GreyNoise App For Google SecOps SIEM

User Guide

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 SIEM

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

This integration enables the seamless ingestion, parsing, and visualization of GreyNoise threat intelligence Indicator data within Google SecOps SIEM, offering comprehensive insights through native Google SecOps dashboards. It supports real-time delivery of indicators from GreyNoise's threat intelligence platform, automatically enriching security events with contextual threat data.

Release Notes

V1.0.0

- Provided the ingestion script which can deploy as a GCP cloud function to collect indicators from the GreyNoise Intelligence Platform and ingest in the Google SecOps.
- Provided the parser that processes data ingested from the GreyNoise Intelligence platform and converts it into the Google SecOps UDM data model.
- Provided Detection Rules for threat detection
- Provided Saved Searches for Lookup
- Provided dashboards for visualization

App Installation & Configuration

Pre-Requisites

- Google SecOps
- GreyNoise Intelligence platform (API Key).
- GCP Project with the below required permissions:
    - GCP user and project service account should have below permissions. Follow these steps to assign these permissions to the service account used for accessing other GCP services within your project.
        - Cloud Scheduler Job Runner
        - Secret Manager Secret Access
        - Storage Admin
        - Chronicle API Editor
        - Role Viewer
        - Cloud Run Invoker
- GCP Services
    - Before using the below services, ensure the Google APIs are enabled in your GCP project
        - Cloud Run Function (4-core CPU or higher is recommended)
        - GCS bucket
        - Secret Manager
        - Cloud Scheduler
- Access to Google SecOps Rules & Detection section.
- Access to Google SecOps Dashboard section.

Creating zip of the cloud function

- You can download the zip directly from the Drive.
  OR
- Create a zip file with the contents of the following files:
  1. Download the common directory from Git repository.
  2. Download the contents of the GreyNoise ingestion script. (Link will be added here once it will be submitted)
  3. Create a zip in the following structure.

  ❖ Example:

```
├── README.md
├── common
│   ├── __init__.py
│   ├── auth.py
│   ├── auth_test.py
│   ├── env_constants.py
│   ├── ingest.py
│   ├── ingest_test.py
│   ├── ingest_v1.py
│   ├── ingest_v1_test.py
│   ├── status.py
│   ├── utils.py
│   └── utils_test.py
├── constant.py
├── exception_handler.py
├── main.py
├── main_test.py
└── requirements.txt
└── utility.py
└── utility_test.py
└── greynoise_client.py
└── greynoise_client_test.py
```

Automatic deployment of the required resources

This section explains the use of the provided bash script to automate the deployment of the GreyNoise Intelligence ingestion function and related resources within your GCP project.

Prerequisites For Script Execution :

- Access to a GCP project with sufficient permissions to enable APIs, create service accounts, IAM bindings, GCS buckets, Cloud Functions, Cloud Scheduler jobs, and Secret Manager secrets.
- The Cloud Function source code packaged as a ZIP file in this format, available on your local machine or accessible within the Cloud Shell environment.

Execution Steps :

1. **Open Google Cloud Shell :**
    - Navigate to the Google Cloud Console (console.cloud.google.com).
    - Click the "**Activate Cloud Shell**" in the upper right corner of the console. A terminal window will open within your browser.
2. **Create the Script File :**
    - Choose a name for the script, for example, `greynoise_deploy.sh`
    - Use a text editor like `nano` to create the file in your Cloud Shell home directory: `nano greynoise_deploy.sh`
3. **Paste the Script Content :**
    - Copy the entire content of the bash script provided from github repo.
    - Paste the script content into the `nano` editor. In Cloud Shell, you can typically right-click and select "Paste" or use `Ctrl+Shift+V.`
4. **Save and Exit :**
    - Press `Ctrl+X` to exit `nano.`
    - When prompted to save, press `Y`.
    - Press `Enter` to confirm the filename (`greynoise_deploy.sh`)
5. **Upload Your Cloud Function ZIP (if not already in Cloud Shell) :**
    - If your function ZIP file is on your local machine, you need to upload it to Cloud Shell.

- Click the three-dot menu in the Cloud Shell terminal window.
- Select "Upload" and choose your ZIP file. It will be uploaded to your Cloud Shell home directory. Note the filename.

6. **Make the Script Executable :**
    - Grant execute permissions to the script file: `chmod +x greynoise_deploy.sh`
7. **Run the Deployment Script :**
    - Execute the script: `./greynoise_deploy.sh`
8. **Follow the Prompts :**
    - The script will prompt you to enter various configuration details, such as:
        - GCP Project ID & Region
        - Local path to your Cloud Function ZIP file (e.g.,`/home/username/your-function.zip`)
        - Chronicle Customer ID
        - Chronicle Region
        - GreyNoise API Key Value (this will be stored securely in Secret Manager)
        - Other optional environment variables.
    - Provide the requested information at each prompt. Required fields must be filled. Optional fields can be left blank to use defaults where applicable.
9. **Monitor the Output :**
    - The script will display progress messages, indicating which steps are being performed (e.g., enabling APIs, creating resources, deploying the function).
    - Error messages will be shown in red. If an error occurs, the script is designed to stop. Review the error message to troubleshoot.
10. **Post-Deployment :**
    - Once the script completes successfully, verify the resources in the GCP Console:
        - Check Cloud Functions to see your deployed function.
        - Check Cloud Scheduler to see the scheduled job.
        - Check GCS for the bucket and upload ZIP.
        - Check the Secret Manager for the GreyNoise API Key secret.
        - Review Cloud Logging for any function execution logs or errors.

Manual deployment of the required resources

Enabling required APIs

Ensure the following Google APIs are enabled in your GCP project (via **APIs & Services** → **Library**).

Service	**APIs needs to enable**
Google SecOps	• chronicle.googleapis.com
Cloud Functions	• cloudfunctions.googleapis.com • run.googleapis.com • cloudbuild.googleapis.com • artifactregistry.googleapis.com • logging.googleapis.com
Cloud Scheduler	• cloudscheduler.googleapis.com • pubsub.googleapis.com
Cloud Storage (Bucket)	• storage-component.googleapis.com
Secret Manager	• secretmanager.googleapis.com

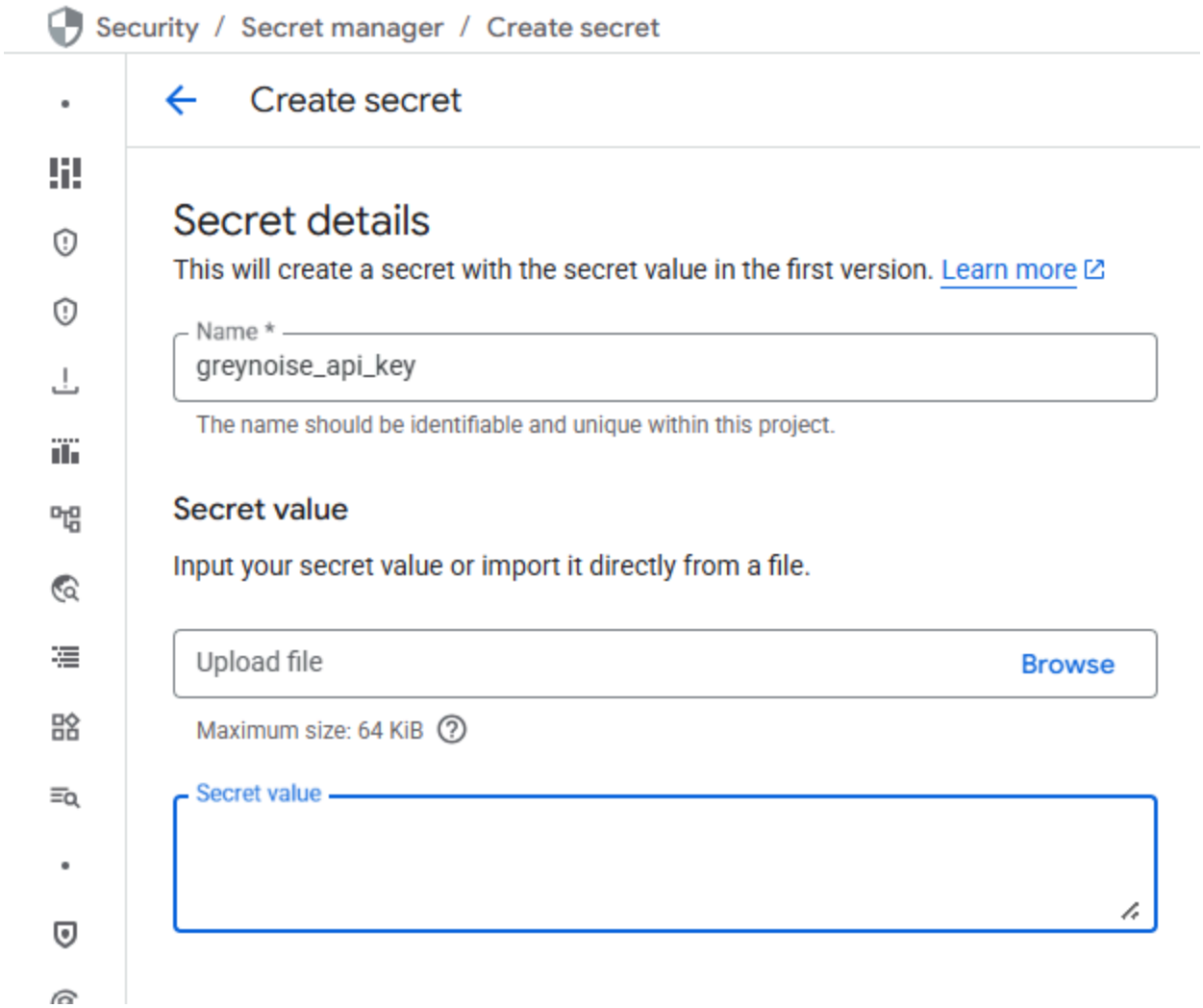
Using Secrets

- Environment variables marked as secret must be configured as secrets on Google Secret Manager. [REF]
- Once the secrets are created on Secret Manager, use the secret's resource ID as the value for environment variables.

  For example:
  CHRONICLE_SERVICE_ACCOUNT:
  projects/{project_id}/secrets/{secret_id}/versions/{version_id}

Add the GreyNoise API Key in Secret Manager

1. Log in to the "https://console.cloud.google.com/" using valid credentials.
2. Navigate to '**Secret Manager**'.
3. Click on '**Create Secret**'.
4. Provide the name for the secret in the `Name` field.
5. Provide the value for the secret in the `Secret Value` field.
6. Keep the other configurations as default, Click on the '**Create Secret**' button.



Refer to this page for more information about how to create secrets.

Once the secrets are created on Secret Manager, use the secret's resource id as the value for environment variables. For example

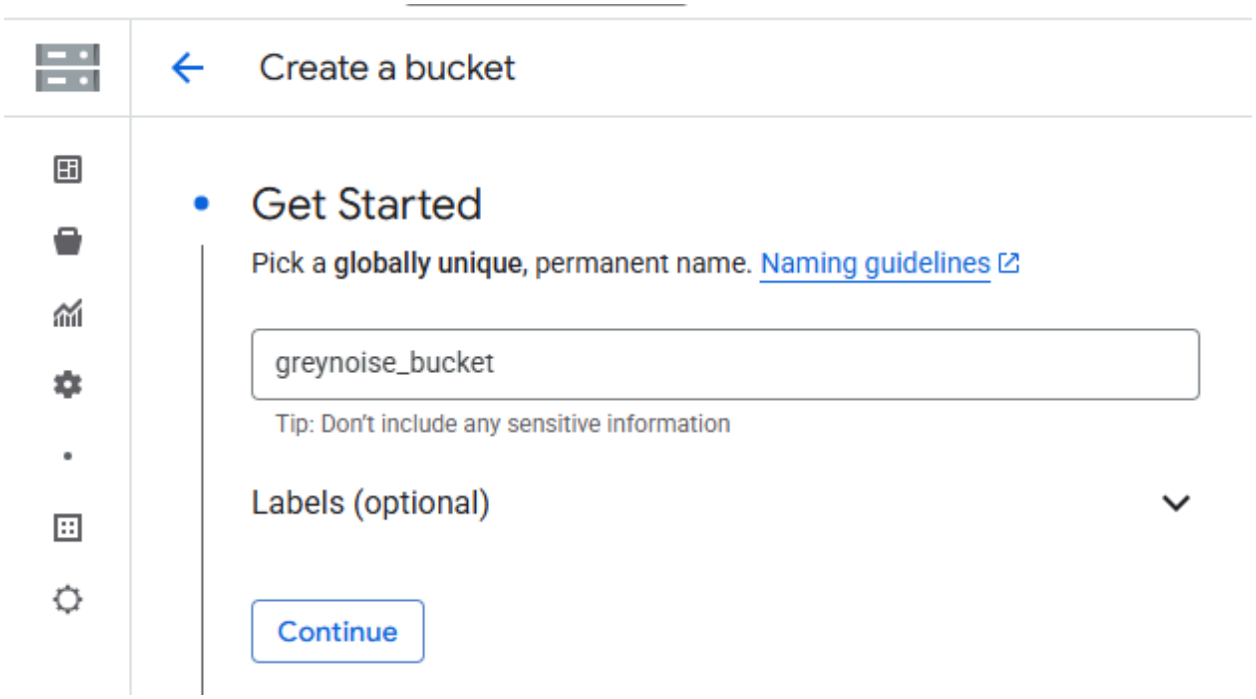
SECRET_KEY: projects/{project_id}/secrets/{secret_id}/versions/{version_id}

Note : Ensure that the resource ID format matches the following structure: projects/{project_id}/secrets/{secret_id}/versions/{version_id}. You can copy the resource name from the secret version details to obtain this value.

Create a GCP Bucket

1. Log in to the "https://console.cloud.google.com/" using valid credentials.
2. Navigate to Buckets in GCP.
3. Click on the Create button.
4. Enter the name of the bucket.
5. Users can select the region and modify the optional parameters if required and then click on the Create button.

Copy the bucket name and provide it in the **GCP_BUCKET_NAME** environment variable.



Cloud Function Deployment

Command based deployment

1. Navigate to the bucket and open the bucket created for the GreyNoise in these steps. Upload the created cloud function zip file in the bucket.
2. Click Activate Cloud Shell at the top right corner of the Google Cloud console.
3. Modify the below command based on your value and run in the terminal.

Command Format :

```
gcloud functions deploy CLOUD_FUNCTION_NAME --set-env-vars
"ENV_NAME1=ENV_VALUE1,ENV_NAME2=ENV_VALUE2,ENV_NAME3=ENV_VALUE3" --gen2 --runtime=python312 --region=REGION
--source=SOURCE_OF_FUNCTION  --entry-point=main
--service-account=SERVICE_ACCOUNT_EMAIL --trigger-http
--no-allow-unauthenticated --memory=8GiB --timeout=3600s
```

- **CLOUD_FUNCTION_NAME**: Unique name of the cloud function.
- **REGION**: A region for your cloud function. (Ex : us-central1, us-west1, etc.)
- **SOURCE_OF_FUNCTION**: gsutil URI of the cloud function zip in cloud storage. (e.g. gs://gti_test_bucket/gti_test.zip) where the gti_test_bucket is the name of the created bucket and gti_test.zip is the cloud function zip file.
- **SERVICE_ACCOUNT_EMAIL**: Email of the created service account of the project. Make sure the selected Service account must have a required permission. Update Service Account Permission following these steps.
- **ENV_NAME1=ENV_VALUE1**: Name and value of the environment variable to be created. Environment variables

Note:

1. When deploying a Cloud Function, ensure that the **--timeout** parameter in the deployment command matches the frequency specified in the Cloud Scheduler **--schedule** parameter. Aligning these values prevents overlapping executions, which could lead to data duplication.
    - **For example**, if you set --timeout=3600s when deploying the Cloud Function, configure the Cloud Scheduler with: --schedule="*/60 * * * *".

- This ensures that each scheduled run starts only after the previous execution has completed.

Example Command,

```
gcloud functions deploy funcusingcmd --set-env-vars
"CHRONICLE_CUSTOMER_ID=ed19f037-2354-43df-bfbf-350362b45844,CHRONICLE
_PROJECT_NUMBER=2134567,CHRONICLE_REGION=us,GCP_BUCKET_NAME=gti_tes
t_bucket,GREYNOISE_API_KEY=projects/1234567890/secrets/gn_api_key/versions/1
," --gen2 --runtime=python312 --region=us-central1
--source=gs://gn_test_bucket/greynoise_test.zip  --entry-point=main
--service-account=1234567890-compute@developer.gserviceaccount.com
--trigger-http --no-allow-unauthenticated --memory=8GiB --timeout=3600s
```

Configure Scheduler

Command based deployment

1. Click Activate Cloud Shell at the top right corner of the Google Cloud console.
2. Modify the below command based on your value and run in the terminal.

Command Format :

```
gcloud scheduler jobs create http SCHEDULER_NAME --schedule="CRON_TIME"
--uri="CLOUD_FUNCTION_URL" --http-method=POST --attempt-deadline=30m
--oidc-service-account-email=SERVICE_ACCOUNT_EMAIL --location=LOCATION
--time-zone=TIME_ZONE --project=CHRONICLE_PROJECT_NUMBER
```

- **SCHEDULER_NAME**: Unique name of the cloud scheduler.
- **CRON_TIME**: Cron time format for the scheduler to run in every interval. (eg. */60 * * * *)
- **CLOUD_FUNCTION_URL**: URL of the created cloud function. Navigate to create cloud function details.
- **SERVICE_ACCOUNT_EMAIL**: Email of the created service account of the project. Make sure the selected Service account must have a required Permission. Update Service Account Permission following these steps.
- **LOCATION**: A region for your connector. (Ex : us-central1, us-west1, etc)
- **TIME_ZONE**: The time zone of your region. (Ex : UTC)

Note:

1. When deploying a Cloud Function, ensure that the **--timeout** parameter in the deployment command matches the frequency specified in the Cloud Scheduler **--schedule** parameter. Aligning these values prevents overlapping executions, which could lead to data duplication.
    - **For example**, if you set --timeout=3600s when deploying the Cloud Function, configure the Cloud Scheduler with: --schedule="*/60 * * * *".
    - This ensures that each scheduled run starts only after the previous execution has completed.

Example Command,

```
gcloud scheduler jobs create http funcusingcmdschedular --schedule="*/60 * * * *" --uri="https://us-central1-test.cloudfunctions.net/funcusingcmd" --attempt-deadline=30m --oidc-service-account-email=1234567890-compute@developer.gserviceaccount.com --location=us-central1 --time-zone=UTC
```

View Events in Google SecOps

---

1. Log in to Google SecOps:
    a. Open a web browser and navigate to the Google SecOps instance URL. For example: https://test.backstory.chronicle.security/
    b. Replace test with your actual Google SecOps instance name.
2. Access SIEM Search:
    a. From the top left corner of the Google SecOps console, select the "Investigation" option.
    b. Within the Investigation section, choose "SIEM Search".
3. Filter Events by Log Type:
    a. In the SIEM Search interface, locate the "UDM Search" section.
    b. Apply a filter for the metadata field "log_type". Set the filter value to metadata.log_type="**GREYNOISE**".
4. View GreyNoise Intelligence Entities:
    a. The SIEM Search results will display GreyNoise Intelligence entities within the "Results" section.

Update Service Account Permission

1. Open **GCP Console**, Then go to **IAM**.
2. In View By **Main Tab** > Click **GRANT ACCESS**.
3. Add Service Account name in **New Principals**. (Example : service_account_name.gserviceaccount.com)
4. In **Assign Role**, assign below roles to service accounts.
    a. Cloud Scheduler Job Runner
    b. Secret Manager Secret Access
    c. Storage Admin
    d. Chronicle API Editor
    e. Cloud Run Invoker
    f. Role Viewer
5. Click **Save**.

Environment Variables to be configured in Cloud Function

Environment variable	Description	Default value	Required	Secret Manager
CHRONICLE_CUSTOMER_ID	Google SecOps customer id. Navigate to settings in the Google SecOps console for the customer id.	-	Yes	No
CHRONICLE_REGION	A region where the Google SecOps instance is located.	us	No	No
CHRONICLE_PROJECT_NUMBER	Specifies the GCP project identifier associated with your Google SecOps environment.	-	Yes	No
GCP_BUCKET_NAME	Name of the created GCP bucket.	-	Yes	No
GREYNOISE_API_KEY	Copied resource name value of API KEY of GreyNoise from secret manager. Generate an API Key from the GreyNoise platform's API key section.	-	Yes	Yes
QUERY	A query to filter GreyNoise Indicators. More details. Examples are below: actor:Censys classification:benign	-	No	No

Detection Rules

Create Detection Rules

After collecting the data into Google SecOps in the form of entities, the user can see the specific entities from predefined saved searches.

Create & Enable Detection Rule

1. From Google SecOps SIEM, navigate to **Detection > Rules & Detections**.
2. Go to the **Rules Editor** Tab**,** click on the **NEW** button**.**
3. Copy and paste the Rule in editor.
4. Click on **SAVE NEW RULE.**
5. Click on 3 dots on the top right of the editor.
6. To generate alerts, Enable **Alerting**
7. Enable **Live Rule** to activate Detection Rule**.**

See Google SecOps: Detection Rules for more information.

Detection Rules

1. **GreyNoise Intelligence Brute Force Attack Detection**

   Description: Detects multiple blocked login attempts from IPs flagged by GreyNoise threat intelligence.

   Rule:

```
rule greynoise_intelligence_brute_force_attack_detection {

  meta:
    author = "GreyNoise Intelligence"
    rule_name = "GreyNoise Intelligence Brute Force Attack Detection"
    description = "Detects multiple blocked login attempts from IPs
flagged by GreyNoise threat intelligence."
    severity = "High"
    priority = "High"
    mitre_attack_tactic = "Credential Access"
    mitre_attack_technique = "T1110 - Brute Force"
    tags = "cloud security, threat intelligence"
```

```
  events:
    $login.metadata.event_type = "USER_LOGIN"
    $login.security_result.action = "BLOCK"
    (
        $login.src.ip = $correlation_ip or
        $login.principal.ip = $correlation_ip
    )

    // GreyNoise entity match
    $greynoise.graph.metadata.event_metadata.base_labels.log_types =
"GREYNOISE"
    $greynoise.graph.metadata.product_name = "GreyNoise Intelligence"
    $greynoise.graph.metadata.entity_type = "IP_ADDRESS"
    $greynoise.graph.entity.ip = $correlation_ip

  match:
    $correlation_ip over 15m

  outcome:
    $principal_ip = array_distinct($login.principal.ip)
    $principal_hostname = array_distinct($login.principal.hostname)
    $principal_user_userid = array_distinct($login.principal.user.userid)
    $principal_mac = array_distinct($login.principal.mac)
    $principal_ip_count = count($login.principal.ip)

    $source_ip = array_distinct($login.src.ip)
    $source_hostname = array_distinct($login.src.hostname)
    $source_user_userid = array_distinct($login.src.user.userid)
    $source_mac = array_distinct($login.src.mac)
    $source_ip_count = count($login.src.ip)

    $target_ip = array_distinct($login.target.ip)
    $target_hostname = array_distinct($login.target.hostname)
    $target_user_userid = array_distinct($login.target.user.userid)
    $target_mac = array_distinct($login.target.mac)

  condition:
    $login and $greynoise and ($principal_ip_count>5 or
$source_ip_count>5)
}
```

2. GreyNoise Intelligence Inbound Network Traffic with ASN Context

Description: Detects allowed inbound firewall connections from malicious IPs flagged by GreyNoise with correlated ASN context.

Rule:

```
rule greynoise_intelligence_inbound_network_traffic_with_asn_context {

  meta:
    author = "GreyNoise Intelligence"
    rule_name = "GreyNoise Intelligence Inbound Network Traffic with ASN
Context"
    description = "Detects allowed inbound firewall connections from
malicious IPs flagged by GreyNoise with correlated ASN context."
    severity = "High"
    priority = "Critical"
    mitre_attack_tactic = "Initial Access"
    tags = "cloud security, threat intelligence"

  events:
    $network.metadata.event_type = "NETWORK_CONNECTION"
    $network.security_result.action = "ALLOW"
    re.regex($network.metadata.product_name, `(?i).*firewall.*`)
    (
        $network.src.ip = $correlation_ip or
        $network.principal.ip = $correlation_ip
    )

    // GreyNoise entity match
    $greynoise.graph.metadata.event_metadata.base_labels.log_types =
"GREYNOISE"
    $greynoise.graph.metadata.product_name = "GreyNoise Intelligence"
    $greynoise.graph.metadata.entity_type = "IP_ADDRESS"
    $greynoise.graph.metadata.threat.threat_verdict = "MALICIOUS"
    $greynoise.graph.entity.ip = $correlation_ip
    (
        re.capture($network.principal.ip_geo_artifact.network.asn,
`(\d+)`) =
re.capture($greynoise.graph.entity.ip_geo_artifact.network.asn, `(\d+)`)
        or
        re.capture($network.src.ip_geo_artifact.network.asn, `(\d+)`) =
re.capture($greynoise.graph.entity.ip_geo_artifact.network.asn, `(\d+)`)
    )
```

```
  match:
    $correlation_ip over 1h

  outcome:
    $principal_ip = array_distinct($network.principal.ip)
    $principal_hostname = array_distinct($network.principal.hostname)
    $principal_user_userid =
array_distinct($network.principal.user.userid)
    $principal_mac = array_distinct($network.principal.mac)

    $source_ip = array_distinct($network.src.ip)
    $source_hostname = array_distinct($network.src.hostname)
    $source_user_userid = array_distinct($network.src.user.userid)
    $source_mac = array_distinct($network.src.mac)

    $target_ip = array_distinct($network.target.ip)
    $target_hostname = array_distinct($network.target.hostname)
    $target_user_userid = array_distinct($network.target.user.userid)
    $target_mac = array_distinct($network.target.mac)

  condition:
    $network and $greynoise
}
```

3. GreyNoise Intelligence IP Match

Description: Detects events where source or principal IP matches a malicious or suspicious IP in GreyNoise intelligence.

Rule:

```
rule greynoise_intelligence_ip_match {

  meta:
    author = "GreyNoise Intelligence"
    rule_name = "GreyNoise Intelligence IP Match"
    description = "Detects events where source or principal IP matches a
malicious or suspicious IP in GreyNoise intelligence."
    severity = "High"
    priority = "Medium"
    tags = "cloud security, threat intelligence"

  events:
```

```
    $network.metadata.event_type != "GENERIC_EVENT"
    (
        $network.src.ip = $correlation_ip or
        $network.principal.ip = $correlation_ip
    )

    // GreyNoise entity match
    $greynoise.graph.metadata.event_metadata.base_labels.log_types =
"GREYNOISE"
    $greynoise.graph.metadata.product_name = "GreyNoise Intelligence"
    $greynoise.graph.metadata.entity_type = "IP_ADDRESS"
    (
        $greynoise.graph.metadata.threat.threat_verdict = "MALICIOUS"
        or
        $greynoise.graph.metadata.threat.threat_verdict = "SUSPICIOUS"
    )
    $greynoise.graph.entity.ip = $correlation_ip

  match:
    $correlation_ip over 1h

  outcome:
    $principal_ip = array_distinct($network.principal.ip)
    $principal_hostname = array_distinct($network.principal.hostname)
    $principal_user_userid =
array_distinct($network.principal.user.userid)
    $principal_mac = array_distinct($network.principal.mac)

    $source_ip = array_distinct($network.src.ip)
    $source_hostname = array_distinct($network.src.hostname)
    $source_user_userid = array_distinct($network.src.user.userid)
    $source_mac = array_distinct($network.src.mac)

    $target_ip = array_distinct($network.target.ip)
    $target_hostname = array_distinct($network.target.hostname)
    $target_user_userid = array_distinct($network.target.user.userid)
    $target_mac = array_distinct($network.target.mac)

  condition:
    $network and $greynoise
}
```

Saved Searches

Create Saved Searches

After collecting the data into Google SecOps in the form of entities, the user can see the specific entities from predefined saved searches.

Create & Execute a Saved Search

1. From Google SecOps SIEM, navigate to **Investigation > SIEM Search**.
2. Go to **Search Manager,** click on **+** icon**.**
3. Copy and paste the Search Query in **UDM SEARCH,** Title in **Title** and Description in **Description** from the Search Queries listed below.
4. Click on **SAVE EDITS.**
5. Click on **LOAD SEARCH**.
6. Select **Time Range.**
7. Click on **Run Search** to execute the search query.

See Google SecOps: Saved Searches for more information.

Saved Searches

1. **GreyNoise - IP Risk & Vulnerability Details**

   Description: Comprehensive security view of GreyNoise indicators including classification, anonymization signals, associated CVEs, and activity timeline.

   Query:

```
graph.metadata.vendor_name = "GreyNoise Intelligence"
graph.metadata.product_name = "GreyNoise Intelligence"
$ip = graph.entity.ip
$classification = graph.metadata.threat.threat_verdict
$bot = strings.to_upper(graph.metadata.threat.detection_fields["bot"])
$tor = strings.to_upper(graph.metadata.threat.detection_fields["tor"])
$spoofable =
strings.to_upper(graph.metadata.threat.detection_fields["spoofable"])
```

```
$vpn_Service = if(graph.metadata.threat.detection_fields["vpn"]= "true",
graph.metadata.threat.detection_fields["vpn_service"], "-")
$time =
timestamp.get_timestamp(graph.metadata.collected_timestamp.seconds,
"%Y-%m-%d %H:%M:%S")
$first_seen_date =
timestamp.get_timestamp(graph.metadata.threat.first_discovered_time.seco
nds, "%Y-%m-%d")
$category = graph.additional.fields["network_category"]
$category = /.*/
graph.entity.asset.vulnerabilities.cve_id = /.*/
match:
    $ip, $classification, $category, $bot, $tor, $vpn_Service,
$spoofable, $first_seen_date
outcome:
    $all_cves =
re.replace((arrays.join_string(array_distinct(if(graph.entity.asset.vuln
erabilities.cve_id!="", graph.entity.asset.vulnerabilities.cve_id,
"-")),", ")), "(, -)|(-, )","")
    $last_seen = window.last(graph.metadata.collected_timestamp.seconds,
$time)
order:
    $last_seen desc
limit:
    10000
```

2. GreyNoise - Indicator Context Summary

Description: Provides actor attribution, geographic details, organization info, and tags for quick indicator triage and contextual analysis.

Query:

```
graph.metadata.vendor_name = "GreyNoise Intelligence"
graph.metadata.product_name = "GreyNoise Intelligence"
$ip = graph.entity.ip
$classification = graph.metadata.threat.threat_verdict
```

```
$actor = graph.additional.fields["internet_scanner_actor"]
$country = graph.entity.ip_geo_artifact.location.country_or_region
$organization = graph.entity.ip_geo_artifact.network.organization_name
$category = graph.additional.fields["network_category"]
$description = graph.metadata.description
$summary = graph.metadata.threat.summary
$time =
timestamp.get_timestamp(graph.metadata.collected_timestamp.seconds,
"%Y-%m-%d %H:%M:%S")
$first_seen_date =
timestamp.get_timestamp(graph.metadata.threat.first_discovered_time.seco
nds, "%Y-%m-%d")
match:
    $ip, $classification, $description, $summary, $actor, $country,
$organization, $category, $first_seen_date
outcome:
    $last_seen = window.last(graph.metadata.collected_timestamp.seconds,
$time)
    $primary_tag =
re.replace((arrays.join_string(array_distinct(if(graph.entity.artifact.t
ags!="", graph.entity.artifact.tags, "-")),", ")), "(, -)|(-, )","")
order:
    $last_seen desc
limit:
    10000
```

3. GreyNoise - High Risk Indicators

Description: Lookup filter to quickly identify indicators classified as `MALICIOUS` or `SUSPICIOUS` that require immediate investigation.

Query:

```
graph.metadata.product_name = "GreyNoise Intelligence"
graph.metadata.vendor_name = "GreyNoise Intelligence"
$classification = graph.metadata.threat.threat_verdict
```

```
$classification = "MALICIOUS" or $classification = "SUSPICIOUS"
```

4. **GreyNoise - All Indicator Lookup**

   Description: Browse all ingested GreyNoise indicators with business service context. Use as a starting point for ad-hoc investigations.

   Query:

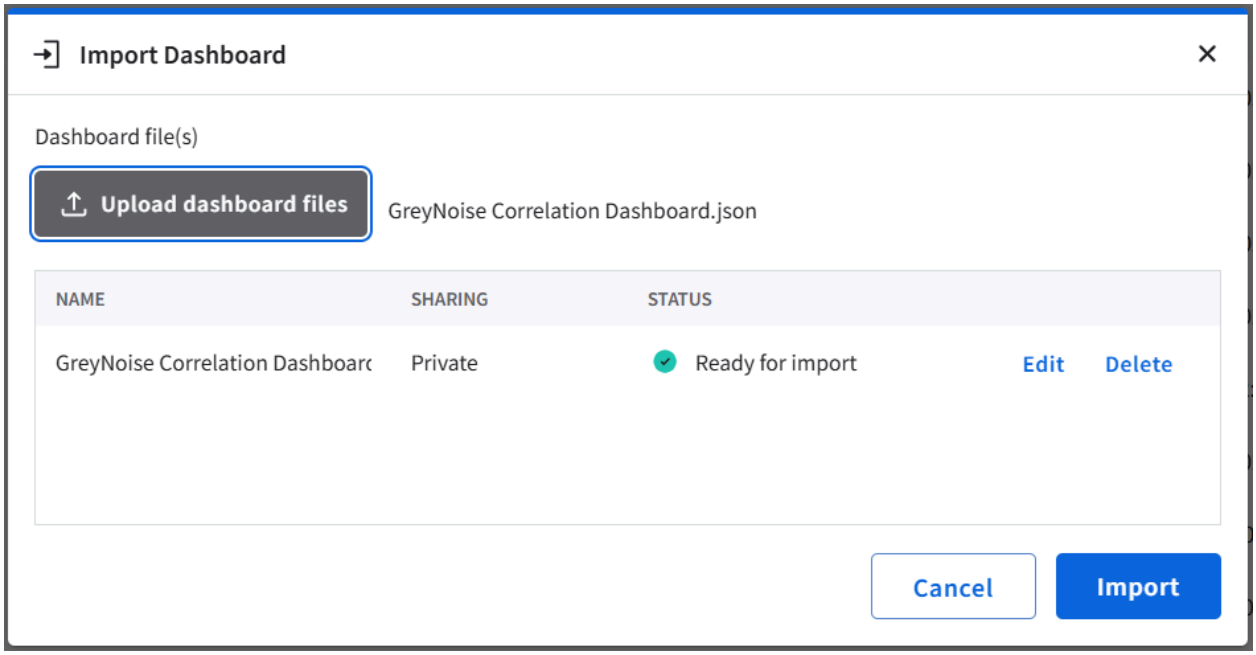
```
graph.metadata.vendor_name = "GreyNoise Intelligence"
graph.metadata.product_name  = "GreyNoise Intelligence"
graph.entity.ip = /.*/
graph.additional.fields["business_service_found"] = /.*/ nocase
```

Dashboard Configuration

Import GreyNoise Dashboard into Google SecOps SIEM

Complete the following steps to import a dashboard:

1. Log in to your Google SecOps instance.
2. In the navigation bar, click **Dashboards**.
3. Click **New dashboard** and then select **Import from JSON**. The **Import dashboard** confirmation dialog appears.
4. Click the **Upload dashboard files** button. The **Select file** dialog appears. Select the dashboard json file. The selected dashboard file will appear in the table.



5. Click **Import** to continue importing the dashboard to a personal or shared dashboard.

Dashboards

Indicator Dashboard

The Indicator Dashboard provides detailed information about Indicators fetched from GreyNoise.

Filters Description

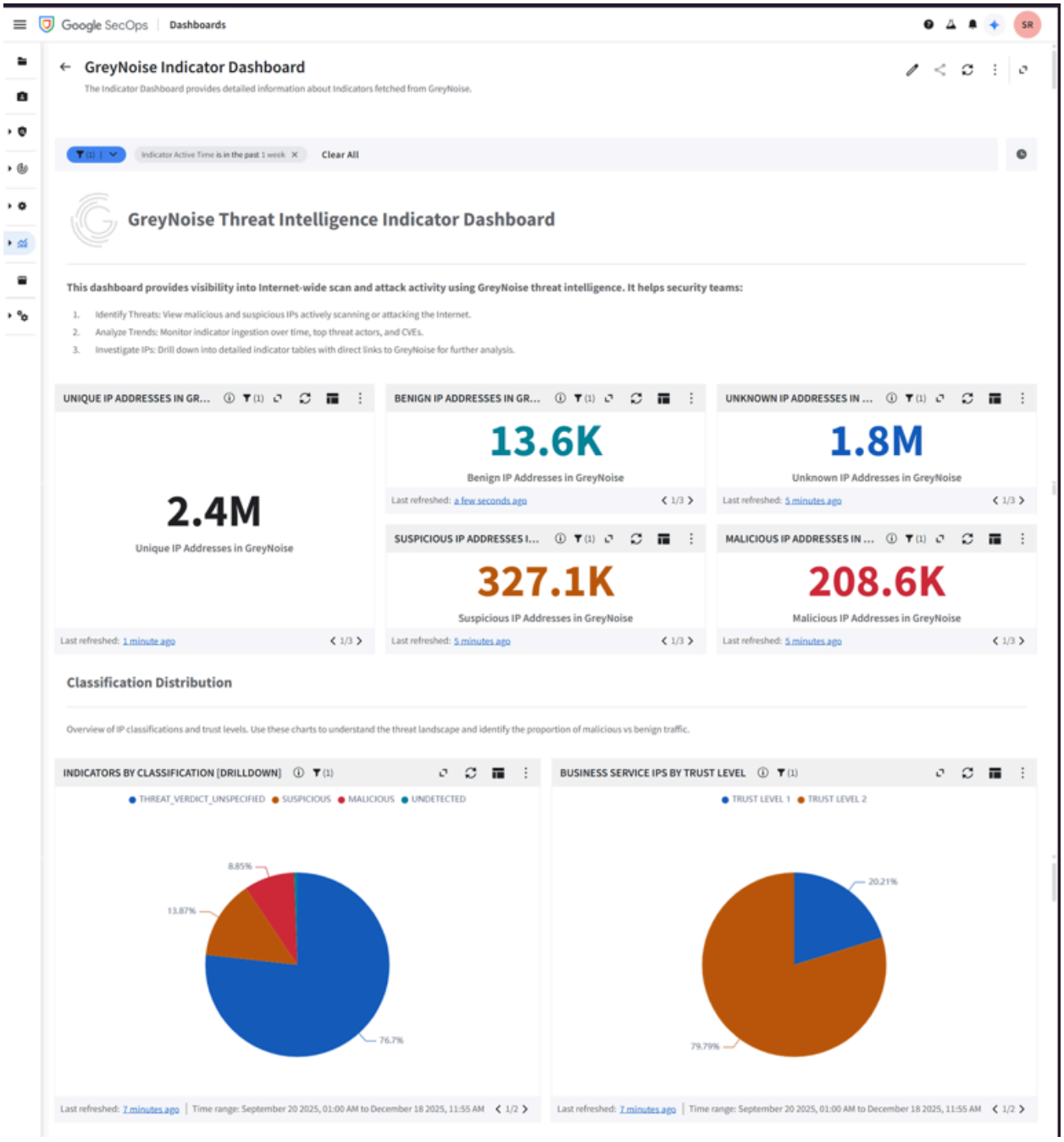
1. **Indicator Active Time**
    a. This filter updated the panel based on the time range selected in it. **Default**: Last 7 days.
2. **Classification Filter**
    a. Filters the Panels according to the selected Classification type. i.e. SUSPICIOUS, MALICIOUS etc. **Default:** All.
3. **Country Filter**
    a. Filters the Panels according to the entered Country. **Default:** All.

Panels Description

1. **Unique IP Addresses in GreyNoise**
    a. Count of distinct IPs in GreyNoise threat intelligence.
2. **Malicious IP Addresses in GreyNoise**
    a. Count of IPs identified as Malicious in GreyNoise threat intelligence.
3. **Suspicious IP Addresses in GreyNoise**
    a. Count of IPs identified as Suspicious in GreyNoise threat intelligence.
4. **Benign IP Addresses in GreyNoise**
    a. Count of IPs identified as Benign in GreyNoise threat intelligence.
5. **Unknown IP Addresses in GreyNoise**
    a. Count of IPs with no definitive classification in GreyNoise threat intelligence.
6. **Indicators by Classification**
    a. Distribution of indicators by classification (malicious, suspicious, benign, unknown).

7. **Business Service IP Intelligence**
    a. List of IP addresses identified as Business Services, including service category, trust level, and descriptive context.
8. **Business Service IPs by Trust Level**
    a. Business service IPs segmented by trust levels
9. **Top 10 Organizations**
    a. Highlights organizations (ISPs/Hosting Providers) most frequently associated with observed indicators.
10. **Top 10 Actors**
    a. Identifies the most frequently observed threat actors across all indicators.
11. **Top 10 Tags**
    a. Lists the most common tags associated with observed indicators.
12. **Top 10 ASN**
    a. Displays the top autonomous system numbers related to indicator activity.
13. **Top 10 Categories**
    a. Displays the most prevalent network categories.
14. **Top 10 OS**
    a. Shows the top operating systems identified from Indicator metadata.
15. **Top 10 Business Service Categories**
    a. Highlights business service categories most often linked to indicators.
16. **Top 10 Source Countries**
    a. Shows the top Source Countries linked to Indicators.
17. **IPs Over Time (Ingested Indicators Trend)**
    a. Trend of ingested indicators over time by classification.
18. **Indicator Details (All Ingested IPs)**
    a. Recently ingested IP indicators with details.
19. **CVE Distribution**
    a. CVEs associated with observed IPs.
20. **Manually Queried Indicator**
    a. Displays indicators that were manually investigated by users.

Screenshots


Google SecOps | Dashboards
SR
GreyNoise Indicator Dashboard
The Indicator Dashboard provides detailed information about Indicators fetched from GreyNoise.
Indicator Active Time is in the past 1 week
Clear All
GreyNoise Threat Intelligence Indicator Dashboard
This dashboard provides visibility into Internet-wide scan and attack activity using GreyNoise threat intelligence. It helps security teams:
1. Identify Threats: View malicious and suspicious IPs actively scanning or attacking the Internet.
2. Analyze Trends: Monitor indicator ingestion over time, top threat actors, and CVEs.
3. Investigate IPs: Drill down into detailed indicator tables with direct links to GreyNoise for further analysis.
UNIQUE IP ADDRESSES IN GR...
2.4M
Unique IP Addresses in GreyNoise
Last refreshed: 1 minute ago
BENIGN IP ADDRESSES IN GR...
13.6K
Benign IP Addresses in GreyNoise
Last refreshed: a few seconds ago
UNKNOWN IP ADDRESSES IN ...
1.8M
Unknown IP Addresses in GreyNoise
Last refreshed: 5 minutes ago
SUSPICIOUS IP ADDRESSES I...
327.1K
Suspicious IP Addresses in GreyNoise
Last refreshed: 5 minutes ago
MALICIOUS IP ADDRESSES IN ...
208.6K
Malicious IP Addresses in GreyNoise
Last refreshed: 5 minutes ago
Classification Distribution
Overview of IP classifications and trust levels. Use these charts to understand the threat landscape and identify the proportion of malicious vs benign traffic.
INDICATORS BY CLASSIFICATION [DRILLDOWN]
THREAT_VERDICT_UNSPECIFIED
SUSPICIOUS
MALICIOUS
UNDETECTED
8.85%
13.87%
76.7%
Last refreshed: 7 minutes ago | Time range: September 20 2025, 01:00 AM to December 18 2025, 11:55 AM
BUSINESS SERVICE IPS BY TRUST LEVEL
TRUST LEVEL 1
TRUST LEVEL 2
20.21%
79.79%
Last refreshed: 7 minutes ago | Time range: September 20 2025, 01:00 AM to December 18 2025, 11:55 AM

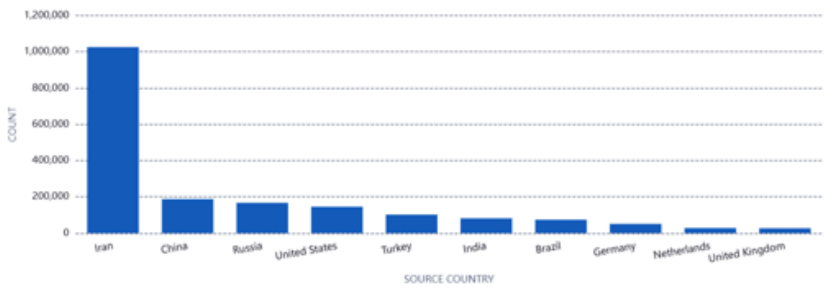
Top 10 Analytics

Key insights into the most prevalent entities in GreyNoise data. Analyze top organizations, threat actors, behavioral tags, ASNs, operating systems, and geographic locations.

TOP 10 ASNS (1)

Last refreshed: 5 minutes ago | Time range: September 20 2025, 01:00 AM to December 18 2025, 11:55 AM | Showing results based on 10 rows

TOP 10 SOURCE COUNTRIES [DRILLDOWN] (1)

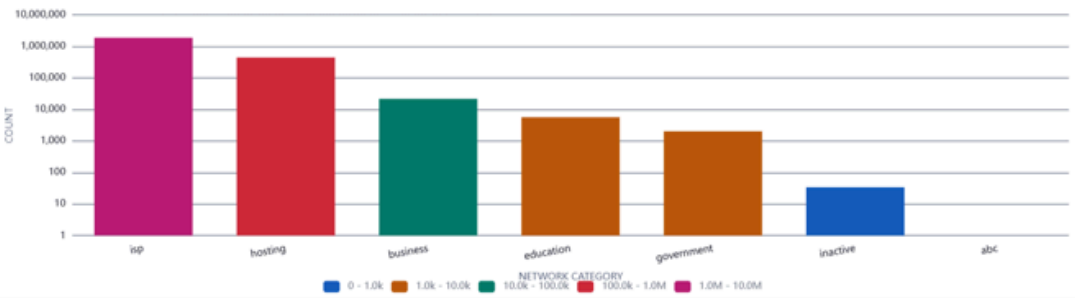


Last refreshed: 5 minutes ago | Time range: September 20 2025, 01:00 AM to December 18 2025, 11:55 AM | Showing results based on 10 rows

TOP 10 OS (1)

Last refreshed: 7 minutes ago 1/3

TOP 10 CATEGORIES (1)

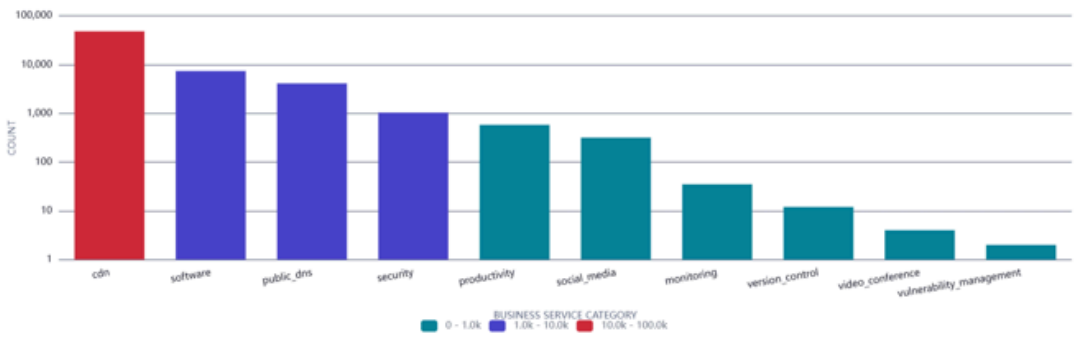


Last refreshed: 5 minutes ago | Time range: September 20 2025, 01:00 AM to December 18 2025, 11:55 AM | Showing results based on 7 rows

TOP 10 TAGS

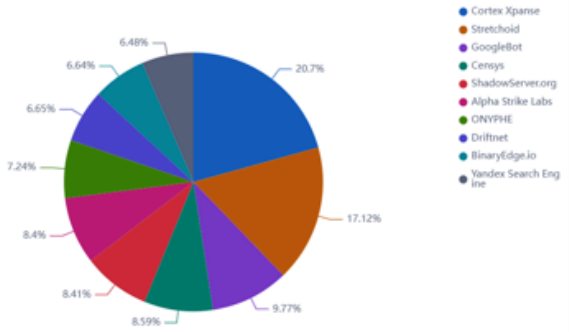
Last refreshed: 5 minutes ago | Time range: September 20 2025, 01:00 AM to December 18 2025, 11:56 AM | Showing results based on 10 rows

TOP 10 BUSINESS SERVICE CATEGORIES



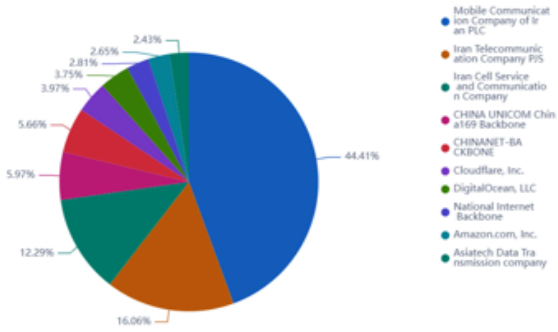
Last refreshed: 5 minutes ago | Time range: September 20 2025, 01:00 AM to December 18 2025, 11:55 AM | Showing results based on 10 rows

TOP 10 ACTORS



Last refreshed: 7 minutes ago | Time range: September 20 2025, 01:00 AM to December 18 2025, 11:55 AM 1/2

TOP 10 ORGANIZATIONS



Last refreshed: 7 minutes ago | Time range: September 20 2025, 01:00 AM to December 18 2025, 11:55 AM 1/2

Detailed Tables

Comprehensive tables for in-depth analysis. Drill down into individual IP indicators, CVE associations, and manually queried IPs.

MANUALLY QUERIED INDICATORS [DRILLDOWN] (1)

IP	CLASSIFICATI...	DESCRIPTION	GREYNOISE L...	IS BUSINESS ...	SOURCE COU...	ORGANIZATION	TRUST LEVEL	ACTOR	ASN	LAST SEEN
1.1.1.2	THREAT_VE...	Public DNS s...	https://viz.g...	TRUE	-	-	1	-	-	2025-12-09 1...
8.8.8.8	THREAT_VE...	Public DNS s...	https://viz.g...	TRUE	-	-	1	-	-	2025-12-09 1...
45.177.117.77	THREAT_VE...	Internet Sca...	https://viz.g...	FALSE	Brazil	Lima e Carv...	-	UNKNOWN	AS268991	2025-12-08 0...
103.83.189....	SUSPICIOUS	Internet Sca...	https://viz.g...	FALSE	Myanmar	IT Spectrum ...	-	UNKNOWN	AS136210	2025-12-08 0...
200.220.164....	THREAT_VE...	Internet Sca...	https://viz.g...	FALSE	Brazil	Paulo Dias d...	-	UNKNOWN	AS27712	2025-12-08 0...
5.122.134.241	THREAT_VE...	Internet Sca...	https://viz.g...	FALSE	Iran	Iran Cell Ser...	-	UNKNOWN	AS44244	2025-12-08 0...

Last refreshed: 5 minutes ago | Time range: September 20 2025, 01:00 AM to December 18 2025, 11:56 AM | Showing results based on 6 rows

BUSINESS SERVICE IP INTELLIGENCE [DRILLDOWN] (1)

IP	CATEGORY	NAME	TRUST LEVEL	SUMMARY	DESCRIPTION	REFERENCE	LAST SEEN
162.158.216.182	cdn	Cloudflare CDN	2	CDNs are geographi...	Cloudflare, Inc. is an...	https://www.cloudfl...	2025-12-12T09:11:06Z
172.64.192.184	cdn	Cloudflare CDN	2	CDNs are geographi...	Cloudflare, Inc. is an...	https://www.cloudfl...	2025-12-12T09:11:06Z
92.122.215.121	software	Akamai Technologie...	1	Trusted commercial...	Akamai is a global c...	https://www.akama...	2025-12-12T09:11:06Z
172.68.210.212	cdn	Cloudflare CDN	2	CDNs are geographi...	Cloudflare, Inc. is an...	https://www.cloudfl...	2025-12-12T09:11:06Z
162.159.104.15	cdn	Cloudflare CDN	2	CDNs are geographi...	Cloudflare, Inc. is an...	https://www.cloudfl...	2025-12-12T09:11:06Z
162.158.233.165	cdn	Cloudflare CDN	2	CDNs are geographi...	Cloudflare, Inc. is an...	https://www.cloudfl...	2025-12-12T09:11:06Z
74.125.47.147	public_dns	Google Public DNS	1	Public DNS services ...	Google's global do...	https://developers.g...	2025-12-12T09:11:06Z

Last refreshed: 5 minutes ago | Time range: September 20 2025, 01:00 AM to December 18 2025, 11:55 AM | Showing results based on 10,000 rows

INDICATOR DETAILS (ALL INGESTED IPS) [DRILLDOWN] (1)

IP	THREAT ...	IS BUSI...	TRUST L...	SUMMARY	GREYNO...	CVES	VPN SE...	SOURCE...	ORGANI...	ACTOR	ASN	BOT	TOR	L
183.78....	MALICI...	FALSE	-	Interne...	https://...	CVE-20...	-	India	XYZ	UNKNO...	AS23	FALSE	FALSE	
220.135....	MALICI...	FALSE	-	Interne...	https://...	CVE-20...	-	Taiwan	Data Co...	UNKNO...	AS3462	FALSE	FALSE	
183.103....	MALICI...	FALSE	-	Interne...	https://...	CVE-20...	-	South K...	Korea T...	UNKNO...	AS4766	FALSE	FALSE	
95.64.0.3	THREAT...	FALSE	-	Interne...	https://...	-	-	Iran	Mobile ...	UNKNO...	AS197207	FALSE	FALSE	
142.91....	THREAT...	FALSE	-	Interne...	https://...	-	-	Singapore	LEASE...	UNKNO...	AS59253	FALSE	FALSE	
117.236....	MALICI...	FALSE	-	Interne...	https://...	-	-	India	Nationa...	UNKNO...	AS9829	FALSE	FALSE	
5.123.1....	THREAT...	FALSE	-	Interne...	https://...	-	-	Iran	Iran Cel...	UNKNO...	AS44244	FALSE	FALSE	

Last refreshed: 5 minutes ago | Time range: September 20 2025, 01:00 AM to December 18 2025, 11:55 AM | Showing results based on 10,000 rows

CVE DISTRIBUTION (1)

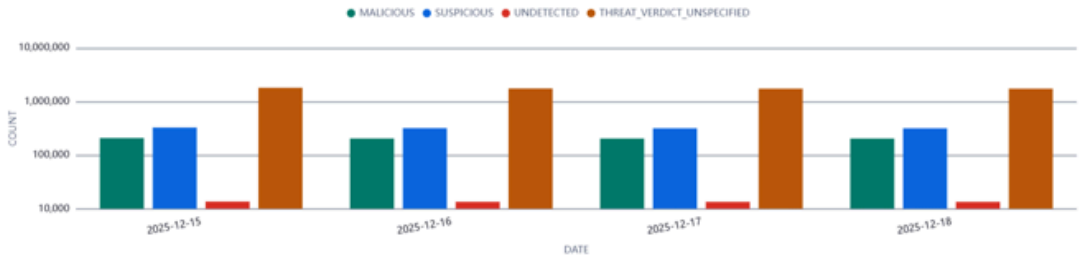
CVE	COUNT	AFFECTED IPs
CVE-2019-8950	27534	110.151.72.232, 85.130.134.26, 72.239.26.94, 114.33.157.233, 84.41.1....
CVE-2021-27144	10890	43.230.106.219, 101.51.251.106, 42.239.125.223, 5.58.206.180, 176.2....
CVE-2021-27172	10890	46.100.61.235, 182.52.100.190, 45.170.21.230, 91.231.202.67, 220.13....
CVE-2021-27145	10890	43.230.106.109, 68.221.121.191, 190.151.157.231, 102.212.41.88, 74....
CVE-2021-27163	10890	103.206.103.140, 115.210.230.185, 103.167.204.232, 154.178.202.51, ...
CVE-2021-27158	10890	36.32.200.14, 122.97.212.0, 118.233.66.253, 175.165.87.40, 156.215.2....
CVE-2021-27154	10890	122.97.212.134, 93.177.148.84, 51.9.59.244, 178.141.201.206, 122.97....

Last refreshed: 5 minutes ago | Time range: September 20 2025, 01:00 AM to December 18 2025, 11:55 AM | Showing results based on 1,849 rows

Trends

Monitor indicator ingestion patterns over time. Track changes in threat activity and identify anomalies or spikes in specific classifications.

IPS OVER TIME (INGESTED INDICATORS TREND) (1)



Last refreshed: 5 minutes ago | Time range: September 20 2025, 01:00 AM to December 18 2025, 11:56 AM | Showing results based on 16 rows

Correlation Dashboard

The Correlation dashboard is designed to identify relationships across indicators and events from other sources.

Filters Description

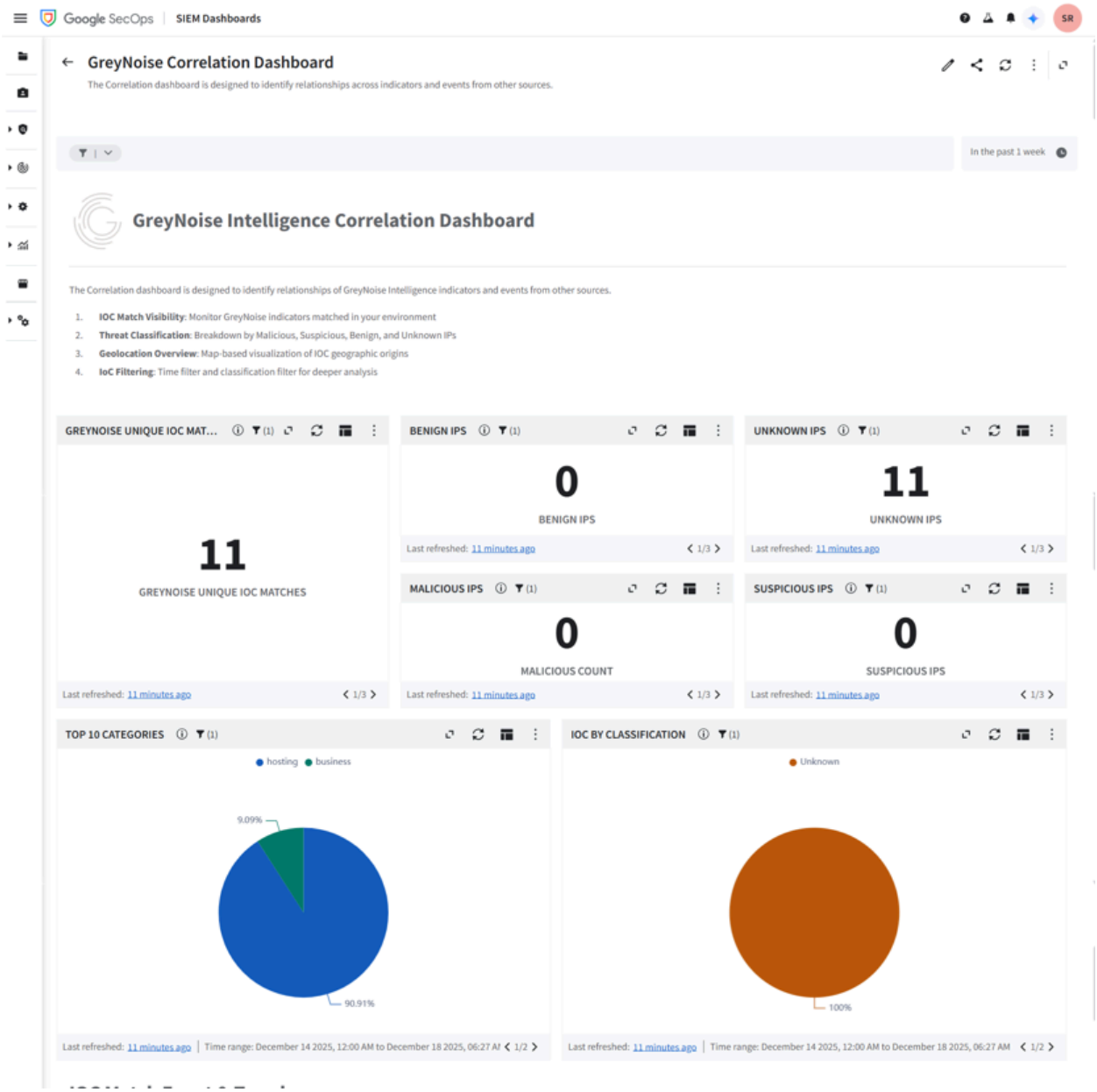
1. **Indicator Active Time Range**
    a. This filter updated the panel based on the time range selected in it. Default: Last 7 days.
2. **Classification Filter**
    b. Filters the Panels according to the selected Classification type. i.e. SUSPICIOUS, MALICIOUS etc. Default: All.

Panels Description

1. **GreyNoise Unique IOC Matches**
    a. Total count of GreyNoise Intelligence IOC matches detected in the environment.
2. **Event Matches In last 24 Hours**
    a. Count of GreyNoise Intelligence IOC matches detected in the environment in the last 24 hours.
3. **Malicious IPs**
    a. Count of IPs classified as malicious by GreyNoise Intelligence.
4. **Suspicious IPs**
    a. Count of IPs classified as suspicious by GreyNoise Intelligence.
5. **Benign IPs**
    a. Count of IPs classified as benign by GreyNoise Intelligence.
6. **Unknown IPs**
    a. Count of IPs classified as unknown by GreyNoise Intelligence.
7. **Total Event Matches**
    a. Count of distinct IP addresses matched against GreyNoise Intelligence.
8. **Top 10 IP Indicators**
    a. Table of the 10 most frequently matched IPs from GreyNoise Intelligence with hostname details.

9. **IOC by Classification**
    a. Pie chart showing GreyNoise Intelligence IOC distribution by classification.
10. **IOC Matches Over Time by Category**
    a. Bar chart showing GreyNoise Intelligence IOC match trends grouped by category.
11. **Top 10 Rarely Seen IP**
    a. Table of the 10 least frequently matched IPs from GreyNoise Intelligence.
12. **Top 10 Categories**
    a. Pie chart visualizing the top 10 IOC categories of GreyNoise Intelligence IOC matches.
13. **IOCs Geolocation Overview**
    a. World map plotting the geographic origin of matched GreyNoise Intelligence IOCs.
14. **Correlation Overview**
    a. Comprehensive investigation table displaying GreyNoise Intelligence IOC matches for in-depth analysis.

Screenshot


Google SecOps | SIEM Dashboards
SR
GreyNoise Correlation Dashboard
The Correlation dashboard is designed to identify relationships across indicators and events from other sources.
In the past 1 week
GreyNoise Intelligence Correlation Dashboard
The Correlation dashboard is designed to identify relationships of GreyNoise Intelligence indicators and events from other sources.
1. IOC Match Visibility: Monitor GreyNoise indicators matched in your environment
2. Threat Classification: Breakdown by Malicious, Suspicious, Benign, and Unknown IPs
3. Geolocation Overview: Map-based visualization of IOC geographic origins
4. IoC Filtering: Time filter and classification filter for deeper analysis
GREYNOISE UNIQUE IOC MAT...
11
GREYNOISE UNIQUE IOC MATCHES
Last refreshed: 11 minutes ago
1/3
BENIGN IPS
0
BENIGN IPS
Last refreshed: 11 minutes ago
1/3
UNKNOWN IPS
11
UNKNOWN IPS
Last refreshed: 11 minutes ago
1/3
MALICIOUS IPS
0
MALICIOUS COUNT
Last refreshed: 11 minutes ago
1/3
SUSPICIOUS IPS
0
SUSPICIOUS IPS
Last refreshed: 11 minutes ago
1/3
TOP 10 CATEGORIES
hosting
business
9.09%
90.91%
Last refreshed: 11 minutes ago | Time range: December 14 2025, 12:00 AM to December 18 2025, 06:27 AM
1/2
IOC BY CLASSIFICATION
Unknown
100%
Last refreshed: 11 minutes ago | Time range: December 14 2025, 12:00 AM to December 18 2025, 06:27 AM
1/2

IOC Match Event & Trends

Monitor the volume and patterns of GreyNoise Intelligence IOC match events across your environment. Includes duplicate matches to show true activity volume, temporal trends, and detailed drill-down capabilities for security investigations.

TOTAL EVENT MATCHES (1)

24

TOTAL EVENT MATCHES

Last refreshed: 11 minutes ago | Time range: December 14 2025, 12:00 AM to December 18 2025, 06:27 AM 1/2

EVENTS MATCHES IN LAST 24 HOURS

4

EVENTS MATCHES IN LAST 24 HOURS

Last refreshed: 11 minutes ago | Time range: September 20 2025, 01:00 AM to December 18 2025, 06:27 AM 1/2

TOP 10 IP INDICATORS (1)

IP	COUNT	HOSTNAME
104.18.32.47	4	-
172.64.155.209	4	-
150.171.27.11	3	-
150.171.28.11	3	-
172.217.194.188	2	-
142.250.4.188	2	-
64.233.170.188	2	-
52.123.129.14	1	-

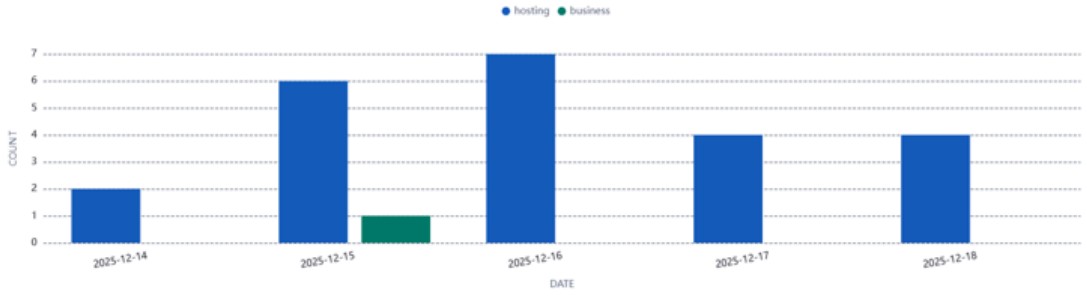
Last refreshed: 11 minutes ago | Time range: December 14 2025, 12:00 AM to December 18 2025, 06:27 AM 1/2

TOP 10 RARELY SEEN IP (1)

IP	COUNT	HOSTNAME
173.252.127.14	1	-
13.107.246.68	1	-
52.123.129.14	1	-
142.251.222.99	1	-
142.250.4.188	2	-
64.233.170.188	2	-
172.217.194.188	2	-
150.171.28.11	3	-

Last refreshed: 11 minutes ago | Time range: December 14 2025, 12:00 AM to December 18 2025, 06:27 AM 1/2

IOC MATCHES OVER TIME BY CATEGORY (1)



Last refreshed: 11 minutes ago | Time range: December 14 2025, 12:00 AM to December 18 2025, 06:27 AM | Showing results based on 6 rows

IOCS GEOLOCATION OVERVIEW (1)

Last refreshed: 11 minutes ago | Time range: December 14 2025, 12:00 AM to December 18 2025, 06:27 AM | Showing results based on 3 rows

CORRELATION OVERVIEW [DRILLDOWN] (1)

IP	CLASSIFICATION	CATEGORY	HOSTNAME	LATITUDE	LONGITUDE	COUNTRY	IOC MATCH DATE
52.123.129.14	Unknown	hosting	-	38.794595199999996	-106.5348379	United States	2025-12-18
64.233.170.188	Unknown	hosting	-	1.352083	103.819836	Singapore	2025-12-18
104.18.32.47	Unknown	hosting	-	38.794595199999996	-106.5348379	United States	2025-12-18
172.64.155.209	Unknown	hosting	-	38.794595199999996	-106.5348379	United States	2025-12-18
104.18.32.47	Unknown	hosting	-	38.794595199999996	-106.5348379	United States	2025-12-17
150.171.28.11	Unknown	hosting	-	38.794595199999996	-106.5348379	United States	2025-12-17
150.171.27.11	Unknown	hosting	-	38.794595199999996	-106.5348379	United States	2025-12-17

Last refreshed: 11 minutes ago | Time range: December 14 2025, 12:00 AM to December 18 2025, 06:27 AM | Showing results based on 24 rows

Limitations

1. If the user does not specify the required environment variable while configuring the Cloud Function, the script deployment will fail.
2. CBN parser will only be able to parse the GreyNoise entity data.
3. The Google SecOps Ingestion API has a payload limit of 4 MB. Logs exceeding this limit will not be ingested and will be skipped. To minimize data loss, please ensure that log sizes remain within the allowed limit.
4. Newly ingested entities often take additional time to appear on the dashboard. This latency impacts real time monitoring and reduces the effectiveness of dashboards for time sensitive investigations. A support ticket was raised regarding this issue for further investigation and resolution and its limitation from Google SecOps side.
5. When Cloud Run functions execute for more than 30 minutes, Cloud Scheduler will show a "Failed" status with 504 Gateway Timeout errors. This is expected behavior and does not indicate actual function failure. The Cloud Run function continues execution despite the timeout in Cloud Scheduler.
   Error Message Example:
   ```
   ERROR <timestamp> [httpRequest.requestMethod:POST]
   [httpRequest.status: 504] [httpRequest.responseSize: 72 B]
   [httpRequest.latency:1,799.798 s]
   [httpRequest.userAgent:Google-Cloud-Scheduler]
   https://<cloud_function_uri>
   ```
6. Google SecOps does not enforce validation on time range filters. Users can configure a start time greater than the end time, or set both values to be identical, without receiving an error. Filter validation logic is platform managed and cannot be extended.
7. Time range granularity in Google SecOps follows calendar-based logic. Selecting "Past 1 Week" on a Monday considers the Monday-to-Sunday range, potentially returning only the current day's data if selected at the start of a week. Similar behavior applies to monthly filters.

Troubleshooting

This section describes the common issues that might happen during the deployment or the running of the app and the steps to resolve the issues.

1. GCloud logs can be used for troubleshooting.
    a. Log in to the "https://console.cloud.google.com/" using valid credentials.
    b. Navigate to 'Cloud functions' and click on the deployed function where you can find the logs module.
    c. Logs can be filtered using severity.
2. If you test the cloud function immediately after deploying it on gcloud, It might be possible that the cloud function will not work as expected. To resolve this, wait for a few seconds and then test it.
3. If the cloud function stops its execution because memory exceeds the limit, reconfigure the cloud function's memory configuration and increase the memory limit.
4. The dashboard may be slow to load or unresponsive - This could be due to a problem with the data source being unavailable or having too much data, the query that is being used, or the way that the dashboard is being rendered.
5. Entities are successfully ingested but not displayed in search results this could be due to following reasons.
    a. The time range for the search is outside of the timestamp associated with the ingested event.
    b. The ingested entity might be a duplicate entity with the same payload.

GCP Resources/Services Approximate Cost Details

Service	Standard Configurations	Purpose	Reference
Cloud Functions	Type: Memory: 8192MB CPU: 4.8GHz Execution time per function (ms): 3600 Invocations per month: 1500 Minimum number of instances: 1	Function / Script which pulls data from Google Threat Intelligence Platform using API and ingest into Google SecOps.	Approx cost ~ $66/month https://cloud.google.com/functions/pricing
Cloud Storage (Bucket)	Total Amount of Storage: 1 GiB	Storage bucket used to manage API checkpoints	Approx cost ~ $0.02/month https://cloud.google.com/storage/pricing
Secret Manager	Access operations: 1500	Used to maintain credentials.	Approx cost ~ $0/month https://cloud.google.com/secret-manager/pricing
Cloud Scheduler	Total amount of jobs: 1	Scheduler which executes above cloud function at a specific time interval.	Approx cost ~ $0/month https://cloud.google.com/scheduler/pricing

Note: Users can also calculate (using pricing calculator) the estimated price of the Google Cloud services used.