Micro Focus Security ArcSight SmartConnectors

SmartConnector for Amazon Web Services Security Hub

Configuration Guide

Document Release Date: February 2022
Software Release Date: February 2022

Legal Notices

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK

https://www.microfocus.com

Copyright Notice

© Copyright 2022 Micro Focus or one of its affiliates

Confidential computer software. Valid license from Micro Focus required for possession, use or copying. The information contained herein is subject to change without notice.

The only warranties for Micro Focus products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein.

No portion of this product's documentation may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's internal use, without the express written permission of Micro Focus.

Notwithstanding anything to the contrary in your license agreement for Micro Focus ArcSight software, you may reverse engineer and modify certain open source components of the software in accordance with the license terms for those particular components. See below for the applicable terms.

U.S. Governmental Rights. For purposes of your license to Micro Focus ArcSight software, “commercial computer software” is defined at FAR 2.101. If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer software documentation and other technical data subject to the terms of the Agreement as specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data) of the Federal Acquisition Regulation (“FAR”) and its successors. If acquired by or on behalf of any agency within the Department of Defense (“DOD”), the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of the Agreement as specified in 48 C.F.R. 227.7202-3 of the DOD FAR Supplement (“DFARS”) and its successors. This U.S. Government Rights Section 18.11 is in lieu of, and supersedes, any other FAR, DFARS, or other clause or provision that addresses government rights in computer software or technical data.

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S.Government rights, patent policy, and FIPS compliance, see https://www.microfocus.com/about/legal/.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number
- Document Release Date, which changes each time the document is updated
- Software Release Date, which indicates the release date of this version of the software

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

https://www.microfocus.com/support-and-services/documentation

Support

Contact Information

Phone	A list of phone numbers is available on the Technical Support Page: https://softwaresupport.softwaregrp.com/support-contact-information
Support Web Site	https://softwaresupport.softwaregrp.com/
ArcSight Product Documentation	https://www.microfocus.com/documentation/arcsight/

Contents

Product Overview 6

Understanding Data Collection 6

Configuration 8

- Prerequisites 8
- Configuring SecurityGroup for Lambda 11
- Configuring an EC2 Instance 12
- Opening Ports 13

Installing Security Hub SmartConnector 14

Upgrading the Security Hub SmartConnector 18

Undeploying the AWS Security Hub SmartConnector 19

ASFF Keys to ArcSight Fields 20

- Header 20
- GuardDuty Default 20
- GuardDuty AWS_API_CALL 21
- GuardDuty DNS_REQUEST 21
- GuardDuty NETWORK_CONNECTION 22
- GuardDuty PORT_PROBE 23
- Inspector Default 24
- Inspector Product Fields 25
- Resource Header 26
- ResourcesDetailsAwsEc2Instance 26
- ResourcesDetailsAwsIamAccessKey 26
- ResourcesDetailsAwsEc2NetworkInterface 27
- ResourcesDetailsAwsEc2SecurityGroup 27
- ResourcesDetailsAwsIamRole 27
- ResourcesDetailsAwsKmsKey 28
- ResourcesDetailsAwsS3Bucket 28
- ResourcesDetailsAwsS3Object 28
- ResourcesDetailsAwsSnsTopic 29
- ResourcesDetailsAwsSqsQueue 29
- ResourcesDetailsAwsLambdaFunction 29

Updating or Overriding Parser 30

- Apply Monthly Parser Updates from Micro Focus 30
- Apply the Parser Updates from a Support Team or Other Sources 31

Enabling New Security Services Support 32

Troubleshooting 34
Enabling Detailed Logging 34
Viewing and Copying Failed Messages 34
The connector configuration on CentOS/RHEL AWS EC2 instances fails and displays the error "Connection refused" or "Unable to get the list of supported connectors for VM [Container1]" 35

Send Documentation Feedback 37

Product Overview

The SmartConnector for AWS Security Hub currently supports Amazon GuardDuty and Amazon Inspector. The Amazon GuardDuty and Amazon Inspector services create security findings after these services are enabled in the AWS cloud environment.

AWS Security Hub is AWS's own security finding aggregator. When enabled in an AWS cloud environment, it aggregates, organizes, and prioritizes these security findings, from multiple AWS security services, and forwards them to AWS EventBridge in ASFF format.

The SmartConnector for AWS Security Hub integrates the Security Hub with the AWS EventBridge to facilitate collection of security findings. The AWS Security Hub connector converts the security findings, originally in ASFF format to CEF format, and forwards them to an ArcSight destination. Consequently, enabling other ArcSight destinations, such as ESM or Logger so that they can process these security findings.

Understanding Data Collection

The following diagram provides a high-level overview of how the AWS Security Hub connector collects and sends GuardDuty and Inspector events through EventBridge to an ArcSight destination.

AWS Cloud
AWS Macie
AWS Inspector
AWS GuardDuty
Step 1: GuardDuty, Inspector, Macie, and so forth are AWS security services. These services create security findings after they are enabled in the AWS cloud environment.
AWS EventBridge
Step 2: Security Hub collects security findings from the enabled AWS security services by default. These security findings are stored in Security Hub for at least 90 days.
AWS Security Hub
Step 4: The rules defined in EventBridge forwards these findings to a target destination. For example, SQS Queue.
Step 3: Secuity Hub forwards the security findings to EventBridge by default. The security findings are created in the AWS Security Finding Format (ASFF).
AWS Source Queue
AWS Dead Letter Queue
Step 8: If there is any error in step 6 and 7, then the incoming message is retried and sent back to an SQS Queue. The default policy is to retry an erroneous message in an SQS Queue for 480 times at every 30 minutes. Messages that cannot be delivered beyond this limit are held in a Dead Letter Queue for 14 days, to be retrieved and analyzed manually.
Step 5: SQS Queue acts as an event source for Lambda. Lambda polls an SQS Queue and triggers a SmartConnector running on it to process a batch of ASFF messages.
Step 7: The CEF messages are sent to Syslog NG Daemon listening on the port 1999 over TLS.
VPC
Private Subnet
Public Subnet
Lambda ENI
NAT Gateway
SmartConnector Running on AWS Lambda
EC2 Instance hosting Syslog NG Daemon running on port 1999
Step 6: SmartConnector running on AWS Lambda reads a mapping files from S3 and uses it to convert ASFF to CEF message.
Step 9: The batched CEF messages are forwarded to a destination through a NAT gateway over HTTPS.
AWS S3
Step 0: All the mapping and configuration files are uploaded to S3 bucket.
Internet Gateway
HTTPS
HTTPS
Customer Network
Port 443
ArcSight destination running on port 443

Permissions to run the CloudFormation Template

The following permission are required to run the CloudFormation Template:

- AmazonS3FullAccess
- CloudwatchEventsFullAccess
- AmazonSQSFullAccess
- LambdaFullAccess
- CloudwatchFullAccess
- EC2FullAccess

Configuration

Note: If your organization has multiple regions, then you must integrate the connector and AWS Security hub only in the aggregation Region.

Prerequisites

1. Enable GuardDuty and Inspector in your AWS environment.
2. Enable Security Hub in your AWS environment.
3. Ensure that you have the AWS permissions to run the Cloud Formation Template.
4. Create an existing VPC with a private and public subnet created.
5. Create an EC2 instance in a private subnet, the route table points to a NAT gateway.

   SyslogNGDaemon is installed here.
6. Create a EC2 instance in a public subnet. A public subnet is associated with an internet gateway.

   SSH connects to the EC2 in a private instance where SyslogNGDaemon is installed.

   Note: For more information, see "Configuring an Amazon VPC and Subnets " on page 10
7. Create two security groups. One for Lambda and the other for the EC2 instance. For more information, see "Configuring SecurityGroup for Lambda" on page 11.
8. Create or select an S3 Bucket. For more information, see Creating and configuring an S3 bucket.

The `external.properties` file, **Maps** folder, ArcSight SmartConnector certificate, `arcsight-aws-securityhub-connector-1.1.0.jar`, and `installer.json` are stored here.

> **Note** : The S3 bucket region must be the same in which the AWS Security Hub SmartConnector is supposed to run and collect process data from.

9. Upload the `arcsight-aws-securityhub-connector-1.1.0.jar` file to the S3 bucket.

   This file is available in the AWS Security Hub Connector installer zip file.

10. Upload the `installer.json` file to your S3 bucket.

    This file is available in the AWS Security Hub Connector installer zip file.

11. Upload the maps folder to your S3 bucket. You can either create a maps folder and upload the content of the folder or copy the `Maps` folder itself.

    This file is available in the AWS Security Hub Connector installer zip file.

12. Create a certs folder and upload the Syslog NG Daemon Connector certificate to your S3 bucket.

    The certificate is located in the installation folder of the Syslog NG Daemon SmartConnector:

    `$INSTALLATIONFOLDER/current/user/agent/remote_management.p12`.

13. Upload the `external.properties` file to your S3 bucket.
    This file is used by a Java function running in Lambda and invokes the SyslogNGDaemon SmartConnector.

14. To update this file:

```
##
## external properties file to upload to s3
##
##
##Valid properties
##
##Arcsight connector hostname or ip address, required parameter
#host.name=0.0.0.0
##
##Arcsight connector port number, required parameter
#port.number=9999
##
##Arsight connector certificate bucket location in s3, required parameter
#certificate.bucket=bucket_name
```

```
##
##Arsight connector certificate key location in s3, required parameter
#certificate.key=path/to/file
##
##Arsight connector certificate password, required parameter
#certificate.password=password
##
##Log Level changes the log level to the specified level
##value can be any of: info debug error all warn fatal trace off
##case insensitive value
##required parameter
#log.level=info debug error all warn fatal trace off
```

host.name: [IP of the private EC2 instance running SyslogNGDaemon]

port.number:1999

certificate.override.bucket.name: [S3bucket name]

certificate.override.key.file: certs/remote_management.p12

certificate.override.keystore.password: crY2cvNdo8wpFdYdDrslv8cdMAV9f33A56hC+zXsqK4=

map.override.bucket.name: [S3bucket name]

map.override.bucket.directory: maps

log.level: info

15. If the private EC2 instance is a Windows machine, then open the relevant ports. For more information, see "Opening Ports " on page 13

    The Linux EC2 machines do not require this step.

Configuring an Amazon VPC and Subnets

To configure an existing Virtual Private Cloud (VPC), you must create a private subnet and associate it with the lambda function.

To create a public subnet:

1. Create an internet gateway if you do not have one.
2. From the VPC console, go to the navigation pane and select **Subnets**.
3. To create a new subnet, select **Create Subnet**. Otherwise, select an existing subnet.
4. Click the **Route Table** tab, then click **Edit**.

5. Click **Change to:**, then type or select an appropriate route table.

   The default route must point to an internet gateway.

To create a NAT gateway:

1. From the VPC console, go to the navigation pane and select **NAT Gateways** > **Create NAT Gateway**.
2. Click **Subnet**, then type or select the public subnet you created.
3. Click **Elastic IP Allocation ID** field, select an existing Elastic IP address or **Create New EIP**, and then click **Create a NAT Gateway**.

   When the status changes to **available**, you can use this EIP allocation ID.

To create a route table:

1. From the VPC console, select **Route Tables** > **Create Route Table**.
2. In the **Name tag** field, enter the appropriate name.
3. Click **VPC** , select your VPC, and then click **Yes, Create**.
4. Select the new route table, then click the **Routes** tab.
5. Click **Edit**, then select **Add another route**.

   Destination: 0.0.0.0/0

   Target: private subnet with the NAT gateway created in the previous step

Configuring SecurityGroup for Lambda

Create a SecurityGroup for Lambda with the following inbound or outbound rules:

Create a SecurityGroup for an EC2 instance with the following inbound or outbound rules:

Inbound rules Edit inbound rules

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	0.0.0.0/0	SSH Management Traffic
SSH	TCP	22	[illegible] (SG_Conn_NGSyslog)	SSH Between EC2 instances
Custom TCP	TCP	1999	[illegible] (SG_Conn_LambdaNGSyslog)	From Lambda
RDP	TCP	3389	0.0.0.0/0	RDP Traffic

Note: SSH is enabled to facilitate the Syslog NG Daemon SmartConnector installation in a private EC2 instance on Linux.
RDP is enabled to facilitate the Syslog NG Daemon installation in a private EC2 instance on Windows.
The Lambda connects to Syslog NG Daemon's port (1999) over TLS as shown in "Understanding Data Collection" on page 6.
After the Syslog NG Daemon SmartConnector is installed in the EC2 instance, you must remove the SSH and RDP rules for security purposes.

Configuring an EC2 Instance

To launch EC2 services from the AWS console:

1. Click **Instances**.
2. Click **Launch Instance**.
3. Click **Community AMI**.
   It displays a list of operating systems.
4. From Red Hat, select **Redhat 7.6** operating system, if you want to deploy an EC2 instance on Linux.
5. Select the Instance type as **t2.micro**.
6. Click **Next: Configure Instance details**, then select network **custom_vpc**.
7. Select the subnet.
8. Enable Auto-assign public IP address.
9. From the network interface, click **Add IP** for a secondary IP address.
10. Click **Next: Add Storage** and keep the original values.
11. Click **Next: Add Tags** and optionally enter the key and value details.
12. Click **Next: Add Security Group**.
13. Click **Create a New Security Group** or select an existing group.
14. Click **Review and Launch**.

15. Click **Launch**.
16. Select **Create a New Key Pair**.
17. Enter the keypair name as **test.pem**.
18. Click **Download Key Pair**.

Opening Ports

You must ensure that the ports on the server on which you installed the Syslog NG Daemon SmartConnector is accessible from AWS. The procedure to open ports varies based on whether you have installed Syslog NG Daemon SmartConnector on a virtual machine or not.

Opening Ports on a Non-Virtual Machine

If you have installed the Syslog NG Daemon SmartConnector on a physical or non-virtual machine, then ensure that the ports on which you installed it are accessible to AWS.

Opening Ports on a Virtual Machine

If you have installed the Syslog NG Daemon SmartConnector on a virtual machine in AWS, then ensure that the ports on which you installed Syslog NG Daemon SmartConnector are open in both AWS and virtual machine.

To open inbound ports on AWS:

1. Log in to the AWS as a user with administrative rights.
2. Go to **Services** and select **EC2**.
3. Select **Instances**.
4. Choose the EC2 instance you want to edit.
5. Click **Launch-Wizard**.
6. Edit the **Inbound** and **Outbound** fields as required.

To open ports in the virtual server:

1. Log in to the virtual AWS machine.
2. Open the AWS Firewall.
3. Click Inbound **Rules** > **New Rule** > **Port** > **Next** > **TCP** > **Specific local ports**.
4. Enter the same port or port range on which you installed the Syslog NG Daemon SmartConnector.
5. Click **Next** > **Allow the connection** > **Next** > **Profile** > **Next**.
6. Enter the name of a rule.

7. Click **Finish**.

> **Note**: You need to follow theses steps only for the Windows instances and not for the Linux EC2 instances.

Installing Security Hub SmartConnector

The Security Hub SmartConnector solution uses AWS cloud resources such as Security Hub, EventBridge (CloudwatchEvent), SQS, and Lambda to source and parse ASFF messages to CEF messages and then to forward them to Syslog NG Daemon.

Syslog NG Daemon batches and forwards these CEF messages to an Arcsight destination.

To complete the Security Hub SmartConnector installation, complete the procedures in the following sections:

Installing Syslog NG Daemon as a Forwarding Agent

1. Launch the EC2 Instance in a public subnet and a private subnet.
2. Log in to the public EC2 instance using the key.

   You can use either PuTTy or MobaXterm software tool.
3. Upload the key of the private EC2 instance to the public EC2 instance.
4. From the public EC2 instance, run the `chmod 600 testprivate.pem` command.
5. SSH to the private instance using the `ssh ec2-user@private-ip-address -i testprivate.pem` command.
6. Upload the Syslog NG Daemon installer to public EC2 instance. You can use MobaXterm to upload this installer.
7. Copy the Syslog NG Daemon installer to the private EC2 instance by using the following command:

   `scp -i testprivate.pem ArcSight-7.15.0.8287.0-Connector-Linux64.bin ec2-user@private-ip-address:/home/ec2-user/.`
8. Configure the Syslog NG Daemon SmartConnector in the private instance.
9. Select **1.0** as the CEF File version.
10. Configure the Protocol as default TLS.
11. Configure the Port 1999.
12. Select **CSV File/CEF File** as the destination, unless you are using any other ArcSight product like Logger or ESM.

> **Note**: To emit the Avro output, you need to select Transformation Hub as the destination. For more information, refer to the *Transformation Hub* section in the Smart Connector User Guide.

13. Run the SmartConnector as a standalone process or as a service.
14. After the deployment is complete, upload the `remote_management.p12` certificate file that is in the `$ARCSIGHT_HOME/current/user/agent` path to the S3 bucket certs folder.

Creating AWS Cloud Resources Using CFT

Cloud Formation Template (CFT) is used to generate the AWS cloud resources. For more information, see "Understanding Data Collection" on page 6. The AWS Security Hub SmartConnector installer is downloaded as the **connector.zip** file.

The **connector.zip** file contains:

- A jar file containing the java function to be deployed in Lambda.
- The **installer.json** file, which is a CFT.
- The **installer_GovCloud.json** file, which is a CFT.

To create AWS Cloud Resources using CFT

1. Open the **connector.zip** file. Use any of the following files to create AWS cloud resources, as required:
   - **installer.json** - If you want to install the connector on AWS standard region.
   - **installer_GovCloud.json** - If you want to install the connector on AWS GovCloud region.
2. Log in to the AWS Web console as a user with appropriate permission. For more information, see "Permissions to run the CloudFormation Template" on page 8.
3. From **Find Services**, search **CloudFormation Service**.
4. From the CloudFormation service console, click **Create Stack**.
5. Click **With New Resources (Standard)**.
   The following window is displayed.

6. For **Amazon S3 URL**, enter the Object URL of the installer.json file uploaded to your S3 bucket.
7. Click the i**nstaller.json** file to get the URL of S3 object.

8. Click **Next**.
   The following window is displayed:

9. Enter the **Stack Name**.

   Note: A stack name must be unique for each region.

10. Enter the name of the S3 bucket in which the installer.json is located.
11. Enter the **Region** code in which you want to install the connector.

    Note: It must be same as the S3 bucket region.

12. Select the Security Group created for Lambda.
13. Select the private subnet you created.
14. Click **Next** and keep the default values.
15. Select the **I acknowledge that AWS CloudFormation might create IAM resources** check box.
16. Click **Create Stack**.

    CloudFormation service starts deploying all the resources as per the CFT. When the resources are successfully deployed, the status changes to UPDATE_COMPLETE.

Security Hub Connector Post-Deployment Configuration

Currently, the SQS queue is configured to re-send failed messages 480 times in a 30 minutes interval for 10 days. This setting can be changed by updating the stack.

Note: This step is optional. You can keep the default settings.

To update the stack:

1. From the stack console, click **Update**.
2. Choose **Edit Template in Designer** and click **View Designer**.
3. Update the **maxReceiveCount** value.
4. Click **Create stack**, continue clicking **Next**, and then click **Update Stack**.

Upgrading the Security Hub SmartConnector

1. Update your Amazon S3 bucket. To update:
   a. Upload the latest version of the `JAR` and `Installer.json` files.
   b. Upload the latest `map` folder.
2. Rename the uploaded JAR file with a new version.

   Example: If the deployed JAR file name is `arcsight-aws-securityhub-connector-1.2.0.jar`, then you can rename it to `arcsight-aws-securityhub-connector-1.3.0.jar`.
3. Update the JAR version mentioned in the installer.json file.

   Example: Change `"S3Key": "arcsight-aws-securityhub-connector-1.1.0.jar"` to `"S3Key": "arcsight-aws-securityhub-connector-1.3.0.jar"`. Save the updated `Installer.json` file to an S3 bucket.
4. From the stack console, click **Update**, then select **Replace Current Template**.
5. For **Amazon S3 URL**, enter the ObjectURL of the Installer.json file that is uploaded to your S3 bucket.
6. Continue clicking **Next**.
7. Select the **I acknowledge that AWS CloudFormation might create IAM resources** check box to acknowledge the capabilities of AWS CloudFormation template.
8. Click **Update Stack** to submit an updated template.

   CloudFormation service starts deploying all the resources as per CFT. When the resources are successfully deployed, the status changes to UPDATE_COMPLETE.

Undeploying the AWS Security Hub SmartConnector

1. Go to CloudFormation console.
2. From the **Filter by Stack name** search box, enter the stack name.
3. From the stack console, click **delete**.
4. Click **Delete Stack**.

   All the AWS cloud resources will be deleted. The VPC components and the EC2 instances must be manually deleted as they were manually created.

ASFF Keys to ArcSight Fields

Header

ASFF Key	ArcSight Fields
Version	deviceCustomFloatingPoint1
Id	devicePayloadId
Detail-type	requestMethod
Account	deviceExternalID
Time	deviceReceiptTime
Region	deviceDnsDomain
Resources	requestCookies

GuardDuty Default

ASFF Key	ArcSight Fields
Product Arn	deviceFacility
Types	deviceCustomString2
Description	requestContext
SchemaVersion	deviceCustomDate2
Generator Id	deviceProcessName
First Observed At	startTime
Created At	fileCreateTime
Record State	oldFileHash
Title	message
Workflow / Status	oldFileId
Last Observed At	endTime
Severity / Normalized	DeviceCustomNumber2
Severity / Label	deviceSeverity

ASFF Key	ArcSight Fields
Updated At	fileModificationTime
Aws Account Id	destinationZoneExternalID
Id	oldFilePermission

GuardDuty AWS_API_CALL

ASFF Key	ArcSight Fields
aws/guardduty/service/action/actionType : aws/guardduty/service/action/resourceRole	name,deviceEventClassId
aws/guardduty/service/action/awsApiCallAction/serviceName	sourceServiceName
aws/guardduty/service/action/awsApiCallAction/callerType	requestClientApplication
aws/guardduty/service/action/awsApiCallAction/remoteIpDetails/ipAddressV4	sourceAddress
aws/guardduty/service/action/awsApiCallAction/remoteIpDetails/organization/	deviceCustomString1
aws/guardduty/service/action/awsApiCallAction/remoteIpDetails/country/countryName	deviceCustomString6
aws/guardduty/service/action/awsApiCallAction/remoteIpDetails/city/cityName	
aws/guardduty/service/resourceRole	
aws/guardduty/service/additionalInfo/	requestUrl
aws/guardduty/service/archived	filePermission
aws/guardduty/service/count	baseEventCount
aws/securityhub/FindingId	fileId
aws/securityhub/ProductName	deviceEventCategory
aws/securityhub/CompanyName	deviceHostName

GuardDuty DNS_REQUEST

ASFF Key	ArcSight Fields
aws/guardduty/service/action/actionType : aws/guardduty/service/action/resourceRole	name,deviceEventClassId
aws/guardduty/service/action/dnsRequestAction/domain	destinationDnsDomain
aws/guardduty/service/action/dnsRequestAction/protocol	transportProtocol
aws/guardduty/service/action/dnsRequestAction/blocked	deviceAction

ASFF Key	ArcSight Fields
aws/guardduty/service/resourceRole	deviceCustomString6
aws/guardduty/service/additionalInfo/	requestUrl
aws/guardduty/service/evidence/threatIntelligenceDetails.0_ /threatNames.0_	deviceCustomString4
aws/guardduty/service/evidence/threatIntelligenceDetails.0_ /threatListName	
aws/guardduty/service/archived	filePermission
aws/guardduty/service/count	baseEventCount
aws/securityhub/FindingId	fileId
aws/securityhub/ProductName	deviceEventCategory
aws/securityhub/CompanyName	deviceHostName

GuardDuty NETWORK_CONNECTION

ASFF Key	ArcSight Fields
aws/guardduty/service/action/actionType : aws/guardduty/service/action/resourceRole	name,deviceEventClassId
aws/guardduty/service/action/networkConnectionAction/connectionDirection	deviceDirection
aws/guardduty/service/action/networkConnectionAction/remoteIpDetails/ipAddressV4	sourceAddress
aws/guardduty/service/action/networkConnectionAction/remoteIpDetails/organization/asn	deviceCustomString1
aws/guardduty/service/action/networkConnectionAction/remoteIpDetails/organization/asnOrg	
aws/guardduty/service/action/networkConnectionAction/remoteIpDetails/organization/isp	
aws/guardduty/service/action/networkConnectionAction/remoteIpDetails/organization/org	
aws/guardduty/service/action/networkConnectionAction/remoteIpDetails/country/countryName	
aws/guardduty/service/action/networkConnectionAction/remoteIpDetails/geoLocation/lat	sourceGeoLatitude
aws/guardduty/service/action/networkConnectionAction/remoteIpDetails/geoLocation/lon	sourceGeoLongitude
aws/guardduty/service/action/networkConnectionAction/remotePortDetails/port	sourcePort

ASFF Key	ArcSight Fields
aws/guardduty/service/action/networkConnectionAction/remotePortDetails/portName	source ServiceName
aws/guardduty/service/action/networkConnectionAction/localPortDetails/port	destinationPort
aws/guardduty/service/action/networkConnectionAction/localPortDetails/portName	destinationServiceNam e
aws/guardduty/service/action/networkConnectionAction/protocol	transportProtocol
aws/guardduty/service/action/networkConnectionAction/blocked	deviceAction
aws/guardduty/service/action/networkConnectionAction/localIpDetails/ipAddressV4	destinationAddress
aws/guardduty/service/resourceRole	deviceCustomString6
aws/guardduty/service/archived	filePermission
aws/guardduty/service/count	baseEventCount
aws/securityhub/FindingId	fileId
aws/securityhub/ProductName	deviceEventCategory
aws/securityhub/CompanyName	deviceHostName

GuardDuty PORT_PROBE

ASFF Key	ArcSight Fields
aws/guardduty/service/action/actionType : aws/guardduty/service/action/resourceRole	name,deviceEventClassI d
aws/guardduty/service/action/portProbeAction/portProbeDetails.0_ /localPortDetails/port	destinationPort
aws/guardduty/service/action/portProbeAction/portProbeDetails.0_ /localPortDetails/portName	destinationServiceName
aws/guardduty/service/action/portProbeAction/portProbeDetails.0_ /remoteIpDetails/ipAddressV4	sourceAddress

ASFF Key	ArcSight Fields
aws/guardduty/service/action/portProbeAction/portProbeDetails.0_ /remoteIpDetails/organization/asn	deviceCustomString1
aws/guardduty/service/action/portProbeAction/portProbeDetails.0_ /remoteIpDetails/organization/asnOrg	
aws/guardduty/service/action/portProbeAction/portProbeDetails.0_ /remoteIpDetails/organization/isp	
aws/guardduty/service/action/portProbeAction/portProbeDetails.0_ /remoteIpDetails/organization/org	
aws/guardduty/service/action/portProbeAction/portProbeDetails.0_ /remoteIpDetails/country/countryNa me	
aws/guardduty/service/action/portProbeAction/portProbeDetails.0_ /remoteIpDetails/city/cityName	
aws/guardduty/service/action/portProbeAction/portProbeDetails.0_ /remoteIpDetails/geoLocation/lat	sourceGeoLatitude
aws/guardduty/service/action/portProbeAction/portProbeDetails.0_ /remoteIpDetails/geoLocation/lon	sourceGeoLongitude
aws/guardduty/service/action/portProbeAction/blocked	deviceAction
aws/guardduty/service/resourceRole	deviceCustomString6
aws/guardduty/service/additionalInfo/	requestUrl
aws/guardduty/service/evidence/threatIntelligenceDetails.0_/threatNames.0_	deviceCustomString4
aws/guardduty/service/evidence/threatIntelligenceDetails.0_/threatListName	
aws/guardduty/service/evidence/threatIntelligenceDetails.1_/threatNames.0_	
aws/guardduty/service/evidence/threatIntelligenceDetails.1_/threatListName	
aws/guardduty/service/count[AJ1]	baseEventCount
aws/securityhub/FindingId	fileId
aws/securityhub/ProductName	deviceEventCategory
aws/securityhub/CompanyName	deviceHostName

Inspector Default

ASFF Key	ArcSight Fields
detail/findings/ProductArn	deviceFacility
detail/findings/Types	deviceCustomString2
detail/findings/Description	requestContext

detail/findings/SchemaVersion	deviceCustomDate2
detail/findings/GeneratorId	deviceProcessName
detail/findings/FirstObservedAt	startTime
detail/findings/CreatedAt	fileCreateTime
detail/findings/RecordState	oldFileHash
detail/findings/Title	message
detail/findings/Workflow/Status	oldFileId
detail/findings/LastObservedAt	endTime
detail/findings/Severity/Normalized	deviceCustomNumber2
detail/findings/Severity/Label	deviceSeverity
detail/findings/UpdatedAt	fileModificationTime
detail/findings/AwsAccountId	oldFilePermission
detail/findings/Remediation/Recommendation/Url or detail/findings/Remediation/Recommendation/Text	deviceCustomString4

Inspector Product Fields

ASFF Key	ArcSight Fields
detail/findings/ProductFields/ResourceOwnerAccount	sourceUserId
detail/findings/ProductFields/aws/securityhub/FindingId	fileId
detail/findings/ProductFields/aws/securityhub/ProductName	deviceEventCategory
detail/findings/ProductFields/aws/securityhub/CompanyName	deviceHostName
detail/findings/ProductFields/attributes/INSTANCE_ID	deviceExternalId
detail/findings/ProductFields/attributes/CVE_ID	deviceCustomString1Label
detail/findings/ProductFields/attributes/OS_INFO	deviceCustomString6
detail/findings/ProductFields/attributes/PROTOCOL	applicationProtocol
detail/findings/ProductFields/serviceAttributes/schemaVersion	deviceCustomNumber3
detail/findings/ProductFields/serviceAttributes/assessmentRunArn	destinationServiceName
detail/findings/ProductFields/serviceAttributes/rulesPackageArn	sourceServiceName
detail/findings/ProductFields/attributes/package_name	sourceProcessName
detail/findings/ProductFields/aws/inspector/RulesPackageName	destinationProcessName
detail/findings/ProductFields/aws/inspector/arn	filePermission
detail/findings/ProductFields/attributes/REACHABILITY_TYPE	requestClientApplication

Resource Header

ASFF Key	ArcSight Fields
Id	filePath
Partition	deviceDomain
Tags	deviceCustomString3
Type	fileType

ResourcesDetailsAwsEc2Instance

ASFF Key	ArcSight Fields
I am Instance Profile Arn	oldFilePath
KeyName	event.oldFileName
Type	fileName
VpcId | SubnetId | ImageId	oldFileType
IpV6Addresses	deviceCustomString5
IpV4Addresses	deviceCustomString5
Launched At	deviceCustomDate1

ResourcesDetailsAwsIamAccessKey

ASFF Key	ArcSight Fields
User Name	sourceUserName
Created At	deviceCustomDate1
Principal Id	fileHash
Principal Name	oldFileName
Principal Type	oldFileType
Status	deviceCustomString5

ResourcesDetailsAwsEc2NetworkInterface

ASFF Key	ArcSight Fields
Attachment/AttachmentId | Attachment/InstanceId	deviceCustomString5
Attachment/AttachTime	deviceCustomDate1
Attachment/DeleteOnTermination | SourceDestCheck	oldFileType
Attachment/DeviceIndex	deviceCustomNumber1
Attachment/InstanceOwnerId	sourceUserId
Attachment/Status	fileName
Security Groups	fileHash
Network Interface Id	oldFilePath
SourceDestCheck	oldFileType

ResourcesDetailsAwsEc2SecurityGroup

ASFF Key	ArcSight Fields
Group Id	oldFilePath
Group Name	fileName
Ip Permissions	oldFileName
Ip Permissions Egress	deviceCustomString5
Owner Id	sourceUserId
Vpc Id	oldFileType

ResourcesDetailsAwsIamRole

ASFF Key	ArcSight Fields
Assume Role Policy Document	oldFileName
Create Date	deviceCustomDate1
Max Session Duration	deviceCustomNumber1
Path	oldfilepath
Role Id	destinationUserId
Role Name	destinationUserName

ResourcesDetailsAwsKmsKey

ASFF Key	ArcSight Fields
AWS Account Id	fileHash
Creation Date	deviceCustomDate1
Key Id	oldFileType
Key Manager	fileName
Key State	oldFilePath
Origin	deviceCustomString5

ResourcesDetailsAwsS3Bucket

ASFF Key	ArcSight Fields
Created At	deviceCustomDate1
Owner Id	sourceUserID
Owner Name	sourceUserName
Server Side Encryption Configuration	deviceCustomString5

ResourcesDetailsAwsS3Object

ASFF Key	ArcSight Fields
Content Type	oldFileType
E Tag	deviceCustomString5
Last Modified	deviceCustomDate1
Server Side Encryption	fileName
SSEKMS Key Id	oldFileName
Version Id	oldFilepath

ResourcesDetailsAwsSnsTopic

ASFF Key	ArcSight Fields
KmsMasterKeyId	fileHash
Owner	destinationUserName
Subscription	deviceCustomString5
Topic Name	fileName

ResourcesDetailsAwsSqsQueue

ASFF Key	ArcSight Fields
Dead Letter Target Arn	oldFilePath
Kms Data Key Reuse Period Seconds	deviceCustomNumber1
Kms MasterKey Id	fileHash
Queue Name	fileName

ResourcesDetailsAwsLambdaFunction

ASFF Key	ArcSight Fields
Code	deviceCustomString5
Code Sha 256	fileHash
Dead Letter Config /Target Arn	oldFileType
Environment / Variables	oldFileName
Environment / Error / Error Code	reason
Function Name	fileName
Handler, KmsKeyArn, Layers, RevisionId, Runtime, Timeout, TracingConfigMode, Version, VpcConfig, MasterArn	oldFilePath
Last Modified	deviceCustomDate1
Memory Size	fileSize
Role	destinationUserName

Updating or Overriding Parser

ArcSight SmartConnector for AWS Security Hub is designed to support parser file updates at run time without any code changes.

The updates either extend events of supported services or provide support for new service events. You can apply parser updates on the basic installation of connectors.

You can apply the following parser updates:

- Updates received from Micro Focus. See "Apply Monthly Parser Updates from Micro Focus" below.
- Updates received from a support team or other sources. See "Apply the Parser Updates from a Support Team or Other Sources" on the next page.

Apply Monthly Parser Updates from Micro Focus

The monthly updates for ArcSight SmartConnector parser releases are available at the ArcSight Marketplace. To set up your administrative account and download the parser updates, refer to ArcSight Marketplace at: https://marketplace.microfocus.com/arcsight

The updates for all parsers are available from Micro Focus as an **ArcSight-8.x.x.xxxx.0-ConnectorParsers.aup** package.

To apply the parser updates:

1. Download the **ArcSight-8.x.x.xxxx.0-ConnectorParsers.aup** package from the ArcSight Marketplace.
2. To apply monthly parser updates to Cloud Connectors:
   a. Download the **ArcSight-8.x.x.xxxx.0-aup-extractor.jar** utility from the location where you have downloaded the connector.

      > **Note**: Your system must have Java 1.8.x or later version installed and Java available in the operating system's path to use the **ArcSight-8.x.x.xxxx.0-aup-extractor.jar** utility.

   b. Specify the following command to use the utility to extract parser files from the package:

      ```
      java -jar ArcSight-8.x.x.xxxx.0-aup-extractor.jar <AUP filename>
      ```

      Examples:

      - `java -jar ArcSight-8.x.x.xxxx.0-aup-extractor.jar ArcSight-8.x.x.xxxx.0-ConnectorParsers.aup` - When the **.aup** package is in the same

directory where the JAR is present.

- `java -jar ArcSight-8.x.x.xxxx.0-aup-extractor.jar c:\MyFolder\ArcSight-8.x.x.xxxx.0.0-ConnectorParsers.aup` - When the **.aup** package is present in other directory.

You can either provide one or both the parameters. If you do not provide any parameters, the utility picks up any available. aup file and creates a new folder named **output** in the directory from where the utility is run and uploads the output files

The following folders will be extracted:

- **aws_cloudwatch**: Contains security parser for AWS Cloudwatch.
- **aws_securityhub**: Contains security parser for AWS Security Hub.
- **azure_emitter**: Contains security parser for Azure emitter.

c. Copy the map files from **output/aws_securityhub** folder and upload them to the AWS environment:

   i. Navigate to your AWS S3 bucket. For more information, see "Prerequisites" on page 8.

   ii. Browse to the **maps** folder and upload the new map files.
   The new map files overwrite the existing files.

   If the updated files require support for the new service events, then you must enable these events in the AWS environment. For more information, see Enabling New Security Services Support.

Apply the Parser Updates from a Support Team or Other Sources

In this case, the updated parser files are already available, which you need to add in the AWS environment.

To add updated parser files provided by a Support Team or other sources to the AWS environment:

a. Navigate to your AWS S3 bucket. For more information, see "Prerequisites" on page 8.

b. Browse to the **maps** folder and upload the new map files.

   If the updated files require support for the new service events, then you must enable these events in the AWS environment. For more information, see Enabling New Security Services Support.

Enabling New Security Services Support

The ArcSight SmartConnector for AWS Security Hub connector is designed to support new security services at run time without any code changes.

To enable a new security service:

1. Navigate to your AWS S3 bucket. For more information, see "Prerequisites" on page 8.
2. Browse to the **maps** folder, then upload the new service related map file to the folder. For example, to support inspector service, add `InspectorDefault.map` to the **maps** folder.
3. Open the **mapping.properties** file in a text editor. Add a new category to support the new service. For example, to support inspector service add the following line:

   ```
   [category=Inspector]
   InspectorDefault.map
   ```
4. Save the updated **mapping.properties** file to your S3 bucket.
5. Open the **Setup.map** file in a text editor. Add a new name string to support the new service. For example, to support inspector service add the following: `inspector_namestring1=/detail/findings/ProductFields/aws/inspector/id`
6. Save the updated **Setup.map** file to your S3 bucket.
7. Open the Installer.json in a text editor and modify the **ProductArn**. For example, to support the inspector service, change:

   ```
   "ProductArn" : [
   {"Fn::Join" : [ "", [ "arn:aws:securityhub:", {"Ref": "Region"},
   "::product/aws/guardduty" ] ]}]
   ```

   to

   ```
   "ProductArn" : [
   {"Fn::Join" : [ "", [ "arn:aws:securityhub:", {"Ref":
   "Region"},"::product/aws/guardduty" ] ]},
   {"Fn::Join" : [ "", [ "arn:aws:securityhub:", {"Ref": "Region"},
   "::product/aws/inspector" ] ]}]
   ```
8. Save the updated `Installer.json` file to your S3 bucket.
9. From the stack console, click **Update**, then select **Replace Current Template**.
10. From Amazon S3 URL, enter the Object URL of the installer.json file uploaded to your S3 bucket.
11. Continue clicking **Next**.
12. Select the **I acknowledge that AWS CloudFormation might create IAM resources** check box to acknowledge the capabilities of AWS CloudFormation template.

13. Click **Update Stack** to submit an updated template.

    CloudFormation service starts deploying all the resources as per the CFT. When the resources are successfully deployed, the status changes to UPDATE_COMPLETE.

Troubleshooting

Enabling Detailed Logging

By default, info level debugging is enabled to reduce logging data size and enhance performance.

To enable detailed level logging to help resolve any runtime issues:

1. Navigate to your S3 bucket
2. Download and open the `external.properties` file.
3. Set `log.level=all`, then save the file.
4. Upload the updated `external.properties` file to S3 bucket.

Viewing and Copying Failed Messages

If there are any errors while converting ASFF messages to CEF messages, the failed messages are returned to the source queue to be retried. If the number of retries exceed the default limit of 480, these failed messages are auto forwarded to a Dead Letter Queue, where it is stored for manual processing.

To view or copy failed messages:

1. In the **Resources** tab, for your stack, get the Dead Letter Queue name .
2. Go to **Simple Queue Services**, then search using the Dead Letter Queue name.
3. If the number of available messages is more than zero, then click the queue and click the **Send and Receive** tab.
4. To display a list of failed messages, click **Poll for Messages**.
5. To view and copy failed messages, click a failed message, then click the **Body** tab to copy and share the message content.

The connector configuration on CentOS/RHEL AWS EC2 instances fails and displays the error "Connection refused" or "Unable to get the list of supported connectors for VM [Container1]"

This issue may occur due to low memory.

If, while installing the connector on an EC2 instance of CentOS or RHEL OS, the error "Unable to get the list of supported connectors" is displayed, change the memory size value.

This error may also be displayed when installing the SyslogNGDaemon SmartConnector in EC2 instances are created with 1GB or 2GB memory.

Workaround:

Recently, we increased the Java heap memory to 1GB; hence, the EC2 instance should be 2GB.

Create the EC2 instance with +2GB and proceed with the connector installation

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this computer, click the link above and an email window opens with the following information in the subject line:

Feedback on Configuration Guide (SmartConnectors 8.3.0)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to Documentation-Feedback@microfocus.com.

We appreciate your feedback!