# ReaQta for IBM QRadar SOAR

## Table of Contents

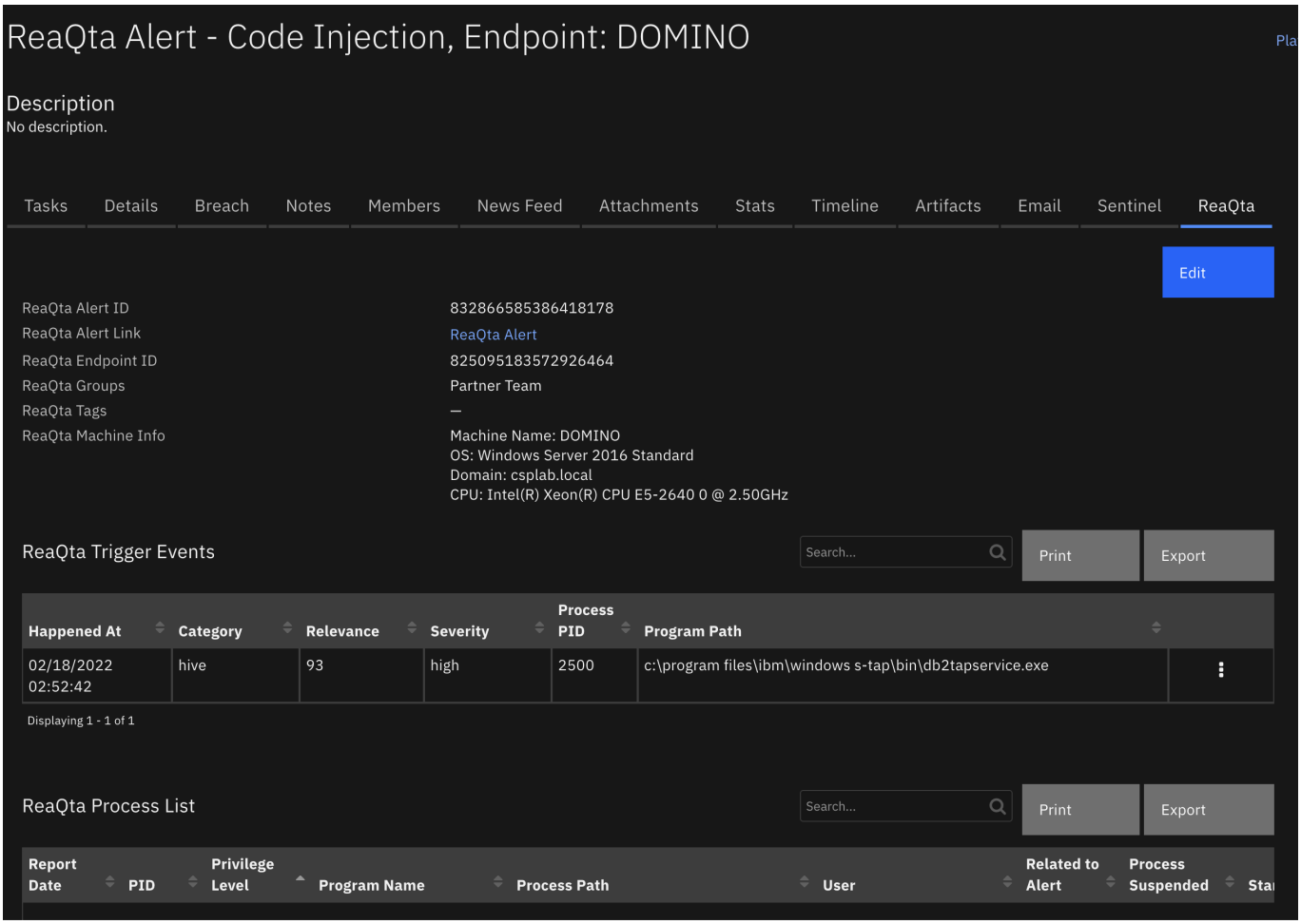


## Release Notes

| Version | Date | Notes |
|---|---|---|
| 1.0.0 | 03/2022 | Initial Release |

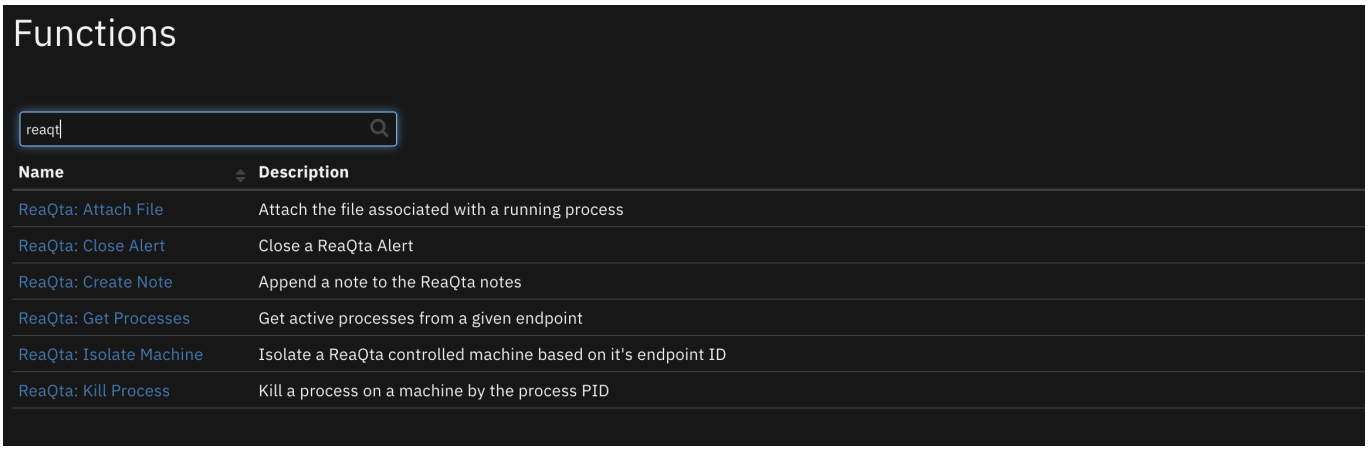
## Overview

**IBM SOAR app for ReaQta**




Bidirectional synchronization of ReaQta Alerts to IBM SOAR.

```
Additional functions exists to list and kill endpoint processes, isolate
the endpiont and synchronize notes and close events.
```




## Key Features

- Sync alerts to SOAR cases via user defined filter criteria
- Sync SOAR case notes to ReaQta
- ReaQta alert closing closes the SOAR case and SOAR case closing closes the ReaQta alert
- Get running processing

- Download ReaQta endpoint process file to an attachment
- Kill ReaQta endpoint process
- Isolate ReaQta endpoint

---

## Requirements

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

### IBM SOAR platform

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

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

- IBM SOAR platform >= `41.2.41`.
- The app is in a container-based format (available from the AppExchange as a `zip` file).

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

- IBM SOAR platform >= `41.2.41`.
- 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 |
| Incidents | Create, Read, Edit, Edit Status |

The following IBM 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 IBM SOAR platform version. On the follow-on page, you can find the *App Host Deployment Guide* or *Integration Server Guide* by expanding **IBM SOAR 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 >= 1.5.
- 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 **does** support a proxy server.

## Python Environment

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

- cachetools
- resilient-circuits>=43.0.0
- resilient-lib

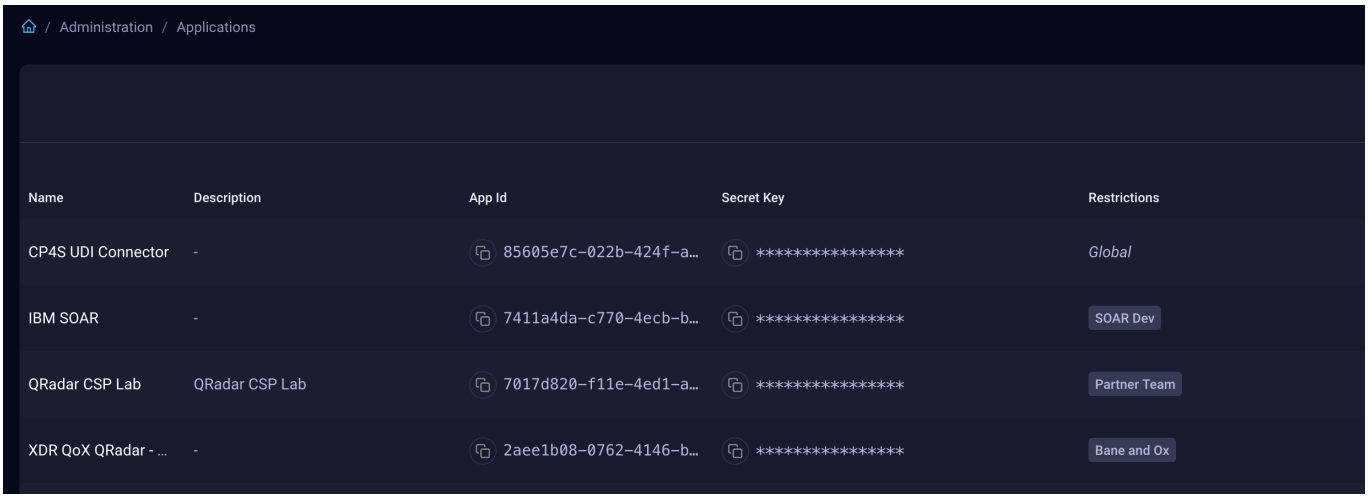
## Endpoint Developed With

This app has been implemented using:

| Product Name | Product Version | API URL | API Version |
|---|---|---|---|
| ReaQta Hive | 1.0 | rqt-api/1/ | 1 |

### Configuration

In order to make API calls to ReaQta, create an API Application, prodiving the endpoint group restrictions as appropiate. The API ID and secret will be copied into your app.config file




# Installation

## Install

- To install or uninstall an App or Integration on the *IBM 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.

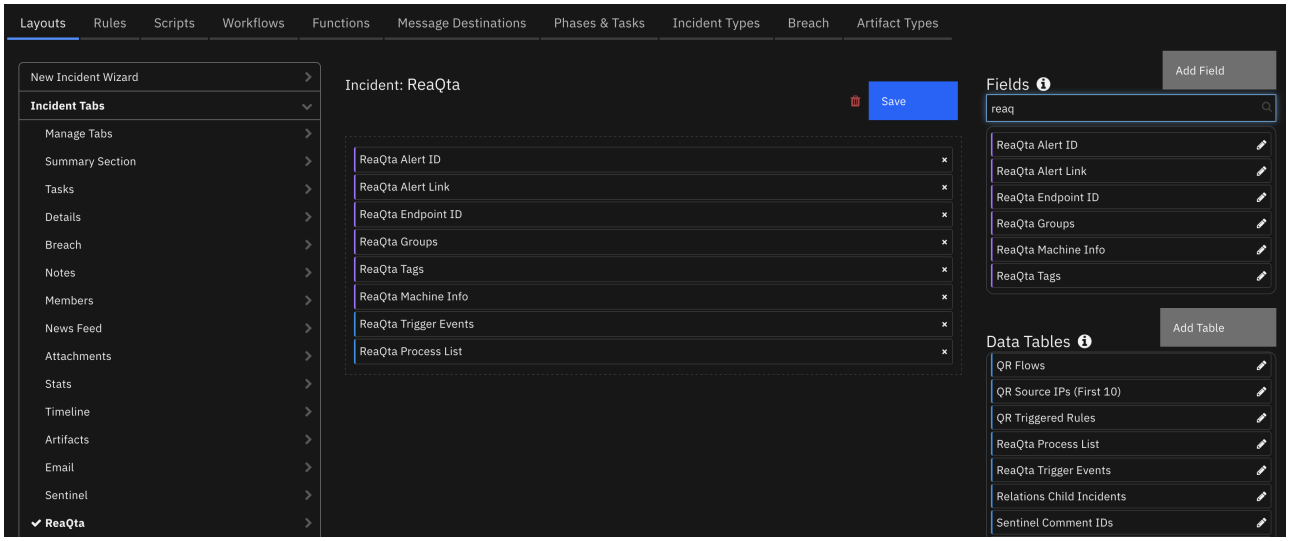
## App Configuration

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 |
|---|---|---|---|
| **api_key** | Yes | `7411a4da-c770-...` | *API Key ID from your configured ReaQta API application* |
| **api_secret** | Yes | `P9zPLkcb-...` | *API Key secret from your configured ReaQta API application* |
| **api_version** | Yes | `rqt_api/1/` | *url path information ending in slash '/'* |
| **cafile** | Yes | `/path/to/cafile.crt or false` | *path to your ReaQta client certification, if needed or false for no certificate verification* |
| **poller_interval** | Yes | `60` | *Number of seconds between polling queries for new alerts * |
| **poller_lookback** | Yes | `120` | *Number of minutes to look back for new alerts the first time the app starts or restarts* |
| **reaqta_url** | Yes | `https://xxx/` | *Base URL to ReaQta instance ending in slash '/'* |

### Custom Layouts

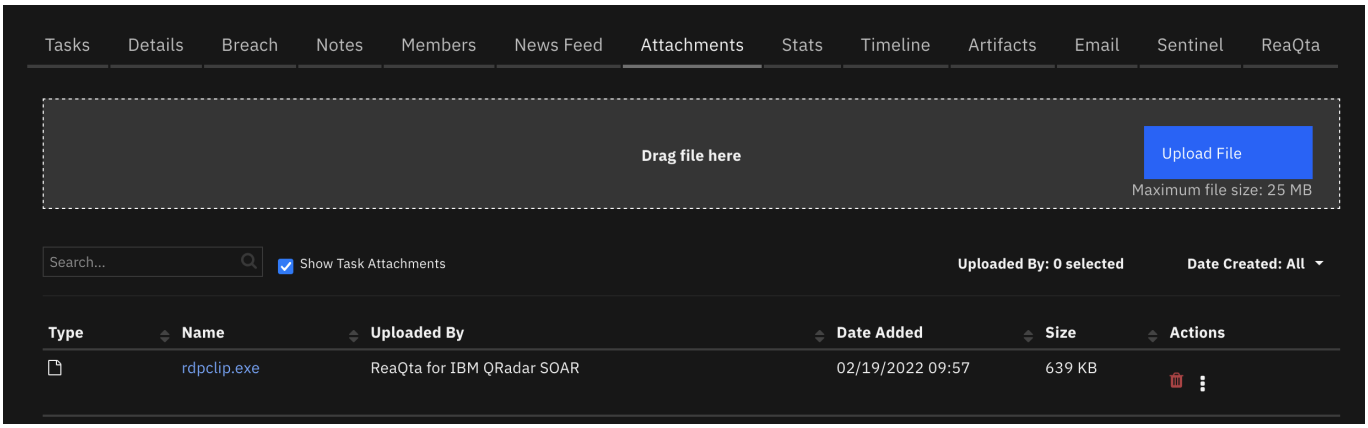
- Import the Data Tables and Custom Fields like the screenshot below:




---

## Function - ReaQta: Attach File

Create a SOAR case attachment associated with a running process




▶ Inputs:

| Name | Type | Required | Example | Tooltip |
|---|---|---|---|---|
| `reaqta_endpoint_id` | `text` | Yes | – | - |
| `reaqta_incident_id` | `number` | Yes | – | - |
| `reaqta_program_path` | `text` | Yes | – | typically taken from the reaqta_trigger_events or reaqta_process_list datatables |

▶ Outputs:

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

```
results = {
  "content": {
    "actions": [
      {
        "enabled": true,
        "id": 90,
        "name": "ReaQta: Kill Process"
      }
    ],
    "content_type": "application/x-msdownload",
    "created": 1645210092202,
    "creator_id": 3,
    "id": 48,
    "inc_id": 2377,
    "inc_name": "ReaQta Alert - Hive alert Title, Endpoint: AUTISTIC1",
    "inc_owner": 3,
    "name": "notepad.exe",
    "size": 334262,
    "task_at_id": null,
    "task_custom": null,
    "task_id": null,
    "task_members": null,
    "task_name": null,
    "type": "incident",
    "uuid": "50df0c2a-e222-4353-90f0-f4f2a2fad3f5",
    "vers": 7
  },
  "inputs": {
    "reaqta_endpoint_id": "831986736375529472",
    "reaqta_incident_id": 2377,
    "reaqta_program_path": "C:\\Windows\\System32\\notepad.exe"
  },
  "metrics": {
    "execution_time_ms": 18981,
    "host": "endpoint.local",
    "package": "fn-reaqta",
    "package_version": "1.0.0",
    "timestamp": "2022-02-18 13:48:11",
    "version": "1.0"
  },
  "raw": null,
  "reason": null,
  "success": true,
  "version": 2.0
}
```


▶ Example Pre-Process Script:

```
inputs.reaqta_program_path = row['process_path'].replace("\\\\", "\\")
inputs.reaqta_endpoint_id = incident.properties.reaqta_endpoint_id
inputs.reaqta_incident_id = incident.id
```

▶ Example Post-Process Script:

```
if results.success:
  incident.addNote("ReaQta Attach File created: {} from program path:
{}".format(results.content['name'],
results.inputs['reaqta_program_path']))
else:
  incident.addNote("ReaQta Attach File failed: {}".format(results.reason))
```

---

## Function - ReaQta: Close Alert

Close a ReaQta alert. This can be triggered when the SOAR case is closed.

▶ Inputs:

| Name | Type | Required | Example | Tooltip |
|---|---|---|---|---|
| reaqta_alert_id | text | Yes | – | - |
| reaqta_is_malicious | boolean | Yes | False | true/ false for malicious/benign |
| reaqta_note | text | No | – | - |

▶ Outputs:

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

```
results = {
  "content": {
    "alertId": "831993910053044226",
    "closed": true
  },
  "inputs": {
    "reaqta_alert_id": "831993910053044226",
    "reaqta_is_malicious": false,
    "reaqta_note": "\u003cdiv
class=\"rte\"\u003e\u003cdiv\u003e\u003cstrong\u003ethis is now
complete\u003c/strong\u003e\u003c/div\u003e\u003c/div\u003e"
  },
  "metrics": {
    "execution_time_ms": 1065,
    "host": "endpoint.local",
    "package": "fn-reaqta",
    "package_version": "1.0.0",
    "timestamp": "2022-02-18 13:54:56",
    "version": "1.0"
  },
  "raw": null,
```

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

▶ Example Pre-Process Script:

```
# Modify this table for custom resolution types
IS_MALICIOUS_LOOKUP = {
  7: False, # Unresolved
  8: False, # Duplicate
  9: False, # Not a Issue
  10: True  # Resolved
}

inputs.reaqta_alert_id = incident.properties.reaqta_id
inputs.reaqta_note = incident.resolution_summary.content
inputs.reaqta_is_malicious =
IS_MALICIOUS_LOOKUP.get(incident.resolution_id, False) # if resolution_id
is not found, set to not malicious
```

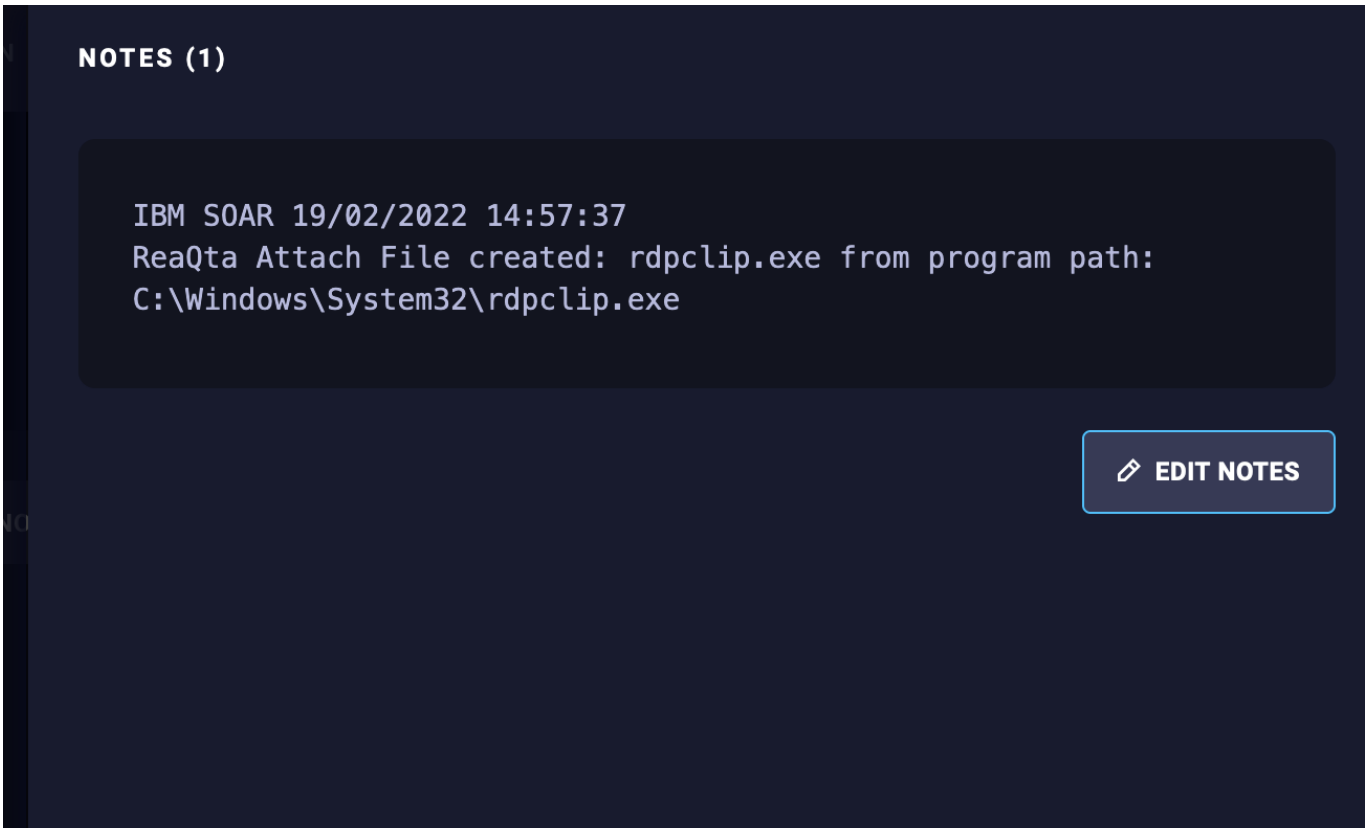
▶ Example Post-Process Script:

```
None
```

---

## Function - ReaQta: Create Note

Append a note to the ReaQta notes. Notes will display in ReaQta with the 'IBM SOAR header.




▶ Inputs:

| Name | Type | Required | Example | Tooltip |
|---|---|---|---|---|
| reaqta_alert_id | text | Yes | - | - |
| reaqta_note | text | No | - | - |

▶ Outputs:

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

```
results = {
  "content": "this is a note in the hive alert\nIBM SOAR 18/02/2022
14:04:57\nReaQta Isolate Machine failed: Endpoint offline",
  "inputs": {
    "reaqta_alert_id": "830607817638412290",
    "reaqta_note": "ReaQta Isolate Machine: Endpoint offline"
  },
  "metrics": {
    "execution_time_ms": 738,
    "host": "endpoint.local",
    "package": "fn-reaqta",
    "package_version": "1.0.0",
    "timestamp": "2022-02-18 14:04:58",
    "version": "1.0"
  },
  "raw": null,
  "reason": null,
  "success": true,
```

```
  "version": 2.0
}
```

▶ Example Pre-Process Script:

```
inputs.reaqta_alert_id = incident.properties.reaqta_id
inputs.reaqta_note = note.text.content
```

▶ Example Post-Process Script:

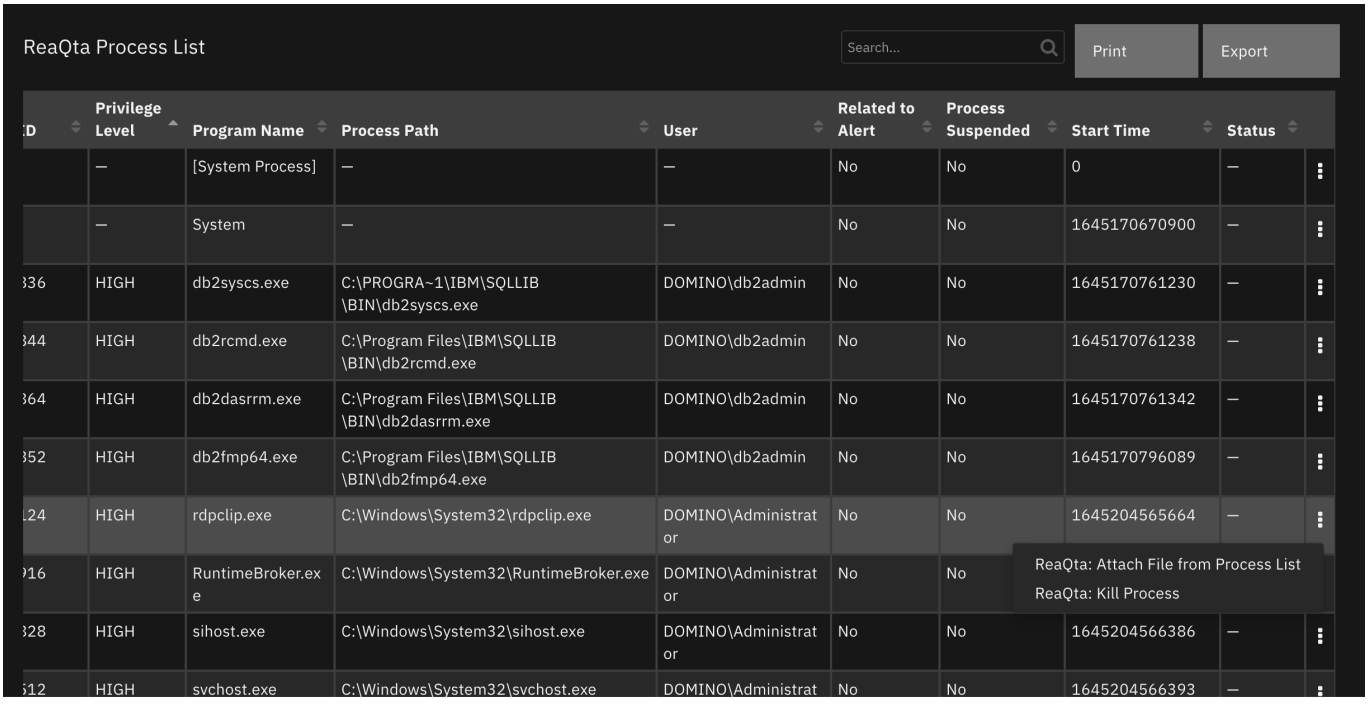
```
from java.util import Date

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

---

## Function - ReaQta: Get Processes

Get active processes from a given endpoint and present in the reaqta_process_list datatable. Two filters are available to reduce the size of the process list, as the list can be quite long.




▶ Inputs:

| Name | Type | Required | Example | Tooltip |
|---|---|---|---|---|
| reaqta_endpoint_id | text | Yes | – | - |

| Name | Type | Required | Example | Tooltip |
|---|---|---|---|---|
| `reaqta_has_incident` | `boolean` | No | `true` | Select only processes associated with an alert |
| `reaqta_suspended` | `boolean` | No | `-` | Select only processes which are suspended |

▶ Outputs:

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


```
results = {
  "content": [
    {
      "hasIncident": false,
      "pid": 0,
      "ppid": 0,
      "processName": "[System Process]",
      "startTime": 0,
      "suspended": false
    },
    {
      "hasIncident": false,
      "pid": 4,
      "ppid": 0,
      "processName": "System",
      "startTime": 1644961331131,
      "suspended": false
    },
    {
      "hasIncident": false,
      "pid": 92,
      "ppid": 4,
      "privilegeLevel": "SYSTEM",
      "processName": "Registry",
      "startTime": 1644961322746,
      "suspended": false,
      "user": "NT AUTHORITY\\SYSTEM"
    },
    {
      "hasIncident": false,
      "pid": 436,
      "ppid": 4,
      "privilegeLevel": "SYSTEM",
      "processName": "smss.exe",
      "programPath": "C:\\Windows\\System32\\smss.exe",
      "startTime": 1644961331143,
      "suspended": false,
      "user": "NT AUTHORITY\\SYSTEM"
    },
```

```
    {
      "hasIncident": false,
      "pid": 556,
      "ppid": 544,
      "privilegeLevel": "SYSTEM",
      "processName": "csrss.exe",
      "programPath": "C:\\Windows\\System32\\csrss.exe",
      "startTime": 1644961338892,
      "suspended": false,
      "user": "NT AUTHORITY\\SYSTEM"
    },
    {
      "hasIncident": false,
      "pid": 628,
      "ppid": 544,
      "privilegeLevel": "SYSTEM",
      "processName": "wininit.exe",
      "programPath": "C:\\Windows\\System32\\wininit.exe",
      "startTime": 1644961339001,
      "suspended": false,
      "user": "NT AUTHORITY\\SYSTEM"
    },
    {
      "hasIncident": true,
      "pid": 6712,
      "ppid": 884,
      "privilegeLevel": "LOW",
      "processName": "Microsoft.Photos.exe",
      "programPath": "C:\\Program
Files\\WindowsApps\\Microsoft.Windows.Photos_2021.21090.10008.0_x64__8weky
b3d8bbwe\\Microsoft.Photos.exe",
      "startTime": 1645130455541,
      "suspended": true,
      "user": "AUTISTIC1\\Administrator"
    },
    {
      "hasIncident": true,
      "pid": 6556,
      "ppid": 884,
      "privilegeLevel": "HIGH",
      "processName": "RuntimeBroker.exe",
      "programPath": "C:\\Windows\\System32\\RuntimeBroker.exe",
      "startTime": 1645130459641,
      "suspended": false,
      "user": "AUTISTIC1\\Administrator"
    },
    {
      "hasIncident": true,
      "pid": 1416,
      "ppid": 884,
      "privilegeLevel": "LOW",
      "processName": "TextInputHost.exe",
      "programPath":
"C:\\Windows\\SystemApps\\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\\TextI
```

```
nputHost.exe",
      "startTime": 1645130848211,
      "suspended": false,
      "user": "AUTISTIC1\\Administrator"
    },
    {
      "hasIncident": false,
      "pid": 7904,
      "ppid": 4772,
      "privilegeLevel": "HIGH",
      "processName": "chrome.exe",
      "programPath": "C:\\Program Files
(x86)\\Google\\Chrome\\Application\\chrome.exe",
      "startTime": 1645130852284,
      "suspended": false,
      "user": "AUTISTIC1\\Administrator"
    }
  ],
  "inputs": {
    "reaqta_endpoint_id": "831986736375529472",
    "reaqta_has_incident": null,
    "reaqta_suspended": null
  },
  "metrics": {
    "execution_time_ms": 953,
    "host": "endpoint.local",
    "package": "fn-reaqta",
    "package_version": "1.0.0",
    "timestamp": "2022-02-18 13:47:22",
    "version": "1.0"
  },
  "raw": null,
  "reason": null,
  "success": true,
  "version": 2.0
}
```


▶ Example Pre-Process Script:

```
inputs.reaqta_endpoint_id = incident.properties.reaqta_endpoint_id
inputs.reaqta_has_incident = rule.properties.reaqta_has_incident
inputs.reaqta_suspended = rule.properties.reaqta_suspended
```

▶ Example Post-Process Script:

```
import java.util.Date as Date
now = Date().getTime()

if results.success:
```

```
  for process in results.content:
    row = incident.addRow("reaqta_process_list")

    row['report_date'] = now
    row["pid"] = process.get("pid")
    row["process_name"] = process.get("processName")
    row["process_path"] = process.get("programPath")
    row["privilege_level"] = process.get("privilegeLevel")
    row["user"] = process.get("user")
    row["has_incident"] = process.get("hasIncident")
    row["suspended"] = process.get("suspended")
    row["start_time"] = process.get("startTime")
else:
  incident.addNote("ReaQta Get Processes failed:
{}".format(results.reason))
```

---

## Function - ReaQta: Isolate Machine

Isolate a ReaQta controlled machine based on it's endpoint ID.

▶ Inputs:

| Name | Type | Required | Example | Tooltip |
|---|---|---|---|---|
| reaqta_endpoint_id | text | Yes | - | - |

▶ Outputs:

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


```
results = {
  "content": {
    "details": {
      "endpointId": "820358261151629312",
      "lastSeenAt": "2022-02-15T18:05:09.113Z"
    },
    "message": "Endpoint offline"
  },
  "inputs": {
    "reaqta_endpoint_id": "820358261151629312"
  },
  "metrics": {
    "execution_time_ms": 498,
    "host": "endpoint.local",
    "package": "fn-reaqta",
    "package_version": "1.0.0",
    "timestamp": "2022-02-18 14:04:55",
    "version": "1.0"
  },
  "raw": null,
```

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

▶ Example Pre-Process Script:

```
inputs.reaqta_endpoint_id = incident.properties.reaqta_endpoint_id
```

▶ Example Post-Process Script:

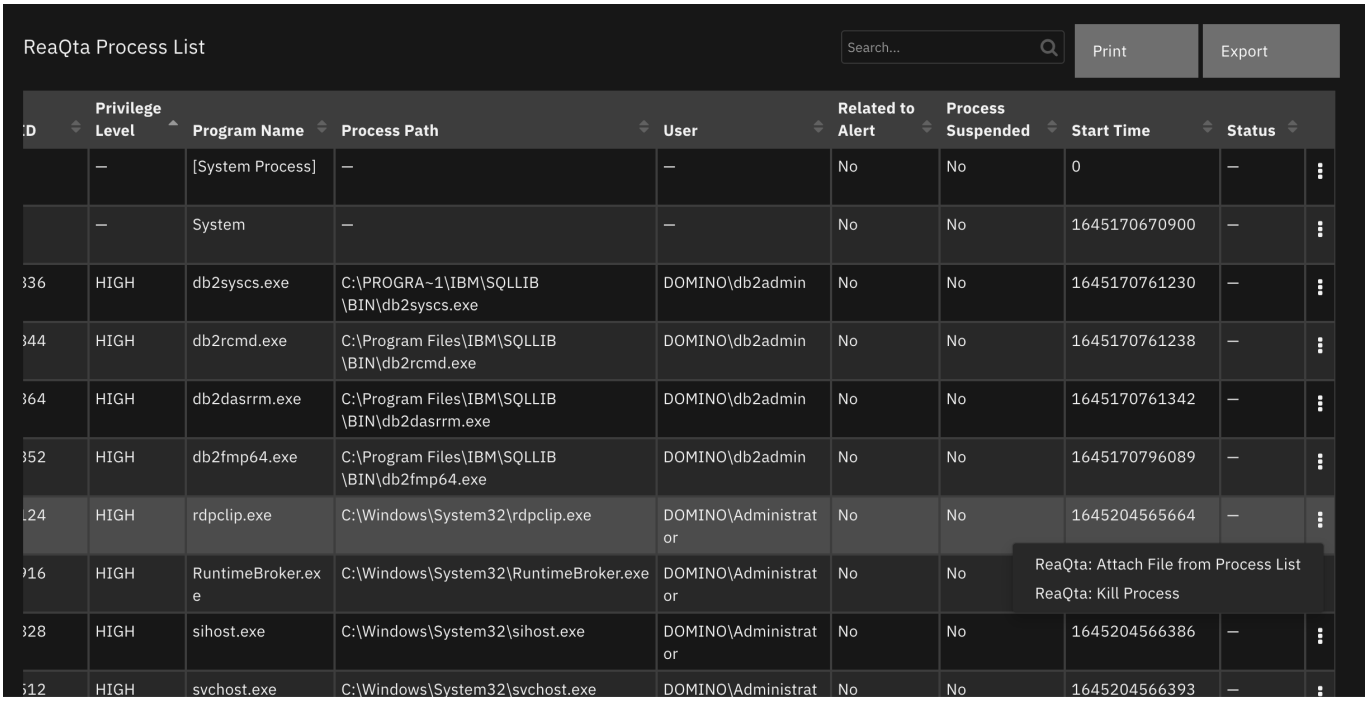
```
if results.success and results.content.get('success'):
  msg = "Endpoint Machine Isolated"
elif results.reason:
  msg = "ReaQta Isolate Machine failed: {}".format(results.reason)
else:
  msg = "ReaQta Isolate Machine failed:
{}".format(results.content.get('message'))

incident.addNote(msg)
```

---

## Function - ReaQta: Kill Process

Kill a process on a machine by the process PID. The process list datatable row is updates to show the kill status.




▶ Inputs:

| Name | Type | Required | Example | Tooltip |
|---|---|---|---|---|
| reaqta_endpoint_id | text | Yes | - | - |
| reaqta_process_pid | number | Yes | - | Collected from the reaQta_process_list datatable. |
| reaqta_starttime | number | Yes | - | Collected from the reaQta_process_list datatable. |

▶ Outputs:

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


```
results = {
  "content": [
    {
      "error": "Process not found",
      "errorCode": -1,
      "killed": false,
      "pid": 4,
      "startTime": 1644961331131
    }
  ],
  "inputs": {
    "reaqta_endpoint_id": "831986736375529472",
    "reaqta_process_pid": 4,
    "reaqta_starttime": 1644961331131
  },
  "metrics": {
    "execution_time_ms": 953,
    "host": "endpoint.local",
    "package": "fn-reaqta",
    "package_version": "1.0.0",
    "timestamp": "2022-02-18 13:48:08",
    "version": "1.0"
  },
  "raw": null,
  "reason": null,
  "success": true,
  "version": 2.0
}
```

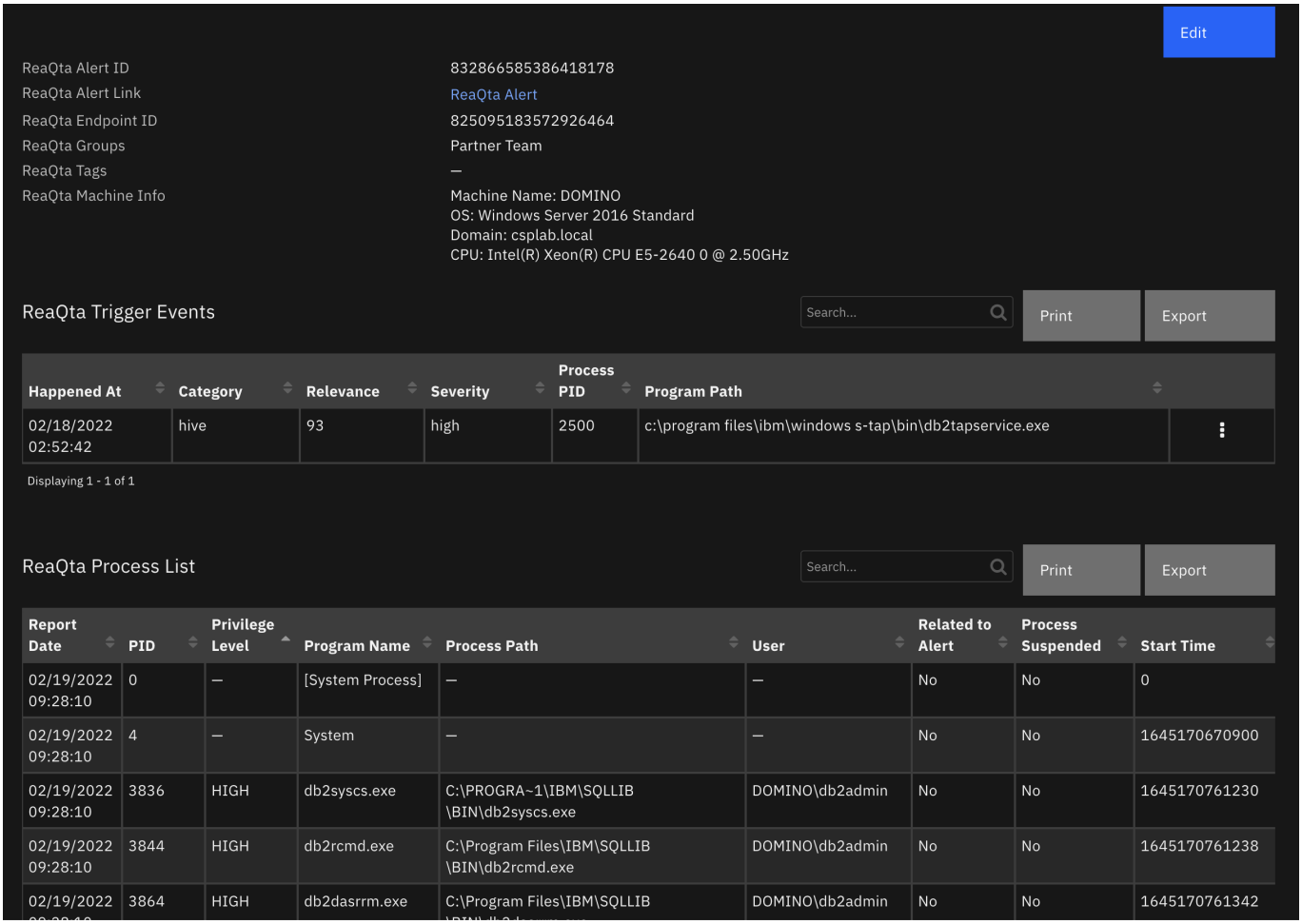

▶ Example Pre-Process Script:

```
None
```

▶ Example Post-Process Script:

None

## Data Table - ReaQta Process List




**API Name:**

reaqta_process_list

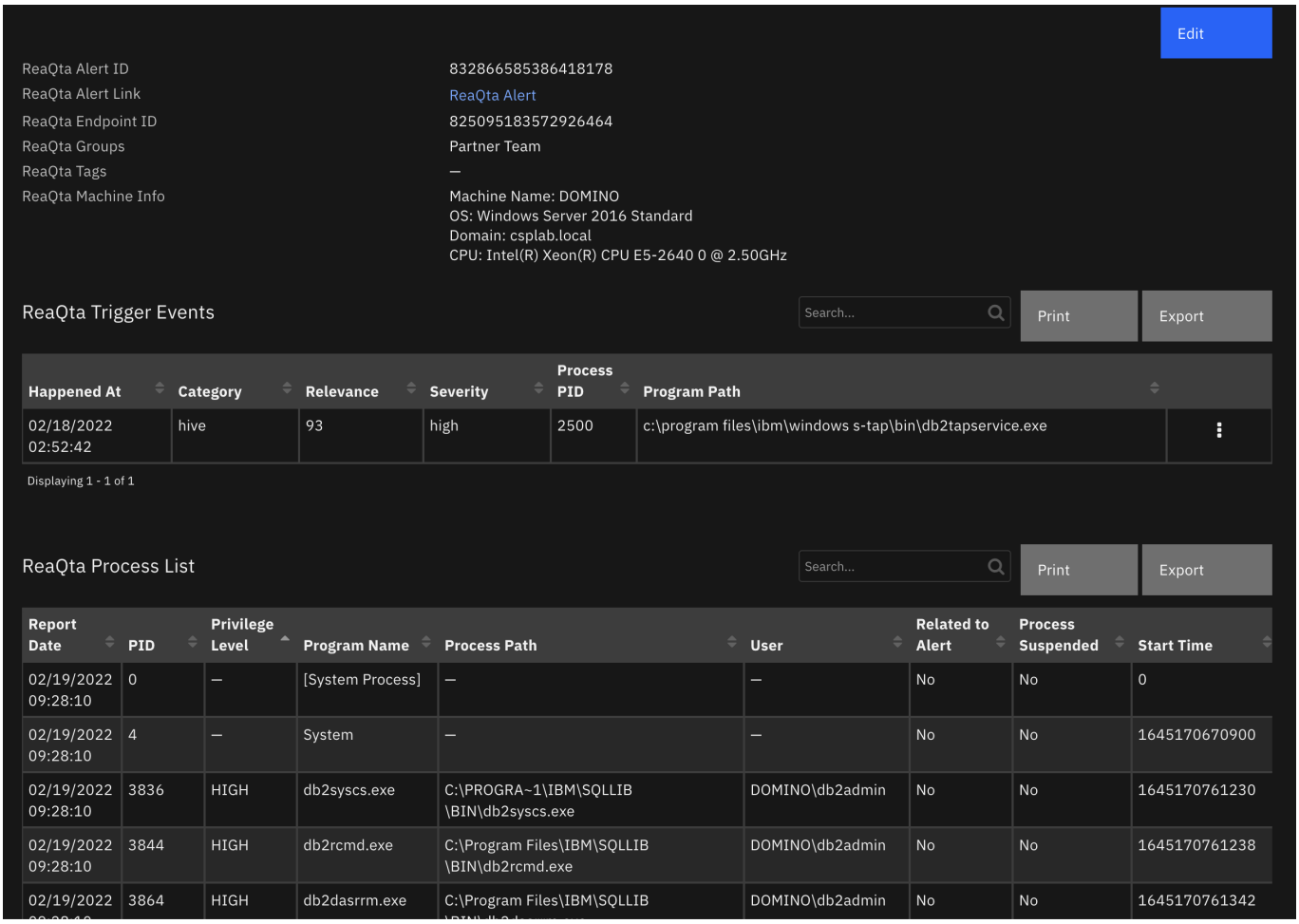
**Columns:**

| Column Name | API Access Name | Type | Tooltip |
|---|---|---|---|
| Report Date | `report_date` | `datetimepicker` | - |
| PID | `pid` | `number` | process ID |
| Privilege Level | `privilege_level` | `text` | - |
| User | `user` | `text` | - |
| Program Name | `process_name` | `text` | - |
| Process Path | `process_path` | `text` | Path to file on endpoint filesystem |

| Column Name | API Access Name | Type | Tooltip |
|---|---|---|---|
| Process Suspended | `suspended` | `boolean` | - |
| Related to Alert | `has_incident` | `boolean` | - |
| Start Time | `start_time` | `number` | - |
| Status | `status` | `text` | Used to identify processes killed |

## Data Table - ReaQta Trigger Events

This table is created when the ReaQta alert is synchronized, showing the specific file events associated with the alert creation.




**API Name:**

reaqta_trigger_events

**Columns:**

| Column Name | API Access Name | Type | Tooltip |
|---|---|---|---|
| Happened At | `happened_at` | `datetimepicker` | - |
| Category | `category` | `text` | - |

| Column Name | API Access Name | Type | Tooltip |
|---|---|---|---|
| Process PID | process_pid | number | - |
| Program Path | program_path | text | - |
| Relevance | relevance | number | - |
| Severity | severity | text | - |

## Custom Fields

| Label | API Access Name | Type | Prefix | Placeholder | Tooltip |
|---|---|---|---|---|---|
| ReaQta Alert Link | reaqta_alert_link | textarea | properties | - | Link back to the Alert for further review |
| ReaQta Alert ID | reaqta_id | text | properties | - | Used for further synchronization with the alert |
| ReaQta Endpoint ID | reaqta_endpoint_id | text | properties | - | Used for further synchronization with the endpoint |
| ReaQta Groups | reaqta_groups | text | properties | - | |
| ReaQta Tags | reaqta_tags | text | properties | - | - |
| ReaQta Machine Info | reaqta_machine_info | textarea | properties | - | - |

## Rules

| Rule Name | Object | Workflow Triggered |
|---|---|---|
| ReaQta: Attach File from Process List | reaqta_process_list | reaqta_attach_file_from_process_list |
| ReaQta: Attach File from Triggered Events | reaqta_trigger_events | reaqta_attach_file_from_triggered_events |
| ReaQta: Close Alert | incident | reaqta_close_alert |
| ReaQta: Create Note | note | reaqta_create_note |

| Rule Name | Object | Workflow Triggered |
| --- | --- | --- |
| ReaQta: Get Processes | incident | `reaqta_get_processes` |
| ReaQta: Isolate Endpoint | incident | `reaqta_isolate_endpoint` |

## 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.