Integrating Tanium Connect with Amazon Security Lake

Version 1.2

PUBLIC

Version History

Version	Date	Pages	Change Agent	Changes
1.0	11/10/2022	all	Technical Media Team	initial document
1.1	11/17/2022	5,7	Technical Media Team	updated background and CLI text
1.2	11/22/2022	1,5,6,10	Technical Media Team	Updated AWS terms, added AWS references

Reviewer	Date
Christopher Nabkey, Chris Horn, Jason Reimer	11/2022

The information in this document is subject to change without notice. Further, the information provided in this document is provided “as is” and is believed to be accurate, but is presented without any warranty of any kind, express or implied, except as provided in Tanium’s customer sales terms and conditions. Unless so otherwise provided, Tanium assumes no liability whatsoever, and in no event shall Tanium or its suppliers be liable for any indirect, special, consequential, or incidental damages, including without limitation, lost profits or loss or damage to data arising out of the use or inability to use this document, even if Tanium Inc. has been advised of the possibility of such damages.

Any IP addresses used in this document are not intended to be actual addresses. Any examples, command display output, network topology diagrams, and other figures included in this document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Please visit https://docs.tanium.com for the most current Tanium product documentation.

Contents

Introduction

Tanium provides a unified endpoint management and security platform that offers customers transformational scale, speed and reliability. From more efficient workflows between teams to eliminating gaps created by standalone solutions, Tanium helps simplify infrastructure in the most demanding IT environments.

Integrate Tanium Connect with AWS S3 and AWS Lambda to create Open Cybersecurity Schema Framework (OCSF) Parquet files for Amazon Security Lake.

Background

Tanium's initial integration with Amazon Security Lake provides device/inventory data to the Amazon Security Lake. This data can be used to understand the device footprint in your environment, and to provide additional context enrichment to other events you see in the Amazon Security Lake.

Requirements for integration

- Tanium Core Platform 7.5 or later
- Tanium Connect
- Tanium AD Query Content sensor installed
- AWS S3
- AWS Lambda

AWS Setup

Create Amazon Security Lake Custom Source

Review the provided AWS documentation to create a custom source for your Amazon Security Lake (SDL). During the setup process for the custom source, complete the following basic steps:

1. Sign in to the AWS Management console.
2. Open the Amazon S3 console.
3. Find and make note of the **Source Name**.
4. Find and make note of the **Bucket Name**.

> **Note**: *The names of the* **Source Name** *and the incoming* **Bucket Name** *are used with the AWS CloudFormation.*

In our example they are

- `tanium-inventory-202211`
- `aws-security-data-lake-us-east-2-o-vp6qmr1234`

Download Tanium components

Download the template and .zip file provided by Tanium:

- AWS Cloud Formation Template
  `TaniumOcsfSecurityDataLakeStack.packaged.template.json`
- A zip with the AWS Lambda code
  `lambda.zip`

Put the files in the same directory.

Create the required AWS resources

Use the provided CloudFormation template to create the required AWS resources. This includes two S3 buckets and an AWS Lambda that converts Tanium's data to the format required for Amazon Security Lake.

1. Use AWS CLI.
2. Enter the commands below:

```
$ aws cloudformation package \
--template-file TaniumOcsfSecurityLakeStack.packageme.template.json \
--s3-bucket test-stage-location \
--output-template-file TaniumOcsfSecurityLakeStack.packaged.template.yaml

$ aws cloudformation deploy \
--template-file TaniumOcsfSecurityLakeStack.packaged.template.yaml \
--stack-name TaniumInventoryStack \
--capabilities CAPABILITY_IAM \
--parameter-overrides \
securityLakeSourceName=tanium-test-20221101 \
securityLakeBucketName=aws-security-data-lake-us-east-2-o-vp6qmr1234
```

3. Once the process is complete, confirm the source name and bucket exist.

Identify the physical ID for the S3 bucket

1. Sign in to the AWS Management console and open the Amazon S3 console.
2. In the **Buckets** list, find the **IncomingReportBucket**.
   In our example it is:
   `taniuminventorystack-incomingreportbucket0582d8a9-148c7ofb31234`

Capture the S3 credentials

1. Sign in to the AWS Management console and open the Amazon S3 console.
2. Browse to the **CloudFormation Stack**.
3. Find **IncomingReportUser**, click through to the IAM page.
4. Under **Security Credentials**, click **Create Access Key**.
5. Annotate the **Access Key ID** and the **Secret Access Key**, which are necessary for Tanium Connect.

Tanium Saved Question

Create a Saved Question named `TaniumInventoryAWSSDL` using this Query:

```
Get Computer ID and Computer Name and AD Query - Primary User and AD Domain and "Computer Serial Number" and CPU and Chassis Type and Short Hostname and Manufacturer from all machines
```

1. Sign in to the Tanium console.
2. From the Main menu, go to **Modules > Interact** to open the Interact Overview page.
3. In the **Explore Data** field, enter the provided query:

   ```
   Get Computer ID and Computer Name and AD Query - Primary User and AD Domain and "Computer Serial Number" and CPU and Chassis Type and Short Hostname and Manufacturer from all machines
   ```
4. Press **Enter**. The **Question Results** page shows the results.
5. Click **Save** above the question field and configure the following:
   a. Name: `TaniumInventoryAWSSDL`
6. Expand the **Preview** section to preview the results of the saved questioned, and then click **Save**.

For details on saved questions, see Managing saved questions.

Tanium Connect

Create a connection named TaniumInventoryAWSSDL using a saved question as the source.

1. Sign in to the Tanium console.
2. On the Connect **Overview** page, scroll to the **Connections** section and click **Create Connection**.
3. Enter `TaniumInventoryAWSSDL` as the name for the connection.
4. Configure the connection source:
   - Source “Saved Question”
   - Saved Question Name TaniumInventoryAWSSDL
5. Select AWS S3 for the destination.
   - Name TaniumInventoryAWSSDL
6. Specify authentication credentials:
   - AWS Access Key (from user IncomingReportUser)
   - AWS Secret Access Key (from user IncomingReportUser)
   - S3 Bucket name: taniuminventorystack-incomingreportbucket0582d8a9-148c7ofb31234 (from IncomingReportBucket)
   - File Name: tanium-inventory.json
7. In the **Configure Output > Format** section, select **JSON** and select the **Generate Document** checkbox .
8. Use the Schedule section to update the schedule:
   - Whatever makes sense to you and select **Enable Schedule**.

Note: If you do not enable the schedule, the connection only runs when you manually run it.

9. After you enter the details for the connection, click **Save and Run.**

For details on connections, see Managing connections.

Result

When the AWS S3 TaniumInventoryAWSSDL connection executes the following happens:

- The Lambda will write a new file named “connect.json.parquet” into the S3 bucket.
- The Parquet format file should have the OCSF Inventory content.

Troubleshooting

There are two main pieces of the Tanium Connect and Amazon Security Lake integration; documentation and logs. Each provides the ability to gather additional data to help troubleshoot any issues you may run into.

For more information on Connect troubleshooting, see the Troubleshooting section of the Connect User Guide.

Review the logs from the provided Lambda created earlier in the document.

When you've confirmed that these two pieces are working appropriately, you may proceed to troubleshoot Amazon Security Lake itself.

Amazon Security Lake Product Page

Amazon Security Lake User Guide

Further Support

If you need further assistance, please reach out to your Tanium Technical Account Manager (TAM).

Appendix: OCSF Mapping

TANIUM QUESTION	OCSF DEVICE INVENTORY
Computer ID	device: uid
Computer Name	device: hostname
AD Query – Primary User	user: name
AD Domain	user: domain
Computer Serial Number	hw_info: serial_number
CPU	hw_info: cpu_type
Chassis Type	hw_info: chassis AND device: type
Short Hostname	device: name
Manufacturer	hw_info: bios_manufacturer

Table 1: OCSF Mapping