GreyNoise

Google SecOps SOAR Integration

User Guide

Version: 7.0

Table of Contents

Overview

GreyNoise Platform

GreyNoise is a threat intelligence platform that monitors internet wide scanning, crawling, and exploitation activity across the Internet. Its core value is helping security teams filter out the “noise” - that is, mass scanning, benign crawlers, or otherwise common scan activity that often triggers alerts but is not necessarily targeting your assets - so you can focus on truly malicious or targeted activity.

Google SecOps

Google SecOps is a modern, cloud-native SecOps platform that empowers security teams to better defend against today’s and tomorrow’s threats. By combining Google’s hyper-scale infrastructure, unparalleled visibility, and understanding of cyber adversaries, Google SecOps provides curated outcomes that proactively uncover the latest threats in near real-time, and enable security teams to detect, investigate and respond with speed and precision.

Google SecOps Integration for GreyNoise

The SOAR component extends GreyNoise capabilities with powerful automation and response features. It enables bi-directional synchronization of threat intelligence between GreyNoise and Google SecOps, allowing security teams to operationalize threat data effectively. Through automated playbooks, security teams can implement rapid response actions such as blocking malicious indicators, updating threat intelligence, and managing security incidents at scale.

Release Notes

V7.0

- Upgraded GreyNoise SDK to version 3.0.1
- Added Get CVE Details action to retrieve CVE information from GreyNoise
- Added the following parameters to the Execute GNQL Query action:
  - Exclude Raw
  - Quick
- Updated parameters in IP Timeline Lookup action:
  - Removed: Limit
  - Added: Field
  - Added: Granularity
- Removed the following actions:
  - Community IP Lookup (use IP Lookup action instead)
  - RIOT IP Lookup (Deprecated use IP Lookup action instead)
  - IP Similarity Lookup (Deprecated)

App Installation & Configuration

Pre-Requisites

Google SecOps

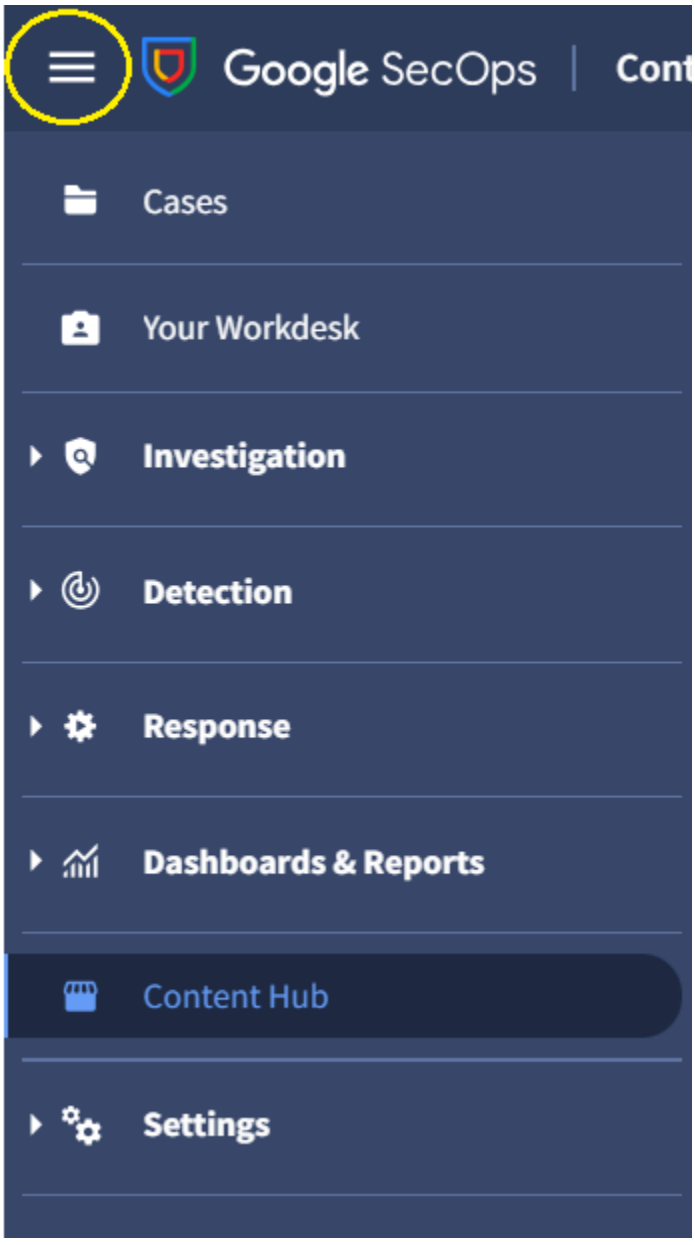
- A Google SecOps SOAR Instance with an Admin role user.

GreyNoise

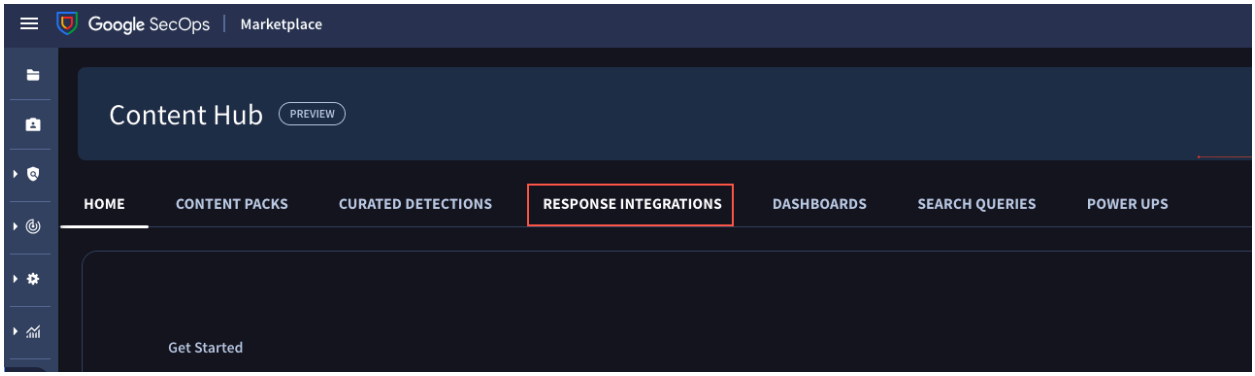
- GreyNoise Intelligence platform (API Key).

Installing the Integration

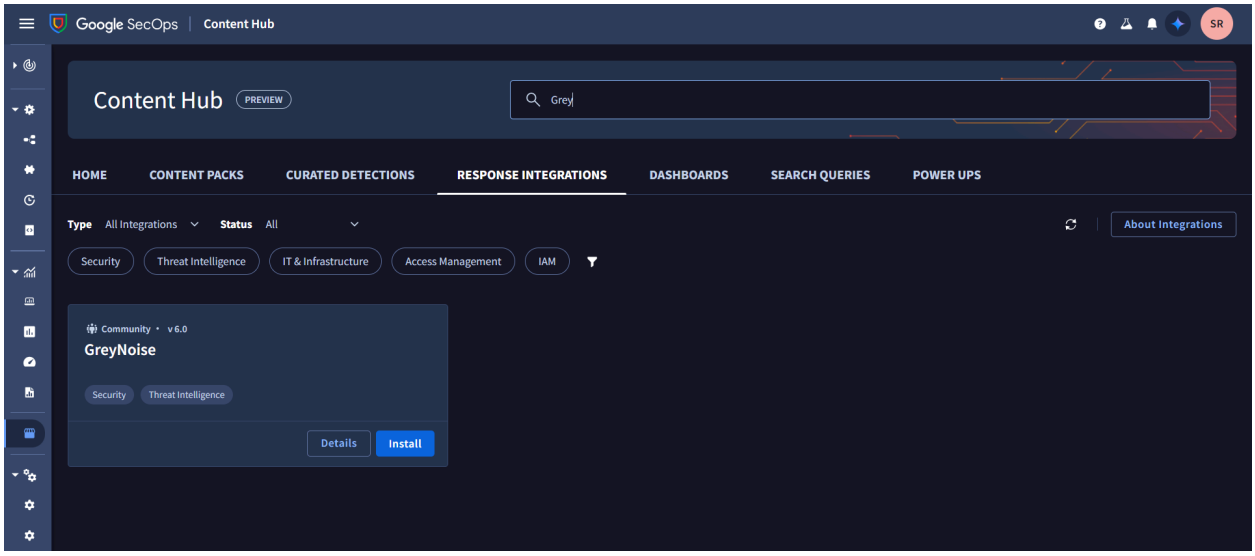
1. Log in to the Google SecOps SOAR Platform.
2. From the Sidebar, click on the **Content Hub** section



3. In the Content Hub, click on the Response Integrations tab.



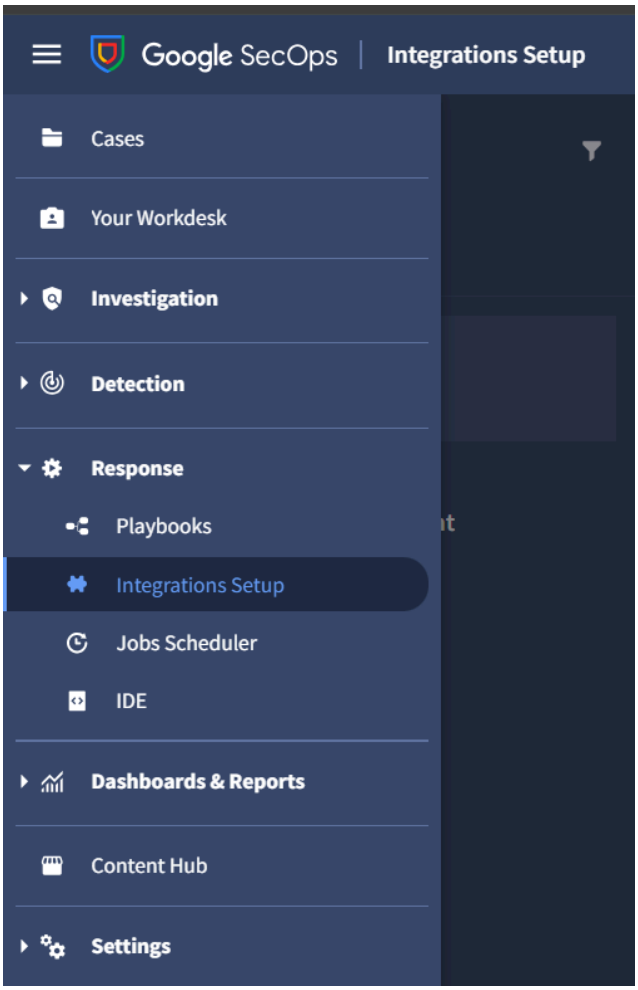
4. From the **Response Integrations** tab, search for **GreyNoise**.



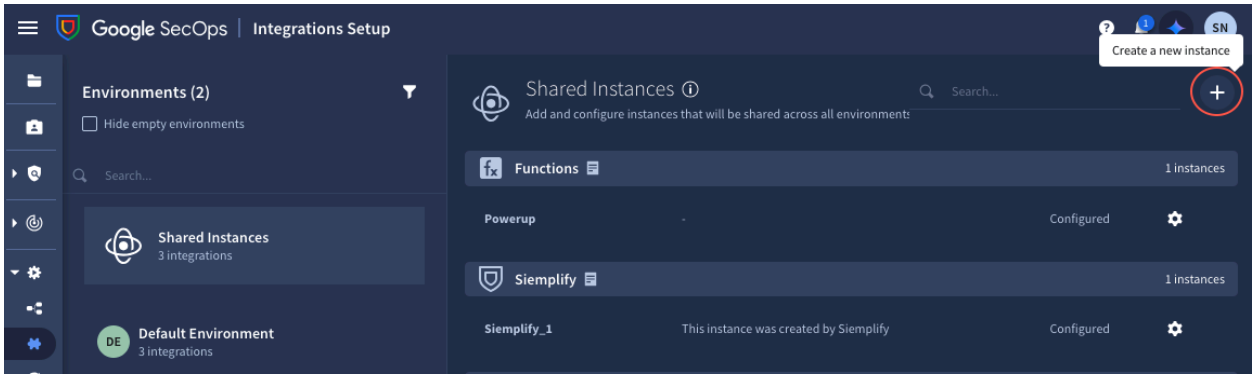
5. Click the **Install** button to install the integration.

Configuration

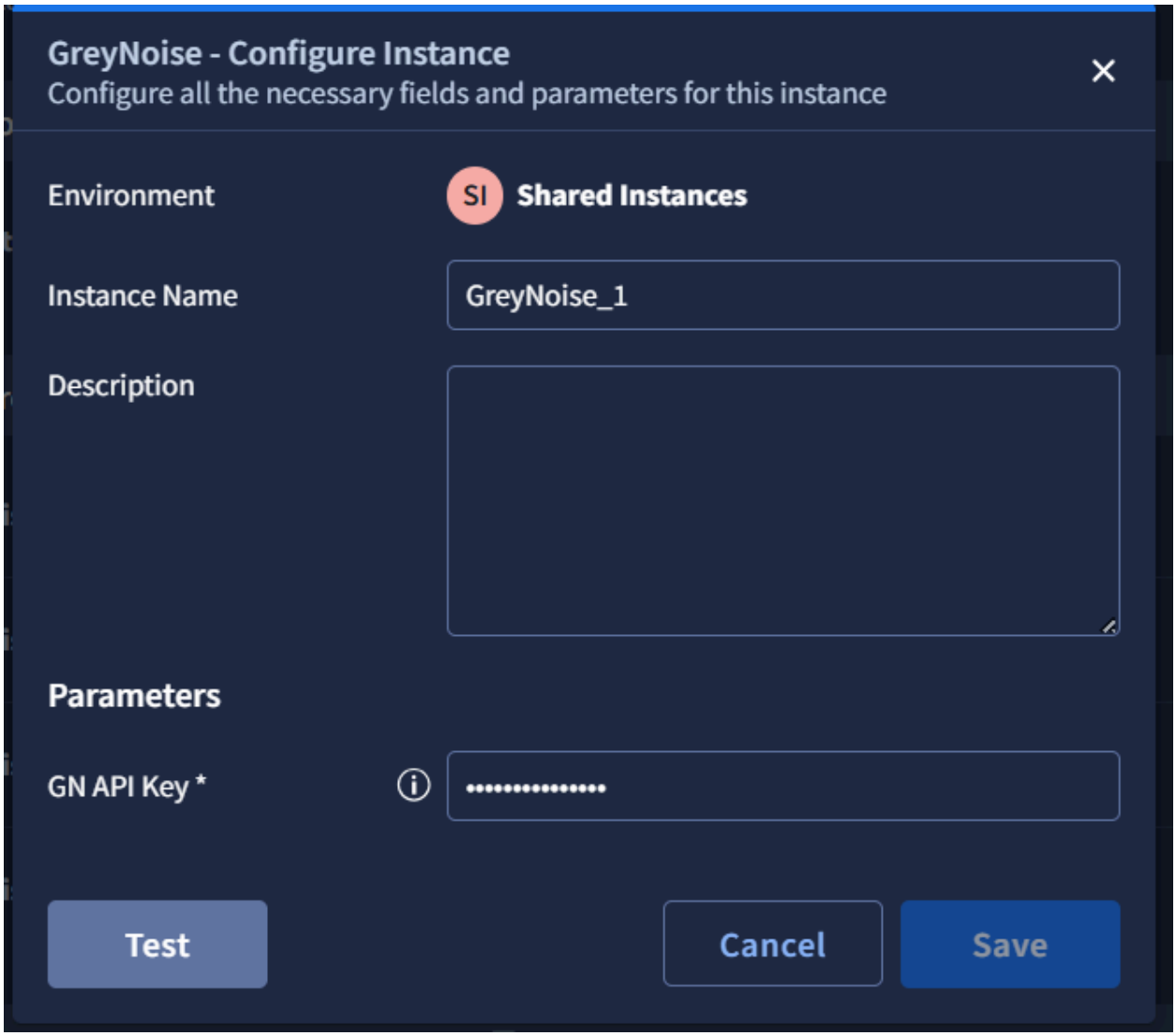
1. Once the integration has been installed successfully, the user needs to configure it to use the actions provided by the integration. These actions can be run manually from a case or can be used in playbooks.
2. To configure the integration, users can navigate to **Response** > **Integrations Setup**.



3. The integration can be configured in any of the desired environments. The integration actions can be used for playbooks and cases belonging to the environment in which the integration is configured.
4. Click on the **Plus** (**+**) button to configure the integration for the desired environment.

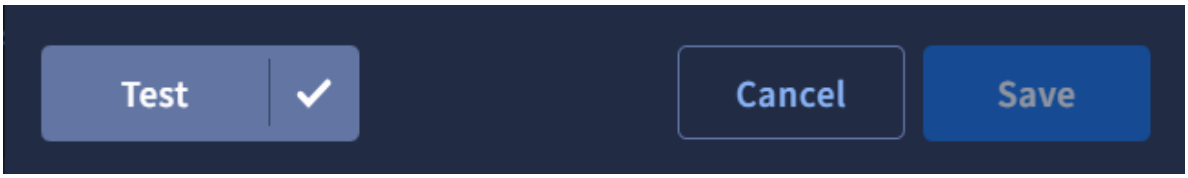


5. Search for GreyNoise integration and select it. Click **Save**.
6. The instance of integration will be saved. However, in the next pop-up, users must add the necessary configuration parameters to use the actions.

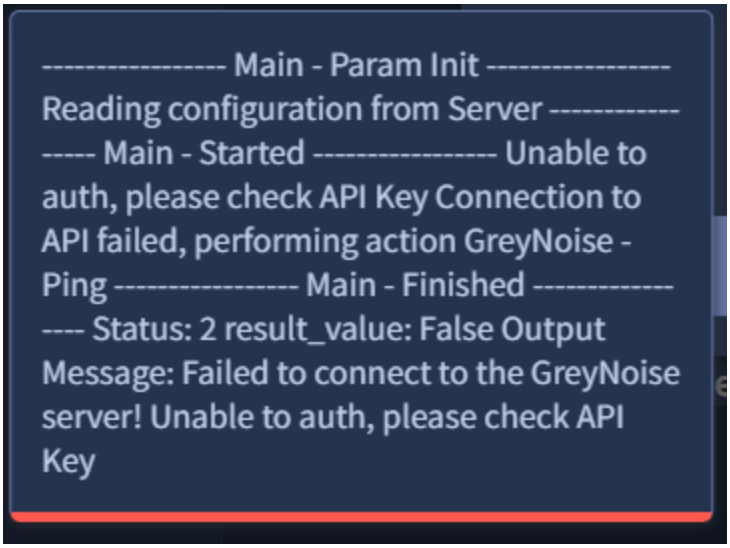


Field name	Description	Default value	Required	Type
GN API Key	GreyNoise API Key	-	True	String

7. After adding the valid configuration parameters, click on the **Save** button to save the configurations. This is required to access the GreyNoise APIs used in the actions.
8. After saving the configurations, the user can validate them using the **Test** button
    a. A green tick mark appears if the credentials provided are valid.



    b. If invalid credentials are provided or any internal issue is faced, a red cross mark on the right side of the Test button will appear. The full error message can be displayed by clicking the red cross button.



Note:

- After making any changes in the configuration parameters, the user needs to save them before testing again using the Test button.
- When updating the credentials, the user must delete the entire Password from the input first before adding the updated Password.

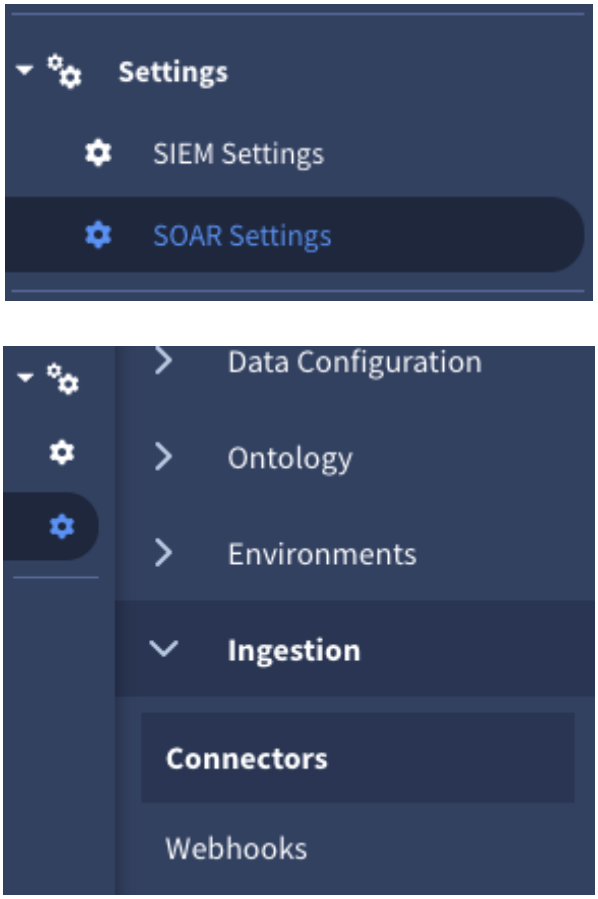
More information about configuring integrations in Google SecOps can be found *here*.

GNQL Connector

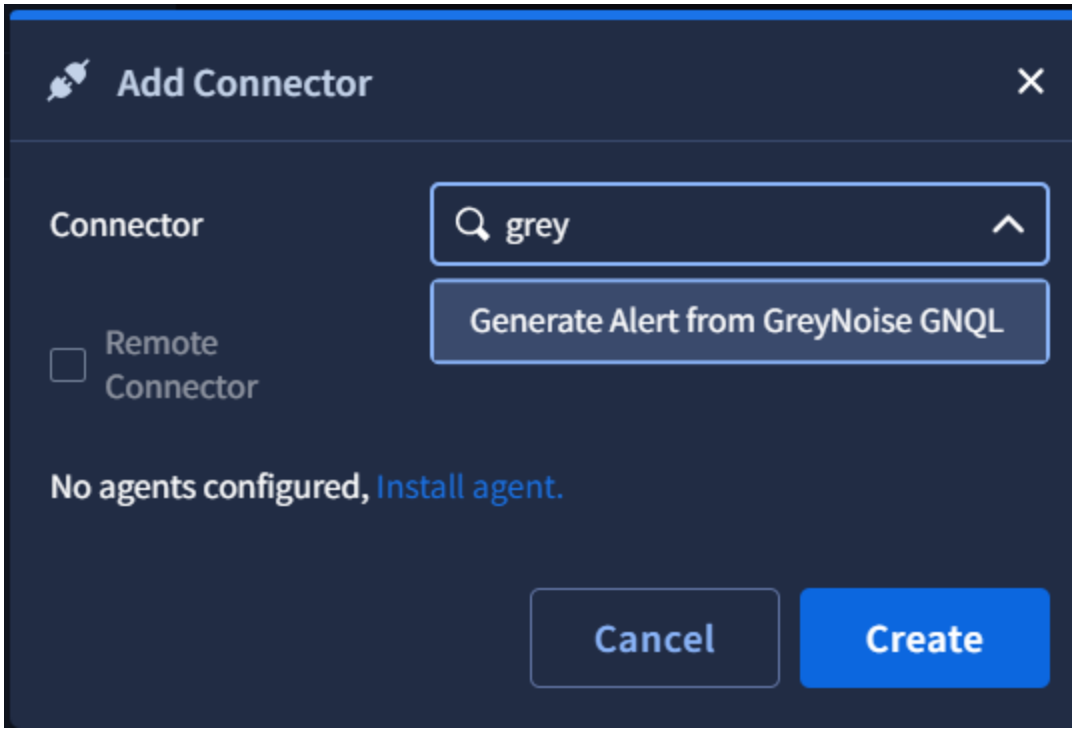
The connector automates the process of querying GreyNoise GNQL API to retrieve threat intelligence data and generate corresponding alerts in Google SecOps. It helps identify scanning or malicious IPs observed by GreyNoise and transforms those findings into actionable alerts.

Connector Configuration

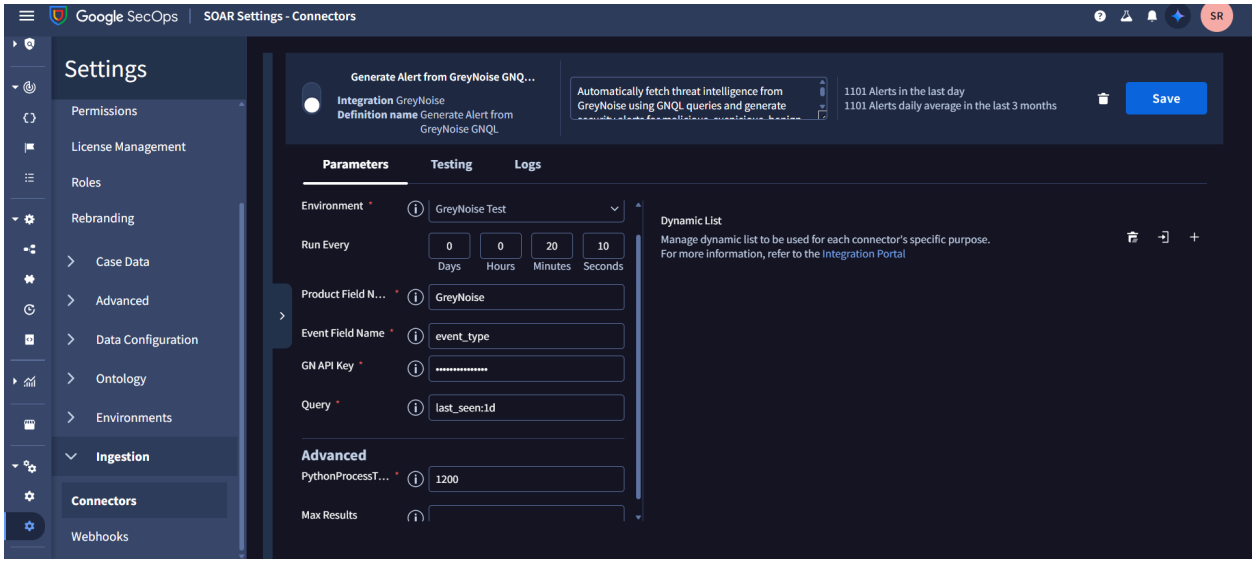
1. Go to **SOAR Settings** > **Ingestion** > **Connectors**.



2. Select the **+** icon to add a new connector and then select Generate Alert from GreyNoise GNQL Connector from the dropdown and then Create.



3. Complete the parameters configuration, enable and **Save** the connector.



Field name	Description	Default value	Required	Type
API Key	API Token	N/A	True	Password

Query	GNQL query	last_seen:1d	True	String
Max Results	The number of max results to fetch.	100	False	Integer

Actions

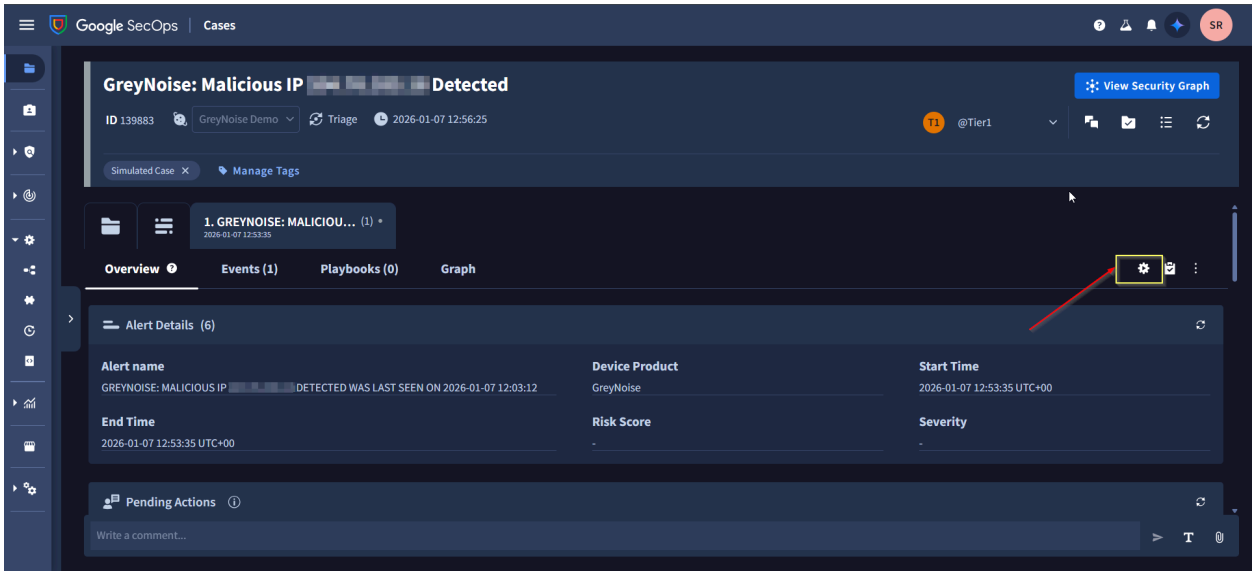
This integration implements investigative and generic actions for the GreyNoise platform on the Google SecOps SOAR Platform. It enables end-users to implement comprehensive network security use cases on the GreyNoise Platform through the below available actions.

1. **Test Connectivity (ping):** Tests the connectivity of the Google SecOps SOAR server to the GreyNoise platform by validating the provided GreyNoise API credentials.
2. **Quick IP Lookup:** Performs a rapid check of an IP address using the GreyNoise SDK to determine if it is associated with internet scanning activity or a known business service. Returns trust level, classification, scanner, and business service fields.
3. **Execute GNQL Query:** Executes a GreyNoise Query Language (GNQL) query, returning a list of IP addresses and metadata that match the query criteria. Supports pagination, partial field omission, and error handling for invalid keys and queries.
4. **IP Lookup:** Performs a comprehensive lookup for one or more IP addresses using the GreyNoise SDK, returning enriched intelligence such as classification, noise/RIOT status, actor, tags, CVEs, behavioral flags, business service information, scanner status, and more.
5. **IP Timeline Lookup:** Retrieves a historical activity timeline for a specified IP address, showing changes over a selected field (e.g., classification, ports) with configurable day/hour granularity using GreyNoise data.
6. **Get CVE Details:** Fetches detailed information about a specific CVE, providing vulnerability info, CVSS/EPSS scores, exploitation stats, threat actor activity, exploitation timeline, and observed related IP activity from GreyNoise sensors.

Invoking Actions

Invoke Action Manually (from Case)

1. Navigate to the Case **Overview** tab of the case on which the action needs to be performed.
2. In the selected case, click Manual Action (‘**gear**’ icon) located on the right side under the Case Top Bar. The Manual Action dialog box appears.



3. Select the required action: **GreyNoise** > [**Action Name**]. Make sure to fill in the required information.
4. Once the information is added, click on the **Execute** button.
5. Once the action is successfully executed, the result (success or failure) will be displayed on the Case wall.
6. The result contains different JSON output and insights based on different actions.

Invoke Action Automatically (from Playbook/Block)

Actions can be added as a step in a playbook. This action will be run when the playbook is executed. Playbooks, in turn, can be triggered on cases and alerts based on the conditions determined by the user.

To add an action as a part of the playbook, follow these steps:

1. Navigate to **Response** > **Playbooks**
2. Select any existing playbook or create a new one.
3. Click on the **+ Open Step Selection** button, and navigate to the Actions tab.
4. Under GreyNoise integration, all the actions available in the integration would be visible. Simply, drag and drop the action required to the playbook step.
5. The action should now be added to the playbook. Make sure to configure the input parameters (if applicable) for the action.

Playbooks

As part of the GreyNoise Integration, we created **Five** playbooks and a block to help the user get started. Users can use and refer to the sample playbook provided in Github Repo

The Playbooks & block included are:

1. CVE-Based Network Containment Orchestrator - GreyNoise
2. Noise Elimination - GreyNoise
3. Network Containment Based on GreyNoise IP Feed - GreyNoise
4. Dynamic Case Naming from Webhook - GreyNoise
5. Network Containment Block - GreyNoise

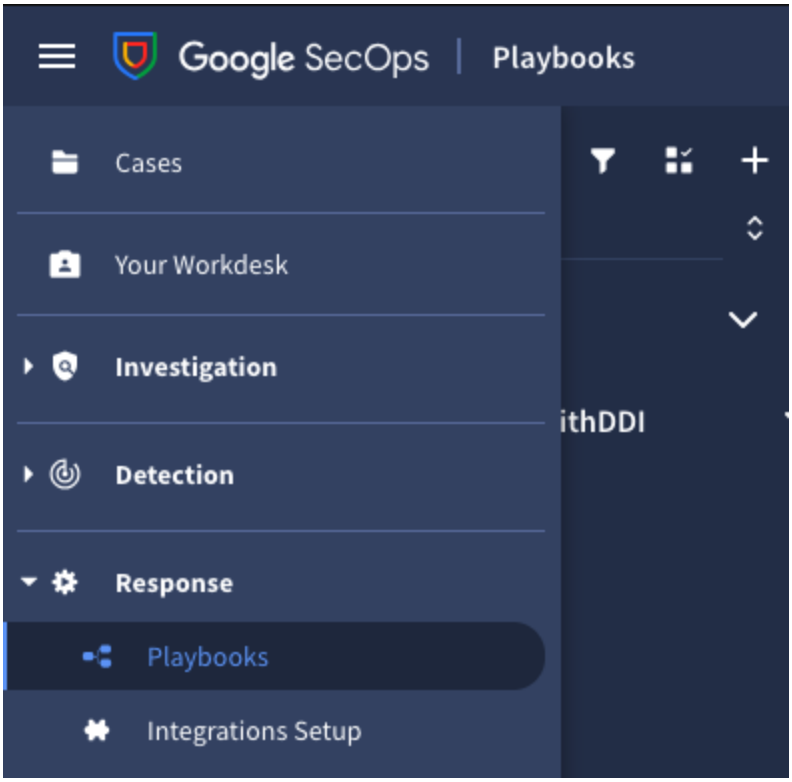
Notes:

- .zip file for every playbook contains the playbook along with all the blocks used in that playbook.
- Each playbook requires a trigger, which decides when the playbook would be run. Users can update the Trigger conditions based on the requirements.
- As the trigger of every playbook is set to run on all ingested alerts into SecOps instance, the imported playbooks are disabled by default. Imported Playbooks and a block can be found in the **Imported Playbooks** folder. These playbooks and blocks can be moved to any other folder of User's choice. Users can easily enable any playbook by simply clicking on the toggle button beside the playbook name. *(Reference)*

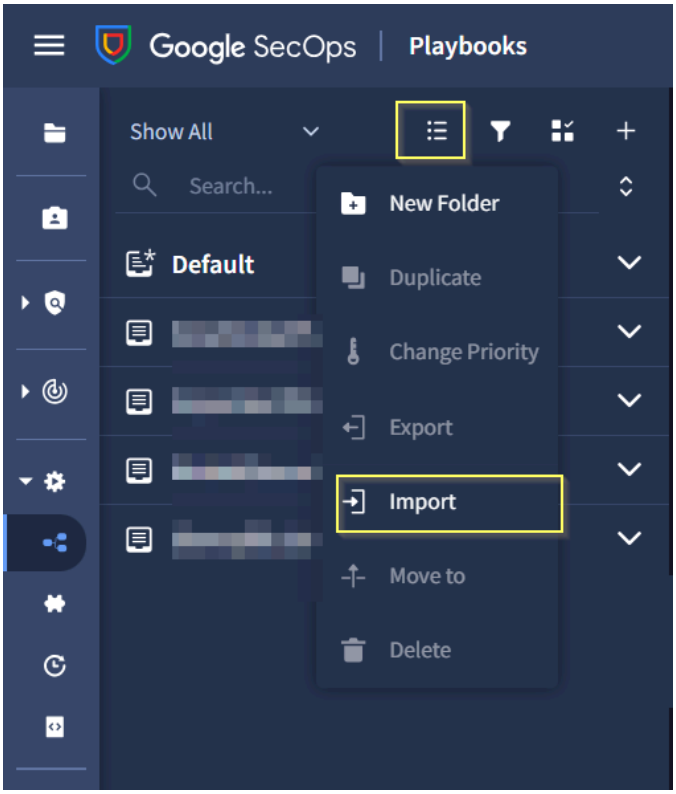
Import Playbooks

To import a playbook into any SecOps instance follow the below steps:

1. Download the .zip file of the playbook that needs to be imported from *Playbooks* . This file would also contain all the blocks that are used in the playbook.
2. Navigate to **Response** > **Playbooks**.



3. Click on the **three dots** icon at the top of the Playbooks, and then click on **Import**.



4. Select the .zip file downloaded earlier.

   The playbooks stored in the file would be imported. We can access these under the **Imported Playbooks** folder. These playbooks and blocks can be moved to any other folder of our choice.

   You can also download the All GreyNoise Playbooks Bundle which includes all the GreyNoise playbooks and blocks.

   Note: *Before using any playbook with an alert, review its default settings (triggers, action parameters, etc.) to ensure they align with your needs*

Webhook

The GreyNoise for Google SecOps SOAR Integration will support ingestion of Alerts and Feeds data via webhook delivery. Webhooks will allow GreyNoise events to be delivered directly into Google SecOps SOAR.

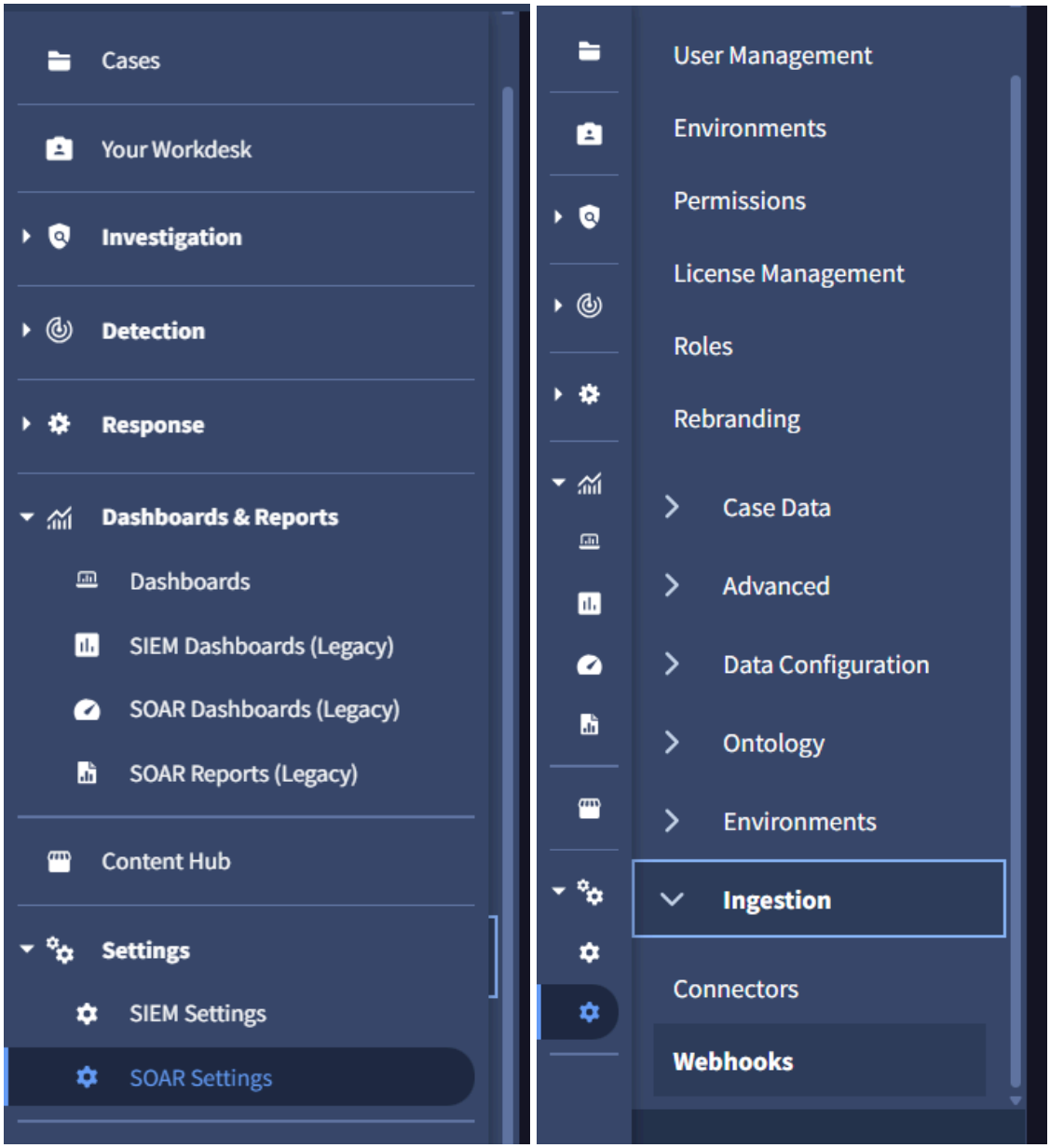
To ingest the Alerts and Feed data using the Webhook we need to create 3 webhooks to support the different schemas.

1. **GreyNoise Alert Webhook**: Supports all the Alert Schemas (IP, CVE, TAGS, GNQL Query)
2. **GreyNoise IP Change Webhook**: Supports only the IP Classification Change Feed.
3. **GreyNoise CVE Change Webhook**: Supports the CVE Spike Activity and CVE Status Change Feed.

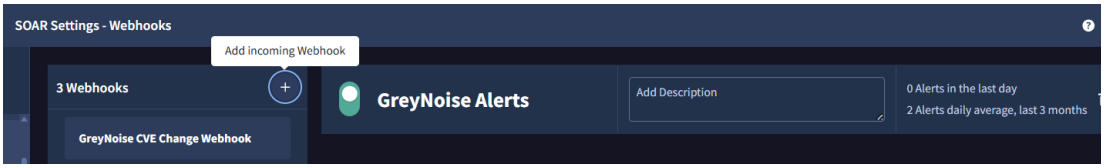
Note: Based on the User's requirement he needs to configure the relevant webhook in the Google SecOps.

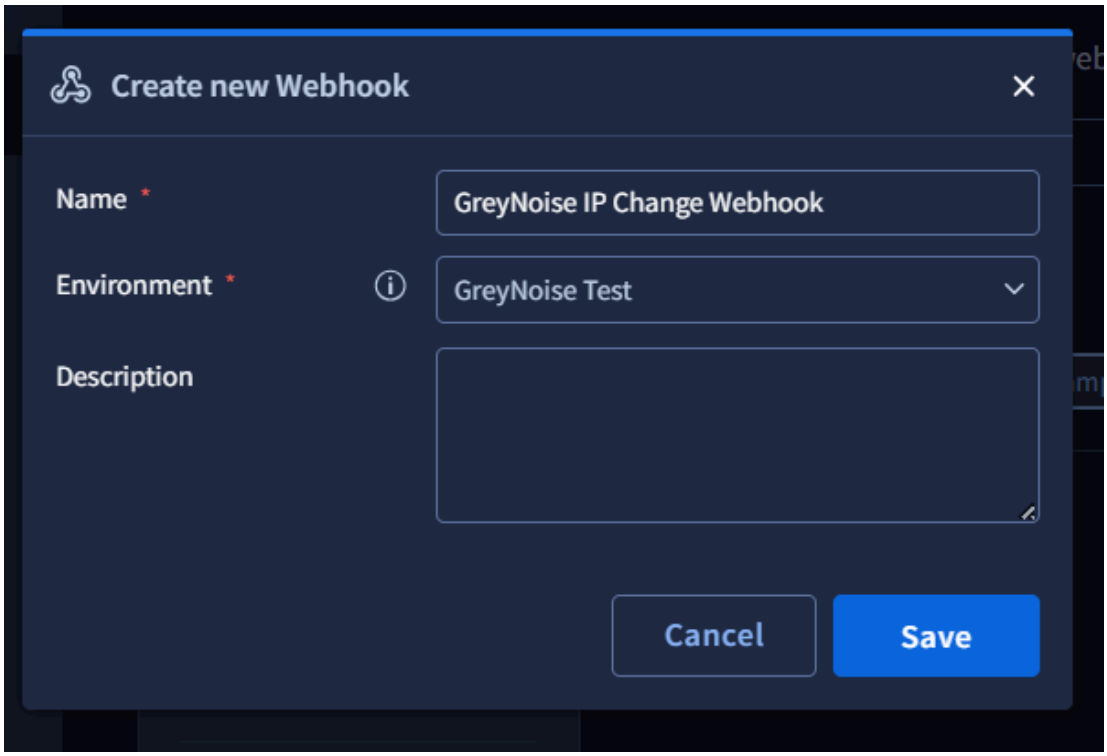
Webhook Configuration

1. **Access Google SecOps SOAR:**
   - Log in to the Google SecOps SOAR UI as an admin.
   - Navigate to **SOAR Settings** → **Ingestion** → **Webhooks**.

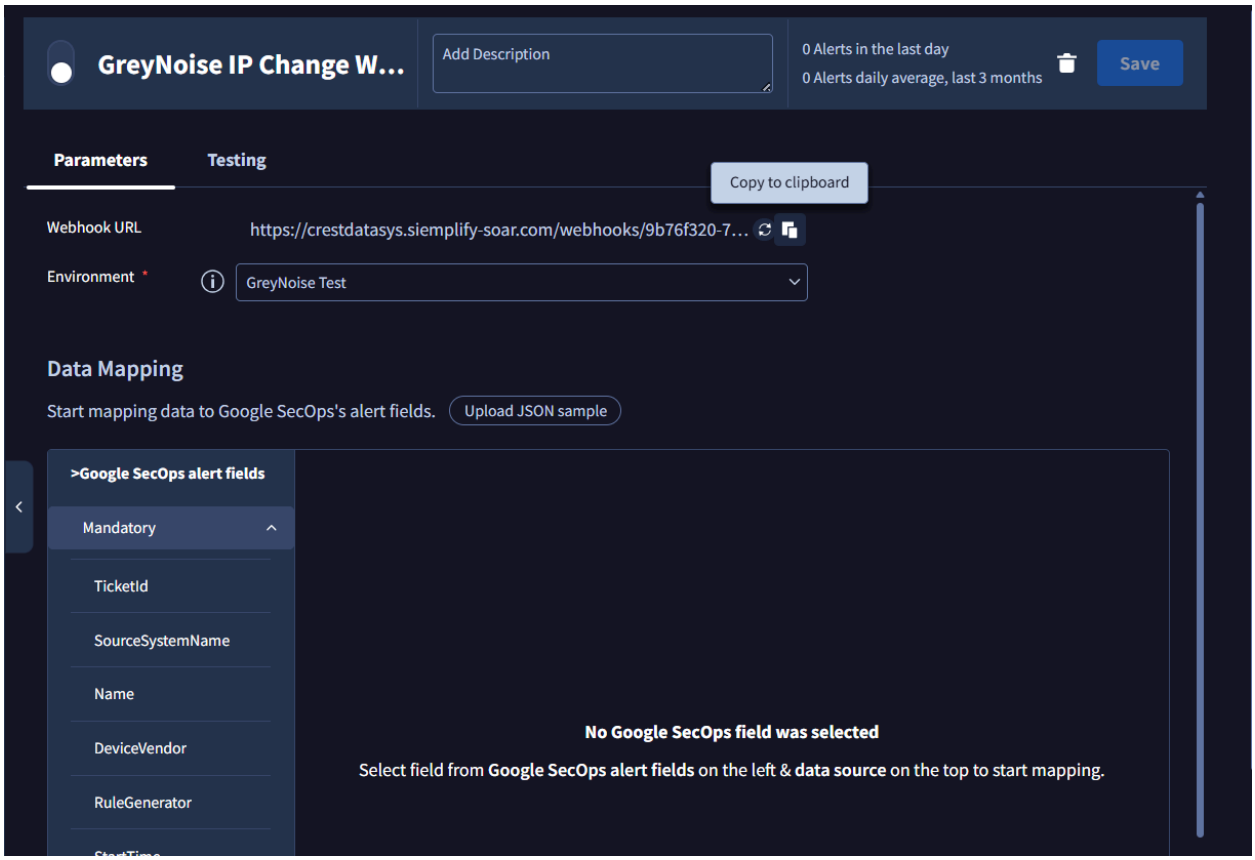


2. **Add Incoming Webhook:**
   - Click **Add Incoming Webhook (+)**.
   - Enter a Name for this webhook (e.g., GreyNoise Alert Webhook).
   - Choose the appropriate Environment. (where user wants to ingest the alerts)
   - Click **Save**.

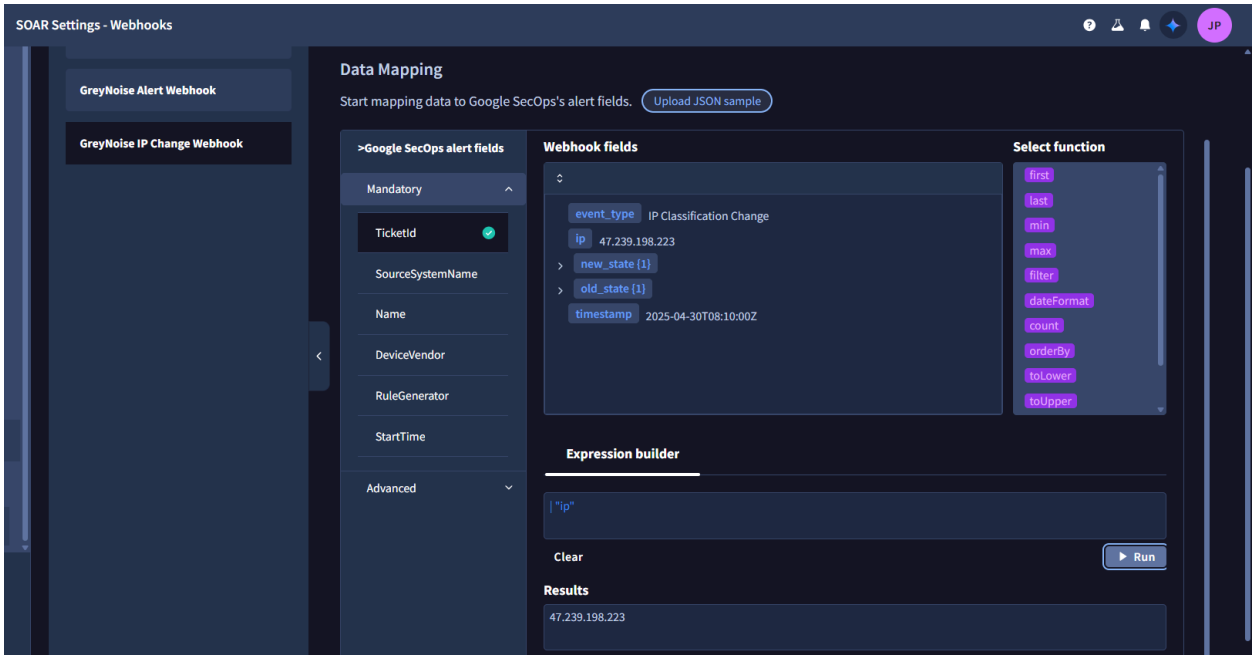




3. **Generate Webhook URL:**
   - After saving, the platform will generate a webhook URL.
   - **Important:** Copy the webhook URL immediately — it cannot be retrieved again once you leave the page. You will need this URL for configuring your source (e.g., GreyNoise).



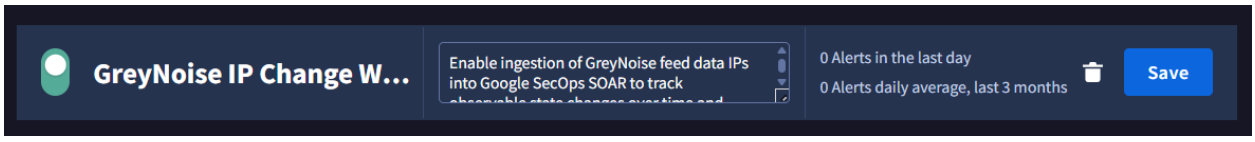
4. **Map the Alert Fields:**
   - In the Data Mapping section, click **Upload JSON sample**.
   - Upload a representative webhook payload from your alert source (e.g., a sample GreyNoise webhook JSON).
   - Use the Expression Builder to map each field in SOAR to a JSON path or literal value.
   - After mapping, use the **Run** button in the Expression Builder to preview output and confirm proper mapping (green checkmark indicates success).



Notes:

- The static values will not be visible in the preview output. However, they will be saved and will be populated during the ingestion process.
- We have temporarily mapped Alert Priority to the classification field even though Priority is an enum; this mapping will not correctly determine priority due to webhook expression builder limitations. The current webhook mapping UI does not support nested if-else logic to map multiple classification values to priority levels (e.g., “malicious → CRITICAL”, “benign → MEDIUM”), as discussed in the community forum.

5. **Enable the Webhook:**
   - Once all necessary fields are mapped, click **Save** and then **Enable** the webhook.



6. **Setup on GreyNoise:**
   - Follow the instructions from GreyNoise to set up Alerts and Feeds.

1. Alerts Webhook

1.1. Goal

To enable ingestion of GreyNoise Alert data into Google SecOps SOAR using webhooks.

1.2. Mapping

Field	Mapping Value	Value Type	Description
TicketID	alert.id	Dynamic	Unique identifier for the alert
SourceSystemName	GreyNoise	Static	Identifies the source system
Name	alert.name	Dynamic	Name of the alert
DeviceVendor	GreyNoise	Static	Vendor information
RuleGenerator	GreyNoise-Alert	Static	Static value for consistency
StartTime	timestamp	Dynamic	Time the alert was generated
Reason	alert.creator	Dynamic	Creator of the alert
DeviceProduct	GreyNoise-Alert	Static	Product information
EndTime	timestamp	Dynamic	Time the alert ended
Priority	data.classification	Dynamic	Classification of the alert
EventsList	data | first(1)	Dynamic	List of events

			Note: Here we are only considering the first record so it can be easily accessible in the playbook.
EventProduct	GreyNoise	Static	Product information
EventName	alert.name	Dynamic	Name of the event

1.3. Supported Webhook Schemas

IP


```
{
   "alert": {
      "creator": "crest.phantom@crestdata.ai",
      "id": "3d19d419-c375-43fd-8f1a-996dcd463af7",
      "name": "IP Alert",
      "type": "ip"
   },
   "alert_link":
"https://viz.greynoise.io/account/alerts?alert=3d19d419-c375-43fd-8f1a-996
dcd463af7",
   "data": [
      {
         "classification": "suspicious",
         "ip": "176.236.29.137"
      }
   ],
   "query_link":
"https://api.greynoise.io/v2/experimental/gnql?query=ip%3A%20176.236.29.
137",
   "timestamp": "2025-07-31T19:01:15Z",
   "viz_link": "https://viz.greynoise.io/query/ip%3A%20176.236.29.137"
}
```

CVE


```
{
   "alert": {
      "creator": "crest.phantom@crestdata.ai",
      "id": "78848d96-ee88-460b-841b-dd4d9c8ff613",
      "name": "CVE Alert",
      "type": "cve"
   },
   "alert_link":
"https://viz.greynoise.io/account/alerts?alert=78848d96-ee88-460b-841b-dd
4d9c8ff613",
   "data": [
      {
         "classification": "malicious",
         "ip": "220.247.174.202"
      }
   ],
   "query_link":
"https://api.greynoise.io/v2/experimental/gnql?query=cve%3A%20CVE-2013-
2134",
   "timestamp": "2025-07-31T19:01:13Z",
   "viz_link": "https://viz.greynoise.io/query/cve%3A%20CVE-2013-2134"
}
```


TAG


```
{
   "alert": {
      "creator": "crest.phantom@crestdata.ai",
      "id": "43b6990e-01b7-468c-afae-0015e3d77095",
      "name": "SIP REGISTER Scanner",
      "type": "tag"
   },
   "alert_link":
"https://viz.greynoise.io/account/alerts?alert=43b6990e-01b7-468c-afae-00
15e3d77095",
   "data": [
      {
```

```
        "classification": "malicious",
        "ip": "168.100.239.215"
      }
   ],
   "query_link":
"https://api.greynoise.io/v2/experimental/gnql?query=tags%3A%20SIP%20
REGISTER%20Scanner",
   "timestamp": "2025-08-06T21:01:25Z",
   "viz_link":
"https://viz.greynoise.io/query/tags%3A%20SIP%20REGISTER%20Scanner"
}
```


GNQL Query


```
{
   "alert": {
      "creator": "crest.phantom@crestdata.ai",
      "id": "0ed61042-6859-4dd7-9ede-cd373d0d7df5",
      "name": "GNQL Alert",
      "type": "query"
   },
   "alert_link":
"https://viz.greynoise.io/account/alerts?alert=0ed61042-6859-4dd7-9ede-cd
373d0d7df5",
   "data": [
      {
         "classification": "malicious",
         "ip": "216.73.216.124"
      }
   ],
   "query_link":
"https://api.greynoise.io/v2/experimental/gnql?query=last_seen%3A1d%20cl
assification%3Amalicious%20bot%3Atrue",
   "timestamp": "2025-07-31T19:01:12Z",
   "viz_link":
"https://viz.greynoise.io/query/last_seen%3A1d%20classification%3Amalicio
us%20bot%3Atrue"
}
```

2. Feeds Webhook

2.1. Goal

Enable ingestion of GreyNoise feed data (IPs and CVEs) into Google SecOps SOAR to track observable state changes over time and trigger appropriate playbooks based on updates.

Note: You Need to create two webhooks, one for the IP Classification Change and another for CVEs (Spike and Status Change)

2.2. Mapping

Mapping for CVE Status Change and CVE Activity Spike Webhook

Field	Mapping Value	Value Type	Description
TicketID	cve	Dynamic	Unique identifier for the CVE event
SourceSystemName	GreyNoise	Static	Identifies the source system
Name	event_type	Dynamic	Type of CVE event
DeviceVendor	GreyNoise	Static	Vendor information
RuleGenerator	GreyNoise-Feed	Static	Static value for consistency
StartTime	timestamp	Dynamic	Time the event was generated
DeviceProduct	GreyNoise-Feed	Static	Product information
EndTime	timestamp	Dynamic	Time the event ended

EventsList	first(1)	Dynamic	List of events
EventProduct	GreyNoise	Static	Product information
EventName	event_type	Dynamic	Name of the event

Mapping for IP Classification Change Webhook

Field	Mapping Value	Value Type	Description
TicketID	ip	Dynamic	Unique identifier for the IP event
SourceSystemName	GreyNoise	Static	Identifies the source system
Name	event_type	Dynamic	Type of IP event
DeviceVendor	GreyNoise	Static	Vendor information
RuleGenerator	GreyNoise-Feed	Static	Static value for consistency
StartTime	timestamp	Dynamic	Time the event was generated
DeviceProduct	GreyNoise-Feed	Static	Product information
EndTime	timestamp	Dynamic	Time the event ended
Priority	new_state.classification	Dynamic	New classification state
EventsList	first(1)	Dynamic	List of events

EventProduct	GreyNoise	Static	Product information
EventName	event_type	Dynamic	Name of the event

2.3. Supported Webhook Schemas

CVE Status Change


```
{
   "cve": "CVE-2022-31718",
   "event_type": "cve-status-change",
   "metadata": {},
   "new_state": {
      "activity_seen": false,
      "benign_ip_count_10d": 0,
      "benign_ip_count_1d": 0,
      "benign_ip_count_30d": 0,
      "threat_ip_count_10d": 1,
      "threat_ip_count_1d": 1,
      "threat_ip_count_30d": 1
   },
   "old_state": {
      "activity_seen": true,
      "benign_ip_count_10d": 0,
      "benign_ip_count_1d": 0,
      "benign_ip_count_30d": 0,
      "threat_ip_count_10d": 0,
      "threat_ip_count_1d": 0,
      "threat_ip_count_30d": 0
   },
   "timestamp": "2025-08-11T10:30:16.972504375Z"
}
```


CVE Activity Spike


```
{
```

```
  "ip": "47.239.198.223",
  "tag": "Palo Alto Networks Login Scanner",
  "event_type": "New IP Activity",
  "event": "added",
  "timestamp": "2025-04-30T08:10:00Z"
}
```


IP Classification Change


```
{
   "event_type": "ip-classification-change",
   "ip": "86.57.2.53",
   "new_state": {"classification": "malicious"},
   "old_state": {"classification": "unknown"},
   "timestamp": "2025-08-11T10:42:39Z",
   "workspace_id": "e4a5be2e-1be0-4105-a5e2-51e6a5525fa0"
}
```

Ontology Mapping

Ontology mapping in Google SecOps SOAR enables automatic entity extraction and relationship building from GreyNoise Webhook and Connector data. This enhances threat intelligence correlation, investigation efficiency, and automated response capabilities.

Import Ontology Manually

- Navigate to **Settings -> SOAR Settings -> Ontology**
- Open **Ontology Status.** Use the Import option available on the right-hand side in the middle of the page.
- Select the downloaded Ontology zip**.** It will be imported.

Note: Ontology will be bundled as part of the integration once it gets published officially.

Ontology Mapping Table

Entity Field	Source Field(s)
SourceAddress	ip
CVE	cve

Limitations

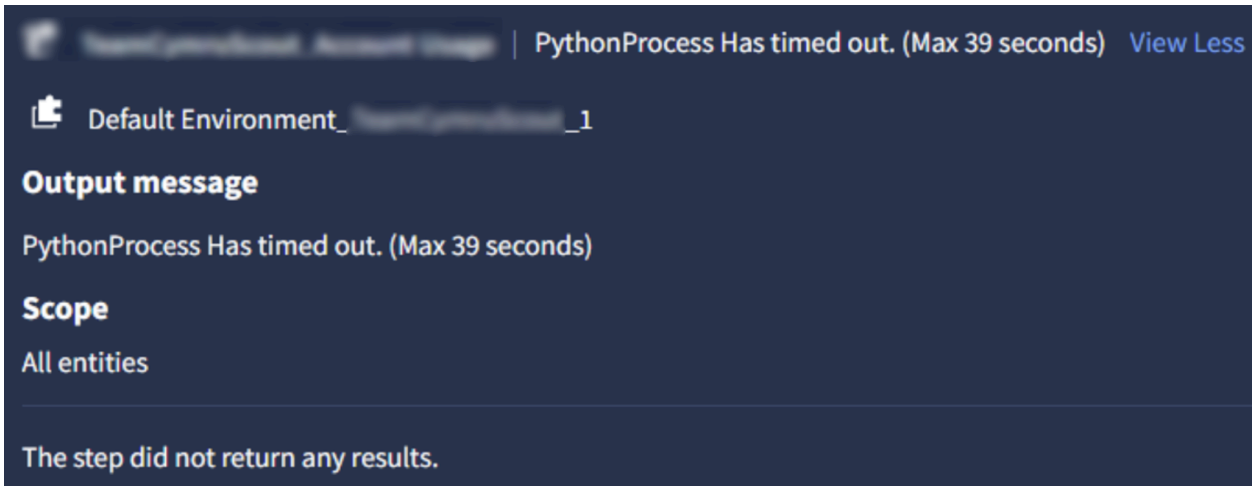
- For **Execute GNQL Query** action, keep max result value under 1,000 to avoid exceeding Google SecOps action payload limitation of **28 MB**.
- Once the indicator is ingested into Google SecOps and an alert is created, it cannot be updated in the next polling interval due to platform limitations in Google SecOps.
- Currently we have temporarily mapped `Alert Priority` to the `classification` field for webhooks, even though Priority is an enum field and the direct mapping will not produce the expected results. The webhook expression builder in Google SecOps does not support nested if-else logic for mapping multiple classification values to priority levels (e.g., mapping “malicious → CRITICAL”, “suspicious → HIGH”, etc.), as discussed in the community. This limitation means priority determination must be handled via playbooks until more flexible mapping is supported in webhook.

Known Behaviors

- We observed that, at times, HTML content (including styles) does not render as expected in the Case Wall, even though the same content renders correctly in the Entity Insight (Overview tab).
- The script timeout might not work as expected. It has been observed that when specifying the timeout to 10 seconds, the action script would timeout after around 39 seconds. The platform adds ~29 seconds of timeout. This behavior differs for every SecOps SOAR instance.
  Reference: Solved: Facing inconsistencies in Script Timeout for Actio... - Google Cloud Community
- The connector execution is not guaranteed to occur exactly at the configured Run Every interval, it will be triggered once at some point during the specified interval window due to platform scheduling behavior.
- Configuring Max Results to a large value for the connector can significantly increase data fetch and processing time. In such cases, the connector execution may exceed the allowed runtime and time out before completion. This behavior depends on the volume of data being retrieved by the connector.

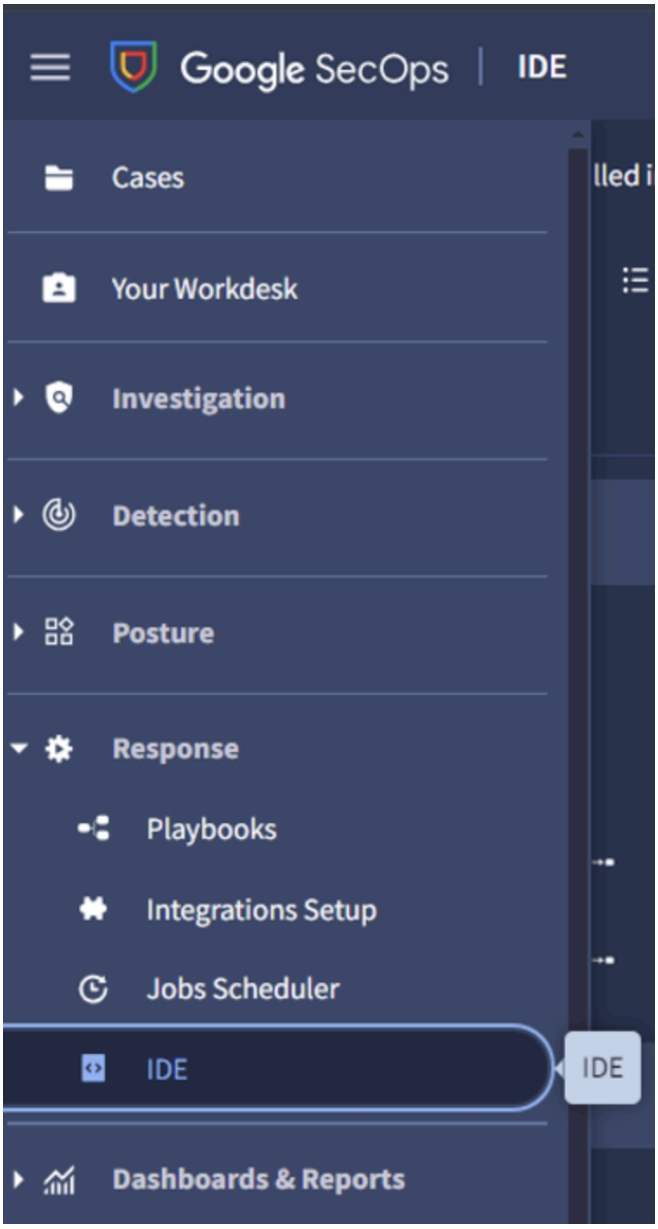
Troubleshooting

Case 1: PythonProcess Has timed out.

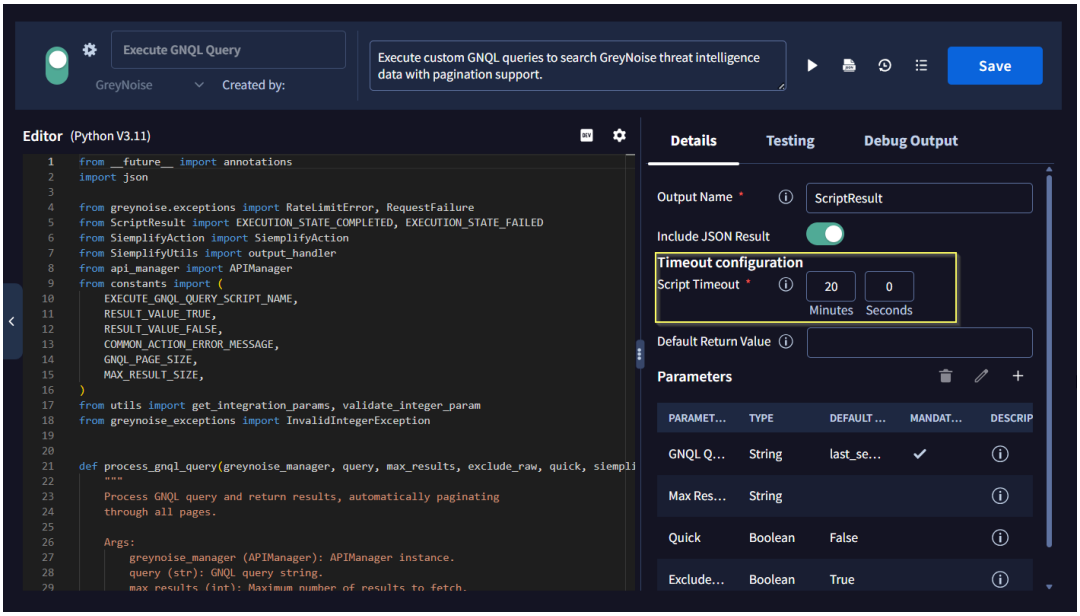


Solution :

1. Increase the script timeout from the Timeout Configuration of an action
    a. From the Sidebar, navigate to **Response** > **IDE**

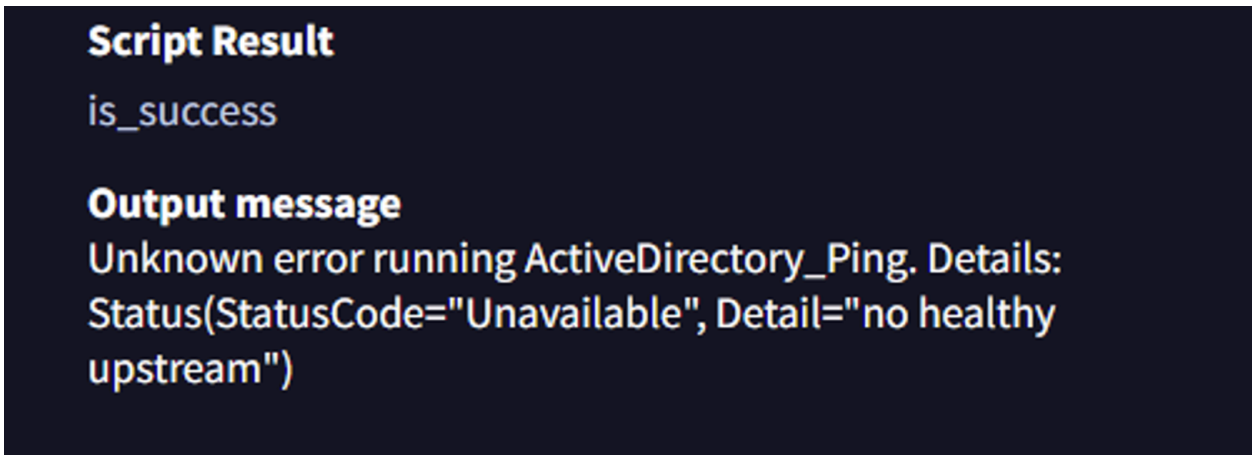


b. Search for the action script and select it. You will be able to see the codebase of the action, and on the right side, there should be an input field named Timeout Configuration under the details tab. Update the value of Script Timeout based on your requirement, and Save the action



2. Another way to increase the execution speed of the action is to tweak the query parameters of the API call by reducing the time frame to search and retrieve only filtered subsets instead of all results

Case 2: Unknown error running Action.



Solution:

The "no healthy upstream" error indicates an internal system issue. The most common cause is that all backend servers are currently unavailable or unresponsive. To resolve this problem, you should contact your system administrator for assistance.