AbuseIPDB Function for IBM SOAR

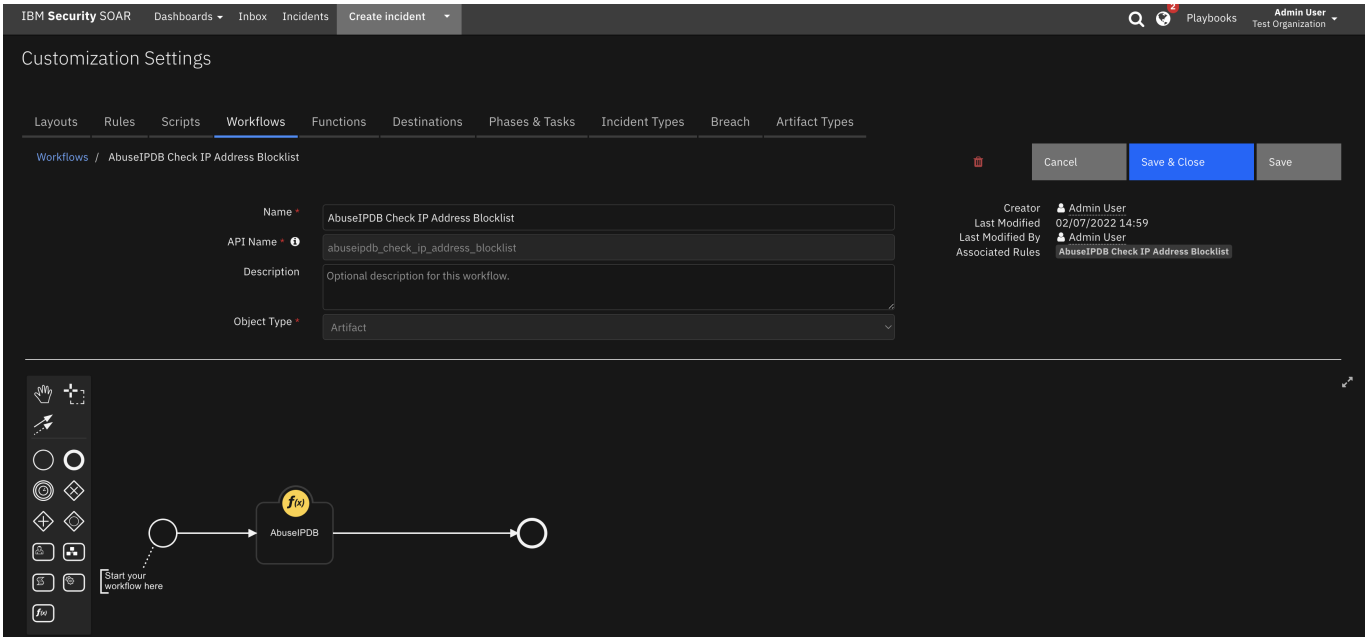
Table of Contents

Release Notes

Version	Date	Notes
1.0.0	02/2022	Initial Release

Overview

IBM Security SOAR app for AbuseIPDB'



This app pulls data from AbuseIPDB (www.abuseipdb.com) and checks if an IP artifact is blacklisted. If so, it will add a hit to the artifact. This app requires an AbuseIPDB account and an v2 api key to work.

Key Features

- The workflow creates a hit in the artifact containing information on the IP address.

Requirements

- resilient-circuits>=43.0.0

This app supports the IBM Security QRadar SOAR Platform and the IBM Security QRadar SOAR for IBM Cloud Pak for Security.

SOAR platform

The SOAR platform supports two app deployment mechanisms, App Host and integration server.

If deploying to a SOAR platform with an App Host, the requirements are:

- SOAR platform >= `43.0.0`.
- The app is in a container-based format (available from the AppExchange as a `zip` file).

If deploying to a SOAR platform with an integration server, the requirements are:

- SOAR platform >= `43.0.0`.
- The app is in the older integration format (available from the AppExchange as a `zip` file which contains a `tar.gz` file).
- Integration server is running `resilient-circuits>=43.0.0`.
- If using an API key account, make sure the account provides the following minimum permissions:

Name	Permissions
Org Data	Read
Function	Read
Artifact	Read, Edit

The following SOAR platform guides provide additional information:

- *App Host Deployment Guide*: provides installation, configuration, and troubleshooting information, including proxy server settings.
- *Integration Server Guide*: provides installation, configuration, and troubleshooting information, including proxy server settings.
- *System Administrator Guide*: provides the procedure to install, configure and deploy apps.

The above guides are available on the IBM Documentation website at ibm.biz/soar-docs. On this web page, select your SOAR platform version. On the follow-on page, you can find the *App Host Deployment Guide* or *Integration Server Guide* by expanding **Apps** in the Table of Contents pane. The System Administrator Guide is available by expanding **System Administrator**.

Cloud Pak for Security

If you are deploying to IBM Cloud Pak for Security, the requirements are:

- IBM Cloud Pak for Security >= 1.4.
- Cloud Pak is configured with an App Host.
- The app is in a container-based format (available from the AppExchange as a `zip` file).

The following Cloud Pak guides provide additional information:

- *App Host Deployment Guide*: provides installation, configuration, and troubleshooting information, including proxy server settings. From the Table of Contents, select Case Management and Orchestration & Automation > **Orchestration and Automation Apps**.
- *System Administrator Guide*: provides information to install, configure, and deploy apps. From the IBM Cloud Pak for Security IBM Documentation table of contents, select Case Management and Orchestration & Automation > **System administrator**.

These guides are available on the IBM Documentation website at ibm.biz/cp4s-docs. From this web page, select your IBM Cloud Pak for Security version. From the version-specific IBM Documentation page, select Case Management and Orchestration & Automation.

Proxy Server

The app does support a proxy server.

Python Environment

Both Python 2.7 and Python 3.6 are supported. Additional package dependencies may exist for each of these packages:

- resilient-circuits>=43.0.0

Endpoint Developed With

This app has been implemented using:

Product Name	Product Version	API URL	API Version
AbuseIPDB	------	https://api.abuseipdb.com/api/v2/check	v2

Prerequisites

- An AbuseIPDB account and a v2 API key

Installation

Install

- To install or uninstall an App or Integration on the *SOAR platform*, see the documentation at ibm.biz/soar-docs.

- To install or uninstall an App on *IBM Cloud Pak for Security*, see the documentation at ibm.biz/cp4s-docs and follow the instructions above to navigate to Orchestration and Automation.

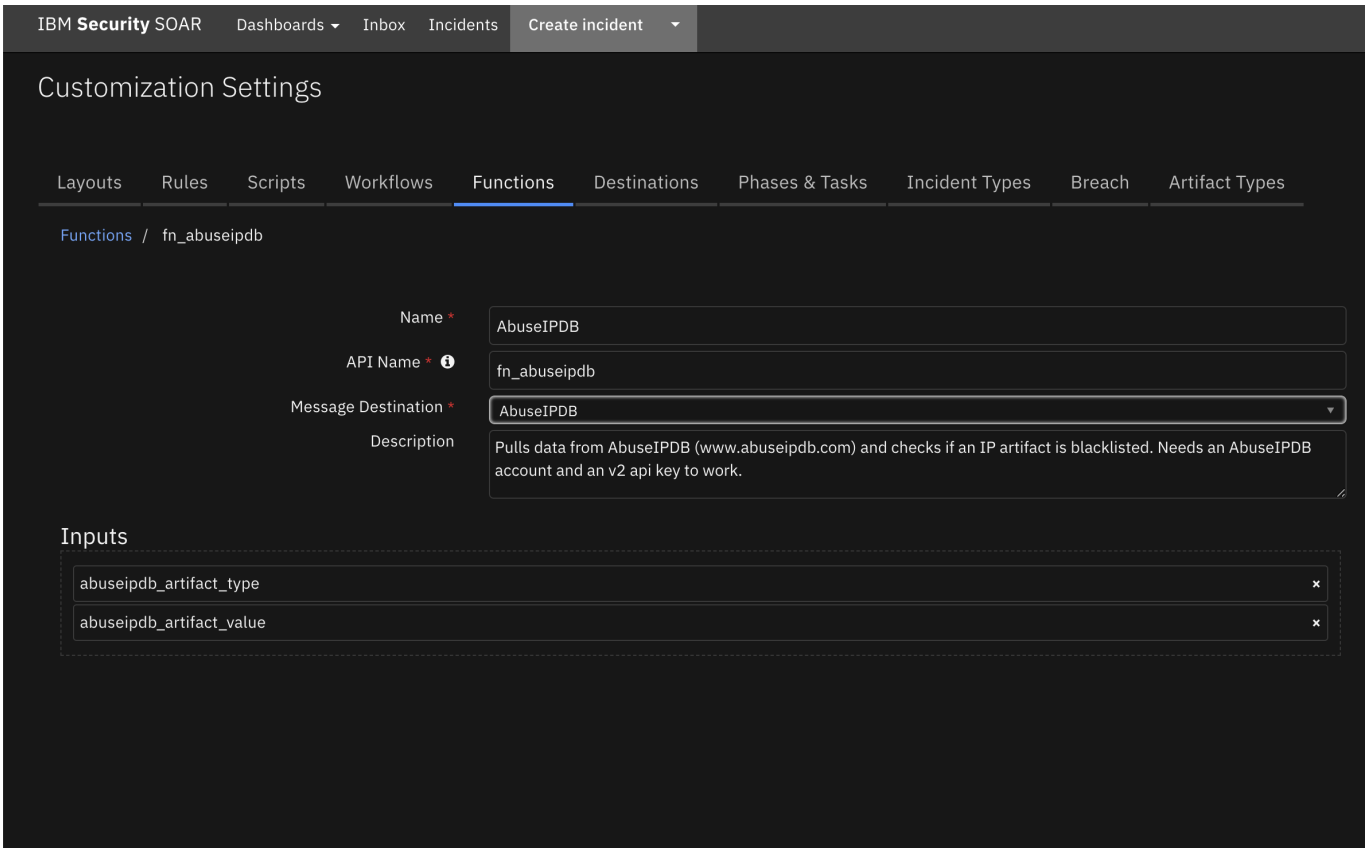
App Configuration

The following table provides the settings you need to configure the app. These settings are made in the app.config file. See the documentation discussed in the Requirements section for the procedure.

Config	Required	Example	Description
abuseipdb_key	Yes	`[your api key from your AbuseIPDB account]`	--
abuseipdb_url	Yes	`https://api.abuseipdb.com/api/v2/check`	--
ignore_white_listed	Yes	`True`	--

Function - AbuseIPDB

Pulls data from AbuseIPDB (www.abuseipdb.com) and checks if an IP artifact is blacklisted. Needs an AbuseIPDB account and an v2 api key to work.



▶ Inputs:

Name	Type	Required	Example	Tooltip
`abuseipdb_artifact_type`	`text`	Yes	`-`	-
`abuseipdb_artifact_value`	`text`	Yes	`-`	-

▶ Outputs:

> **NOTE:** This example might be in JSON format, but `results` is a Python Dictionary on the SOAR platform.


```
results = {
  "content": {
    "data": {
      "abuseConfidenceScore": 100,
      "countryCode": "TH",
      "countryName": "Thailand",
      "domain": "cattelecom.com",
      "hostnames": [],
      "ipAddress": "110.77.136.226",
      "ipVersion": 4,
      "isPublic": true,
      "isWhitelisted": false,
      "isp": "CAT Telecom Public Company Ltd",
      "lastReportedAt": "2022-02-08T17:10:55+00:00",
      "numDistinctUsers": 32,
      "reports": [
        {
          "categories": [
            18
          ],
          "comment": "Attempted Brute Force (dovecot)",
          "reportedAt": "2022-02-08T17:10:55+00:00",
          "reporterCountryCode": "GB",
          "reporterCountryName": "United Kingdom of Great Britain and
Northern Ireland",
          "reporterId": 34703
        },
        {
          "categories": [
            18
          ],
          "comment": "Email Auth Brute force attack 1/1 in last day",
          "reportedAt": "2022-02-06T19:27:40+00:00",
          "reporterCountryCode": "GB",
          "reporterCountryName": "United Kingdom of Great Britain and
Northern Ireland",
          "reporterId": 49881
        }
      ],
      "totalReports": 2,
      "usageType": null
    }
  },
  "inputs": {
    "abuseipdb_artifact_type": "IP Address",
    "abuseipdb_artifact_value": "110.77.136.226"
  },
```

```
  "metrics": {
    "execution_time_ms": 298,
    "host": "My Host",
    "package": "fn-abuseipdb",
    "package_version": "1.0.0",
    "timestamp": "2022-02-11 14:18:53",
    "version": "1.0"
  },
  "raw": null,
  "reason": null,
  "success": true,
  "version": 2.0
}
```


▶ Example Pre-Process Script:

```
inputs.abuseipdb_artifact_type = artifact.type
inputs.abuseipdb_artifact_value = artifact.value
```

▶ Example Post-Process Script:

```
CATEGORIES= {
  3: "Fraud Orders",
  4: "DDoS Attack",
  5: "FTP Brute-Force",
  6: "Ping of Death",
  7: "Phishing",
  8: "Fraud VoIP",
  9: "Open Proxy",
  10: "Web Spam",
  11: "Email Spam",
  12: "Blog Spam",
  13: "VPN IP",
  14: "Port Scan",
  15: "Hacking",
  16: "SQL Injection",
  17: "Spoofing",
  18: "Brute-Force",
  19: "Bad Web Bot",
  20: "Exploited Host",
  21: "Web App Attack",
  22: "SSH",
  23: "IoT Targeted",
}

if results.success:
  resp_data = results.content['data']
  number_of_reports = resp_data['totalReports']
  country_name = resp_data['countryName']
```

```
  most_recent_report = resp_data['lastReportedAt']
  confidence_score = resp_data.get("abuseConfidenceScore", 0)

  hit = []

  # get clean list of de-duped categories
  categories_names = ""
  if resp_data.get('reports'):
      categories_list = []
      for report in resp_data['reports']:
          categories_list.extend(report["categories"])
      categories_set = set(categories_list)  # dedup list
      categories_names = u', '.join(CATEGORIES.get(item, 'unknown') for
item in categories_set)

  # only return data if there's anything useful
  if number_of_reports or confidence_score:
      hit = [
        {
          "name": "Confidence Score",
          "type": "number",
          "value": "{}".format(confidence_score)
        },
        {
          "name": "Number of Reports",
          "type": "number",
          "value": "{}".format(number_of_reports)
        },
        {
          "name": "Country",
          "type": "string",
          "value": "{}".format(country_name)
        },
        {
          "name": "Most Recent Report",
          "type": "string",
          "value": "{}".format(most_recent_report)
        },
        {
          "name": "Categories",
          "type": "string",
          "value": "{}".format(categories_names)
        }
        ]
  artifact.addHit("AbuseIPDB Function hits added", hit)
else:
  incident.addNote("AbuseIPDB Check IP Address Blocklist failed:
{}".format(results.reason))
```

Rules

Rule Name	Object	Workflow Triggered
AbuseIPDB Check IP Address Blocklist	artifact	`abuseipdb_check_ip_address_blocklist`

Troubleshooting & Support

Refer to the documentation listed in the Requirements section for troubleshooting information.

For Support

This is a IBM Community provided App. Please search the Community ibm.biz/soarcommunity for assistance.