RSA NetWitness Platform

Event Source Log Configuration Guide

Universal CloudWatch Logs

Last Modified: Thursday, December 31, 2020

Event Source Product Information:

Vendor: AWS
Event Source: Amazon CloudWatch Logs
Versions: API v1.0

Note: RSA is qualifying support for the major version. In case of any configuration changes or logs not parsing in a minor version, please open a case and we will add support for it.

RSA Product Information:

Supported On: Netwitness Platform 11.5.0 and later
Event Source Log Parser: aws, aws_cloudtrail, aws_route53resolver
Collection Method: Plugin Framework
Event Source Class.Subclass: Host.Cloud

Introduction

The Amazon CloudWatch Logs service allows you to collect and store logs from your resources, applications, and services in near real-time. There are three main categories of logs:

1) **Vended logs**- These are natively published by Amazon Web Services (AWS) on behalf of the customer. Currently, AWS supports Amazon Virtual Private Cloud (VPC) Flow and Amazon Route 53 logs.

2) **Logs published by AWS services**- Currently, over 30 AWS services publish logs to CloudWatch. These services include Amazon API Gateway, AWS Lambda, AWS CloudTrail, and others.

3) **Custom logs**- These are logs from your own application and on-premise resources.

For more information, see https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/WhatIsCloudWatchLogs.html

In RSA Netwitness Universal CloudWatch logs integration, AWS logs or external application logs must be forwarded to your AWS CloudWatch log group before configuring the RSA Netwitness plugin. You can then configure the RSA Netwitness plugin using CloudWatch log group access details. It collects logs from CloudWatch log group through AWS API calls. The route of these logs before being parsed in RSA Netwitness is:

Source → AWS CloudWatch Log group → RSA Netwitness Universal CloudWatch Plugin → RSA NetWitness AWS JSON log parsers/RSA user custom log parsers.

Note: Currently RSA NetWitness parsers supports event sources like AWS CloudTrail, VPC Flow Logs and Route53. You can still forward any type of event source logs from your CloudWatch log group to Netwitness Universal CloudWatch plugin and parse it either by creating a custom parser of your own or open a case with RSA to add support for parsing.

Setting up AWS CloudWatch Logs

AWS Regions: Login to your Amazon Web Service (AWS) account and select a CloudWatch supported AWS region before configuring the CloudWatch Logs.

Note: Select the region closest to your country. Refer https://docs.aws.amazon.com/general/latest/gr/cwl_region.html . Note the AWS Region code of your CloudWatch log group as it is required when you configure the RSA Netwitness Universal CloudWatch Plugin.

Programmatic access: You need the access key and the secret key to collect AWS CloudWatch logs through Application Programming Interface (API) calls in RSA Netwitness.

Note: As the access key provides access to your AWS services, it should be kept secure. To know more about how to get access to your AWS account, referhttps://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html.

AWS Access Key permissions: This is a one-time configuration as once you configure the access key to read logs from CloudWatch, it works for all AWS service Logs collected in the CloudWatch log group. You need read-only access to read logs from CloudWatch Logs using user programmatic access.

For information on accessing CloudWatch logs, see 'Allow Read-Only Access to CloudWatch Logs'.

For information on embedding an inline policy see Amazon AWS Identity and Access Management User Guide.

Add the following policy to your user role to allow a read-only access to CloudWatch Logs data.

```
{
"Version":"2012-10-17",
"Statement":[
{
"Action":[
"logs:Describe*",
"logs:Get*"
],
"Effect":"Allow",
"Resource":"*"
```

```
}
]
}
```

Route Logs to AWS CloudWatch

RSA Supported Event Source	Steps to route Logs to AWS CloudWatch	Expectation
AWS CloudTrail	https://docs.amazonaws.cn/en_us/awscloudtrail/latest/userguide/send-cloudtrail-events-to-cloudwatch-logs.html	
AWS VPC Flow Logs	https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs-cwl.html	Select AWS default format in VPC flow logs configuration in AWS
AWS Route 53	https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/logging-monitoring.html	

Setting Up Amazon Universal CloudWatch Plugin

To set up Amazon Universal CloudWatch plugin in RSA Netwitness Suite, perform the following tasks.

• Deploy RSA Netwitness Universal CloudWatch files from RSA Live

• Configure Universal CloudWatch plugin in RSA Netwitness Suite UI

Deploying RSA Universal CloudWatch config files

Universal CloudWatch Logs plugin requires resources available in RSA Live in order to collect logs and parse it.

To deploy the amazoncloudwatch content from Live:

1. In the **RSA NetWitness Suite** menu, select **Live**. Browse **Live** for **Universal CloudWatch** Logs plugin by typing ***amazoncloudwatch*** into the Keywords text box and click **Search**.

2. Select the item returned from the Search.

3. Click **Deploy** to deploy the **Universal CloudWatch** plugin to the appropriate Log Collectors, using the Deployment Wizard.

4. Log Parsers '**aws, aws_cloudtrail, aws_route53resolver**' have been added as required resources of amazoncloudwatch in RSA Live. Deploy these parsers to appropriate Log Decoders when you deploy plugin log collection file.

For more details, please refer the Add or Update Supported Event Source Log Parsers topic, or the ***Live Resource Guide*** on RSA community Link.

Note: On a hybrid installation, you need to deploy the package on both the VLC and the LC. Also, Netwitness aws JSON parsers will be enabled in your LD automatically after deployment.

Configure Universal CloudWatch plugin in RSA NetWitness Suite UI

1. In RSA NetWitness Suite menu, select **ADMIN > Services**.

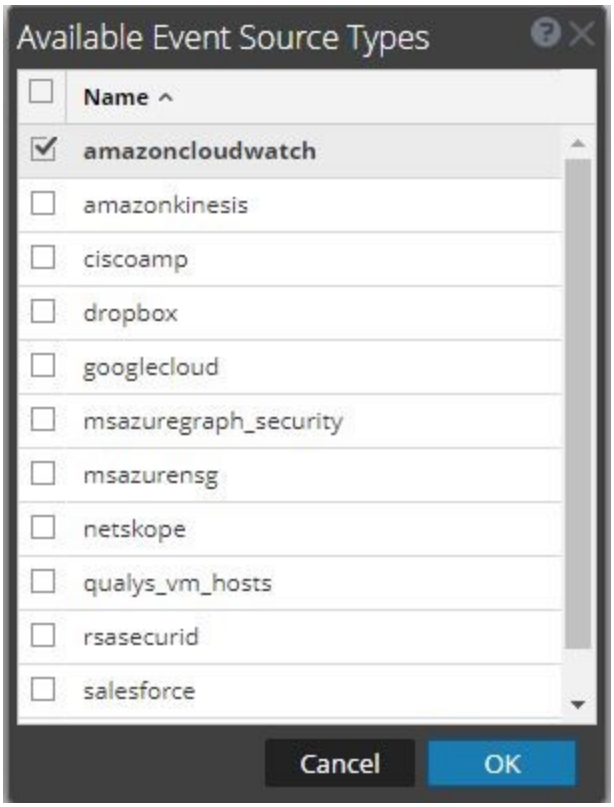
2. In the **Services** grid, select a **Log Collector service**, and choose **Config** option from the system menu.

3. In the **Event Sources** tab, select **plugins** from the dropdown menu. The Event categories panel displays the File event sources that are configured, if any.

4. In the Event Categories panel tool bar, click +,

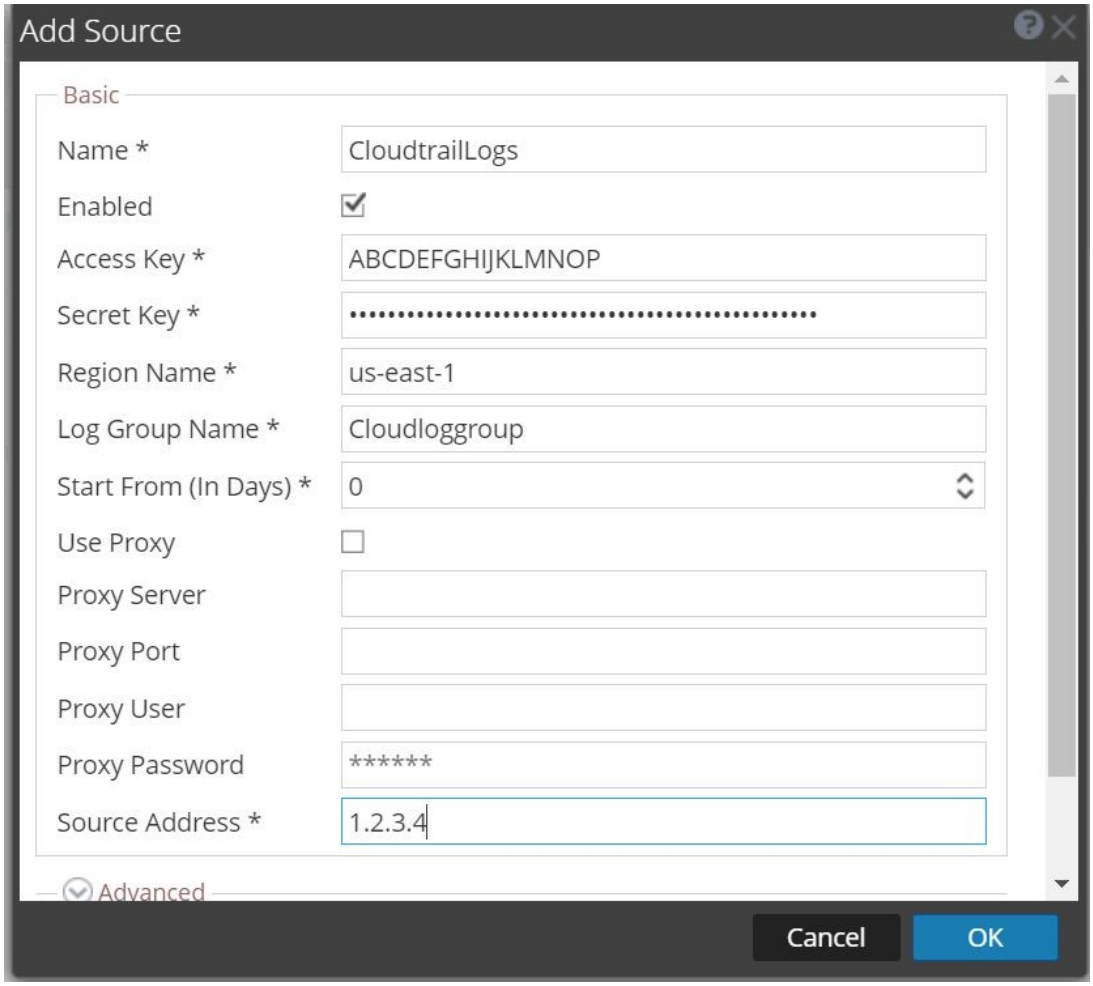
The available Event Source Types dialog is displayed.

Note: You can skip this step and go to step 6 directly, if you already configured an event source using Universal CloudWatch plugin.



5. Select **amazoncloudwatch** from the list and click **OK**. The newly added event source type is displayed in the **Event Categories** panel.

6. Select the new type in the **Event Categories** panel and click +, in the Source panel tool bar, the Add Source dialog is displayed.



7. Define parameter values, as described in Amazon CloudWatch Collection Configuration.

8. Click Test Connection, the result of the test will display in the dialog box. If the test is not successful, edit the device or service information based on error messages displayed and retry.

Note: The log collector takes approximately 60 seconds to return the test results. If it exceeds the time limit, the test times out and RSA Netwitness Suite displays a Request Timed Out Error.

9. If the test is successful, click **OK**. The new event source will be displayed in the **Sources** panel.

10. Repeat steps 6–9 to add another instance of Amazon CloudWatch plugin type.

Amazon CloudWatch Collection Configuration Parameters

This section describes the Amazon CloudWatch plugin configuration parameters.

Basic Parameters

Name	Description
Name*	Enter an alpha-numeric, descriptive name for the source. This value is only used for displaying the name on the screen.
Enabled	Select the checkbox to enable the event source configuration to start collection. The checkbox is selected by default.
Access Key*	AWS IAM Access Key belonging to your AWS user account
Secret Key*	AWS Secret Key corresponding to the above access key.

Name	Description
Region Name*	AWS Region code where the CloudWatch log group is located. Example: us-east-1
Log Group Name*	Input the AWS CloudWatch LogGroup Name here.
Start From (In Days)	'N' number of days to backtrack and pull data from the Log Group. Default value is 0 which collects logs of last 30 mins of current day. Max supported value is 180 days. **Note:** Do not edit the "Start From" value of a running instance. Please create a new event source instance for new "Start From" values because this is used to keep track of logs collection time and bookmark it in plugin to avoid duplicate logs collection

Name	Description
Use Proxy	Check to enable proxy configuration.
Proxy Server	If you are using a proxy, enter the proxy server address.
Proxy Port	Enter the proxy port.
Proxy User	Username for the proxy (leave empty if using anonymous proxy).
Proxy Password	Password for the proxy (leave empty if using anonymous proxy).
Source Address	IP address that is to be given for the current plugin event source. You can use any valid IP value. This IP will be mapped to device.ip meta in RSA Netwitness aws JSON parser and will be useful in indexing of logs
Test Connection	Checks the configuration parameters specified in this dialog to make sure that those are correct to collect logs from the AWS server.

Advanced Parameters

Click **Advanced** to view and edit the advanced parameters, if necessary.

Name	Description
Polling Interval	Interval (amount of time in seconds) between each poll. The default value is 180. For example, if you specify 180, the collector schedules a polling of the event source every 180 seconds. If the previous polling cycle is still underway, the collector waits for that cycle to finish. If you have many event sources that you are polling, it may take longer than 180 seconds for the polling to start because the threads are busy.
Max Duration Poll	Maximum duration, in seconds, of a polling cycle. A zero value indicates no limit. The default is set to 600.
Max Events Poll	The maximum number of events per polling cycle (how many events collected per polling cycle).
Max Idle Time Poll	Maximum Idle time, in seconds, of a polling cycle. A zero value indicates no limit.
Command Args	Optional arguments to be added to the script invocation.

Name	Description
Debug	**Caution**: Only enable debugging (set this parameter to On or Verbose) if you have a problem with an event source and you need to investigate this problem. **Caution**: Enabling debugging will adversely affect the performance of the Log Collector. Enables or disables debug logging for the event source. Valid values are: **Off** = (default) disabled **On** = enabled **Verbose** = enabled in verbose mode - adds thread information and source context information to the messages. This parameter is designed to debug and monitor isolated event source collection issues. If you change this value, the change takes effect immediately (no restart required). The debug logging is verbose, so limit the number of event sources to minimize performance impact. **Note:** If you enable debug, it adds a few more debug info from CloudWatch logs API as extra fields in RSA Netwitness logs. Extra debug fields are nw.eventTimestamp, nw.ingestionTime **nw.eventTimestamp** – The time the event occurred, expressed as the number of milliseconds after Jan 1, 1970 00:00:00 UTC. **nw.ingestionTime-** The time the event was ingested to CloudWatch, expressed as the number of milliseconds after Jan 1, 1970 00:00:00 UTC.
Trail By (In Minutes)	Specifies the lag between current time and log collection in RSA Netwitness Plugin. The default value is 60 minutes. Range is set between 5 to 120 minutes. You need to tune this value if you are seeing logs which are near to real time is missing in RSA Netwitness Investigation page. **Note:** It was observed that some CloudWatch logs take a while to be available in AWS log group. If you collect too close to the current time, there is a possibility of missing some events. Please refer following link for more details https://docs.aws.amazon.com/AmazonCloudWatchLogs/latest/APIReference/API_LogStream.html#CWL-Type-LogStream-lastEventTimestamp

Name	Description
AWS Event Type	Optional Parameter. Default value is 'NA'. This field should be filled only when the user wants to route logs belonging to AWS LogGroup to a custom log parser of their own in RSA Netwitness. Filled value will come as one of fields in header of your log in the RSA Netwitness log decoder after collection. If you do not give any value here and event source is RSA supported one, logs will be parsed automatically using RSA supported log parsers given in the page 1 of this doc.
SSL Enable	Uncheck to disable SSL certificate verification which is not suggested by RSA.

Additional Information

Following information is used to parse logs in RSA supported log parsers into metas. We expect the default aws format for logs from AWS server to parse logs in RSA log parsers.

Logs Formats

vpcflowlogs	https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs-records-examples.html
cloudtrail	https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-log-file-examples.html
route 53	a) https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/query-logs.html#query-logs-format b) https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resolver-query-logs-example-json.html

November 2020

Trademarks