ThreatConnect™

Micro Focus® ArcSight Integration Package

User Guide

Software Version 1.0

July 27, 2020

30034-03 EN Rev. A

ThreatConnect, Inc.
3865 Wilson Blvd., Suite 550, Arlington, VA 22203

P: 1.800.965.2708 | F: .703.229.4489
www.ThreatConnect.com

Table of Contents

OVERVIEW

The ThreatConnect® integration package for Micro Focus ArcSight Enterprise Security Management (ESM) allows ArcSight ESM users to interact with threat intelligence in ThreatConnect directly from the ArcSight Console. The integration allows users to look up Indicators, create Indicators, report false positives, and report observations.

NOTE: Users running a Dedicated Instance of ThreatConnect should have the ThreatConnect Environment Server installed in order to use the ArcSight Integration Package. The Environment Server allows an organization to utilize a user interface (UI) to execute the jobs using ThreatConnect integration applications available from TC Exchange™. For example, if data need to be pushed to a device such as a SIEM-, firewall-, or host-based system, the Environment Server runs as an intermediary between the external ThreatConnect instance and the user's internal network.

DEPENDENCIES

ThreatConnect Dependencies

- Active ThreatConnect Application Programming Interface (API) key
- Python 2.7.11, including the following Python modules:
  - **enum34**
  - **python-dateutil**
  - **requests**
  - **threatconnect**

ArcSight Dependencies

- ArcSight ESM
- ArcSight Console
- ArcSight SmartConnector Installer

APPLICATION SETUP

ThreatConnect Administrative Configuration

There are no changes required by the Administrator for this integration.

ThreatConnect Organization Configuration

There are no changes required by the Organization Administrator for this integration.

ArcSight Configuration

There are two possible configuration options to use the integration from within the ArcSight console. The first option is to install an Action Connector (an instance of an ArcSight SmartConnector) on an ArcSight server, which allows the Integration Commands to run remotely. The second option is to install the integration scripts on the workstation that runs the ArcSight Console.

Action Connector: Overview

The Action Connector option requires that an ArcSight SmartConnector be installed in the ArcSight infrastructure. Once the Connector is installed, it will be configured as an Action Connector to run the Integration Commands remotely via a connector from the ArcSight Console. When using the Integration Commands, the ArcSight Console will send the request to the Action Connector, which will execute the Integration Commands and return the results to the ArcSight Console.

The ArcSight administrator is responsible for configuring each action command, the Integration Commands in the ArcSight Console, and their supported parameters. Any changes to the configuration of the commands or parameters will require a restart of the Action Connector and may require an update of the Integration Commands configuration.

Action Connector: Installation

NOTE: See the ArcSight Action Connector documentation provided by Micro Focus for all available options and for more details about Action Connectors.

1. Create a directory called **threatconnect** in the ArcSight connector directory (e.g., **<ArcSight Home>/connectors**).
2. Follow the standard procedures for installing the ArcSight SmartConnector on the platform for your environment (e.g., on 64-bit Linux servers, use **ArcSight-7.x.x.xxx.x-Connector-Linux64.bin**). When prompted for the path during

installation, the fully qualified path of the **threatconnect** directory created in Step 1 should be entered.

3. Extract the ThreatConnect ArcSight Integration package in the **<ArcSight Home>/connectors/threatconnect/current** directory.
4. Install Python dependencies **pip install threatconnect**.
5. In the extracted directory, copy the **tc.conf-template** file to **tc.conf**. Edit the **tc.conf** configuration file, and supply the appropriate values for ThreatConnect API connectivity.
6. Copy the **threatconnect.counteract.properties-template** configuration file to **<ArcSight Home>/connectors/threatconnect/current/user/agent/flexagent/threatconnect.counteract.properties**. If any modifications to the command are required, they can be made now by editing the properties. Updating the commands in the properties files can be done later if required. See the "Action Connector: Command Configuration" section for more details.
7. Change directory to the **<ArcSight Home> /connectors/threatconnect/current/bin** directory. Run the agent setup script (**runagentsetup.sh**, or **runagentsetup.bat** on a Windows system).
    a. Select the **Add a Connector** option.
    b. Select **ArcSight FlexConnector CounterAct** as the Type.
    c. Enter **threatconnect** as the configuration file. (Ensure that **.counteract.properties** is *excluded* when entering the configuration file name.)
    d. Select **ArcSight Manager (encrypted)** as the destination.
    e. Complete the setup.

Once the setup is completed, the new Action Connector should be available in the ArcSight Console under the **Navigator** panel by going to **Connectors > Shared > All Connector > Site Connectors** (Figure 1).

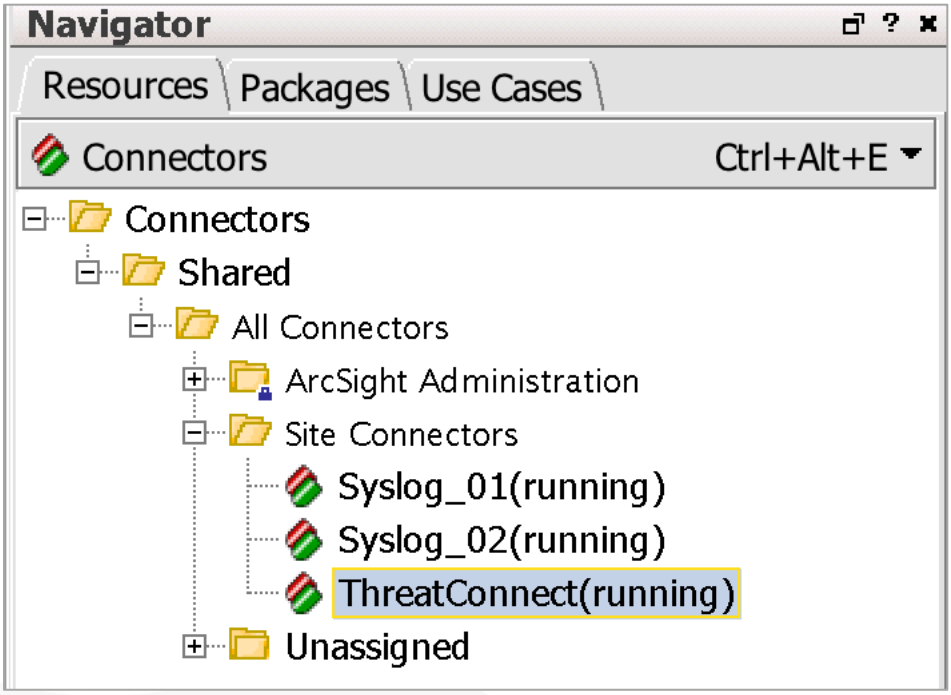

Figure 1

Action Connector: Command Configuration

The provided configuration file (**threatconnect.counteract.properties**) is configured with all the commands supported by the integration using default parameters. Each configured parameter is required when using these commands. In some instances, it may be preferable to alter the parameters or create new commands. For instance, the **tc-add-indicator** command has Tags and Attributes required by default. It might be decided that Tags are not required or that multiple Tags are required. Administrators with write access to the configuration file can update the commands to suit the needs of the organization.

When updating the **threatconnect.counteract.properties** file, always ensure that the **command.count** value is updated to the number of commands in the configuration file and that, if parameters are added or removed, the **parameter.count** value is updated to the correct number of parameters for the command being modified.

For example, if an organization needs to have **quick add indicator**, **add indicator**, and **add indicator enhance** commands, the **threatconnect.counteract.properties** file would require the following changes:

1. Remove the existing **tc-add-indicator** command.
2. Create the following new command configuration for **tc-quick-add-indicator**:

```
command[3].name=tc-quick-add-indicator
command[3].displayname=ThreatConnect_Quick_Add_Indicator
command[3].parameter.count=4
command[3].parameter[0].name=indicator
command[3].parameter[0].displayname=indicator
command[3].parameter[1].name=api_access_id
command[3].parameter[1].displayname=API Access Id
command[3].parameter[2].name=api_secret_key
command[3].parameter[2].displayname=API Secret Key
command[3].parameter[3].name=api_base_url
command[3].parameter[3].displayname=API Base URL
command[3].action=python ${_ARCSIGHT_HOME}/threatconnect-
arcsight/add_indicator.py --indicator ${indicator} --owner MyOrg --
rating 3 --confidence 75 --tag ArcSight --api_access_id
${api_access_id} --api_secret_key ${api_secret_key} --api_base_url
${api_base_url}
```

In this command configuration, the owner, Threat Rating (**rating**), Confidence Rating (**confidence**), and a Tag have all been predefined. The user will not be required to add these parameters in the console when using the Integration Command.

3. Create the following new command configuration for **tc-add-indicator**:

```
command[4].name=tc-add-indicator
command[4].displayname=ThreatConnect_Add_Indicator
command[4].parameter.count=7
command[4].parameter[0].name=indicator
command[4].parameter[0].displayname=indicator
command[4].parameter[1].name=owner
command[4].parameter[1].displayname=Owner
command[4].parameter[2].name=rating
command[4].parameter[2].displayname=Rating
command[4].parameter[3].name=confidence
command[4].parameter[3].displayname=Confidence
command[4].parameter[4].name=api_access_id
command[4].parameter[4].displayname=API Access Id
command[4].parameter[5].name=api_secret_key
command[4].parameter[5].displayname=API Secret Key
command[4].parameter[6].name=api_base_url
command[4].parameter[6].displayname=API Base URL
command[4].action=python ${_ARCSIGHT_HOME}/threatconnect-
arcsight/add_indicator.py --indicator ${indicator} --owner ${owner} --
rating ${rating} --confidence ${confidence} --api_access_id
${api_access_id} --api_secret_key ${api_secret_key} --api_base_url
${api_base_url}
```

In this command configuration, the user will be prompted for the owner, Threat Rating, and Confidence Rating. There will be no Tag or Attribute added with this Indicator.

4. Create the following new command configuration for **tc-add-indicator-enhanced**:

```
command[5].name=tc-add-indicator-enhanced
command[5].displayname=ThreatConnect_Add_Indicator_Enhanced
command[5].parameter.count=9
command[5].parameter[0].name=indicator
command[5].parameter[0].displayname=indicator
command[5].parameter[1].name=owner
command[5].parameter[1].displayname=Owner
command[5].parameter[2].name=rating
command[5].parameter[2].displayname=Rating
command[5].parameter[3].name=confidence
command[5].parameter[3].displayname=Confidence
command[5].parameter[4].name=attribute
command[5].parameter[4].displayname=Attribute
command[5].parameter[5].name=tag
command[5].parameter[5].displayname=Tag
command[5].parameter[6].name=api_access_id
```

```
command[5].parameter[6].displayname=API Access Id
command[5].parameter[7].name=api_secret_key
command[5].parameter[7].displayname=API Secret Key
command[5].parameter[8].name=api_base_url
command[5].parameter[8].displayname=API Base URL
command[5].action=python ${_ARCSIGHT_HOME}/threatconnect-
arcsight/add_indicator.py --indicator ${indicator} --owner ${owner} --
rating ${rating} --confidence ${confidence} --attribute ${attribute} --
tag ${tag} -tag ArcSight --api_access_id ${api_access_id} --
api_secret_key ${api_secret_key} --api_base_url ${api_base_url}
```

In this command configuration, the user will be prompted to add the owner, Threat Rating, Confidence Rating, Attribute, and Tag. An additional Tag of "ArcSight" has already been added to the configuration. Any number of Tags and Attributes can be added as parameters to the command.

5. Increment the **command.count** value in the configuration to account for all commands added to the configuration.
6. Restart the Action Connector.

Action Connector: Integration Commands

To configure the corresponding Integration Commands, log into the ArcSight Console with a user in the Administrator Groups.

1. In the **Navigator** panel, switch to the **Integration Commands** section.
2. From the **Commands** tab, right-click on the **Public** folder under **Integration Commands > Shared > All Integration Commands** and select the **New Command** option (Figure 2). A new tab will open in the **Inspect/Edit** panel.

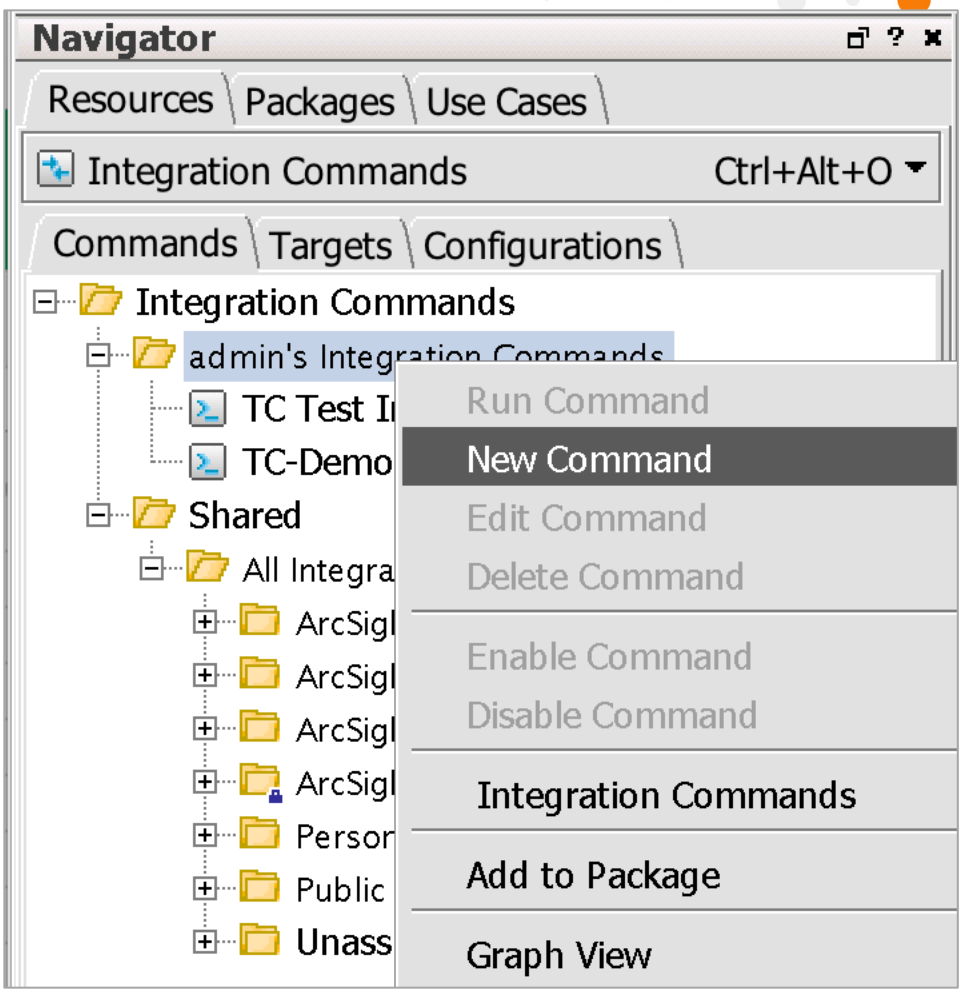

Figure 2

3. In the **Command Editor** tab of the **Inspect/Edit** panel, select **Connector** as the Integration Target type (Figure 3).

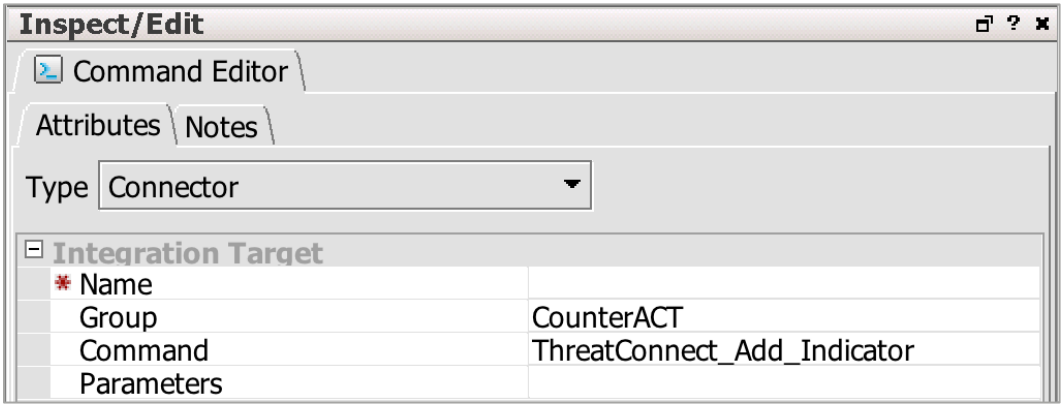

Figure 3

Note: All required parameters have a red asterisk to the left of the parameter name.

4. Enter **TC – Add Indicator** for the name.

5. Select **CounterAct** as the Group, and select **ThreatConnect_Add_Indicator** as the command.
6. Click in the **Parameters** value area. An ellipsis icon will appear (Figure 4).

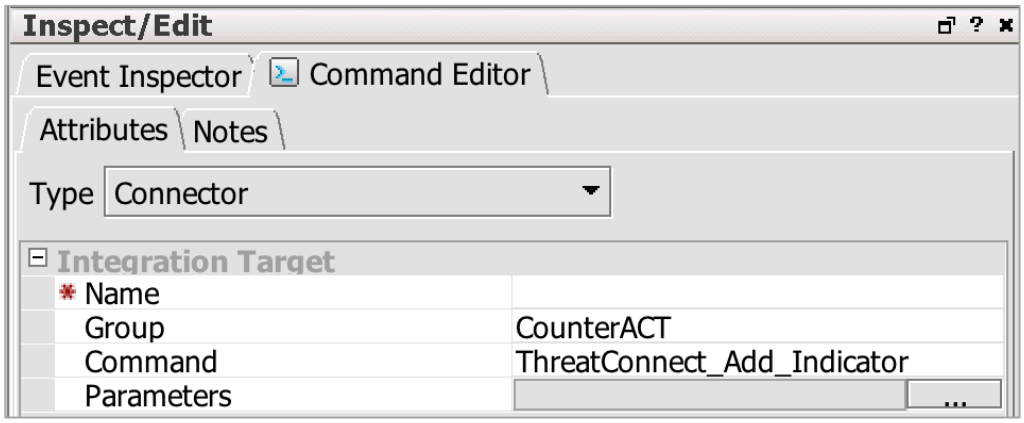

Figure 4

7. Click on the ellipsis, and a **Parameters** dialog box will be displayed (Figure 5).

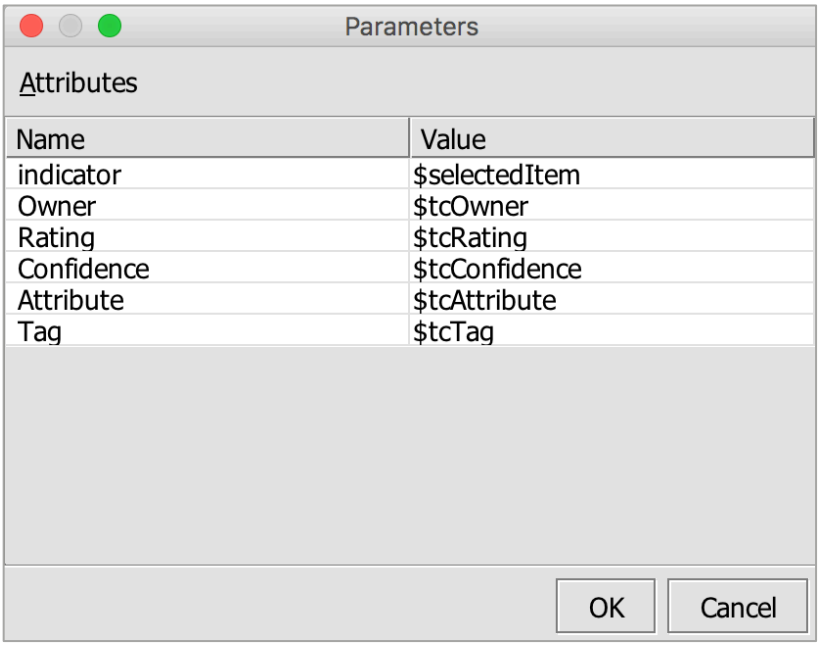

Figure 5

8. In the Parameters dialog box, add $selectedItem as the Indicator. This designation allows the Indicator to be auto-populated by clicking on a cell. For the remaining fields, a variable can be defined or a value can be entered. For all parameters populated with a variable (e.g., $tcOwner), the user will be prompted to enter values when selecting the integration commands. (See the "Running Integration Commands" section.)

Action Connector: Integration Commands Configuration

After the Integration Commands setup has been completed, a new configuration needs to be added for the command to be displayed in the context menu.

1. In the **Navigator** panel, switch to the **Integration Commands** section and then to the **Configurations** tab.
2. Right-click on the **Public** folder under **Integration Configurations > Shared > All Integration Configurations**, and select the **New Configuration** option (Figure 6).

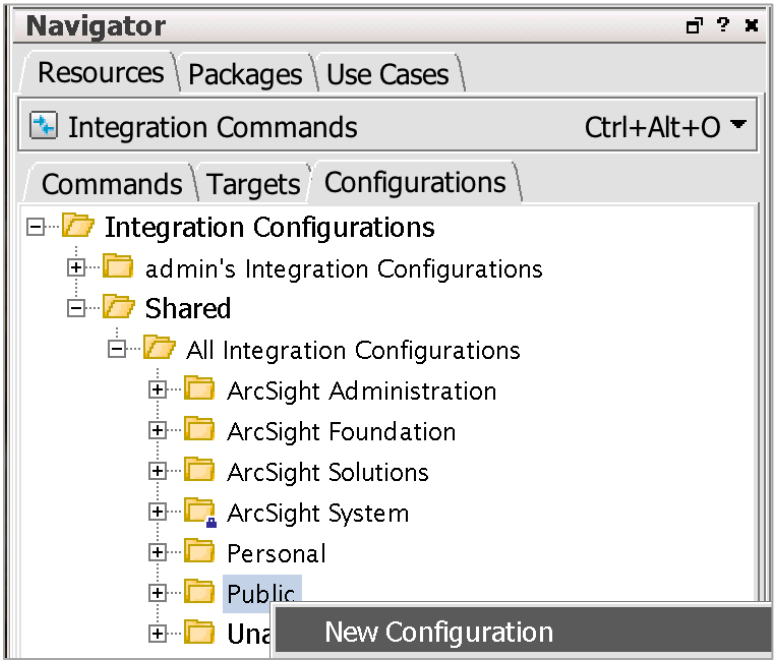

Figure 6

3. In the **Configuration Editor** tab that opens in the **Inspect/Edit** panel, select **Connector** as the Integration Configuration type (Figure 7).

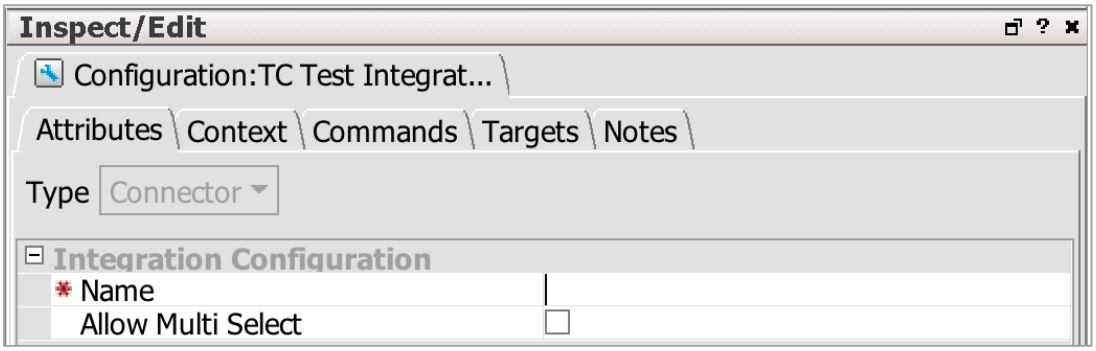

Figure 7

Note: All required parameters have a red asterisk to the left of the parameter name.

4. Enter TC – Add Indicator Config for the name.
5. Switch over to the Context tab, and click the Add button. A new entry will be added to the list (Figure 8).

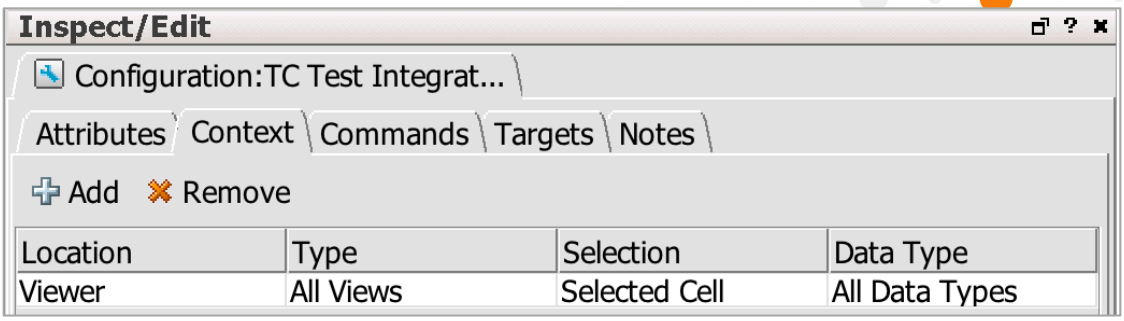

Figure 8

6. Click on the fields and populate them with the following values:
   - Location: **Viewer**
   - Type: **All Views**
   - Selection: **Selected Cell**
   - Data Type: **All Data Types**

   These values can be changed to accommodate an organization's requirements, with the exception of the **Selection** field, which is required to have a value of **Selected Cell** for the integration command to function properly.
7. Switch to the **Commands** tab, and click the **Add** button. In the new dialog window, select the command created in the "Action Connector: Integration Commands" section (**TC – Add Indicator**) from the file tree (Figure 9).

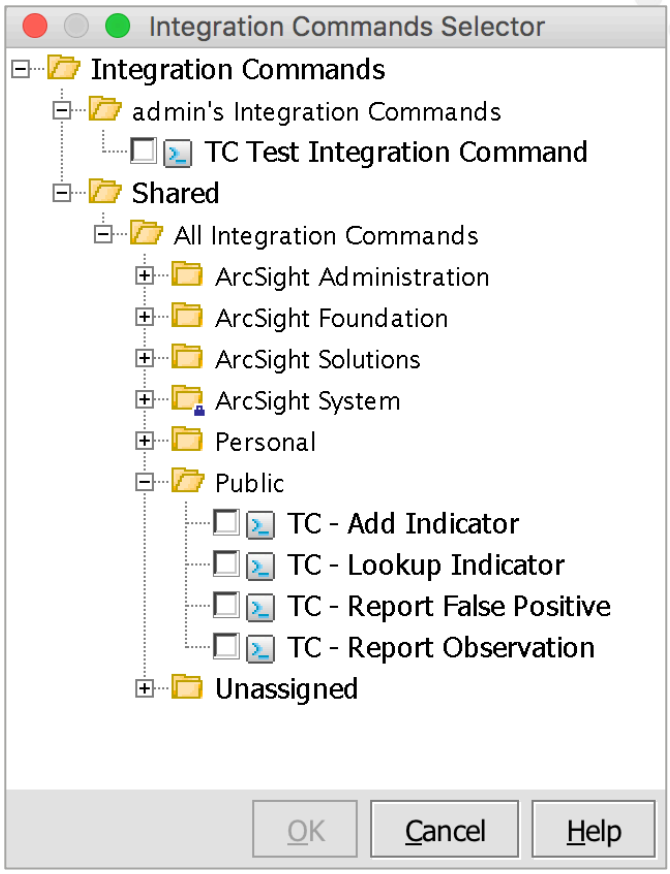

Figure 9

8. Switch to the **Targets** tab under the **Configuration Editor** tab in the **Inspect/Edit** panel for **TC – Add Indicator**, and click the **Add** button. In the **Connectors Selector** window, browse to **Connector > Shared > All Connectors > Site Connectors** and check the box next to **ThreatConnect** (Figure 10).

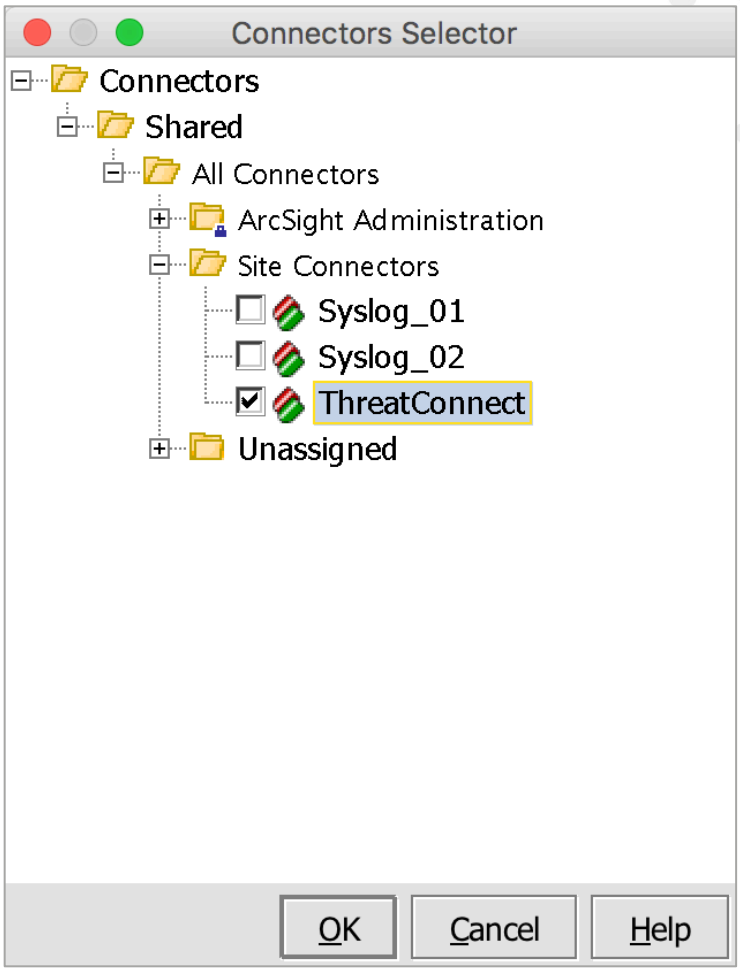

Figure 10

Local Workstation: Overview

This method requires Python and dependencies to be installed on the workstation running the ArcSight Console. The scripts will be run from the local workstation and require network connectivity to the ThreatConnect API.

Local Workstation: Integration Commands

Any user can use Integration Commands via the **Script** Type.

1. In the **Navigator** panel, switch to the **Integration Commands** section.
2. From the **Commands** tab, right-click on the personal folder under **Integration Commands** and select the **New Command** option (Figure 11).

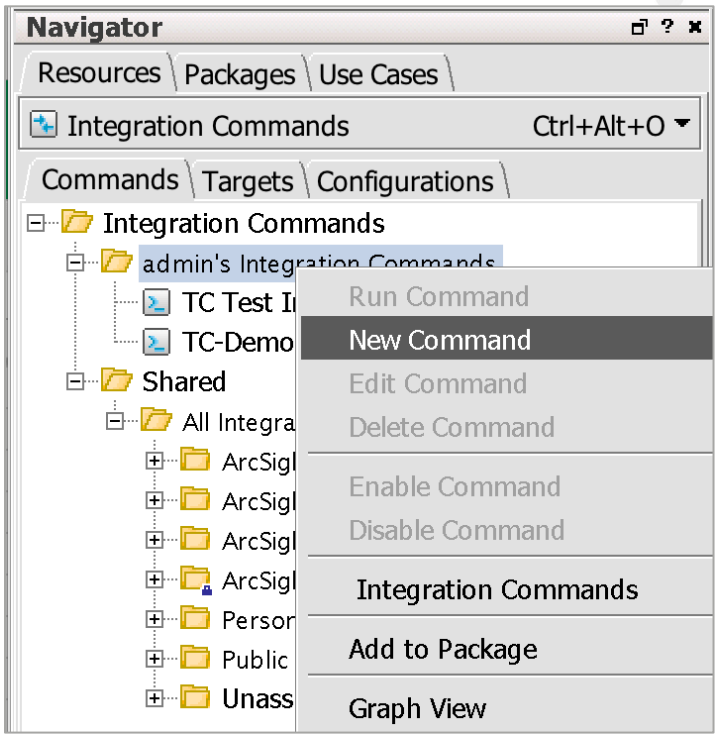

Figure 11

3. In the **Command Editor** tab that opens in the **Inspect/Edit** panel, select **Script** as the Integration Target type (Figure 12).

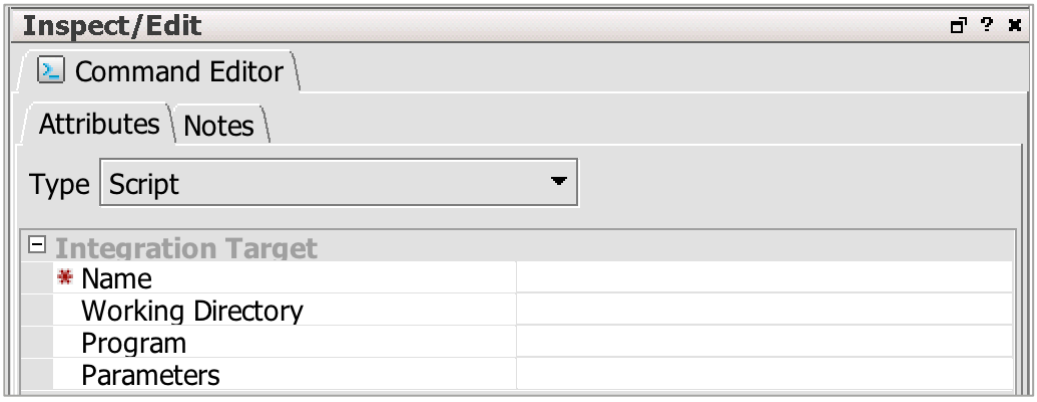

Figure 12

Note: All required parameters have a red asterisk to the left of the parameter name.

4. Enter **TC – Lookup Indicator Local** for the name.
5. Click in the **Working Directory** value area, and an ellipsis icon will appear (Figure 13).

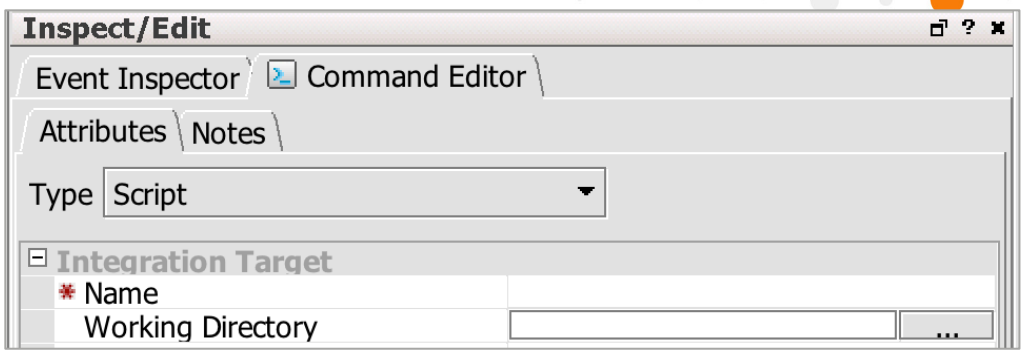

Figure 13

6. Click on the ellipsis, and a **Select** dialog box will be displayed (Figure 14). Browse to the directory where the ThreatConnect ArcSight Package was extracted and click the **Select** button.

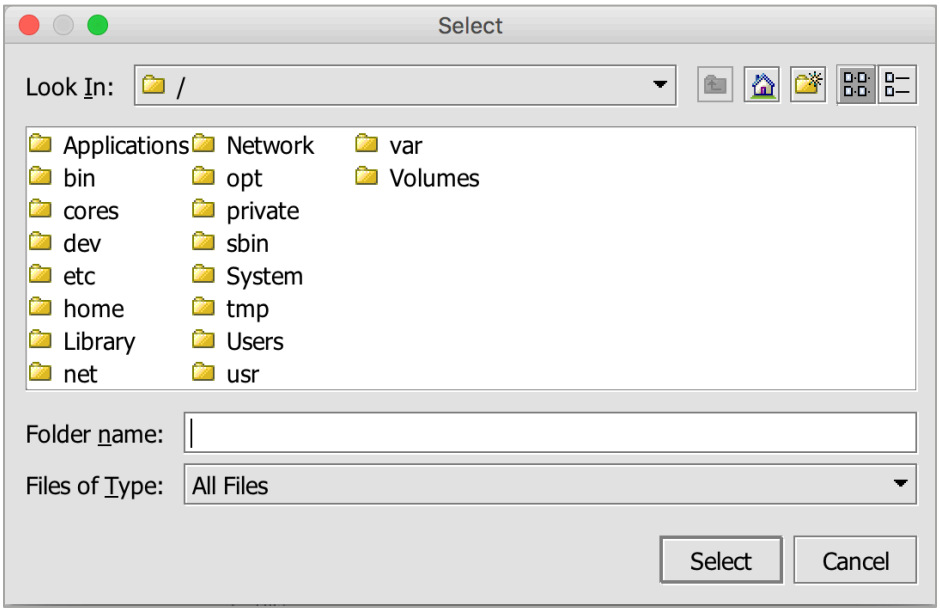

Figure 14

7. In the **Program** field, add **python** (Figure 15).

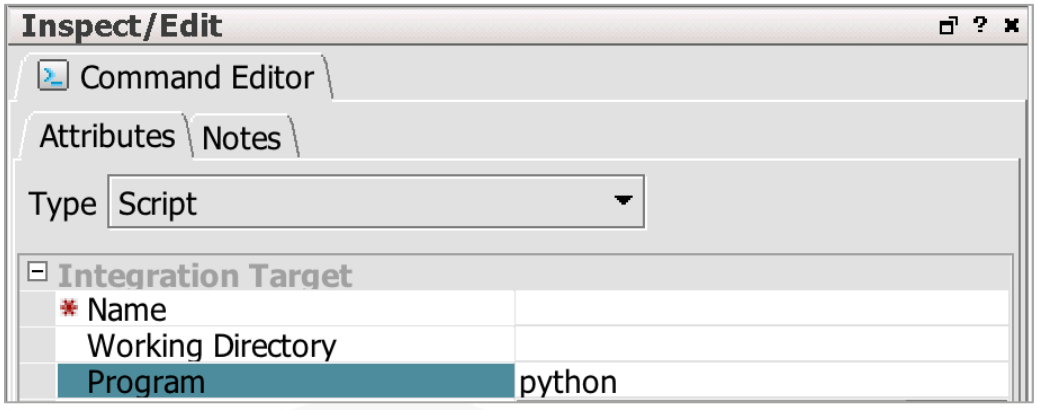

Figure 15

8. Click in the **Parameters** value area, and an ellipsis icon will appear (Figure 16).

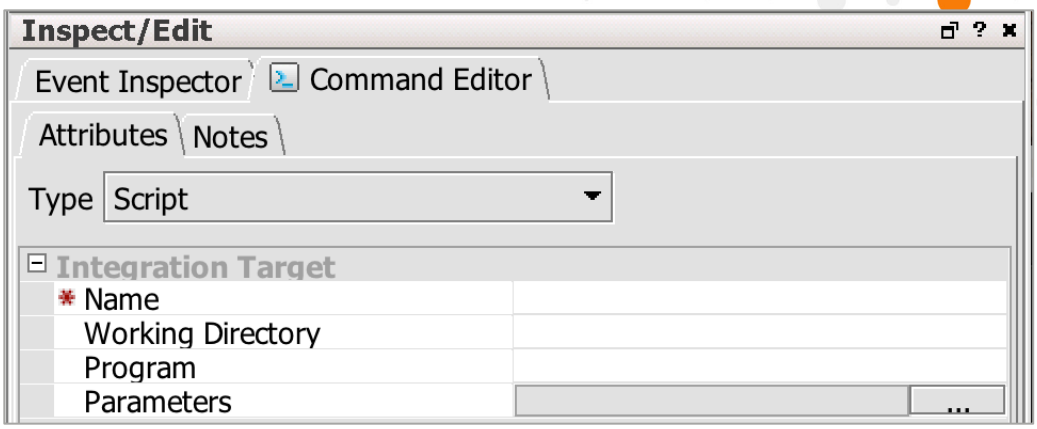

Figure 16

9. Click on the ellipsis, and a **Parameters** dialog box will be displayed (Figure 17).

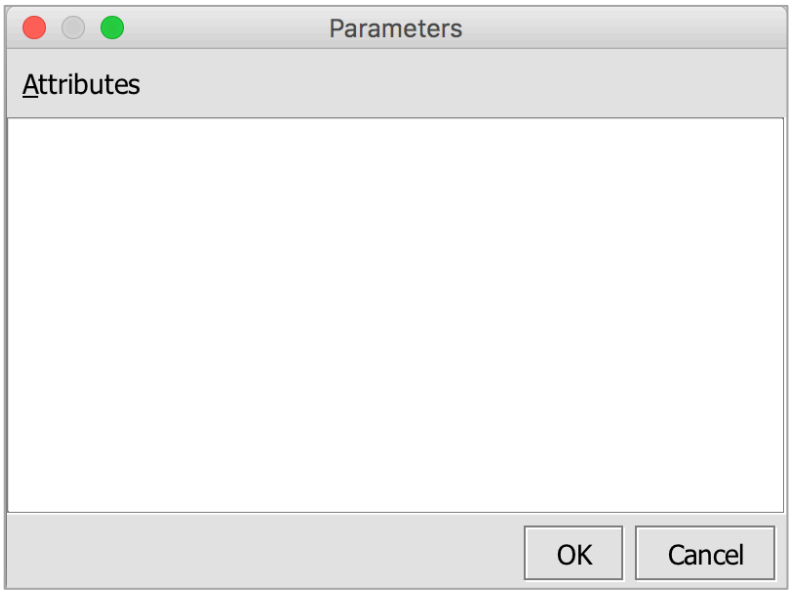

Figure 17

10. In the **Parameters** dialog box, add the script name with the **--indicator** parameter and a value of **$selectedItem** (Figure 18). This designation allows the Indicator to be auto-populated by clicking on a cell.

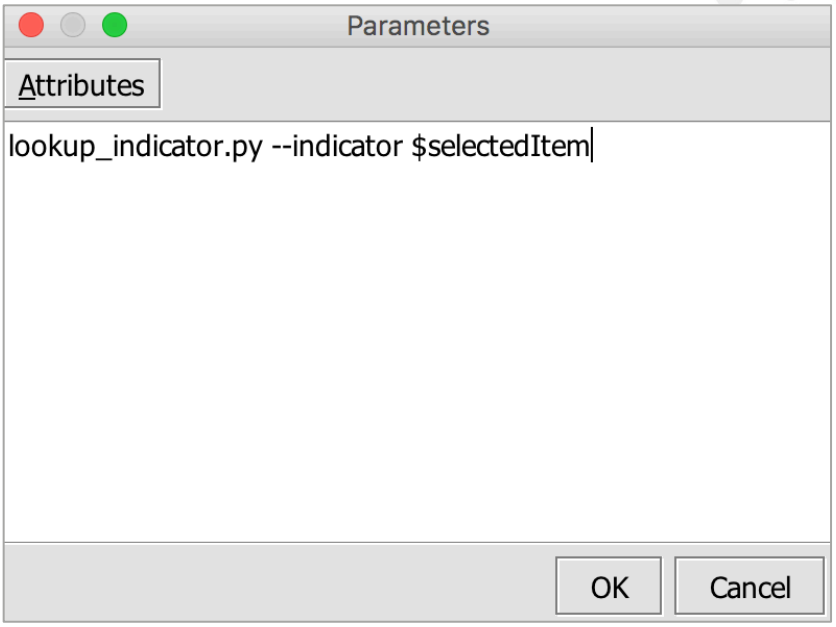

Figure 18

Local Workstation: Integration Commands Configuration

After the Integration Command setup for Script Type has been completed, a new configuration needs to be added for the command to be displayed in the context menu.

1. In the **Navigator** panel, switch to the **Integration Commands** section and then to the **Configurations** tab.
2. From the **Commands** tab, right-click on the personal folder under **Integration Configuration** and select the **New Configuration** option (Figure 19).

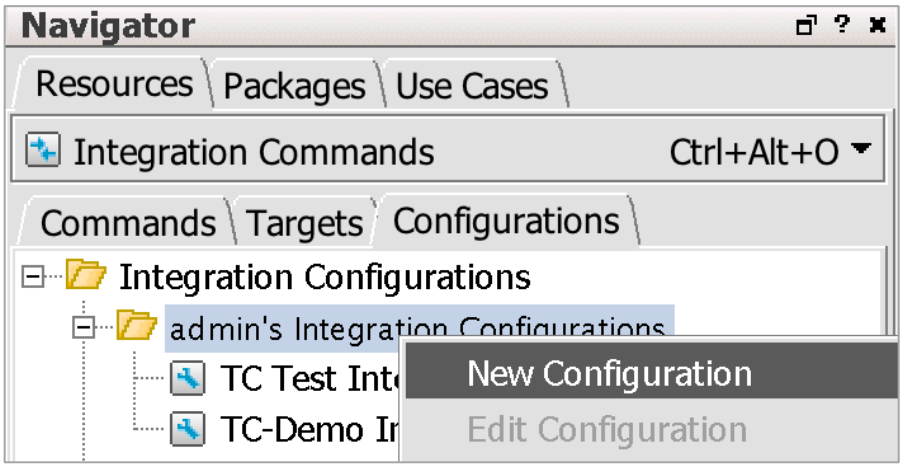

Figure 19

3. In the **Configuration Editor** tab that opens in the **Inspect/Edit** panel, select **Script** as the Integration Target type (Figure 20).

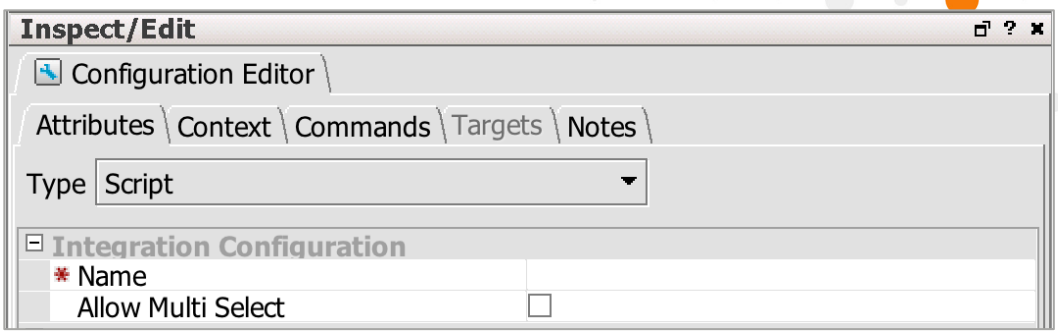

Figure 20

Note: All required parameters have a red asterisk to the left of the parameter name.

4. Enter **TC – Add Lookup Local Config** for the name.
5. Switch to the **Context** tab, and click the **Add** button. A new entry will be added to the list (Figure 8). Click on the fields and populate them with the following values:
   - Location: **View**
   - Type: **All View**
   - Selection: **Selected Cell**
   - Data Type: **All Data Types**

   These values can be changed to accommodate an organization's requirements, with the exception of the **Selection** field, which is required to have a value of **Selected Cell** for the Integration Command to function properly.
6. Switch to the **Commands** tab, and click the **Add** button. In the new dialog window, select the command created in the "Local Workstation: Integration Commands" section (**TC – Lookup Indicator Local**) from the file tree (Figure 21).

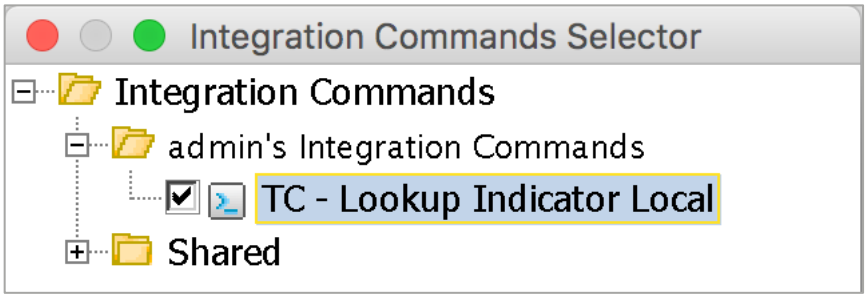

Figure 21

TC – Add Indicator

The availability of the **TC – Add Indicator** Integration Command depends on the context specified in the Command Configuration. For more information on the context setting, see the "Integration Commands Configuration" sections of this document. To access the Integration Commands, right-click on the Indicator that should be added and select **Integration Commands > TC – Add Indicator** (Figure 22).

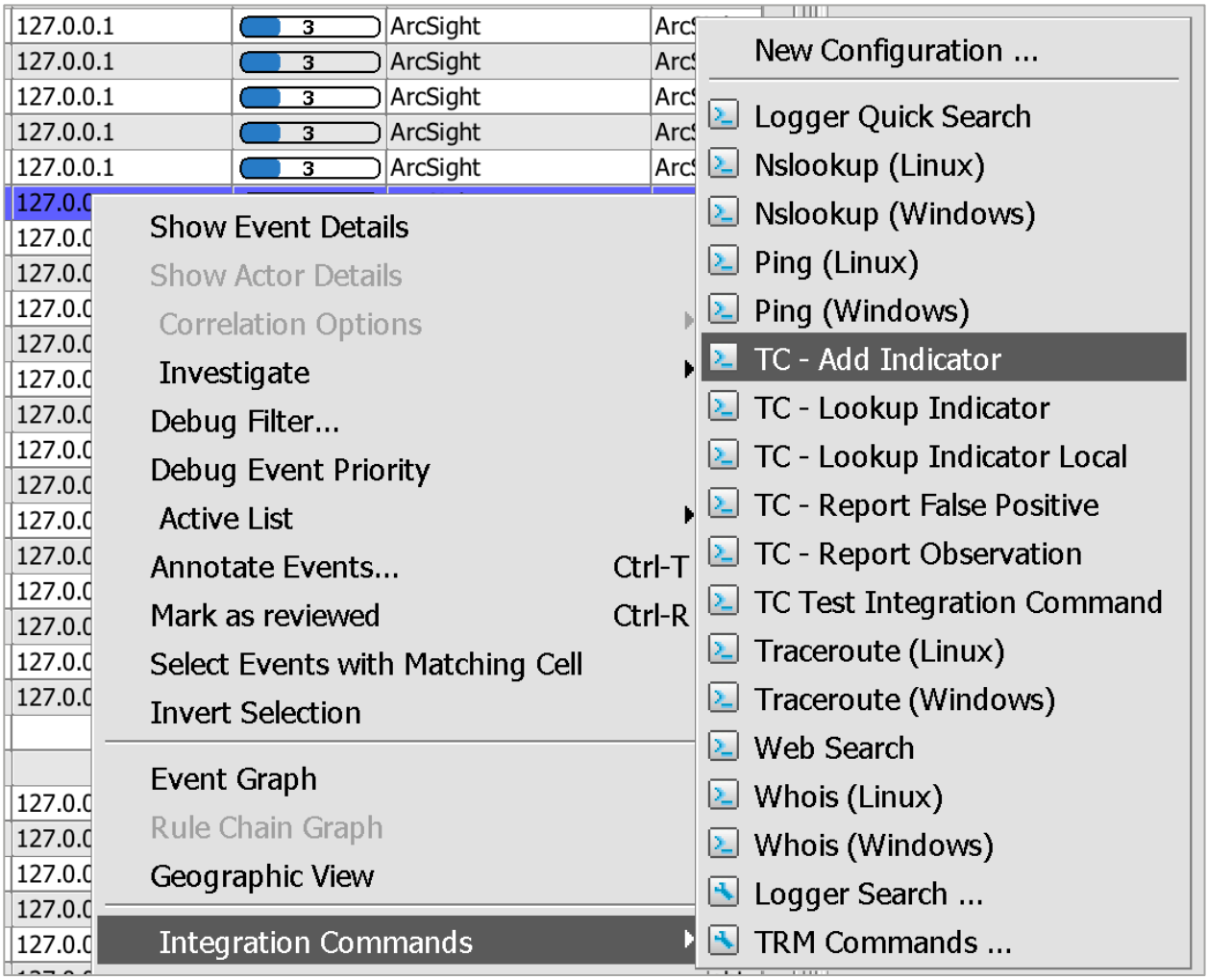

Figure 22

TC – Lookup Indicator

The availability of the **TC – Lookup Indicator** Integration Command depends on the context specified in the Command Configuration. For more information on the context setting, see the "Integration Commands Configuration" sections of this document. To access the Integration Commands, right-click on the Indicator for which the lookup should be performed and select **Integration Commands > TC – Lookup Indicator** (Figure 23).

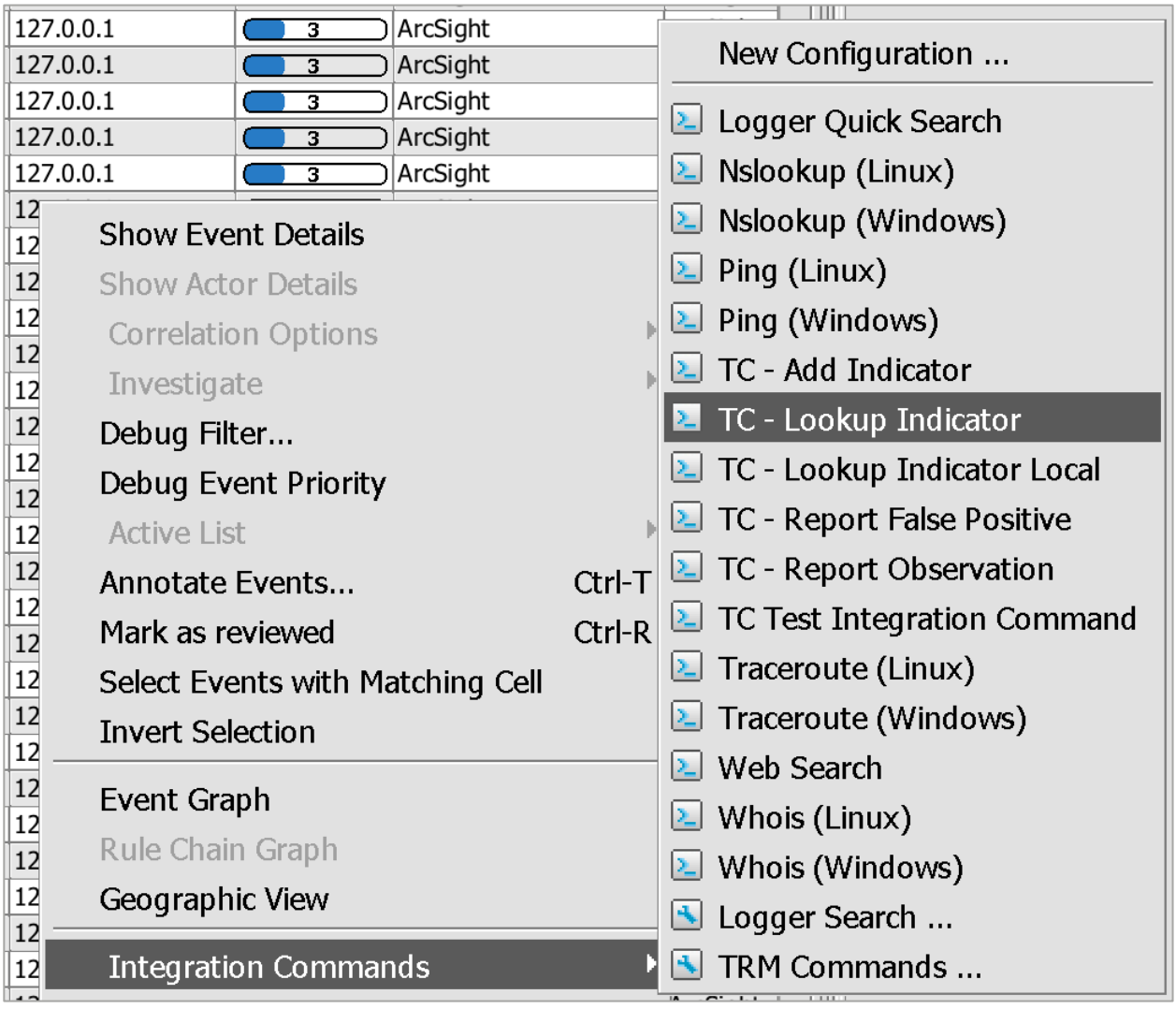

Figure 23

TC – Report False Positive

The availability of the **TC – Report False Positive** Integration Command depends on the context specified in the Command Configuration. For more information on the context setting, see the "Integration Commands Configuration" sections of this document. To access the Integration Commands, right-click on the Indicator to report and select **Integration Commands > TC – Report False Positive** (Figure 24).

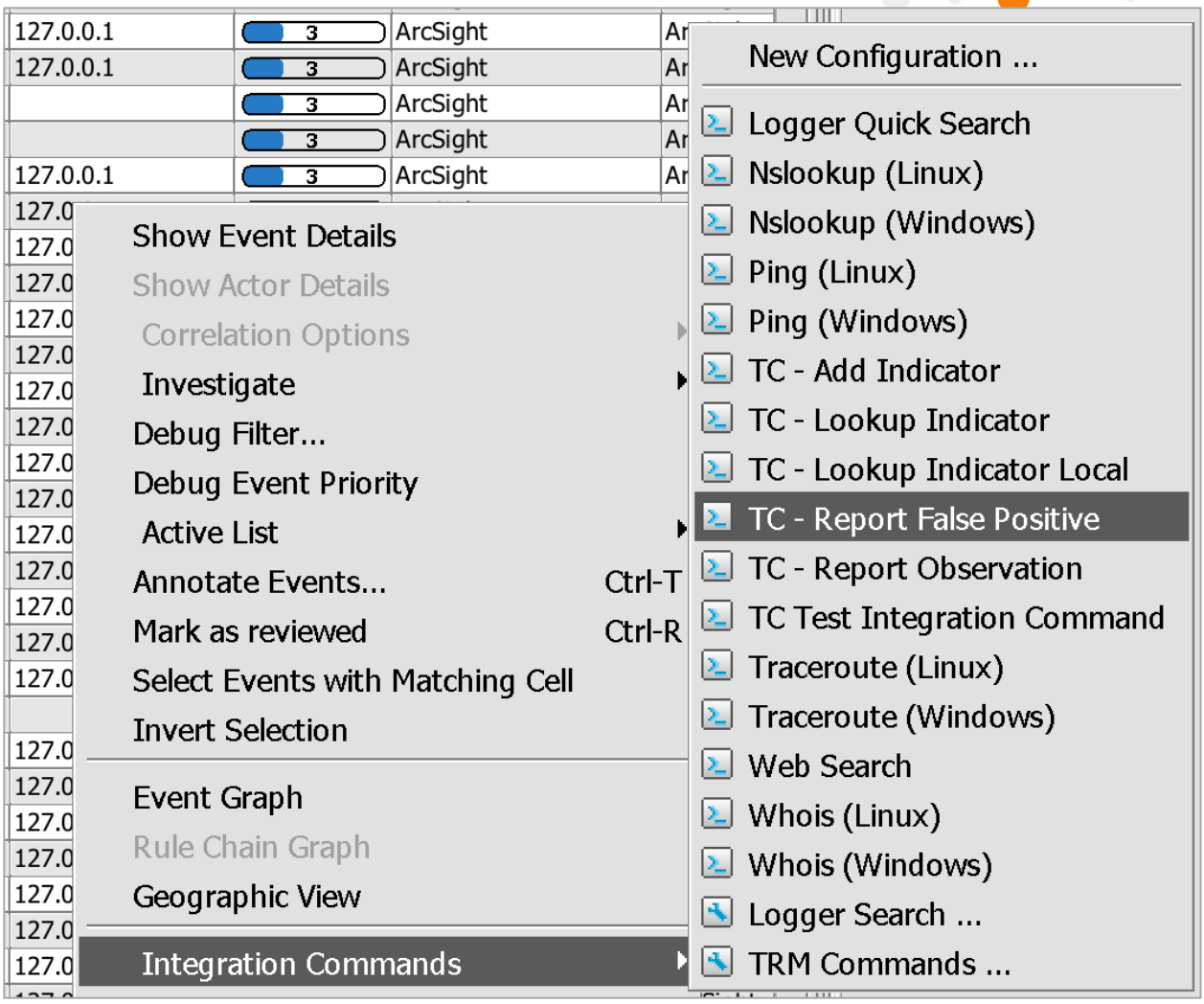

Figure 24

Rule Integration

Action Connector commands can be used in rules for automatically reporting observations or to automatically add an Indicator into ThreatConnect. These commands can be added to new rules or to existing rules already running in the current ArcSight environment.

1. In the **Navigator** panel, switch to the **Rules** section.
2. Right-click on a rule, and select the **Edit Rule** option (Figure 25).

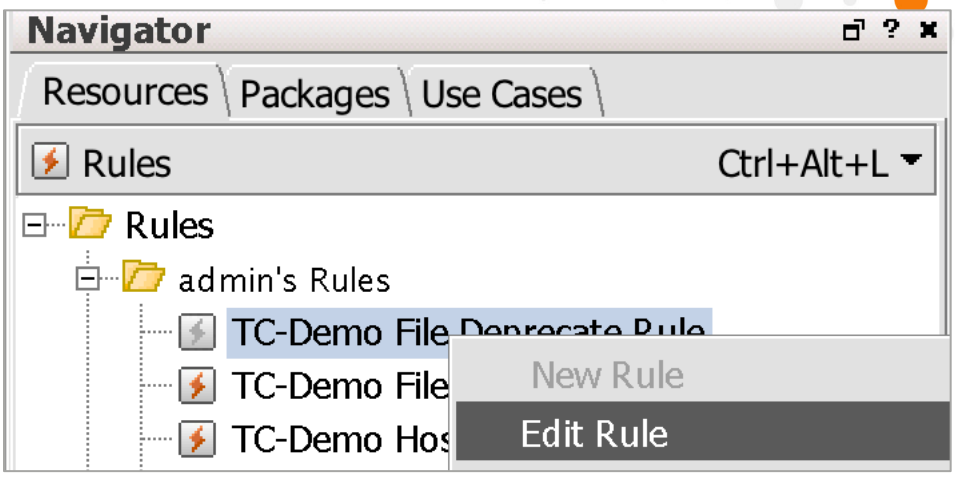

Figure 25

3. In the **Rule** tab that opens in the **Inspect/Edit** panel, select the **Actions** sub-tab. Right-click on the event to which the command should be added and select **Add > Execute Connector Command** (Figure 26). If this option is grayed out and you are not able to select it, you may need to convert your rule from a lightweight rule to a standard rule.

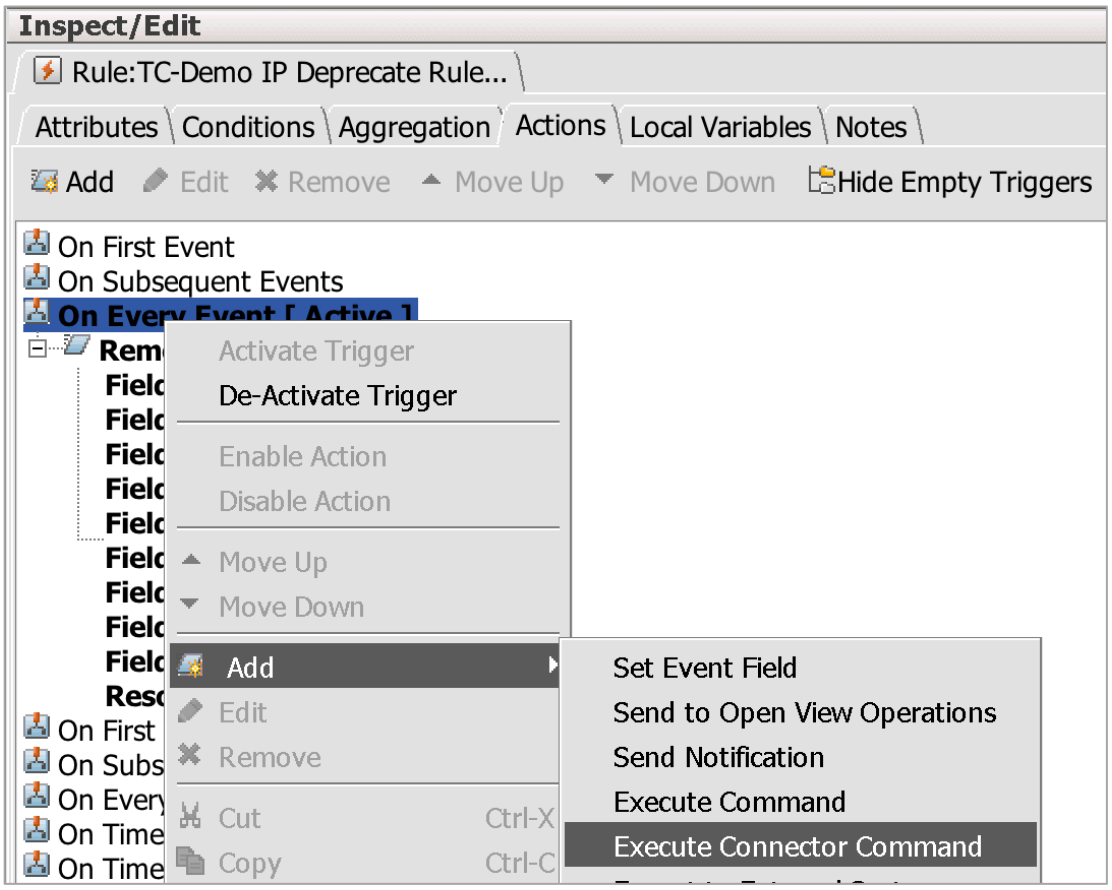

Figure 26

4. Select **ThreatConnect** as the Site Connector (Figure 27).

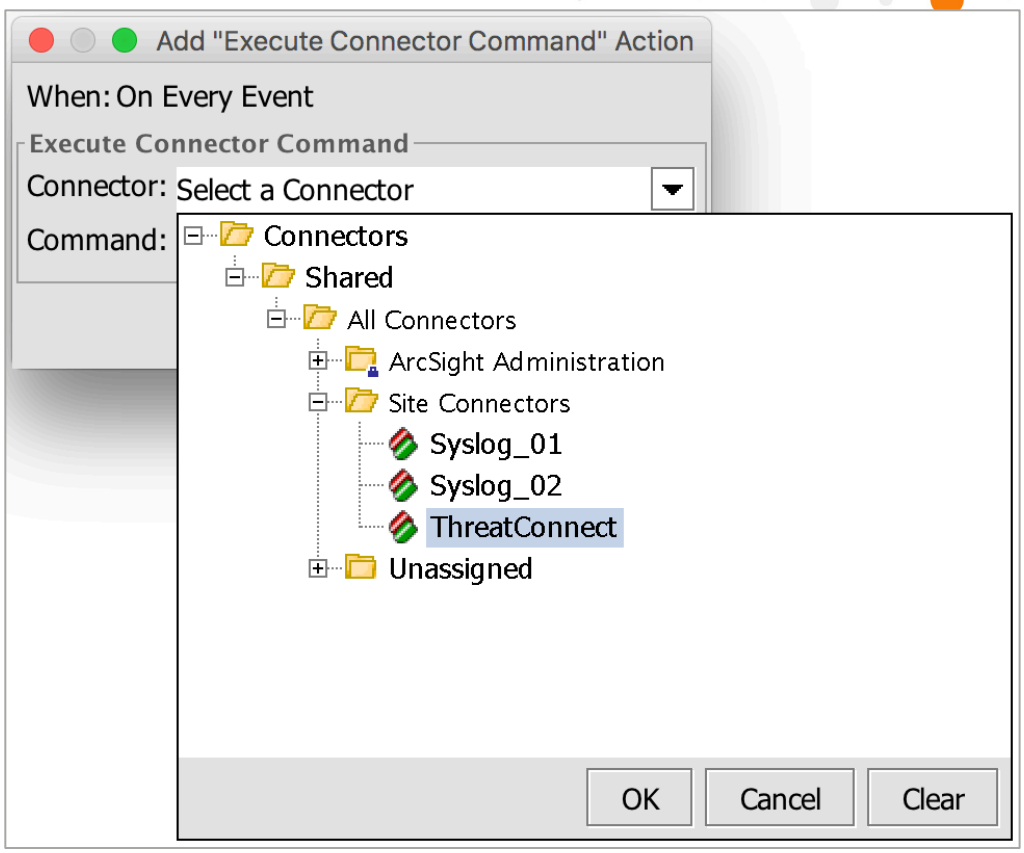

Figure 27

5. Select **counteract.Threatconnect_Report_Observation** as the command (Figure 28). Enter **1** for the **Count** unless using an aggregated event, in which case the aggregated-count variable should be used if available.

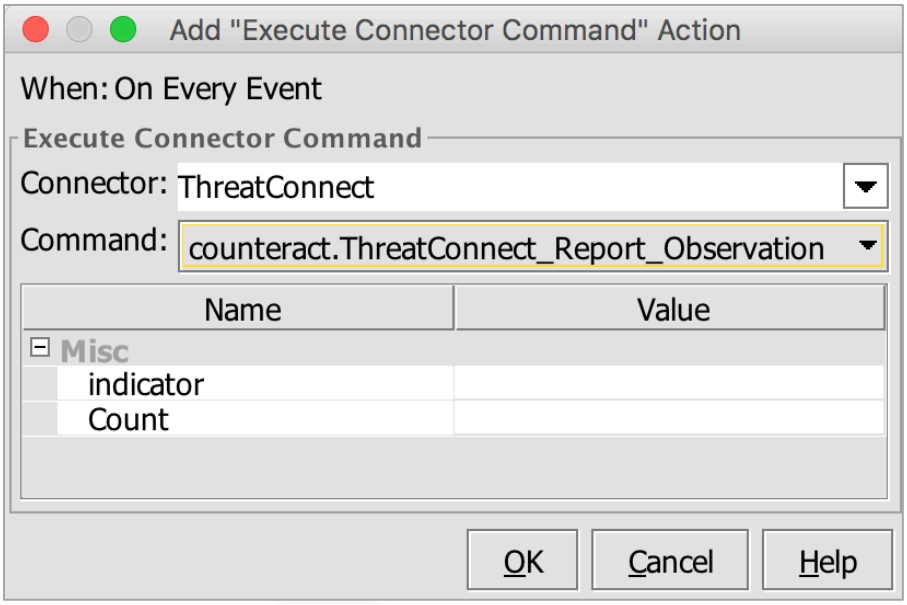

Figure 28

Commands Scripts

All of the Integration Command scripts require Command Line Interface (CLI) parameters. Some of the parameters, such as **api_access_id**, **api_secret_key**, and **api_base_url**, can be put into a **tc.conf** configuration file in the same directory as the scripts. This configuration allows the script to be called without having to provide those parameters on the CLI. All other CLI-required parameters must be passed to the script at each execution. A few of the scripts also support optional CLI parameters that can be passed while calling the scripts.

Add Indicator

```
usage: add_indicator.py [-h] --owner OWNER --indicator INDICATOR
                        --api_access_id API_ACCESS_ID --api_secret_key
                        API_SECRET_KEY --api_base_url API_BASE_URL
                        [--rating RATING] [--confidence CONFIDENCE]
                        [--attribute ATTRIBUTE] [--tag TAG]
                        [--logging LOGGING] [--activity_log]
```

The **add_indicator.py** command will add an Indicator to ThreatConnect. The command requires that an owner, Indicator, **api_access_id**, **api_secret_key**, and **api_base_url** be provided unless using the **tc.conf** configuration file. The command also supports providing optional Threat Rating and Confidence Rating values for the Indicator. Multiple Tags and Attributes can be provided as well. Attributes require an Attribute Type and value separated by a colon (e.g., **Description:Added via ArcSight ESM**).

Lookup Indicator

```
usage: lookup_indicator.py [-h] --indicator INDICATOR --api_access_id
                           API_ACCESS_ID --api_secret_key API_SECRET_KEY
                           --api_base_url API_BASE_URL [--logging LOGGING]
                           [--activity_log]
```

The **lookup_indicator.py** command will look up an Indicator in ThreatConnect. The command requires that an Indicator, **api_access_id**, **api_secret_key**, and **api_base_url** be provided unless using the **tc.conf** configuration file.

Report False Positive

```
usage: report_false_positive.py [-h] --indicator INDICATOR --api_access_id
                                API_ACCESS_ID --api_secret_key API_SECRET_KEY
                                --api_base_url API_BASE_URL
                                [--logging LOGGING] [--activity_log]
```

The **report_false_positive.py** command will add a false-positive count in ThreatConnect to the provided Indicator. The command requires that an Indicator, **api_access_id**, **api_secret_key**, and **api_base_url** be provided unless using the **tc.conf** configuration file.

Report Observation

```
usage: report_observation.py [-h] --indicator INDICATOR --api_access_id
                             API_ACCESS_ID --api_secret_key API_SECRET_KEY
                             --api_base_url API_BASE_URL [--count COUNT]
                             [--date_observed DATE_OBSERVED]
                             [--logging LOGGING] [--activity_log]
```

The **report_observation.py** command will add an observation count in ThreatConnect to the provided Indicator. The command requires that an Indicator, **api_access_id**, **api_secret_key**, and **api_base_url** be provided unless using the **tc.conf** configuration file. The command also supports providing optional count (default value is **1**) and date-observed (default value is the current date) values for the observation.

Default Configuration File

NOTE: See the HP ArcSight Action Connector documentation provided by HP for a description of each property.

The default configuration file provides four commands: **tc-lookup**, **tc-report-false-positive**, **tc-report-observation**, and **tc-add-indicator**. Administrators with write permission for this file can modify the default configuration and create new configurations.

Default Configuration Details

```
command.count=4
command[0].name=tc-lookup
command[0].displayname=ThreatConnect_Lookup
command[0].parameter.count=4
command[0].parameter[0].name=indicator
command[0].parameter[0].displayname=Indicator
command[0].parameter[1].name=api_access_id
command[0].parameter[1].displayname=API Access Id
command[0].parameter[2].name=api_secret_key
command[0].parameter[2].displayname=API Secret Key
command[0].parameter[3].name=api_base_url
command[0].parameter[3].displayname=API Base URL
command[0].action=python ${_ARCSIGHT_HOME}/threatconnect-
arcsight/lookup_indicator.py --indicator ${indicator} --api_access_id
```

```
${api_access_id} --api_secret_key ${api_secret_key} --api_base_url
${api_base_url}

command[1].name=tc-report-false-positive
command[1].displayname=ThreatConnect_Report_False_Positive
command[1].parameter.count=4
command[1].parameter[0].name=indicator
command[1].parameter[0].displayname=indicator
command[1].parameter[1].name=api_access_id
command[1].parameter[1].displayname=API Access Id
command[1].parameter[2].name=api_secret_key
command[1].parameter[2].displayname=API Secret Key
command[1].parameter[3].name=api_base_url
command[1].parameter[3].displayname=API Base URL
command[1].action=python ${_ARCSIGHT_HOME}/threatconnect-
arcsight/report_false_positive.py --indicator ${indicator} --api_access_id
${api_access_id} --api_secret_key ${api_secret_key} --api_base_url
${api_base_url}

command[2].name=tc-report-observation
command[2].displayname=ThreatConnect_Report_Observation
command[2].parameter.count=5
command[2].parameter[0].name=indicator
command[2].parameter[0].displayname=indicator
command[2].parameter[1].name=count
command[2].parameter[1].displayname=Count
command[2].parameter[2].name=api_access_id
command[2].parameter[2].displayname=API Access Id
command[2].parameter[3].name=api_secret_key
command[2].parameter[3].displayname=API Secret Key
command[2].parameter[4].name=api_base_url
command[2].parameter[4].displayname=API Base URL
command[2].action=python ${_ARCSIGHT_HOME}/threatconnect-
arcsight/report_observation.py --indicator ${indicator} --count ${count} --
api_access_id ${api_access_id} --api_secret_key ${api_secret_key} --
api_base_url ${api_base_url}

command[3].name=tc-add-indicator
command[3].displayname=ThreatConnect_Add_Indicator
command[3].parameter.count=9
command[3].parameter[0].name=indicator
command[3].parameter[0].displayname=indicator
command[3].parameter[1].name=owner
```

```
command[3].parameter[1].displayname=Owner
command[3].parameter[2].name=rating
command[3].parameter[2].displayname=Rating
command[3].parameter[3].name=confidence
command[3].parameter[3].displayname=Confidence
command[3].parameter[4].name=attribute
command[3].parameter[4].displayname=Attribute
command[3].parameter[5].name=tag
command[3].parameter[5].displayname=Tag
command[3].parameter[6].name=api_access_id
command[3].parameter[6].displayname=API Access Id
command[3].parameter[7].name=api_secret_key
command[3].parameter[7].displayname=API Secret Key
command[3].parameter[8].name=api_base_url
command[3].parameter[8].displayname=API Base URL
command[3].action=python ${_ARCSIGHT_HOME}/threatconnect-
arcsight/add_indicator.py --indicator ${indicator} --owner ${owner} --rating
${rating} --confidence ${confidence} --attribute ${attribute} --tag ${tag} --
api_access_id ${api_access_id} --api_secret_key ${api_secret_key} --
api_base_url ${api_base_url}
```