a-b0ba-a0a136e9aaaa"
        }
    ]
}
```

Sample Role Policy, including KMS (click to expand)

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "0480ecd1228e41ec98342970a90bec4c",
            "Effect": "Allow",
            "Action": [
                "sqs:DeleteMessage",
                "s3:GetObject",
                "sqs:GetQueueUrl",
                "sqs:ReceiveMessage",
                "sqs:SendMessage",
                "sqs:GetQueueAttributes",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:sqs:us-east-1:535601802221:cbc-demo-queue-alerts",
                "arn:aws:sqs:us-east-1:535601802221:cbc-demo-queue-events",
                "arn:aws:sqs:us-east-1:535601802221:cbc-demo-queue-watchlist-hits",
                "arn:aws:s3:::cbc-demo-bucket/*",
                "arn:aws:kms:us-east-1:535601802221:key/fb1aaaaa-8c3a-428a-b0ba-a0a136e9aaaa"
            ]
        },
        {
            "Sid": "17102cece21c40bb8f5c16f8a9ffb8bc",
            "Effect": "Allow",
            "Action": "sqs:ListQueues",
            "Resource": "*"
        }
    ]
}
```

Additional References

1. Carbon Black Cloud Technical Documentation
   - Add a Data Forwarder
   - Create an S3 Bucket in the AWS Console
   - Configure the Bucket Policy to Allow Access
2. Developer Network
   - Data Forwarder API
   - Data Forwarder Data Guide & Schema
   - Subscribe to the monthly Dev Network Newsletter
   - Developer Network Community / User Exchange
3. Carbon Black Cloud Splunk
   - Splunk App Documentation & Deployment Guide
   - Splunk App Configuration Video
   - VMware Carbon Black Cloud App
   - VMware Carbon Black Cloud Input Add-on (IA)
   - VMware Carbon Black Cloud Tech Add-on (TA)
4. Tech Zone
   - Getting Started: Custom Filters for the Data Forwarder
   - Useful Queries for the VMware Carbon Black Cloud Splunk App

Change Log

The following updates were made to this guide:

Date	Description of Changes
2022-01-19	

About the Author and Contributors

Bruce Deakyne is a Product Line Manager at VMware Carbon Black Cloud, focused on improving the ecosystem of APIs & integrations. Outside of cyber security, he enjoys cycling through the mountains of Boulder, CO.