SentinelOne

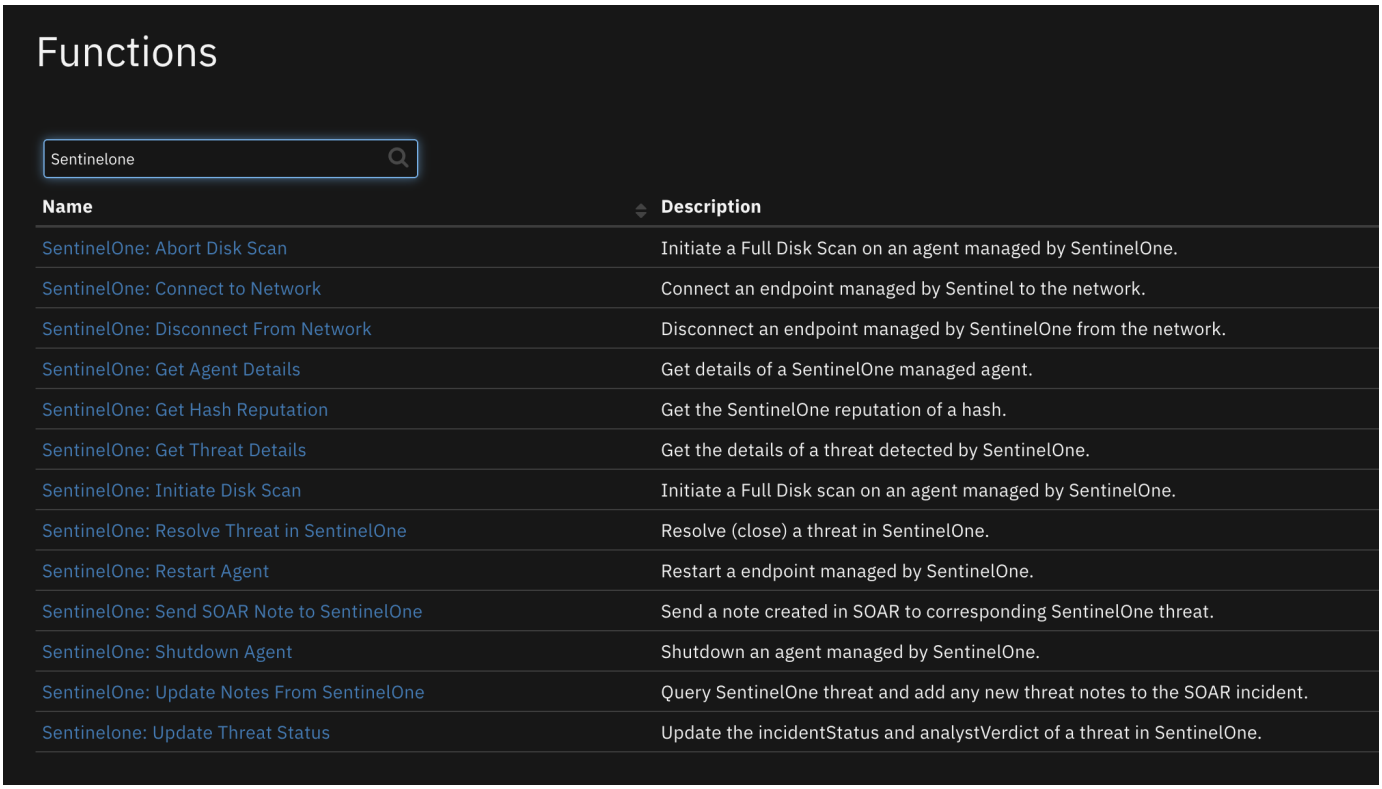
Table of Contents

Release Notes

Version	Date	Notes
1.0.0	1/2022	Initial Release

Overview

IBM Security SOAR app for SentinelOne



The SentinelOne platform provides AI-powered prevention, detection, response, and threat hunting across user endpoints, containers, cloud workloads, and IoT devices.

This app escalates SentinelOne threat incidents into IBM Security SOAR as an incident/case.

Key Features

The SentinelOne app implements the following functionality in the IBM SOAR platform:

- Poll SentinelOne for threats and create a corresponding incident/case in the IBM SOAR platform for each threat.
- Get SentinelOne threat notes and add them as notes in corresponding IBM SOAR incident/case.
- Resolve a SentinelOne threat when the corresponding IBM SOAR incident is closed.
- Close an IBM SOAR incident/case when the corresponding SentinelOne threat is resolved in SentinelOne.
- Create artifacts from the SentinelOne threat in the IBM SOAR platform.
- Provide information on the SentinelOne agent (endpoint) in a data table.
- Allow the IBM SOAR user to perform the following actions on the SentinelOne agent:
  - connect/disconnect agent from network
  - shutdown/restart agent
  - initiate/abort full disk scan
- Get the agent or threat details and write the JSON returned from SentinelOne in a formatted incident note.

Requirements

This app supports the IBM SOAR Platform and the 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 >= `40.2.81`.
- 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 >= `40.2.81`.

- 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>=40.0.0`.
- If using an API key account, make sure the account provides the following minimum permissions:

Name	Permissions
Org Data	Read
Function	Read
Incident	Read, Edit, Create, Owner, Status
Incident Notes	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 Knowledge Center 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 **Resilient 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 Knowledge Center table of contents, select Case Management and Orchestration & Automation > **System administrator**.

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

Proxy Server

The app supports a proxy server via the https_proxy and http_proxy app.config settings.

Python Environment

Python 3.6+ is supported. Additional package dependencies may exist for each of these packages:

- jinja2
- resilient-circuits>=40.0.0
- resilient-lib

Endpoint Developed With

This app has been implemented using:

Product Name	Product Version	API URL	API Version
SentinelOne	Queensland#11	https://SentinelOne-server/web/api/v2.1	2.1

Prerequisites

- A SentinelOne cloud account with Admin role.

Configuration

- To run the app, obtain an API token from the SentinelOne **My User** menu item in the **Options** menu and place in the app.config file.

Permissions

- SentinelOne Admin role is required in the SentinelOne account that IBM SOAR is communicating with as specified in the App configration file.

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

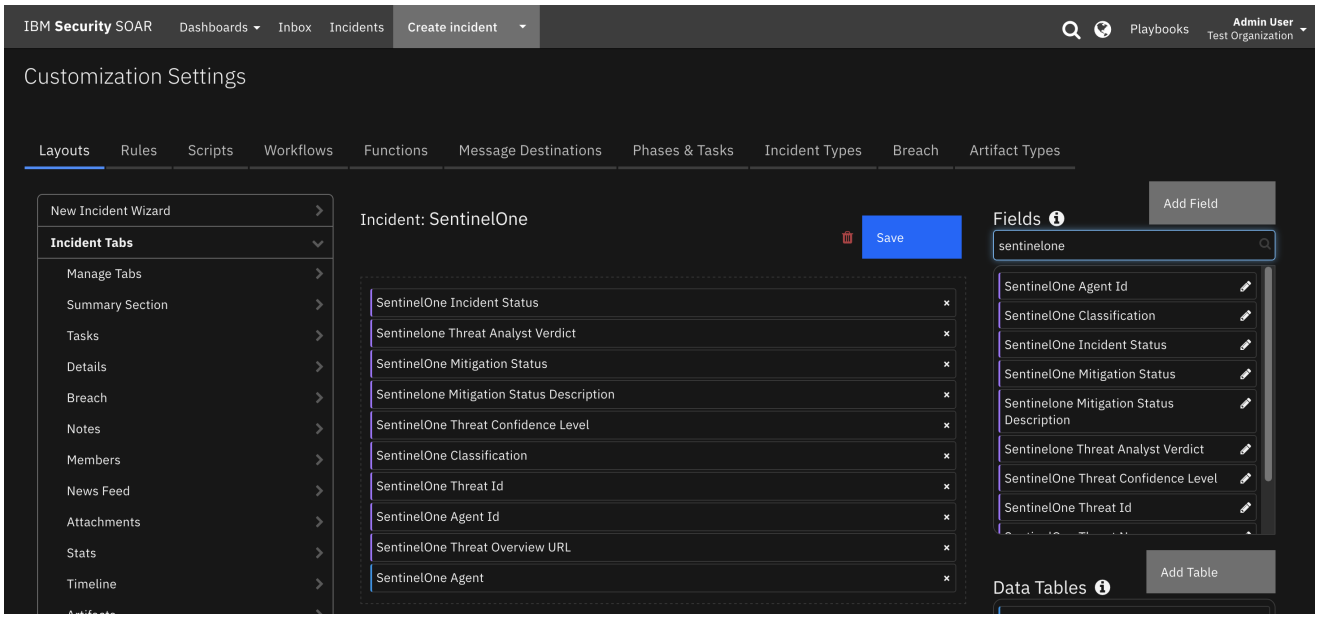
The following table provides the settings you need to configure the app. These settings are made in the app.config file. See the documentation discussed in the Requirements section for the procedure.

Config	Required	Example	Description
sentinelone_server	Yes	`xxx.sentinelone.net`	*SentinelOne server*
api_version	Yes	`2.1`	*SentinelOne REST API version*
api_token	Yes	`xxxxxx`	*SentinelOne user API token*
polling_interval	Yes	`60`	*Poller interval time in seconds. Value of zero to turn poller off.*
polling_lookback	Yes	`120`	*Number of minutes to look back for threat updates. Value is only used on the first time polling when the app starts.*
account_ids	No	`123456789012345678`	*Comma separated list of SentinelOne accountIds to query for threats.*
site_ids	No	`987654321098765432`	*Comma separated list of SentinelOne siteIds to query for threats.*
incident_statuses	No	`resolved,in_progress,unresolved`	*Comma separated list of SentinelOne incidentStatuses to query for threats. NOTE: Include incidentStatus **resolved** if the poller should close SOAR incidents that are marked as **resolved** in SentinelOne.*
limit	No	`25`	*Limit number of threats to return from query.*
sort_by	No	`createdDate`	*The column to sort results by when querying threats.*

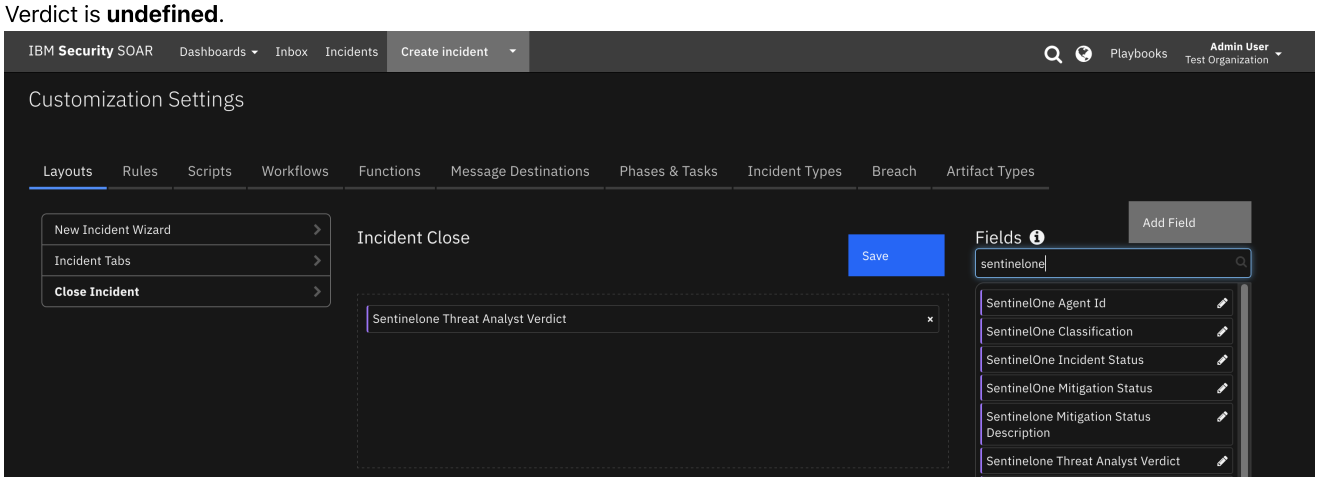
Config	Required	Example	Description
sort_order	No	`desc`	*Sort direction to return threat query results: 'asc' or 'desc'.*
query_param	No	`threat details`	*Full text search for fields when querying threats.*
send_soar_link_to_sentinelone	No	true	*Send SOAR incident URL live link via threat note to SentinelOn.e*
create_incident_template	No	/path/create_incident_template.jinja	*Path to custom create incident jinja template.*
close_incident_template	No	/path/close_incident_template.jinja	*Path to custom close incident jinja template.*
update_incident_template	No	/path/update_incident_template.jinja	*Path to custom update incident jinja template.*
verify	No	/path/toclient_certificate.pem	*Path to client SSL certificate.*

Custom Layouts

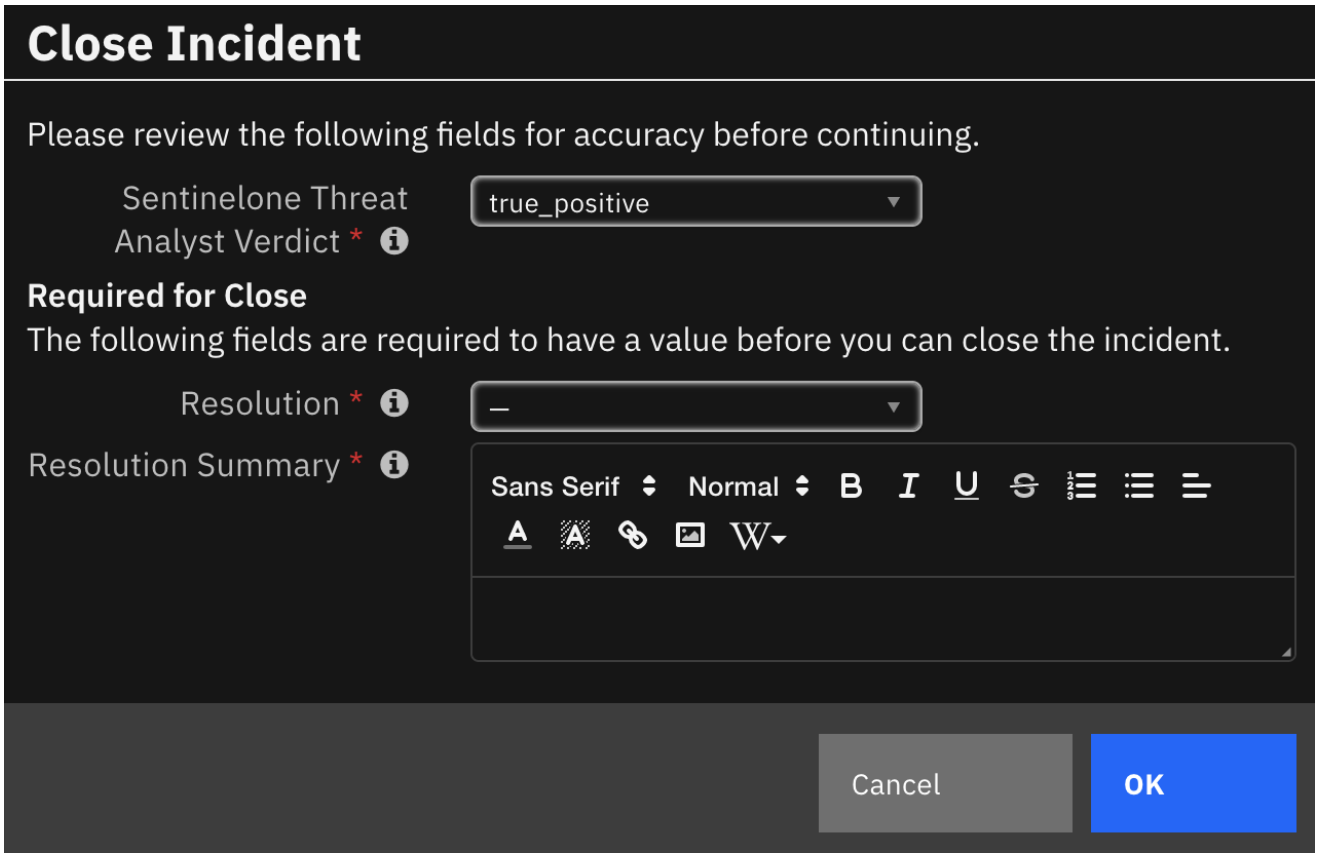
- Create a SentinelOne incident tab and import the data table and custom fields as shown in the screenshot:



- Create a Close Incident tab so that when a SOAR incident corresponding to a SentinelOne threat is closed in SOAR, the user can select a SentinelOne Threat Analyst Verdict to send to SentinelOne. The SentinelOne threat cannot be closed if the Analyst Verdict is **undefined**.



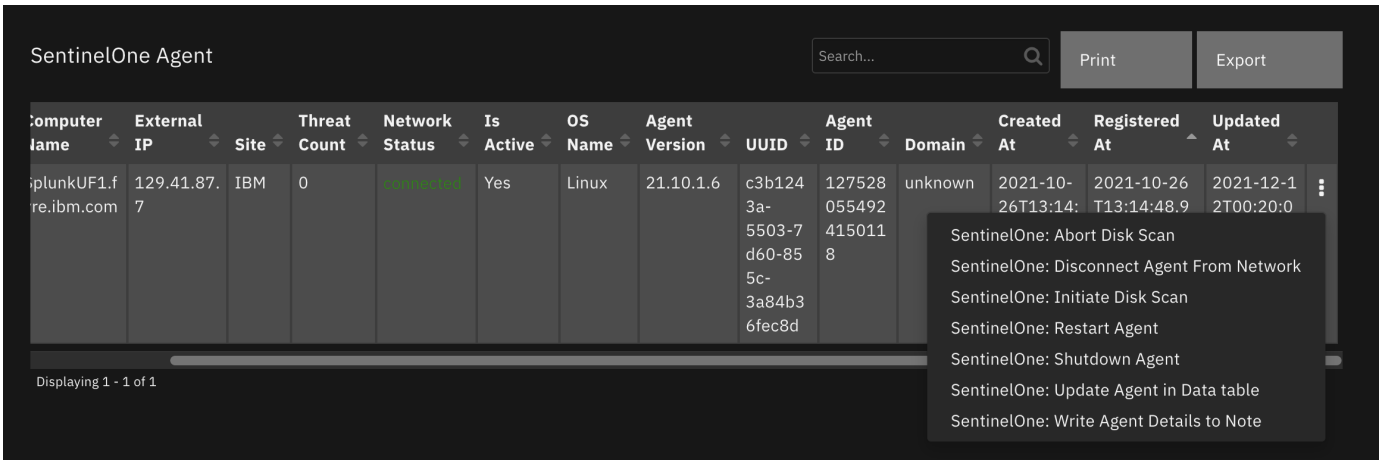
Here is a screenshot of the SentinelOne custom Close Incident popup:



NOTE: If a Close incident tab is created, all non-SentinelOne incidents also contain the SentinelOne Threat Analyst Verdict select field in the Close menu pop-up. To avoid this, you can alternatively, use the provided incident menu item rule **SentinelOne: Update Analyst Verdict and Threat Status**, which a user can run manually to update the Analyst Verdict and Threat status in SentinelOne. When the rule is run to resolve a SentinelOne threat, choose an Analyst Verdict and set the Threat Status to **resolved**. SOAR updates the SentinelOne threat which in turn causes the incident to be closed in IBM SOAR on the next poll.

Function - SentinelOne: Abort Disk Scan

Initiate a Full Disk Scan on an agent managed by SentinelOne. If the agent *isActive* field is **No**, the agent action to abort a disk scan is not an option.



▶ Inputs:

Name	Type	Required	Example	Tooltip
`sentinelone_agent_id`	`text`	Yes	`–`	-

▶ Outputs:

> **NOTE:** this example may be in JSON format, yet on the SOAR Platform `results` will be a Python Dictionary

```
results = {
  "content": {
    "data": {
      "affected": 1
    }
  },
  "inputs": {
    "sentinelone_agent_id": "1212121212121212121"
  },
  "metrics": {
    "execution_time_ms": 4108,
    "host": "myHost",
    "package": "fn-sentinelone",
    "package_version": "1.0.0",
    "timestamp": "2021-12-13 11:22:36",
    "version": "1.0"
  },
  "raw": null,
  "reason": null,
  "success": true,
  "version": 2.0
}
```

▶ Example Pre-Process Script:

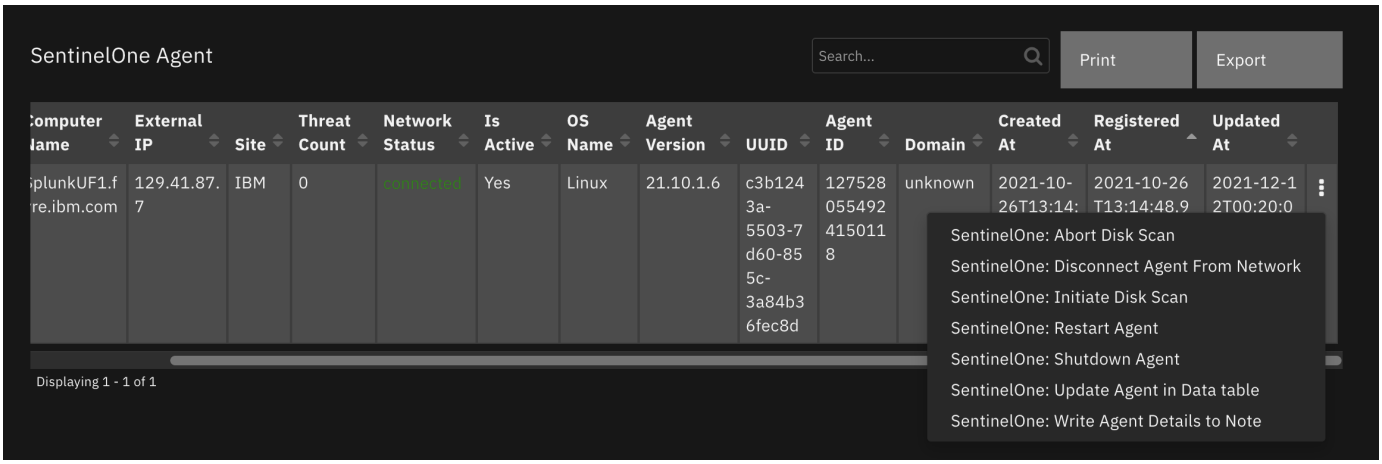
```
inputs.sentinelone_agent_id = row.sentinelone_dt_agent_id
```

▶ Example Post-Process Script:

```
so_inputs = results.get("inputs")
agent_id = so_inputs.get("sentinelone_agent_id")
note = u"<b>SentinelOne: Abort Full Disk Scan </b><br>  SentinelOne Agent Id:
{0}".format(agent_id)
content = results.get("content")
if content:
  data = content.get("data")
  if data:
    if int(data.get("affected")) <= 0:
      note = u"{0} Full Disk Scan was NOT aborted.".format(note)
    else:
      note = u"{0} Full Disk Scan aborted.".format(note)
  else:
    note = u"{0} Full Disk Scan was NOT aborted. No 'data' returned from
function".format(note)
else:
    note = u"{0} Full Disk Scan was NOT aborted. No content returned from
function".format(note)

incident.addNote(helper.createRichText(note))
```

Function - SentinelOne: Connect to Network

Connect an endpoint managed by Sentinel to the network. If the agent *isActive* field is **No**, the agent action to connect is not an option.



▶ Inputs:

Name	Type	Required	Example	Tooltip
sentinelone_agent_id	text	Yes	–	-

▶ Outputs:

> **NOTE:** this example may be in JSON format, yet on the SOAR Platform `results` will be a Python Dictionary

```
results = {
  "content": {
    "data": {
      "affected": 1
    }
  },
  "inputs": {
    "sentinelone_agent_id": "1212121212121212121"
  },
  "metrics": {
    "execution_time_ms": 226,
    "host": "myHost",
    "package": "fn-sentinelone",
    "package_version": "1.0.0",
    "timestamp": "2021-12-13 11:21:53",
    "version": "1.0"
  },
  "raw": null,
  "reason": null,
  "success": true,
  "version": 2.0
}
```

▶ Example Pre-Process Script:

```
inputs.sentinelone_agent_id = row.sentinelone_dt_agent_id
```

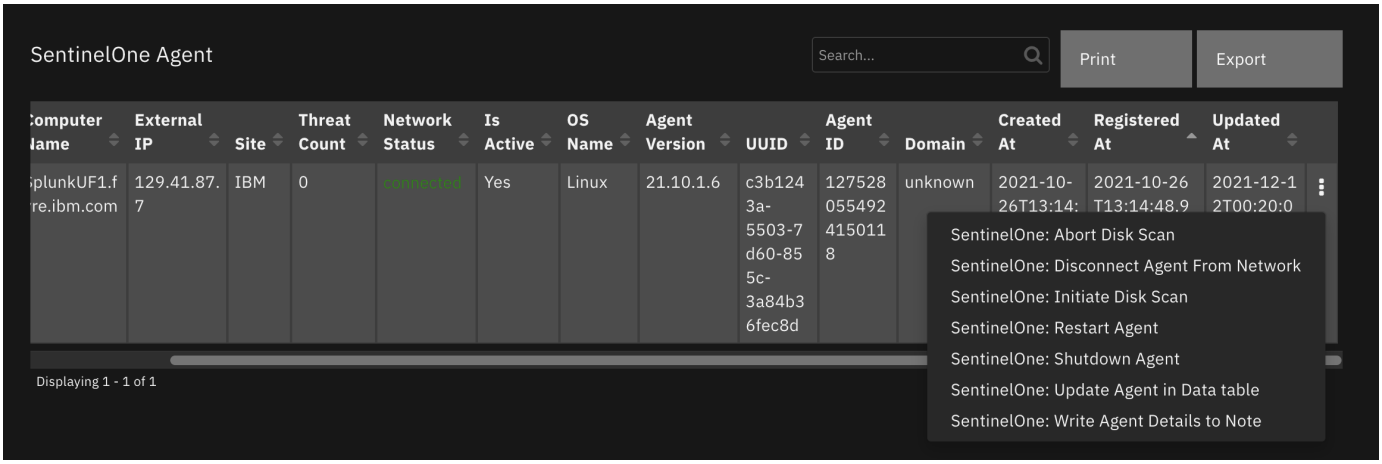
▶ Example Post-Process Script:

```
so_inputs = results.get("inputs")
agent_id = so_inputs.get("sentinelone_agent_id")
note = u"<b>SentinelOne: Connect to Network </b><br>  SentinelOne Agent Id:
{0}".format(agent_id)
content = results.get("content")
if content:
```

```
  data = content.get("data")
  if data:
    if int(data.get("affected")) <= 0:
      note = u"{0} is NOT connected to network".format(note)
    else:
      networkStatus = u"""<p style= "color:{color}">{status}</p>""".format(color="green",
status="connected")
      row["sentinelone_dt_network_status"] = helper.createRichText(networkStatus)
      note = u"{0} is connected to network".format(note)
  else:
    note = u"{0} no data returned from function".format(note)
else:
    note = u"{0} no content data returned from function".format(note)

incident.addNote(helper.createRichText(note))
```

Function - SentinelOne: Disconnect From Network

Disconnect an endpoint managed by SentinelOne from the network. If the agent *isActive* field is **No**, the agent action to disconnect is not an option.



▶ Inputs:

Name	Type	Required	Example	Tooltip
sentinelone_agent_id	text	Yes	–	-

▶ Outputs:

> **NOTE:** this example may be in JSON format, yet on the SOAR Platform `results` will be a Python Dictionary

```
results = {
  "content": {
    "data": {
      "affected": 1
    }
  },
  "inputs": {
    "sentinelone_agent_id": "1275282318251495460"
  },
  "metrics": {
    "execution_time_ms": 242,
    "host": "myHost",
    "package": "fn-sentinelone",
    "package_version": "1.0.0",
    "timestamp": "2021-12-13 11:21:08",
```

```
    "version": "1.0"
  },
  "raw": null,
  "reason": null,
  "success": true,
  "version": 2.0
}
```


▶ Example Pre-Process Script:

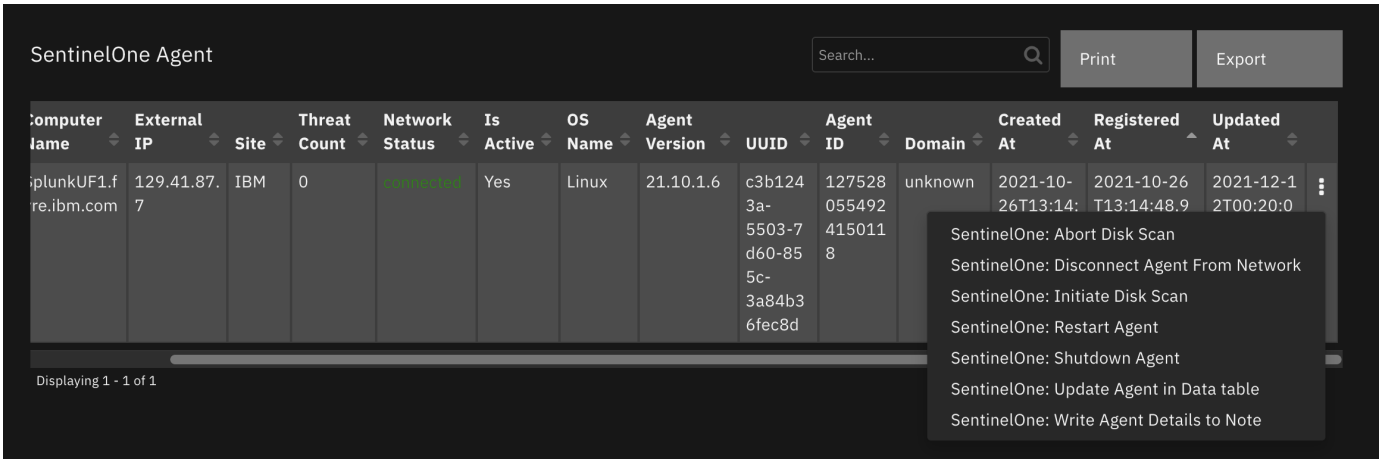
```
inputs.sentinelone_agent_id = row.sentinelone_dt_agent_id
```

▶ Example Post-Process Script:

```
so_inputs = results.get("inputs")
agent_id = so_inputs.get("sentinelone_agent_id")
note = u"<b>SentinelOne: Disconnect From Network </b><br>  SentinelOne Agent Id:
{0}".format(agent_id)
content = results.get("content")
if content:
  data = content.get("data")
  if data:
    if int(data.get("affected")) <= 0:
      note = u"{0} was not disconnected from network".format(note)
    else:
      networkStatus = u"""<p style= "color:{color}">{status}</p>""".format(color="red",
status="disconnected")
      row["sentinelone_dt_network_status"] = helper.createRichText(networkStatus)
      note = u"{0} is disconnected from network".format(note)
  else:
    note = u"{0} no data returned from function".format(note)
else:
    note = u"{0} no content data returned from function".format(note)

incident.addNote(helper.createRichText(note))
```

Function - SentinelOne: Get Agent Details

Get details of a SentinelOne managed agent. The example workflow writes the SentinelOne agent JSON details to a formatted incident note.



▶ Inputs:

Name	Type	Required	Example	Tooltip
sentinelone_agent_id	text	Yes	–	-

▶ Outputs:

> **NOTE:** this example may be in JSON format, yet on the SOAR Platform `results` will be a Python Dictionary


```
results = {
  "content": {
    "data": [
      {
        "accountId": "434343434343434343",
        "accountName": "SentinelOne",
        "activeDirectory": {
          "computerDistinguishedName": null,
          "computerMemberOf": [],
          "lastUserDistinguishedName": null,
          "lastUserMemberOf": []
        },
        "activeThreats": 0,
        "agentVersion": "21.10.1.6",
        "allowRemoteShell": false,
        "appsVulnerabilityStatus": "not_applicable",
        "cloudProviders": {},
        "computerName": "computerName",
        "consoleMigrationStatus": "N/A",
        "coreCount": 2,
        "cpuCount": 2,
        "cpuId": "AMD EPYC Processor (with IBPB)",
        "createdAt": "2021-10-26T13:18:19.135249Z",
        "detectionState": null,
        "domain": "unknown",
        "encryptedApplications": false,
        "externalId": "",
        "externalIp": "12.4.8.3",
        "firewallEnabled": true,
        "firstFullModeTime": null,
        "groupId": "607447413805059643",
        "groupIp": "12.4.8.x",
        "groupName": "Default Group",
        "id": "1275282318251495460",
        "inRemoteShellSession": false,
        "infected": false,
        "installerType": ".rpm",
        "isActive": true,
        "isDecommissioned": false,
        "isPendingUninstall": false,
        "isUninstalled": false,
        "isUpToDate": true,
        "lastActiveDate": "2021-12-13T16:23:27.067226Z",
        "lastIpToMgmt": "10.21.10.88",
        "lastLoggedInUserName": "",
        "licenseKey": "",
        "locationEnabled": false,
        "locationType": "not_supported",
        "locations": null,
        "machineType": "server",
        "mitigationMode": "protect",
        "mitigationModeSuspicious": "detect",
        "modelName": "QEMU Standard PC (i440FX + PIIX, 1996)",
        "networkInterfaces": [
          {
            "gatewayIp": null,
            "gatewayMacAddress": null,
```

```
            "id": "1275282318259884069",
            "inet": [
              "10.21.10.88"
            ],
            "inet6": [],
            "name": "eth0",
            "physical": "00:00:0A:15:0A:58"
          },
          {
            "gatewayIp": "9.46.92.1",
            "gatewayMacAddress": "00:00:0c:9f:f0:01",
            "id": "1275282318268272678",
            "inet": [
              "9.46.92.211"
            ],
            "inet6": [],
            "name": "eth1",
            "physical": "00:00:09:2E:5C:D3"
          }
        ],
        "networkQuarantineEnabled": false,
        "networkStatus": "connected",
        "operationalState": "na",
        "operationalStateExpiration": null,
        "osArch": "64 bit",
        "osName": "Linux",
        "osRevision": "CentOS release 7.9.2009 (Core) 3.10.0-1160.49.1.el7.x86_64",
        "osStartTime": "2021-12-11T22:40:32Z",
        "osType": "linux",
        "osUsername": "root",
        "rangerStatus": "NotApplicable",
        "rangerVersion": null,
        "registeredAt": "2021-10-26T13:18:19.132188Z",
        "remoteProfilingState": "disabled",
        "remoteProfilingStateExpiration": null,
        "scanAbortedAt": "2021-12-10T21:57:56.711226Z",
        "scanFinishedAt": "2021-12-10T02:33:00.768224Z",
        "scanStartedAt": "2021-12-13T16:22:40.612822Z",
        "scanStatus": "started",
        "siteId": "607447413779893818",
        "siteName": "mySite",
        "storageName": null,
        "storageType": null,
        "threatRebootRequired": false,
        "totalMemory": 3789,
        "updatedAt": "2021-12-13T16:22:40.616308Z",
        "userActionsNeeded": [],
        "uuid": "8329e587-bbe9-b906-a6a5-646e2686eba9"
      }
    ],
    "pagination": {
      "nextCursor": null,
      "totalItems": 1
    }
  },
  "inputs": {
    "sentinelone_agent_id": "12121212121212121"
  },
  "metrics": {
    "execution_time_ms": 277,
    "host": "myHost",
    "package": "fn-sentinelone",
    "package_version": "1.0.0",
    "timestamp": "2021-12-13 11:23:55",
    "version": "1.0"
  },
```

```
  "raw": null,
  "reason": null,
  "success": true,
  "version": 2.0
}
```

▶ Example Pre-Process Script:

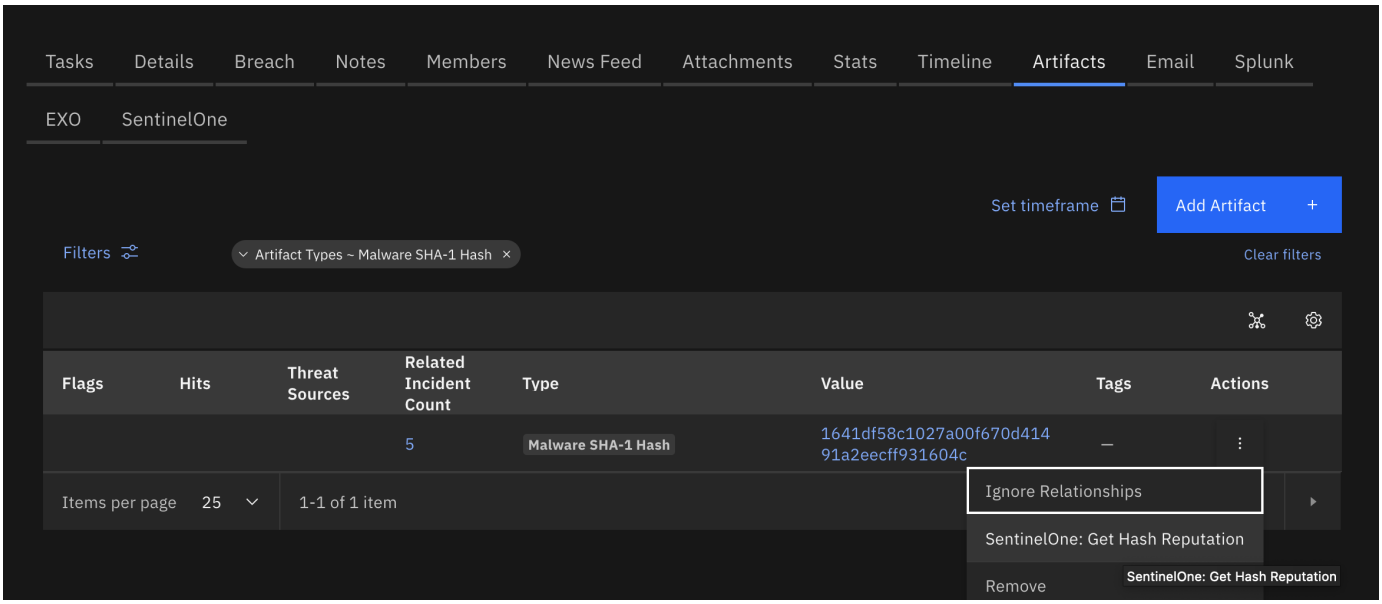
```
inputs.sentinelone_agent_id = incident.properties.sentinelone_agent_id
```

▶ Example Post-Process Script:

```
from java.util import Date

note = u"<b>SentinelOne: Add Agent to Data Table: </b> \n"
content = results.get("content")
if content:
  data = content.get("data")
  if data:
    data_len = len(data)
    note = u"{0} {1} agent added to SentinelOne Agent data table.".format(note, data_len)
    for agent in data:
      agent_row = incident.addRow("sentinelone_agents_dt")
      agent_row.sentinelone_dt_query_date = Date()
      agent_row.sentinelone_dt_agent_id = agent.get("id")
      networkStatus = agent.get("networkStatus")
      if networkStatus == "connected":
        display_color = "green"
      else:
        display_color = "red"
      networkStatus = u"""<p style= "color:{color}">{status}
</p>""".format(color=display_color, status=networkStatus)
      agent_row.sentinelone_dt_network_status = helper.createRichText(networkStatus)
      agent_row.sentinelone_dt_computername = agent.get("computerName")
      agent_row.sentinelone_dt_external_ip = agent.get("externalIp")
      agent_row.sentinelone_dt_site = agent.get("siteName")
      agent_row.sentinelone_dt_agent_version = agent.get("agentVersion")
      agent_row.sentinelone_dt_threat_count = agent.get("activeThreats")
      agent_row.sentinelone_dt_domain = agent.get("domain")
      agent_row.sentinelone_dt_os_name = agent.get("osName")
      agent_row.sentinelone_dt_uuid = agent.get("uuid")
      agent_row.sentinelone_dt_is_active = agent.get("isActive")
      agent_row.sentinelone_dt_registered = agent.get("registeredAt")
      agent_row.sentinelone_dt_created = agent.get("createdAt")
      agent_row.sentinelone_dt_updated = agent.get("updatedAt")
  else:
    note = u"{0} No data returned from function.".format(note)
else:
  note = u"{0} No content data returned from function.".format(note)

incident.addNote(helper.createRichText(note))
```

Function - SentinelOne: Get Hash Reputation

Get the SentinelOne reputation of a hash. The example workflow writes the results to an incident note.



▶ Inputs:

Name	Type	Required	Example	Tooltip
sentinelone_hash	text	No	–	-

▶ Outputs:

> **NOTE:** this example may be in JSON format, yet on the SOAR Platform `results` will be a Python Dictionary

```
results = {
  "content": {
    "data": {
      "rank": "0"
    }
  },
  "inputs": {
    "sentinelone_hash": "1641df58c1027a00f670d41491a2eecff931604c"
  },
  "metrics": {
    "execution_time_ms": 202,
    "host": "myHost",
    "package": "fn-sentinelone",
    "package_version": "1.0.0",
    "timestamp": "2021-12-13 11:27:14",
    "version": "1.0"
  },
  "raw": null,
  "reason": null,
  "success": true,
  "version": 2.0
}
```

▶ Example Pre-Process Script:

```
inputs.sentinelone_hash = artifact.value
```

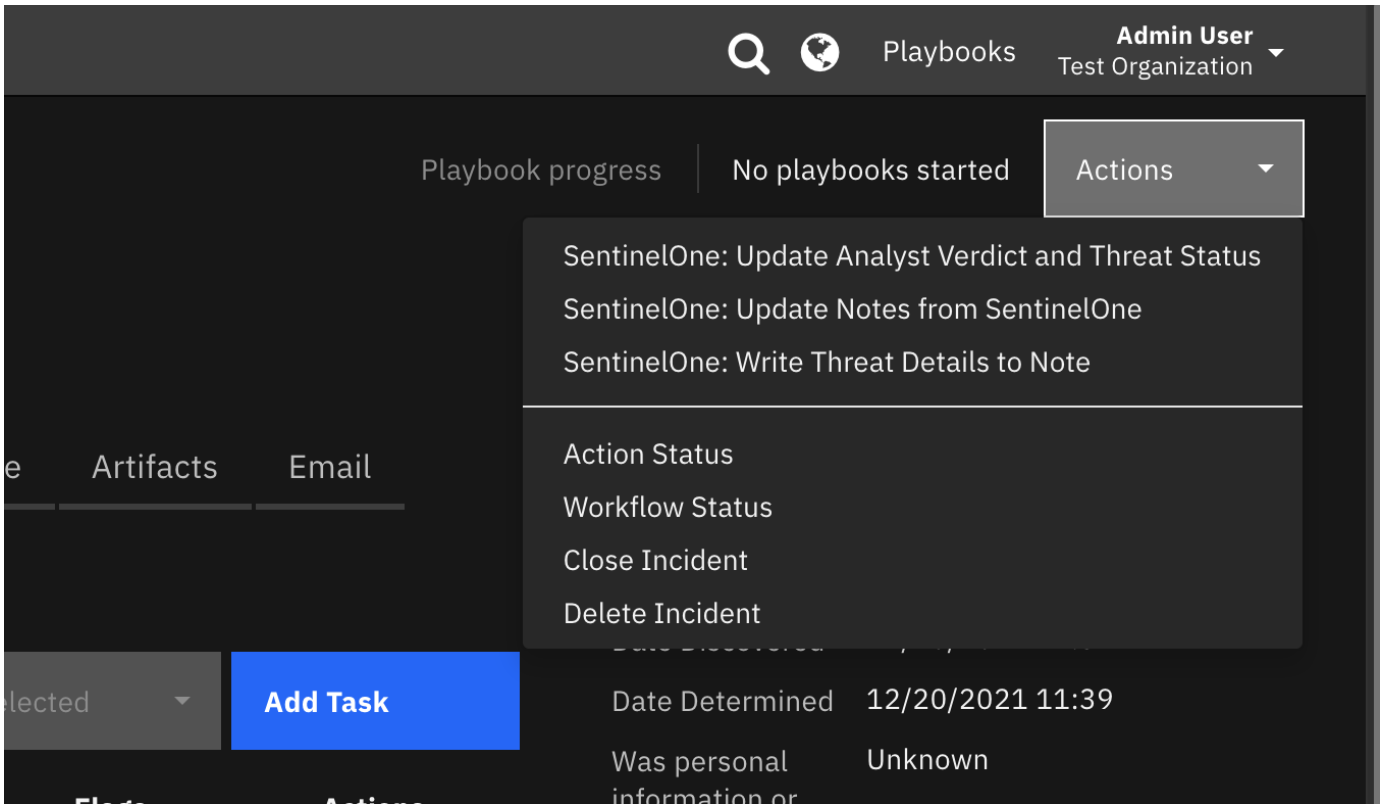
▶ Example Post-Process Script:

```
note = u"<b>SentinelOne: Get Hash Reputation: </b><br>"
```

```
content = results.get("content")
inputs = results.get("inputs")
hash_value = inputs.get("sentinelone_hash")
if content:
  data = content.get("data")
  if data:
    rank = data.get("rank")
    note = u"{0} Hash <b>{1}</b> has rank: <b>{2}</b>".format(note, hash_value, rank)
  else:
    note = u"{0} No data returned from function.".format(note)
else:
  note = u"{0} No content data returned from function.".format(note)

incident.addNote(helper.createRichText(note))
```

Function - SentinelOne: Get Threat Details

Get the details of a threat detected by SentinelOne. Results are written to a formatted incident note in the example workflow.



▶ Inputs:

Name	Type	Required	Example	Tooltip
sentinelone_threat_id	text	No	–	-

▶ Outputs:

> **NOTE:** this example may be in JSON format, yet on the SOAR Platform `results` will be a Python Dictionary

```
results = {
  "content": {
    "data": [
      {
        "agentDetectionInfo": {
          "accountId": "433241117337583618",
          "accountName": "SentinelOne",
          "agentDetectionState": null,
```

```
            "agentDomain": "unknown",
            "agentIpV4": "10.21.10.88,9.46.92.211",
            "agentIpV6": "",
            "agentLastLoggedInUserName": "",
            "agentMitigationMode": "detect",
            "agentOsName": "Linux",
            "agentOsRevision": "CentOS release 7.9.2009 (Core) 3.10.0-1160.49.1.el7.x86_64",
            "agentRegisteredAt": "2021-10-26T13:18:19.132188Z",
            "agentUuid": "8329e587-bbe9-b906-a6a5-646e2686eba9",
            "agentVersion": "21.10.1.6",
            "externalIp": "129.41.87.3",
            "groupId": "607447413805059643",
            "groupName": "Default Group",
            "siteId": "606060606060606060",
            "siteName": "mySite"
          },
          "agentRealtimeInfo": {
            "accountId": "434343434343434343434343434343",
            "accountName": "SentinelOne",
            "activeThreats": 1,
            "agentComputerName": "SplunkHF1.fyre.ibm.com",
            "agentDecommissionedAt": null,
            "agentDomain": "unknown",
            "agentId": "1275282318251495460",
            "agentInfected": true,
            "agentIsActive": true,
            "agentIsDecommissioned": false,
            "agentMachineType": "server",
            "agentMitigationMode": "detect",
            "agentNetworkStatus": "connected",
            "agentOsName": "Linux",
            "agentOsRevision": "CentOS release 7.9.2009 (Core) 3.10.0-1160.49.1.el7.x86_64",
            "agentOsType": "linux",
            "agentUuid": "8329e587-bbe9-b906-a6a5-646e2686eba9",
            "agentVersion": "21.10.1.6",
            "groupId": "6060606060606060606",
            "groupName": "Default Group",
            "networkInterfaces": [
              {
                "id": "1275282318268272678",
                "inet": [
                  "9.46.92.211"
                ],
                "inet6": [],
                "name": "eth1",
                "physical": "00:00:09:2E:5C:D3"
              },
              {
                "id": "1275282318259884069",
                "inet": [
                  "10.21.10.88"
                ],
                "inet6": [],
                "name": "eth0",
                "physical": "00:00:0A:15:0A:58"
              }
            ],
            "operationalState": "na",
            "rebootRequired": false,
            "scanAbortedAt": "2021-12-10T21:57:56.711226Z",
            "scanFinishedAt": "2021-12-10T02:33:00.768224Z",
            "scanStartedAt": "2021-12-13T16:22:40.612822Z",
            "scanStatus": "started",
            "siteId": "606060606060606060",
            "siteName": "IBM",
            "storageName": null,
```

```
      "storageType": null,
      "userActionsNeeded": []
    },
    "containerInfo": {
      "id": null,
      "image": null,
      "labels": null,
      "name": null
    },
    "id": "1308905355630511064",
    "indicators": [
      {
        "category": "Persistence",
        "description": "Unsigned kernel module was loaded.",
        "ids": [
          384
        ],
        "tactics": [
          {
            "name": "Persistence",
            "source": "MITRE",
            "techniques": [
              {
                "link": "https://attack.mitre.org/techniques/T1547/006/",
                "name": "T1547.006"
              }
            ]
          }
        ]
      },
      {
        "category": "Persistence",
        "description": "New kernel module was added.",
        "ids": [
          399
        ],
        "tactics": [
          {
            "name": "Persistence",
            "source": "MITRE",
            "techniques": [
              {
                "link": "https://attack.mitre.org/techniques/T1547/006/",
                "name": "T1547.006"
              }
            ]
          }
        ]
      }
    ],
    "kubernetesInfo": {
      "cluster": null,
      "controllerKind": null,
      "controllerLabels": null,
      "controllerName": null,
      "namespace": null,
      "namespaceLabels": null,
      "node": null,
      "pod": null,
      "podLabels": null
    },
    "mitigationStatus": [],
    "threatInfo": {
      "analystVerdict": "true_positive",
      "analystVerdictDescription": "True positive",
      "automaticallyResolved": false,
```

```
            "browserType": null,
            "certificateId": null,
            "classification": "Malware",
            "classificationSource": "Static",
            "cloudFilesHashVerdict": "provider_unknown",
            "collectionId": "1140024784343285701",
            "confidenceLevel": "suspicious",
            "createdAt": "2021-12-11T22:41:17.533077Z",
            "detectionEngines": [
              {
                "key": "executables",
                "title": "Behavioral AI"
              }
            ],
            "detectionType": "dynamic",
            "engines": [
              "DBT - Executables"
            ],
            "externalTicketExists": false,
            "externalTicketId": null,
            "failedActions": false,
            "fileExtension": null,
            "fileExtensionType": null,
            "filePath": "/opt/CrowdStrike/falcon-sensor12803",
            "fileSize": 1617904,
            "fileVerificationType": null,
            "identifiedAt": "2021-12-11T22:41:17.461397Z",
            "incidentStatus": "in_progress",
            "incidentStatusDescription": "In progress",
            "initiatedBy": "agent_policy",
            "initiatedByDescription": "Agent Policy",
            "initiatingUserId": null,
            "initiatingUsername": null,
            "isFileless": false,
            "isValidCertificate": null,
            "maliciousProcessArguments": " falcon-sensor",
            "md5": null,
            "mitigatedPreemptively": false,
            "mitigationStatus": "not_mitigated",
            "mitigationStatusDescription": "Not mitigated",
            "originatorProcess": "systemd",
            "pendingActions": false,
            "processUser": "root",
            "publisherName": null,
            "reachedEventsLimit": null,
            "rebootRequired": false,
            "sha1": "1641df58c1027a00f670d41491a2eecff931604c",
            "sha256": null,
            "storyline": "2588b11a-e3cd-1677-7746-3f85cd99c850",
            "threatId": "1308905355630511064",
            "threatName": "falcon-sensor12803",
            "updatedAt": "2021-12-13T16:25:55.851553Z"
          },
          "whiteningOptions": [
            "path",
            "hash"
          ]
        }
      ],
      "pagination": {
        "nextCursor": null,
        "totalItems": 1
      }
    },
    "inputs": {
      "sentinelone_threat_id": "1308905355630511064"
```

```
  },
  "metrics": {
    "execution_time_ms": 260,
    "host": "myHost",
    "package": "fn-sentinelone",
    "package_version": "1.0.0",
    "timestamp": "2021-12-13 11:26:02",
    "version": "1.0"
  },
  "raw": null,
  "reason": null,
  "success": true,
  "version": 2.0
}
```


▶ Example Pre-Process Script:

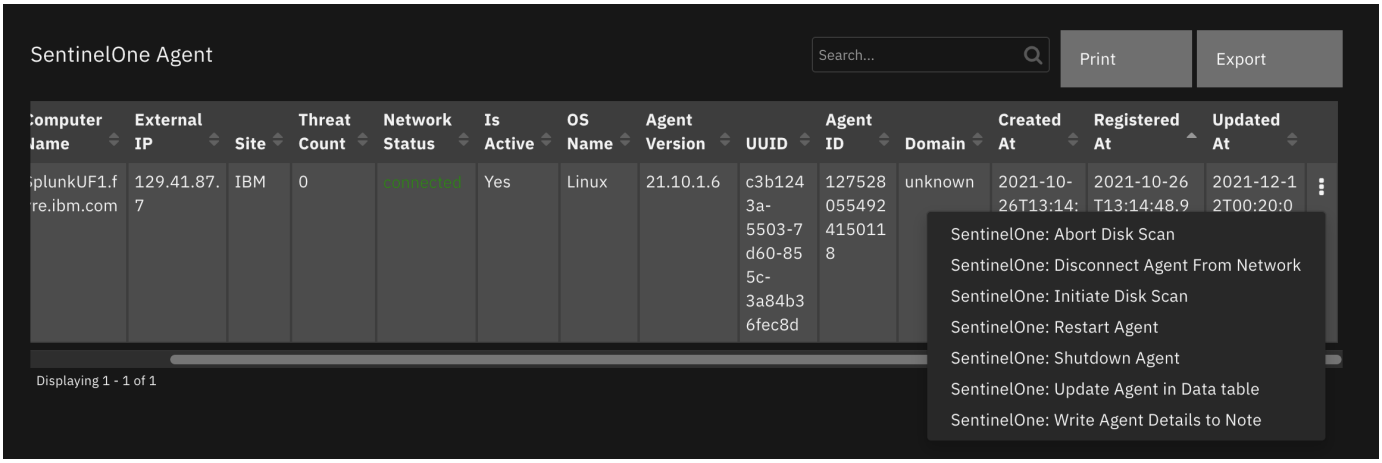
```
inputs.sentinelone_threat_id = incident.properties.sentinelone_threat_id
```

▶ Example Post-Process Script:

```
# Put the results json into a workflow property so we can call the
# convert_json_to_rich_text script to print readable formatted json in an incident note.
inputs = results.get("inputs")
threat_id = inputs.get("sentinelone_threat_id")
content = results.get("content")
data = content.get("data")

header = u"SentinelOne Threat Id: {0} Details:".format(threat_id)

json_note = {
              "version": "1.1",
              "header": header,
              "json": data,
              "sort": False
            }

workflow.addProperty('convert_json_to_rich_text', json_note)
```

Function - SentinelOne: Initiate Disk Scan

Initiate a Full Disk scan on an agent managed by SentinelOne. If the agent *isActive* field is **No**, the agent action to initiate a full disk scan is not an option.



▶ Inputs:

Name	Type	Required	Example	Tooltip
sentinelone_agent_id	text	Yes	-	-

▶ Outputs:

> **NOTE:** this example may be in JSON format, yet on the SOAR Platform `results` will be a Python Dictionary

```
results = {
  "content": {
    "data": {
      "affected": 1
    }
  },
  "inputs": {
    "sentinelone_agent_id": "1275282318251495460"
  },
  "metrics": {
    "execution_time_ms": 5678,
    "host": "myHost",
    "package": "fn-sentinelone",
    "package_version": "1.0.0",
    "timestamp": "2021-12-13 11:22:20",
    "version": "1.0"
  },
  "raw": null,
  "reason": null,
  "success": true,
  "version": 2.0
}
```

▶ Example Pre-Process Script:

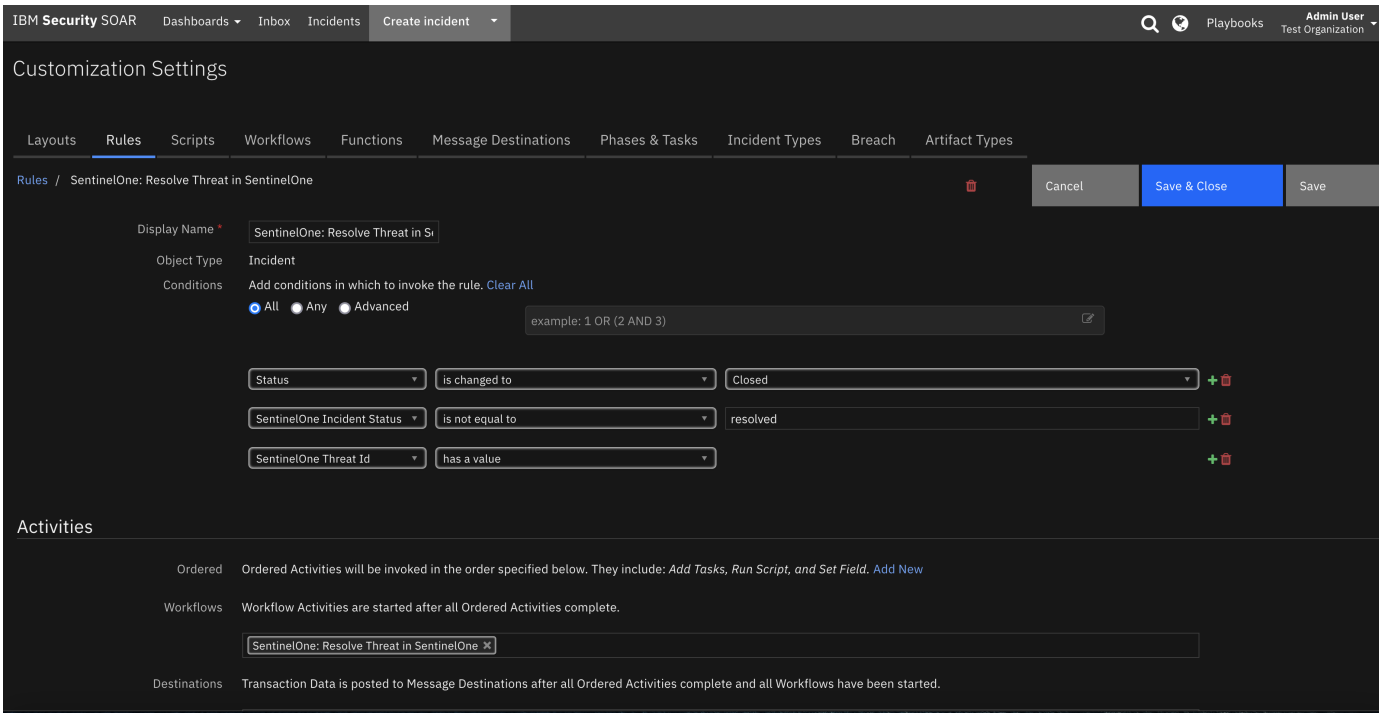
```
inputs.sentinelone_agent_id = row.sentinelone_dt_agent_id
```

▶ Example Post-Process Script:

```
so_inputs = results.get("inputs")
agent_id = so_inputs.get("sentinelone_agent_id")
note = u"<b>SentinelOne: Initiate Full Disk Scan </b><br>  SentinelOne Agent Id:
{0}".format(agent_id)
content = results.get("content")
if content:
  data = content.get("data")
  if data:
    if int(data.get("affected")) <= 0:
      note = u"{0} Full Disk Scan was NOT initiated.".format(note)
    else:
      note = u"{0} Full Disk Scan initiated.".format(note)
  else:
    note = u"{0} Full Disk Scan was NOT initiated. No 'data' returned from
function".format(note)
else:
    note = u"{0} Full Disk Scan was NOT initiated. No content returned from
function".format(note)

incident.addNote(helper.createRichText(note))
```

Function - SentinelOne: Resolve Threat in SentinelOne

Resolve (close) a threat in SentinelOne.



▶ Inputs:

Name	Type	Required	Example	Tooltip
incident_id	number	No	–	-

▶ Outputs:

> **NOTE:** this example may be in JSON format, yet on the SOAR Platform `results` will be a Python Dictionary

```
results = {
  "version": 2.0,
  "success": true,
  "reason": null,
  "content": {
    "success": true,
    "threat_id": "131313131313131313"
  },
  "raw": null,
  "inputs": {
    "incident_id": 2338
  },
  "metrics": {
    "version": "1.0",
    "package": "fn-sentinelone",
    "package_version": "1.0.0",
    "host": "myHost",
    "execution_time_ms": 1208,
    "timestamp": "2021-12-14 11:21:41"
  }
}
```

▶ Example Pre-Process Script:

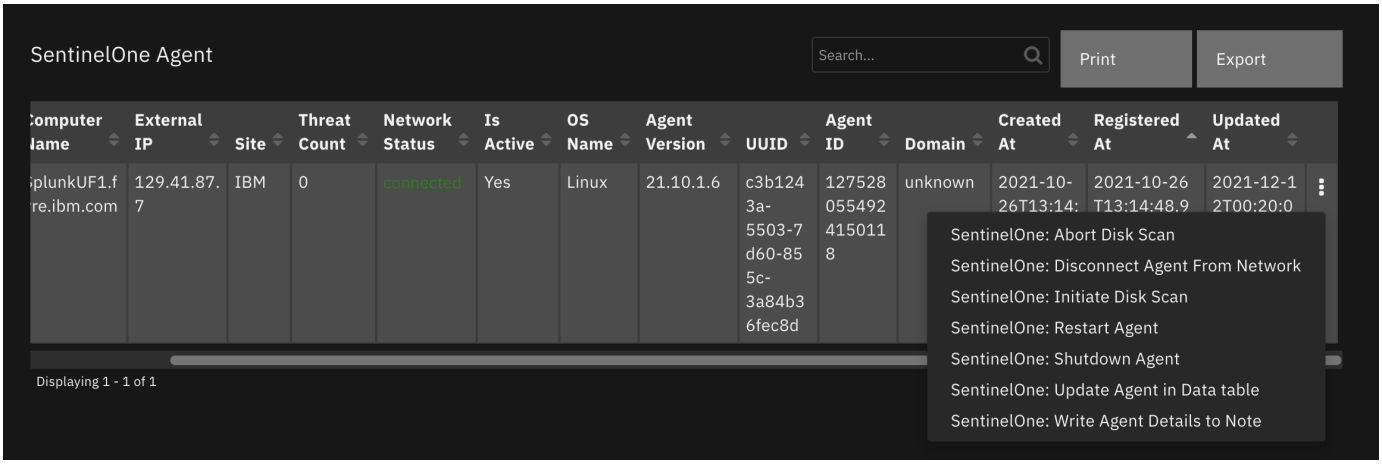
```
inputs.incident_id = incident.id
```

▶ Example Post-Process Script:

```
if results.success:
  noteText = u'SentinelOne threat {0} resolved.'.format(results.content['threat_id'])
elif:
  noteText = u'ERROR: unable to resolve SentinelOne threat
{0}.'.format(results.content['threat_id'])

incident.addNote(noteText)
```

Function - SentinelOne: Restart Agent

Restart a endpoint managed by SentinelOne. If the agent *isActive* field is **No**, the agent action to restart is not an option.



▶ Inputs:

Name	Type	Required	Example	Tooltip
sentinelone_agent_id	text	Yes	–	-

▶ Outputs:

> **NOTE:** this example may be in JSON format, yet on the SOAR Platform `results` will be a Python Dictionary

```
results = {
  "content": {
    "data": {
      "affected": 0
    }
  },
  "inputs": {
    "sentinelone_agent_id": "121212121212121212"
  },
  "metrics": {
    "execution_time_ms": 3624,
    "host": "myHost",
    "package": "fn-sentinelone",
    "package_version": "1.0.0",
    "timestamp": "2021-12-13 11:23:30",
    "version": "1.0"
  },
  "raw": null,
  "reason": null,
  "success": true,
  "version": 2.0
}
```

▶ Example Pre-Process Script:

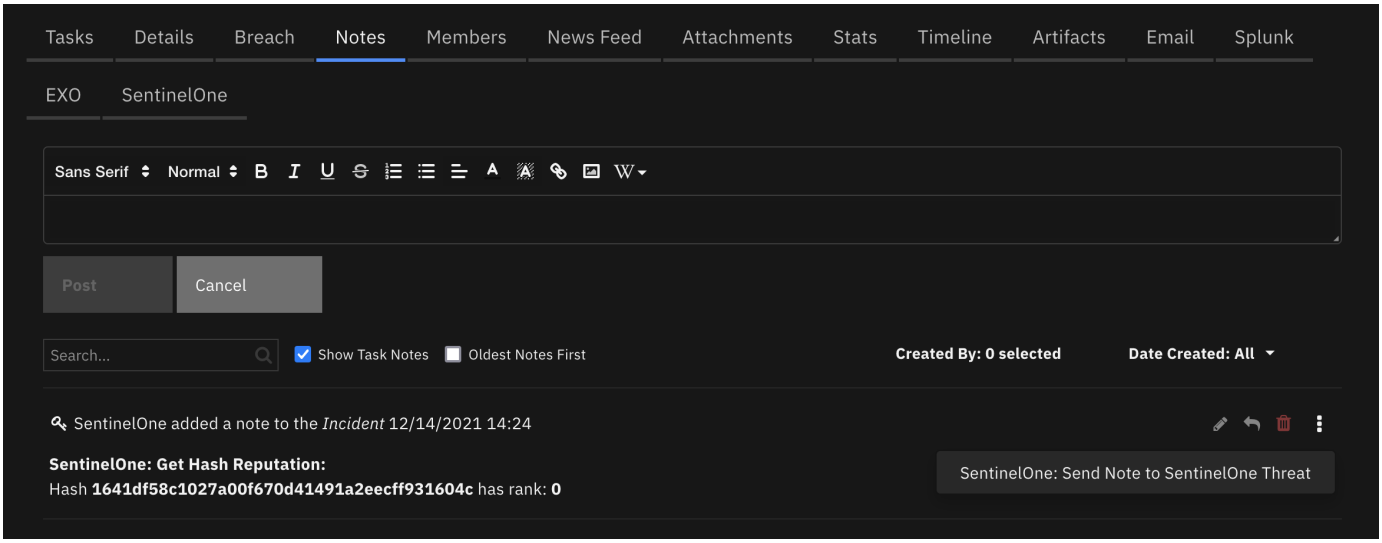
```
inputs.sentinelone_agent_id = incident.properties.sentinelone_agent_id
```

▶ Example Post-Process Script:

```
None
```

Function - SentinelOne: Send SOAR Note to SentinelOne

Send a note created in SOAR to corresponding SentinelOne threat. The example rule is a menu item rule run off a SOAR note.



▶ Inputs:

Name	Type	Required	Example	Tooltip
sentinelone_note_text	text	No	–	-
sentinelone_threat_id	text	No	–	-

▶ Outputs:

> **NOTE:** this example may be in JSON format, yet on the SOAR Platform `results` will be a Python Dictionary

```
results = {
  "content": {
    "reason:": null,
    "success": true
  },
  "inputs": {
    "sentinelone_note_text": "\u003cb\u003eSentinelOne: Get Hash Reputation:
\u003c/b\u003e\u003cbr /\u003e Hash
\u003cb\u003e1641df58c1027a00f670d41491a2eecff931604c\u003c/b\u003e has rank:
\u003cb\u003e0\u003c/b\u003e",
    "sentinelone_threat_id": "1308905355630511064"
  },
  "metrics": {
    "execution_time_ms": 5590,
    "host": "myHost",
    "package": "fn-sentinelone",
    "package_version": "1.0.0",
```

```
    "timestamp": "2021-12-13 11:28:04",
    "version": "1.0"
  },
  "raw": null,
  "reason": null,
  "success": true,
  "version": 2.0
}
```


▶ Example Pre-Process Script:

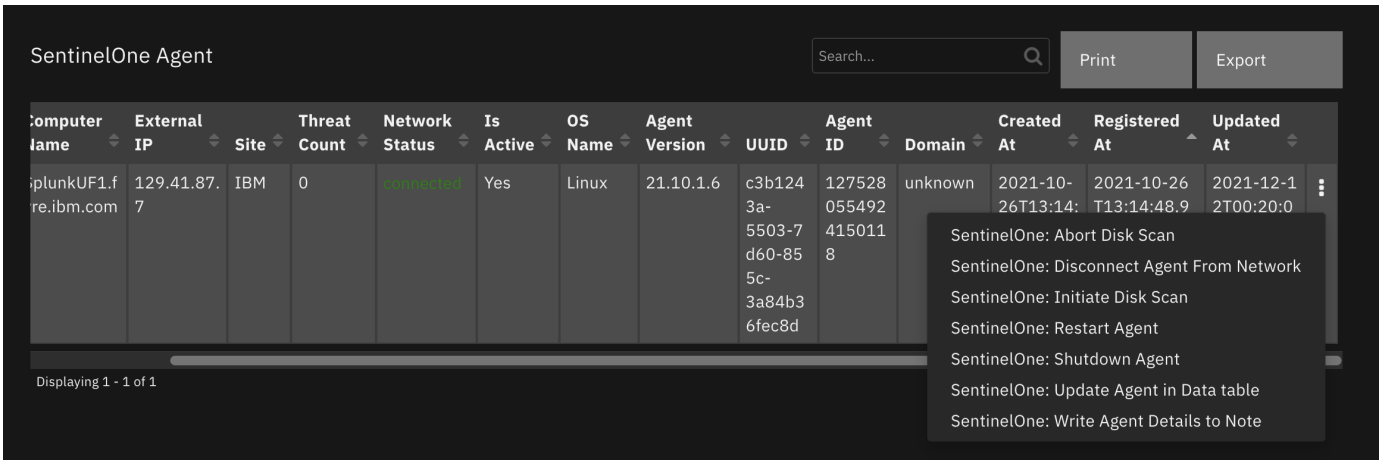
```
inputs.sentinelone_threat_id = incident.properties.sentinelone_threat_id
inputs.sentinelone_note_text = note.text.content
```

▶ Example Post-Process Script:

```
# Import Date
from java.util import Date

# Edit note in SOAR to indicate it was sent to SentinelOne
if results.success:
  # Get the current time
  dt_now = Date()
  note.text = u"<b>Sent to SentinelOne at {0}</b><br>{1}".format(dt_now,
unicode(note.text.content))
```

Function - SentinelOne: Shutdown Agent

Shutdown an agent managed by SentinelOne. If the agent *isActive* field is **No**, the agent action to shutdown is not an option.



▶ Inputs:

Name	Type	Required	Example	Tooltip
sentinelone_agent_id	text	Yes	–	-

▶ Outputs:

> **NOTE:** this example may be in JSON format, yet on the SOAR Platform `results` will be a Python Dictionary


```
results = {
  "content": {
    "data": {
```

```
      "affected": 0
    }
  },
  "inputs": {
    "sentinelone_agent_id": "1275282318251495460"
  },
  "metrics": {
    "execution_time_ms": 3426,
    "host": "myHost",
    "package": "fn-sentinelone",
    "package_version": "1.0.0",
    "timestamp": "2021-12-13 11:23:17",
    "version": "1.0"
  },
  "raw": null,
  "reason": null,
  "success": true,
  "version": 2.0
}
```


▶ Example Pre-Process Script:

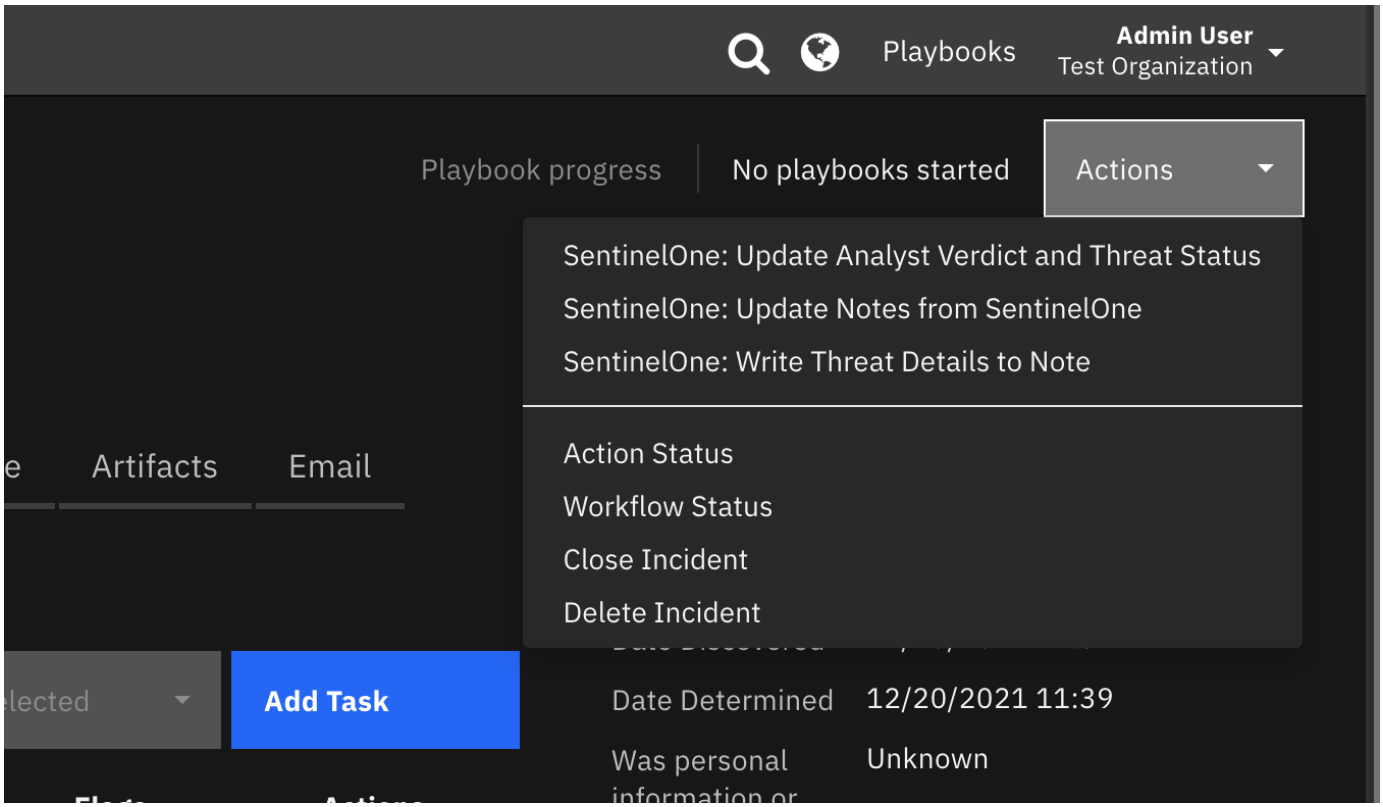
```
inputs.sentinelone_agent_id = row.sentinelone_dt_agent_id
```

▶ Example Post-Process Script:

```
so_inputs = results.get("inputs")
agent_id = so_inputs.get("sentinelone_agent_id")
note = u"<b>SentinelOne: Shutdown Agent </b><br>  SentinelOne Agent Id: {0}".format(agent_id)
content = results.get("content")
if content:
  data = content.get("data")
  if data:
    if int(data.get("affected")) <= 0:
      note = u"{0} Agent was NOT shutdown.".format(note)
    else:
      note = u"{0} Agent shutdown initiated.".format(note)
  else:
    note = u"{0} Agent shutdown was NOT initiated. No 'data' returned from
function".format(note)
else:
    note = u"{0} Agent shutdown was NOT initiated. No content returned from
function".format(note)

incident.addNote(helper.createRichText(note))
```

Function - SentinelOne: Update Notes From SentinelOne

Query SentinelOne threat and add any new threat notes to the SOAR incident.



▶ Inputs:

Name	Type	Required	Example	Tooltip
incident_id	number	No	–	-
sentinelone_threat_id	text	No	–	-

▶ Outputs:

> **NOTE:** this example may be in JSON format, yet on the SOAR Platform `results` will be a Python Dictionary

```
results = {
  "content": {
    "notes_created": 1,
    "success": true
  },
  "inputs": {
    "incident_id": 2334,
    "sentinelone_threat_id": "1308905355630511064"
  },
  "metrics": {
    "execution_time_ms": 1668,
    "host": "myHost",
    "package": "fn-sentinelone",
    "package_version": "1.0.0",
    "timestamp": "2021-12-13 11:28:59",
    "version": "1.0"
  },
  "raw": null,
  "reason": null,
  "success": true,
  "version": 2.0
}
```

▶ Example Pre-Process Script:

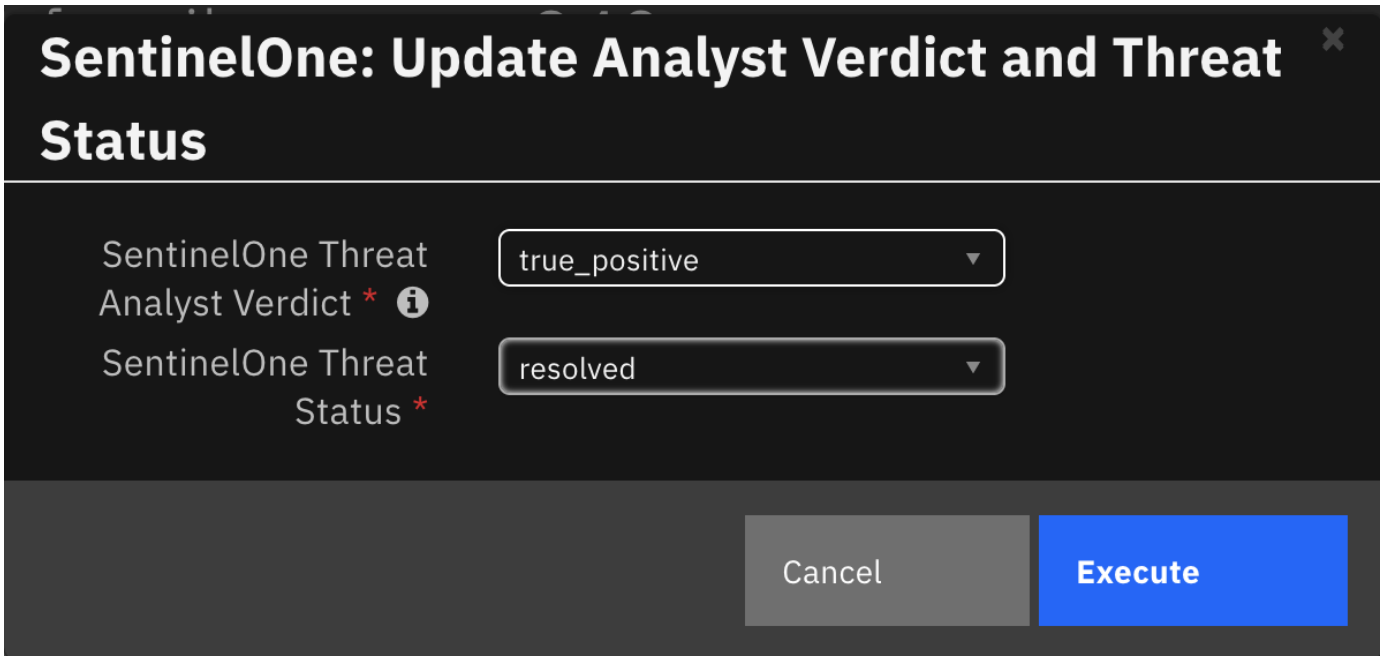
```
inputs.incident_id = incident.id
inputs.sentinelone_threat_id = incident.properties.sentinelone_threat_id
```

▶ Example Post-Process Script:

```
None
```

Function - Sentinelone: Update Threat Status

Update the analyst verdict and the status of a threat in SentinelOne.



▶ Inputs:

Name	Type	Required	Example	Tooltip
sentinelone_threat_id	text	Yes	–	-
sentinelone_threat_status	select	Yes	–	-
sentinelone_threat_analyst_verdict	select	Yes	–	-

▶ Outputs:

> **NOTE:** this example may be in JSON format, yet on the SOAR Platform `results` will be a Python Dictionary


```
results = {
  "content": {
    "data": {
      "affected": 1
    }
  },
  "inputs": {
    "sentinelone_threat_id": "1308905355630511064",
    "sentinelone_threat_status": "in_progress"
  },
  "metrics": {
    "execution_time_ms": 5215,
    "host": "myHost",
    "package": "fn-sentinelone",
```

```
    "package_version": "1.0.0",
    "timestamp": "2021-12-13 11:25:54",
    "version": "1.0"
  },
  "raw": null,
  "reason": null,
  "success": true,
  "version": 2.0
}
```


▶ Example Pre-Process Script:

```
inputs.sentinelone_threat_id = incident.properties.sentinelone_threat_id
inputs.sentinelone_threat_status = rule.properties.sentinelone_threat_status
```

▶ Example Post-Process Script:

```
so_inputs = results.get("inputs")
threat_id = so_inputs.get("sentinelone_threat_id")
status = so_inputs.get("sentinelone_threat_status")
note = u"<b>SentinelOne: Update Threat Status </b><br>  SentinelOne Threat Id:
{0}".format(threat_id)
content = results.get("content")
if content:
  data = content.get("data")
  if data:
    if int(data.get("affected")) <= 0:
      note = u"{0} Threat Status <b>{1}</b> NOT updated in SentinelOne threat.".format(note,
status)
    else:
      note = u"{0} Threat Status <b>{1}</b> updated in SentinelOne threat".format(note,
status)
  else:
    note = u"{0} Threat Status <b>{1}</b> NOT updated. No 'data' returned from
function".format(note, status)
else:
    note = u"{0} Threat Status <b>{1}</b> NOT updated. No content returned from
function".format(note, status)

incident.addNote(helper.createRichText(note))
```

Script - Convert JSON to rich text v1.1

This script converts a json object into a hierarchical display of rich text and adds the rich text to an incident's rich text (custom) field or an incident note. A workflow property is used to share the json to convert and identify parameters used on how to perform the conversion.

Typically, a function will create the workflow property 'convert_json_to_rich_text', and this script will run after that function to perform the conversion.

Features:

- Display the hierarchical nature of json, presenting the json keys (sorted if specified) as bold labels
- Provide links to found URLs
- Create either an incident note or add results to an incident (custom) rich text field.

Object: incident

▶ Script Text:

```
# (c) Copyright IBM Corp. 2010, 2020. All Rights Reserved.
VERSION = 1.1
"""
  This script converts a json object into a hierarchical display of rich text and adds the
rich text to an incident's rich text (custom) field or an incident note.
  A workflow property is used to define the json to convert and identify parameters used on
how to perform the conversion.
  Typically, a function will create workflow property and this script will run after that
function to perform the conversion.
  Features:
    * Display the hierarchical nature of json, presenting the json keys as bold labels
    * Provide links to found URLs
    * Create either an incident note or add results to an incident (custom) rich text field.

  In order to use this script, define a workflow property called: convert_json_to_rich_text,
to define the json and parameters to use for the conversion.
  Workflow properties can be added using a command similar to this:
  workflow.addProperty('convert_json_to_rich_text', {
    "version": 1.1,
    "header": "Artifact scan results for: {}".format(artifact.value),
    "padding": 10,
    "separator": u"<br />",
    "sort": True,
    "json": results.content,
    "json_omit_list": ["omit"],
    "incident_field": None
  })

  Format of workflow.property.convert_json_to_rich_text:
  {
    "version": 1.1, [this is for future compatibility]
    "header": str, [header line to add to converted json produced or None. Ex: Results from
scanning artifact: xxx. The header may contain rich text tags]
    "padding": 10, [padding for nested json elements, or defaults to 10]
    "separator": u"<br />"|list such as ['<span>','</span>'], [html separator between json
keys and lists or defaults to html break: '<br />'.
                                                If a list, then the data is brackets by the
pair specified]
    "sort": True|False, [sort the json keys at each level when displayed]
    "json": json, [required json to convert]
    "json_omit_list": [list of json keys to exclude or None]
    "incident_field": "<incident_field>" [indicates a builtin rich text incident field, such
as 'description'
                                            or a custom rich text field in the format:
'properties.<field>'. default: create an incident note]
  }
"""

import re

# needed for python 3
try:
    unicode("abc")
except:
    unicode = str


rc = re.compile(r'http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+#\?]|[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-
F]))+')

class ConvertJson:
    """Class to hold the conversion parameters and perform the conversion"""

    def __init__(self, omit_keys=[], padding=10, separator=u"<br />", sort_keys=False):
        self.omit_keys = omit_keys
```

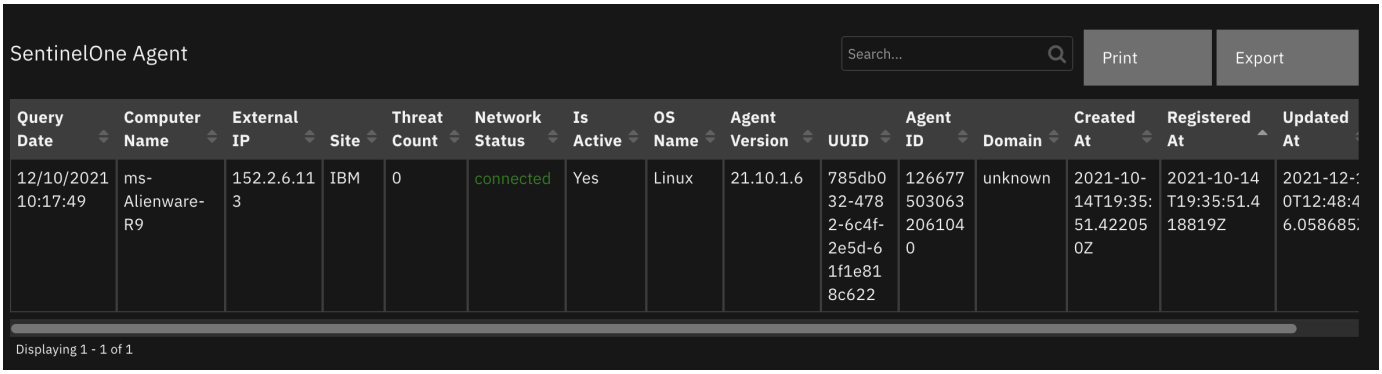
```
        self.padding = padding
        self.separator = separator
        self.sort_keys = sort_keys


    def format_link(self, item):
        """[summary]
          Find embedded urls (http(s)) and add html anchor tags to display as links
          Args:
              item ([string])

          Returns:
              [str]: None|original text if no links|text with html links
        """
        formatted_item = item
        if item and not isinstance(item, (int, bool, float)):
            list = rc.findall(item)
            if list:
                for link in list:
                    formatted_item = formatted_item.replace(link, u"<a target='blank'
href='{0}'>{0}</a>".format(link))

        return formatted_item

    def expand_list(self, list_value, is_list=False):
        """[summary]
          convert items to html, adding indents to nested dictionaries.
          Args:
              list_value ([dict|list]): json element

          Returns:
              [str]: html converted code
        """
        if not isinstance(list_value, list):
            return self.format_link(list_value)
        elif not list_value:
            return u"None<br>"

        try:
            items_list = []  # this will ensure list starts on second line of key label
            for item in list_value:
                if isinstance(item, dict):
                    result = self.convert_json_to_rich_text(item)
                    if is_list:
                        items_list.append(u"<li>{}</li>".format(result))
                    else:
                        items_list.append(result)
                elif isinstance(item, list):
                    items_list.append(self.expand_list(item, is_list=True))
                elif is_list:
                    items_list.append(u"<li>{}</li>".format(self.format_link(unicode(item))))
                else:
                    items_list.append(self.format_link(unicode(item)))

            expand_list_result = self.add_separator(self.separator if not is_list else u"",
                                                    items_list,
                                                    is_list=is_list)

            if is_list:
                return u"<ul>{}</ul>".format(expand_list_result)
            else:
                return u"<div style='padding:5px'>{}</div>".format(expand_list_result)
        except Exception as err:
            return str(err)

    def convert_json_to_rich_text(self, sub_dict):
```

```
        """[summary]
          Walk dictionary tree and convert to html for better display
          Args:
              sub_dict ([type]): [description]

          Returns:
              [type]: [description]
        """
        notes = []
        if sub_dict:
            if isinstance(sub_dict, list):
                expanded_list = self.expand_list(sub_dict, is_list=True)
                notes.append(self.add_separator(self.separator, expanded_list))
            else:
                keys = sorted (sub_dict.keys()) if self.sort_keys else sub_dict.keys()

                for key in keys:
                    if key not in self.omit_keys:
                        value = sub_dict[key]
                        is_list = isinstance(value, list)
                        item_list = [u"<strong>{0}</strong>: ".format(key)]
                        if isinstance(value, dict):
                            convert_result = self.convert_json_to_rich_text(value)
                            if convert_result:
                                item_list.append(u"<div style='padding:{}px'>{}
</div>".format(self.padding, convert_result))
                            else:
                                item_list.append(u"None<br>")
                        else:
                            item_list.append(self.expand_list(value, is_list=is_list))
                        notes.append(self.add_separator(self.separator, u"".join(unicode(v)
for v in item_list), is_list=is_list))

        result_notes = u"".join(notes)
        if isinstance(self.separator, list):
            return result_notes
        else:
            return result_notes.replace(
                u"</div>{0}".format(self.separator), u"</div>").replace(
                u"{0}</div>".format(self.separator), u"</div>"
            )  # tighten up result

    def add_separator(self, separator, items, is_list=False):
        """
        apply the separator to the data
        :param separator: None, str or list such as ['<span>', '</span>']
        :param items: str or list to add separator
        :return: text with separator applied
        """
        _items = items

        if not _items:
            return "<br>"

        if not isinstance(_items, list):
            _items = [_items]

        if isinstance(separator, list):
            return u"".join([u"{}{}{}".format(separator[0], item, separator[1]) for item in
_items])

        return u"{}{}".format(separator.join(_items), separator if not is_list else u"")

def get_properties(property_name):
    """
    Logic to collect the json and parameters from a workflow property.
```

```
    Args:
      property_name: workflow property to reference
    Returns:
      padding, separator, header, json_omit_list, incident_field, json, sort_keys
    """
    if not workflow.properties.get(property_name):
        helper.fail("workflow.properties.{} undefined".format(property_name))

    padding = int(workflow.properties[property_name].get("padding", 10))
    separator = workflow.properties[property_name].get("separator", u"<br />")
    if isinstance(separator, list) and len(separator) != 2:
        helper.fail("list of separators should be specified as a pair such as ['<div>',
'</div>']: {}".format(separator))

    header = workflow.properties[property_name].get("header")
    json_omit_list = workflow.properties[property_name].get("json_omit_list")
    if not json_omit_list:
        json_omit_list = []
    incident_field = workflow.properties[property_name].get("incident_field")
    json = workflow.properties[property_name].get("json", {})
    if not isinstance(json, dict) and not isinstance(json, list):
        helper.fail("json element is not formatted correctly: {}".format(json))
    sort_keys = bool(workflow.properties[property_name].get("sort", False))

    return padding, separator, header, json_omit_list, incident_field, json, sort_keys


## S T A R T
if 'workflow' in globals():
    padding, separator, header, json_omit_list, incident_field, json, sort_keys =
get_properties('convert_json_to_rich_text')

    if header:
        if isinstance(separator, list):
            hdr = u"{0}{1}{2}".format(separator[0], header, separator[1])
        else:
            hdr = u"{0}{1}".format(header, separator)
    else:
        hdr = u""

    convert = ConvertJson(omit_keys=json_omit_list, padding=padding, separator=separator,
sort_keys=sort_keys)
    converted_json = convert.convert_json_to_rich_text(json)
    result = u"{}{}".format(hdr, converted_json if converted_json else "\nNone")

    rich_text_note = helper.createRichText(result)
    if incident_field:
        incident[incident_field] = rich_text_note
    else:
        incident.addNote(rich_text_note)
```

Data Table - SentinelOne Agent

Here is an example of the SentinelOne Agent data table:



API Name:

sentinelone_agents_dt

Columns:

Column Name	API Access Name	Type	Tooltip
Agent ID	sentinelone_dt_agent_id	text	-
Agent Version	sentinelone_dt_agent_version	text	-
Computer Name	sentinelone_dt_computername	text	-
Created At	sentinelone_dt_created	text	-
Domain	sentinelone_dt_domain	text	-
External IP	sentinelone_dt_external_ip	text	-
Is Active	sentinelone_dt_is_active	boolean	-
Network Status	sentinelone_dt_network_status	textarea	-
OS Name	sentinelone_dt_os_name	text	-
Query Date	sentinelone_dt_query_date	datetimepicker	-
Registered At	sentinelone_dt_registered	text	-
Site	sentinelone_dt_site	text	-
Threat Count	sentinelone_dt_threat_count	number	-
Updated At	sentinelone_dt_updated	text	-
UUID	sentinelone_dt_uuid	text	-

Custom Fields

Label	API Access Name	Type	Prefix	Placeholder	Tooltip
SentinelOne Agent Id	sentinelone_agent_id	text	properties	-	-
SentinelOne Classification	sentinelone_classification	text	properties	-	-
SentinelOne Threat Confidence Level	sentinelone_confidence_level	text	properties	-	-

Label	API Access Name	Type	Prefix	Placeholder	Tooltip
SentinelOne Incident Status	sentinelone_incident_status	text	properties	-	-
SentinelOne Mitigation Status	sentinelone_mitigation_status	text	properties	-	-
Sentinelone Mitigation Status Description	sentinelone_mitigation_status_description	text	properties	-	-
Sentinelone Threat Analyst Verdict	sentinelone_threat_analyst_verdict	select	properties	-	SentinelOne threat analyst verdict
SentinelOne Threat Id	sentinelone_threat_id	text	properties	-	-
SentinelOne Threat Name	sentinelone_threat_name	text	properties	-	-
SentinelOne Threat Overview URL	sentinelone_threat_overview_url	textarea	properties	-	-

Rules

Rule Name	Object	Workflow Triggered
SentinelOne: Abort Disk Scan	sentinelone_agents_dt	sentinelone_abort_disk_scan
SentinelOne: Add Agent to Data Table	incident	sentinelone_add_agent_to_data_table
SentinelOne: Connect Agent to Network	sentinelone_agents_dt	sentinelone_connect_to_network
SentinelOne: Disconnect Agent From Network	sentinelone_agents_dt	sentinelone_disconnect_from_network
SentinelOne: Get Hash Reputation	artifact	sentinelone_get_hash_reputation
SentinelOne: Initiate Disk Scan	sentinelone_agents_dt	sentinelone_initiate_disk_scan
SentinelOne: Resolve Threat in SentinelOne	incident	sentinelone_resolve_threat_in_sentinelone
SentinelOne: Restart Agent	sentinelone_agents_dt	sentinelone_restart_agent
SentinelOne: Send Note to SentinelOne Threat	note	sentinelone_send_soar_note_to_sentinelone
SentinelOne: Send SOAR Note to SentinelOne	note	sentinelone_send_soar_note_to_sentinelone
SentinelOne: Shutdown Agent	sentinelone_agents_dt	sentinelone_shutdown_agent
SentinelOne: Update Agent in Data table	sentinelone_agents_dt	sentinelone_update_agent_in_data_table
SentinelOne: Update Notes from SentinelOne	incident	sentinelone_update_notes_from_sentinelone
SentinelOne: Update Analyst Verdict and Threat Status	incident	sentinelone_update_threat_status
SentinelOne: Write Agent Details to Note	sentinelone_agents_dt	sentinelone_write_agent_details_to_note

Rule Name	Object	Workflow Triggered
SentinelOne: Write Threat Details to Note	incident	`sentinelone_write_threat_details_to_note`

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.

Template Appendix

Below are examples of templates for creating, updating, and closing IBM SOAR incidents. Customize these templates and refer to them in the app.config file. These default jinja templates map IBM SOAR fields to SentinelOne threat and agent fields.

Each template should be reviewed for correctness in your enterprise. For instance, closing a SOAR incident may include additional custom fields which the default template does not include.

- ▶ incident_creation_template
- ▶ incident_close_template
- ▶ incident_update_template